N° d'ordre : 10/2014-D/INF

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et d'Informatique

THESE

Présentée pour l'obtention du grade de DOCTEUR

En : INFORMATIQUE

Spécialité : Informatique Mobile

Par : SENOUCI Mustapha Reda

Sujet :

APPROCHES AUTONOMES POUR LE DEPLOIEMENT ET LE CONTROLE DE PLUSIEURS ENTITES COMMUNICANTES DANS UN ENVIRONNEMENT CONTRAINT

Soutenue publiquement, le 25 Janvier 2014, devant le jury composé de :

M. N. BADACHE, Professeur à l'USTHB	Président
M. A. AISSANI, Professeur à l'USTHB	Directeur de thèse
M. A. MELLOUK, Professeur à l'UPEC - France	Co-directeur
M. J-M. THIRIET, Professeur à l'UJF/GIPSA-Lab - France	Examinateur
M. A. APPRIOU, Directeur de Recherche à l'ONERA - France	Examinateur
M. M. BENCHAIBA, Maître de Conférences/A à l'USTHB	Examinateur
Mme L. OUKHELLOU, Directrice de Recherche à l'UPE - France	Invitée

Titre : Approches autonomes pour le déploiement et le contrôle de plusieurs entités communicantes dans un environnement contraint.

Résumé : Cette thèse porte sur les problèmes de déploiement des Réseaux de Capteurs sans Fil (RCsF) et est centrée autour de trois directions principales : (1) le déploiement aléatoire, (2) le déploiement déterministe, et (3) l'auto-déploiement. En premier lieu, nous présentons une étude sur le placement aléatoire des capteurs et nous élaborons une stratégie pratique de déploiement aléatoire. Ensuite, nous analysons le problème de gestion des imperfections liées à la collecte des données par les capteurs dans le cas d'un déploiement déterministe. Nous discutons les modèles de couverture et les algorithmes de placement existants et nous exploitons la théorie de l'évidence pour concevoir des stratégies de déploiement plus efficaces. Enfin, nous explorons les stratégies d'auto-déploiement existantes et nous présentons un protocole en deux phases, léger et complet, pour assurer une couverture optimisée de la zone contrôlée en utilisant un RCsF mobile. Les résultats obtenus montrent l'efficacité des approches proposées qui ont été étudiées et validées à la fois sur des données synthétiques et aussi sur un banc d'essai expérimental.

Mots-clés : Réseaux de Capteurs sans Fil, Déploiement, Couverture, Connectivité, Banc d'essai, Fusion, Fonctions de Croyance, Modèle des Croyances Transférables.

Title: Autonomous approaches for the deployment and control of several communicating entities in a constrained environment.

Abstract: This dissertation is an in-depth investigation of the Wireless Sensor Networks (WSNs) deployment problems that follows three general directions: (1) random deployment, (2) deterministic deployment, and (3) self-deployment. First, we present a survey and taxonomy of random node placement and we devise a practical random deployment strategy. Second, we analyze the uncertainty-aware deterministic WSNs deployment problem where sensors may not always provide reliable information. We discuss sensor coverage models and placement algorithms found in the literature and we investigate the evidence theory to design better deployment strategies. We devise evidence-based sensor coverage models and we propose several polynomial-time uncertainty-aware deployment algorithms. Third, we explore the published self-deployment strategies and we devise a lightweight and comprehensive two-phase protocol, for ensuring area coverage employing a mobile WSN. Experimental results based on synthetic data sets, data traces collected in a real deployment, and an experimental testbed, show that the proposed approaches outperform the state-of-the-art deployment strategies.

Keywords: Wireless Sensor Networks, Deployment, Coverage, Connectivity, Testbed, Fusion, Belief Functions, Transferable Belief Model.

Table of Contents

Table of Contents i

Acknowledgements xi

Dedication xii

Publications personnelles faites dans le cadre de la thèse xiii

1 Introduction 1
- 1.1 Wireless Sensor Networks Overview . . . 1
  - 1.1.1 Sensor nodes . . . 2
  - 1.1.2 Wireless Sensor Networks . . . 3
- 1.2 WSNs Deployment . . . 7
  - 1.2.1 Deployment Strategies . . . 7
  - 1.2.2 Objectives and Constraints . . . 8
- 1.3 Problem Statement . . . 12
- 1.4 Dissertation Contributions . . . 14
- 1.5 Dissertation Organization . . . 16

2 Literature Review & Related Work 17
- 2.1 Introduction . . . 17
- 2.2 Random Deployment . . . 18
  - 2.2.1 Simple Random Node Placement Strategies . . . 19
  - 2.2.2 Compound Random Node Placement Strategies . . . 21
  - 2.2.3 Related work & Discussion . . . 24
- 2.3 Deterministic Deployment . . . 26
  - 2.3.1 Sensor Coverage Models . . . 26
  - 2.3.2 Placement Algorithms . . . 31
- 2.4 Self-Deployment . . . 38
  - 2.4.1 Virtual forces based approach . . . 39
  - 2.4.2 Coverage pattern based approach . . . 41
  - 2.4.3 Grid quorum based approach . . . 42

2.4.4 Computational geometry based approach . . . 43
2.4.5 Fuzzy based approach . . . 44
2.4.6 Optimization based approach . . . 45
2.4.7 Miscellaneous approaches . . . 46
2.4.8 A comparison between the various approaches . . . 46
2.5 Conclusion . . . 48

3 Random Deployment of WSNs 49
3.1 Introduction . . . 49
3.2 Simulation Settings . . . 50
3.3 Deployment Metrics Analysis . . . 51
3.3.1 Coverage Analysis . . . 51
3.3.2 Connectivity Analysis . . . 53
3.4 Functional Metrics Analysis . . . 54
3.4.1 Routing-related Metrics Analysis . . . 54
3.4.2 Fault-Tolerance Analysis . . . 55
3.4.3 Network Lifespan Analysis . . . 57
3.5 A Practical Random Deployment Strategy . . . 59
3.5.1 Deployment Metrics Analysis . . . 61
3.5.2 Functional Metrics Analysis . . . 62
3.6 Conclusion . . . 65

4 Uncertainty-aware WSNs Deployment 67
4.1 Introduction . . . 67
4.2 Evidence-based Sensor Coverage Model . . . 70
4.2.1 Evidence construction . . . 70
4.2.2 Decision Making . . . 72
4.2.3 Handling Deployment-related Issues . . . 72
4.3 Application: Evidence-based Detection Coverage Model . . . 74
4.3.1 Evidence Combination . . . 74
4.3.2 Decision making . . . 76
4.4 Problem Formalization . . . 76
4.4.1 Network Model and Assumptions . . . 77
4.4.2 Uniform Coverage and Cost Objectives . . . 78
4.4.3 Preferential Coverage and Cost Objectives . . . 78
4.4.4 Uniform Coverage, Cost and Connectivity Objectives . . . 78
4.4.5 Preferential Coverage, Cost and Connectivity Objectives . . . 79
4.5 The Proposed Sensor Placement Algorithms . . . 79
4.5.1 EBDA . . . 80
4.5.2 S-E2BDA . . . 81
4.5.3 E2BDA-P . . . 83
4.5.4 E2BDA-C . . . 83

4.5.5 E2BDA . . . 84
4.6 Performance Evaluation . . . 84
4.6.1 The Effects of Different Parameters on the System Performance . . . 85
4.6.2 EBDA *vs.* Random . . . 86
4.6.3 EBDA *vs.* MIN_MISS: Numerical Experiments . . . 89
4.6.4 EBDA *vs.* MIN_MISS: Trace-driven Simulations . . . 91
4.6.5 E2BDA *vs.* State of the Art Deployment Strategies . . . 93
4.7 Conclusion . . . 97

5 Uncertainty-aware Fusion-based WSNs Deployment 98
5.1 Introduction . . . 98
5.2 Evidence Fusion Model . . . 99
5.2.1 Evidence-based Sensing Model . . . 100
5.2.2 Building Belief Functions . . . 103
5.3 Discussion . . . 104
5.3.1 Handling Sensors Reliability . . . 105
5.3.2 Handling Harsh Deployment Environments . . . 105
5.4 Probabilistic Connectivity and Communication Cost Models . . . 106
5.5 Sensor Placement Problems . . . 109
5.5.1 Network Model and Assumptions . . . 109
5.5.2 Problems Formulation . . . 109
5.6 Sensor Placement Algorithms . . . 112
5.6.1 GEBDA . . . 113
5.6.2 MGEBDA . . . 118
5.7 Performance Evaluation . . . 122
5.7.1 GEBDA: Numerical Results . . . 122
5.7.2 GEBDA: Trace-driven Simulations . . . 128
5.7.3 MGEBDA . . . 130
5.8 Conclusion . . . 135

6 Design and Implementation of a Surveillance WSN Testbed 136
6.1 Introduction . . . 136
6.2 Hardware and Software . . . 136
6.2.1 Hardware . . . 137
6.2.2 Software . . . 140
6.3 System Architecture . . . 141
6.4 Experiments and Results . . . 141
6.5 Coverage Models Analysis . . . 142
6.5.1 Building Belief Functions From Raw Sensory Data . . . 142
6.5.2 Sensing Performance of the Phidgets Sensor . . . 143
6.6 Connectivity Model Analysis . . . 143
6.6.1 Regions of Connectivity . . . 144

6.6.2 Calibrating the Communication Cost Model . . . 145
6.7 EBDA *vs.* MIN_MISS . . . 145
6.8 GEBDA . . . 146
6.9 Conclusion . . . 147

7 General Conclusion and Perspectives **148**
7.1 Summary of Contributions . . . 149
7.2 On-going Works . . . 150
7.2.1 Dynamic WSNs Deployment . . . 150
7.2.2 Post-deployment Strategies . . . 152
7.3 Future Research Directions . . . 153

8 Version Française Abrégée **155**
8.1 Introduction . . . 155
8.2 Etude des propriétés intrinsèques du déploiement aléatoire des RCsF . . . 157
8.3 Proposition d'une stratégie pratique de déploiement aléatoire . . . 158
8.4 Modèles de couverture basés sur les évidences . . . 159
8.5 Plateformes prenant en compte l'incertitude pour le déploiement des RCsF . 160
8.6 Déploiement autonome des capteurs dans un RCsF mobile . . . 161
8.7 Les approches de post-déploiement . . . 162
8.8 Conclusion . . . 163

A The Belief Functions Theory Overview **164**
A.1 Introduction . . . 164
A.2 Overview of the TBM . . . 164
A.2.1 Representation of Information . . . 164
A.2.2 Handling Information . . . 167
A.2.3 Decision Making . . . 169

Bibliography **170**

List of Figures

1.1 A sensor node. 2
1.2 A typical WSN architecture. 3
1.3 WSNs scenarios. 5

2.1 Random node placement strategies taxonomy. 18
2.2 An example of simple diffusion. 19
2.3 An example of continuous diffusion. 20
2.4 An example of discontinuous diffusion (5 throws). 21
2.5 An example of discontinuous diffusion (10 throws). 21
2.6 An example of constant diffusion. 22
2.7 An example of R-random. 23
2.8 An example of Exponential diffusion. 24
2.9 An example of Stensor placement [102]. 24
2.10 A taxonomy for sensor coverage models. 27
2.11 Illustration of the disk coverage model. 28
2.12 Illustration of the relation between the coverage measure and the sensor-point distance for (a) Binary model; (b) Probabilistic model; and (c) Truncated probabilistic model. 29
2.13 A taxonomy for deterministic deployment strategies. 32
2.14 A taxonomy for self-deployment strategies. 39
2.15 An example of virtual forces. 40
2.16 Commonly used coverage patterns. 41
2.17 Examples of grid quorum based approach. 43
2.18 An example of voronoi diagram. 44

3.1 Achieved k-coverage by different strategies. 51
3.2 Achieved connectivity by different strategies. 53

3.3 Fault-tolerance related to detection errors. 56
3.4 Network lifespan for different strategies. 58
3.5 An example of the hybrid diffusion. 60
3.6 Achieved coverage by different strategies. 61
3.7 Achieved connectivity by different strategies. 62
3.8 Fault-tolerance related to detection errors. 62
3.9 Network lifespan based on the detection rate. 63
3.10 Network lifespan based on the number of alive sensors. 64
3.11 Network lifespan based on coverage. 65
3.12 Network lifespan based on connectivity. 65

4.1 Detection probability *vs.* the sensor-vehicle distance. 68
4.2 Evidence-based sensor coverage model. 70
4.3 2D RoI model. 77
4.4 The effects of R_s, R_u and Th. 86
4.5 Required number of sensors *vs.* sensor reliability. 86
4.6 EBDA *vs.* Random in terms of convergence. 87
4.7 EBDA *vs.* Random using different thresholds. 88
4.8 EBDA *vs.* Random in terms of cost. 89
4.9 EBDA *vs.* Random in terms of achieved coverage. 90
4.10 EBDA *vs.* MIN_MISS using different thresholds. 90
4.11 EBDA *vs.* MIN_MISS in terms of deployment cost ($Th = 0.9$). 91
4.12 Example of a set of mass functions associated to a WINS NG 2.0 sensor in the case of AAV vehicle detection. 92
4.13 EBDA *vs.* MIN_MISS in terms of achieved coverage. 93
4.14 Sensor placements using EBDA and MIN_MISS (RoI=$400m \times 400m$ and $Th = 0.9$). 94
4.15 Preferential coverage requirements. 96

5.1 The false alarm rate *vs.* the number of sensors. 103
5.2 Example of a set of mass functions associated to a WINS NG 2.0 sensor in the case of AAV vehicle detection. 104
5.3 Channel Model, $\gamma = 2$, $\sigma_\epsilon = 4$, $P_t = 0dBm$. 107
5.4 Connectivity model. 108
5.5 Binary representation of a sensor placement in GEBDA. 114

5.6 Crossover techniques used in GEBDA. 116
5.7 Binary representation of a sensor placement in MGEBDA. 119
5.8 Crossover techniques used in MGEBDA. 120
5.9 The number of sensors *vs.* fusion radius. 123
5.10 Results for different population size. 125
5.11 L-mapping *vs.* H-mapping. 126
5.12 Pareto frontier for MGEBDA with different parameter settings. 131
5.13 The impact of fusion radius on performance. 132
5.14 The impact of fusion radius on the generated topologies. 133

6.1 The Arduino Uno board. 137
6.2 The Arduino WiFi (left) and the Wireless Proto Shields (right). 138
6.3 The XBee 802.15.4 (left), the XBee ZB (center), and the Bluetooth Bee (right). 138
6.4 Static node. 139
6.5 Mobile node. 140
6.6 The Arduino IDE running in Ubuntu. 140
6.7 System architecture. 141
6.8 Example of a set of mass functions associated to a PIR Phidgets 1111_0 sensor in an indoor environment. 142
6.9 Example of a set of mass functions associated to a PIR Phidgets 1111_0 sensor in an outdoor environment. 143
6.10 Detection probability *vs.* the sensor-target distance. 144
6.11 Regions of Connectivity. 145
6.12 Measured P_r *vs.* theoretical P_r 146

7.1 Principle of the healing process. 151

8.1 Schéma général d'un RCsF. 155
8.2 Déploiement aléatoire *vs.* déploiement déterministe. 156
8.3 Taxonomie des stratégies de déploiement aléatoire des RCsF. 157
8.4 Un exemple de la diffusion hybride de 300 capteurs dans une zone circulaire de rayon $R = 150m$. 158
8.5 Détection de cible/évènement dans le cadre de la théorie des fonctions de croyance. 160

8.6 Différents angles d'études pour le problème du déploiement des RCsF. . . . 163

List of Tables

2.1 A comparison between the various probabilistic deployment approaches. . . 36
2.2 A comparison between the various fusion-based deployment approaches. . . 37
2.3 A comparison between the various self-deployment approaches. 47

3.1 The second scenario parameters. 51
3.2 Summary of first stage simulations results. 54
3.3 Routing-related simulations results. 55
3.4 The differences due to different errors. 57
3.5 Summary of second stage simulations results. 59
3.6 The differences due to different errors. 63

4.1 Simulations parameters 87
4.2 Simulations parameters 94
4.3 Cost of full uniform coverage 95
4.4 Cost of full preferential coverage 96
4.5 Cost of coverage and connectivity 97

5.1 Results for different crossover probabilities 124
5.2 Results for different evolution engines 125
5.3 GEBDA *vs.* Complete Enumeration 127
5.4 GEBDA *vs.* E2BDA 128
5.5 Trace-driven simulations results 129
5.6 Simulations Parameters 130
5.7 The effect of the requested connectivity reliability. 134
5.8 MGEBDA *vs.* GEBDA. 135

6.1 Experimental Results 146
6.2 Experimental Results 147

A.1 Examples of mass functions in the FoD $\Theta = \{\theta_1, \theta_2, \theta_3\}$ 166
A.2 The connection between mass, belief, and plausibility functions 167

Acknowledgements

"He who does not thank people, does not thank Allah."

The Messenger of Allah (peace be upon him).

First of all, I humbly thank **Allah** (God) Almighty, the Merciful and the Beneficent, who blessed me with knowledge, health, thoughts and cooperative people to enable me achieve this research. My most sincere gratitude to all the people that made this dissertation possible.

I would like to express my deep gratitude to Prof. Abdelhamid Mellouk and Prof. Amar Aissani, my research supervisors, for their patient guidance, enthusiastic encouragement and useful critiques of this research work. Thanks go again to Prof. Abdelhamid Mellouk for being much more than an academic adviser, for being a mentor and a friend.

I am extremely grateful to my research advisers, Dr. Latifa Oukhellou and Dr. Nadjib Aitsaadi, for their valuable advice and assistance.

I am obligated to all the members of my jury for accepting to evaluate my work. Thanks go to Prof. Nadjib Badache, Prof. Jean-Marc Thiriet, Dr. Alain Appriou, and Dr. Mahfod Benchaiba.

Many thanks to all the colleagues and friends at the LiSSi Laboratory and the Ecole Militaire Polytechnique, especially those of the UERI & CECA, for their always friendly help and valuable discussions.

The author is grateful to the many anonymous reviewers that with their comments greatly improved the content of the papers from which this thesis has been partly extracted.

Last but not least, I am extremely grateful to my family for their support, and for all the little things that made me be who and what I am, and allowed me to achieve all this.

Dedication

To my wife and my daughters.

Publications personnelles faites dans le cadre de la thèse

Articles dans des revues internationales ou nationales avec comité de lecture répertoriées dans JCR (ISI Web of Science) :

1. M. R. Senouci, A. Mellouk, K. Assnoune: Localized Movement-Assisted Sensor Deployment Algorithm For Hole Detection and Healing. IEEE Transactions on Parallel and Distributed Systems, to appear 2013.
2. M. R. Senouci, A. Mellouk, A. Aissani: Random Deployment of Wireless Sensor Networks: A Survey and Approach. International Journal of Ad Hoc and Ubiquitous Computing, Indersciences Publishing Ed., to appear, 2013.
3. M. R. Senouci, Abdelhamid Mellouk, Latifa Oukhellou, Amar Aissani: An Evidence-Based Sensor Coverage Model. IEEE Communications Letters, vol. 16, no. 9, pp. 1462-1465, 2012.
4. M. R. Senouci, A. Mellouk, H. Senouci, A. Aissani: Performance Evaluation of Network Lifetime Spatial-Temporal Distribution for WSN Routing Protocols. Journal of Network and Computer Applications, Elsevier, vol. 35, no 4, pp. 1317-1328, 2012.
5. M. Aissani, A. Mellouk, W. Demigha, M. R. Senouci: Link Failure Resilience in the Dynamic Source Routing Protocol. Wireless Communications and Mobile Computing, Wiley, vol. 9, no. 5, pp. 591-605, 2009.

Communications avec actes dans un congrès international :

1. M. R. Senouci, A. Mellouk, A. Aissani, "An analysis of intrinsic properties of stochasticnode placement in sensor networks", in Proc. Of the IEEE Global Communications Conference (GLOBECOM), Anaheim - California, USA, December 3-7, 2012, pp. 494-499.

2. M. R. Senouci, A. Mellouk, L. Oukhellou, A. Aissani, "Efficient uncertainty-aware deployment algorithms for wireless sensor networks", in Proc. Of the IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, April 10-14, 2012, pp. 2163-2167.
3. M. R. Senouci, A. Mellouk, L. Oukhellou, A. Aissani, "Uncertainty-Aware Sensor Network Deployment", in Proc. Of the IEEE Global Communications Conference (GLOBECOM), Houston, USA, December 5-9, 2011, pp. 1-5.
4. M. R. Senouci, A. Mellouk, A. Aissani, H. Senouci, "Performance Evaluation of the Network Lifetime for Routing Protocols in WSN", in Proc. Of the IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), Istanbul, Turkya, July 5-8, 2011, pp. 1-6.

Chapitres dans des recueils de communications de conférences internationales :

1. M. R. Senouci, A. Mellouk, L. Oukhellou, A. Aissani, "Using the Belief Functions Theory to Deploy Static Wireless Sensor Networks", in Springer Lecture Notes in Advances in Intelligent and Soft Computing -Belief Functions: Theory and Applications, vol. 164, 2012, pp. 425-432. 2nd Int. Conf. on Belief Functions, Compiègne, France, May 9-11, 2012.

Conférences données à l'invitation du comité d'organisation dans un congrès national ou international :

1. L. Oukhellou, A. Mellouk, M. R. Senouci, Belief Functions: Applications in Telecommunication, CNRS ARC MACS Journey, Paris, May 22, 2012.

Communications dans un congrès national

1. M. R. Senouci, A. Mellouk, N. Aitsaadi, L. Oukhellou, Déploiement déterministe multi-objectif des réseaux de capteurs sans fil. Ecole de Recherche ResCom 2013, GDR ASR, Porquerolles, 13-17 Mai, 2013.
2. M. R. Senouci, A. Mellouk, Déploiement des réseaux de capteurs de surveillance à base de fusion : une approche basée sur le théorie des fonctions de croyance, 6e Journée du GT ARC, GDR MACS, Paris, 28 Mars, 2013.
3. M. Boudali, M. R. Senouci, M. Aissani Couverture de barrière pour la détection d'intrusion par réseau de capteurs sans fil, CGE'07, pp. 61-66, Algérie, 12-13 Avril, 2011.

4. M. R. Senouci, A. Mellouk, L. Oukhellou, A. Aissani, Uncertainty-aware sensor placement framework for surveillance applications, in RSACS'11, Algérie, Juin, 2011.

Travaux soumis :

1. M. R. Senouci, A. Mellouk, L. Oukhellou, A. Aissani, WSNs deployment framework based on the theory of belief functions. Submitted to IEEE/ACM Transactions on Networking.
2. M. R. Senouci, A. Mellouk, N. L. Oukhellou, Fusion-based WSN surveillance applications deployment: an evidence-based approach. Submitted to INFOCOM 2014.

Chapter 1

Introduction

Recent advances in Micro Electro-Mechanical Systems (MEMS) have enabled the development of small-scale, low-power, low-cost sensor nodes (also called sensors or motes) that integrate processing, storage, sensing, and communication capabilities. Such devices may be deployed in vast numbers over large geographical areas to form a Wireless Sensor Network (WSN).

The development of Wireless Sensor Networks (WSNs) was motivated by military applications such as battlefield surveillance; today such networks are used in many environmental, industrial, home, and medical applications. According to different application requirements, a WSN may consist of just a few or as many as thousands of sensors, operating in a collaborative and coherent manner to fulfill a common task.

In this Chapter, we provide a brief overview of WSNs, the primary topic of interest in this dissertation. We will first highlight the major characteristics of sensors and WSNs, followed by an introduction to typical WSNs scenarios and applications. We further explain the research problems explored in this dissertation, i.e., WSNs deployment problems, and their importance to WSNs applications. The dissertation contributions are also summarized briefly. In the end, the organization of this dissertation is outlined.

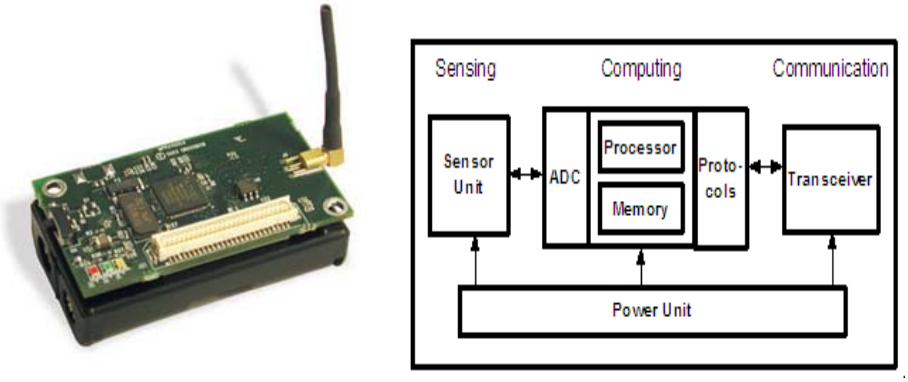
1.1 Wireless Sensor Networks Overview

Because WSNs are composed of sensor nodes, we first outline the typical architecture of a sensor node and then we highlight the major characteristics of WSNs.

1.1.1 Sensor nodes

Generally speaking, a sensor is a device which responds to physical stimulus (such as heat, light, sound, pressure, magnetism, etc.) and converts the quantity or parameter of a physical stimulus into recordable signals (such as electrical signals, mechanical signals, etc.). A sensor node (also called sensor or mote), which is the basic unit of a WSN, not only integrates sensors for sensing the physical world but also includes other units to process and deliver sensing data. Thus, a sensor generates data from sensing physical characteristics such as temperature, pressure and sound, and then transmits this data to a main location.

For better understanding of WSNs, it is important to know about all the components of sensor. A typical architecture of a sensor node is shown in Fig. 1.1(b), which consists of sensor unit, communication unit, microcontroller unit, memory and power unit. A sensor may also include some other units, depending on application scenarios and requirements, such as a locomotive unit that enables a sensor node to move around, an energy scavenging unit that obtains energy from the physical world, a GPS unit that acquires node geographical location, etc.



(a) Photograph of a MICAz sensor. (b) A typical sensor hardware architecture.

Figure 1.1: A sensor node.

A sensor node is implemented by a sensor board which integrates all the aforementioned components and other necessary circuitry. Fig. 1.1(a) presents a photograph of a MICAz sensor node by Crossbow Corporation [36]. The top layer of the node is the sensor board MPR2400TM [36], of 58 $mm \times 32\ mm \times 7\ mm$ large and of 18 g weight. The microcontroller is based on the Atmel ATmega128L, and three memory types are used, namely, program flash memory of 128 KB, configuration EEPROM 4 KB, and measurement flash 512 KB.

The sensor board can support many sensors types, including acoustic, light, magnetometer, and accelerometer sensors, and contains a 10-bit ADC. A MICAz node runs on 2.4 GHz at up to 250 $kbps$ data rate and conforms with IEEE 802.15.4 standard. The bottom is a battery case which can include two AA batteries. The current draw of the microcontroller in active mode is about 8 mA and in sleep mode is less than 15 μA. The current draw of the transceiver is 19.7 mA in receive mode, up to 17.4 mA in transmit mode, 20 μA in idle mode (with voltage regulator on), and 1 μA in sleep mode (with voltage regulator off). These tiny sensors leverage the idea of wireless sensor networks.

1.1.2 Wireless Sensor Networks

A Wireless Sensor Network (WSN) consists of a spatially distributed sensors and one or more *sinks* (or base stations). Sensors monitor environmental conditions, such as temperature, vibration, or motion and produce sensory data. A sink, on the other hand, collects data from sensors [13]. For example, in an event monitoring application, sensors are required to send data to the sink(s) when they detect unusual behavior in the network. A sink can be regarded as a gateway between a WSN and other networks, or an interface between a WSN and the people operating the WSN. A sink is often assumed as resource abundant, without energy supply limit and with advanced computation capability. Aside from data collection, a sink is also a central controller which can execute network management algorithms and instruct sensors only the computation results. Fig. 1.2 depicts a typical WSN architecture.

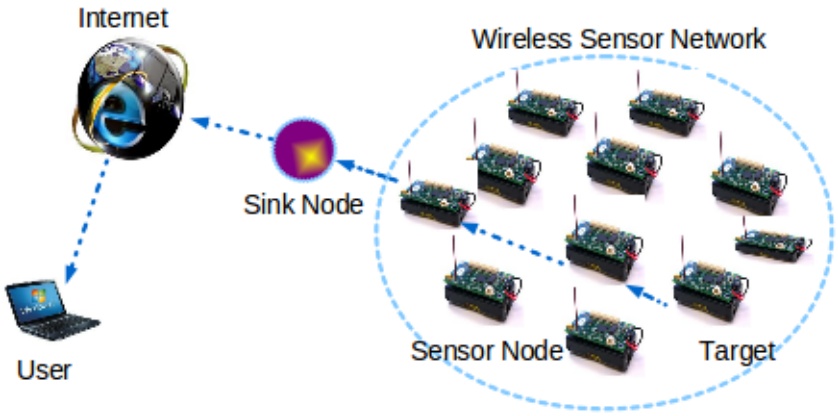


Figure 1.2: A typical WSN architecture.

Sensor Network Scenarios

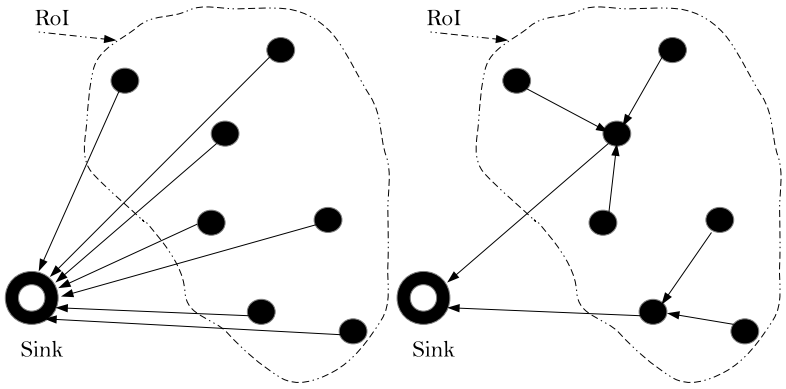
Sensors are usually deployed in a Region of Interest (RoI) to monitor some physical phenomena. Diverse scenarios exist in the architecture and management of WSNs. In what follows, we introduce several typical sensor network scenarios.

- **Homogeneous *vs.* Heterogeneous Networks** In a homogeneous WSN, all sensors have the same sensing, processing, communication, and other capabilities (excluding the initial power supply). In a heterogeneous WSN, sensors have different capabilities. For example, a node may have a stronger sensor unit and can cover a larger area.
- **Stationary *vs.* Mobile Networks** In a stationary WSN, all sensors are fixed and cannot move around after they have been deployed. A mobile sensor is equipped with a locomotive unit and can move around after deployment. In general, a mobile sensor is more expensive than its stationary compeer. It is usually considered in the literature that a mobile sensor network consists of only mobile nodes, while a hybrid sensor network is a network consisting of both stationary nodes and mobile nodes.
- **Single-Hop *vs.* Multi-Hop Networks** There are two basic communication paradigms between sensors and sinks, namely, *single-hop* communication and *multi-hop* communication. In a single-hop WSN, all sensors transmit directly to the sink as illustrated in Fig. 1.3(a). In a multi-hop WSN, some sensors use a multi-hop path consisting of other sensors as relays to deliver their data to the sink. A multi-hop WSN can be flat or hierarchical. In flat networks, all the nodes in the networks take the same responsibility, while in hierarchical networks the cluster-heads perform several special functions such as maintaining the clusters and aggregation. Fig. 1.3(b) illustrates a hierarchical multi-hop WSN. In a cluster, each cluster member transmits to its cluster head via either single-hop or multi-hop transmission.

Sensor Network Applications

As almost all kinds of physical phenomena can be sensed by different types of sensors, it is not surprising to find applications of WSNs in almost any domain. Application domains of WSNs include military, environmental, industrial, home, medical applications, etc. Numerous WSNs applications are expected to emerge in the near future. In what follows, we present a brief introduction and examples of WSNs applications.

- **Military Applications** WSNs, which can be rapidly scattered to critical terrains,



(a) A single-hop WSN. (b) A hierarchical multi-hop WSN.

Figure 1.3: WSNs scenarios.

routes, and straits, can provide battlefield intelligence, such as the location, numbers, and movement of troops and vehicles, and can be used to detect, classify, and track the trajectory of enemy vehicles. Some example military projects of WSNs include vehicles detection and tracking [18], sniper shoot detection and localization [101].

– **Environmental Applications** An important benefit of WSNs is that they enable non-intrusive and large-scale monitoring, which is critical to many environmental monitoring applications. There are lots of examples of projects using WSNs for environmental monitoring. For example, A WSN has been deployed in the Volcán Reventador Volcano for studying volcanoes' interior structure [131]. In [74], Mainwaring and colleagues present a real-life experiment of deploying a 32-node WSN to monitor the habitat of the nesting petrels on Great Duck Island.
– **Industrial Applications** WSNs can also find many industrial applications, such as manufacture automation, warehouses and products management, lighting control, the structural health monitoring, etc. Some example of industrial projects include the Pipenet project which uses a WSN to monitor public water supply pipes [107], the Wisden project which monitors the structural health [135], and the project that deploys a WSN to monitor the structural health for the Golden Gate Bridge [63].
– **Home Applications** WSNs can help to create a smart environment. An example of smart environment is the Smart kindergarten project [31], which uses WSNs to create a smart environment for early childhood education.

– **Medical Applications** Sensors can monitor body temperature, pulse rate, perspiration, oxygen and glucose level in blood, and other physical and biochemical parameters. These sensors can be wear in clothes or even implanted into human bodies. A wireless body sensor network integrates these wearable or implanted sensor nodes into a network and connects them with external networks for monitoring patients with chronic disease, hospital patients, and elderly patients [138]. An example project is the CodeBlue which develops a WSN for monitoring and tracking of patients [73].

Along with these *sense-only* scenarios, a new breed of applications emerged where some nodes are equipped with actuators. Wireless Sensor and Actuator Networks (WSANs) [14], refer to a group of sensors and actors linked by wireless medium to perform distributed sensing and acting tasks. These networks can be an integral part of systems such as battlefield surveillance, home automation and environmental monitoring. For example, in the case of home automation, motion and light sensors in a room can detect the presence of people and then command the appropriate actors to execute actions based on the pre-specified user preferences.

In this dissertation, we are interested in WSNs for detection applications. Detection refers to the process of making a yes/no decision regarding the existence of a certain phenomena of interest (e.g., target). In fact, in most WSNs applications, detection is usually the first step or even the sole goal of the system. For example, in tracking and classification systems, one cannot perform tracking nor classification without first asserting the presence of the target or phenomena of interest.

For all the applications mentioned above, the deployment of sensors is an important issue to deal with. In fact, deployment is of the utmost importance in the process of developing WSNs solutions for real-life applications as it decides the available resources and their configuration for system setup. This in turn plays a key role in the network performance. A significant amount of research has been made to enhance the network performance during its operational time by optimizing the WSNs protocol stack [13]. Nevertheless, even with the best WSNs protocol stack, deployed WSN cannot meet the desired performance goals unless it has been appropriately configured in advance. For instance, if the deployed sensors are insufficient or there are architecture deficiencies due to ineffective deployment plan, the network coverage could be insufficient as well as the network connectivity which will impair the WSNs desired functionalities. Accordingly, recent research efforts are moving toward enhancing the network performance by optimizing the network deployment [120, 121, 143].

1.2 WSNs Deployment

For the most part, WSNs are highly application-dependent. As a result, details such as node design, processing algorithms, network protocols, and deployment scheme are customized for the proposed application. Among these, deployment scheme, both deterministic and stochastic, is considered extremely important, since it directly influences fundamental efficiency metrics such as coverage, connectivity, cost, and lifetime. Delivery of large-scale WSN presents a major problem in the application of WSN technology, since large networks rely mainly on stochastic deployment schemes. Achieving acceptable network efficiency (e.g., connectivity, coverage, lifetime, cost, etc.) in both deterministic and stochastic deployment schemes is formally referred to as the deployment problem [13, 120, 143].

Before explaining the research problems explored in this dissertation, we first introduce some basic concepts and terminologies related to WSNs deployment.

1.2.1 Deployment Strategies

One of the fundamental design issues in WSNs is where to place the sensors in the RoI. The location of a sensor may affect the fulfillment of the system's requirements and multiple network performance metrics. Careful sensor placement can be a very effective optimization means for achieving the desired design goals. For example, the coverage objective is regarding how to ensure that each of the target points in the RoI is covered by the WSN. The sensors need to be placed not too close to each other so that the sensing capability of the network is fully utilized and at the same time they must not be located too far from each other to avoid the formation of coverage holes. A good deployment will enable a better performance on information gathering and communication.

Sensors can generally be placed in the RoI either deterministically or randomly. The choice of the deployment scheme depends highly on the type of sensors, application and the environment that the sensors will operate in [120, 121, 143]. Deterministic deployment schemes are optimal as sensors are placed at predetermined coordinates to guarantee the network efficiency; however they are impractical and sometimes impossible for large-scale WSN applications. For these applications, sensors may be deployed from a plane, delivered in an artillery shell, rocket or missile, or catapulted from a shipboard [13]. In these cases, the WSN has the utmost challenge of guaranteeing proper area coverage and connectivity upon deployment [13, 143]. Such cases require implementation of additional

complex protocols to ensure efficient network operation, which maximize network lifetime and decrease frequency of re-deployment.

Using mobile sensors, some deployment strategies have advocated dynamic adjustment of sensors' location which allows a dynamic deployment reconfiguration that improves the network performance. One of the major techniques is to schedule mobile sensors to move to the designated locations according to the result computed by the placement strategy such that the energy consumption due to movement is minimized. Another major approach is to schedule mobile sensors in order to cope with coverage hole issues.

A more in-depth discussion about all of the aforementioned deployment strategies is provided in Chapter 2.

1.2.2 Objectives and Constraints

WSNs deployment strategies should take into account the optimization of one or more objectives relating to the application needs under one or more constraints. In this section, we introduce the most important objectives and constraints generally considered in the literature.

Number of deployed sensors

The more sensors are used the higher is the overall cost of the network. As the latter must be taken into consideration, the number of deployed sensors is one of the important metrics that needs to be considered within WSNs deployment process. Almost, all the papers dedicated to optimal sensor deployment consider to achieve the specified goals with minimum cost.

Coverage

The first challenge encountered in WSNs is how to cover a RoI perfectly. Coverage is one of the most fundamental issues in WSNs, which have a great impact on the performance of WSNs. Several coverage formulations arise naturally in many domains. Typically considered formulations are area coverage, point/target coverage and barrier coverage. Coverage of an entire area otherwise known as area or blanket coverage means that every single point within the RoI is within the sensing range of at least one sensor node. Point/Target coverage refers to observing a fixed number of targets. This type of coverage has obvious military applications such as those covered in [18]. A variation of point/target coverage

known as sweep coverage is also discussed in [69]. Barrier coverage refers to the detection of movement across a barrier of sensors. This problem was defined as the maximal breach path in [76].

Assessing the coverage varies based on the underlying sensor coverage model and the metric used to measure the collective coverage of deployed sensors. Some of the published papers use the worst case coverage, usually referred to as least exposure, measuring the probability that a target would travel across the RoI or an event would happen without being detected. However, most work has focused on the ratio of the covered area to the size of the RoI as a metric for the rate of coverage.

In the case of detection applications, the performance of a WSN is usually specified in terms of its overall detection and false alarm rate. Detection rate refers to the probability that a "yes" detection decision is made when the phenomena of interest is present. Obviously, correct detection decisions are always desirable. False alarm rate refers to the probability that a detection decision of "yes" is made when the phenomena of interest is absent. False alarms can occur due to imperfection in sensor design or noise and are undesirable. Maximizing the overall detection rate and minimizing the overall false alarm rate are competing objectives. Therefore, the end user of a WSN detection system usually provides a set of minimum detection requirements to satisfy and a set of false alarm requirements that are not to be exceeded.

It should be noted that in the literature, there is a confusion between coverage or coverage rate and the quality of coverage. Existing literature does not always make a clear distinction between the two terms which are often used interchangeably. In this dissertation, a distinction is made between these two terms. Coverage or coverage rate refers to the ratio of the covered area, in terms of detection rate, to the size of the RoI. Whereas, the quality of coverage takes account of the false alarm rate in addition to the coverage rate.

k-coverage

Applications requiring k-coverage ($k > 1$) may occur in situations where a stronger environmental monitoring capability is desired, such as military applications. Such a problem can be formulated as a decision problem, whose goal is to determine whether every point in the RoI is covered by at least k sensors, where k is a predefined value.

Preferential coverage

The preferential/differentiated/non-uniform coverage, which considers the satisfaction of different coverage levels in different geographical areas of the RoI, is also an important issue. The distribution of the requested event detection probability will not be uniform as it will depend on the application, RoI, and the event characteristics. The requested event-detection probability distribution can also be influenced by geographical factors. In many real world WSNs applications, such as surveillance applications, the supervised RoI may require high detection probabilities at certain sensitive areas. However, for some not so sensitive areas, relatively low detection probabilities are required to reduce the number of deployed sensors. For example, in a fire-detection system, high detection probabilities (close to 1) are requested for high-risk areas (e.g. those close to chemical deposits), and low detection probabilities for low-risk areas. In this case, the coverage rate is the percentage of the RoI where the generated detection probabilities are equal or greater than the required detection probabilities.

Network connectivity

Another issue in WSNs design is the connectivity of the network. Usually, we say that a network is connected if any sensor can communicate with any other sensor (possibly using other sensors as relays: multi-hop path). This definition can be relaxed with regards to communication among sensors: one-to-one, one-to-many, and many-to-one. For instance, in surveillance applications, only many-to-one communication is required. Network connectivity is necessary to ensure that information collected by sensors is sent to the sink. In this case, we say that a network is connected if any sensor can communicate with the sink (either directly or via multi-hop path).

It should be noted that connectivity and coverage are related since they are both affected by the position of sensors. This issue will be discussed in the next chapter.

Network lifetime

One major challenge in the design of WSNs is the fact that energy resources are very limited. Recharging or replacing the battery of the deployed sensors may be difficult or impossible. The number and positions of sensors have a direct impact on the WSN lifetime. When few sensors are deployed, a weak lifetime results. However, deploying many sensors does not automatically increase the lifetime. In fact, when the number of sensors increases,

volume of traffic also increases. Consequently, the energy consumption increases and the lifetime decreases. Thus, deploying more sensors does not necessary solve the problem. To increase the network lifetime, sensors must be located in the right places by taking into consideration factors such as the position of the sink(s), event frequency, and any specific mechanism implemented in the sensors to build paths to the sink (routing protocol).

Energy efficiency

This criterion is often used interchangeably with the network lifetime. Due to the limited energy resource in each sensor, we need to use the sensors in an efficient manner to increase the network lifetime. There are at least two approaches to the problem of conserving energy in sensor networks connected with optimal placement. The first approach is to plan a schedule of active sensors that enables other sensors to go into a sleep mode using overlaps among sensing ranges. The second approach is adjusting the sensing range of sensors for energy conservation.

In the case of mobile sensors, energy efficiency mainly is to schedule mobile sensors to move to the designated locations according to the result computed by the placement strategy such that the energy consumption due to movement is minimized.

Data fidelity

Ensuring the credibility of the gathered data is obviously an important design goal of WSNs. A sensor network basically provides a collective assessment of the detected phenomena by fusing the readings of multiple independent (and sometimes heterogeneous) sensors. Data fusion boosts the fidelity of the reported incidents by lowering the probability of false alarms and of missing a detectable object. Increasing the number of sensors reporting in a particular region will surely boost the accuracy of the fused data. However, redundancy in coverage would require an increased sensor density, which can be undesirable due to increased cost or decreased survivability (e.g., the potential of detecting the sensors in a combat field).

Fault tolerance and load balancing

Since WSNs are subject to failures, fault-tolerance becomes an important requirement for many WSN applications. In fact, the availability of the services provided by a WSN

to a large extent is affected by faults that may occur due to various reasons such as malfunctioning hardware, software glitches, dislocation, or environmental hazards, e.g., fire or flood. A WSN that is not prepared to deal with such situations may suffer a reduction in overall lifetime, or lead to hazardous consequences in critical application contexts.

One of the major fault tolerance techniques is the exploitation of redundancy, which is often a default condition in WSNs. For dealing with failures, techniques for reliable routing in WSNs have been proposed and are well-understood, but to be effective these depend on a physical network topology that ensures alternative routes to the sink are available. This requires sensor network deployment to be planned with an objective of ensuring some measure of robustness in the topology, so when failures occur that routing protocols can continue to offer a reliable delivery. Thus, a network should be k-connected, which allows $k - 1$ sensors to fail while the network would still be connected. A second approach is sensor value fusion that seeks to provide high-level information derived from a number of low-level sensor inputs. There, the inherent redundancy of sensors can be used to provide fault-tolerant data aggregation. Another major approach is the involvement of sinks or other resourceful sensors to maintain operations after failures.

1.3 Problem Statement

This dissertation addresses one of the most fundamental issues in WSNs designing: the deployment problem. This specific problem has different appellations in the literature, e.g., placement, layout, coverage or positioning problem in WSNs. Mainly, deployment's aim is to determine the number and locations of sensors. In most practical applications, sensors do not change their locations once they are deployed on the sensing field. We call this kind of deployment practice *static sensor deployment*. However, when sensors are able to move on their own, dynamic deployment reconfiguration can be exploited to enhance the network performance. We call this kind of deployment practice *dynamic sensor deployment*.

In this dissertation, the first problem we will study is the most naive approach to deploy static WSNs: random deployment. We conduct an empirical study to analyze the intrinsic properties of several random node placement strategies and devise a practical random deployment strategy that mixes the simple diffusion and the constant diffusion strategies.

A problem which impinges upon the success of any WSN deployment is the fact that there is an inherent imperfection associated with sensor readings. Indeed, sensors may

not always provide reliable information, either due to hardware configuration and/or environmental conditions. Some practical examples include AWAIRS at UCLA/RSC [6], Smart Dust at UC Berkeley [60], the USC-ISI network [44], and the DARPA SensIT systems/networks [41]. As an example, for omnidirectional acoustic sensors or ultrasonic sensors, a longer distance between the sensor and the target generally implies a greater loss in the signal strength or a lower signal-to-noise ratio [120]. This suggests that we can build an abstract sensor coverage model to express the uncertainty in sensor readings. Therefore, it is very important to take into account this issue in the deployment process to anticipate this sensing behavior. In the second problem, we will focus on a more elaborate static deployment approach: deterministic deployment, where each sensor is placed at predetermined coordinates to guarantee the quality of surveillance provided by the WSN. The first fundamental question that arises is: How well do the sensors observe the world? In other words, how can one characterize "observability" of phenomena and subsequently sensor coverage? What are the errors and confidence levels associated with the reported results? Can there be any guarantees on the levels of coverage that the sensors provide in a given scenario? The second fundamental question is: can information about coverage be used to design better deployment strategies? This is the most challenging problem we will solve in this dissertation. Our research aims to establish the best network topology while also taking into account:

1. Deployment cost: Minimizing the number of sensors deployed.
2. Coverage rate/quality: Ensuring that each point in the RoI is "properly" covered while considering uncertainties related to sensors readings.
3. Network connectivity/lifetime: Calibrating the distances separating sensors to ensure reliable communication while minimizing the communication cost.
4. Data fidelity: Ensuring the credibility of the gathered data.
5. Sensors reliability: Considering sensors reliability in the deployment process.
6. Harsh deployment environments: Considering harsh deployment environments in the deployment process.

In this dissertation, we will deal with the issue of non-linear and non-convex multi-objective optimization which is NP-complete [61]. To the best of our knowledge, this dissertation is the first piece of research that studies WSNs deployment including the above objectives.

Till now, our emphasis is on static sensor deployment approaches. However, there is still the case of mobile sensors. In the third problem, we will focus on dynamic sensor deployment approaches. Specifically, we consider the problem of hole detection and healing in mobile WSNs.

1.4 Dissertation Contributions

Our research aims primarily at optimizing the WSNs deployment in surveillance applications by exploiting the belief functions theory. The contributions of this dissertation can be broken up into the following areas.

- **Random deployment**: We survey existing random node placement strategies. We categorize random placement strategies into simple and compound. An empirical study has been carried out yielding a detailed analysis of random deployment intrinsic properties, such as coverage, connectivity, fault-tolerance and network lifespan. We also investigate the performance of a hybridization of the simple diffusion model that places a large number of nodes around the sink and the constant diffusion that provides high coverage and connectivity rates. We show that such hybridization should ensure best performance. The obtained results give helpful design guidelines in using random deployment strategies.
- **Deterministic deployment**: We survey existing deterministic deployment strategies. A unified approach for robust uncertainty-aware deployment of WSNs is proposed.
  - **Sensor coverage models**: Existing sensor coverage models such as the binary model and the probabilistic model may not be realistic in many cases and remain limited. Based on the transferable belief model, we define an evidence-based sensor coverage model that reflects reality well and can be easily extended to include deployment-related issues, such as sensor reliability. As an example of application, we devise an evidence-based detection coverage model. Experimental results based on both synthetic data sets and data traces collected in a real experiment for vehicle detection are provided to demonstrate the benefits of the evidence-based sensor coverage model over state-of-the-art coverage models.
  - **Placement algorithms**: By exploiting the belief functions reasoning framework, we present a flexible framework for target/event detection within the transferable belief model. Using the developed framework, we propose five uncertainty-aware deployment algorithms, namely, *EBDA*, *E2BDA-S*, *E2BDA-C*, *E2BDA-P*,

and *E2BDA*. Related issues, such as connectivity, preferential coverage, challenging environments and sensor reliability, are also discussed. Experimental results based on both synthetic data sets and data traces collected in a real deployment for vehicle detection are provided to demonstrate the ability of our approach to achieve an efficient WSN deployment.

- **Fusion-based deterministic deployment**: We survey existing fusion-based deterministic deployment strategies. A unified approach for robust uncertainty-aware deployment of fusion-based WSNs is proposed.
  - **Sensor coverage models**: Unlike prior efforts that rely on simple fusion schemes, a generic evidence fusion scheme, that captures several characteristics of real-world applications, is defined on the basis of Dempster-Shafer theory.
  - **Placement algorithms**: The fusion-based uncertainty-aware WSNs deployment problems are formulated as binary non-linear and non-convex optimization problems and two efficient heuristics, *GEBDA* and *MGEBDA*, using genetic algorithms are investigated. The proposed approach is validated through extensive experiments carried out on both synthetic data set and real data traces collected in a military vehicle detection experiment. The obtained results demonstrate the interest of the evidence fusion model that considers in a meaningful way the information on the quality of sensors decisions as well as the reliability of these sensors along with their uncertain and imprecise decisions. Also, the devised approaches outperform the state-of-the-art deployment strategies.
- **Dynamic deployment**: We discuss the main drawbacks of existing solutions and we identify four key elements that are critical for ensuring effective coverage in mobile WSNs. We devise a lightweight and comprehensive two-phase protocol, HEAL, for ensuring area coverage employing a mobile WSN. The first identifies the boundary nodes and discovers holes using a lightweight localized protocol over the Gabriel Graph of the network. The second treats the hole healing, with novel concept, hole healing area. We propose a distributed virtual forces-based local healing approach where only the nodes located at an appropriate distance from the hole will be involved in the healing process. Through extensive simulations we show that HEAL deals with holes of various forms and sizes, and provides a cost-effective and an accurate solution for hole detection and healing.
- **Post-deployment**: We discuss post-deployment approaches that can be exploited to enhance the performances of the WSN once is deployed. Specifically, we propose a

new technique aimed at providing a good spatial-temporal distribution of the WSN lifetime. The proposed technique effectively balances the energy consumption of the entire WSN, and avoids the energy hole problem. Simulation results show that the proposed approach can be every effective for long-lived WSNs.

- **Simulation Tools**: Multiple protocols were simulated in TOSSIM [67]. Further, multiple simulators in C++, NS2 [45] and Matlab were created in the early stages of this dissertation to help estimate network performance. Also, many Perl scripts were developed to exploit the DARPA SensIT dataset [41].
- **Embedded Software**: Multiple protocols were implemented in TinyOS [68] and Arduino [16].
- **Deployment**: We deployed, *ArduiNet*, a real WSN testbed for motion detection comprising more than 40 sensor nodes. The ArduiNet was designed, developed, and implemented in the LiSSi laboratory.

In summary, this dissertation provides guidelines for WSNs deployment.

1.5 Dissertation Organization

The rest of this dissertation is organized as follows. Chapter 2 highlights relevant background and related work in the literature. In Chapter 3, we investigate the random deployment of WSNs. We present a survey and taxonomy of random node placement in WSNs and a detailed analysis of their intrinsic properties. We also investigate the performance of a hybrid random deployment model. Our WSNs uncertainty-aware deployment framework is proposed in Chapter 4. It includes: defining an evidence-based sensor coverage model based on the transferable belief model [106], conceiving a detection coverage model build on our evidence-based coverage model, and formulating four different forms of the uncertainty-aware wireless sensor networks deployment problem, in addition to devising five uncertainty-aware deployment algorithms. In Chapter 5, on the basis of Dempster-Shafer theory [106], we propose an uncertainty-aware fusion-based WSNs deployment framework. We investigate coverage quality, communication reliability, and network lifespan. Deployment-related issues such as sensor reliability and challenging environments are also discussed. Chapter 6 presents the results obtained by deploying an experimental testbed for motion detection. Finally, Chapter 7 concludes the dissertation, presents our ongoing work and provides directions for future work. Background information about the belief functions theory is presented in Appendix A.

Chapter 2

Literature Review & Related Work

2.1 Introduction

Sensor placement is a fundamental issue in WSNs design [13, 143]. It determines many intrinsic properties of the WSN, such as its coverage, connectivity, cost and lifespan. In most practical applications, sensors do not change their locations once they are deployed in the Region of Interest (RoI). We call this kind of deployment practice *static sensor deployment*. In the literature, static sensor deployment is typically implemented in one of two approaches: *random deployment* or *deterministic deployment*. The choice of an approach depends on several factors, such as the deployment objectives and constraints, the type of sensors, application and the RoI [13, 143]. With the advances in mobile devices, some of these sensors are able to move on their own, this is achieved by embedding the sensors on mobile platforms such as Robomote [100] and iMouse [112]. Mobile sensors permit a dynamic deployment reconfiguration that improves the network performance. We call this kind of deployment practice *dynamic sensor deployment* or *self-deployment*. The random deployment usually serves as an initial phase of the self-deployment.

In this Chapter, we discuss these three deployment approaches, namely, random deployment, deterministic deployment and self-deployment. We begin with the most trivial one which is random deployment. First, we present a survey and taxonomy of random node placement in WSNs and we discuss some related work. Second, we discuss a more elaborated deployment approach which is deterministic deployment. We highlight the two components involved in the deterministic deployment strategy, namely, the sensor coverage model and the placement algorithm. We also discuss sensor coverage models and

placement algorithms found in the literature. Finally, we consider the case of mobile sensors. We explore the published self-deployment strategies and we classify the identified strategies into six classes. For each class, we present its principle and the most important algorithms belonging to the class.

2.2 Random Deployment

Random deployment is the most practical way in placing the sensors. In random (or stochastic) node placement, sensor-positions are defined by a Probability Density Function (PDF) [1]. Depending on the deployment strategy, the coordinates of the sensor positions may follow a particular distribution. In this section, we define the PDF and briefly describe the characteristics of each random node placement strategy. We categorize the random placement strategies into *simple* and *compound* (Fig. 2.1). Simple strategies are mere variants of the simple diffusion strategy, whereas compound strategies are realized by repeated simple diffusion.

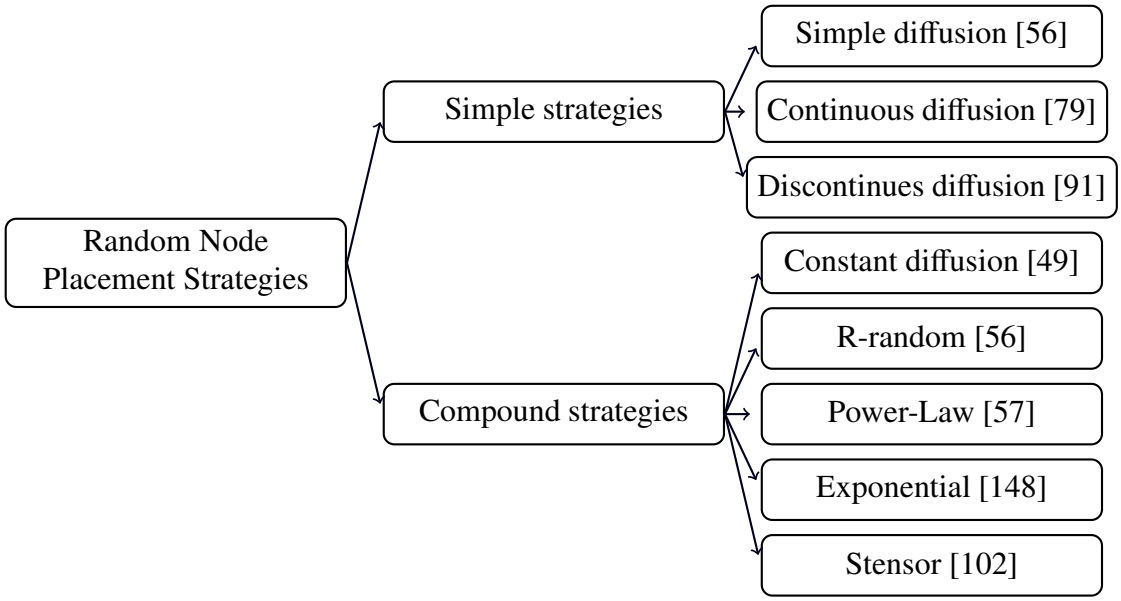


Figure 2.1: Random node placement strategies taxonomy.

1. The probability of a sensor being within the RoI = $\{x_1 \leq X \leq x_2, y_1 \leq Y \leq y_2\}$ can be written in terms of PDF as follows: $P(x_1 \leq X \leq x_2, y_1 \leq Y \leq y_2) = \int_{x_1}^{x_2} \int_{y_1}^{y_2} f(x, y)dxdy$.

2.2.1 Simple Random Node Placement Strategies

Simple diffusion

The simplest way to deploy sensors is to scatter them from the air [13, 57, 122]. Since all the information must reach the sink, the distribution is centered on the sink. Lightweight sensors will have a high air resistance, this randomizes their placement, we call the resulting distribution a *simple diffusion*. In [56], this deployment process was modeled by a linear diffusion equation, whose solution is a two-dimensional normal distribution. The PDF of sensor-positions is:

$$\begin{aligned} f(x) &= \frac{1}{2\pi\sigma^2} h(\|x-c\|), \quad x \in \mathbb{R}^2 \\ h(r) &\triangleq e^{(\frac{-r^2}{2\sigma^2})} \end{aligned} \tag{2.1}$$

In Equation 2.1, $c \in \mathbb{R}^2$ is the mean position at ground which is just under the point where sensors are scattered and σ^2 is the variance of the distribution. The variance is determined by various factors (e.g., shape or weight of sensors, or the height at which sensors are released). It should be pointed out that when air currents are strong, sensor-positions are governed by another formula [56]. Fig. 2.2 shows an example of a simple diffusion of 498 nodes in a RoI of $300m \times 300m$.

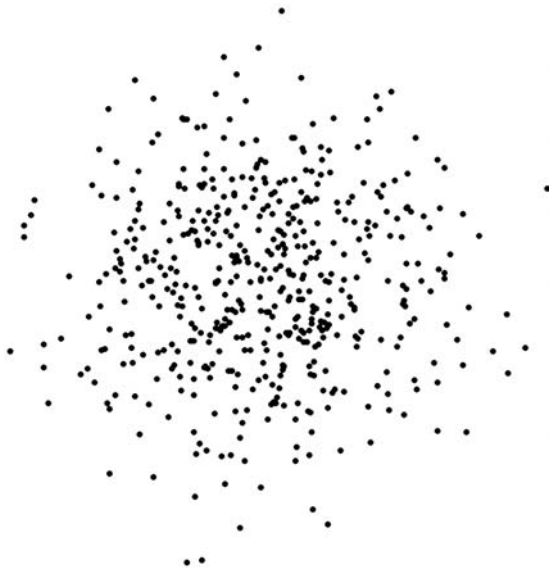


Figure 2.2: An example of simple diffusion.

We consider that any random node placement can be realized by repeated simple diffusion with different means and variances. The theoretical basis for this consideration is described in [57].

Continuous diffusion

If sensors are thrown off an aircraft that flies over the middle of the RoI, most sensors are expected to fall somewhere close to the central line (denoted by B), and several sensors are likely to end up further out. Along the axis of flight, the node distribution is Uniform, while it is Gaussian in the orthogonal direction [79]. We call the resulting distribution a *continuous diffusion*. The PDF of sensor-positions is (the x-axis is the axis of flight):

$$\begin{aligned} f(x) &= \frac{1}{|B|}, \quad x \in \mathbb{R} \\ f(y) &= \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-m)^2}{2\sigma^2}}, \quad y \in \mathbb{R} \end{aligned} \tag{2.2}$$

where $|B|$ is the length of the axis of flight. Fig. 2.3 shows an example of a continuous diffusion of 497 sensors in a RoI of $600m \times 200m$.

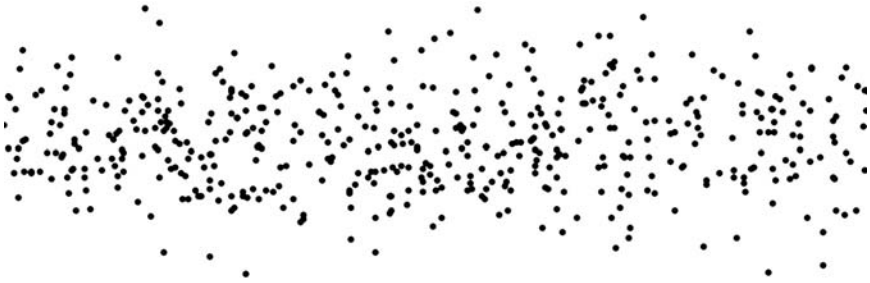


Figure 2.3: An example of continuous diffusion.

Discontinues diffusion

In this model, the sensors are dropped by an aircraft that flies over the middle of the RoI. We propose a discontinuous dropping of the sensors defined as follow: sensors are thrown discontinuously in a single flying over. In each throw, n sensors will be dropped. In the end, we will have N dropped sensors, with $N = n \times$ *number of throws*. Fig. 2.4 shows an example of a discontinuous diffusion with $N = 98$ and the *number of throws* $= 5$ in a

RoI of $600m \times 200m$. If we increase the number of throws, the discontinuous diffusion converges to the continuous diffusion, as shown in Fig. 2.5.

Figure 2.4: An example of discontinuous diffusion (5 throws).

Figure 2.5: An example of discontinuous diffusion (10 throws).

2.2.2 Compound Random Node Placement Strategies

A compound random deployment model is a random deployment model realizable by repeated simple diffusion with different means and variances. Therefore, to obtain practically a compound random deployment model we may need several flying over the RoI. In this section, we survey some existing compound random node placement strategies.

Constant diffusion

In many works [49, 143], the sensors are placed in such a way that their density is constant. Such random distribution is called *constant diffusion*. The PDF of the sensor-positions is given by the following equation:

$$f(x) = \frac{1}{|RoI|}, \quad x \in \mathbb{R}^2 \tag{2.3}$$

An example of constant diffusion is illustrated in Fig. 2.6. Here, the number of sensors is 400 and the RoI is $300m \times 300m$.

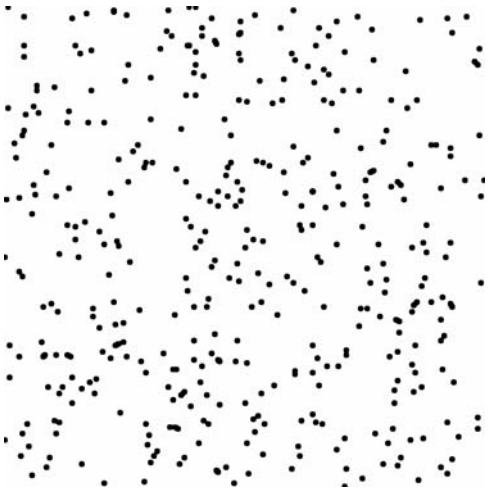


Figure 2.6: An example of constant diffusion.

R-random

This distribution was proposed in [56], where the nodes are uniformly scattered with respect to the radial and angular directions from the sink. The R-random node distribution pattern simulates the effect of an exploded shell and follows the following PDF for sensor-positions in polar-coordinates within a distance R from the sink:

$$f(r,\theta) = \frac{1}{2\pi R} \quad 0 \le r \le R, \;\; 0 \le \theta \le 2\pi \tag{2.4}$$

An example of R-random placement is illustrated in Fig. 2.7. Here, the number of sensors is 394 and the RoI is $300m \times 300m$.

Power-Law

This distribution is characterized by the following two features [57]: First, the density of sensors is higher near the sink. Second, the degree of the sensors follows a power law. The PDF of the sensor-positions in polar-coordinates is:

$$f(r,\theta) = \frac{\alpha+1}{2\pi R}(\frac{r}{R})^{\alpha} \tag{2.5}$$

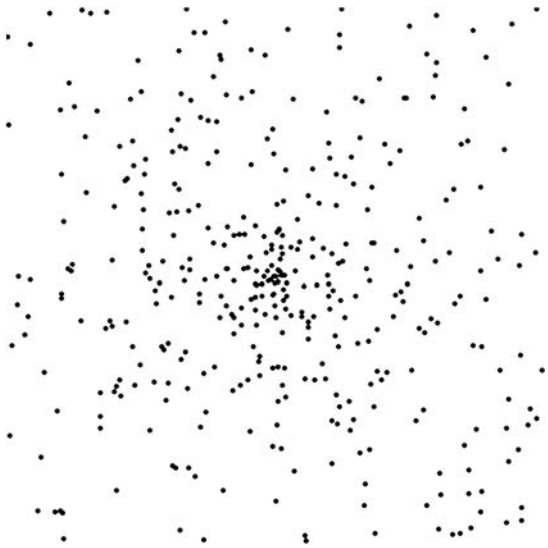


Figure 2.7: An example of R-random.

$$0 \leq r \leq R, \;\; 0 \leq \theta \leq 2\pi, \;\; -1 \leq \alpha \leq 1$$

The characteristics of the Power-Law placement are similar to those of the R-random placement.

Exponential

In this model, the distribution follows an Exponential law [148]. The PDF of sensor-positions is:

$$f(x) = \lambda e^{-\lambda x} \tag{2.6}$$

An example of Exponential diffusion is illustrated in Fig. 2.8. Here, the number of sensors is 436, the RoI is $300m \times 300m$ and $\lambda = 100$.

Stensor

In [102], the authors propose Stensor, a partition-based random node placement algorithm. They assume a rectangular area classified into small cells. Each cell cannot host more than one sensor. Sensors are distributed in those cells according to the following PDF:

$$f(x) = \frac{e^{-\sqrt{\lambda}} \lambda^{\frac{x}{2}}}{x!}, \;\; x \geq 0 \tag{2.7}$$

An example of Stensor placement is illustrated in Fig. 2.9. Here, the number of sensors

Figure 2.8: An example of Exponential diffusion.

is 16 and the RoI contains 1000 cells.

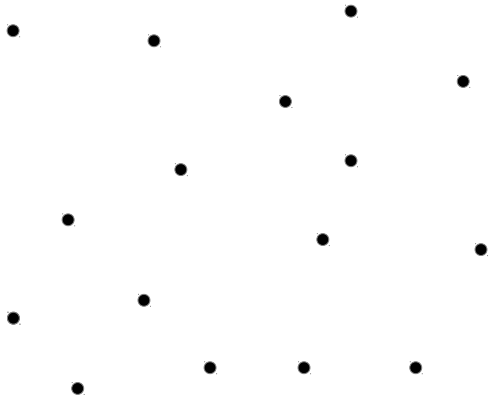


Figure 2.9: An example of Stensor placement [102].

2.2.3 Related work & Discussion

Random deployment has been studied extensively over the last decade. In [56], the authors evaluate the fault-tolerance against random failure from the random deployment viewpoint. Results prove that the tolerance against failure is low in constant placement, when the R-random placement has high fault-tolerance. In a more recent paper [57], the same authors show that the Power-law placement can raise fault-tolerance with appropriately selected control parameters.

Gupta and Kumar [49] studied necessary conditions on the transmission range needed for asymptotic connectivity of constant diffusion. Under the assumptions that nodes are deployed as a Poisson point process in a square region, the authors in [148] derive the node density required in order to maintain full coverage with high probability. Wan et al. [119] study how the probability of the k-coverage changes with the sensing radius or the number of sensors. They assume that the sensors are deployed as either a Poisson point process or a Uniform point process in a square region. Wang et al. [122] identify intrinsic properties of coverage/lifespan of a WSN that follows two-dimensional Gaussian distribution. They show that Gaussian distribution can effectively increase the network lifespan.

In [115], the authors discuss a performance study of congestion control algorithms when nodes are deployed under four different topologies, namely: simple diffusion, constant placement, R-random placement, and grid placement. The results depict that congestion control algorithms are highly affected by the node placement. In a more recent paper [97], the same authors evaluate the energy utilization performance of a congestion control and avoidance algorithm while considering the grid placement, biased random placement, simple diffusion, and random placement. Obtained results show that the best performance is obtained when the sensors are densely deployed near hot spots like the sink.

Generally speaking, studies on random deployment can be classified into two types: theoretical studies and empirical studies. The former is usually conducted for very few random deployment models under several simplifying assumptions [49, 119, 122, 148]. On the other hand, simulations analysis considers a couple of metrics and several random deployment models [56, 57, 79]. With both approaches, the effectiveness of random deployment has been shown to be certain. However, there has been little research on the analysis of intrinsic properties of random deployment. By intrinsic properties, we mean a full set of metrics such as coverage, connectivity, fault-tolerance, and lifespan. Furthermore, previous works fail to provide a holistic view of random deployment properties. Therefore, in Chapter 3, we conduct an empirical study to analyze the intrinsic properties of several random deployment strategies. Also, we devise a practical random deployment strategy that mixes the simple diffusion and the constant diffusion strategies, and we show that such hybridization ensures best performance.

As mentioned before, the main advantage of random deployment is its feasibility. In harsh environments such as disaster region, random deployment often becomes the only option. However, additional sensors are needed in random deployment as compared to deterministic deployment.

2.3 Deterministic Deployment

In deterministic deployment, each sensor is placed at predetermined coordinates which is usually pursued for indoor applications [13, 143], when sensors are expensive or when their operation is significantly affected by their position. In contrast to random deployment, deterministic deployment provides optimum network configuration, since positioning of sensors is determined beforehand to meet the design goals such as reducing cost and increasing coverage, connectivity, and lifetime. The most important metric considered in the literature is coverage. In fact, the research community has, in large part, investigated how to maximize the coverage rate while minimizing the usage of sensors (cost).

Usually, the deterministic deployment of WSNs involves two components: (i) a *sensor coverage model* and (ii) a *placement algorithm*. A sensor coverage model is an abstraction model trying to quantify how well sensors can sense physical phenomena at some locations in the RoI [120]. On the other hand, a placement algorithm determines the minimum number of sensors and their locations to achieve the desired design goals. The sensors locations are computed based on a sensor coverage model. This section reviews the related work of sensor coverage models and placement algorithms.

2.3.1 Sensor Coverage Models

When dealing with WSNs, a key component is the sensor coverage model which is used to reflect sensors' sensing capability and quality. Sensor coverage models are abstraction models trying to quantify how well sensors can sense physical phenomena at some locations, or in other words, how well sensors can cover such locations. In almost all cases, a sensor coverage model can be formulated as a function of the Euclidean distances (and the angles) between a space point and sensors.

The analysis of sensor coverage models shows the existence of different classifications based on distinct goals [120]. In this dissertation, these models are classified into three groups: i) binary coverage models, ii) probabilistic coverage models, and iii) fusion-based detection coverage models. A taxonomy for sensor coverage models is depicted in Fig. 2.10. In what follows, we elaborate some commonly used coverage models in details.

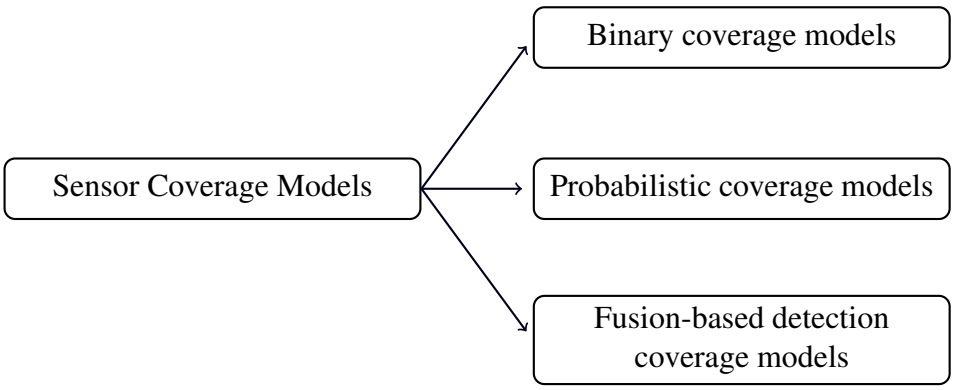


Figure 2.10: A taxonomy for sensor coverage models.

Binary Coverage Model

For its simplicity, the most widely used sensor coverage model in the literature is the binary model. In this model, sensors are modeled as having a predetermined range of effectiveness. Coverage within the range which is typically characterized by a disk (but can also be any arbitrary shape or a collection of shapes), is assumed to be full (1) and coverage outside of the given range is assumed to be non-effective (0). The specific semantics of the sensing region depends on the application at hand. For example, for a target-detection application, an effective sensing range indicates a region within which a potential target of interest can be detected with a given confidence level.

A well-known variant of the binary coverage model is the disk coverage model, depicted in Fig. 2.11. In this model, the sensing region of a sensor is often modeled as a disk with radius R_s centered at the position of the sensor, where R_s is referred to as the sensing range. Generally, R_s is determined by the sensor's hardware characteristics. For an event that occurs at point p, the following Equation computes the event-detection probability by a sensor s:

$$P_{s/p} = \begin{cases} 1 & \text{if } \|sp\| \leq R_s \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

where $\|sp\|$ is the Euclidean distance between a sensor s and the point p. In Fig. 2.11, the location A would be considered "covered", whereas the location B would be considered "uncovered".

Sensor coverage model can be considered as a measure of the sensing capability and quality of individual sensors. *Network coverage*, on the other hand, can be regarded as a

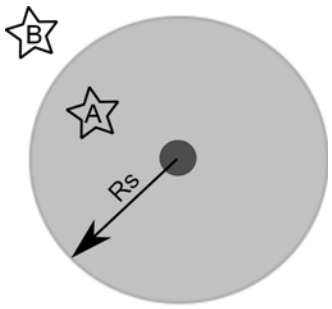


Figure 2.11: Illustration of the disk coverage model.

collective measure of the quality of service provided by a network of sensors at different geographical locations. The network coverage is defined as the ratio of covered area by the network to the RoI.

We derive the network coverage for the disk coverage model as follows: assuming the disk model, any event in the RoI will be detected by any arbitrary sensor if it is within R_s distance from the event. The probability that the event will be detected by an arbitrary sensor is $P = \frac{\pi {R_s}^2}{RoI}$ (neglecting boundary effect). The probability that the event will be undetected by the arbitrary sensor is equal to $(1 - P)$. Assume that N sensors are deployed randomly. Thus, the probability that the event will not be detected by any one of the sensor is $P_m = (1 - P)^N$. The probability that the event will be detected by at least one of the N sensors is equal to the coverage fraction and is:

$$\mathbb{P} = 1 - P_m = 1 - (1 - P)^N \tag{2.9}$$

While the binary model is appropriate for defining the foundation of research, as it allows a geometric treatment to the coverage problem, the binary coverage model is overly simplistic, it fails to capture the stochastic nature of sensing and it could cause wrong estimation of system performance in the real world.

Probabilistic Coverage Model

Some researchers argue that the sensing quality of a sensor will decay with the increase of the distance away from the sensor, environmental conditions, hardware configurations, and other problem specific attributes [39, 120, 152]. *Probabilistic coverage models* [27, 32, 39, 120, 121, 143, 152] are used to capture such attenuated sensing qualities. An example of a probabilistic coverage model is given by:

$$P_{s/p} = \frac{C}{||sp||^{\alpha}} \tag{2.10}$$

where α is the path attenuation exponent, and C a constant.

As the coverage measure becomes very small when the distance between a space point and a sensor becomes very large, the coverage measure might be neglected, and some approximations can be made by truncating the coverage measure for larger values of distance. For example, Zou and Chakrabarty [154] propose the following truncated probabilistic coverage model:

$$P_{s/p} = \begin{cases} Ce^{-\alpha||sp||} & \text{if } ||sp|| \leq R_s \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

where α is a parameter representing the physical characteristics of the sensor unit, and R_s the sensing range. Fig. 2.12 illustrates the relation between the coverage measure and the sensor-point distance for the aforementioned coverage models.

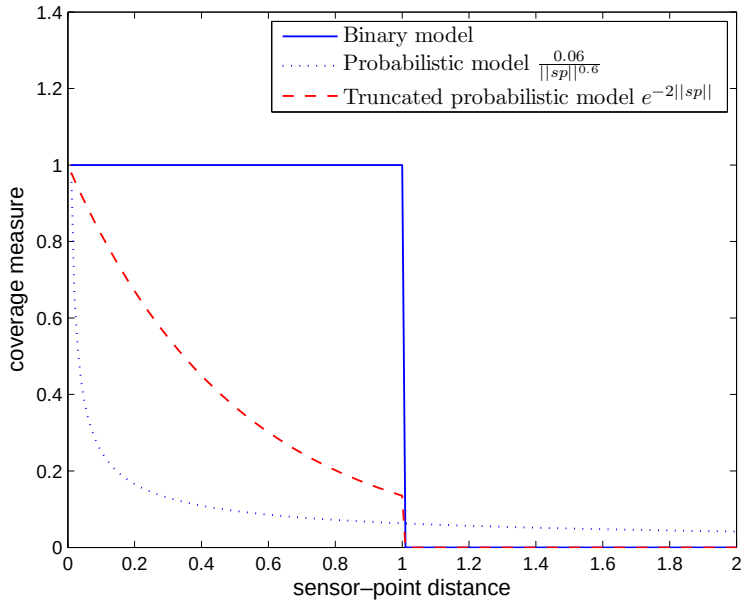


Figure 2.12: Illustration of the relation between the coverage measure and the sensor-point distance for (a) Binary model; (b) Probabilistic model; and (c) Truncated probabilistic model.

The network coverage or the overall detection probability by a set of sensors at point p,

denoted $\mathbb{P}_p$, is estimated as:

$$\mathbb{P}_p = 1 - \prod_{s \in RoI} (1 - P_{s/p}) \tag{2.12}$$

where $P_{s/p}$ is the detection probability of a sensor s at point p, expressed by Equations (2.10)-(2.11).

Fusion-based Detection Coverage Model

An important application of WSNs is to detect some event occurred at some location. In the context of detection application, the sensing quality of a sensor can be represented by its detection probability. The detection probability of a space point by a single sensor is related, among other factors, to the distance between them. However, the detection probability of a space point relative to a set of sensors is no longer simply computed as the addition of the detection probability of the point relative to each individual sensor (otherwise, it might be larger than one). Instead, a value fusion or decision fusion can be used to derive the detection probability. In contrast to the aforementioned coverage models, the fusion-based detection coverage model considers collaboration among sensors.

In decision fusion, each sensor makes a local decision based on its measurements and sends its decision to the fusion center, which makes a system decision according to the local decisions. The optimal decision fusion rule has been obtained in [26]. In value fusion, each sensor sends its measurements to the fusion center, which makes the detection decision based on the received measurements. Value and decision fusion were compared in [33, 34]. In [33], it was found that the former performs better in terms of detection probability for low noise levels; however, for noisy energy measurements decision fusion proves more robust. Clouqueur et al. [34] show that decision fusion become superior to value fusion as the ratio of faulty sensors to fault free sensors increases. Further, the communication cost is lower in decision fusion than in value fusion.

Based on different event scenarios and detection techniques, many fusion-based detection coverage models have been proposed in the literature [110, 117, 137]. In [110], authors describe a value fusion scheme as follows. Suppose sensor s is d_s meters from the target/event, the signal energy it measures is given by $U_s = W(d_s) + {N_s}^2$. Where $W(d_s)$ (resp. N_s) denotes the signal energy (resp. the noise strength) measured by the sensor s. N_s follows the zero-mean normal distribution with a variance of σ^2, i.e., $N_s \sim N(0, \sigma^2)$. Sensors send their energy measurements to the fusion center, which in turn compares the

average of all measurements against a threshold η to make a decision regarding the presence of the target/event. The threshold η is referred to as the detection threshold. In practice, the parameters of target and noise models are often estimated using a training dataset before deployment.

Under the aforementioned value fusion scheme, the detection probability is given by $P_d = \mathbb{P}(\frac{1}{n}\sum_{i=1}^{n}(W(d_s) + {N_s}^2) > \eta)$. As $N_s/\sigma \sim N(0,1)$, $\sum_{i=1}^{n}(N_s/\sigma)^2$ follows the Chi-square distribution with n degrees of freedom whose Cumulative Distribution Function (CDF) is denoted as $\chi_n(.)$. Hence, P_d can be calculated by:

$$P_d = 1 - \chi_n(\frac{n\eta - \sum_{i=1}^{n} W(d_s)}{\sigma^2}) \tag{2.13}$$

2.3.2 Placement Algorithms

In this section, we will describe relevant deterministic WSNs deployment strategies found in the literature. The analysis of WSNs deployment approaches shows the existence of different classifications based on distinct goals [120, 143]. In this dissertation, we classify the published strategies based on the used sensor coverage model. Thus, these strategies can be classified into three groups: i) binary deployment strategies, ii) probabilistic deployment strategies, and iii) fusion-based deployment strategies. A taxonomy for deterministic deployment strategies is depicted in Fig. 2.13.

Optimal deterministic deployment is a very challenging problem that has been proven to be NP-Hard for most of the formulations of sensor deployment [120, 143]. To tackle such complexity, several heuristics have been proposed to find sub-optimal solutions [9, 39, 92, 93, 120, 143, 153].

Binary Deployment Strategies

We talk about binary deployment strategies when the used sensor coverage model is the binary model. When considering a binary coverage model, the sensor deployment problem can be formulated as the famous *Art Gallery Problem* (AGP) addressed by the art gallery theorem [80]. The AGP seeks to determine the minimum number of cameras that can be placed in a polygonal environment, such that every point in the environment is monitored. Similarly, the deployment problem basically deals with placing a minimum number of sensors, such that every point in the RoI is covered. The AGP has been solved optimally in two-dimensional and shown to be NP-hard in the 3-dimensional case [80].

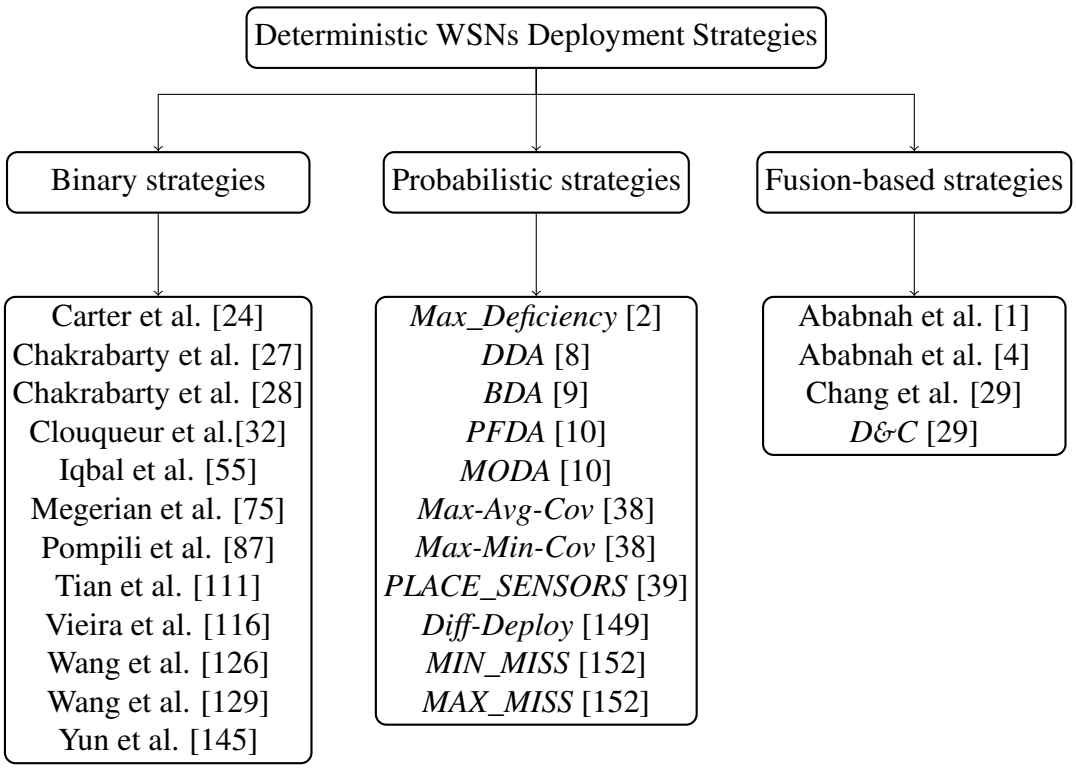


Figure 2.13: A taxonomy for deterministic deployment strategies.

Several variants of the AGP have been studied in the literature, including mobile guards, exterior visibility, and polygons with holes. Some other formulations include the minimal disk covering problem [62].

In a regular deployment, sensor-positions follow a regular topology, e.g., hexagonal, ring, star, etc. Many papers have studied regular deployment and compared it with constant diffusion (cf. Section 2.2.2). In [75], the authors proved that the quality of monitoring in regular deterministic deployment is better than constant diffusion. A comparison is made with different regular topologies such as triangle, square and hexagon. In [129], the authors proposed to deploy the sensors according to a grid topology. In [87], the authors have investigated the problem of achieving maximal coverage with the lowest sensor count in the context of underwater WSNs, a triangular grid has been proposed. In [116], an incremental deployment process is proposed. The main idea is to deploy a new sensor s at a point p, if no sensors are placed inside the sensing disk of s. In [55], the proposed deployment process builds a hexagon-structured topology. The authors showed that the hexagon topology obtains better performances in terms of the minimum number of sensors required when compared with the triangular and square topologies. Authors in [24] try to

minimize costs subjects to covering all target points. Yun et al. [145] propose a set of deployment patterns to achieve full area coverage and k-connectivity ($k \leq 6$).

In [32], the authors use a sequential deployment of sensors, *i.e.*, a limited number of sensors is deployed in each step until the desired minimum exposure or probability of detection of a target is achieved. In most practical applications however, we need to deploy the sensors in advance without any prior knowledge of the target and sequential deployment is often infeasible. Moreover, sequential deployment may be undesirable when the number of sensors or the RoI is large. Thus, a single step-deployment scheme is more advantageous in such scenarios.

Sensor placement on two- and three-dimensional grids was formulated as a combinatorial optimization problem and solved using integer linear programming [27, 28]. This approach suffers from two main drawbacks. First, computational complexity makes the approach infeasible for large problem instances. Second, the grid coverage approach relies on "perfect" sensor detection, i.e., a sensor is expected to yield a binary yes/no detection outcome in every case.

When dealing with binary deployment strategies, it should be noted that there is a relationship between connectivity and coverage. To the best of our knowledge, Wang et al. [126] and Zhang et al. [50] are the earliest papers that independently proved the same conclusion: if a convex RoI is completely covered by a set of sensors, the communication graph consisting of these sensors is connected when $R_c \geq 2R_s$. In other words, under the condition that $R_c \geq 2R_s$, a WSN only needs to be configured to guarantee coverage in order to satisfy both coverage and connectivity. Hence, if the transmission range R_c of a sensor is much longer than its sensing range R_s then connectivity is not an issue. This result has been generalized for the case of k-coverage [111]. For short, also k-coverage implies k-connectivity when $R_c \geq 2R_s$.

Probabilistic Deployment Strategies

In probabilistic deployment strategies, the sensor detection is modeled probabilistically. In [38, 39, 152], authors consider a probabilistic coverage model. The RoI is assumed as a two-dimensional $n \times n$ grid of points. Dhillon et al. [39] formulate the sensor placement as an optimization problem and propose a greedy heuristic, called *PLACE_SENSORS*, to optimize the number of sensors and determine their placement to provide sufficient grid coverage of the RoI. *PLACE_SENSORS* complexity is equal to $O(n^4)$. In [38], the same authors proposed two deployment algorithms. The first algorithm, *Max-Avg-Cov*, is identical

to the *PLACE_SENSORS* algorithm proposed in [39]. *Max-Avg-Cov* aims to maximize the average coverage of the grid points. The second one, *Max-Min-Cov*, aims to maximize the coverage of the grid points that are the least effectively covered.

Dhillon et al. [38, 39] defined four main variables D, M, M^* and M_{min}. $D = [P_{i/j}]$ is a sensor detection matrix for all pairs of grid points in the RoI. For an $n \times n$ RoI, we have a total of n^2 grid points, hence the matrix D consists of n^2 rows and n^2 columns, and a total of n^4 elements. $M = m_{ij} = 1 - P_{i/j}$ is a miss probability matrix. The vector $M^* = (M_1, M_2, ..., M_N)$ is the set of miss probabilities for the $N = n^2$ grid points in the RoI. At the start of the placement algorithm, no sensor is deployed so M^* is initialized to the all-1 vector, i.e., $M^* = (1, 1, ..., 1)$. Each sensor placed in the RoI decreases one or more entries in this vector. M_{min} is the maximum value of the miss probability that is permitted for any grid point. In *Max-Avg-Cov* a sensor is deployed at a grid point i where $\sum_j M_{ij}$ is minimal. However, in *Max-Min-Cov* a sensor is deployed at the grid point of minimum coverage in the RoI. The placement algorithms are stopped when the generated detection probabilities are greater than the requested detection probabilities, or the number of sensors deployed exceeds the allocated budget for sensors. The pseudo code of *Max-Avg-Cov* (resp. *Max-Min-Cov*) is outlined in Algorithm 1 (resp. Algorithm 2).

Algorithm 1 Max-Avg-Cov (M, M^*, M_{min})

1: $num_sensors = 1$;
2: **repeat**
3: **for** $i = 1$ to N **do**
4: $\sum_i = m_{i1} + m_{i2} + ... + m_{iN}$;
5: **end for**
6: Place sensor on grid point k such that $\sum_k$ is minimum;
7: **for** $i = 1$ to N **do**
8: $M_i = M_i m_{ki}$;
9: **end for**
10: Delete k^{th} row and column from the M matrix;
11: $num_sensors = num_sensors + 1$;
12: **until** ($M_i < M_{min}$ for all i, $1 \leq i \leq N$) $\vee$ ($num_sensors > limit$)

The computational complexity for both *Max-Avg-Cov* and *Max-Min-Cov* is $O(n^4)$. In addition, both algorithms create a matrix that contains n^4 elements, so the memory complexity for both *Max-Avg-Cov* and *Max-Min-Cov* is $O(n^4)$.

The uncertainty associated with the predetermined sensor locations is considered in [152]. The authors propose two sensor placement algorithms: *MIN_MISS* and *MAX_MISS*,

Algorithm 2 Max-Min-Cov (M, M^*, M_{min})

```
1: Place first sensor randomly;
2: num_sensors = 1;
3: repeat
4:    for i = 1 to N do
5:       M_i = M_i m_ki;
6:    end for
7:    Place sensor at grid point k such that M_k is max;
8:    Delete k^th row and column from the M matrix;
9:    num_sensors = num_sensors + 1;
10: until (M_i < M_min for all i, 1 ≤ i ≤ N ) ∨ (num_sensors > limit)
```

where the sensor location is modeled as a random variable with a Gaussian probability distribution. *MIN_MISS* and *MAX_MISS* are iterative algorithms, where in each iteration, only one sensor is deployed. In *MIN_MISS*, a new sensor is deployed at the grid point with the lowest over miss probability. Whereas, in *MAX_MISS*, a new sensor is deployed at the grid point that maximizes the over miss probability. *MIN_MISS* and *MAX_MISS* computational complexity is equal to $O(n^6)$.

The aforementioned probabilistic deployment strategies addressed the uniform deployment problem, in which the monitored event has the same importance at every point of the RoI. This means that the specified detection probability threshold for the entire RoI is the same. In [5, 8, 9, 10, 149], the authors addressed explicitly the differentiated deployment problem, in which the required detection probability thresholds at different locations are different. Aitsaadi et al. [8] proposed a deterministic deployment algorithm named differentiated deployment algorithm (DDA). It is based on mesh representation used in image processing and 3D modeling. *DDA* suffers from high computational complexity which is equal to $O(n^6)$. In a more recent paper [9], the same authors proposed a pseudo-random deployment strategy, Bernoulli Deployment Algorithm (BDA), based on Tabu search metaheuristic. *BDA* has a quadratic complexity but it cannot guarantee full area coverage even with a high number of iterations.

Authors in [149] formalized the deployment problem as a Linear Shift-Invariant (LSI) system. A heuristic algorithm, *Diff-Deploy*, is proposed. The disadvantage of this solution is the computational complexity which is equal to $O(\frac{4}{3}n^6)$. Given a pre-determined number of sensors, the deployment problem has been formalized as a linear quadratic regulator in [2, 5]. As the complexity of the optimal control based solution is high, Ababnah et al. [2, 5] proposed an approximation heuristic called *Max_Deficiency* whose computational

complexity is equal to $O(n^6)$.

Table 2.1 provides a comparative summary of the various probabilistic deployment approaches discussed above.

Table 2.1: A comparison between the various probabilistic deployment approaches.

Strategy	Computational complexity	Memory complexity	Primary objective	Secondary objective	Constraint
PLACE_SENSORS	$O(n^4)$	$O(n^4)$	Max. the average coverage	Min. the number of sensors	-
Max-Avg-Cov	$O(n^4)$	$O(n^4)$	Max. the average coverage	Min. the number of sensors	-
Max-Min-Cov	$O(n^4)$	$O(n^4)$	Min. miss detection proba	Min. the number of sensors	-
MIN_MISS	$O(n^6)$	$O(n^2)$	Min. the over miss proba	Min. the number of sensors	Locations uncertainty
MAX_MISS	$O(n^6)$	$O(n^2)$	Max. the over miss proba	Min. the number of sensors	Locations uncertainty
DDA	$O(n^6)$	$O(n^3)$	Satisfies preferential coverage	Min. the number of sensors	-
BDA	$O(n^2)$	$O(n^2)$	Satisfies preferential coverage	Min. the number of sensors	Connectivity
Diff-Deploy	$O(n^6)$	$O(n^4)$	Satisfies preferential coverage	Min. the number of sensors	-
Max_Deficiency	$O(n^6)$	$O(n^2)$	Satisfies preferential coverage	-	Fixed number of sensors

Fusion-based Deployment Strategies

We refer to deployment strategies that exploit a fusion-based detection coverage model as fusion-based deployment strategies. There has been little research in this area. Previous deployment approaches have used simplistic combination techniques such as averaging. Recently, Xing et al. [134] show by mean of probability theory that value fusion improves the coverage of WSNs. Based on a value fusion scheme [110], Chang et al. [29], formulate the deployment problem as a non-linear and non-convex optimization problem and propose the use of a high computational cost non-linear optimization algorithm, Constrained Simulated Annealing (CSA) and a relatively low computational cost Divide-and-Conquer (D&C) heuristic.

Chang et al. [29] assume that targets appear at a set of known physical locations referred to as surveillance spots $T = \{t_j, 1 \leq j \leq K\}$, where $t_j = (x_j, y_j) \in RoI$ is the coordinates of the j^{th} spot. They also assume that only the sensors close to a surveillance spot participate in the data fusion. For any surveillance spot, the fusion region is a disk of radius R centered at the spot. Finally, they define the impact region of a surveillance spot as the disk of radius $2R$ centered at the spot. The fusion-based deployment problem is defined as follows. Given a RoI and a set of surveillance spots T, find a sensor placement $S = \{(x_i, y_i), 1 \leq i \leq N\}$, such that the number of sensors $|S|$ is minimized subject to $min_{1 \leq j \leq K}\{P_{d_j}\} \geq \beta$. Where P_{d_j} is the detection probability which can be calculated by Equation (2.13) and β is the required minimum event detection probability threshold.

The straightforward optimal solution for this problem, proposed by Chang et al. [29],

is to iterate incrementally N from 1 to search for the optimal sensor placement. In each iteration, $min_{1\leq j\leq K}\{P_{d_j}\}$ is maximized. Once the constraint $min_{1\leq j\leq K}\{P_{d_j}\} \geq \beta$ is satisfied, the global optimal solution is found. To solve the non-linear and non-convex optimization problem in each iteration, Chang et al. [29] apply a non-linear programming solver based on the Constrained Simulated Annealing (CSA) algorithm [118], which is a global optimal algorithm that converges asymptotically to a constrained global optimum (Theorem 1 of [118]). However, the complexity of CSA increases exponentially with respect to the number of variables [118]. Therefore, for a large-scale deployment problem, the global optimal solution becomes prohibitively expensive.

Chang et al. [29] propose also a relatively low computational cost Divide-and-Conquer (D&C) heuristic. In the divide step, for each surveillance spot t_j, they find the set of spots within the impact region of t_j. In the conquer step, for each surveillance spot t_j, the fewest additional sensors are placed within the fusion region of t_j to cover t_j and the neighboring spots in its impact region. The optimization is implemented by the aforementioned CSA solver.

Ababnah et al. [3] treated the deployment problem in the context of optimal control theory. Specifically, they model the deployment problem as a linear quadratic regulator problem, with the deployment locations serving as control parameters. Assuming a value fusion scheme and under several approximations, Ababnah et al. [4] try to linearize the formulation proposed in [110] and solve the deployment problem as an optimal control problem. Within this framework, they propose a sequential sensor deployment algorithm whose computational complexity is equal to $O(3n^6 + 4n^4)$. In a recent work, Ababnah et al. [1] introduced a similar algorithm while considering the majority or counting rule as a decision fusion rule. The computational complexity of the proposed algorithm is equal to $O(3n^6 + 4n^4)$.

Table 2.2 provides a comparative summary of the various fusion-based deployment approaches discussed above.

Table 2.2: A comparison between the various fusion-based deployment approaches.

Paper	Computational complexity	Fusion model	Primary objective	Secondary objective	Constraint
[4]	$O(3n^6 + 4n^4)$	Value fusion	Max. detection performance	-	Fixed number of sensors
[1]	$O(3n^6 + 4n^4)$	Decision fusion	Max. detection performance	-	Fixed number of sensors
[29]	$O(e^{n^2})$	Value fusion	Max. detection performance	Min. the number of sensors	-
[29]	$O(n^2 m e^m)^*$	Value fusion	Max. detection performance	Min. the number of sensors	-

* m is the average number of spots in the impact region.

The aforementioned works are based on simple fusion schemes and ignore completely deployment-related issues such as sensor reliability and network connectivity. In Chapter 5, we investigate the evidence theory also known as Dempster-Shafer theory [106], which is well suited to this type of problem for two reasons: first, it reflects uncertainty or a lack of complete information and second, Dempster's rule for combination gives a convenient way for fusing together multiple uncertain sensory data from multiple sensors that can varies in their reliability.

2.4 Self-Deployment

All of the approaches discussed above initially compute the optimal location for the sensors and do not consider moving them once they have been positioned. Moreover, the context of the pursued optimization strategies is mainly static in the sense that assessing the quality of candidate positions are based on performance metrics like the sensing range, connectivity range, etc. In addition, the placement decision is made at the time of network setup and does not consider dynamic changes during the network operation.

Some schemes have advocated dynamic adjustment of nodes' location since the optimality of the initial positions may become void during the operation of the network depending on the network state and various external factors [143]. For example, traffic patterns can change based on the monitored events, or the load may not be balanced among the nodes, causing bottlenecks. Also, application-level interest can vary over time and the available network resources may change as new nodes join the network, or as existing nodes run out of energy. For instance, when many of the sensors in the vicinity of the sink stop functioning due to the exhaustion of their batteries, some redundant sensors from other parts of the monitored region can be identified and relocated to replace the dead sensors in order to improve the network lifetime.

Most researches about sensors mobility focuses on designing algorithms to relocate sensors in order to enhance the quality of monitoring. The proposed algorithms, mainly, strive to spread sensors in the RoI to maximize the covered area. The main difference among these schemes is *how exactly the sensors' new positions are computed*. Sensor mobility is exploited essentially to obtain a new stationary configuration, after the displacement of sensors to their desired locations, which improves coverage.

In this section, we explore the published self-deployment strategies. We classify the

identified strategies into six classes which are: 1) virtual forces-based approach, 2) pattern-based approach, 3) grid-quorum based approach, 4) computational geometry-based approach, 5) fuzzy logic-based approach, and 6) optimization-based approach. A taxonomy for self-deployment strategies is depicted in Fig. 2.14. For each class, we present its principle and the most important algorithms belonging to the class.

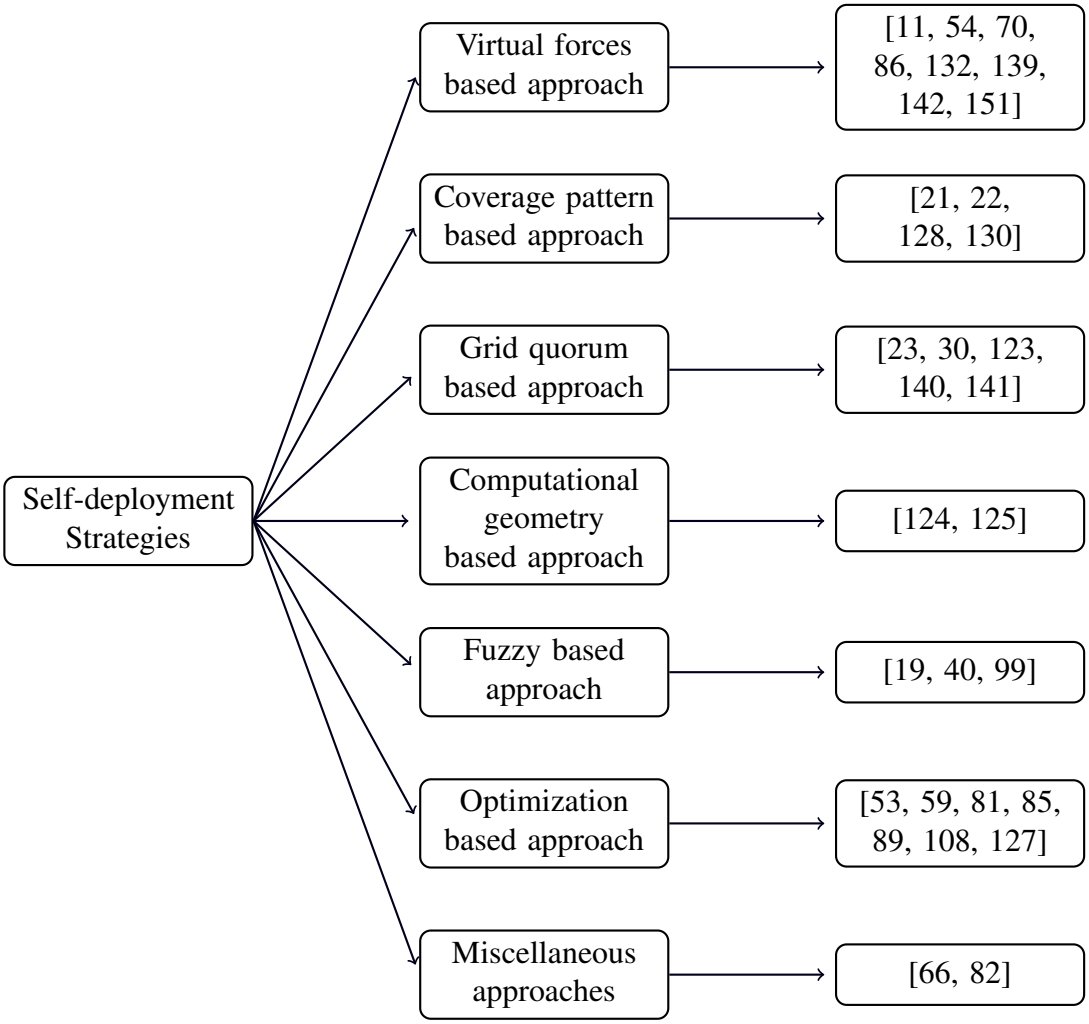


Figure 2.14: A taxonomy for self-deployment strategies.

2.4.1 Virtual forces based approach

In the virtual forces nodes self-deployment approach [11, 54, 70, 86, 132, 139, 142, 151], sensors behave as electromagnetic particles. A virtual force concept is adopted in order to relocate sensors to their next position. The inter-sensor force can either be attractive

or repulsive. The nature of the force depends essentially on the Euclidean distance between each pair of sensors, where each sensor is supposed to know its position and the list of all its neighbors. If two sensors are too close to each other, where the closeness is measured by a pre-determined threshold, they exert repulsive forces on each other. This ensures the minimization of the overlapping coverage area between the two sensors. On the other hand, if a pair of sensors are too far apart from each (according to the pre-determined threshold distance), they exert an attractive force on each other, and which will ensure a global uniform sensor placement. The final movement of a sensor is dictated by the combined force at that sensor due to sensors in its neighborhood. In Fig. 2.15, the overall force $\overrightarrow{F_1}$ on s_1 is given by $\overrightarrow{F_1} = \overrightarrow{f_{12}} + \overrightarrow{f_{13}}$. Where d_{th} is the threshold distance, $\|s_1 s_2\| > d_{th}$, $\|s_1 s_3\| < d_{th}$, and $\|s_1 s_4\| = d_{th}$.

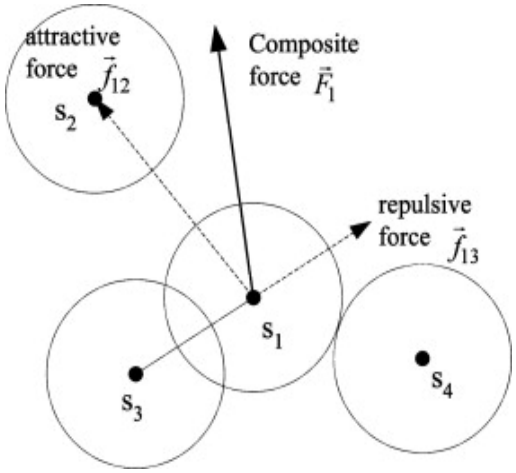


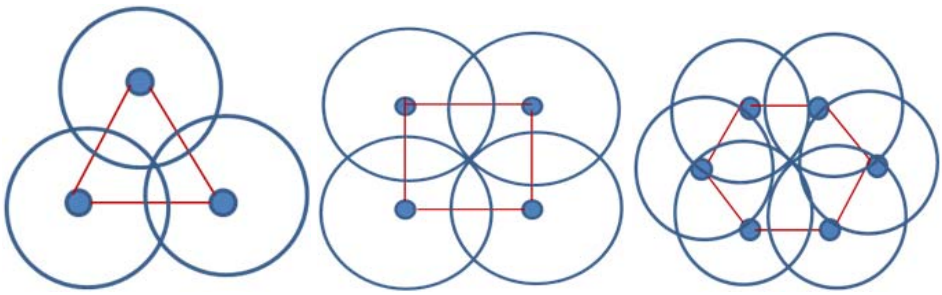
Figure 2.15: An example of virtual forces.

In the literature, we find an important number of algorithms which exploit the virtual forces principle for the relocation of mobile sensors. Zou and Chakrabarty [151] propose a cluster model where each sensor is exposed to three sources of force which are its neighbors, obstacles and holes on the RoI. Yang et al. [139] add in their force model a new source of force which is a virtual point situated inside the RoI named Deployment Center (DC). DC exerts an attracting force on all sensors. Sensors receiving forces from DC move toward it. Wong et al. [132] adopt a concept of back-off delay in order to solve the problem of simultaneous movement of sensors. This model reduces the collision between the sensors, but it suffers of excessive energy consumption. Yong and Li [142] propose a distributed algorithm based on the definition of the local and global density around a sensor to calculate the total force exerted by all its neighbors. Poduri and Sukhatme [86]

assume that every sensor has at least k neighbors in order to insure the connectivity of the whole network after the sensors movement. Howard et al. [54] exploit a model based on the potential field theory where each sensor attempts to find a balance state that result of the forces exerted by its neighborhood. Akkaya and Younis [11] use the sensors mobility in order to guaranty the overall network connectivity. Li et al. [70] exploit the sensors relocation in a purpose of target tracking where they attempt to find the ideal combination of forces which insure the overall coverage of the network.

2.4.2 Coverage pattern based approach

According to the coverage pattern based approach [21, 22, 128, 130], the final positions where the mobile sensors will be relocated are pre-calculated based on a pre-selected coverage pattern which can ensure the overall coverage and connectivity of the whole network. The RoI is partitioned into a virtual grid in the form of the selected coverage pattern. There are three types of grids commonly used: triangular lattice (Fig. 2.16(a)), square grid (Fig. 2.16(b)) and hexagonal grid (Fig. 2.16(c)).



(a) Triangular lattice. (b) Square grid. (c) Hexagonal grid.

Figure 2.16: Commonly used coverage patterns.

Triangular lattice is the best among the three kinds of grids as it has the smallest overlapping area. Square grid provides fairly good performance for any parameters while hexagonal grid is the worst among all since it has the biggest overlapping area. In addition to the type of grid, the size of grid also plays an important role. The size of grid need to be chosen based on how dense the WSN is going to be. For a highly dense network, small size grids help in reducing coverage holes thus providing better result. But in a sparse network, large grid size is better as it will avoid overlapping of sensors' sensing range therefore ensuring full utilization of their sensing capabilities. The vertices of the final grid are the target positions where the mobile sensors will be relocated. This approach presents the advantage

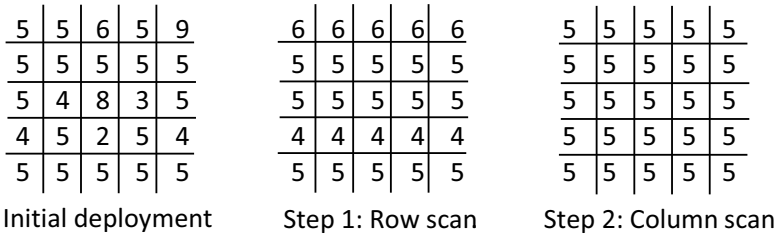
that based on the sensing range of each sensor we can compute the exact number of sensors to deploy initially which can insure the overall coverage of the RoI.

2.4.3 Grid quorum based approach

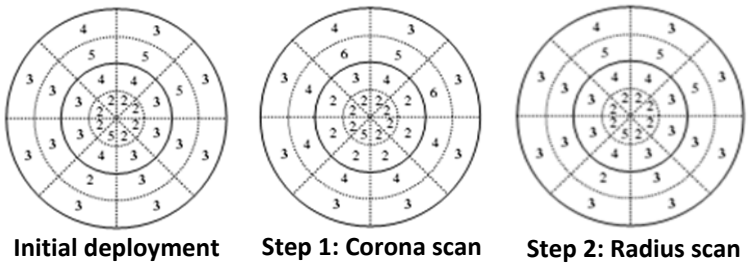
In the Grid quorum based approach [23, 30, 123, 140, 141], the sensor deployment problem is brought to a load balance problem as in traditional parallel processing, where each region corresponds to a processor and the number of sensors in a region corresponds to the load. The RoI is partitioned into a two-dimensional grid of cells through clustering. Each cluster corresponds to a square region and has a cluster-head that is in charge of bookkeeping and communication with adjacent cluster-heads. In contrast to the pattern-based and the virtual forces based schemes, the Grid quorum based approach doesn't try to find a final target position where the mobile node must be relocated, instead it attempts to define a target cluster in the grid where the node can be localized anywhere. A scan operation is then used to calculate the average load (number of sensors in a cluster) and then to determine the amount of overload and under-load in clusters. Load is shifted from overloaded clusters to underloaded clusters under the constraint of minimizing the overall traversed distance.

Yang et al. [141] transform the load balance problem to a matching problem in a bipartite graph. They propose a centralized solution named the Hungarian method which they compare with a distributed approach named SMART [140]. SMART uses a scan technique. The scan is performed on two steps first on rows and then on columns to settle the direction of the node movement. As shown in Fig. 2.17(a), after the first round, all rows are balanced. After the second round, all columns are balanced, as is the whole RoI. In [23], Cardei et al. present a centralized algorithm which uses a sector scan. The final deployment obtained by the algorithm is not uniform. The density of sensors around the sink is greater than the density in the other regions. Two scans are used in sequence: corona scans followed by radius scans as shown in Fig. 2.17(b). A corona scan will balance the number of sensors per corona. At the end of this scan, regions in the same corona will have the same number of sensors. In the radius scan, sensors are redistributed on segments according to the desired sensor densities.

Chellappan et al. [30] reject the fact that a sensor can move an unlimited distance to reach its final position. They propose a model where each sensor can only make a flip movement to a limited distance. Wang et al. [123] propose an algorithm named grid-quorum.



(a) SMART operating principal.



(b) Corona radius scanning operating principal.

Figure 2.17: Examples of grid quorum based approach.

They first construct a virtual grid of clusters. Then they identify the redundant sensors on a cluster to assign them to the empty adjacent clusters through a cascade movement.

2.4.4 Computational geometry based approach

In this approach [124, 125], a geometric computation is used in order to discover the existence of coverage holes in the RoI and settle the target position where the mobile sensor will be relocated from densely deployed areas to sparsely ones. The most commonly used computational geometry approaches are Voronoi diagram and Delaunay triangulation.

The Voronoi diagram [47] is an important data structure in computational geometry. It represents the proximity information about a set of geometric nodes. The Voronoi diagram of a collection of nodes partitions the space into polygons. Every point in a given polygon is closer to the node in this polygon than to any other node (Fig. 2.18). Based on the sensors positions, the Voronoi diagram of the network is constructed and the decision whether the sensors need to reposition or to stay will be made based on the diagram.

Wang et al. [124] propose three relocation protocols based on the use of the Voronoi diagram they are entitled *VEC*, *VOR* and *MiniMax*. The three algorithms attempt to reach a uniform distribution of sensors on the RoI by the relocation of the sensors from dense regions to sparse ones relying on the geometric properties of the Voronoi diagram. *VOR*

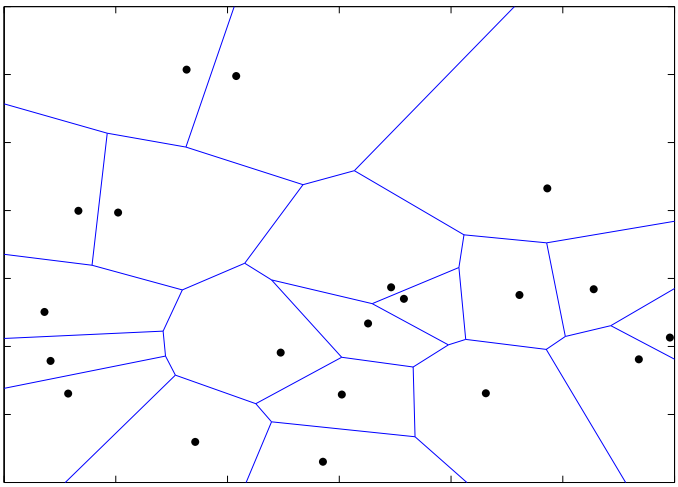


Figure 2.18: An example of voronoi diagram.

is a pull-based algorithm which pulls sensors to their local maximum coverage holes. In *VOR*, if a sensor detects the existence of coverage holes, it will move toward its farthest Voronoi vertex. Similar to *VOR*, *MiniMax* fixes holes by moving closer to the farthest Voronoi vertex, but it does not move as far as *VOR* to avoid the situation that the vertex which was originally close becomes the new farthest vertex.

In [125], Wang et al. present a protocol named Bidding protocol. The authors suppose a hybrid network composed of static and mobile sensors. The static sensors identify the coverage holes by means of the Voronoi diagram then the mobile sensors will relocate in order to fill the voids identified by the static ones.

2.4.5 Fuzzy based approach

The fuzzy based approach [19, 40, 99] proposes to use fuzzy logic system to control the sensor movement. It parts from the conviction that Fuzzy reasoning is a powerful tool that has the capability to handle the uncertainty and ambiguity; it can present more smooth results than common proposed methods in reasoning. The fuzzy system fixes several rules based on the definition of a selection of antecedents that can be for example the number of neighbors or the Euclidean distance between a pair of sensors. The output provided by the system is the next-step move distance and direction for each sensor.

Shu et al. [99] present a relocation algorithm based on a fuzzy system. The system uses two antecedents which are the Euclidean distance and the number of neighbors in order

to identify the movement distance for each node. To determine the movement direction the authors appeal to the Coulomb's law. Bahareh et al. [19] use the number of nodes in the neighborhood of a node and the distance from the closest obstacle to settle the system output which is a priority order of movement for all the sensors. Du et al. [40] propose a fuzzy logic based distributed decision making algorithm. The mobile sensor uses the fuzzy logic algorithm to determine the best location that it should move to. Their utility function includes three factors: the movement cost, the coverage gain, and the connectivity gain.

2.4.6 Optimization based approach

Algorithms belonging to this class [53, 59, 81, 85, 89, 108, 127] exploit the power of metaheuristics in order to settle the position, the direction and the movement speed of a mobile sensor. Ant Colony (AC), Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) are examples of such algorithms.

Genetic Algorithms (GA) are very robust optimization techniques used to find a global solution for a complex problem. Contrary to the other relocation approaches, the GA based approach doesn't give a unique solution to the problem. It first generates a population of candidate solutions to the problem. Each individual of the population is characterized by its genome which is composed of one or several chromosomes represented by the speed, position and movement destination of the sensor. The genome is used in a fitness function which evaluates the candidate solutions to settle the best solution for the current iteration. Jiang et al. [59] propose a distributed algorithm that uses the node position and the positions of its neighbor's to represent the sensor in the fitness function. The algorithm supposes a dense initial deployment to ensure the coverage of the whole deployment area. In [108], the author proposes a clustering based algorithm. It's executed on five steps: Initialization, Collect, Recombination, Mutation and Termination. Sahin et al. [89] present a genetic algorithm based on the use of virtual forces. The node's genome is defined as its speed and movement direction which are calculated by means of virtual inter-node forces.

In addition to GA, various metaheuristics have been exploited by researchers for WSNs self-deployment. In [85], the ant colony theory is used to find the optimization scheme for path planning and deployment of mobile sensors. Wang et al. [127] proposed a new approach based on parallel particle swarm optimization (PPSO). The Simulated annealing approach has been considered in [53]. In [81], the artificial bee colony algorithm, a swarm based intelligent method inspired by modeling the foraging behavior of honey bees, is

used for the self-deployment problem of WSNs. Simulation results show that the proposed algorithm obtains better deployments for WSNs than the PSO algorithm.

2.4.7 Miscellaneous approaches

Our discussion so far has concerned mainly deployment approach classes where several self-deployment algorithms share the same principle. However, in the literature, we found few strategies that do not share their principle with the six classes discussed earlier. In this section, we will discuss such strategies.

Event-driven sensor deployment using self-organizing maps

Koutsougeras et al. [66] try to take into account the realistic consideration of the probability density for events to be sensed, termed as event-driven coverage. The objective is to distribute sensors so that the distribution density of the sensors matches that of the probability density of events to be sensed. In this context, they explore the concept of Self-Organizing Maps (SOM) to address the coverage problem. The SOM is a method for unsupervised learning, based on a grid of artificial neurons whose weight vectors are adapted to topologically match the distribution of vectors in a sample set that is used as exemplar or training set.

An algorithm based on market competition for wireless sensor network connectivity and coverage

Inspired by the market competition of human society, Peng et al. [82] proposed a distributed algorithm to handle the wireless sensor network connectivity and coverage issues. In this algorithm, the sensors in the network are seen as enterprises in economic activities, the interested areas are taken as resources, and network configurations are seen as market competitions. The advantages of the proposed algorithm are the reduced complexity of the exchanged information and the total distance traveled by sensors. However, the deployment environment is oversimplified.

2.4.8 A comparison between the various approaches

Table 2.3 provides a comparative summary of the merits and limits of the six classes of self-deployment approaches discussed above.

Table 2.3: A comparison between the various self-deployment approaches.

Class	**Critics**	**Merits**
Coverage pattern	Doesn't takes into account the presence of obstacles. We can't all the time adapt the shape of the RoI to the pattern grid. Excessive movement of the nodes to reach their pre-determined positions because it doesn't take into account the initial deployment. A uniform deployment is not preferable all the time.	The overall coverage of the RoI at the end of the process.
Virtual forces	The oscillation of the sensors around the same position. Collisions between nodes resulting from the simultaneous movement of several nodes. A uniform deployment is not preferable all the time.	Takes into account the initial deployment. Takes into account the presence of obstacles. We can control the coverage degree.
Grid quorum	A clustering is not feasible all the time. Strong assumption, i.e., pre-knowledge about the sensor field. The approach is centralized which evokes the single point failure problem and the scalability problem.	Balanced distribution of nodes in the different parts of the deployment area.
Fuzzy logic	Doesn't takes into account the presence of obstacles. The adjustment of the parameters is difficult. Not based on a mathematical formalism (no tools to validate the system of rules).	Takes into account the initial nodes deployment. Based on the human reasoning. The rules system constitute an effective solution to deal with the uncertainty and ambiguity which characterize the movement of the nodes.
Optimization	High temporal and spatial complexity. Problem of the local optimums. Hard to use (uses multiple parameters which are delicate to adjust).	Gives a solution close to the optimal solution.
Geometric computation	Don't takes into account the presence of obstacles. The convergence time can be very long.	Minimization of the coverage holes.

2.5 Conclusion

This Chapter provided an overview of WSNs deployment strategies. We discussed the three deployment approaches, namely, random deployment, deterministic deployment, and self-deployment. First, we presented a survey and taxonomy of random node placement in WSNs and we discussed some related work. Second, we focused on the deterministic deployment. We highlighted the two main components involved in the deterministic deployment strategy, namely, the sensor coverage models and the placement algorithms. For both components, we presented a survey and taxonomy that captures the fundamental differences among existing solutions. Finally, we considered the case of mobile sensors. We explored the published self-deployment strategies. We classified the identified strategies into six classes. For each class, we presented its principle and the most important algorithms belonging to the class. We further performed qualitative comparisons between the different approaches. Therefore, providing not only a complete view of the state-of-the-art, but also useful insights for selecting the deployment strategy most appropriate to the application at hand.

In the next Chapter, we will focus on the random deployment of WSNs. We will provide an in-depth analysis of intrinsic properties, thereby giving a holistic view of random deployment properties. Further, we will devise a practical random deployment strategy.

Chapter 3

Random Deployment of WSNs

3.1 Introduction

Deterministic deployment typically results in optimal efficiency; however, due to the size and density required to provide appropriate network coverage in large geographical areas, careful positioning of sensors is impractical. Furthermore, several applications of WSNs are expected to operate in hostile environments [12]. This makes deterministic deployment in some cases impossible and consequently, stochastic deployments become the only feasible alternative [12, 13, 120, 121, 143]. Random deployment can be achieved by dropping sensors from a plane, delivered in an artillery shell, rocket or missile, or throwing by a catapult from a ship board [12].

In the previous Chapter, we have surveyed and categorized existing random node placement strategies. Random deployment has been studied extensively over the last decade. Two approaches, are commonly used, namely: *theoretical studies* and *empirical studies*. The former is usually conducted for very few random deployment models under several simplifying assumptions [49, 119, 122, 148]. On the other hand, simulations analysis considers a couple of metrics and several random deployment models [56, 57, 79]. With both approaches, the effectiveness of random deployment has been shown to be certain. However, there has been little research on the analysis of *intrinsic properties of random deployment*. By intrinsic properties, we mean a full set of metrics such as coverage, connectivity, fault-tolerance, and lifespan. Furthermore, previous works fail to provide a holistic view of random deployment properties. Therefore, in this Chapter, we conduct an empirical study to analyze the intrinsic properties of several random node placement strategies. In order to capture the intrinsic properties of random deployment, first, we propose to use a

set of metrics. We distinguish two classes: (i) *deployment metrics* and (ii) *functional metrics*. Second, we carry out extensive simulations yielding a detailed analysis of the intrinsic properties of random node placement strategies. Finally, we devise a practical random deployment strategy that mixes the simple diffusion and the constant diffusion strategies, and we show that such hybridization ensures best performance. Experimental results provide helpful design guidelines in using random deployment strategies.

The rest of this Chapter is organized as follows. In Section 3.2, we present the simulations settings and analyze the experimental results in Sections 3.3 and 3.4. Section 3.5 discusses a practical random deployment strategy. In Section 3.6, we provide a conclusion and outline perspectives for further works.

3.2 Simulation Settings

To capture intrinsic properties of random deployment, we propose to use a set of metrics. We distinguish two classes: (i) *deployment metrics* and (ii) *functional metrics*. Deployment metrics are related to sensor-positions in the RoI, they are relatively independent of the protocol stack of the WSNs. We consider the following deployment metrics: coverage, connectivity and connected coverage. On the other hand, functional metrics are related to the network operations and therefore dependent on its protocol stack. We consider the following significant functional metrics: network lifespan, fault-tolerance and routing-related metrics.

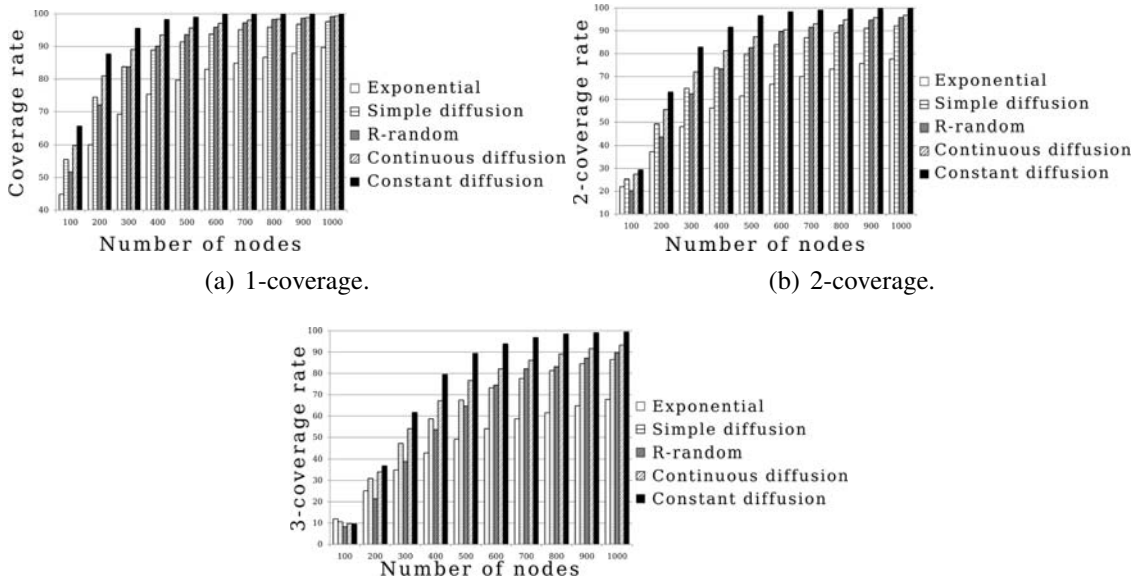
We have carried out a two-stage simulation assuming that (x, y) coordinates of the sensors are independent and the deployed sensors are homogeneous. The first simulation stage assesses deployment metrics using our simulation tool. In the second simulation stage, we have conducted extensive simulations using the network simulator NS2 [45] to analyze functional metrics. We use the well known SPEED [52] as a routing protocol. We repeated hundred times each simulation with variation of the seed; arithmetic means are reported on the graphs and tables. We observed that for a degree of confidence of 95%, the simulations results remain in (2-9)% compared to the average. The communication radius ($R_c = 40m$), sensing radius ($R_s = 18m$) and characteristics of the protocol stack have been chosen to represent an off-the-shelf device. Table 3.1 summarizes the parameters of our second stage simulation.

Table 3.1: The second scenario parameters.

Parameters	**Values**
Channel type	WirelessChannel
Radio-propagation model	Two-Ray Ground
Antenna type	OmniAntenna
Network interface type	Phy/WirelessPhy
MAC type	IEEE 802.11
Interface queue type	Queue/DropTail/PriQueue
Size of the queue	50 packets
Energy model	EnergyModel
txPower, rxPower	$0.8W$, $0.4W$

3.3 Deployment Metrics Analysis

3.3.1 Coverage Analysis



(a) 1-coverage.

(b) 2-coverage.

(c) 3-coverage.

Figure 3.1: Achieved k-coverage by different strategies.

Simple coverage

As depicted in Fig. 3.1(a), we see clearly that the constant diffusion ensures the highest coverage. This can be explained by the fact that the sensing capability of the network

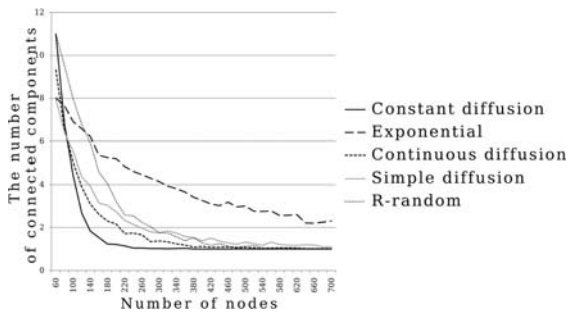
is fully utilized in the constant diffusion where sensors are uniformly distributed in the RoI. The continuous diffusion approaches the constant diffusion in term of coverage rate; moreover for a number of sensors higher than 600, the difference between them is 2%. The continuous diffusion obtains the highest coverage rate in comparison to those of other simple random node placement strategies. This is explained by the fact that the continuous diffusion is a combination of the Uniform and the Gaussian distributions.

When considering the relation between the number of deployed sensors and the coverage rate, we notice a threshold, in each model, from which the coverage variation is very small. In the constant diffusion, the coverage rate is 95% for 300 sensors and 100% for 600 sensors. Thus, 300 additional sensors were deployed to cover the remaining 5%. The first drawn conclusion is that full area coverage is very costly. This explains the fact that many researchers currently investigate *partial coverage* [71] and *sweep coverage* [69] because they allow a deployment with an acceptable quality of surveillance at a reduced cost. The second drawn conclusion is that the 300 additional sensors inevitably generate redundancies in coverage, which motivates the study of k-coverage.

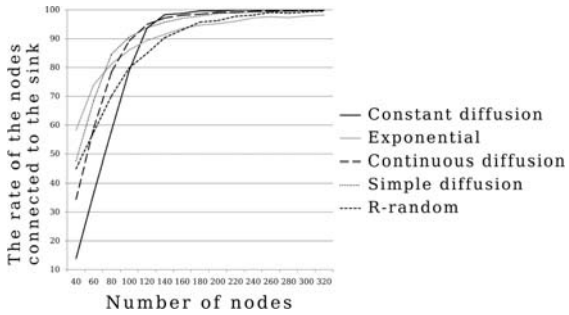
k-coverage

According to the results depicted in Fig. 3.1(b), if we consider the 2-coverage rate for 100 deployed sensors, we have the following order: the constant diffusion (29%), the continuous diffusion (27%), the simple diffusion (25%), the Exponential distribution (21%) and finally the R-random distribution (20%). By deploying 200 sensors, the 2-coverage rate of the R-random exceeds that of the Exponential distribution by 6%, for the other distributions the order remains the same. For 300 and 400 sensors, the order is always the same and we notice that the 2-coverage rate of the R-random reaches that of the simple diffusion quickly and it exceeds it for a number of sensors equal to 500.

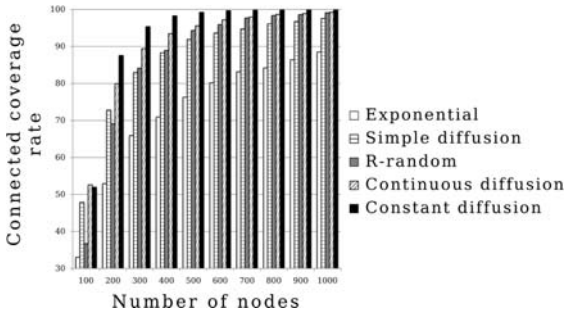
According to the results depicted in Figures 3.1(a)-3.1(c), we can classify the random node placement strategies according to the k-coverage rate as follow: (1) constant diffusion, (2) continuous diffusion, (3) R-random (4) simple diffusion and (5) Exponential distribution.



(a) The number of connected components.



(b) The % of sensors connected to the sink.



(c) Connected coverage.

Figure 3.2: Achieved connectivity by different strategies.

3.3.2 Connectivity Analysis

Number of connected components

According to the results depicted in Fig. 3.2(a), we can say that the number of deployed sensors is inversely proportional to the number of connected components. For 60 sensors, we have an average of 11, 11, 9, 8 and 8 connected components for the constant distribution, R-random, continuous diffusion, Exponential distribution, and simple diffusion, respectively. Starting from 180 sensors, we obtain only one connected component. In this case, all the sensors are connected to the sink. We know according to [49] that a network is strongly connected if the following condition is satisfied: $R_c \geq a\sqrt{\frac{C\ln(n)}{n}}$, where a is the length of the RoI, C is a constant ($C > \frac{1}{\pi}$), and n is the number of sensors. In our case, a network of $300m \times 300m$, a number of sensors greater than 180 and R_c of $40m$ represents a network with a good connectivity which explains the existence of only one connected component. The Exponential distribution always gives a higher number of connected components compared to the other distributions and it never reaches one. We can say that it is the worst model in terms of connectivity.

The rate of sensors connected to the sink

According to Fig. 3.2(b), the rate of sensors connected to the sink varies proportionally with the number of deployed sensors. For all placement strategies, we find that they provide an acceptable rate of sensors connected to the sink, especially after the deployment of 80 sensors as the worst among them exceeds 60%. This enables us to say that most of the deployed sensors are connected to the sink.

Connected coverage

On Fig. 3.2(c), for 100 sensors deployed, we note that the connected coverage rate has decreased significantly compared to the coverage rate, (-14%) for the constant diffusion, (-15%) for R-random, (-7.5%) for the simple diffusion, (-7%) for the continuous diffusion, and (-11%) for the Exponential distribution. These results show that there are some sensors covering parts of the RoI and they are unable to communicate their readings, they are isolated sensors and therefore useless. Increasing the number of sensors reduces the gap between coverage and connected coverage.

The first stage simulations results are summarized in Table 3.2. We use the following notation: Very Bad ($--$), Bad ($-$), Fair ($\pm$), Good ($+$), and Very Good ($++$).

Table 3.2: Summary of first stage simulations results.

	Coverage	**Connectivity**	**Connected coverage**
Constant diffusion	$++$	$++$	$++$
Continuous diffusion	$+$	$+$	$+$
R-random	$\pm$	$\pm$	$\pm$
Simple diffusion	$-$	$-$	$-$
Exponential distribution	$--$	$--$	$--$

3.4 Functional Metrics Analysis

3.4.1 Routing-related Metrics Analysis

We generate sensors-to-sink CBR traffic, such as sources are randomly selected, and we evaluate routing-related metrics. Table 3.3 summarizes the obtained results. We notice that the three considered models provide a high delivery rate (DR) (over 94%). It should be

pointed out that for the same number of sensors the three models provide a high connected coverage rate.

As the energy is considered as a critical resource in WSNs, research in this area tends to minimize energy consumption to increase the network lifespan. The consumed energy (CE) by a uniformly deployed network is much less than in the case of a simple diffusion or R-random, these results do not determine the model that consumes more energy. To get a clear idea, we calculate the energy consumed per packet (CEP). According to the obtained results (cf. Table 3.3), we can say that the constant diffusion uses less energy than the other two models to route a packet to the sink. Simple diffusion and R-random consume almost twice the energy consumed by the constant diffusion to route a packet to the destination. This is explained by the fact that each node has more routing neighbors' candidates in the case of a constant diffusion; therefore the routing protocol generates few control packets which leads to reduced energy consumption.

Obtained results show that the simple diffusion has the smallest end-to-end delay (EED); the other two models have almost the same delay. This is explained by the fact that in the simple diffusion when approaching the sink, each sensor has more routing neighbors' candidates which allows the routing protocol to have a larger set of routing candidates and thus a better selection which reduces the EED.

Table 3.3: Routing-related simulations results.

	Constant diffusion	**Simple diffusion**	**R-random**
DR	97.65	96.46	94.29
CE	171.62	318.72	279.04
CEP	0.046	0.073	0.078
EED	214.03	187.19	208.61

3.4.2 Fault-Tolerance Analysis

In this section, we evaluate fault-tolerance related to detection errors, transient and global errors. In the simulations, 1600 targets are generated periodically in the RoI in random positions. A target is detected if its position is within the detection range of a sensor. Each sensor that detects a target sends a detection message to the sink. In what follows, we consider a target as detected if it was detected by at least one sensor and at least one detection message was received by the sink.

Fault-tolerance related to detection errors (FT-DE)

We vary the probability of detection errors from 0.2 to 0.8, Fig. 3.3 shows the variation of detection rate as a function of detection errors.

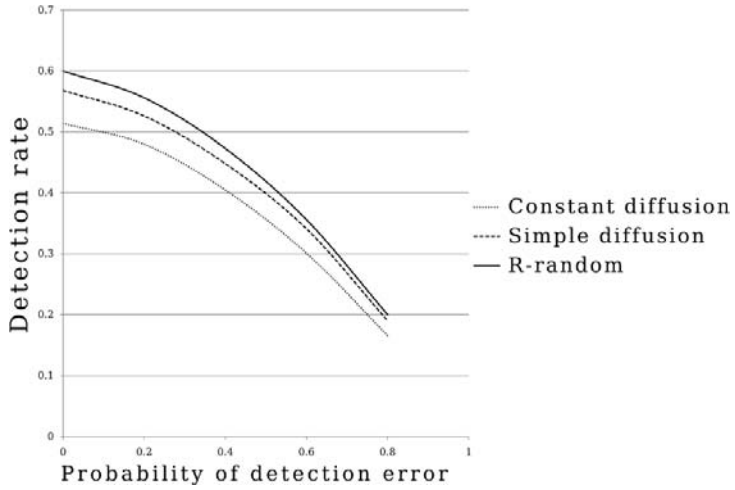


Figure 3.3: Fault-tolerance related to detection errors.

The R-random distribution provides a higher detection rate than other distributions. By increasing the probability of detection errors, we notice that the detection rate decreases, *i.e.*, the detection rate is inversely proportional to the probability of detection errors. At first sight, these results may not seem very logical, since the constant diffusion provides the highest coverage and connected coverage in comparison to the other two models, so it should provides the highest detection rate. To explain this result, a very important parameter is involved which is the problem of the *non-uniform energy consumption* also known as the energy hole problem [133]. Over time, sensors deplete their energy as they send and route messages toward the sink. Sensors surrounding the sink consume more energy than the others, thus they quickly exhaust their energy; this is called the energy hole problem. Due to this problem, the signaling messages sent by other sensors will be transmitted to the sink with a delivery rate that decreases with time, until the sink becomes isolated from the network because of the death of all its neighboring sensors. Recall that R-random and simple diffusion have a high sensors density around the sink which allow them to be more resistant to the problem of the non-uniform energy consumption; consequently, they ensure a greater detection rate in comparison to the other models.

Fault-tolerance related to transient and global errors

Regarding fault-tolerance related to global errors (FT-GE), we simulate the random failure of 12% of the sensors after a time t (after the generation of 200 targets). For fault-tolerance related to transient errors (FT-TE), we simulate the temporary failure of random sensors for a specific time intervals. More precisely, we simulate the temporary failure of 5 sensors during the time of generation of 100 targets; this scenario will be repeated after each generation of 200 targets. We compare the obtained results with those of the reference case (no sensor falls down). Table 3.4 shows the differences due to different types of errors compared to the reference case. We notice that the impact of errors on the constant diffusion is much larger than the other two models. The simple diffusion has a greater robustness against different types of errors and the R-random provides results that are quite similar to those of the simple diffusion.

Table 3.4: The differences due to different errors.

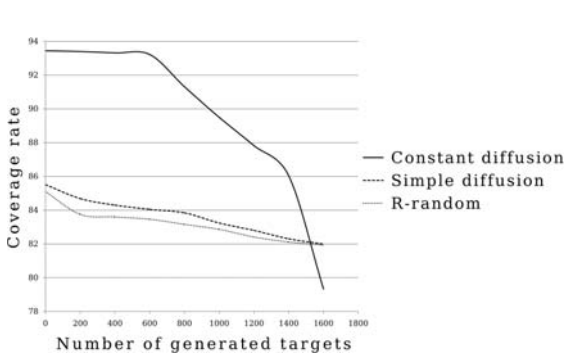
	Transient errors	**Global errors**
Constant diffusion	18.10%	17.60%
Simple diffusion	7.10%	9.80%
R-random	10.50%	9.30%

With 130 deployed sensors, the k-coverage rate is reduced (according to the results obtained in the first stage simulation), so the failures of sensors impact the functioning of the network. It reduces the coverage, so fewer targets are detected, and reduces connectivity, so fewer packets are successfully routed toward the sink. The importance of the k-coverage and k-connectivity appears clearly in the case of errors that disrupt the proper functioning of the network. Depending on the application, a redundancy in coverage and connectivity must be provided to ensure fault-tolerance.

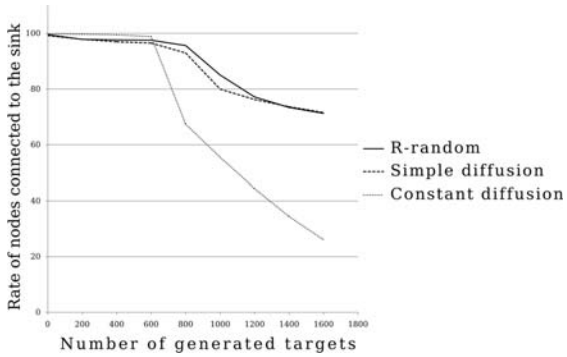
3.4.3 Network Lifespan Analysis

Network lifespan based on coverage (NL-CV)

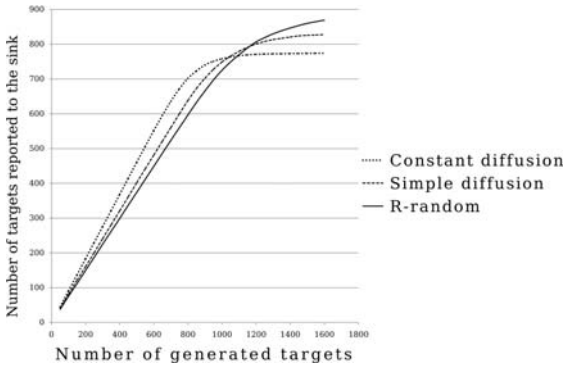
First, we consider network lifespan based on coverage. When setting up the network, the coverage rate of the constant diffusion is superior to those of other models, which coincides with the previous results. After generating a number of targets, sensors' battery exhaustion affects the network coverage; the constant diffusion is highly affected (see



(a) Network lifespan based on coverage.



(b) Network lifespan based on connectivity.



(c) Network lifespan based on the detection rate.

Figure 3.4: Network lifespan for different strategies.

Fig. 3.4(a)). The coverage rate decreases less strongly in R-random and simple diffusion due to the high density of sensors around the sink, which ensure the *k*-coverage of this area. On the other side, when sensors are uniformly distributed, sensors death expands the not-covered area quickly. In terms of network coverage, both models R-random and simple diffusion provide a longer network lifespan than that offered by the constant diffusion.

Network lifespan based on connectivity (NL-CN)

Ensuring a quite good coverage during network operation does not mean providing a high detection rate, the absence of paths to the sink makes the signalization of detected targets impossible. We evaluate the network lifespan based on connectivity. As depicted in Fig. 3.4(b), at the beginning, the three models provide a high connectivity rate. Over time, the number of sensors connected to the sink decreases rapidly in the constant diffusion to reach 23% at the end of the simulation. The other two models guarantee a high connectivity rate that exceeds 70%. In terms of network connectivity, both models R-random and

simple diffusion provide a longer network lifespan compared to that offered by the constant diffusion.

Network lifespan based on the detection rate (NL-DR)

Fig. 3.4(c) illustrates the number of targets reported to the sink against generated targets. The curve of constant diffusion tends to a constant value. In this case, the network has reached a saturation mode; no detection message can reach the isolated sink. In the other two models, the number of reported targets continues to grow, even with the death of some sensors surrounding the sink. R-random model provides a better detection rate than the simple diffusion. This could be justified by the fact that in the simple diffusion 68.3% of the sensors are at a distance $d \leq \sigma$, so the density in the rest of the network will be less than that of the R-random distribution which places sensors uniformly according to the polar coordinates (r, θ). The R-random ensures the longest network lifespan in terms of detection rate, followed by simple diffusion, and finally constant diffusion.

The second stage simulations results are summarized in Table 3.5.

Table 3.5: Summary of second stage simulations results.

	Constant diffusion	**Simple diffusion**	**R-random**
DR	+	+	+
CEP	+	±	±
EED	±	+	±
FT-DE	−	±	+
FT-TE	−	+	±
FT-GE	−	+	+
NL-CV	±	+	+
NL-CN	−	+	+
NL-DR	−	±	+

3.5 A Practical Random Deployment Strategy

According to the results discussed in Sections 3.3 and 3.4, in one hand, the constant diffusion guarantees a high coverage and connectivity rates during the deployment of sensors. However, after a while these performances reach a lower level compared to the other models such as simple diffusion or R-random. The constant diffusion reaches quickly the

saturation mode, where no message could be transmitted to the sink located at the center of the RoI. In this case, the sink becomes isolated from the rest of the network. In the other hand, simple diffusion and R-random place a large number of sensors around the sink which prolongs the network lifespan in terms of detection rate and connected coverage rate.

We propose a hybridization of the simple diffusion model that places a large number of sensors around the sink and the constant diffusion that provides high coverage and connectivity rates. We chose the simple diffusion instead of the R-random whereas the latter ensures a greater detection rate, because the simple diffusion is characterized by the standard deviation σ which makes it possible to control the density of the sensors around the sink. We call such hybridization the *hybrid diffusion*.

Definition 1. *The hybrid diffusion of N sensors is defined as the deployment of* $\alpha.N$ *sensors according to the simple diffusion strategy and* $\beta.N$ *sensors according to the constant diffusion model where* $0 < \alpha < 1$, $0 < \beta < 1$, *and* $\alpha + \beta = 1$.

Fig. 3.5 shows an example of a hybrid diffusion ($\alpha = 0.35, \beta = 0.65$) of 300 sensors in a circular RoI of a radius $R = 150m$.

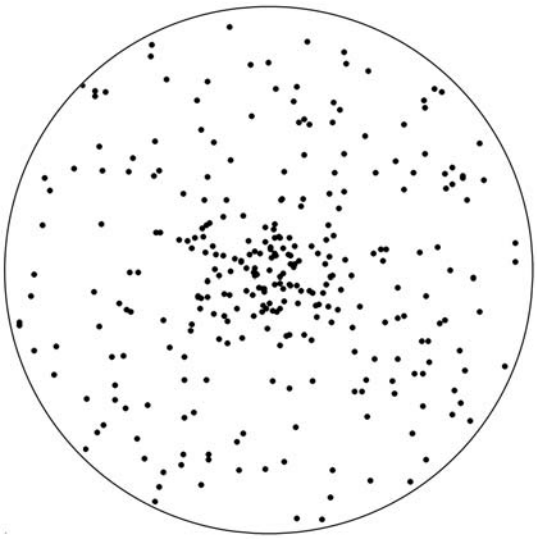


Figure 3.5: An example of the hybrid diffusion.

We conduct extensive simulations to evaluate the performance of the hybrid diffusion in comparison to those of other deployment strategies. In what follows, we present the obtained results for a circular RoI of a radius $R = 150m$, $\alpha = 7/20$, $\beta = 13/20$ and $\sigma = 0.3 * (R/2)$.

3.5.1 Deployment Metrics Analysis

Simple coverage

We vary the number of deployed sensors (N) from 100 to 600. Fig. 3.6 shows the coverage rate of the hybrid diffusion in comparison to those of simple diffusion, constant diffusion and R-random. According to the obtained results, the hybrid diffusion approaches the constant distribution in terms of coverage and outperforms the other two models.

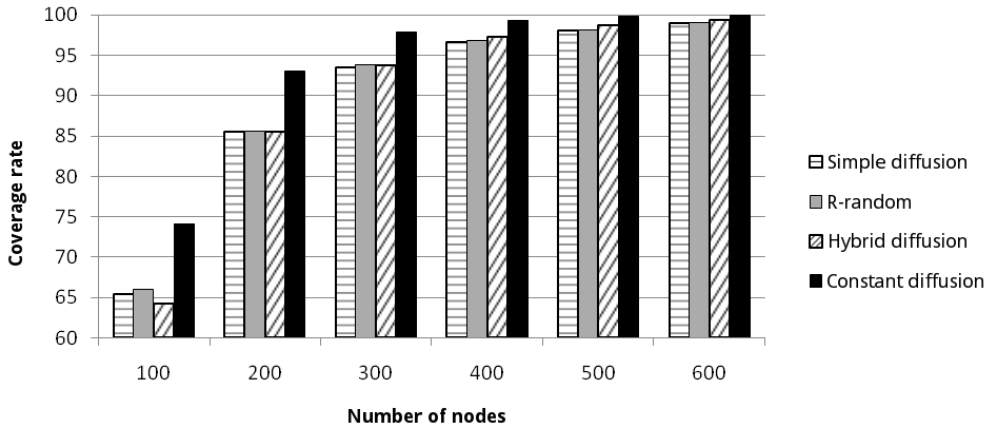


Figure 3.6: Achieved coverage by different strategies.

The rate of sensors connected to the sink

Fig. 3.7 shows the rate of sensors connected to the sink when using four deployment strategies, namely: hybrid diffusion, simple diffusion, constant diffusion and R-random.

For 60 sensors, the hybrid diffusion ensures better performance in comparison to the constant diffusion. We know from previous results (cf. Section 3.3.2) that the constant diffusion gives the lowest rate of sensors connected to the sink, our model improves this metric (a difference of 32% in this case). This is explained by the fact that our model has the properties of simple diffusion which provides a high rate of sensors connected to the sink even with a reduced number of sensors. When increasing the number of sensors, our model gives a similar rate of sensors connected to the sink to those of other models. Based on these results, we can deduce that to ensure a high connected coverage rate (over 90%), our model is the best candidate after the constant diffusion.

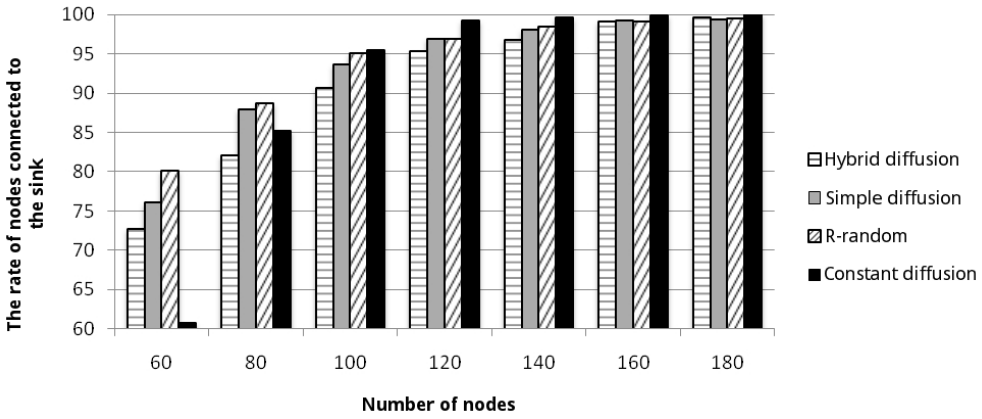


Figure 3.7: Achieved connectivity by different strategies.

3.5.2 Functional Metrics Analysis

Fault-tolerance related to detection errors

Regrading fault-tolerance related to detection errors, we compare our hybrid model to R-random model which gave the highest detection rate in the previous simulations (cf. Section 3.4.2). According to the results depicted in Fig. 3.8, for small detection errors probabilities both models have the same results. When increasing detection errors probability, our hybrid model outperforms the R-random model.

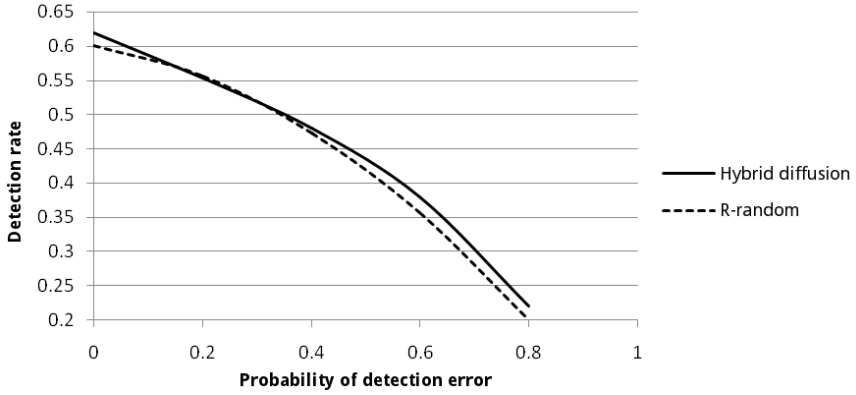


Figure 3.8: Fault-tolerance related to detection errors.

Fault-tolerance related to transient and global errors

We compare our hybrid model to simple diffusion and R-random in terms of fault-tolerance related to transient and global errors with the same scenarios as those of Section 3.4.2. Table 3.6 shows the differences due to different types of errors in comparison to the reference case.

Table 3.6: The differences due to different errors.

	Transient errors	**Global errors**
Hybrid diffusion	10.8%	8.6%
Simple diffusion	7.1%	9.8%
R-random	10.5%	9.3%

The performance of our hybrid model are close to those of other models. Indeed, the performance of hybrid diffusion is almost similar to that of R-random in the case of transient errors. In the case of global errors, hybrid diffusion achieves the best performance.

Network lifespan based on the detection rate

Fig. 3.9 presents the number of targets reported to the sink against generated targets. The obtained results are compared to those of R-random which gave the highest detection rate compared to other models. We notice that our hybrid model achieves a higher detection rate than R-random model and ensures a greater network lifespan by providing a better detection rate.

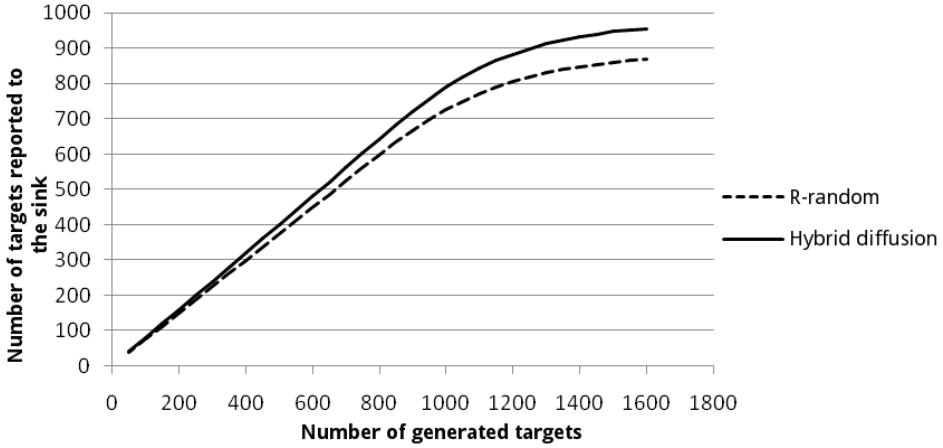


Figure 3.9: Network lifespan based on the detection rate.

Network lifespan based on the number of alive sensors

Fig. 3.10 shows the evolution of the number of the dead sensors as a function of time. In the hybrid model, the number of dead sensors is slightly higher than that of R-random. That's because our hybrid model gives the greatest detection rate, therefore a greater number of packets routed to the sink, which causes the exhaustion of a great number of sensors especially those around the sink. According to the definition of the network lifespan based on the number of alive sensors, the hybrid model tends to exhaust a greater number of sensors compared to the other models, and thus presents a slightly shorter network lifespan.

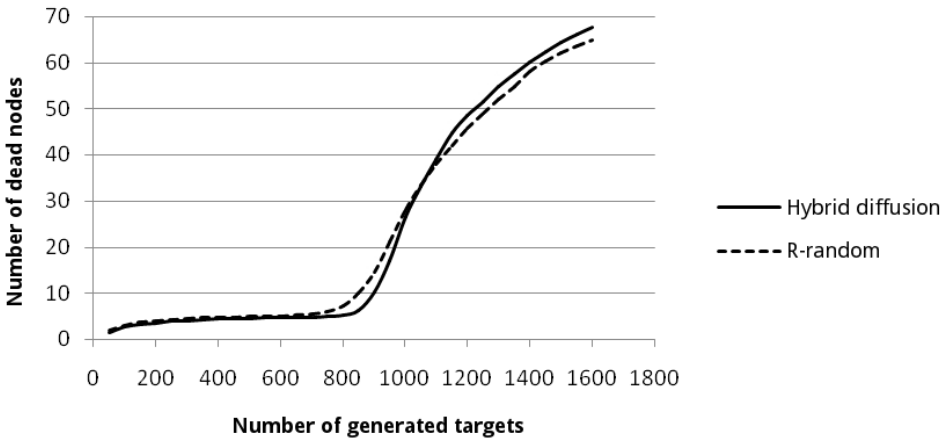


Figure 3.10: Network lifespan based on the number of alive sensors.

Network lifespan based on coverage

According to the results depicted in Fig. 3.11, initially, our hybrid model provides a lower coverage rate in comparison to that of constant diffusion and slightly higher coverage rate in comparison to those of simple diffusion and R-random. Over time, our hybrid model outperforms the three other models. We also notice that our hybrid model keeps a nearly constant coverage, which is an interesting property.

Network lifespan based on connectivity

Fig. 3.12 shows the rate of sensors connected to the sink as a function of time. We notice that the hybrid model provides the highest rate of sensors connected to the sink, which explains the highest detection rate obtained previously. According to the definition

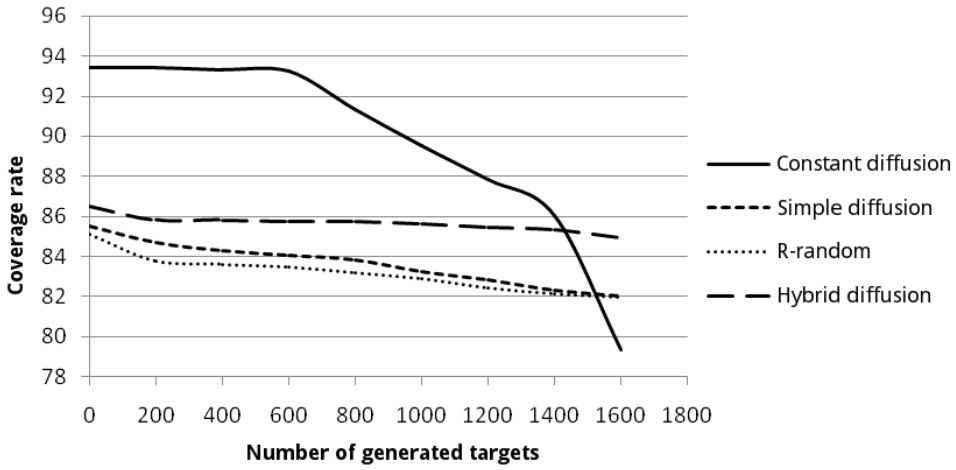


Figure 3.11: Network lifespan based on coverage.

of the network lifespan based on connectivity, the hybrid diffusion ensures the longest network lifespan.

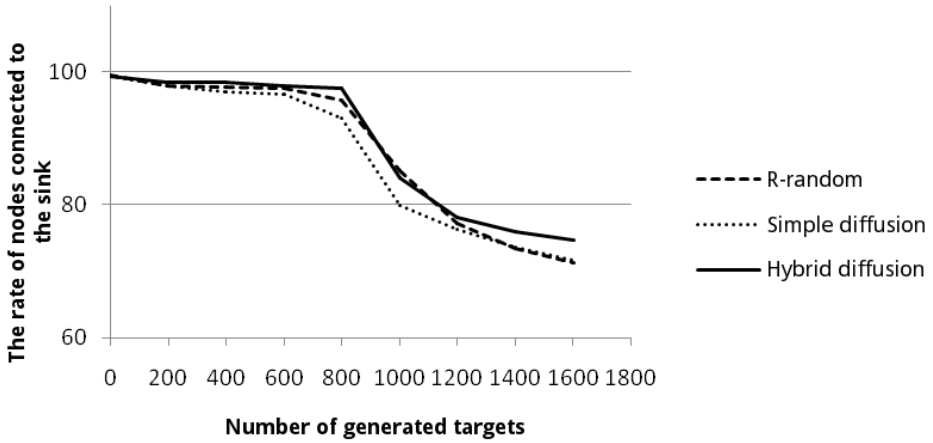


Figure 3.12: Network lifespan based on connectivity.

3.6 Conclusion

In this Chapter, an empirical study has been carried out yielding a detailed analysis of intrinsic properties of random node placement in WSNs and provides a holistic view of random deployment properties. The obtained results give helpful design guidelines in using random deployment strategies. Based on our findings, we further proposed a hybridization

of the simple diffusion model that places a large number of sensors around the sink and the constant diffusion that provides high coverage and connectivity rates. We also investigated the performance of the hybrid model and we showed that such hybridization ensures better performance.

Although, the feasibility of random deployment is appealing, random deployment is not cost effective as compared to the deterministic deployment. In the next Chapter, we will focus on the deterministic deployment of WSNs. In contrast to the existing literature, we will exploit the evidence theory also known as Dempster-Shafer theory [98] to design better deterministic deployment strategies.

Chapter 4

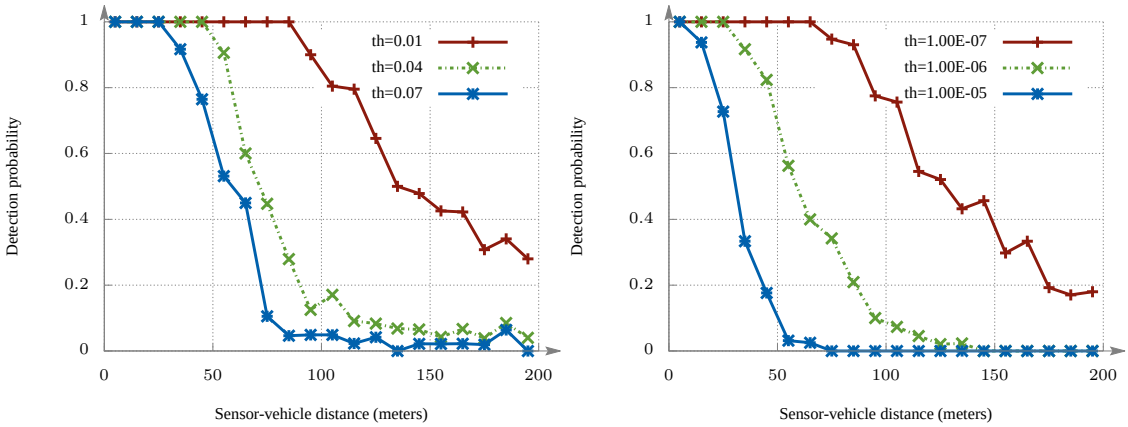
Uncertainty-aware WSNs Deployment

4.1 Introduction

In the previous Chapter, we have explored the random deployment of WSNs while considering a rich set of metrics. The main advantage of random deployment is its feasibility. In harsh environments such as disaster region, random deployment often becomes the only option. However, additional sensors are needed in random deployment as compared to the deterministic deployment. For that, in this Chapter we focus on deterministic deployment while considering, in the first place, the most important metric, namely: coverage.

As mentioned in Chapter 2, in deterministic deployment, the sensor locations are precomputed based on a "perfect" sensor coverage model. The most widely used sensor coverage model in the literature is the *binary coverage model* (often simplified as the disk model), which assumes that an event happening within the sensing radius of a node is always detected, while any event outside this disk is assumed not to be detected (cf. Section 2.3.1). However, sensors may not always provide reliable information, either due to operational tolerance levels or environmental factors. In reality, it is well known that there is an inherent uncertainty and imprecision associated with sensor readings. Some practical examples include AWAIRS at UCLA/RSC [6], Smart Dust at UC Berkeley [60], the USC-ISI network [44], and the DARPA SensIT systems/networks [41]. To illustrate the inaccuracy of the binary coverage model, we plot in Fig. 4.1 the average sensing performance of two types of sensors, namely: acoustic (Fig. 4.1(a)) and seismic (Fig. 4.1(b)) sensors using the data traces collected from a real vehicle detection experiment [41]. In the experiment, the sensor detects moving vehicles by comparing its signal energy measurement against a threshold (denoted by *th*). As shown in Figures 4.1(a)-4.1(b), for both

modalities (acoustic and seismic) there is no clear cut-off boundary between successful and unsuccessful detection.



(a) Acoustic sensors. (b) Seismic sensors.

Figure 4.1: Detection probability *vs.* the sensor-vehicle distance.

These results confirm that, for many types of sensors, a longer distance between the sensor and the target generally implies a greater loss in the signal strength or a lower signal-to-noise ratio [27, 120]. This issue impinges upon the success of any WSN deployment, as the sensor locations are computed while ignoring this issue and it is imperative to have practical considerations at the design stage to anticipate this sensing behavior. This suggests that we can build an abstract sensor coverage model to express the uncertainty in sensor readings. Thus, a key challenge in the deployment of WSNs is to consider the imperfections associated with sensor readings. Or in other words, to consider the *uncertainty-aware WSNs deployment*.

To avoid the aforementioned limitations, recent research works have adopted a probabilistic coverage model [8, 9, 38, 39, 149, 152] that considers the probabilistic sensing behavior of sensors. Although this model captures the behavior of sensors more realistically, it does not exploit the collaboration among sensors. In addition, it remains limited. For example, the reliability of sensors in the deployment process still not taken into account in this kind of model [120]. Thus, a generic and flexible coverage model that expresses the uncertainty in sensor readings is important, and it is one of our motivations in this work.

The representation and management of the imperfections associated with sensor readings can be done using mathematical formalisms such as fuzzy sets theory [146] in the case of imprecise data or Probability theory [65], Possibility theory [147], and Belief Functions theory [98] in the case of uncertain data. The probabilistic coverage models

[8, 9, 38, 39, 149, 152] exploit the Probability theory to handle the imperfections associated with sensor readings. However, these approaches suffer from major limitations. For instance, uncertainty cannot be expressed in the Probability theory, except by an equidistribution of probabilities for the different hypotheses. The advantage of the belief functions theory [98] is its ability to separate the two sources of imperfection, imprecision and uncertainty, and its fairly simple modeling of doubt and lack of information. These advantages seem valuable in the context of wireless sensor networks deployment.

Because the deployment should be application-driven, we focus on a particular application of WSNs: target/event detection. For instance, to detect a target moving through the RoI, sensors have to make local observations of their surrounding environment and collaborate to produce a global decision that reflects the status of the RoI. This collaboration requires the local processing of observations, communication between nodes, and information fusion [114].

In this Chapter, we explore the problem of uncertainty-aware WSNs deployment. First, we detail a new evidence-based sensor coverage model based on the Transferable Belief Model (TBM) [106], a model for the representation of quantified uncertainty based on belief functions. This abstract sensor coverage model not only considers the imperfections associated with sensor readings, but can be easily extended to include deployment-related issues, such as sensor reliability. Here, we focus our attention on the isotropic class of sensors, deployed in a field to detect certain activities. An example of such a scenario may be seismic sensors deployed in a battlefield to detect enemy movements. We also discuss some related issues such as sensor reliability and challenging environments. Second, as an example of potential applications, we define a detection coverage model build on our evidence-based coverage model. Third, we present a flexible framework for collaborative target/event detection within the Transferable Belief Model (TBM) [106]. We formalize four different forms of the uncertainty-aware wireless sensor networks deployment problem. Fourth, using the developed framework, we conceive five uncertainty-aware deployment algorithms that determine the minimum number of sensors and their locations to satisfy the user requirements. Finally, we provide experimental results based on both synthetic data sets and data traces collected in a real deployment for military vehicle detection to demonstrate the ability of our approach to achieve an efficient WSNs deployment by exploiting a collaborative target/event detection scheme.

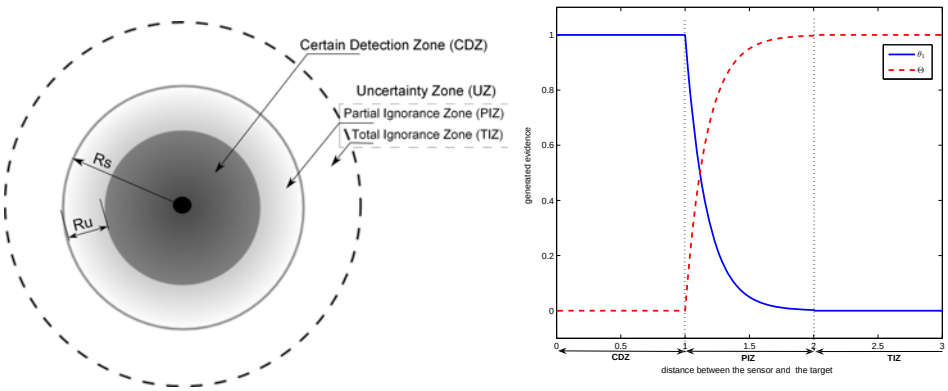
4.2 Evidence-based Sensor Coverage Model

Sensor coverage models are abstraction models trying to quantify how well sensors can sense physical phenomena at some locations in the RoI, or in other words, how well sensors can cover such locations [120].

The TBM introduced in [106], is a particular interpretation of the so-called Dempster-Shafer theory [98]. This uncertain reasoning framework can be seen as an extension of Bayesian probability theory. The TBM is a two-level model: there is a *credal level* where beliefs are entertained and a *pignistic level* where beliefs are used to make decisions. In what follows, we assume the reader to be familiar with the TBM (please see Appendix A for an overview).

4.2.1 Evidence construction

Two states are required to specify whether a space point $p \in RoI$ is covered: $\theta_1(covered)$ and $\theta_0(not\ covered)$. Thus, the Frame of Discernment (FoD) is the set $\Theta = \{\theta_0, \theta_1\}$. Let s be a sensor, R_s be its sensing range and R_u be a distance ($0 \leq R_u \leq R_s$) as illustrated in Fig. 4.2.



(a) Illustration of the different zones.

(b) Illustration of the relation between the generated evidence and sensor-target distance.

Figure 4.2: Evidence-based sensor coverage model.

Definition 2. *the Certain Detection Zone (CDZ) of s is a disk centered at s with a radius of* $(R_s - R_u)$.

Definition 3. *the Uncertainty Zone (UZ) of s is the complement of its CDZ relative to the RoI;* $UZ = \overline{CDZ}_{RoI}$. *UZ is divided into two sub-zones: the Partial Ignorance Zone (PIZ) and the Total Ignorance Zone (TIZ).*

Definition 4. *the Zone of Partial Ignorance (PIZ) is defined as an annulus centered at s with an inner radius of* $R_s - R_u$ *and an outer radius of* R_s.

Definition 5. *the Zone of Total Ignorance (TIZ) is defined as the complement of PIZ relative to* $(RoI - CDZ)$; $TIZ = \overline{PIZ_{RoI-CDZ}}$.

Each sensor s provides information on the coverage of a space point $p \in RoI$ with a belief $b_{s/p}$. The complementary information $1 - b_{s/p}$ is assigned to the whole FoD because it encodes the sensor ignorance. The output from the sensor s about a space point $p \in RoI$ can thus be represented as a basic belief assignment (*bba*) $m_{s/p}$ with two focal sets: the singleton $\{\theta_1\}$ and the FoD Θ.

$$
\begin{aligned}
&m_{s/p}(\{\theta_1\}) = b_{s/p},\ b_{s/p} \in [0, 1] \\
&m_{s/p}(\Theta) = 1 - b_{s/p}, \\
&m_{s/p}(\emptyset) = 0
\end{aligned}
$$

The idea behind this formulation is that we consider that sensory data collected by a sensor is a support given to the hypothesis $\{\theta_1\}$. A sensor cannot support the hypothesis $\{\theta_0\}$, but it can support Θ, which represents its degree of ignorance. This enables the sensor to not be in contradiction with itself, i.e., to not have mass on the hypothesis $\{\theta_1\}$ and on the hypothesis $\{\theta_0\}$ simultaneously.

The parameter $1 - b_{s/p}$ reflects the sensor's degree of ignorance. This model reflects the behavior of range sensing devices such as infrared and ultrasound sensors. One can assume, as in the truncated attenuated disk model [120], that $b_{s/p} = e^{-\delta(d(s,p)-(R_s-R_u))^\beta}$. where $d(s, p)$ is the Euclidean distance between a sensor s and a target p, R_s is the sensing range, R_u is the uncertain range, δ is a sensor technology related parameter and β is an event characteristic-dependent parameter. An example of this model is depicted in Fig. 4.2(b).

More precisely, depending on the distance between a sensor s and a space point p, the sensor s provides the following belief functions:

– Within CDZ, s produces a *categorical belief function* defined as follows:

$$
\begin{aligned}
&m_{s/p}(\{\theta_1\}) = 1 \\
&m_{s/p}(\emptyset) = m_{s/p}(\Theta) = 0
\end{aligned}
$$

– Within PIZ, s produces a *simple support function* defined as follows:

$$\begin{aligned} &m_{s/p}(\{\theta_1\}) = b_{s/p},\ b_{s/p} \in [0,1] \\ &m_{s/p}(\Theta) = 1 - b_{s/p}, \\ &m_{s/p}(\emptyset) = 0 \end{aligned} \tag{4.1}$$

– Within TIZ, s produces a *vacuous belief function* defined as follows:

$$\begin{aligned} m_{s/p}(\Theta) &= 1 \\ m_{s/p}(\emptyset) &= 0 \end{aligned}$$

4.2.2 Decision Making

Relatively to a space point p, a sensor s provides $m_{s/p}$ as a belief function. To decide whether p is covered by s, the pignistic transformation of $m_{s/p}$ (denoted by $BetP_{s/p}$) is constructed. The decision is based on selecting the hypothesis $\hat{\theta}$ with the largest pignistic probability: $\hat{\theta} = \arg\max_{i=0,1} BetP_{s/p}(\{\theta_i\})$.

Definition 6. *A space point p is covered by a sensor s if:*

$$\begin{aligned} \hat{\theta} &= \theta_1 \\ BetP_{s/p}(\{\theta_1\}) &\geq Th_p \end{aligned} \tag{4.2}$$

The threshold (Th_p) value is an application-specific user-specified parameter. For example, in critical applications where a false negative is considered harmful, Th_p should be close to 1.

4.2.3 Handling Deployment-related Issues

In this section, two possible extensions are discussed to show the many strengths of the proposed model compared to the probabilistic model. More precisely, we show how the proposed model can be extended to include deployment-related issues, such as *sensor reliability* and *challenging environments*.

The behavior of sensors is strongly scenario dependent, referring to i) the quality of the acquired data, ii) the reliability of each of the deployed sensors under particular weather conditions and types of RoIs, and iii) the type of target under analysis [43, 106]. In this discussion, we consider three sources of *sensor decision unreliability*: i) sensor reliability, ii) challenging environments, and iii) sensor reliability and challenging environments.

Sensor Reliability

In this case, sensors are vulnerable to misreading or malfunctioning due to their quality. We consider such sensors as only partially reliable. For $\alpha \in [0, 1]$, let $(1 - \alpha)$ be the degree of "confidence" we assign to the sensor. It can be encoded into a *bba* defined on the set $\{reliable, not\ reliable\}$ such that:

$$m(x) = \begin{cases} 1 - \alpha & \text{if } x = \{reliable\}, \\ \alpha & \text{if } x = \{not\ reliable\} \end{cases} \tag{4.3}$$

Suppose that the *bba* m on Θ represents the sensor report about the actual value of Θ. The result of combining the sensor report with the *bba* given in Equation (4.3) is a new *bba* denoted by m^α, defined as:

$$\begin{aligned} m^\alpha(A) &= (1 - \alpha)m(A) \quad for \ A \subset \Theta \\ m^\alpha(\Theta) &= \alpha + (1 - \alpha)m(\Theta) \end{aligned} \tag{4.4}$$

Sensor reliability can be different from one sensor to another and depend on a priori knowledge. Discounting factors associated to the sensors will be used in the evidence combination step.

Challenging Environments

Although sensors may be of good quality and provide accurate readings, external factors can greatly influence their correct functioning [43]. In this case, sensors are vulnerable to misreading or malfunctioning due to their locations in the RoI; we call such locations "*challenging locations*". A fully reliable sensor is considered as only partially reliable if it is deployed in a challenging location.

Let $\beta \in [0, 1]$ be the degree of "challenge" that we assign to a location p in the RoI. It can be encoded into a *bba* defined on the set $\{challenging, not\ challenging\}$ such that:

$$m(x) = \begin{cases} \beta & \text{if } x = \{challenging\}, \\ 1 - \beta & \text{if } x = \{not\ challenging\} \end{cases} \tag{4.5}$$

Suppose that the *bba* m on Θ represents the report of the sensor (deployed at location p) about the actual value of Θ. The result of combining the sensor report with the *bba* given in Equation (4.5) is a new *bba* denoted by m^β, defined as:

$$m^{\beta}(A) = \quad (1-\beta)m(A) \quad for \ A \subset \Theta$$
$$m^{\beta}(\Theta) = \quad \beta + (1-\beta)m(\Theta)$$

Thus, discounting factors will be associated with deployment points. The available a priori knowledge on environmental factors is used to compute the discounting factors. In the combination step, the received evidence will be discounted using the associated discounting factors.

Sensor Reliability and Challenging Environments

In this case, some sensors are more vulnerable to misreading or malfunctioning due to their quality and/or their location in the RoI. Suppose the *bba* m on Θ represents the sensor's report about the actual value of Θ. The result of combining the sensor's report with the *bba* given in Equation (4.3) and the *bba* given in Equation (4.5) is a new *bba* denoted by $m^{\alpha\beta}$, defined as:

$$m^{\alpha\beta}(A) = \quad (1-\alpha)(1-\beta)m(A) \quad for \ A \subset \Theta$$
$$m^{\alpha\beta}(\Theta) = \quad \beta + \alpha(1-\beta) + (1-\alpha)(1-\beta)m(\Theta)$$

Discounting factors will be associated with sensors and deployment points.

4.3 Application: Evidence-based Detection Coverage Model

One current application of WSNs is target/event detection; the sensors collaborate to arrive at a consensus decision as to whether a target is present in the RoI. We consider a TBM-based approach for reaching this consensus: one of the sensors (called a fusion center: it can be a sink, a cluster-head or any sensor), gathers the evidence from the other sensors, combines the evidence and decides whether an event is present. In this section, we define a detection coverage model build on our evidence-based coverage model presented in the previous section.

4.3.1 Evidence Combination

Using a conjunctive combination, the fusion center constructs new evidence representing the consensus of the evidence obtained from sensors. To obtain evidence from sensors, one can imagine several scenarios:

– *On-demand scenario*: A request is sent to a specific area of the RoI, and the sensors involved send back their observations to a fusion center;
– *Reactive scenario*: Upon the detection of an event, some sensors send their observations to a fusion center;
– *Proactive scenario*: Periodically, some sensors send their observations to a fusion center.

For N sensors, the combination of the N *bbas* $m_{1/p}, \ldots, m_{N/p}$ using the unnormalized Dempster's rule yields a *bba* m_p with 2 focal sets: the singleton $\{\theta_1\}$ and the FoD Θ. This *bba* has the following expression:

$$\begin{aligned} m_p(\{\theta_1\}) &= \prod_{i=1}^{N} b_{i/p} + \underbrace{b_{j/p} b_{k/p} \ldots b_{L/p}}_{\substack{1:N-1 \quad terms \\ j,k,\ldots,L=1\ldots N \\ j\neq k\neq \ldots \neq L\neq i}} \sum_{i=1}^{N} (1 - b_{i/p}) \\ m_p(\Theta) &= \prod_{i=1}^{N} (1 - b_{i/p}) \end{aligned} \tag{4.6}$$

Note here that the evidence can be fused in any order because the combination is performed by the conjunctive rule (which is both commutative and associative).

Example 1: let us consider a hypothetical system composed of 2 sensors. The FoD is then $\Theta = \{\theta_0, \theta_1\}$. The different *bbas* provided by each sensor are:

Sensor 1

$$\begin{aligned} m_{1/p}(\{\theta_1\}) &= b_{1/p} \\ m_{1/p}(\Theta) &= 1 - b_{1/p} \\ m_{1/p}(\{\emptyset\}) &= 0 \end{aligned}$$

Sensor 2

$$\begin{aligned} m_{2/p}(\{\theta_1\}) &= b_{2/p} \\ m_{2/p}(\Theta) &= 1 - b_{2/p} \\ m_{2/p}(\{\emptyset\}) &= 0 \end{aligned}$$

The combined *bba* is then

$$
\begin{aligned}
m_p(\{\theta_1\}) &= b_{1/p}b_{2/p} + b_{2/p}(1 - b_{1/p}) + b_{1/p}(1 - b_{2/p}) \\
m_p(\Theta) &= (1 - b_{1/p})(1 - b_{2/p}) \\
m_p(\{\emptyset\}) &= 0
\end{aligned}
$$

4.3.2 Decision making

Relatively to a space point p, the fusion center constructs the pignistic transformation (denoted by $BetP_p$) that permits the construction of the probabilities needed for decision making. The decision is based on selecting the hypothesis $\hat{\theta}$ with the largest pignistic probability.

$$\hat{\theta} = \arg\max_{i=0,1} BetP_p(\{\theta_i\}) \tag{4.7}$$

If the selected hypothesis is $\hat{\theta} = \theta_1$ and the corresponding pignistic probability $BetP_p(\{\theta_1\})$ exceeds a predefined threshold (Th_p), we consider the target/event located at p to be detected; otherwise, the target/event is not detected. Thus, the target/event detection conditions are:

$$
\begin{aligned}
\hat{\theta} &= \theta_1 \\
BetP_p(\{\theta_1\}) &\geq Th_p
\end{aligned}
\tag{4.8}
$$

The threshold (Th_p) value is a user-specified application-specific parameter. A study on the effects of R_s, R_u, and Th_p on the system performance is reported in Section 4.6.

Example 2: Let us come back to the *bba m* of Example 1. Its pignistic probability function is:

$$
\begin{aligned}
BetP_p(\{\theta_0\}) &= \tfrac{(1-b_{1/p})(1-b_{2/p})}{2} \\
BetP_p(\{\theta_1\}) &= b_{1/p}b_{2/p} + b_{2/p}(1 - b_{1/p}) + b_{1/p}(1 - b_{2/p}) + \tfrac{(1-b_{1/p})(1-b_{2/p})}{2}
\end{aligned}
$$

4.4 Problem Formalization

In this section, we formalize four different forms of the uncertainty-aware wireless sensor networks deployment problem while considering our evidence-based coverage models. Specifically, we tackle the deployment problem in four stages. In the first stage, we consider uniform coverage and the cost of deployment only. We give a formal definition of the deployment problem where the preferential coverage and network connectivity objectives

were relaxed. In the second and the third stage, we will build on the work carried out in the first stage by introducing the preferential coverage objective and the network connectivity objective, respectively. Finally, in the fourth stage, all of the objectives will be studied together (i.e. preferential coverage, cost, and connectivity objectives).

4.4.1 Network Model and Assumptions

We assume that targets appear at a set of known locations referred to as targets points defining the set $T \subseteq RoI$. T is determined according to user requirements. If full coverage is required, targets points can be chosen densely and uniformly in the RoI. We constrain the deployment of sensors in the set $D \subseteq RoI$ consisting of deployment points in the RoI. If full coverage is required and sensors can be deployed anywhere in the RoI, then the RoI is discretized by a (two- or three-dimensional) grid of points. The granularity of the grid, i.e., the distance between consecutive grid points, can be adjusted to trade off computation time against accuracy. In the two-dimensional case, the RoI is represented by a $m \times n$ grid. Each cell's barycenter is a possible location for a sensor/target as illustrated in Fig. 4.3.

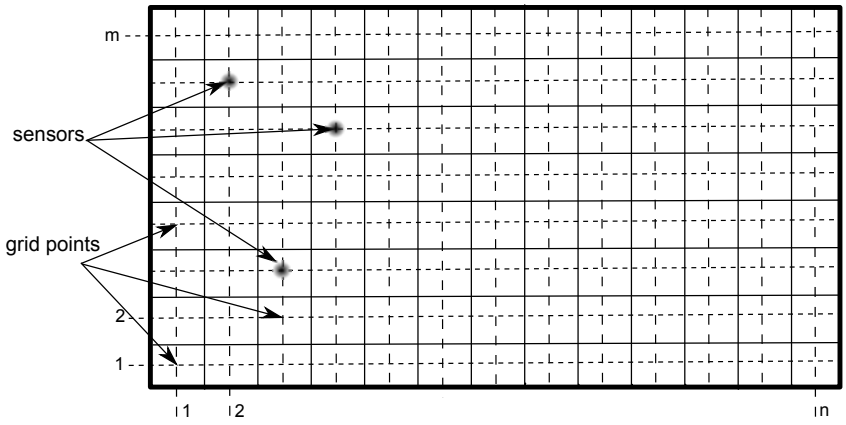


Figure 4.3: 2D RoI model.

To guarantee the area coverage of the RoI, all the target points must be covered under the considered evidence-based coverage model. It is straightforward to define the coverage in our sensor deployment scenario.

Definition 7. *A target point $p \in T$ is covered if a fusion center can reach a detection decision using evidence from deployed sensors, indicating that the selected hypothesis $\hat{\theta} = \theta_1$ and $BetP_p(\{\theta_1\})$ exceeds a predefined threshold (Th_p).*

4.4.2 Uniform Coverage and Cost Objectives

In our first WSN deployment problem, the number of sensors should be kept to a minimum while satisfying the coverage requirements for all points. We assume a uniform area coverage; each target point $p \in T$ is associated with a required minimum event detection probability threshold, denoted by th. Thus, our first deployment problem can be formulated as follows: $\forall p \in T$, the measured event detection pignistic probability of that point, $BetP(\{\theta_1\})_p$, must be greater than or equal to th. More formally, the first deployment problem can be formalized as follows.

$$\begin{aligned} \min \quad & \sum_{p \in D} x_p \\ \text{s.t.} \quad & BetP(\{\theta_1\})_p \geq th, \ \forall p \in T \\ & x_p = 0 \ or \ 1, \ \forall p \in D. \end{aligned}$$

Note that x_p is a zero-one variable. If $x_p = 1$ than we decide to deploy a sensor at the deployment point p.

4.4.3 Preferential Coverage and Cost Objectives

In this second WSN deployment problem, we assume a non-uniform area coverage (or a preferential coverage); thus each target point $p \in RoI$ is associated with a required minimum event detection probability threshold, denoted by th_p. Our second deployment problem can be formalized as follows.

$$\begin{aligned} \min \quad & \sum_{p \in D} x_p \\ \text{s.t.} \quad & BetP(\{\theta_1\})_p \geq th_p, \ \forall p \in T \\ & x_p = 0 \ or \ 1, \ \forall p \in D. \end{aligned}$$

4.4.4 Uniform Coverage, Cost and Connectivity Objectives

In this third WSN deployment problem, we will build on the first deployment problem by adding the network connectivity objective. Our goal is to deploy a WSN while optimizing the deployment cost and simultaneously ensuring uniform area coverage and guaranteeing network connectivity.

To ensure network connectivity, the connectivity graph $G = (V, E)$ must be connected. V is the set of vertex formed by the deployed sensors, and E is the set of edges. Assuming a

transmission disk model [120], a unit disk graph can be used to describe the network. CC_G denotes the set of connected components of $G = (V, E)$. The cardinal of CC_G must be equal to 1 in order to guarantee the network connectivity. More formally, the third deployment problem can be formalized as follows.

$$\begin{aligned} \min \quad & \sum_{p \in D} x_p \\ \text{s.t.} \quad & BetP(\{\theta_1\})_p \geq th, \ \forall p \in T \\ & |CC_G| = 1 \\ & x_p = 0 \ or \ 1, \ \forall p \in D. \end{aligned}$$

4.4.5 Preferential Coverage, Cost and Connectivity Objectives

In this fourth WSN deployment problem, we will build on the third deployment problem by adding the preferential coverage objective. Our goal is to deploy a WSN while optimizing the deployment cost and simultaneously ensuring preferential coverage and guaranteeing network connectivity. More formally, the fourth deployment problem can be formalized as follows.

$$\begin{aligned} \min \quad & \sum_{p \in D} x_p \\ \text{s.t.} \quad & BetP(\{\theta_1\})_p \geq th_p, \ \forall p \in T \\ & |CC_G| = 1 \\ & x_p = 0 \ or \ 1, \ \forall p \in D. \end{aligned}$$

4.5 The Proposed Sensor Placement Algorithms

The four combinatorial optimization problems formalized in Section 4.4 are NP-hard [61]. Thus, an optimal sensor placement algorithm cannot handle large-scale problems due to high time complexity. We devise two polynomial-time algorithms: i) a greedy-based heuristic called Evidence-Based WSNs Deployment Algorithm (*EBDA*), and ii) a dynamic-programming-based heuristic named Efficient Evidence-based WSNs Deployment Algorithm (*E2BDA*).

First, we detail EBDA and a simplified version of E2BDA, called *S-E2BDA*, where the preferential coverage and network connectivity objectives were relaxed. Second, we will build on the S-E2BDA placement algorithm by introducing the preferential coverage objective and the network connectivity objective, respectively. This will lead us to propose

two deployment strategies, *E2BDA-P* and *E2BDA-C*. Finally, all of the objectives will be studied together (i.e. preferential coverage, cost and connectivity objectives). This will lead us to devise the E2BDA placement algorithm.

4.5.1 EBDA

The main goal of a placement algorithm is to determine the minimum number of sensors and their locations so that every grid point is covered. Let S be the set of sensors and s be an individual sensor located at a grid point p. For any grid point p, the sensor detection outcome is an evidence as described before. We denote by SC_p an $m \times n$ matrix that corresponds to evidence of each grid point in the RoI when a sensor is located at grid point p. SC_p is associated with each grid point and can be pre-computed using our framework.

Because a certain number of sensors are placed to ensure coverage, we would like to know the coverage of each grid point due to a set of deployed sensors, namely, the *collective coverage*. We denote by CC an $m \times n$ matrix that corresponds to decisions reached by a "fictitious" fusion center using evidences from a set of deployed sensors over the grid points.

We propose an incremental deployment algorithm that determines the locations of the sensors one at a time. In each step, we would find all possible locations that are available on the grid for a sensor, and calculate the collective coverage matrix CC associated to this sensor in addition to those already deployed. A sensor is placed at the position that allows the maximum number of detection decisions. When the best location is found for the current sensor, CC is updated to include the newly introduced sensor. As our objective is to ensure that every grid point is covered, this intended outcome is our first termination criterion. We define the Number of Available Sensors (NAS) as a second termination criterion. The pseudo-code of EBDA is shown in Algorithm 3.

It is important to point out that because the vacuous belief function is a neutral element with respect to the combination operation, our algorithm does not check all grid points to compute the achieved coverage but only a limited zone using a sliding window based technique. EBDA has a quadratic computational complexity $O(w^2.m.n)$. Where w is the window size, which depends on the sensor coverage model. Here, w is equal to the sensor's sensing rang R_s.

Algorithm 3 EBDA

Input: RoI, S ($set\ of\ sensors$), Th, NAS
Output: PR ($Placement\ Result$), U

```
1: Set CC = [0], PR = [0], U = {∅}
2: for (every target point p ∈ T) do
3:    Compute SC_p
4: end for
5: repeat
6:    for (every deployment point p ∈ D) do
7:       Evaluate CC
8:       Place sensor s ∈ S on deployment point p such that CC is maximum
9:       Set S = S − {s}, U = U ∪ {s}
10:      Update PR
11:   end for
12: until (RoI is covered using PR) ∨ (|U| > NAS)
```

4.5.2 S-E2BDA

The first step in solving the deployment problem by dynamic programming is to characterize the structure of an optimal solution. The optimal substructure of our problem is as follows. Suppose that an optimal placement strategy of N sensors $s_1, ..., s_N$ is OP. Then the placement strategy of n sensors ($n < N$) within OP must be an optimal placement strategy of n sensors. Why? If there were a more efficient strategy to deploy $s_1, ..., s_n$, substituting that placement strategy in OP would produce another placement strategy whose "quality" is better than the optimum: a contradiction.

Our optimal substructure can be used to construct an optimal solution to the problem from optimal solutions to subproblems. However, for this deployment problem the total number of distinct subproblems is an exponential in the input size. This will lead us to an exponential-time algorithm. We propose a polynomial-time algorithm to construct a suboptimal solution as follows.

Let S be the set of sensors and s be an individual sensor located at grid point p. We denote SC_p the matrix that corresponds to evidences of each grid point in the RoI when a sensor is located at grid point p. SC_p is associated with each grid point and can be pre-computed using our framework.

Suppose that a placement strategy of n sensors $s_1, ..., s_n$ is D_n. A grid point p is covered if is satisfy Equation (4.8). We denote $score_{D_n}$ the number of coverage decisions reached by a "fictitious" fusion center using evidences from the set of deployed sensors $s_1, ..., s_n$ in the

RoI according to the placement strategy D_n. The deployment of an additional sensor s_{n+1}, generates a set of possible placement strategies D^q_{n+1}, where q corresponds to the location of the newly deployed sensor s_{n+1}. Obtained $score_{D^q_{n+1}}$ can be regrouped in a matrix we denote PS.

At the beginning, we consider the placement of one sensor s_1 in the RoI. For all possible locations on the grid, the algorithm computes the associated PS matrices and save the k-best placements of s_1 denoted BP. k is considered as an input parameter, which can take any value greater than 0. After that, in each step i, we would find all possible locations that are available on the grid for s_i, and calculate the matrices PS associated to s_i in addition to the sensors already deployed $s_1, ..., s_{i-1}$. BP are updated to save the k-best placements found for $s_1, ..., s_i$. As our objective is to ensure that every grid point is covered, this intended outcome is our first termination criterion. We define the Number of Available Sensors (NAS) as a second termination criterion. In the last step, the algorithm selects the best placement over the k-best placements BP found for $s_1, ..., s_n$, where $n \leq NAS$. The pseudo-code of S-E2BDA is illustrated in Algorithm 4.

Algorithm 4 S-E2BDA

Input: RoI, S, th, k, NAS
Output: oP ($optimalPlacement$), dS ($deployedSensors$)

{Initialization}
$dS = 0$
Set BP = [0]
for (every grid point $p \in RoI$) **do**
  Compute SC_p
end for
{Sensors placement}
repeat
  Using BP compute $U = \{p_a, p_a \in RoI$ and p_a *available*$\}$
  for (every grid point $p_a \in U$) **do**
    Compute the associated PS matrices using BP, SC_{p_a} and th
  end for
  Update BP to save the k-best placements
  $dS = dS + 1$
  oP = the best placement in BP
until (RoI is covered using oP) $\vee$ ($dS > NAS$)

It is important to point out that since the vacuous belief function is a neutral element

with respect to the combination operation, S-E2BDA does not check all grid points to compute the achieved coverage but only a limited zone using a sliding window based technique. If NAS is large, the computational complexity of S-E2BDA is equal to $O(kw^2mn)$, where k is the number of the best placements considered at each iteration and w is the window size. In fact, w depends on the sensor coverage model and is equal to the sensor's sensing rang R_s. The product kw^2 is equal to constant C, hence the computational complexity of S-E2BDA is equal to $O(Cmn)$.

4.5.3 E2BDA-P

For each grid point $p \in RoI$, there is an associated predefined minimum event detection probability threshold th_p. A grid point $p \in RoI$ is covered if the pignistic probability of θ_1 is greater than the associated threshold th_p. More formally, $p \in RoI$ *is covered* if $BetP(\{\theta_1\})_p \geq th_p$. We will build on our proposed solution S-E2BDA by adding the preferential coverage objective, the resulting protocol is denoted by E2BDA-P. In S-E2BDA we did not assume a uniform coverage, hence preferential coverage can be handled by simply considering the value of the pignistic probability associated with the hypothesis θ_1. E2BDA-P is identical to S-E2BDA except that E2BDA-P uses a detection threshold matrix Th as an input parameter instead of a single value th.

4.5.4 E2BDA-C

As previously done, we will build on our proposed solution S-E2BDA by adding the network connectivity objective, the resulting protocol is denoted E2BDA-C.

As our deployment algorithm S-E2BDA is incremental, connectivity can be easily ensured inherently. In each step i, using BP we find all available grid point $p_a \in RoI$ and are also within the connectivity area of already deployed sensors $s_1, ..., s_{i-1}$. We denote such points $\{p_{ac}\}$. More formally, assume that $G' = (V', E')$ is the connectivity graph of already deployed sensors $s_1, ..., s_{i-1}$ using E2BDA-C. G' is a connected graph by construction. Let s_i be a new sensor deployed at a location p_a, and $G'' = (V'', E'')$ the resulting graph. The location p_a must satisfy the condition: G'' is a connected graph.

In comparison to S-E2BDA, E2BDA-C considers $\{p_{ac}\}$ grid points instead of $\{p_a\}$ grid points. The transmission range R_c is used as an input parameter to E2BDA-C.

4.5.5 E2BDA

To simultaneously handle the preferential coverage and the network connectivity objectives, we will build on our previous proposed solutions by combining E2BDA-P and E2BDA-C, the resulting algorithm is denoted E2BDA. The pseudo-code of E2BDA is illustrated in Algorithm 5. The computational complexity of E2BDA is also equal to $O(Cmn)$.

Algorithm 5 E2BDA

Input: RoI, S, Th, k, NAS, R_c
Output: $optimalPlacement$, $deployedSensors$

{Initialization}
1: $deployedSensors = 0$
2: Set $BP = [0]$
3: **for** (every target point $p \in RoI$ such as $Th_p > 0$) **do**
4: Compute SC_p
5: **end for**
{Sensors placement}
6: **repeat**
7: Using BP and R_c compute $U = \{p_{ac}, p_{ac} \in RoI$ and $Th_{p_{ac}} > 0$ and p_{ac} *available* and $p_{ac} \in$ *the connectivity area of* $BP\}$
8: **for** (every grid point $p_{ac} \in U$) **do**
9: Compute the associated PS matrices using BP, SC_{p_a} and Th
10: **end for**
11: Update BP to save the k-best placements
12: $deployedSensors = deployedSensors + 1$
13: $optimalPlacement$ = the best placement in BP
14: **until** (RoI is covered using $optimalPlacement$) $\vee$ ($deployedSensors > NAS$)

4.6 Performance Evaluation

The objective of this performance analysis is threefold. (1) It permits an evaluation of the effect of different parameters of the proposed framework on the system performance. (2) It leads to an assessment of the proposed placement algorithms. (3) It permits an objective assessment of the proposed approach in comparison to the state of the art approaches. Starting with the study of the effect of different parameters of the proposed framework on the system performance, two evaluation stages are performed. In the first one, we analyze the effects of R_s, R_u, and Th parameters. In the second experiment, we analyze the required number of sensors while varying gradually the sensor reliability based on the Equations

(4.3) and (4.4). In the third experiment, we compare EBDA with the random placement strategy. A comparison between our approach and the probabilistic one is reported in the fourth experiment. The first two experiments are based on our own synthetic data, whereas the third experiment is based on real data traces collected in the DARPA SensIT vehicles detection experiments [41]. Finally, in the last experiment, we evaluate the effectiveness of our deployment algorithm E2BDA. We compare our proposal to the state of the art deployment strategies, namely: Max-Avg-Cov [38], Max-Min-Cov [38], MIN_MISS [152] and our deployment algorithm EBDA.

4.6.1 The Effects of Different Parameters on the System Performance

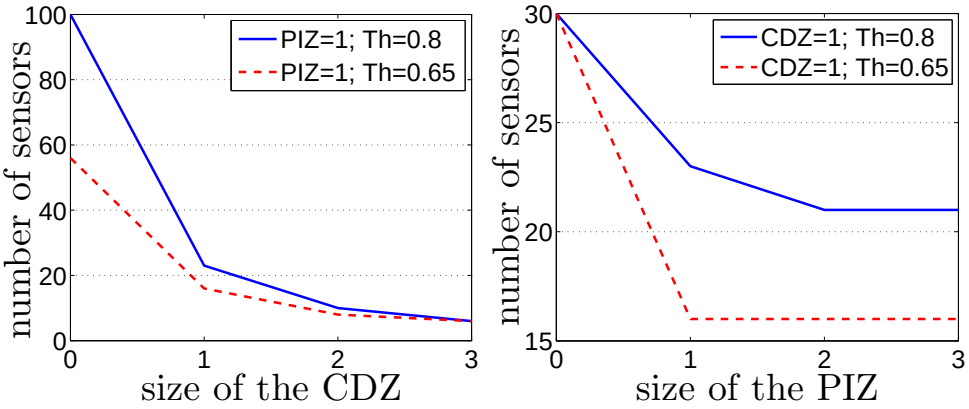
The Effects of R_s, R_u, and Th on the performance

In this experiment, we analyze the effects of R_s, R_u and Th parameters. The RoI is assumed to be a 10×10 grid. On Fig. 4.4(a), we see that increasing the size of the CDZ reduces the required number of sensors. This reduction is more significant when the parameter Th is more important. Intuitively, as the CDZ increases, more grid points will be covered by the sensor resulting in better sensing quality.

The same behavior is observed when increasing the size of the PIZ as depicted in Fig. 4.4(b). It should be noted that the first result depicted in Fig. 4.4(b) shows the number of sensors when considering a binary model (PIZ=0). For example, for a PIZ=1, this number is reduced by 46%. Therefore, considering noisy sensory data of multiple sensors (partial ignorance zones) can significantly improve sensing coverage by exploiting the collaboration among sensors. These results demonstrate the advantage of the evidence-based model compared to the binary model.

The Effects of sensor reliability on the performance

In the second experiment, we analyze the required number of sensors while varying gradually the sensor reliability (Fig. 4.5) based on Equations (4.3) and (4.4). We see clearly that this parameter has a significant impact on the required number of sensors. This impact is more significant when the Th is more important, which is normal: the application is more critical and needs more precision in the collected measurements. Therefore, it is very important to consider this issue in the deployment process. It should be noted here that the probabilistic model ignores completely this issue while the evidence-based model integrates it seamlessly.



(a) Varying the size of the CDZ. (b) Varying the size of the PIZ.

Figure 4.4: The effects of R_s, R_u and Th.

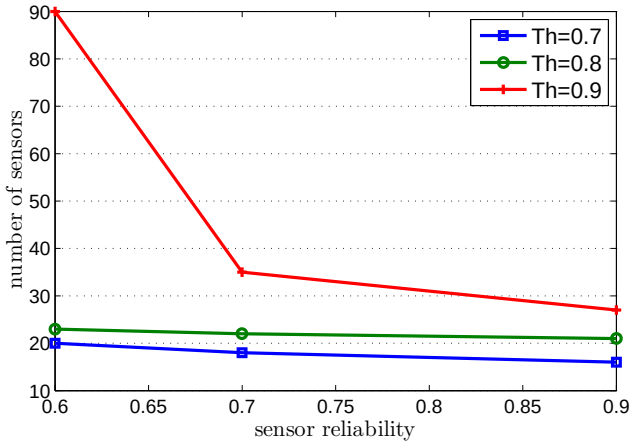


Figure 4.5: Required number of sensors *vs.* sensor reliability.

4.6.2 EBDA *vs.* Random

As in [38, 39, 152], we use the random placement of sensors as a baseline. The Random strategy consists of deploying sensors in the RoI according to a uniform distribution (cf. Chapter 2). For both strategies (Random and EBDA), we analyze their convergence and study the effect of the decision threshold (Th) on the number of deployed sensors. In a second phase, we examine the cost of full area coverage. Finally, we analyze the coverage achieved when the deployment problem is defined as the placement of a fixed number of sensors to ensure maximum coverage.

The RoI is assumed to be a square, with a side of n units length. We present our simulations results for different sets of parameters. The simulation parameters (in units

of grid point) are presented in Table 4.1. The experiments were repeated ten times with varying random seed, as suggested in [58]. Whenever confidence intervals are shown in plots, the confidence level on these intervals is 95%.

Table 4.1: Simulations parameters

Parameters	**Values**
n	$5 - 15$
R_s, R_u	$2, 1$
Th	$0.500 - 0.999$

Evaluation Stage 1: Study of the Convergence

In a 10×10 RoI, we analyze the convergence of each algorithm in terms of the coverage achieved. As depicted in Fig. 4.6, for each newly deployed sensor, EBDA improves coverage while it is not always the case for the Random strategy. To achieve a 100% rate of coverage, EBDA needs only 23 sensors, while for the Random strategy, we need 68 ± 7 sensors.

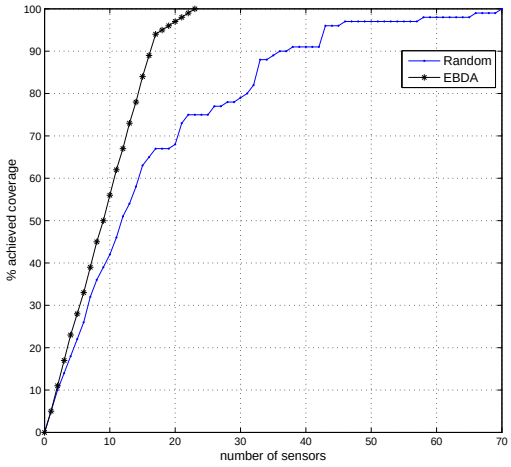


Figure 4.6: EBDA *vs.* Random in terms of convergence.

Evaluation Stage 2: Analysis of the Effect of Th on the Deployment Cost

In a 10×10 RoI, we vary the threshold Th used in the decision step defined in Equation (4.8). As expected, we see clearly in Fig. 4.7 that the needed number of sensors to

achieve full coverage is an increasing function of the threshold. When the threshold is about 1, we are asking for almost certain detection, which explains the large number of required sensors. In all cases, EBDA requires fewer sensors than Random to achieve full area coverage.

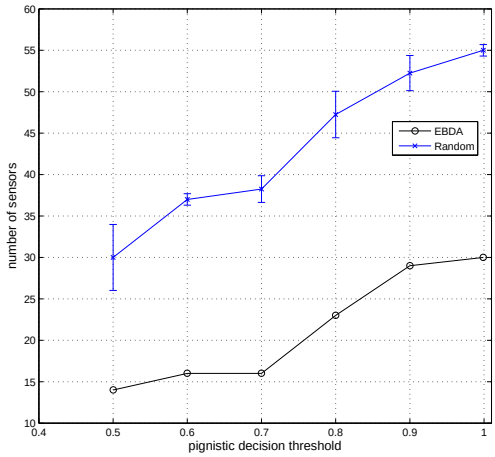


Figure 4.7: EBDA *vs.* Random using different thresholds.

Evaluation Stage 3: Study of the Deployment Cost

We vary the size of the RoI and we compare EBDA to Random in terms of deployment cost for full area coverage. The results plotted in Fig. 4.8 show that fewer sensors are required when EBDA is used. In addition, we see clearly that EBDA presents a linear curve, while Random presents an exponential one. Contrary to the results presented in [39, 152], EBDA outperforms Random in all cases.

Evaluation Stage 4: Study of the Achieved Coverage for a Fixed Cost

In this scenario, we fix the number of available sensors (NAS) to $\lfloor n^2/5 \rfloor$ and we analyze the coverage achieved by EBDA and Random. In Fig. 4.9, we see clearly that the average achieved coverage provided by EBDA exceeds 97%, while that provided by Random is about 71%. This allows us to see that EBDA is still effective when the deployment problem is defined as the placement of a fixed number of sensors to ensure maximum area coverage.

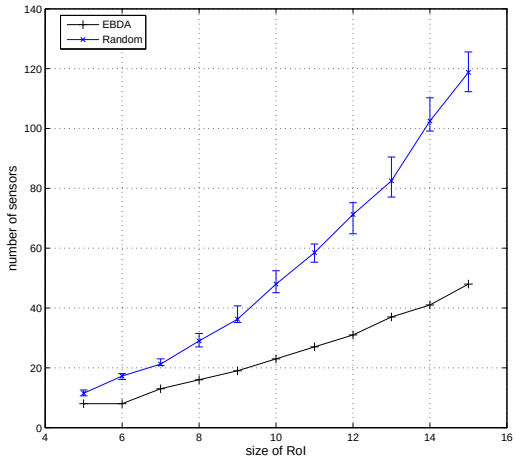


Figure 4.8: EBDA *vs.* Random in terms of cost.

4.6.3 EBDA *vs.* MIN_MISS: Numerical Experiments

In this section, we report extensive simulations based on synthetic data as well as real data traces to evaluate our approach. To quantify the real benefit of the proposed methodology, we compare our evidence-based placement algorithm (EBDA) to a state of the art probabilistic deployment strategy MIN_MISS [152]. The choice of this two algorithms is based on the following reasons: (i) both algorithms are iterative, where in each iteration, only one sensor is deployed, (ii) EBDA deploys a new sensor at a deployment point p that maximizes the number of detection decisions while MIN_MISS deploys a new sensor at a deployment point p, that minimizes the over miss probability value, (iii) EBDA exploits our evidence-based sensor coverage model while MIN_MISS is based on the probabilistic sensor coverage model. This choice allows us to quantify the real benefit of our approach while minimizing the differences due to the efficiency of the sensor placement algorithms.

Based on the real data traces (cf. Fig. 4.1(a)), we estimate that the detection probability of a target located at point p by a sensor located at s is: $p_d(s/p) = e^{-0.01d(s,p)}$. For a fair comparison, this estimation is used for both strategies.

Analysis of the effect of Th on the deployment cost

For both strategies, we analyze the cost of full area coverage as a function of the required minimum target detection probability threshold Th in $300m \times 300m$ and $400m \times 400m$ RoI. As expected, we see clearly in Figures 4.10(a)-4.10(b) that the number of sensors

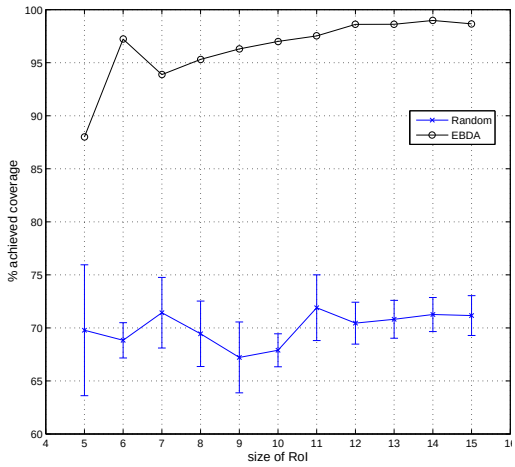
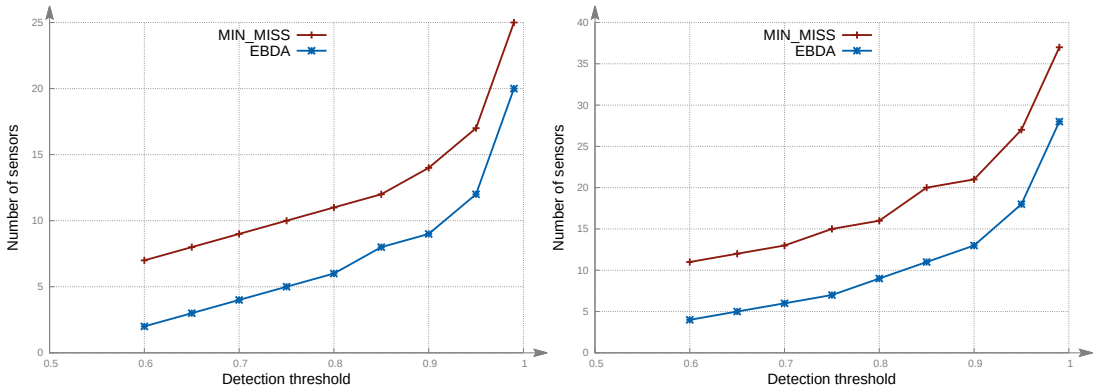


Figure 4.9: EBDA *vs.* Random in terms of achieved coverage.

needed to achieve full coverage is an increasing function of the detection threshold. When this threshold is about 1, we are asking for almost certain detection, which explains the large number of required sensors. Similarly, the larger the RoI, the greater is the needed number of sensors to achieve full coverage. EBDA consistently outperforms MIN_MISS in all cases.



(a) $300m \times 300m$ RoI. (b) $400m \times 400m$ RoI.

Figure 4.10: EBDA *vs.* MIN_MISS using different thresholds.

Study of the deployment cost

We vary the size of the RoI and we compare EBDA to MIN_MISS in terms of deployment cost for full area coverage. The threshold Th is set to 0.9. The results plotted in Fig. 4.11 show that fewer sensors are required when EBDA is used. EBDA consistently

outperforms MIN_MISS in all cases. The average performance gain is about 40%. It should be noted that this gain depends heavily on the size of the RoI, the larger the RoI, the greater is the performance gain. These results confirm that evidence combination improves significantly sensing coverage by exploiting the collaboration among sensors.

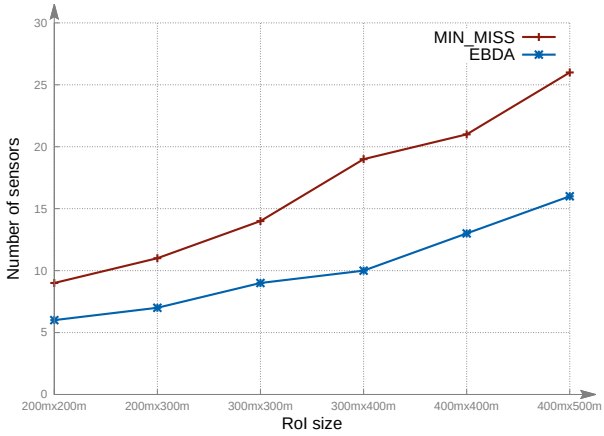


Figure 4.11: EBDA *vs.* MIN_MISS in terms of deployment cost ($Th = 0.9$).

4.6.4 EBDA *vs.* MIN_MISS: Trace-driven Simulations

In this section, we present results obtained when conducting extensive simulations using real data traces collected in the DARPA SensIT military vehicles detection experiments [41]. The dataset used in our simulations include the location information for 18 nodes, the ground truth information of the vehicles recorded by a GPS device and the acoustic signal energy measurements recorded by 18 nodes at a sampling period of 0.75 s, when military vehicle drives through the RoI. For detailed setup of the experiments, we refer to [41].

In Section 4.2, we have detailed how belief functions are modeled and manipulated. In the next section, we show how they are build from raw sensing data.

Building belief functions

To build belief functions from sensing data, it is possible to use methods exploiting statistics [17] or simple intuition-based methods [84, 113]. The former is not appropriate to the problem at hand; the latter may be unrealistic or at least may not reflect reality well. We choose a simple method based on the analysis of available real data traces. The proposed method reflects reality well and is generic enough to be adapted for different sensor types. This section describes our effort to build belief functions from raw sensing data.

We propose a two-step method: First, we analyze the average performance of sensors in terms of detection probability as a function of the target/event-sensor distance. Second, we analyze the signal energy as a function of the target/event-sensor distance. Using the real data traces collected in the DARPA SensIT military vehicle detection experiments [41], we plot in Fig. 4.12 a set of mass functions associated to a WINS NG 2.0 sensor in the case of Assault Amphibian Vehicle detection. For simplicity's sake, the probability mass assigned to target detection is chosen linear with the detection probability. A projection on the set of mass functions is done in order to obtain the corresponding mass function each time a raw data from that sensor is received. For instance, if a sensor sample an acoustic signal energy of 0.1193, then the resulting mass function would have two focal elements: $m(\{\theta_1\}) = 0.8187$ and $m(\Theta) = 0.1812$.

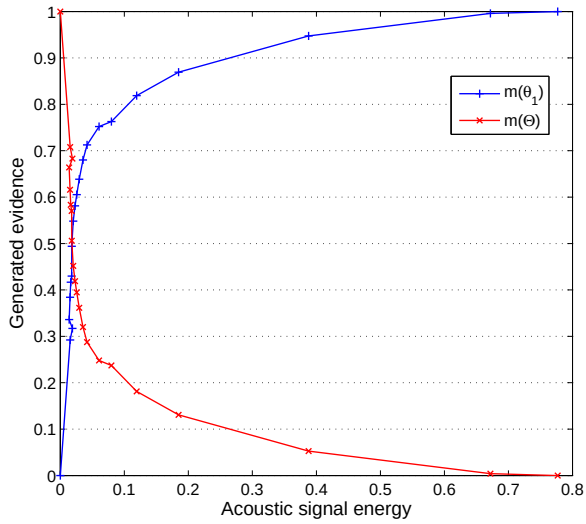


Figure 4.12: Example of a set of mass functions associated to a WINS NG 2.0 sensor in the case of AAV vehicle detection.

We vary the size of the RoI and we compare EBDA to MIN_MISS in terms of real achieved coverage. In the trace-driven simulations, each simulated sensor is associated with a real sensor randomly selected. When a simulated sensor samples an energy measurement, we compute the sensor-vehicle distance. The sensor's measurement is then set to be the real measurement gathered by the associated real sensor at the same distance to the vehicle. This approach has been used in many recent research works [29, 41, 110]. Afterward, this energy measurement is used by the sensor to build its beliefs about vehicle presence. The experiments were repeated hundred times with varying random seed. The obtained results

are summarized in Fig. 4.13, with 95% confidence intervals.

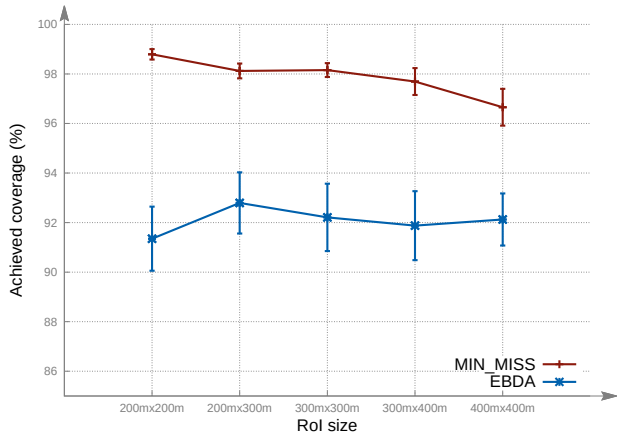


Figure 4.13: EBDA *vs.* MIN_MISS in terms of achieved coverage.

In all considered scenarios, both strategies ensure the minimum required detection probability (0.9). Further, we notice that MIN_MISS outperforms EBDA in terms of detection rate. This is due to the fact that MIN_MISS uses additional sensors in comparison to EBDA. For instance, to guarantee a 0.9 detection probability in a $400m \times 400m$ RoI, MIN_MISS employs 21 sensors, whereas EBDA uses only 13 sensors. This represents a significant saving in terms of deployment cost. To conclude, EBDA reduces the deployment cost while ensuring the full satisfaction the user detection requirements.

It is worth pointing out that EBDA saves almost 40% of the deployment cost over MIN_MISS while ensuring the minimum required detection probability. A detailed analysis of the sensor placements generated by the EBDA and MIN_MISS clearly shows that EBDA generates a wider and sparser placement which allows the sensing capability of the network to be fully utilized. A visualization of a sensor placement generated by EBDA and MIN_MISS is depicted in Fig. 4.14.

4.6.5 E2BDA *vs.* State of the Art Deployment Strategies

The objective of this experiment is to evaluate the effectiveness of our deployment algorithm E2BDA. We compare our proposal to the state of the art deployment strategies, namely: Max-Avg-Cov [39], Max-Min-Cov [38], MIN_MISS [152] and our deployment algorithm EBDA. In order to perform a fair comparison, Max-Avg-Cov, Max-Min-Cov and MIN_MISS have been adapted to use the evidence-based coverage model. Some recent

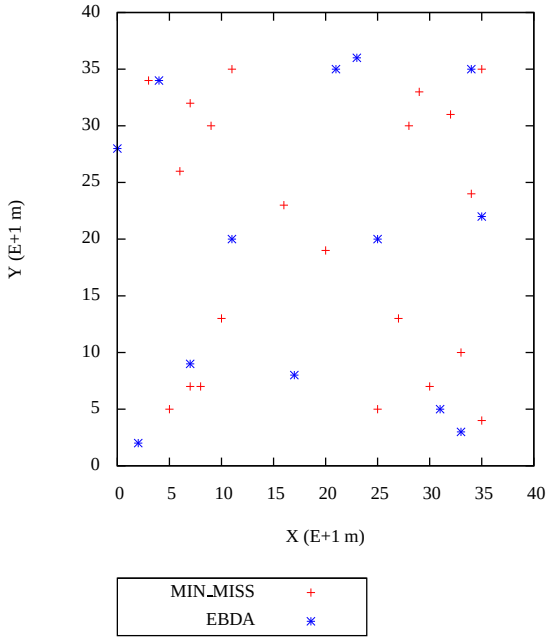


Figure 4.14: Sensor placements using EBDA and MIN_MISS (RoI=400$m \times 400m$ and $Th = 0.9$).

related work have been intentionally ignored in this comparison, such as BDA [9] which cannot guarantee full area coverage even with a high number of iterations.

We perform a three-stage simulation. In the first stage, we compare the performance of different deployment strategies for uniform coverage requirements. In the second stage, the performance of different deployment strategies for preferential coverage requirements is evaluated. Finally, we investigate the performance of different deployment strategies when the network connectivity is considered.

The RoI is assumed to be a square, with a side of n units length. In stage 1 and stage 2, R_c has been set large to relax the network connectivity constraint. The simulation parameters (in units of grid point) are presented in Table 4.2.

Table 4.2: Simulations parameters

Parameters	**Values**
n	20
R_s, R_u	2, 1
k	3
NAS	n^2

Evaluation Stage 1: Uniform Coverage Requirements

We first study the performance of different deployment strategies with three uniform coverage requirements scenarios. In scenarios 1, 2, and 3 *th* is equal to 0.65, 0.8, and 0.95 respectively. Table 4.3 shows the number of sensors required to guarantee full uniform coverage.

Table 4.3: Cost of full uniform coverage

	Value of Th		
	0.65	0.8	0.95
Deployment strategy	**Number of sensors**		
E2BDA	49	81	112
EBDA	52	82	130
MIN_MISS	73	92	211
Max-Min-Cov	49	98	134
Max-Avg-Cov	110	235	235

The first observation we draw from Table 4.3 is that for all deployment strategies, the deployment cost increases as the coverage requirement increases. The second observation is that both EBDA and E2BDA use much fewer sensors than MIN_MISS, Max-Min-Cov and Max-Avg-Cov. However, E2BDA achieves better performance than EBDA. The performance advantage of E2BDA over EBDA is more significant when the coverage requirement is larger.

Evaluation Stage 2: Preferential Coverage Requirements

In this section, we study the performance of different deployment strategies when the coverage requirements at different locations are different. As in [9], the preferential coverage requirements are generated according to a multivariate normal distribution, as illustrated in Fig. 4.15.

Table 4.4 shows the performance of the five deployment algorithms. We can observe that E2BDA outperforms all other strategies. The performance advantage of E2BDA over EBDA, MIN_MISS, Max-Min-Cov and Max-Avg-Cov is more significant for preferential coverage requirements.

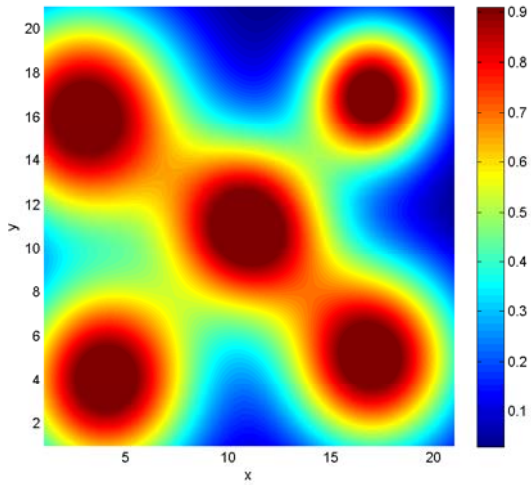


Figure 4.15: Preferential coverage requirements.

Table 4.4: Cost of full preferential coverage

Deployment strategy	**Number of sensors**
E2BDA	110
EBDA	132
MIN_MISS	209
Max-Min-Cov	130
Max-Avg-Cov	143

Evaluation Stage 3: Network Connectivity

In this section, we show the performance of the deployment strategies when the network connectivity is considered. It is important to point out that EBDA, MIN_MISS, Max-Min-Cov and Max-Avg-Cov does not take into consideration the issue of network connectivity. Only E2BDA addresses explicitly this issue. For different values of R_c, we check the connectivity graphs of the topologies generated by the different deployments strategies. Table 4.5 summarizes the obtained results.

The obtained results show that only E2BDA strategy guarantees the network connectivity in all scenarios. For instance, in the case when R_c is equal to 1, only E2BDA strategy guarantees the network connectivity. We notice that in this case, E2BDA incurs an additional cost in comparison to the result obtained in Evaluation Stage 1. This additional cost is due to the network connectivity constraint and depends on the value of R_c. In fact, there is an inverse correlation between this additional cost and the value of R_c.

Table 4.5: Cost of coverage and connectivity

	Value of R_c		
	1	2	4
Deployment strategy	**Number of sensors | Connectivity**		
E2BDA	113 | Yes	60 | Yes	49 | Yes
EBDA	52 | No	52 | No	52 | Yes
MIN_MISS	73 | No	73 | Yes	73 | Yes
Max-Min-Cov	49 | No	49 | No	49 | Yes
Max-Avg-Cov	110 | No	110 | Yes	110 | Yes

4.7 Conclusion

In this Chapter, we focused on uncertainty-aware WSNs deployment where sensors may not always provide reliable information. We investigated the evidence theory to design better deployment strategies. First, we devised an evidence-based sensor coverage model. Second, based on this coverage model, we formalized four variants of the uncertainty-aware WSNs deployment problem as combinatorial optimization problems, which are NP-hard. Third, we proposed five polynomial-time uncertainty-aware deployment algorithms. Experimental results, based on both synthetic data sets and data traces collected in a real deployment for military vehicle detection, show that our approach outperforms the state-of-the-art deployment strategies.

Although, obtained results are satisfactory, many issues still not addressed. For instance, we have no idea on the rate of false alarm generated by the above deployment strategies. In addition, connectivity unreliability and network lifespan were not addressed. The next Chapter is an attempt to handle these challenges.

Chapter 5

Uncertainty-aware Fusion-based WSNs Deployment

5.1 Introduction

In the previous Chapter, we have considered the coverage rate or detection rate as the main metric. Thus, we have focused on the coverage rate and not on the quality of coverage. However, in mission-critical wireless sensor networks surveillance applications, a high detection rate coupled with a low false alarm rate is essential. In other words, mission-critical WSNs surveillance applications require a good quality of coverage not a good coverage. In contrast to the deployment problem treated in the previous Chapter, here we consider both false alarm and detection requirements. In addition, we address the issues of connectivity and network lifespan. It is worth noting that related work [1, 4, 29] ignore completely the issue of connectivity although sensors are supposed to collaborate. Also, the network lifespan issue is not addressed.

Critical applications such as military target detection impose stringent requirements for event detection accuracy such as a low false alarm rate coupled with a high detection rate. To detect a target, sensors have to make local observations of their surrounding environment and collaborate to produce a global decision that reflects the status of the RoI. It is well known that collaboration among sensors improve the sensing quality by jointly considering noisy measurements of multiple unreliable sensors. For example, He et al. [51] show that the false alarm rate of a real-world Mica2 WSN can be reduced from 60% (when sensors decide independently) to near zero by adopting a fusion scheme.

As we said before, the deployment is a mandatory and critical step in the process of developing WSNs solutions for real-life applications. Even if many optimizations can be done once the WSN is deployed to enhance its performance, we believe that the most important optimizations are those done at the design step to build the best topology possible that meets specific user requirements. In this Chapter, we investigate the following question: what is the best way to deploy a finite number of unreliable sensors characterized by uncertain readings in order to satisfy the user detection requirements while extending the network lifetime. Unlike prior efforts that rely on simple fusion schemes, a generic evidence fusion scheme, that captures several characteristics of real-world applications, is defined on the basis of Dempster-Shafer theory. We also define a communication cost prediction approach to address the issues of connectivity and network lifespan. The fusion-based uncertainty-aware wireless sensor networks deployment problems are formulated as binary non-linear and non-convex optimization problems and efficient heuristics using genetic algorithms are investigated. The proposed approaches are validated through extensive experiments carried out on both synthetic data set and real data traces collected in a military vehicle detection experiment. The obtained results demonstrate the interest of the evidence fusion model that considers in a meaningful way the information on the quality of sensors decisions as well as the reliability of these sensors along with their uncertain and imprecise decisions. Also, the devised approaches outperform the state of the art deployment strategies.

5.2 Evidence Fusion Model

Two commonly fusion schemes are considered in the literature: value fusion and decision fusion [114]. In the value fusion scheme, the nodes exchange their raw energy measurements whereas in the decision fusion scheme the nodes exchange local detection decisions based on their energy measurement. As a compromise, in this work, an information-rich cost-effective approach is formulated: an evidence could be transmitted to the fusion center containing uncertain decision along with the information quality related to the degree of confidence that a sensor has about its decision. This approach combines the advantages of both value and decision fusion approaches. In this section, we provide a formal definition of the evidence-based sensing model and a description of how to build evidence from raw sensing data.

5.2.1 Evidence-based Sensing Model

Contrasting with related work, in this work, we consider a more general decision scheme wherein each sensor produces a belief function instead of a binary decision. We do not suppose a particular sensing model which highly depends on the used sensors and the RoI. In this section, we detail how belief functions, also called basic belief masses (bbm), are modeled and manipulated, and in the next section we show how they are build from raw sensing data.

Evidence construction

When a target is present in the RoI, two states are required to specify whether the target is detected: θ_0 (target present and not detected) and θ_1 (target present and detected). Thus, the FoD is the set $\Theta^t = \{\theta_0, \theta_1\}$. To translate the uncertain sensory data into belief functions, we proceed as follow: for a sensor s_i and relatively to a point $p \in RoI$, we define three inputs: the *bbm* assigned to no detection $m^{\Theta^t}_{d_{i/p}}(\theta_0)$, the *bbm* assigned to detection $m^{\Theta^t}_{d_{i/p}}(\theta_1)$, and the unassigned *bbm* $m^{\Theta^t}_{d_{i/p}}(\theta_0, \theta_1)$. The sum of these masses always equals one, so there are only two independent masses: $m^{\Theta^t}_{d_{i/p}}(\theta_0)$ and $m^{\Theta^t}_{d_{i/p}}(\theta_1)$. The mass $m^{\Theta^t}_{d_{i/p}}(\theta_1)$ represents a belief in a target presence at p; the mass $m^{\Theta^t}_{d_{i/p}}(\theta_0)$ represents the opposite and the mass $m^{\Theta^t}_{d_{i/p}}(\theta_0, \theta_1)$ reflects the uncertainty of the sensor s_i.

Given the belief $b_{d_{i/p}}$ of a sensor s_i that claims that there is a target located at p, the following mapping is used to acquire the probability masses:

$$m^{\Theta^t}_{d_{i/p}}(x) = \begin{cases} (1 - u_i) \cdot (1 - b_{d_{i/p}}) & \text{if } x = \{\theta_0\}, \\ (1 - u_i) \cdot b_{d_{i/p}} & \text{if } x = \{\theta_1\}, \\ u_i & \text{if } x = \{\theta_0, \theta_1\}, \\ 0 & \text{if } x = \{\emptyset\}. \end{cases}$$

where $u_i \in [0, 1]$ is the uncertainty of sensor s_i about its decision.

Alternatively, when a target is not present in the RoI, two states are required to specify whether a target is detected: θ_2 (target not present and not detected), θ_3 (target not present and detected). Thus, the FoD is the set $\Theta^{nt} = \{\theta_2, \theta_3\}$. Given the false alarm rate b_{f_i} of a sensor s_i, we translate the uncertain sensory data into belief functions by defining a belief function $m^{\Theta^{nt}}_f$ on the FoD Θ^{nt} as follows:

$$m_{f_i}^{\Theta^{nt}}(x) = \begin{cases} (1-u_i)\cdot(1-b_{f_i}) & \text{if } x = \{\theta_2\}, \\ (1-u_i)\cdot b_{f_i} & \text{if } x = \{\theta_3\}, \\ u_i & \text{if } x = \{\theta_2, \theta_3\}, \\ 0 & \text{if } x = \{\emptyset\}. \end{cases}$$

It should be noted here, that both formulations $m_{d_{i/p}}^{\Theta^t}$ and $m_{f_i}^{\Theta^{nt}}$ meet Appriou's axioms [15].

Decision combination

When several decisions are obtained through distinct sensors, new decision representing the consensus of those disparate decisions can be obtained through the combination operation. Let $m_{d_{1\oplus 2}}^{\Theta^t}$ be the combination of the two bbas $m_{d1}^{\Theta^t}$ and $m_{d2}^{\Theta^t}$ produced by two sensors s_1 and s_2, respectively. Using the Dempster's rule of combination, the bba $m_{d_{1\oplus 2}}^{\Theta^t} = m_{d1}^{\Theta^t} \oplus m_{d2}^{\Theta^t}$ has the following expression:

$$\begin{aligned} m_{d_{1\oplus 2}}^{\Theta^t}(\theta_0) &= k_{d_{1\oplus 2}} \cdot [m_{d_1}^{\Theta^t}(\theta_0)\cdot m_{d_2}^{\Theta^t}(\theta_0) + m_{d_1}^{\Theta^t}(\theta_0)\cdot m_{d_2}^{\Theta^t}(\theta_0,\theta_1) + m_{d_1}^{\Theta^t}(\theta_0,\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_0)] \\ m_{d_{1\oplus 2}}^{\Theta^t}(\theta_1) &= k_{d_{1\oplus 2}} \cdot [m_{d_1}^{\Theta^t}(\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_1) + m_{d_1}^{\Theta^t}(\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_0,\theta_1) + m_{d_1}^{\Theta^t}(\theta_0,\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_1)] \\ m_{d_{1\oplus 2}}^{\Theta^t}(\theta_0,\theta_1) &= k_{d_{1\oplus 2}} \cdot [m_{d_1}^{\Theta^t}(\theta_0,\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_0,\theta_1)] \\ m_{d_{1\oplus 2}}^{\Theta^t}(\emptyset) &= 0 \end{aligned} \tag{5.1}$$

where

$$k_{d_{1\oplus 2}} = \frac{1}{1 - (m_{d_1}^{\Theta^t}(\theta_0)\cdot m_{d_2}^{\Theta^t}(\theta_1) + m_{d_1}^{\Theta^t}(\theta_1)\cdot m_{d_2}^{\Theta^t}(\theta_0))}$$

Similarly, $m_{f_{1\oplus 2}}^{\Theta^{nt}} = m_{f_1}^{\Theta^{nt}} \oplus m_{f_2}^{\Theta^{nt}}$ is the combination of two bbas $m_{f_1}^{\Theta^{nt}}$ and $m_{f_2}^{\Theta^{nt}}$ produced by two sensors s_1 and s_2, respectively. The bba $m_{f_{1\oplus 2}}^{\Theta^{nt}}$ has the following expression:

$$\begin{aligned} m_{f_{1\oplus 2}}^{\Theta^{nt}}(\theta_2) &= k_{f_{1\oplus 2}} \cdot [m_{f_1}^{\Theta^{nt}}(\theta_2)\cdot m_{f_2}^{\Theta^{nt}}(\theta_2) + m_{f_1}^{\Theta^{nt}}(\theta_2)\cdot m_{f_2}^{\Theta^{nt}}(\theta_2,\theta_3) + m_{f_1}^{\Theta^{nt}}(\theta_2,\theta_3)\cdot m_{f_2}^{\Theta^{nt}}(\theta_2)] \\ m_{f_{1\oplus 2}}^{\Theta^{nt}}(\theta_3) &= k_{f_{1\oplus 2}} \cdot [m_{f_1}^{\Theta^{nt}}(\theta_3)\cdot m_{f_2}^{\Theta^{nt}}(\theta_3) + m_{f_1}^{\Theta^{v}}(\theta_3)\cdot m_{f_2}^{\Theta^{nt}}(\theta_2,\theta_3) + m_{f_1}^{\Theta^{nt}}(\theta_2,\theta_3)\cdot m_{f_2}^{\Theta^{nt}}(\theta_3)] \\ m_{f_{1\oplus 2}}^{\Theta^{nt}}(\theta_2,\theta_3) &= k_{f_{1\oplus 2}} \cdot [m_{f_1}^{\Theta^{nt}}(\theta_2,\theta_3)\cdot m_{f_2}^{\Theta^{nt}}(\theta_2,\theta_3)] \\ m_{f_{1\oplus 2}}^{\Theta^{nt}}(\emptyset) &= 0 \end{aligned} \tag{5.2}$$

where

$$k_{f_{1\oplus 2}} = \frac{1}{1 - (m_{f_1}^{\Theta^{nt}}(\theta_2) \cdot m_{f_2}^{\Theta^{nt}}(\theta_3) + m_{f_1}^{\Theta^{nt}}(\theta_3) \cdot m_{f_2}^{\Theta^{nt}}(\theta_2))}$$

For N sensors, the combination of the N *bbas* $m_{d_1}^{\Theta^t}, \ldots, m_{d_N}^{\Theta^t}$ using the $\oplus$ rule yields a *bba* $m_d^{\Theta^t}$ that has the following expression:

$$\begin{array}{ll} m_d^{\Theta^t}(\theta_0) & = 1 - \prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_0)) \cdot k_d \\ m_d^{\Theta^t}(\theta_1) & = 1 - \prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_1)) \cdot k_d \\ m_d^{\Theta^t}(\Theta^t) & = \prod_{i=1}^{N} m_{d_i}^{\Theta^t}(\Theta^t) \cdot k_d \end{array} \tag{5.3}$$

where

$$k_d = \frac{1}{\prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_0)) + \prod_{i=1}^{N}(1 - m_{d_i}^{\Theta^t}(\theta_1)) - \prod_{i=1}^{N} m_{d_i}^{\Theta^t}(\Theta^t)}$$

Similarly, the combination of the N *bbas* $m_{f_1}^{\Theta^{nt}}, \ldots, m_{f_N}^{\Theta^{nt}}$ using the $\oplus$ rule yields a *bba* $m_f^{\Theta^{nt}}$ that has the following expression:

$$\begin{array}{ll} m_f^{\Theta^{nt}}(\theta_2) & = 1 - \prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_2)) \cdot k_f \\ m_f^{\Theta^{nt}}(\theta_3) & = 1 - \prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_3)) \cdot k_f \\ m_f^{\Theta^{nt}}(\Theta^{nt}) & = \prod_{i=1}^{N} m_{f_i}^{\Theta^{nt}}(\Theta^{nt}) \cdot k_f \end{array} \tag{5.4}$$

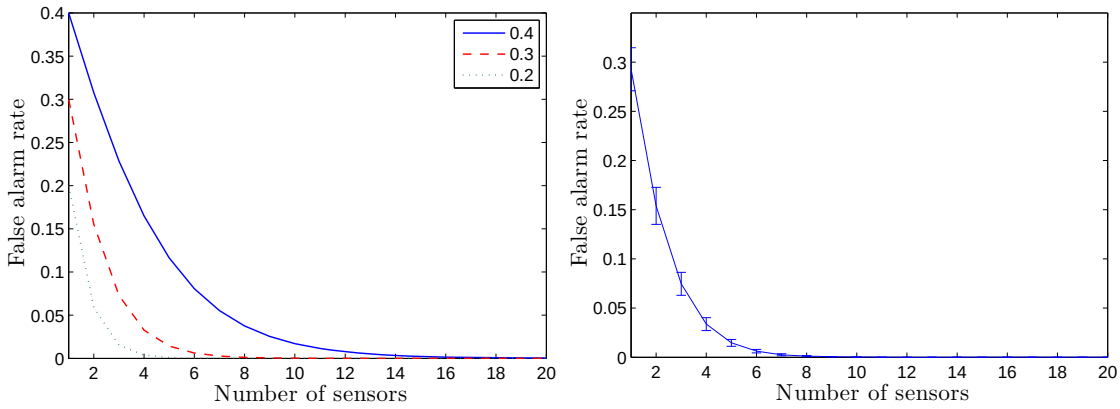
where

$$k_f = \frac{1}{\prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_2)) + \prod_{i=1}^{N}(1 - m_{f_i}^{\Theta^{nt}}(\theta_3)) - \prod_{i=1}^{N} m_{f_i}^{\Theta^{nt}}(\Theta^{nt})}$$

Using Equation (5.4), we plot in Fig. 5.1, with 95% confidence intervals, the evolution of the false alarm rate as a function of the number of sensors (N) involved in the fusion process. Two cases are considered: (i) false alarm rate is a fixed value for all sensors (Fig. 5.1(a)) and (ii) false alarm rate is generated by a Gaussian distribution $\mathcal{N}(1, 1)$ (Fig. 5.1(b)). In both cases, obtained results show that the false alarm rate of the WSN is reduced to zero by adopting our evidence-fusion scheme.

Decision making

Relatively to a space point $p \in RoI$, two pignistic transformations (denoted by $BetP_{d/p}$ and $BetP_{f/p}$) that permit the construction of the probabilities needed for decision making are constructed.



(a) Constant false alarm rate. (b) $\mathcal{N}(1,1)$ false alarm rate.

Figure 5.1: The false alarm rate *vs.* the number of sensors.

The pignistic transformation of the bba $m_d^{\Theta^t}$ denoted by $BetP_{d/p}$ is defined as follows:

$$\begin{aligned} BetP_{d/p}^{\Theta^t}(\theta_0) &= m_d^{\Theta^t}(\theta_0) + \tfrac{1}{2} m_d^{\Theta^t}(\theta_0, \theta_1) \\ BetP_{d/p}^{\Theta^t}(\theta_1) &= m_d^{\Theta^t}(\theta_1) + \tfrac{1}{2} m_d^{\Theta^t}(\theta_0, \theta_1) \end{aligned} \tag{5.5}$$

Similarly, the pignistic transformation of the bba $m_f^{\Theta^{nt}}$ denoted by $BetP_{f/p}$ is defined as follows:

$$\begin{aligned} BetP_{f/p}^{\Theta^{nt}}(\theta_2) &= m_f^{\Theta^{nt}}(\theta_2) + \tfrac{1}{2} m_f^{\Theta^{nt}}(\theta_2, \theta_3) \\ BetP_{f/p}^{\Theta^{nt}}(\theta_3) &= m_f^{\Theta^{nt}}(\theta_3) + \tfrac{1}{2} m_f^{\Theta^{nt}}(\theta_2, \theta_3) \end{aligned} \tag{5.6}$$

To quantify the quality of coverage provided by a WSN, we define the concept of (α,β)-coverage as follows.

Definition 8. *Given two constants $\alpha \in [0,1]$ and $\beta \in [0,1]$, a physical point $p \in RoI$ is (α,β)-covered if:*

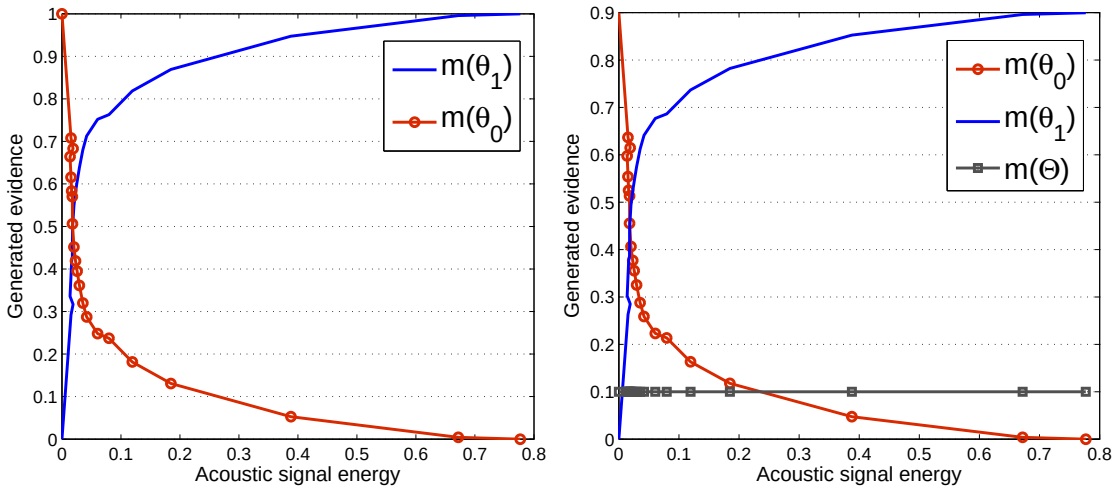
$$\begin{aligned} BetP_{d/p}^{\Theta^t}(\theta_1) &\geq \alpha \\ BetP_{f/p}^{\Theta^{nt}}(\theta_3) &\leq \beta \end{aligned} \tag{5.7}$$

5.2.2 Building Belief Functions

To build belief functions form raw sensing data, we exploit our two-step method described in the previous Chapter. Using the real data traces collected in DARPA SensIT military vehicle detection experiments [41], we plot in Fig. 5.2 a set of mass functions

associated to a WINS NG 2.0 sensor in the case of assault amphibian vehicle detection [41]. For each measurement delivered by a sensor, a distribution of masses, which take into account the uncertainty of its functioning, will be inferred.

For simplicity's sake, the probability mass assigned to target detection is chosen linear with the detection probability. A projection on the set of mass functions is done in order to obtain the corresponding mass function each time a raw data from that sensor is received. For example, according to Fig. 5.2(b), if a sensor sample an acoustic signal energy of 0.3879, then the resulting mass function would have three focal elements: $m(\{\theta_0\}) = 0.0474$, $m(\{\theta_1\}) = 0.8526$, and $m(\Theta) = 0.1$.



(a) Without sensor uncertainty.

(b) With constant sensor uncertainty.

Figure 5.2: Example of a set of mass functions associated to a WINS NG 2.0 sensor in the case of AAV vehicle detection.

Our evidence fusion model can be easily extended to handle deployment-related issues. In the next section, we discuss some deployment-related issues and how they can be included in the framework.

5.3 Discussion

As mentioned previously, our proposal can be extended in many ways. In this section, we discuss how correction mechanisms for weakening or reinforcing belief functions [43, 77, 106] can be included in our framework in order to handle deployment-related issues such as *sensor reliability* and *challenging environments*.

5.3.1 Handling Sensors Reliability

When sensors are vulnerable to misreading or malfunctioning due to their quality, we consider such sensors as only partially reliable. The idea consists in using the reliability rate of a sensor in order to discount its decision. In practice, we do not know for sure whether a sensor is reliable or not, but we have some degree of belief, equal to $(1 - \gamma)$ in this hypothesis, where $\gamma \in [0, 1]$. This can be encoded into a *bba* defined on the set $\{reliable, not\ reliable\}$ such that:

$$m(reliable) = 1 - \gamma \ and \ m(not\ reliable) = \gamma \tag{5.8}$$

Suppose that the *bba* m on Θ represents the sensor report about the actual value of Θ. The result of combining the sensor report with the *bba* given in Equation (5.8) is a new *bba* denoted m^{γ}, defined as:

$$m^{\gamma}(A) = (1 - \gamma) \cdot m(A) \quad for \ \ A \subset \Theta$$

$$m^{\gamma}(\Theta) = \gamma + (1 - \gamma) \cdot m(\Theta)$$

If a priori knowledge on sensor reliability is available, discounting factors associated to the sensors will be used in the evidence combination step. Further, the reliability of a sensor can be updated by the fusion center using methods such as [43].

5.3.2 Handling Harsh Deployment Environments

Although sensors may be of good quality and provide accurate readings, several factors may affect the signal level received by a sensor at a given location [43]. In this case, sensors are vulnerable to misreading due to their locations in the RoI; we call such locations "*challenging locations*".

So far, few researches have tried to deal with this issue in the WSN deployment process. In [46], authors try to handle environment factors such obstacles and weather condition at the post-deployment stage. They assume that sensors know the position of gathered event, by deploying samples in the network. After gathering samples, the sensing range of sensors are determined by a revised geometry structure (α-shape). Different from this work, we aim to consider jointly different forms of challenging environments at the pre-deployment stage. We define two forms of challenging environments and show how our framework can be extended to handle such issues.

Definition 9. ***Type-I obstacles***. *Any energy signal propagating through such an obstacle will be partially or totally absorbed depending on the obstacles' surface characteristics such as trees and man-made structures. This type of obstacles does not generate any form of noise (signal) but rather absorb signals.*

Definition 10. ***Type-II obstacles***. *Any energy signal propagating through such an obstacle will be contaminated. In fact, this type of obstacle generates a form of white noise which will decrease the signal-to-noise ratio (SNR).*

On the one hand, near Type-I obstacles, the signal strength received by a sensor is diminished, so it could be beneficial to give less weight to the information provided by that sensor at the pre-deployment stage, i.e., to discount the sensor's beliefs. The same approach discussed in Section 5.3.1 could be used to handle Type-I obstacles by discounting the belief function $m_d^{\Theta^t}$.

On the other hand, near Type-II obstacles, a sensor tends to generate more false positive than expected. To anticipate this behavior, it could be beneficial to give more weight to the information it provides, i.e., to reinforce the sensor's beliefs. This can be achieved by using the de-discounting operation [77]. Thus, to handle Type-II obstacles, the belief function $m_f^{\Theta^{nt}}$ could be de-discounted. If varying noise levels at the same location at different times is likely, a conservative measure is to consider the worst case scenario.

5.4 Probabilistic Connectivity and Communication Cost Models

As we have done in the previous Chapter, it is customary to assume a binary connectivity model according to which two sensors are able to communicate directly if they are within one communication hop of each other. Although, this model considered the network connectivity problem mainly from a geometric perspective which simplify the analysis it remains limited and not realistic. Empirical studies [78] show that there is no clear cut-off boundary between successful and unsuccessful communication.

In practice, the attenuation experienced by a wireless signal at a given distance is described by the path loss whereas shadowing describes random fluctuations in signal strength at a known path loss. Empirical measurements have indicated that shadowing is a zero-mean normally distributed random variable with standard deviation σ_ϵ. Due to the unique

characteristics of each environment, most radio propagation models use a combination of analytical and empirical methods. One of the most common radio propagation models is the log-normal shadowing path loss model [88] which is given by:

$$PL(d) = PL(d_0) + 10\gamma \log_{10}(\frac{d}{d_0}) + \epsilon \tag{5.9}$$

Where d is the transmitter-receiver distance, d_0 a reference distance, γ the path loss exponent (rate at which signal decays), and ϵ a zero-mean Gaussian distributed random variable with standard deviation σ_ϵ (in dB) that represents the shadowing effects.

The received signal strength (P_r) at a distance d is the output power of the transmitter minus $PL(d)$. Formally:

$$P_r(d) = P_t - PL(d) = P_t - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}) - \epsilon \tag{5.10}$$

Fig. 5.3 shows an analytical propagation model for $\gamma = 2$, $\sigma_\epsilon = 4$, $PL(d_0) = 55dB$, $d_0 = 1$ and an output power $P_t = 0dBm$ (e.g, the Chipcon CC2420 IEEE 802.15.4, 2.4 GHz).

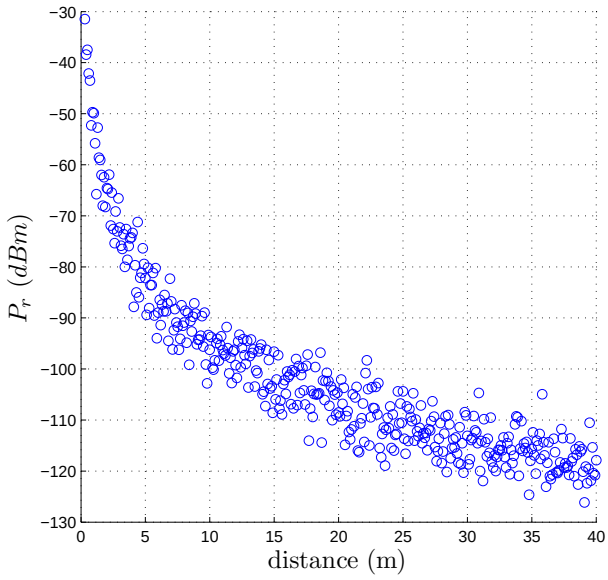


Figure 5.3: Channel Model, $\gamma = 2$, $\sigma_\epsilon = 4$, $P_t = 0dBm$.

From Equation (5.10) we have $P_r(d) \sim \mathcal{N}(P_t - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}), \sigma_\epsilon)$. Since $P_r(d)$ is Gaussian, the probability of a successful communication between two sensors s_i and s_j located at distance d from each other is:

$$\mathcal{P}[P_r(d) > ASS_{min}] = Q(\frac{ASS_{min} - (P_t - PL(d_0) - 10\gamma \log_{10}(\frac{d}{d_0}))}{\sigma_\epsilon})$$

where ASS_{min} represents the minimum acceptable signal strength and Q is the complementary cumulative distribution function (CCDF) of a standard Gaussian, *i.e.*,

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{+\infty} e^{\frac{-t^2}{2}} \mathrm{d}t$$

Hence, the expected number of retransmissions for a successful communication between s_i and s_j is:

$$nRT(d) = \frac{1}{\mathcal{P}[P_r(d) > ASS_{min}]} \tag{5.11}$$

Fig. 5.4 shows the connectivity model related to this formulation. We see clearly that some area within the connectivity range has received power lower than ASS_{min}. Whereas, some area outside the connectivity range has received power higher than ASS_{min}. As mentioned before, this model reflects the fact that there is no clear cut-off boundary between successful and unsuccessful communication.

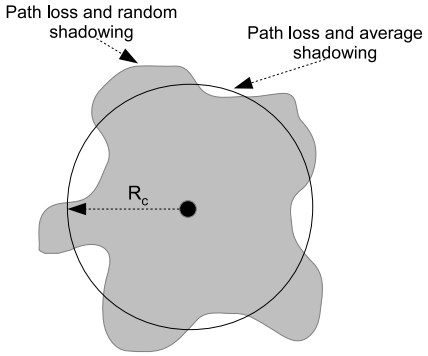


Figure 5.4: Connectivity model.

On the basis of the connectivity model described above, we define the communication cost of a link and a cluster as follows.

Definition 11. *The communication cost of a link i j (denoted by Lc_{ij}) is the expected number of retransmissions needed for a successful communication between the nodes s_i and s_j.*

Formally:

$$Lc_{ij} = nRT(\|ij\|) = \frac{1}{\mathcal{P}[P_r(\|ij\|) > ASS_{min}]}$$

Definition 12. *The communication cost of a cluster (denoted by Cc) is the sum of all communication cost of the links connecting cluster members to the cluster-head (denoted by h). Formally:*

$$Cc_h = \sum_{s_i \in \text{ cluster members}} Lc_{ih}$$

5.5 Sensor Placement Problems

In this section, we formalize different variants of the fusion-based uncertainty-aware wireless sensor networks deployment problem while considering our evidence-based sensing and communication cost models described above.

5.5.1 Network Model and Assumptions

Targets are assumed to appear at a set of known locations referred to as *target points* defining the set $T \subseteq RoI$. T is determined according to user requirements. If full coverage is required, target points can be chosen densely and uniformly in the RoI. The deployment of sensors is constrained to belong to the set $D \subseteq RoI$ consisting of deployment points in the RoI. For every target point $t \in T$, we assume that only a subset of deployed sensors, referred to as Neighbor Sensors where $NS(t) \subseteq D$, participate in the fusion process. NS is a design parameter of the fusion model which is mainly constrained by the communication overhead and will be discussed in Section 5.7.1.

5.5.2 Problems Formulation

We define the fusion-based uncertainty-aware sensor networks deployment (FUSD) problem as the problem of covering a set of target points while considering our evidence-based sensing model defined in Section 5.2.1. Deployment-related issues such as connectivity and network lifespan may be included by considering the probabilistic connectivity and communication cost models defined in Section 5.4.

Handling uniform (α,β)-coverage and cost

In the simplest form of the FUSD problem, the number of sensors should be kept to a minimum while also satisfying the (α,β)-coverage requirements for all target points. This problem (denoted by FUSD-U) can be formulated as an optimization problem:

$$\min \sum_{p \in D} x_p \tag{5.12}$$

$$\text{s.t.} \quad BetP^{\Theta^t}_{d/p}(\theta_1) \geq \alpha, \ \forall p \in T \tag{5.13}$$

$$BetP^{\Theta^{nt}}_{f/p}(\theta_3) \leq \beta, \ \forall p \in T \tag{5.14}$$

$$x_p \in \{0, 1\}, \ \forall p \in D \tag{5.15}$$

Equation (5.12) defines the optimization problem where the objective is to minimize the deployment cost. The solution is constrained in equations (5.13) and (5.14), which requires that each target point is (α,β)-covered. Equation (5.15) defines x_p as a zero-one variable. If $x_p = 1$ than a sensor will be deployed at the deployment point p.

Handling preferential (α,β)-coverage and cost

In this second problem, for each target point $p \in T$, there is an associated predefined minimum event detection probability threshold α_p and a maximum false alarm rate β_p. The second FUSD problem (denoted by FUSD-P) is formulated as follows:

$$\min \sum_{p \in D} x_p$$

$$\text{s.t.} \quad BetP^{\Theta^t}_{d/p}(\theta_1) \geq \alpha_p, \ \forall p \in T$$

$$BetP^{\Theta^{nt}}_{f/p}(\theta_3) \leq \beta_p, \ \forall p \in T$$

$$x_p \in \{0, 1\}, \ \forall p \in D$$

The size of the search space in both formulations, FUSD-U and FUSD-P, can be up to $2^{|D|}$.

Handling preferential (α,β)-coverage, cost, connectivity, and lifespan

In this third problem, we will build on the second deployment problem by adding the communication cost and lifespan objectives. We have to simultaneously optimizes sensing quality, communication cost and network lifespan. Since transmission consumes more energy than any other process, we have to ensure that our sensor placements have reliable links and the overall number of unnecessary retransmissions is minimized. For that, we have to ensure that formed clusters minimize the communication cost as defined in Section 5.4. These objectives are competing. On the one hand the (α,β)-coverage objective will desire "spread out" network topology, where sensors are as far apart from each other as possible in order to minimize the sensing overlap. On the other hand, in order to minimize communication cost and enhance the network lifespan, sensors must not be located too far from each other. This implies a clustered configuration around cluster-heads with a lot of sensing overlap between sensors, yielding a poor (α,β)-coverage.

Our aim is to generate a good cluster-based topology that consume less energy while satisfying the (α,β)-coverage and connectivity constraints. We already know that a good clustering algorithm should guarantee that nodes in the same clusters are highly connected and between clusters are less connected. There are other issues to be addressed when applying clustering approaches, such as the cluster size, the number of orphan (isolated) nodes, etc. Thus, in addition to the objectives considered in the FUSD-P problem, we seek to find the best network topology that:

- Minimizes the overall number of unnecessary retransmissions.
- Minimizes the number of cluster-heads.
- Minimizes the number of isolated sensors (ensures that each sensor gets included in a cluster).
- Minimizes the number of isolated cluster-heads (ensures that each cluster-head has some cluster members).
- Minimizes the overlapping areas among cluster-heads (ensures that the cluster-heads are so distributed or chosen such that there is a minimum overlapping).

This formulation not only enhances the cluster connectivity but also the cluster lifespan, and thus the network lifespan. It is worth noting that the position of the sink(s) could be easily considered in this formulation. For instance, the straightforward approach is to minimize the distances between the cluster-heads and the sink(s). Formally, the third FUSD problem (denoted by FUSD-C) is defined as follows:

$$
\begin{aligned}
&\min \quad \sum_{p \in D} x_p \\
&\min \quad \sum_{p \in \text{ cluster-heads}} x_p \\
&\min \quad \sum_{p \in \text{ isolated sensors}} x_p \\
&\min \quad \sum_{p \in \text{ isolated cluster-heads}} x_p \\
&\min \quad \sum_{\text{cluster-heads}} \text{cluster-heads overlapping} \\
&\min \quad \max_{h \in \text{ cluster-heads}} (Cc_h) \\
&\text{s.t.} \quad BetP^{\Theta^t}_{d/p}(\theta_1) \geq \alpha_p \qquad , \forall p \in T \\
&\qquad\quad BetP^{\Theta^{nt}}_{f/p}(\theta_3) \leq \beta_p \qquad , \forall p \in T \\
&\qquad\quad x_p \in \{0, 1\} \qquad , \ \forall p \in D
\end{aligned}
$$

It is worth noting that the problem in hand is a multi-objective binary non-linear and non-convex optimization problem. In the next section, we investigate the mono-objective cases (FUSD-U and FUSD-P) where the aim is to find the best possible solution. After that, we consider the FUSD-C problem. In this case, our aim is to find the global Pareto-optimal solutions.

5.6 Sensor Placement Algorithms

The three combinatorial non-linear non-convex optimization problems formalized in Section 5.5 are NP-hard [61]. For the mono-objective cases (FUSD-U and FUSD-P), a non-linear programming solver can be applied to the FUSD problem as done in [29] where authors used the CSA solver [118]. The limitations of this technique are as follow: first, only a local optimum is obtained and second, there is a limitation on the number of constraints and hence the size of the RoI that can be handled.

Several interesting approaches like Neural Networks, Artificial Intelligence, Swarm

Optimization, and Ant Colony Optimization have been implemented to tackle such problems. Genetic Algorithm (GAs) are one of the most powerful heuristics for solving optimization problems [109]. In this dissertation, we propose efficient algorithms based on GAs to solve the FUSD problem. GAs have several advantages over other mathematical programming techniques [48]. In the context of mono-objective optimization of sensor placement, some advantages include the following:

– GAs deal directly with a population of solutions at any time. These are spread throughout the solution space, so the chance of reaching the global optimum is increased significantly.
– GAs identify a set of solutions of sensor placements that are close to the minimum cost solution. These sensor placements may correspond to quite different designs that can be then compared in terms of other important but nonquantifiable objectives.

In the multi-objective case, GAs find multiple Pareto-optimal solutions in one single simulation run [109]. In fact, the presence of multiple objectives in a problem, in principle, gives rise to a set of optimal solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution.

In the following, we first discuss a GA-based approach for the mono-objective case, i.e., the FUSD-U and FUSD-P problems, and then we devise another GA-based approach for the multi-objective case.

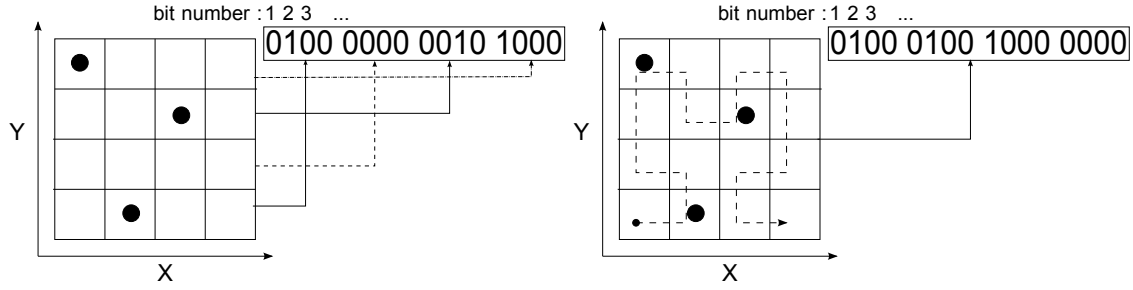
5.6.1 GEBDA

The following subsections describe the detailed steps for modeling of a genetic algorithm to solve the mono-objective FUSD problems, namely: FUSD-U and FUSD-P. This algorithm will be called from now on *GEBDA* (Genetic Evidence-Based Deployment Algorithm).

Problem Representation

In GEBDA, an individual is a candidate sensor placement that specifies the number and locations of sensors. GEBDA encodes the solutions as strings of bits with length $L = |D|$ from a binary alphabet. Each bit corresponds to a deployment point $p \in D$ and "1" means that a sensor will be deployed at that point. A mapping algorithm between 2D (or even 3D) and 1D space is required. A straightforward approach is a linear horizontal (or

vertical) coding (Fig. 5.5(a)). Other mapping algorithms that fairly well preserves locality such as Hilbert space-filling curve (Fig. 5.5(b)) and Z-order curve could also be used. Fig. 5.5 shows the binary representation of a placement of 3 sensors within a 4×4 RoI (here $D = RoI$) using two mapping algorithms. The impact of the mapping algorithm on the performance will be discussed in Section 5.7.1.



(a) Linear horizontal mapping. (b) Hilbert curve mapping.

Figure 5.5: Binary representation of a sensor placement in GEBDA.

For a population of N chromosomes, GEBDA considers each of the N chromosomes in the population in turn. It decodes each chromosome into the corresponding sensor placement and computes its fitness. For next generations, if modification (crossover or mutation) does not occur the previously computed fitness is used.

Constraints Handling

We use penalties [35] to handle our nonlinear inequality constraints. GEBDA assigns a penalty cost for each sensor placement if it does not satisfy the user detection requirements. The infeasibility of an individual represents both the number of active constraints and the extent to which each constraint is violated. As the optimum solution lies on the boundary between feasible and infeasible solutions [48, 109], the penalty cost should be such that near-optimal infeasible solutions are highly fit so that the optimum solution will be approached from both above and below.

A $viol_d$ function that measures the degree of detection rate constraints dissatisfaction is defined according to Equation (5.16).

$$viol_d = \sum_{p \in T} ([\alpha_p - \tilde{\alpha_p}]^+ * \alpha_p). \tag{5.16}$$

Here, $\tilde{\alpha_p}$ denotes the generated detection probability at the target point $p \in T$ (computed

by Equation (5.5)), and $[\alpha_p - \tilde{\alpha_p}]^+$ denotes the projection of $(\alpha_p - \tilde{\alpha_p})$ in $\mathbb{R}^+$. Formally,

$$[\alpha_p - \tilde{\alpha_p}]^+ = (\alpha_p - \tilde{\alpha_p}) \times 1_{\{\alpha_p > \tilde{\alpha_p}\}}.$$

$1_{\{\alpha_p > \tilde{\alpha_p}\}}$ is the indicator function that is equal to 1 if $\alpha_p > \tilde{\alpha_p}$, otherwise it is equal to 0.

Similarly, a $viol_f$ function that measures the degree of false alarm rate constraints dissatisfaction is defined according to Equation (5.17).

$$viol_f = \sum_{p \in T} \frac{[\tilde{\beta_p} - \beta_p]^+}{\beta_p}. \tag{5.17}$$

Here, $\tilde{\beta_p}$ denotes the generated false alarm rate at the target point $p \in T$ (computed using Equation (5.6)).

The *viol* functions values translate how far the solution is from the user detection requirements. The weighting schemes introduced in equations (5.16) and (5.17) allows us to give more weight to risky areas.

Fitness Function

The fitness function combines the deployment cost and the dissatisfaction of detection and false alarm constraints. The goal of this fitness function (Equation 5.18) is to obtain a sensor placement which satisfies the user detection requirements with the minimum number of sensors.

$$fitness = \sum_{p \in D} x_p + C_d \times viol_d + C_f \times viol_f. \tag{5.18}$$

The $viol_d$ function (resp. $viol_f$) is weighted by a penalty factor C_d (resp. C_f). It should be noted that very large values of penalty factors C_d and C_f lead to the death penalty method [35] that just rejects unfeasible solutions from the population.

Genetic Operators

Genetic operators modify the genotype of individuals and thus move them in the search space [109]. In GEBDA, individuals undergo different crossover and mutation operators. There are several crossover techniques, but all involve swapping sequences of bits in the

genotype between two individuals. GEBDA combines three crossover operators with specified relative rates, namely: one-point crossover, two-point crossover and the Uniform crossover (Fig. 5.6). The crossover rate p_c gives the probability that GEBDA will invoke crossover.

After crossover, individuals are subjected to mutation. Mutation of a bit involves flipping it: changing a 0 to 1 or vice versa. GEBDA uses the bitwise mutation with a small probability (p_m). GEBDA treats mutation only as a secondary operator for two reasons: first, this restores lost genetic material and second, it explores parts of the space that crossover might miss and helps prevent premature convergence.

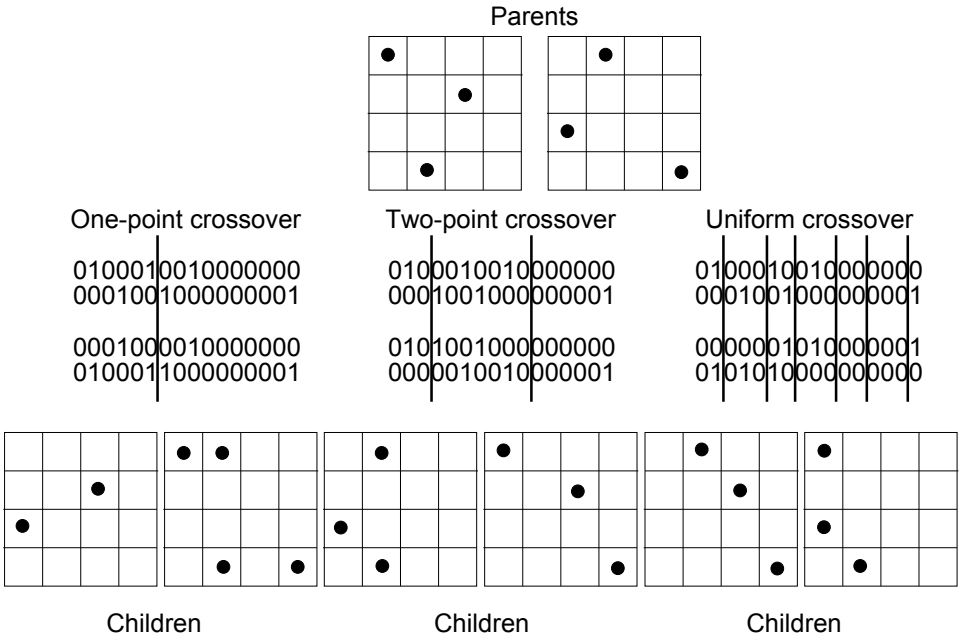


Figure 5.6: Crossover techniques used in GEBDA.

Evolution Engine

The term evolution engine denotes the selection and the replacement schemes [109]. The selection scheme determines how individuals are chosen for mating or mutating, based on their fitness scores. Fitter solutions survive while weaker ones perish. Further, that worse individuals of the population still have a small probability of being selected, and this is important to ensure that the search process is global and does not simply converge to the nearest local optimum. There are several ways for selection [48, 109]. GEBDA employs the binary tournament selection strategy that has several advantages which include

efficient time complexity, low susceptibility to takeover by dominant individuals, and no requirement for fitness scaling or sorting [48, 109].

The replacement phase concerns the survivor selection of both the parent and the offspring populations. In GEBDA, the parents to be killed are chosen by a reverse deterministic tournament. The impact of different selection/replacement strategies on the performance will be discussed in Section 5.7.

Description of GEBDA

The executional steps of GEBDA are as follow:

1. **Step 1**: Create an initial population (of size N) based on random sensor placements.
2. **Step 2**: Iteratively perform the following sub-steps (called a generation) on the population until a stopping criterion is satisfied:
   (a) Analyze the (α,β)-coverage and compute the penalties costs of each sensor placement.
   (b) Compute the fitness of each sensor placement: the fitness of each deployment in the population is taken as the weighted sum of the deployment cost plus the penalties costs.
   (c) Select sensor placements from the population using the evolution engine.
   (d) Create new sensor placements by applying the genetic operators with specified probabilities.
   (e) Generate the next population using the evolution engine.
3. **Step 3**: After the termination criterion is satisfied, the single best sensor placement is harvested and designated as the result of the run.

The stopping criterion can be implemented in several ways such as:
– the number of generations.
– the amount of time.
– minimum fitness threshold.
– fitness has reached a plateau.

The general structure of GEBDA is shown in Algorithm 6.

Algorithm 6 GEBDA

Input: *RoI*, Coverage requirement
Output: Optimal placement
1: Set GEBDA parameters
2: Generate random initial population
3: **repeat**
4: Evaluate (α,β)-coverage for each individual in current population using Equations (5.3)-(5.7)
5: Assign fitness value to each individual using Equations (5.16)-(5.18)
6: Select sensor placements from the population using the evolution engine
7: Create new sensor placements by applying the genetic operators
8: Generate the next population using the evolution engine
9: **until** (Stopping criterion is satisfied)
10: **return** best individual in current population

5.6.2 MGEBDA

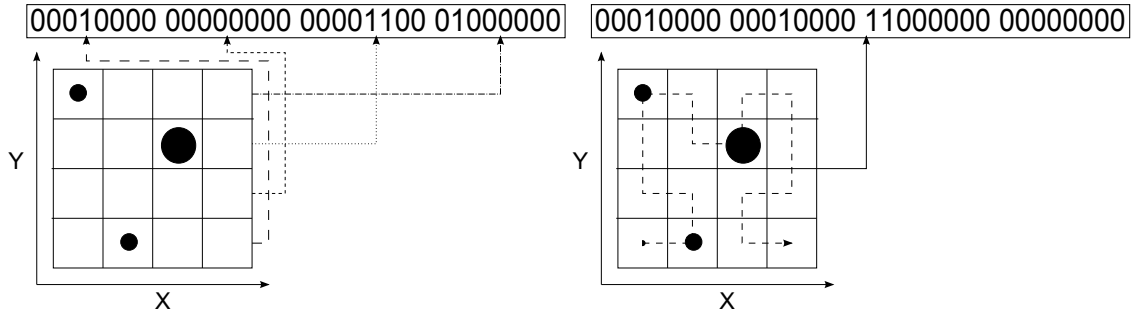
The non-linear and non-convex optimization FUSD-C problem is characterized by the presence of many conflicting objectives. Therefore, it is necessary to look at that problem as a multi-objective optimization problem. Pareto-optimal (non dominated or non inferior) solutions can be obtained for the FUSD-C problem using multi-objective optimization techniques. The solutions belonging to the Pareto-optimal solution set are not dominated by the rest of solutions in the search space. Any solution of the Pareto-optimal front cannot be said to be better than the other solutions in absence of any further information on preference ordering. Therefore, there is a demand to generate Pareto-optimal solutions as much as possible to give more options to the network designer. Classical optimization methods have limitation in handling multi-objective optimization problems, as they convert the multi-objective optimization problem into a single objective optimization problem to generate only a single Pareto-optimal solution.

The advantage of the multi-objective GAs is that it can generate the whole Pareto-optimal solutions in a single simulation [109]. A number of multi-objective GAs are developed in last few years. Among the existing GAs for solving multi-objective optimization problems the most prominent ones are the Pareto archived evolutionary strategy (PAES) [64], Strength pareto evolutionary algorithm (SPEA-2) [150] and the Non-dominated sorting genetic algorithm (NSGA-II) [37]. It has been reported that NSGA-II can converge to the global Pareto-optimal front and can maintain the diversity of population on the Pareto-optimal front [37]. The computational complexity of NSGA-II is also lower than other

multi-objective GAs. For that, on the basis of NSGA-II we devise a new algorithm to solve the multi-objective FUSD problem (FUSD-C). This algorithm will be called from now on *MGEBDA* (Multi-Objective Genetic Evidence-Based Deployment Algorithm).

Problem Representation

In MGEBDA, an individual is a candidate sensor placement that specifies the number and locations of regular sensors and cluster-heads. The length of this bit string is $L = |D|^2$ as two bits correspond to a deployment point $p \in D$ with the following semantic: "00" no sensor deployed, "01" or "10" a regular sensor deployed, and "11" a cluster-head sensor deployed. A mapping algorithm between 2D (or even 3D) and 1D space is required. A straightforward approach is a linear horizontal (or vertical) coding (Fig. 5.7(a)). Other mapping algorithms that fairly well preserves locality such as Hilbert space-filling curve (Fig. 5.7(b)) and Z-order curve could also be used. Fig. 5.7 shows the binary representation of a placement of 3 sensors within a 4×4 RoI (here $D = RoI$) using two mapping algorithms.



(a) Linear horizontal mapping. (b) Hilbert curve mapping.

Figure 5.7: Binary representation of a sensor placement in MGEBDA.

Constraints Handling

As a constraint-handling method, we use the constrained-domination principle [37]. The main idea of this approach is based on a simple modification of the definition of *domination* between two solutions p and q.

Definition 13. *A solution p is said to constrained-dominate a solution q, if any of the following conditions is true.*

1. *Solution p is feasible and solution q is not.*
2. *Solutions p and q are both infeasible, but solution p has a smaller overall constraint violation.*
3. *Solutions p and q are feasible and solution p dominates solution q.*

The effect of using this constrained-domination principle is that any feasible solution has a better non-domination rank than any infeasible solution. All feasible solutions are ranked according to their non-domination level based on the objective function values. However, among two infeasible solutions, the solution with a smaller constraint violation has a better rank [37].

Genetic Operators

As in NSGA-II, MGEBDA uses the bitwise mutation with a small probability (p_m). However, unlike NSGA-II that uses the one-point crossover, MGEBDA combines three crossover operators with specified relative rates, namely: one-point crossover, two-point crossover and the Uniform crossover (Fig. 5.8). The crossover rate p_c gives the probability that MGEBDA will invoke crossover.

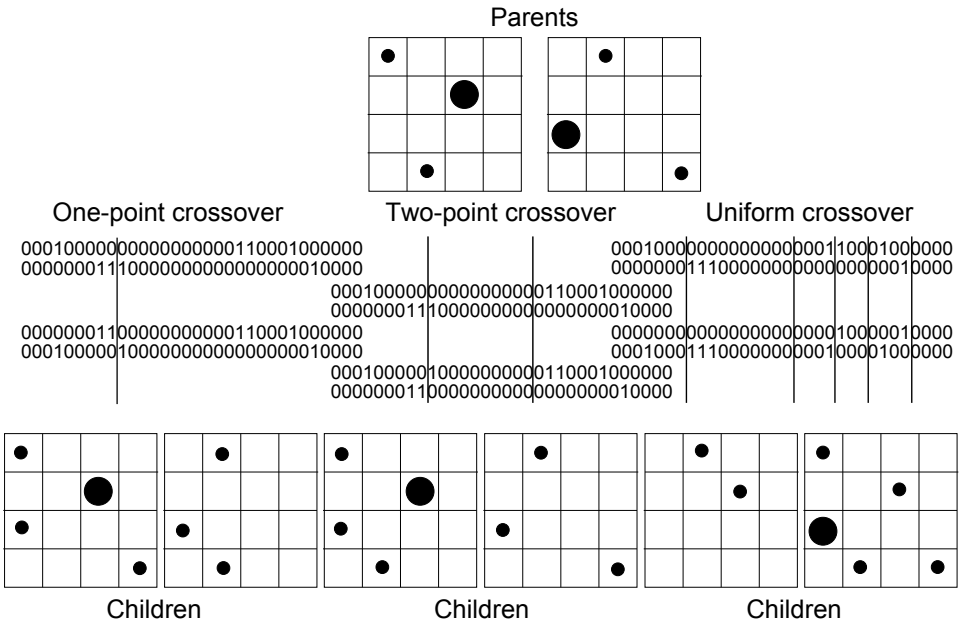


Figure 5.8: Crossover techniques used in MGEBDA.

Evolution Engine

MGEBDA uses the evolution engine of NSGA-II. Selection is done according to the crowding distance operator as follows: a sensor placement p wins the tournament with another sensor placement q if:

1. p has better rank than q, or
2. p and q have the same rank, but p has a large crowding distance than q.

Regarding the replacement strategy, the children and parent population are combined together in order to implement elitism and the constrained non-dominating sorting is applied on the combined population. The old parents are replaced by the better members of the combined population. The solutions of the lower ranking fronts are selected initially to replace the parent population. If all the solutions of a front cannot be accommodated in the parent population, the solutions having large crowding distance will get preference to replace the parent solutions.

Description of MGEBDA

The steps involved in the MGEBDA algorithm are described below.

- **Step 1:** Initialize the population. The initial population (of size N) is generated using random sensor placements.
- **Step 2**: Iteratively perform the following sub-steps (called a generation) on the population until the stopping criterion is satisfied:
    1. Calculate all the objective functions values, separately.
    2. Rank the population using the constrained non-dominating criteria.
    3. Calculate the crowding distance of each solution.
    4. Select sensor placements from the population using the evolution engine.
    5. Create new sensor placements by applying the genetic operators with specified probabilities.
    6. Generate the next population using the evolution engine.
- **Step 3**: After the termination criterion is satisfied, the first front of the final population is harvested and designated as the result of the run.

5.7 Performance Evaluation

The objective of this performance analysis is threefold. (1) It permits an evaluation of the effect of different parameters of the proposed framework on the system performance. (2) It leads to an assessment of the proposed placement algorithms. (3) It permits an objective assessment of the proposed approach in comparison to the state of the art approaches. To evaluate the performance of the proposed approaches, both numerical experiments and trace-driven simulations have been conducted. In the first experiment, we analyze the performance of GEBDA. First, we evaluate the impact of the fusion range on the deployment cost. Second, we analyze GEBDA performances for different parameters settings. Third, we compare GEBDA with the global optimal algorithm and our previous algorithm E2BDA. In the second experiment, in order to quantify the benefit of our approach, we compare GEBDA with two state-of-the-art approaches, namely: *Min_Miss* [152] and *D&C* [29]. In the third experiment, we analyze the performance of MGEBDA. First, we evaluate the impact of the requested connectivity reliability on the deployment cost. Second, we analyze MGEBDA performances for different parameters settings. Third, we compare MGEBDA with our previous approach GEBDA.

5.7.1 GEBDA: Numerical Results

In this section, numerical experiments are conducted to evaluate the performance of our fusion-based uncertainty-aware deployment algorithm, GEBDA, proposed in Section 5.6. First, we evaluate the impact of the fusion range on the deployment cost. Second, we analyze GEBDA performances for different parameters settings. Third, we compare GEBDA with the global optimal algorithm (complete enumeration) and our previous algorithm E2BDA.

GEBDA is coded in ANSI-C++ under the ParadisEO framework [72, 109]. For each experiment, GEBDA was run ten times and the lowest deployment cost is taken as a final result. We consider regularly spaced deployment/target points on a grid thus, the RoI is $n \times n$ grid. Unless otherwise specified, false alarm and detection requirements are uniform over the RoI, with $\alpha = 0.9$ and $\beta = 0.01$.

Impact of fusion radius

The fusion radius is a design parameter of the fusion model, which is mainly constrained by the communication overhead. In this section, we provide several numerical results to help understand the impact of fusion radius. We change the fusion radius from 2 to 9 in a 20×20 RoI. Fig. 5.9 plots the number of sensors required for full $(0.9, 0.1)$-coverage computed by GEBDA versus the fusion radius.

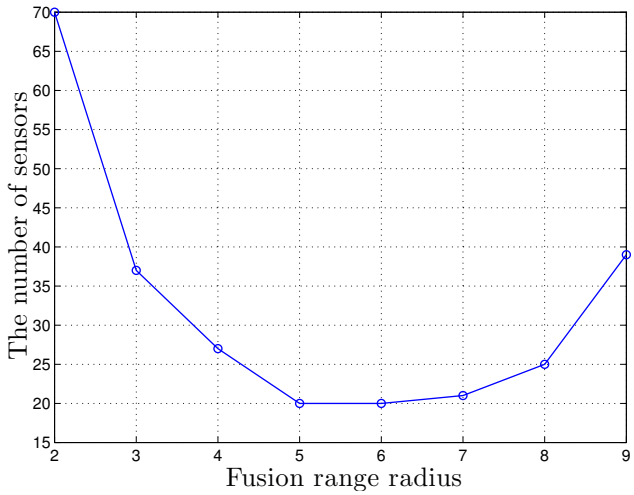


Figure 5.9: The number of sensors *vs.* fusion radius.

On Fig. 5.9, we can see that the number of required sensors drops rapidly from 70 to 20 when the fusion radius increases from 2 to 5, and gradually increases to 39 when the fusion radius becomes larger. Intuitively, as the fusion range increases, more sensors contribute to the data fusion resulting in better sensing quality which reduces the deployment cost. However, as the fusion range becomes very large, the measurements of sensors far away from the target point contain low-quality information and fusing them leads to lower detection performance hence the increase in the deployment cost. This simulation result is consistent with the probabilistic-based numerical results reported in [29, 134].

Impact of the key parameters

One main difficulty in building a practical GA is in choosing suitable values for parameters such as population size, probability of crossover (p_c), and probability of mutation (p_m) [48]. The aim of these experiments is twofold. First, is to figure out suitable values

for GEBDA parameters. Second, is to investigate the performance of GEBDA with different strategies such as: selection strategies, replacement strategies and mapping algorithms. More specifically, the total of our parameters was:

1. Population sizes: 20, 30, 50, 100, 200, 400, 800, and 2000
2. Crossover probability: [0.6 - 0.95] with step 0.05
3. Mutation probability: [0.01 - 0.20] with step 0.01
4. Crossover mechanism: $\frac{1}{4}$ one point crossover + $\frac{1}{4}$ two point crossover + $\frac{1}{2}$ uniform crossover
5. Selection strategies: Binary tournament, Ranking
6. Replacement strategies: Generational, Steady-state
7. Elitism: Yes/No
8. Mapping algorithms: Linear horizontal (L-mapping) and Hilbert space-filling curve (H-mapping).

In a 20×20 RoI, we vary each of the above parameters once at the time to analyze its impact on GEBDA performance. In what follows, we summarize the obtained results.

Fig. 5.10 shows that GEBDA performance improves when the population size is changed. A small size provides an insufficient sample which makes GEBDA performs poorly. A large population size will undoubtedly raise the convergence of GEBDA. However, a large population requires more evaluations per generation which makes GEBDA execution slow. We also analyzed the impact of population size for different RoI. We found a population size of $2 \times L$ to be a good trade-off.

Table 5.1 shows GEBDA performance for different crossover probabilities. We found a crossover rate about 0.6 to be the best. On the other side, best mutation rate should be very low about 0.01-0.03.

Table 5.1: Results for different crossover probabilities

p_c	**0.6**	0.65	0.7	0.75	0.8	0.85	0.9	0.95
No. of sensors	21	21	21	22	21	21	21	21
Running time (s)	111	132	130	189	163	190	191	220

We investigate the performance of GEBDA with different evolution engines in terms of obtained deployment cost and running time. Two selection strategies are considered:

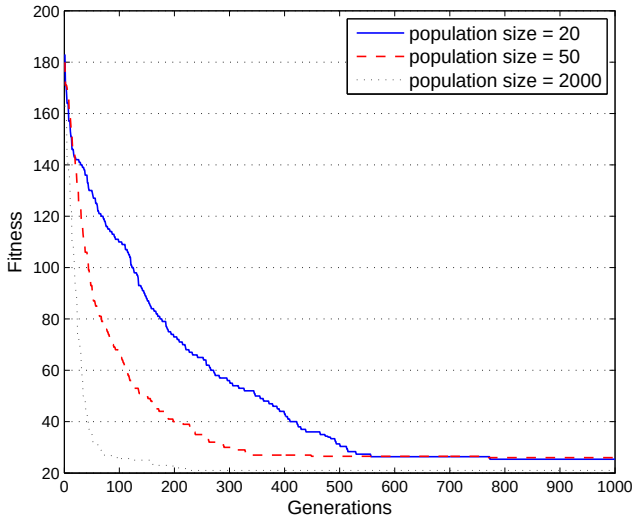


Figure 5.10: Results for different population size.

the binary tournament selection (denoted by Tour) and the rank-based selection (denoted by Rank). The selective pressure used in the rank-based selection is set to 2 for all runs (because we found a value of 2 to be the best). Three replacement strategies are considered: first, the generational replacement (denoted by Gen) which selects the best offspring. In the second replacement strategy (denoted by Worst), the worse parents are killed and replaced by all offsprings. In the third replacement strategy (denoted by Det), the parents to be killed are chosen by a reverse deterministic tournament. We also consider the impact of weak elitism on non-elitist replacement strategies. Obtained results are summarized in Table 5.2.

Table 5.2: Results for different evolution engines

Evolution Engine		**20x20 RoI**		**30x30 RoI**		**50x50 RoI**	
Selection	Replacement	Cost	Time	Cost	Time	Cost	Time
Tour	Gen	22	97	49	1044	111	21664
Tour	Gen+Eli	22	116	50	824	115	18860
Rank	Gen	22	122	50	894	115	28612
Rank	Gen+Eli	21	115	49	1013	116	16735
Tour	Worst	23	96	48	1047	113	25945
Tour	**Det**	**21**	110	**48**	916	**111**	21716
Rank	Worst	22	124	49	1072	111	19818
Rank	Det	22	124	51	777	113	22650

Table 5.2 shows that the evolution engine composed by the binary tournament selection and the Det replacement strategy offers a good performance in all considered cases.

We also perform extensive simulations to analyze the impact of the two mapping algorithms (L-mapping and H-mapping) on GEBDA performance. Fig. 5.11 shows the result obtained for a 64×64 RoI. The overall obtained results indicate no significant difference.

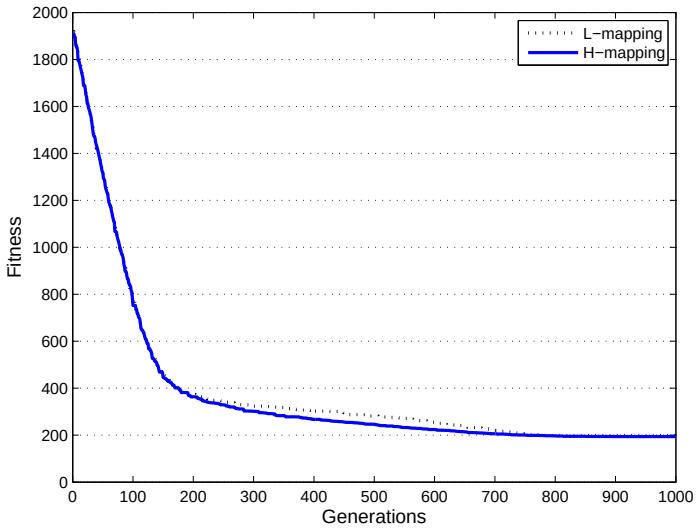


Figure 5.11: L-mapping *vs.* H-mapping.

GEBDA *vs.* Complete Enumeration

To evaluate the efficiency of GEBDA, we compare it with the global optimal algorithm. We launched height scenarios with different RoI sizes: 3×3, 4×4, ..., 10×10. Table 5.3, shows the minimum number of sensors needed to meet the full $(0.9, 0.1)$-coverage and the running time for different problem sizes when using the two algorithms: GEBDA and the global optimal algorithm.

From Table 5.3, we can clearly see that the running time of the global optimal algorithm increases exponentially with respect to the RoI size. This is mainly due to the size of the candidate solutions set. Also, we note that GEBDA offers a comparable performance to that of the global optimal algorithm but with very lower computational complexity. For example, for a 10×10 RoI, GEBDA finds the global optimum solution in 4.06 *s*, whereas the global optimal algorithm needs more than four days. In all cases, GEBDA finds the global optimum in relatively few evaluations compared to the size of the search space.

Table 5.3: GEBDA *vs.* Complete Enumeration

	GEBDA		**Complete enumeration**	
RoI	Running time	No. of sensors	Running time	No. of sensors
3×3	≈ 0 ms	2	≈ 0 ms	2
4×4	31 ms	2	15 ms	2
5×5	78 ms	3	140 ms	3
6×6	218 ms	3	750 ms	3
7×7	312 ms	4	37.66 s	4
8×8	765 ms	5	25.53 m	5
9×9	2.73 s	6	3.15 h	6
10×10	4.06 s	7	4.23 d	7

The complexity of the global optimal algorithm increases exponentially with respect to the RoI size. Therefore, for a large-scale placement problem, the global optimal solution becomes prohibitively expensive. For further comparison, we employ our previous algorithm E2BDA as another baseline.

GEBDA *vs.* E2BDA

In this experiment, we compare the performance of GEBDA to that of E2BDA which is based on a dynamic programming approach. It should be noted that E2BDA has been compared to the state-of-the-art deployment strategies. The results reported in the previous Chapter show that E2BDA obtains the best performances. Table 5.4 shows the minimum number of sensors needed to meet the full $(0.9, 0.1)$-coverage and the running time for different problem sizes when using the two algorithms: GEBDA and E2BDA.

From Table 5.4, we see clearly that GEBDA consistently outperforms E2BDA in terms of required number of sensors. Regarding the running time for moderate problem size (up to 40×40 RoI), E2BDA still run faster than GEBDA but, beyond that GEBDA consistently outperforms E2BDA. Further, the difference in terms of performance increases with the problem size which clearly shows the advantage of the genetic approach for very large problems.

Table 5.4: GEBDA *vs.* E2BDA

	GEBDA		**E2BDA**	
RoI	Running time	No. of sensors	Running time	No. of sensors
10×10	4.06 s	7	359 ms	10
15×15	20.43 s	13	3.78 s	16
20×20	115.19 s	21	23.81 s	24
30×30	21.73 m	44	8.09 m	51
40×40	1.45 h	73	1.15 h	85
50×50	5.01 h	112	6.38 h	129
60×60	14.84 h	161	28.21 h	189
100×100	$\approx$ 2 d	373	$\approx$ 40 d	444

5.7.2 GEBDA: Trace-driven Simulations

We conduct extensive simulations using the real data traces collected in the DARPA SensIT military vehicle detection experiments [41]. The dataset used in our simulations include the location information for 18 nodes, the ground truth information of the vehicles recorded by a GPS device and the acoustic signal energy measurements recorded by 18 nodes when military vehicle drives through the RoI. Two types of military vehicles are considered, namely: Assault Amphibious Vehicle (AAV) and Dragon Wagon (DW). For detailed setup of the experiments, we refer to [41]. The data traces used in our simulations include time series recorded for nine vehicles (AAV3-11) recorded by 18 nodes at a sampling period of 0.75 *s*.

To quantify the benefit of our approach, we compare it with two state-of-the-art approaches, namely: *Min_Miss* (denoted by MM) [152] and *D&C* [29]. The MM algorithm [152] is a non-fusion sensor placement algorithm based on a probabilistic coverage model. On the other hand, the D&C algorithm [29] is a recent fusion-based sensor placement algorithm based on a value-fusion approach. In the simulations, each simulated sensor is associated with a real sensor randomly selected. When a simulated sensor samples an energy measurement, we compute the sensor-vehicle distance. The sensor's measurement is then set to be the real measurement gathered by the associated real sensor at the same distance to the vehicle. This approach has been used in many recent research works [29, 41, 134]. Afterward, this energy measurement is used by the sensor to build its beliefs about vehicle presence as described in Section 5.2.2.

For a fair comparison, the parameters of MM, D&C and GEBDA are estimated on the

same dataset [41]. For different RoI sizes, the three approaches are compared in terms of three metrics: (i) the minimum number of sensors needed to meet the full $(0.9, 0.01)$-coverage (deployment cost), (ii) the rate of successful detection (true positive), and (iii) the rate of false alarm (false positive). The obtained results are summarized in Table 5.5, with 95% confidence intervals. It should be noted that there exists deviation between user detection requirements and generated performance when using real data. This deviation is due to differences between the measurements of sensors and the analytical coverage model estimated by the training data and is caused by various reasons including terrain and changing noise levels due to wind as already reported in [29].

Table 5.5: Trace-driven simulations results

		RoI			
		7x7	**10x10**	**15x15**	**20x20**
MM	Deployment cost	3	4	6	9
	True positive (%)	100	97.51 ± 1.12	99.28 ± 0.25	99.56 ± 0.20
	False positive (%)	36.75 ± 6.64	41.23 ± 7.92	44.5 ± 8.41	39.22 ± 8.08
D&C	Deployment cost	4	5	14	30
	True positive (%)	98.71 ± 1.44	95.79 ± 2.16	99.51 ± 0.40	99.71 ± 0.17
	False positive (%)	21.3 ± 6.35	15.04 ± 5.84	31.31 ± 7.53	26.51 ± 6.27
GEBDA	Deployment cost	4	8	14	24
	True positive (%)	96.76 ± 0.95	92.5 ± 1.46	95.83 ± 1.86	92.66 ± 1.10
	False positive (%)	**4.95 ± 3.01**	**7.4 ± 3.94**	**8.67 ± 4.17**	**2.36 ± 1.19**

Regarding the true positive, the three strategies ensure the user detection requirements. Further, the MM strategy tends to employ fewer sensors than the two other strategies as no fusion scheme is used. However, on average the MM strategy generates 40.42 ± 7.76 % false alarm which is unacceptable for critical applications that require a low rate of false alarm. The value-fusion scheme used in D&C reduces the false alarm rate, on average to 23.54 ± 6.50. Our evidence-fusion scheme not only improves sensing coverage by wisely exploiting the collaboration among sensors but also reduces the false alarm rate, on average to 5.84 ± 3.08.

The first drawn conclusion is that fusion improves coverage quality not coverage. If false alarms are not an issue for the target application, probabilistic placement without fusion should be used. Otherwise, a fusion-based placement should be employed. In both cases, the analytical coverage model assumed at the pre-deployment stage should approximate real coverage properties well enough to lead to accurate estimation of application

performance. The second drawn conclusion is that evidence-fusion improves significantly coverage quality by exploiting the collaboration among sensors and it outperforms the value-fusion approach.

5.7.3 MGEBDA

In this section, numerical experiments are conducted to evaluate the performance of MGEBDA. First, we analyze MGEBDA performances for different parameters settings. Second, we evaluate the impact of the requested connectivity reliability on the deployment cost. Third, we compare MGEBDA with our previous approach GEBDA.

MGEBDA is coded in ANSI-C++ under the ParadisEO framework [72, 109]. We consider regularly spaced deployment/target points on a grid thus, the RoI is $n \times n$ grid. Unless otherwise specified, false alarm and detection requirements are uniform over the RoI, with $\alpha = 0.9$ and $\beta = 0.01$. The simulations parameters are described in Table 5.6.

Table 5.6: Simulations Parameters

Parameters	**Values**
n	$10 - 100$
α, β	$0.9, 0.01$
Fusion radius	5
ASS_{min}, γ, σ_ϵ	$-70dBm$, 2, $4dB$
$PL(d_0)$, d_0, P_t	$55dB$, $1m$, $0dBm$
$popSize$, $maxGen$	400, 250

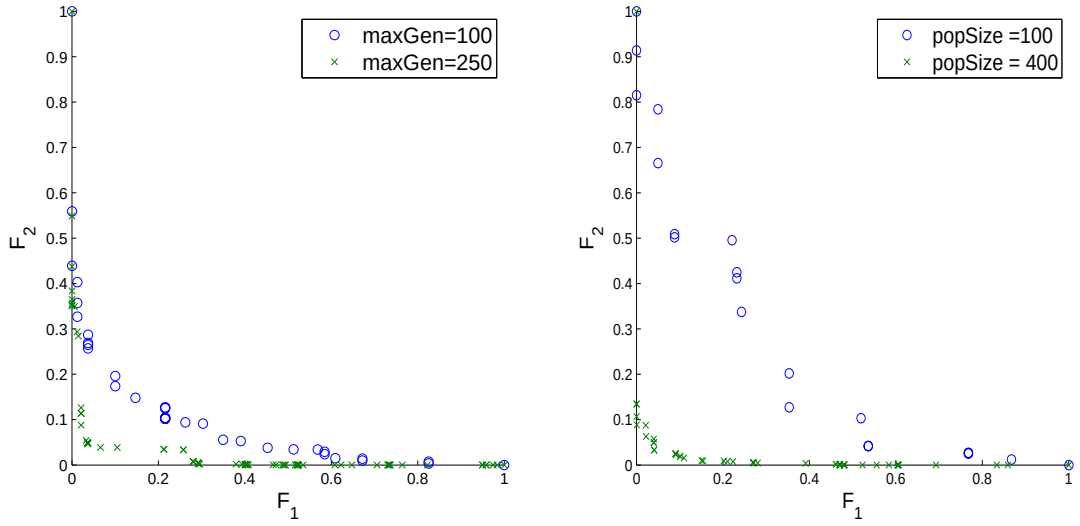
To assess the performance of MGEBDA, or multi-objective algorithms in general, several metrics appeared in the literature [90]. However, most of the current proposals [90] assume that the global Pareto front of the multi-objective optimization under study is known which is not our case. Further, the few metrics that can be employed when the global Pareto front is unknown, such as the spread metrics, only judge the uniformity of solution points over different subregions without indicating the overall quality of solutions.

For the reasons discussed previously, we propose to plot the Pareto fronts and inspect visually their relative locations, relative number of points in each front and their distribution to assess the performance of MGEBDA for different parameter settings. For illustration purpose, the objectives considered in the FUSD-C problem were grouped into two objective

functions F_1 and F_2. The function F_1 measures the deployment cost along with the degree of detection rate constraints dissatisfaction, and the degree of false alarm rate constraints dissatisfaction. The function F_2 measures all other objectives. Both functions, F_1 and F_2, are to be minimized.

Impact of the key parameters

In this section, we perform some experiments to show the effect of a couple of different parameter settings on the performance of MGEBDA. Precisely, we analyze the effect of population size and the number of maximum generations on the performance of MGEBDA. For that, first we keep all other parameters, but increase the number of maximum generations from 100 to 250. Second, we increase the population size from 100 to 400 while keeping all other parameters unchanged.



(a) Pareto frontier with different number of maximum generations.

(b) Pareto frontier with different population size.

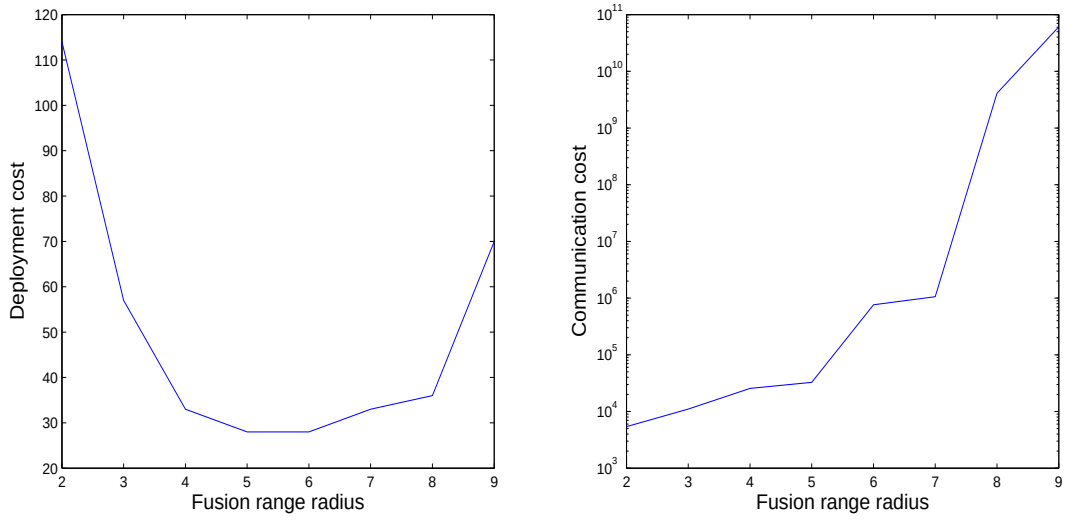
Figure 5.12: Pareto frontier for MGEBDA with different parameter settings.

Fig. 5.12 shows one of the ten runs of MGEBDA with different parameter settings. First, this figure demonstrates the abilities of MGEBDA in converging to the true front and in finding diverse solutions in the front. Second, increasing the population size and/or the number of maximum generations help MGEBDA to converge very close to the true Pareto-optimal front and maximize the number of elements of the Pareto-optimal set found.

For instance, with 100 (resp. 250) generations, MGEBDA finds 35 (resp. 81) Pareto-optimal solutions. A large population size and/or number of maximum generations will undoubtedly raise the convergence of MGEBDA. However, it will require more evaluations per generation which makes MGEBDA execution slow. We also analyzed the impact of these parameters for different RoI. We found a population size of $2 \times L$ and a number of maximum generations of 500 to be a good trade-off.

Impact of fusion radius

As discussed previously, the fusion radius is a design parameter of the fusion model, which is mainly constrained by the communication overhead. In this section, we provide several numerical results to help understand the impact of fusion radius. We change the fusion radius from 2 to 9 in a 20×20 RoI. Fig. 5.13 plots the deployment cost and the communication cost (as stated by Definitions 11 and 12) required for full $(0.9, 0.01)$-coverage computed by MGEBDA versus the fusion radius. It should be noted that as MGEBDA generates several Pareto-optimal solutions, we consider, each time, the best solution in terms of deployment cost.



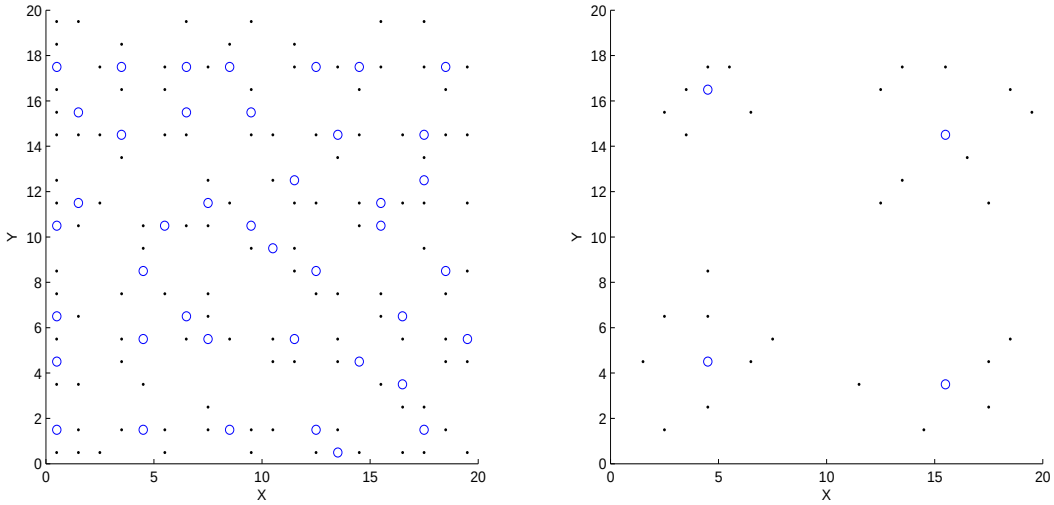
(a) Deployment cost.

(b) Communication cost.

Figure 5.13: The impact of fusion radius on performance.

On Fig. 5.13, we can see that: (1) the deployment cost drops rapidly from 114 to 32 when the fusion radius increases from 2 to 6, and gradually increases to 70 when the fusion

radius becomes larger. (2) as the fusion radius increases the communication cost increases exponentially. Intuitively, as the fusion range increases, more distant sensors contribute to the data fusion resulting in better sensing quality which reduces the deployment cost in terms of regular sensors and cluster-heads. A visualization of a sensor placement generated by MGEBDA is depicted in Fig. 5.14. We see clearly that fewer regular sensors and cluster-heads are employed in the case of a fusion radius of 6 as compared to the case of a fusion radius of 2.



(a) Fusion radius of 2.

(b) Fusion radius of 6.

Figure 5.14: The impact of fusion radius on the generated topologies.

However, as the fusion range becomes very large, the measurements of sensors far away from the target point contain low-quality information and fusing them leads to lower detection performance hence the increase in the deployment cost. In all cases, as the distance between sensors and their cluster-head increases, the communication cost increases.

The effect of the requested connectivity reliability

In this experiment, we analyze the effect of the requested connectivity reliability on the deployment cost (i.e., the number of deployed sensors and cluster-heads). For that, in a 20×20 RoI, we vary the minimum acceptable signal strength (ASS_{min}) and we analyze the deployment cost. Table 5.7 summarizes the obtained results.

We notice that increasing the requested connectivity reliability does not increase the

Table 5.7: The effect of the requested connectivity reliability.

ASS_{min} **(dBm)**	**Number of sensors**	**Number of CHs**
-90	28	5
-60	27	8
-50	27	9

number of regular sensors. However, in terms of number of clusters, the structure of the generated topology was highly affected. In fact, we can clearly see that increasing the requested connectivity reliability increases the number of cluster-heads and therefore increases the number of clusters. This can be explained by the fact that to ensure a high connectivity reliability, the distances separating sensors members and their corresponding cluster-heads are required to be small to guarantee links with good quality which, inevitably, reduces the size of the clusters. However, these good links reduce the communication cost of the network. To summarize, when increasing the requested connectivity reliability, MGEBDA tends to generate small clusters (in terms of distances separating sensors members and their corresponding cluster-heads).

MGEBDA *vs.* GEBDA

As mentioned above, in addition to GEBDA, all related deployment strategies [1, 4, 29] ignores completely the issues of network connectivity and network lifespan. Previously, we show that GEBDA outperforms related deployment strategies. In this experiment, we vary the size of the RoI and we compare MGEBDA to GEBDA in terms of deployment cost. For MGEBDA, we set the minimum acceptable signal strength to $-70dBm$.

It should be noted that unlike GEBDA, MGEBDA generates a set of non-dominated solutions. Thus, we are able to select one topology among several. For this purpose, it is necessary to define a priority for each of the above considered objectives. In this experiment, we select the topology which has the lowest deployment cost. Obtained results are summarized in Table 5.8

We notice that MGEBDA ensures full satisfaction of (α,β)-coverage and minimizes the communication cost of the network which enhances the network lifespan. However, MGEBDA does not offer the lowest deployment cost. Indeed, more sensors are deployed by MGEBDA when compared with GEBDA. This is due to the consideration of the additional constraints: connectivity and lifespan.

Table 5.8: MGEBDA *vs.* GEBDA.

RoI	MGEBDA	GEBDA
10×10	10	8
15×15	18	14
20×20	32	24
30×30	69	50
40×40	118	86

5.8 Conclusion

In this Chapter, a unified approach for robust deployment of fusion-based WSNs is proposed. Our approach is based on the Dempster-Shafer theory that captures several characteristics of real-world applications and allows an efficient management of the uncertainty related to sensors readings. This distinguishes our effort from existing sensor deployment algorithms that are either based on ideal detection models or rely on simple fusion schemes. Some possible extensions to this work were discussed. For instance, correction mechanisms for weakening or reinforcing belief functions can be included in our framework in order to handle deployment-related issues such as sensor reliability and challenging environments.

We also devised probabilistic connectivity and communication cost models, and we addressed the fusion-based WSNs deployment problem, taking into the account a rich set of objectives and constraints, namely, cost of deployment, (α,β)-coverage, network connectivity, data fidelity, sensors reliability, harsh deployment environments, and network lifetime. Experiments carried out on both synthetic data set and data traces collected in a real experiment for vehicle detection demonstrate the interest of the evidence fusion model that considers in a meaningful way the information on the quality of sensors' decisions as well as the reliability of these sensors along with their uncertain and imprecise decisions. The obtained results show that our proposal outperforms all the existing strategies.

To further quantify the real benefit of our approaches, we developed, *ArduiNet*, a real WSN testbed for motion detection comprising more than 40 static and mobile sensor nodes. In the next Chapter, we will present the work carried out toward the design, development and implementation of ArduiNet. We will also report the obtained results, supported by comparison with other works.

Chapter 6

Design and Implementation of a Surveillance WSN Testbed

6.1 Introduction

In the previous Chapters, our proposals were validated through extensive experiments carried out on both synthetic data set and real data traces collected in a vehicle detection experiment. To further quantify the real benefit of the proposed approaches, we developed, *ArduiNet*, a real WSN testbed for motion detection comprising more than 40 static and mobile sensor nodes. In this Chapter, we present the work carried out toward the design, development and implementation of ArduiNet in the LiSSi laboratory. We also report the obtained results, supported by comparison with other works.

6.2 Hardware and Software

Our testbed is based on the Arduino platform [1] which is an open-source fast prototyping platform based on flexible, easy-to-use hardware and software. This section provides the details of hardware and software components of our testbed.

1. http://arduino.cc/

6.2.1 Hardware

The need for a programmable and extensible platform drove us toward the Arduino platform. Its core consists of a programmable microcontroller that can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. Additional modules, called "shields", can be used in order to extend the Arduino capabilities. Particularly relevant for our testbed are the extensions providing networking functionality (e.g., WiFi, Bluetooth, IEEE 802.15.4, and ZigBee).

Arduino UNO board

The Arduino Uno (Fig. 6.1) is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs (each of which provide 10 bits of resolution, i.e., 1024 different values), a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. The ATmega328 has 32 KB (with 0.5 KB used for the bootloader), 2 KB of SRAM, and 1 KB of EEPROM.

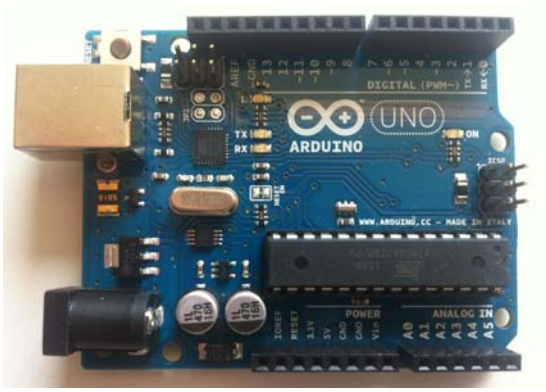


Figure 6.1: The Arduino Uno board.

Shields

Shields are boards that can be plugged on top of the Arduino PCB [2] extending its capabilities. Two shields are used (Fig. 6.2): First, the Wireless Proto shield which allows an Arduino board to communicate wirelessly using a wireless module. It is based on the

2. Printed Circuit Board

Xbee modules from Digi [3], but can use any module with the same footprint. Second, the Arduino WiFi shield that allows an Arduino board to connect to wireless networks using the 802.11b and 802.11g specifications. It contains also an onboard micro-SD card slot, which can be used to store files for serving over the network.

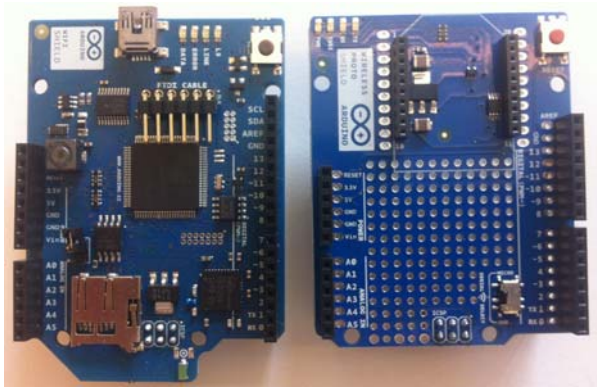


Figure 6.2: The Arduino WiFi (left) and the Wireless Proto Shields (right).

Communication modules

In addition, to the Arduino WiFi shield, the following communication modules are used with the Wireless Proto shield: the XBee 802.15.4 (series 1), the XBee ZB (series 2), and the Bluetooth Bee (Fig. 6.3).

Figure 6.3: The XBee 802.15.4 (left), the XBee ZB (center), and the Bluetooth Bee (right).

Sensors

For motion detection, we connect a passive infrared (PIR) motion sensor to the Arduino Uno. PIR sensor detects changes in infrared radiation which occur when there is movement

3. http://www.digi.com/

by a person (or object) which is different in temperature from the surroundings. As sensor detects temperature differences, it is well suited to detecting the motion of people by their body temperature. We have configured an analog (Phidgets 1111_0) and a digital (Parallax 555-28027) sensor for our testbed. Additional components such as a color serial JPEG camera module can be easily integrated. Fig. 6.4 shows a typical ArduiNet static node.

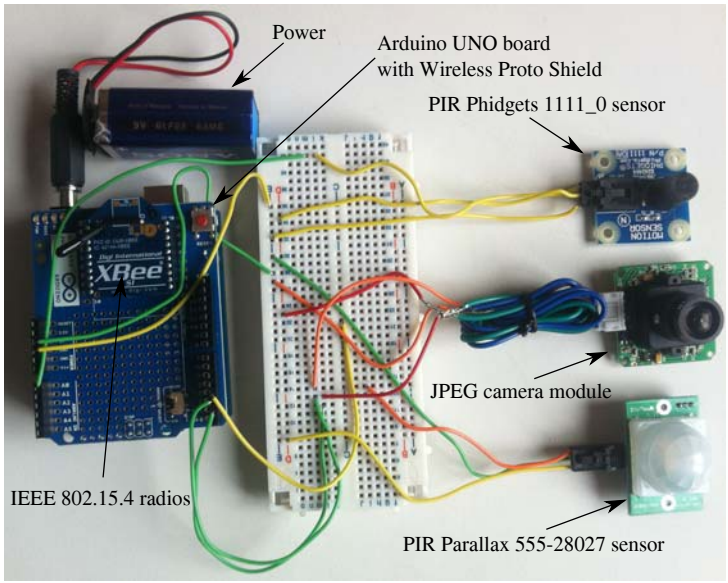


Figure 6.4: Static node.

Mobile nodes

The mobile node (Fig. 6.5) is based on the DFRobotShop Rover which is a versatile robot tank based on the Arduino microcontroller. It uses the popular Tamiya twin motor gearbox and the Tamiya track and wheel set. The DFRobotShop Rover PCB incorporates a standard Arduino (ATMega328 SMT), an L293B motor driver (connected to pins 5 to 8), voltage regulator and prototyping area, and contributes to the mechanical structure of the robot.

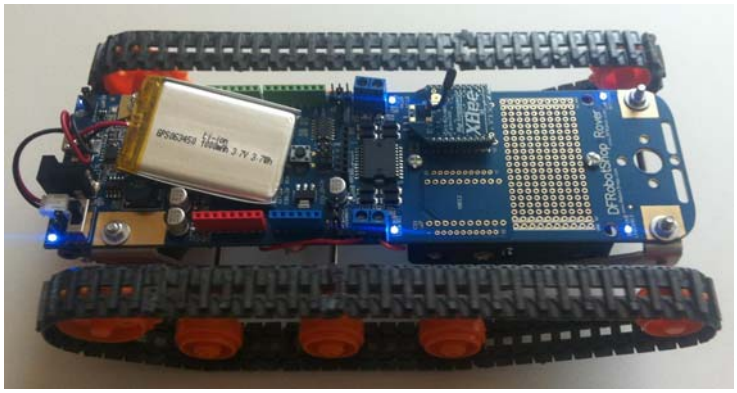


Figure 6.5: Mobile node.

6.2.2 Software

The microcontroller is programmed using the Arduino programming language (based on Wiring [4]) and the open-source Arduino development environment (based on Processing [5], avr-gcc, and other open source software). The Arduino IDE makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. We use the Arduino IDE to create a sketch [6] which is the name that Arduino uses for a program. Fig. 6.6 shows the Arduino IDE running in Ubuntu.

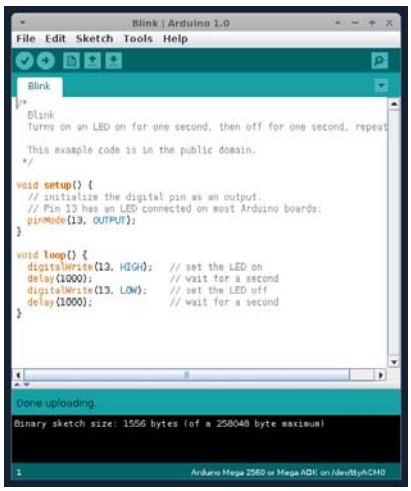


Figure 6.6: The Arduino IDE running in Ubuntu.

4. http://wiring.org.co/
5. http://www.processing.org/
6. It's the unit of code that is uploaded to and run on an Arduino board.

6.3 System Architecture

ArduiNet is deployed in a lab room as shown in Fig. 6.7. A $2.4m \times 4m$ RoI is discretized by two-dimensional metallic grid. The granularity of the grid, i.e., the distance between consecutive grid points is $0.4m$. Each cell's barycenter is a possible location for a sensor as illustrated in Fig. 6.7.

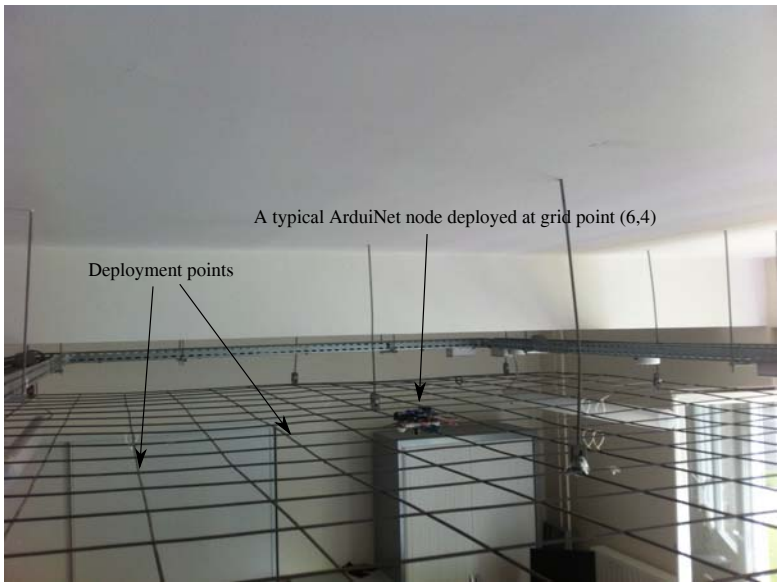


Figure 6.7: System architecture.

The topologies generated by our framework can be easily deployed on that metallic grid for testing purposes. In the next section, we report the obtained results, supported by comparison with other works.

6.4 Experiments and Results

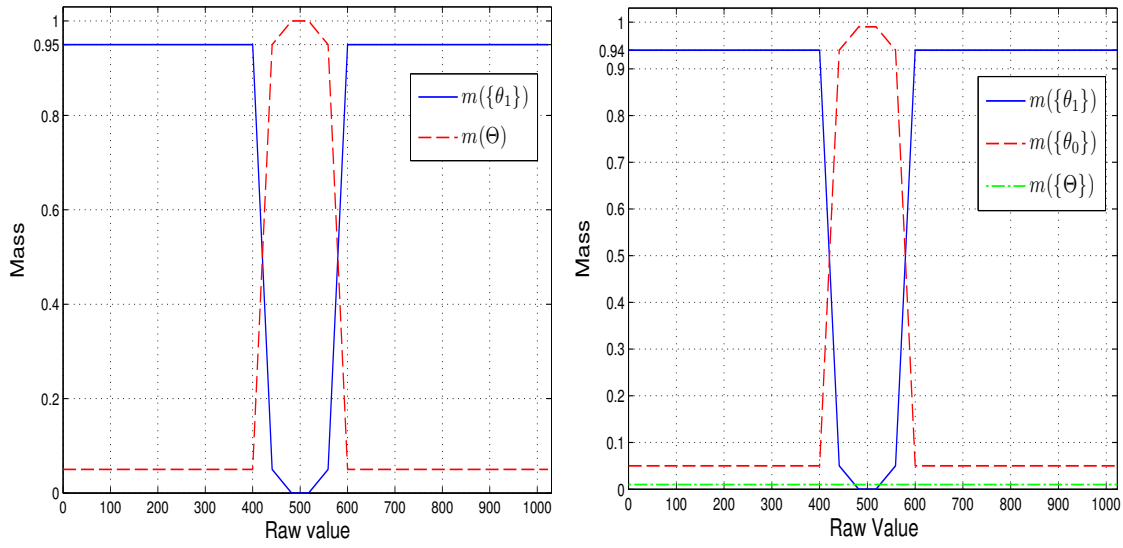
In this section, we present the experiments and discuss the obtained results. First, we discuss our effort to build belief functions from raw sensory data of the PIR Phidgets 1111_0 sensor. Second, we examine the properties and performance of the IEEE 802.15.4 to calibrate the proposed communication cost model. Finally, we analyze ArduiNet as a surveillance system while using classical approaches and our evidence-based approaches.

6.5 Coverage Models Analysis

We analyze the performance of the PIR Phidgets sensor in both indoor and outdoor environments. This study will allow us to correctly calibrate the sensor coverage model which is an input to our deployment framework.

6.5.1 Building Belief Functions From Raw Sensory Data

Fig. 6.8 gives an example of set of mass functions associated to the PIR Phidgets sensor in an indoor environment. When there is no movement, this sensor provides a value around 500 (value gauged by the manufacturer). Any other measure is equivalent to its symmetrical around 500. Numbers outside the 400 to 600 range denote the detection of a moving object with a high certainty. A projection on the set of mass functions is done in order to obtain the corresponding mass function each time a raw data from that sensor is received. For instance, in the case of the first formulation, if the Phidgets sensor returns a value of 100, then the resulting mass function would have two focal elements: $m(\{\theta_1\}) = 0.95$ and $m(\Theta) = 0.05$.

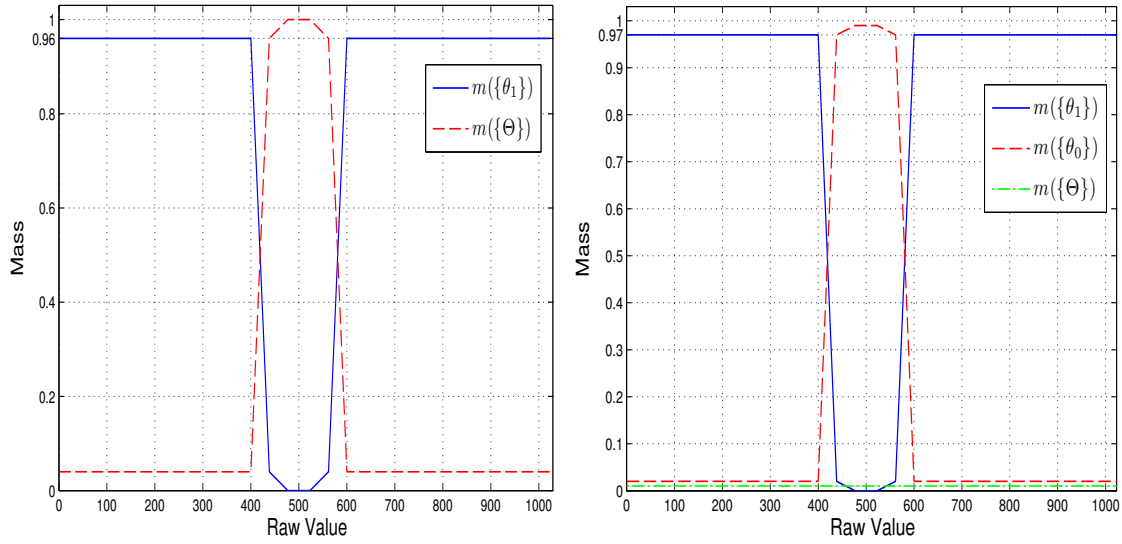


(a) First formulation. (b) Second formulation.

Figure 6.8: Example of a set of mass functions associated to a PIR Phidgets 1111_0 sensor in an indoor environment.

We notice that in the outdoor environment, the PIR Phidgets 1111_0 sensor provides a slightly better performance than the indoor environment case. Fig. 6.9 gives an example of

set of mass functions associated to the PIR Phidgets sensor in an outdoor environment.



(a) First formulation.

(b) Second formulation.

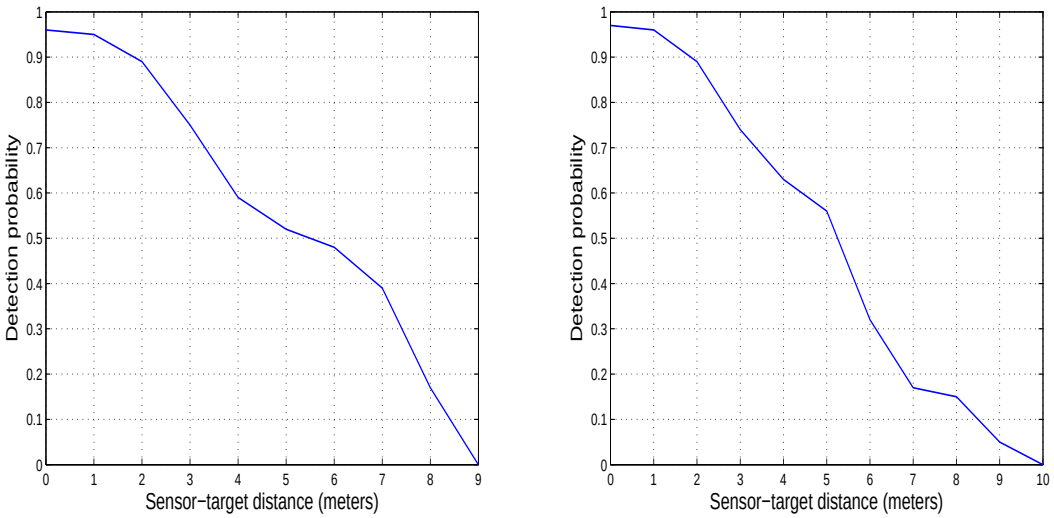
Figure 6.9: Example of a set of mass functions associated to a PIR Phidgets 1111_0 sensor in an outdoor environment.

6.5.2 Sensing Performance of the Phidgets Sensor

Fig. 6.10 shows the average sensing performance of the Phidgets sensor in both indoor and outdoor environments. As shown in Fig. 6.10, there is no clear cut-off boundary between successful and unsuccessful detection which illustrates the inaccuracy of the binary sensor coverage model. This has been pointed out by our previous empirical study discussed in Chapter 4.

6.6 Connectivity Model Analysis

In this section, we evaluate through measurements, using the XBee serie 1 (IEEE 802.15.4 radio), the PRR (Packet Reception Ratio) and the RSSI (Received Signal Strength Indicator) in both indoor and outdoor environments in order to examine the basic characteristics of the IEEE 802.15.4 communication channel. This study will allow us to correctly calibrate the communication cost model which is an input to our deployment framework.



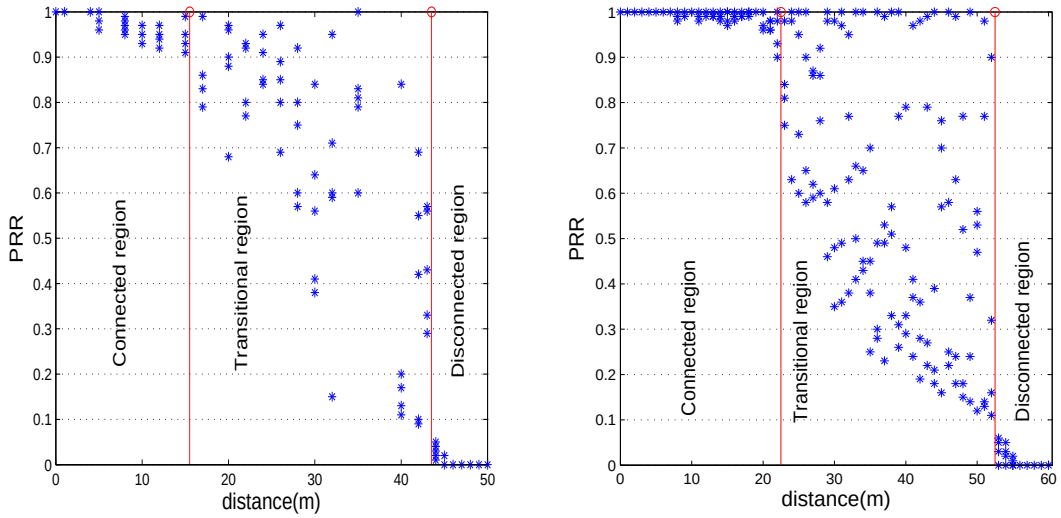
(a) Indoor environment. (b) Outdoor environment.

Figure 6.10: Detection probability *vs.* the sensor-target distance.

6.6.1 Regions of Connectivity

Since the distance and the direction are fundamental factors that affect the link quality, in this experiment, we place the transmitter at different distances and directions from the receiver, and compute the PRR. We set the packet size to 11 bytes (in the testbed, the size of packets sent by sensors is 7 bytes). The obtained results are reported on Fig. 6.11. Note that the reported PRR is the average over different PRR samples.

This result confirms that there is no clear cut-off boundary between successful and unsuccessful communication which illustrates the inaccuracy of the binary communication model as discussed in the previous Chapter. This has been also pointed out by previous empirical studies such as [25, 78]. Further, we see clearly three distinct reception regions: connected, transitional, and disconnected. This has been revealed by many recent studies such as [83]. The transitional region is characterized by high variances in the reception rates [25], which means that the links have moderate connectivity. We notice that in the outdoor environment, the XBee serie 1 provides a slightly better performance than the indoor environment case. The main purpose of our communication cost model is to wisely exploit the transitional region to enhance the network connectivity and lifetime.



(a) Indoor environment.

(b) Outdoor environment.

Figure 6.11: Regions of Connectivity.

6.6.2 Calibrating the Communication Cost Model

The first step in validating the proposed communication cost model is to verify the log-normal path loss model in the lab environment. Fig. 6.12 shows the average measured P_r and the theoretical P_r described by Equation (5.10) with $\gamma = 3.2$, $\sigma_\epsilon = 4$, $PL(d_0) = 40dB$, $d_0 = 1$ and an output power $P_t = 0dBm$. One can see that experimental data match well the log-normal path loss model.

6.7 EBDA *vs.* MIN_MISS

We are interested in the ability of ArduiNet to detect walking persons, as demanded by the application. In individual tests, we deploy ArduiNet as suggested by the output of the two placement algorithms (EBDA and MIN_MISS), we let a single person walk by at different paths in the lab room, and measure the detection rate. The experiments were repeated more than thousand times. The obtained results are summarized in Table 6.1.

The results of these further experimental tests are very convincing. We see clearly that fewer sensors are required when EBDA is used. EBDA consistently outperforms MIN_MISS in all cases. In fact, while both strategies guarantee the minimum required detection probability, our evidence-based approach saves almost 25% of the deployment

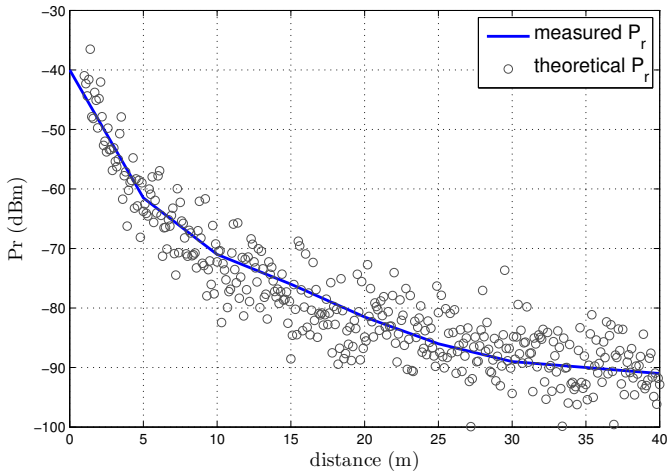


Figure 6.12: Measured P_r *vs.* theoretical P_r

Table 6.1: Experimental Results

	Requested detection rate	**Achieved detection rate**	**Deployment cost**
EBDA	90%	91%	6
	95%	97%	8
MIN_MISS	90%	94%	8
	95%	98%	11

cost over the probabilistic one. As mentioned before, the rate of reduction in the deployment cost achieved by the evidence-based approach is related among others to the size of the RoI and the minimum required detection probability. For instance, in a $2.4m \times 4m$ RoI, our evidence-based approach saves 25% (resp. 27%) of the deployment cost over the probabilistic one for a 0.9 (resp. 0.95) minimum required detection probability. Also, the larger the RoI, the greater is the performance gain. In conclusion, the present results confirm so far our previous results.

6.8 GEBDA

In these experiments, we deploy ArduiNet as suggested by the output of GEBDA. We let a single person walk by at different paths in the testbed room, and measure the rate of successful detection and the rate of false alarm. The experiments were repeated more than three thousand times. The obtained results are summarized in Table 6.2.

Table 6.2: Experimental Results

Requested (α,β)-coverage	**Deployment cost**	**Achieved (α,β)-coverage**
(0.90, 0.01)	14	(0.94, 0.00)
(0.95, 0.01)	16	(0.97, 0.00)

The results of these further experimental tests confirm so far our previous results. The proposed approach was able to meet the desired user detection requirements. Further, a zero false positive rate was achieved. These experimental results confirm that evidence combination improves significantly sensing coverage by wisely exploiting the collaboration among sensors.

6.9 Conclusion

In this Chapter, the effectiveness and efficiency of the proposed approaches were evaluated using experiments. We have developed, *ArduiNet*, a real WSN testbed for motion detection comprising more than 40 static and mobile sensor nodes. Obtained results were reported, supported by comparison with other works. The results of these further experimental tests confirm so far our previous results.

Chapter 7

General Conclusion and Perspectives

Sensor deployment is a mandatory and critical step in the process of developing WSNs solutions for real-life applications. Even if many optimizations can be done once the WSN is deployed to enhance its performance, we believe that the most important optimizations are those done at the design step to build the best topology possible that meets specific performance requirements. In fact, the types, numbers, and locations of sensors have to be judiciously planned so that performance requirements such as sensing coverage quality, network connectivity, lifetime, and reliability are all met while keeping the cost affordable. However, deploying WSNs into the real-world could be a very challenging task. In addition to the intrinsic properties of WSNs, many challenges stem from the close interactions between the WSN and the physical environment. Such challenges include imperfections related to sensors readings, sensors reliability, and harsh deployment environments to name a few. By considering such fundamental challenges in early design steps of the WSN, several issues can be expected thus avoiding unpleasant surprises and saving effort, time and money.

In this dissertation, we propose deployment frameworks under which deployment-related issues are incorporated into the deployment models and the problem formulations. Our proposals are based on the Dempster-Shafer theory that captures several characteristics of real-world applications and allows an efficient management of the challenges above. We also propose a series of methods and tools to address a set of deployment problems. In the following Sections, we summarize our main contributions and discuss on-going and future works.

7.1 Summary of Contributions

The work presented in the preceding Chapters has concentrated on investigating new ideas and techniques for the deployment of WSNs, particularly the issue of uncertainty-aware WSNs deployment. Hereafter, we will summarize our main contributions.

The first contribution is a survey of WSNs deployment strategies. We discussed the three deployment approaches, namely, random deployment, deterministic deployment, and self-deployment. For each approach, we presented a survey and taxonomy that captures the fundamental differences among existing solutions. We further performed qualitative comparisons between the different approaches. Therefore, providing not only a complete view of the state-of-the-art, but also useful insights for selecting the deployment strategy most appropriate to the application at hand.

In the second contribution, an empirical study has been carried out yielding a detailed analysis of intrinsic properties of random node placement in WSNs and provides a holistic view of random deployment properties. The obtained results give helpful design guidelines in using random deployment strategies. Based on our findings, we further proposed a hybrid model and we showed that such hybridization ensures better performance.

The third contribution involves uncertainty-aware WSNs deployment where sensors may not always provide reliable information. We investigated the evidence theory to design better deployment strategies. Specifically, we defined new evidence-based sensor coverage models and we devised five polynomial-time uncertainty-aware deployment algorithms. The first algorithm, *EBDA*, is a greedy heuristic that considers only uniform coverage and cost objectives. Afterward, we have considered two additional constraints, namely preferential coverage and network connectivity. On the basis of a dynamic programming approach, four polynomial-time uncertainty-aware algorithms, *E2BDA-S*, *E2BDA-C*, *E2BDA-P*, and *E2BDA* were proposed. Experimental results were provided to demonstrate the ability of our approach to achieve an efficient sensor deployment.

In the fourth contribution, we devised probabilistic connectivity and communication cost models, and we addressed the fusion-based WSNs deployment problem, taking into the account a rich set of objectives and constraints, namely, cost of deployment, (α,β)-coverage, network connectivity, data fidelity, sensors reliability, harsh deployment environments, and network lifetime. This problem was formulated as a binary non-linear and non-convex optimization problem, and two efficient heuristics, *GEBDA* and *MGEBDA*, using genetic algorithms were investigated. Experiments carried out on both synthetic data set

and data traces collected in a real experiment for vehicle detection show that our proposal outperforms all the existing strategies.

The fifth contribution considers the deployment of mobile sensors. Specifically, we devised a lightweight and comprehensive two-phase protocol, *HEAL*, for ensuring area coverage employing a mobile WSN.

In the sixth contribution, we discussed post-deployment approaches that can be exploited to enhance the performances of the WSN once is deployed. In particular, we proposed a new technique aimed at providing a good spatial-temporal distribution of the WSN lifetime. The proposed technique effectively balances the energy consumption of the entire WSN, and avoids the energy hole problem.

In our final contribution, we deployed, *ArduiNet*, a real WSN testbed for motion detection comprising more than 40 sensor nodes. The ArduiNet was designed, developed and implemented in the LiSSi laboratory [1].

7.2 On-going Works

7.2.1 Dynamic WSNs Deployment

All of the approaches discussed above initially compute the optimal location for the sensors and do not consider moving them once they have been positioned. In this work, we focus on designing algorithms to relocate sensors in order to enhance the quality of monitoring.

Several anomalies can occur in WSNs that impair their desired functionalities resulting in the formation of different kinds of holes, namely: coverage holes, routing holes, jamming holes, and worm holes [7]. In this work, we are interested in large bounded holes, i.e., large holes that are circumscribed by sensor nodes. Specifically, we seek to address the problem of hole detection and healing. Most of the existing solutions [94] use global operations to calculate the size of a big hole and then relocate a group of mobile sensors to heal the hole. While some existing localized solutions [94] requires strong assumptions or even unrealistic ones. The incompleteness of previous works motivates our research.

We propose a comprehensive solution, called *HEAL* (Holes dEtection and heALing) that has a very low complexity and avoid some drawbacks noticed in previous works. HEAL is a distributed and localized algorithm that operates in two distinct phases. The

1. http://lissi.fr/lab/

first phase consists of three sub-tasks; hole identification, hole discovery and border detection. Unlike prior efforts, we propose a Distributed and localized Hole Detection algorithm (*DHD*) that operates over the Gabriel Graph of the network. The computation complexity of DHD is $O(v^2)$, where v is the average number of 1-hop neighbors. Compared to the existing schemes, DHD has a very low complexity and deals with holes of various forms and sizes despite the nodes distribution and density. The second phase treats the hole healing with novel concept, hole healing area (Fig. 7.1). It consists of two sub-tasks; hole healing area determination and node relocation. We propose a distributed virtual forces-based local healing approach based on the Hole Healing Area (HHA), in which the forces will be effective. This allows a local healing where only the nodes located at an appropriate distance from the hole will be involved in the healing process.

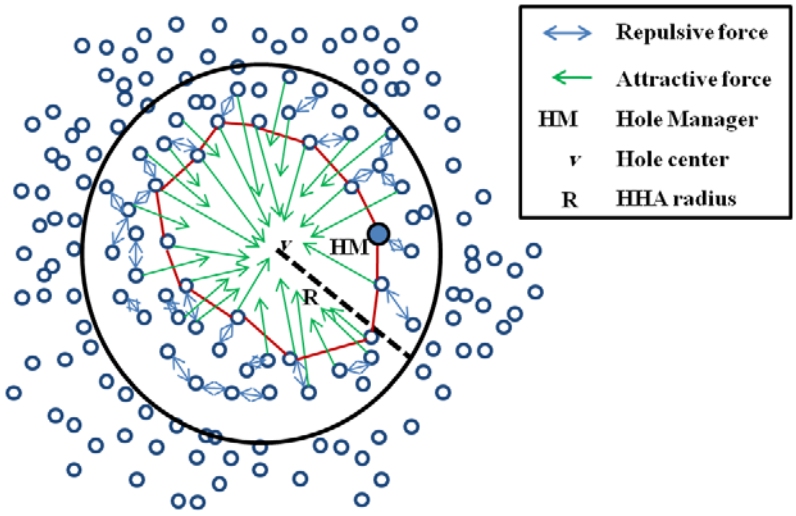


Figure 7.1: Principle of the healing process.

Through the performance evaluation, we validated HEAL, using different criteria and showed that it detects and heals the holes despite their number or size with less mobility in various situations. The evaluation results demonstrate that HEAL provides a cost-effective and an accurate solution for hole detection and healing in mobile WSNs.

We are currently working on holes detection while considering our evidence-based coverage models. In the future, we plan to investigate the interaction between HEAL and the network layer for on-demand hole detection and healing.

7.2.2 Post-deployment Strategies

Effectively managing WSNs is a major challenge starting from deployment phase and during the entire lifetime of the network. In the approaches discussed previously, we strive to build the best *physical topology* possible that meets specific performance requirements. Recent research has focused on methods to improve the initial physical configuration. One possible method is to use mobile sensors as discussed in our previous on-going work. Another method is to focus on the *logical topology* of the WSN which is formed by the communication graph of the network. Choosing the right logical topology helps to reduces the radio interference, the probability of message being lost, and the waiting time for sensors to transmit data. Also, it facilitates data aggregation, which greatly reduces the amount of processing cycles and energy consumption, resulting in a longer network lifetime.

The network lifetime strongly depends on the routing protocol used and can be defined in several ways. For instance, the lifetime can be defined as the time elapsed until the first node (or the last node) in the network depletes its energy (dies) [144]. In some scenarios such as intrusion detection, it is necessary that all nodes stay alive as long as possible, since service quality decreases considerably as soon as one node dies. In these scenarios, it is important to know when the first node dies. The *FND* metric (First Node Dies) gives an estimated value for this event. In some applications, the loss of a single or few nodes does not automatically reduce the service quality. The *HNA* metric (Half of the Nodes Alive) denotes an estimated value for the half-life period of a WSN whereas the *LND* metric (Last Node Dies) gives an estimated value for the overall lifetime of a WSN. Using these three metrics (FND, HNA, and LND), the network lifetime distribution over time can be analyzed. Moreover, sensor nodes are placed at different distances of the sink so the network lifetime distribution over space can be analyzed also.

In this work, the two aspects are considered: the network lifetime distribution over time and over space, or, in other words, the spatial-temporal (ST) distribution of the network lifetime. We propose a new technique aimed at providing a good ST distribution of the WSN lifetime [95]. Obtained results show that the proposed technique effectively balances the energy consumption of the entire WSN.

The neighboring nodes which may exhaust their residual energies are called energy hole. The ST distribution of the network lifetime can be very useful in predicting and mitigating the formation of energy-hole in time-driven many-to-one WSNs. This study can also be useful in the design of node distribution strategies used to mitigate the energy hole problem in WSNs.

7.3 Future Research Directions

This dissertation addressed the challenging and fundamental issues related to the deployment of WSNs. Several future research directions and open issues can be derived from our work so far. We first discuss extensions to our work discussed in the preceding Chapters.

1. Throughout this dissertation, we assumed that sensors have a common coverage and connectivity models. We can extend our work to the case of heterogeneous sensors with different coverage and/or connectivity models.
2. In Chapter 3, we examined the intrinsic properties of random node placement in WSNs when disk sensor coverage model and binary connectivity are assumed. Future work could investigate these properties under other coverage and connectivity models, such as the evidence-based coverage model and the probabilistic connectivity model.
3. In Chapter 4, we have studied deployment for a basic terrain model. We can extend our work to more realistic terrain models under which the various proposed deployment algorithms can be evaluated.
4. There are more theoretical works to be done. One of the future works is to investigate the relationship between connectivity and coverage while considering the evidence-based coverage model and the probabilistic connectivity model. Another future work is to analyze the optimal fusion radius discussed in Chapter 5.
5. More experiments with ArduiNet are to be done to quantify the real benefit of the proposed deployment approaches. For instance, the validation of the self-deployment approach proposed in the paper could be a valuable experience.

In addition to the extensions discussed above, we can investigate the deployment problems under new dimensions. A subset of possible next steps are discussed below:

1. There is an unquestionable need for sensor coverage models which are close enough to reality as to represent networks performance, being at the same time concise enough to promote strong theoretical results. We expect realistic coverage models to be a promising research direction.
2. All our current works are limited to two-dimensional terrain. A natural idea is to investigate the possible solutions for these problems in the three-dimensional case.

3. Other WSNs deployment formulations such as activity scheduling, point coverage and barrier coverage could be investigated while considering our evidence-based coverage models.
4. Our current works only consider WSNs, one direction of future works is to extend them to WSANs.

Chapitre 8

Version Française Abrégée

8.1 Introduction

Un Réseau de Capteurs sans Fil (RCsF), ou Wireless Sensor Network (WSN), est un ensemble d'entités déployées de façon à couvrir un territoire donné. Ils sont capables d'opérer en toute autonomie afin de collecter, traiter et envoyer des données relatives à leur environnement vers une station de base (Fig. 8.1). Les noeuds capteurs disposent de faibles capacités énergétiques et communiquent entre eux via des liaisons sans fil [13].

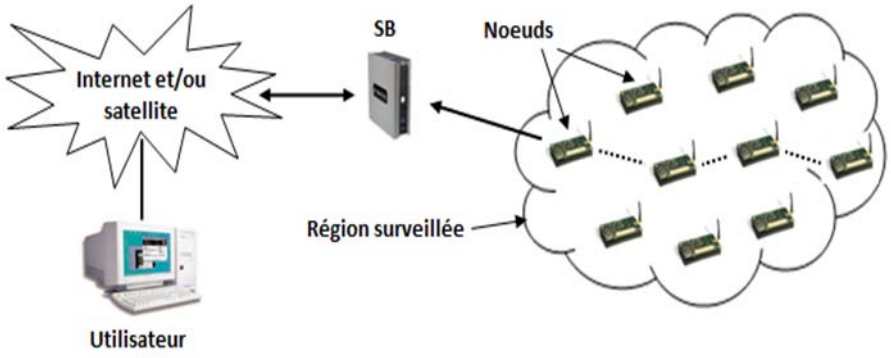
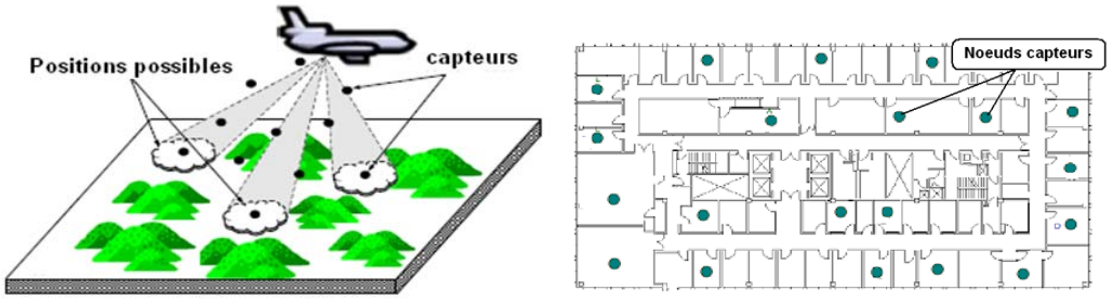


FIGURE 8.1 – Schéma général d'un RCsF.

Les RCsF sont utilisés dans de nombreuses applications militaires et civiles. Ils peuvent être intégrés dans le commandement militaire, le contrôle, les communications, le renseignement, la surveillance, et les systèmes de reconnaissance et détection de cibles. Le déploiement rapide, l'auto-organisation et la tolérance aux fautes, principales caractéristiques des réseaux de capteurs, en font une technique de détection très prometteuse.

Les capteurs peuvent être placés dans la zone à surveiller de trois manières : aléatoire (Fig. 8.2(a)), déterministe (Fig. 8.2(b)), ou dynamique (l'auto-déploiement) [12, 120, 121, 143]. Le choix de la méthodologie de déploiement dépend fortement du type des capteurs, de l'application envisagée et de l'environnement de déploiement. Le déploiement déterministe des capteurs est viable et est souvent nécessaire lorsque les capteurs sont coûteux ou lorsque leur exploitation est sensiblement affectée par leurs positions. De plus, dans certaines applications, la répartition aléatoire des capteurs est la seule option possible. Ceci est particulièrement vrai pour les environnements difficiles et hostiles tels qu'un champ de bataille ou une zone sinistrée. Dans le cas des capteurs mobiles, le déploiement dynamique exploite cette mobilité pour améliorer les performances du réseau.



(a) Illustration d'un déploiement aléatoire.

(b) Illustration d'un déploiement déterministe des capteurs dans un immeuble.

FIGURE 8.2 – Déploiement aléatoire *vs.* déploiement déterministe.

Le déploiement des RCsF est un sujet de recherche qui a suscité beaucoup d'attention ces dernières années. En effet, le nombre et les positions des capteurs, déployés dans une zone d'intérêt, déterminent la topologie du réseau. Ceci influence sur les propriétés intrinsèques du réseau, comme : la couverture, la connectivité, le coût et la durée de vie. Par conséquent, les performances d'un RCsF dépendent en grande partie de son déploiement. Cette thèse porte sur les problèmes de déploiement des RCsF. Elle suit trois directions principales : (1) le déploiement aléatoire, (2) le déploiement déterministe, et (3) l'auto-déploiement. Dans les sections suivantes, nous décrivons brièvement les principaux travaux menés dans le cadre de cette thèse.

8.2 Etude des propriétés intrinsèques du déploiement aléatoire des RCsF

Dans un grand nombre d'applications des RCsF, notamment les applications de surveillance en extérieur, le positionnement direct des capteurs semble fastidieux, difficile, voir parfois impossible. Cette difficulté est liée essentiellement au nombre important des capteurs et à la nature de la zone de déploiement. Dans le cas d'un terrain caractérisé par des reliefs difficiles ou potentiellement dangereux, il est difficile de procéder à ce type de déploiement. La seule option possible, dans ces cas, est le déploiement aléatoire des capteurs. Celui-ci peut se faire de différentes manières : largage par un avion (ou un drone), lancement par artillerie, etc. Par ailleurs, la nature de la distribution des capteurs sur le terrain change en fonction des cas.

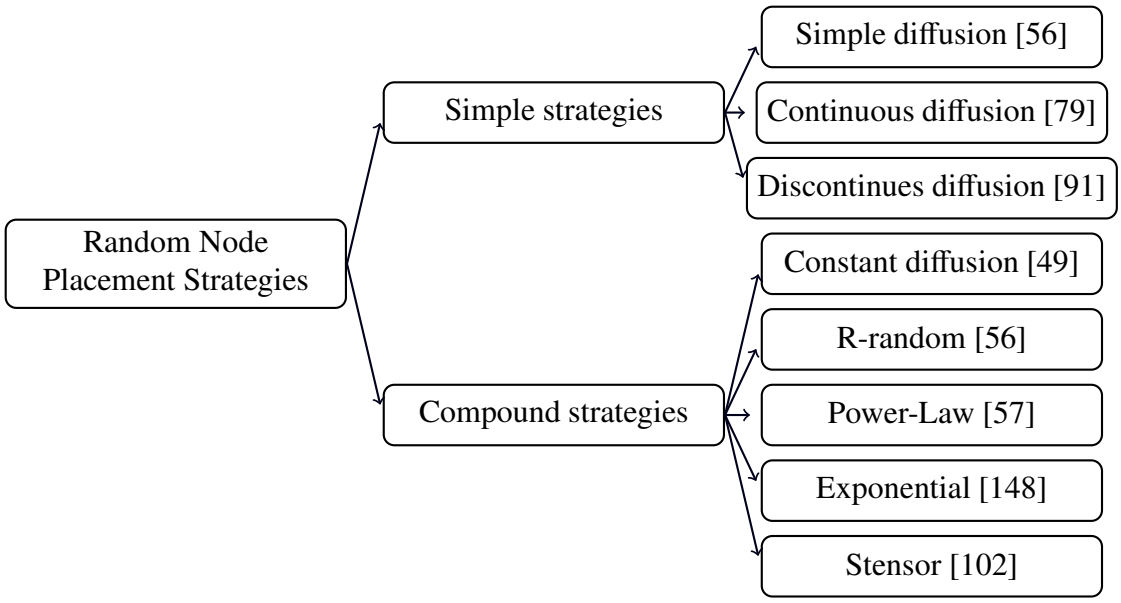


FIGURE 8.3 – Taxonomie des stratégies de déploiement aléatoire des RCsF.

Dans ce travail, nous avons étudié les propriétés intrinsèques des modèles de déploiement stochastique des RCsF, tels que la couverture, la connectivité, la tolérance aux fautes, et la durée de vie. Nous avons présenté un état de l'art sur les modèles de déploiement stochastique et nous avons proposé une classification de ces modèles (Fig. 8.3). Nous avons ensuite proposé des métriques, que nous jugeons importantes pour l'évaluation des modèles de déploiement étudiés, auxquelles nous avons adjoint un ensemble d'algorithmes permettant de les calculer. En mesurant plusieurs métriques, nous avons évalué les performances

d'un réseau de capteurs déployé suivant plusieurs modèles de déploiement stochastique. L'évaluation des modèles, basée sur des simulations, prend en compte aussi bien les métriques de fonctionnement que les métriques de déploiement. Les résultats obtenus aboutissent à des directives de conception utiles pour l'utilisation des modèles de déploiement stochastique.

8.3 Proposition d'une stratégie pratique de déploiement aléatoire

Les résultats obtenus dans la première partie ont permis de proposer une hybridation entre deux modèles étudiés améliorant ainsi les performances du réseau. En effet, d'une part, le modèle uniforme garantit un taux de couverture et de connectivité important lors du déploiement du réseau. Après un certain temps de fonctionnement du réseau, ces performances atteignent un niveau inférieur par rapport aux autres modèles tels que la diffusion simple ou R-random. Le modèle uniforme atteint rapidement le régime de saturation, la station de base située au centre de la zone d'intérêt s'isole rapidement et aucun message ne pourra être lui transmis. D'autre part, les deux modèles R-random et la diffusion simple assurent une grande densité au centre de la zone d'intérêt, ce qui prolonge la durée de vie du réseau en termes de taux de détection.

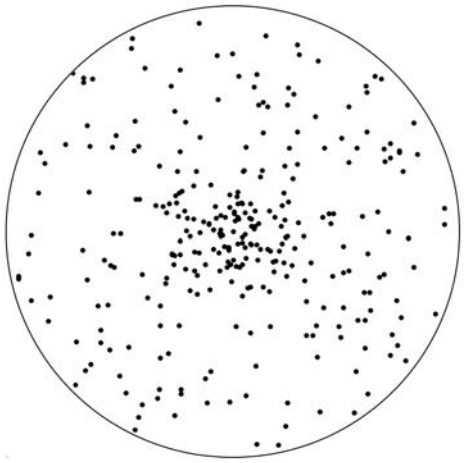


FIGURE 8.4 – Un exemple de la diffusion hybride de 300 capteurs dans une zone circulaire de rayon $R = 150m$.

Nous avons proposé une hybridation entre le modèle de diffusion simple qui génère

un grand nombre de capteurs autour de la station de base et le modèle uniforme qui assure une plus grande couverture du champ de surveillance (Fig. 8.4). Nous avons choisi la diffusion simple par rapport à la distribution R-random alors que cette dernière assure un plus grand taux de détection, car la diffusion simple est caractérisée par l'écart-type σ qui permet de contrôler la densité des capteurs autour de la station de base. Nous avons validé cette proposition par une multitude de comparaisons avec les modèles de déploiement déjà étudiés.

8.4 Modèles de couverture basés sur les évidences

Un défi clé dans la réussite d'un déploiement de RCsF consiste à considérer les imperfections liées aux lectures opérées par les capteurs. Dans ce travail, l'objectif est de proposer des mécanismes de déploiement déterministe des RCsF avec une prise en charge des imperfections liées aux lectures des capteurs. Pour prendre en compte ces imperfections, des formalismes mathématiques ont été proposés par le passé tel que la théorie des sous-ensembles flous [146] lorsqu'il s'agit d'informations imprécises, ou la théorie des probabilités [65], des possibilités [147], et des fonctions de croyance [98] dans le cas de données incertaines.

La théorie des fonctions de croyance (ou théorie de Dempster-Shafer) a été initialement proposée par Dempster et Shafer, puis largement développée par P. Smets grâce à son modèle des croyances transférables [106]. Le principal avantage de cette théorie est sa flexibilité quant à la représentation de l'information. Elle permet à la fois de représenter et de manipuler des informations aussi bien incertaines qu'imprécises. En séparant ces deux sources d'inexactitude dans la représentation de l'information, cette théorie permet de mieux formaliser certains problèmes, et propose également, une modélisation assez simple du doute et de l'absence d'information. Ces avantages semblent pertinents à exploiter dans le contexte du déploiement des RCsF où les données sont souvent de nature incertaine.

Dans un premier temps, notre travail a consisté à modéliser la couverture d'un capteur dans le cadre du modèle des croyances transférables, de proposer par la suite un modèle de couverture d'un RCsF basé sur les croyances et de concevoir une approche de déploiement qui exploite les deux modèles (couverture de capteur/réseau).

Un modèle de couverture mesure la capacité et la qualité de détection en considérant la relation géométrique entre un point de l'espace et le capteur [120]. Les modèles de couverture existants, comme le modèle binaire ou le modèle probabiliste [8, 9, 38, 39, 149, 152]

peuvent ne pas être réalistes dans de nombreux cas et restent limités. Sur la base du modèle des croyances transférables, nous avons développé, plusieurs modèles de couverture capteur/réseau. Ces modèles représentent fidèlement la réalité et peuvent par ailleurs être facilement étendus pour inclure d'autres facteurs liés au déploiement, tels que la fiabilité des capteurs ou encore les environnements difficiles. Les résultats obtenus démontrent la supériorité de ces modèles par rapport aux modèles classiques.

8.5 Plateformes prenant en compte l'incertitude pour le déploiement des RCsF

Une application typique des RCsF est la détection de cibles/évènements. Les capteurs collaborent pour une prise de décision par consensus quant à la présence ou non d'un(e) cible/évènement dans la région surveillée. Dans le cadre de nos travaux, nous avons développé une approche basée sur la théorie des fonctions de croyance. L'observation du système est assurée par des capteurs qui fournissent une opinion pondérée, représentée par une fonction de croyance, concernant l'état réel du système. Dans ce cas, les capteurs jouent le rôle de sources de croyance. Ils émettent une opinion sur l'état du système observé (présence de cible/évènement). Un des capteurs appelé centre de fusion (ça peut être un cluster-head, la station de base ou tout autre capteur), collecte les croyances (données sensorielles) des autres capteurs, combine ces évidences, effectue une transformation pignistique et décide si un(e) cible/évènement est présent(e) ou non dans la zone surveillée. La figure 8.5 illustre un tel scénario.

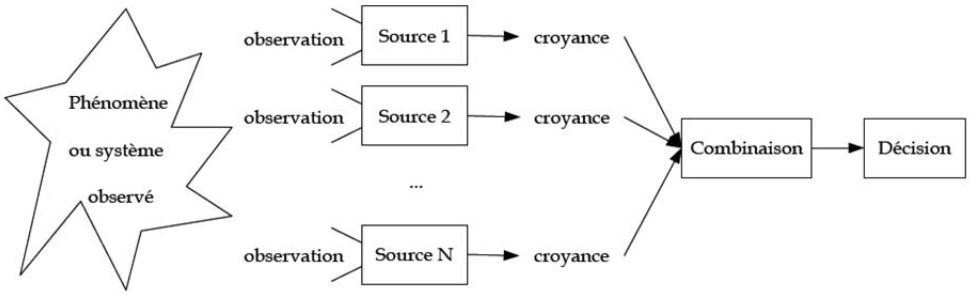


FIGURE 8.5 – Détection de cible/évènement dans le cadre de la théorie des fonctions de croyance.

Nos principales contributions dans le cadre de ce travail peuvent se résumer ainsi :

1. Nous avons formulé six variantes du problème de déploiement déterministe des réseaux de capteurs statiques.
2. En exploitant les règles d'affaiblissement/renforcement des fonctions de croyance, nous avons montré comment intégrer la fiabilité des capteurs et les environnements difficiles dans le processus de déploiement.
3. En utilisant nos modèles de couverture basés sur les croyances, nous avons conçu plusieurs algorithmes de déploiement qui permettent de déterminer le nombre minimum de capteurs ainsi que leur emplacement, afin d'assurer la couverture uniforme d'une zone. Nous avons aussi considéré d'autres objectifs, comme la couverture préférentielle, la connectivité, et la durée de vie du réseau.

8.6 Déploiement autonome des capteurs dans un RCsF mobile

Une bonne partie des travaux existants dans le domaine des RCsF se sont focalisés sur le développement d'algorithmes de traitement des signaux collectés par les capteurs, ou encore sur la minimisation de l'énergie consommée par ces derniers. Ces algorithmes supposent que les capteurs sont statiques et que le nombre de capteurs déployés est supposé être suffisamment grand pour couvrir toute la zone surveillée. Toutefois, un déploiement statique des capteurs ne mène pas certainement à une bonne couverture de la zone surveillée.

Dans une stratégie de déploiement statique, la décision sur les emplacements des capteurs est faite au moment de la configuration du réseau et ne tient pas compte des changements dynamiques dans sa phase d'exploitation. En effet, plusieurs situations peuvent se produire : les routes du trafic peuvent changer par cause de la disparition d'un ou de plusieurs capteurs, la charge peut ne pas être répartie équitablement entre tous les capteurs, provoquant ainsi des goulots d'étranglement, l'usage d'une application peut varier dans le temps, et enfin les ressources réseau disponibles peuvent changer à mesure que de nouveaux capteurs rejoignent le réseau. Ces situations montrent que le déploiement statique ne répond pas à une bonne partie des applications des RCsF. Par conséquent, le repositionnement dynamiquement des capteurs, dans un environnement opérationnel, est nécessaire afin d'améliorer les performances du réseau. Par exemple, lorsque de nombreux capteurs dans les environs de la station de base fonctionnelle tombent en panne dû à l'épuisement de

leurs batteries, des capteurs redondants dans d'autres parties de la région surveillée peuvent être identifiés et redéployés pour remplacer les capteurs éteints, afin d'améliorer la durée de vie du réseau et assurer ainsi la continuité du service.

Un déploiement dynamique peut également être très bénéfique dans un objectif de suivi où la cible est mobile, des capteurs peuvent être rapprochés de l'objectif afin d'accroître l'acuité des données transmises par les capteurs. Aussi, dans certaines applications, la station de base doit être maintenue à une distance de sécurité des objectifs nuisibles en l'a déplaçant par exemple vers des zones plus sûres afin d'assurer son bon fonctionnement.

Le mouvement des capteurs pendant le fonctionnement régulier du réseau impose l'existence de règles réactives. Contrairement au placement initial, la relocalisation est exercée en réponse à un stimulus du réseau ou de l'environnement de surveillance, ce qui fait qu'il requiert une surveillance continue de l'état du réseau et de ses performances, ainsi que l'analyse des événements qui se déroulent à proximité du capteur pendant son déplacement. En outre, le processus de relocalisation doit se manipuler avec précaution, cela peut en effet potentiellement causer des perturbations dans le routage des données ou dans la détection de nouveaux événements ayant lieu lors du redéploiement.

Dans le cadre de cette thèse, nous avons présenté un état de l'art sur les modèles de déploiement dynamique. Nous avons proposé une classification des différentes approches proposées dans la littérature, et nous avons mené une étude comparative entre ces approches. Un nouveau mécanisme de détection et de réparation des vides dans les réseaux de capteurs mobiles fait l'objet d'une de nos contributions. Ce dernier est basé sur le concept des forces virtuelles, il permet une réparation locale des vides avec le minimum des délais et un coût réduit.

8.7 Les approches de post-déploiement

Dans cette partie, nous discutons des approches post-déploiement qui peuvent être exploitées pour améliorer les performances du RCsF une fois la phase de déploiement effectuée.

Plus précisément, nous proposons une nouvelle technique visant à fournir une bonne répartition spatio-temporelle de la durée de vie du RCsF. La technique proposée équilibre efficacement la consommation énergétique de l'ensemble du RCsF, et permet d'éviter le problème des vides d'énergie. Les résultats obtenus montrent l'efficacité de l'approche proposée dans le but de prolongation de la durée de vie des RCsF.

8.8 Conclusion

Les travaux développés dans le cadre de cette thèse s'intéressent à un problème fondamental dans les RCsF, à savoir : le déploiement. Vu la diversité des facteurs impliqués, tels que le type des capteurs, l'application envisagée, et l'environnement de déploiement, ce problème peut être étudié sous des angles différents, comme le montre la figure 8.6.

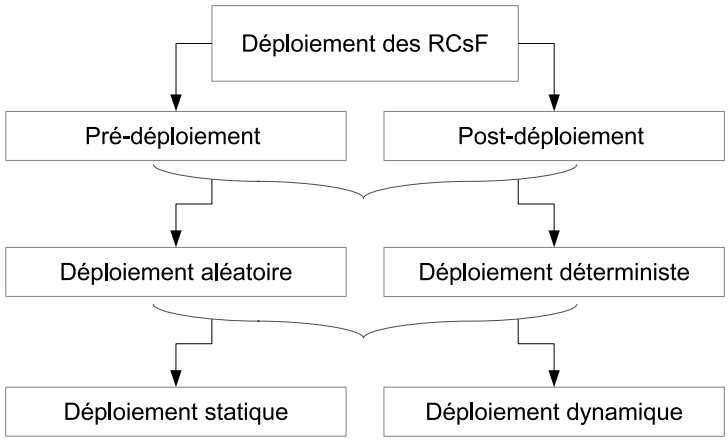


FIGURE 8.6 – Différents angles d'études pour le problème du déploiement des RCsF.

Dans le cadre de cette thèse, ces différents angles d'études ont été explorés. Afin de concevoir des stratégies de déploiement plus efficaces, nous avons axé nos travaux sur l'usage de la théorie de l'évidence. Les solutions proposées ont été validées et comparées aux approches existantes par des expérimentations menées à la fois sur des données synthétiques et aussi sur un testbed expérimental.

Appendix A

The Belief Functions Theory Overview

A.1 Introduction

In this Appendix, we cover some background information related to the belief functions theory. We focus our discussions on the Transferable Belief Model (TBM), a particular interpretation of the belief functions theory.

A.2 Overview of the TBM

The TBM, introduced by Smets [106], is a particular interpretation of the so-called Dempster-Shafer theory [136]. This uncertain reasoning framework can be seen as an extension of the Bayesian probability theory.

The TBM is a two-level model in which knowledge representation is non-Bayesian while decision making is Bayesian. At the *credal level*, beliefs are represented and updated in a non-Bayesian fashion; at the *pignistic level* (from the Latin pignus, the bet), beliefs are casted into a structure that obeys the axioms of probability theory. The key propositions of this theory are summarized below. Further details can be found in [104, 105, 106].

A.2.1 Representation of Information

Frame of Discernment

In a TBM-based reasoning system, the frame of discernment is a fundamental concept. The evidence theory represents a problem domain by a set called the Frame of Discernment

(FoD) Θ which is the set of possible states of the system.

$$\Theta = \{\theta_1 \dots \theta_k \dots \theta_N\} = \bigcup_{k=1}^{N} \{\theta_k\}$$

It must be exhaustive and mutually exclusive in the sense that the system is certainly in one, and only one, state $\theta_k \in \Theta$. The elements of the power set [1] 2^Θ are called *hypotheses*.

Basic Belief Assignment

The central element of the TBM is the basic belief assignment (*bba*). A *bba* is a function m (mass function) from the power set 2^Θ to $[0, 1]$ satisfying [2]:

$$\begin{aligned} m(\emptyset) &= 0 \\ m(A) &\geq 0, \quad \forall A \subseteq \Theta \\ \sum_{A:A\subseteq\Theta} m(A) &= 1 \end{aligned} \tag{A.1}$$

The basic belief mass (*bbm*) $m(A)$ is the part of belief that supports A and does not support any strict subset of A [106]. If $m(A) > 0$, the subset $A \subseteq \Theta$ is called a *focal set* of m. A *categorical bba* has only one focal set. The *vacuous belief function* satisfies $m(\Theta) = 1$. It represents a state of total ignorance regarding the question under consideration. A *simple bba* has at most two focal sets, including Θ. A *Bayesian bba* is a *bba* whose focal sets are singletons. A *bba* is said to be *consonant* if its focal sets are nested. Table A.1 shows some examples.

A *bbm* can be committed to either a singleton or a subset of Θ. This property makes TBM more expressive than probability theory. When a *bbm* is committed to a subset that has more than one element, it is explicitly stating that there is not enough information to distribute this belief more precisely among the individual elements in the subset. In particular, the total belief is assigned to Θ when there is no evidence about Θ at all. In contrast, probability theory lacks this ability by dividing the total belief equally among elements of Θ (Principle of Insufficient Reason).

1. The power set of any set S, written $\mathcal{P}(S)$ or 2^S, is the set of all subsets of S, including the empty set and S itself.

2. Under closed world assumption $m(\emptyset) = 0$. Under open world assumption, $m(\emptyset)$ may be positive. For the difference between the open and closed world assumptions, see Smets [103].

Table A.1: Examples of mass functions in the FoD $\Theta = \{\theta_1, \theta_2, \theta_3\}$

$A \subseteq \Theta$	Categorical	Bayesian	Consonant	Simple	Vacuous
$\emptyset$	0	0	0	0	0
$\{\theta_1\}$	0	0.3	0.6	0.7	0
$\{\theta_2\}$	0	0.5	0	0	0
$\{\theta_3\}$	1	0.2	0	0	0
$\{\theta_1, \theta_2\}$	0	0	0.3	0	0
$\{\theta_1, \theta_3\}$	0	0	0	0	0
$\{\theta_2, \theta_3\}$	0	0	0	0	0
$\{\theta_1, \theta_2, \theta_3\}$	0	0	0.1	0.3	1

Belief and Plausibility Functions

Two other evidential functions can be used to represent an agent's belief. The belief function (*bel*) represents the belief assigned to an event $A \subseteq \Theta$ given the available evidence. It is obtained by summing all the basic belief masses $m(B)$ for $B \subseteq A, B \neq \emptyset$. We have:

$$bel(A) = \sum_{B:\emptyset \neq B \subseteq A} m(B), \ \forall A \subseteq \Theta, A \neq \emptyset \tag{A.2}$$

where $bel(\emptyset) = 0$, and $bel(\Theta) = 1$. The belief function is also called the lower probability function. The dual of *bel*, called plausibility function *pl* or upper probability function is defined as:

$$pl(A) = \sum_{X \subseteq \Theta : X \cap A \neq \emptyset} m(X), \ \forall A \subseteq \Theta \tag{A.3}$$

The quantity *pl* is thus equal to the sum of the basic belief masses assigned to propositions that are not in contradiction with A. It corresponds to the maximum degree of support that could be given to A, if further evidence becomes available.

bel is often 0 on all or most of the atomic hypotheses in complex domains unless a large number of evidence sources are available. However, *pl* generally provides some discrimination even when the evidence is sparse. Therefore, *pl* is a more robust guide to decision making than is *bel* [20]. Other functions have been defined such as the *commonality function* Q which is generally used as a technical device to simplify proofs of computational theorems. Shafer [98] showed that a one-to-one correspondence exists between mass, belief, plausibility and commonality, meaning that given any one of m, *bel*, *pl* or Q, the other three can be calculated. Thus, a body of evidence in any of the four forms may be called a

belief function since *bel* is uniquely determined and can be recovered. Table A.2 shows an example of the connection between mass, belief, and plausibility functions.

Table A.2: The connection between mass, belief, and plausibility functions

$A \subseteq \Theta$	$m(A)$	$bel(A)$	$pl(A)$
$\emptyset$	0	0	0
θ_1	0.35	0.35	0.56
θ_2	0.25	0.25	0.45
θ_3	0.15	0.15	0.34
θ_1, θ_2	0.06	0.66	0.85
θ_1, θ_3	0.05	0.55	0.75
θ_2, θ_3	0.04	0.44	0.65
$\theta_1, \theta_2, \theta_3$	0.1	1.0	1.0

A.2.2 Handling Information

In addition to the different ways to represent bodies of evidence provided by the evidence theory, a set of methods and tools to handle these bodies of evidence is also provided.

Combination

When several pieces of evidence are obtained through distinct sources (agents) over the same FoD Θ, new evidence representing the consensus of those disparate opinions can be obtained through the combination operation. Many combination rules have been proposed [96]. Among them, we find:

- The conjunctive rule of combination, also referred to as the unnormalized Dempster's rule, which constructs the *bba* if it is accepted that sources are fully reliable;
- The disjunctive rule of combination, which constructs the *bba* if at least one source is accepted as fully reliable.

Let m_1 and m_2 be mass functions on Θ representing two pieces of evidence from independent sources. A and B represent focal elements of m_1 and m_2 respectively.

- The conjunctive combination (denoted by $m^{\Theta}_{1\textcircled{\cap}2}$) is calculated from the aggregation of two *bba*'s m_1 and m_2 in the following manner:

$$m^{\Theta}_{1\textcircled{\cap}2}(C) = \sum_{A\subseteq\Theta, B\subseteq\Theta: A\cap B=C} m_1(A)\cdot m_2(B) \quad \forall C \subseteq \Theta \tag{A.4}$$

– The disjunctive combination (denoted by $m^{\Theta}_{1\textcircled{\cup}2}$) is calculated from the aggregation of two *bba*'s m_1 and m_2 in the following manner:

$$m^{\Theta}_{1\textcircled{\cup}2}(C) = \sum_{A\subseteq\Theta, B\subseteq\Theta: A\cup B=C} m_1(A)\cdot m_2(B) \quad \forall C \subseteq \Theta \tag{A.5}$$

As mentioned before, in the situation where all sources are considered reliable, a conjunctive operation is appropriate (A and B and $C \ldots$). In the case where there is one reliable source among many, we can justify the use of a disjunctive combination operation (A or B or $C \ldots$). However, many combination operations lie between these two extremes (A and B or C, A and C or B, etc.). Dubois and Prade [42] describe these three types of combinations as conjunctive pooling ($A \cap B$, if $A \cap B \neq \emptyset$), disjunctive pooling ($A \cup B$), and tradeoff (there are many ways a tradeoff between $A \cap B$ and $A \cup B$ can be achieved). Examples of such combination operations include Yager's rule; Inagaki's unified combination rule; Zhang's center combination rule; and Dubois and Prade's disjunctive pooling rule [96]. It is worth noting that the *bbms'* from mass functions that contradict each other can be used to obtain a measure of conflict between the independent belief sources.

Discounting of Information

Sometimes, sources (agents) are vulnerable to misreading or malfunctioning due to their type and/or their deployment environment [43]. In this case, we consider such sources as only partially reliable. This source reliability can be taken into account by discounting their original *bba* by a discount rate α between 0 and 1. The lower the reliability is, the larger the discount rate will be.

For $\alpha \in [0, 1]$, let $(1 - \alpha)$ be the degree of "confidence" we assign to the source. It can be encoded into a *bba* defined on the set $\{reliable, not\ reliable\}$ such that:

$$m(x) = \begin{cases} 1-\alpha & \text{if } x = \{reliable\}, \\ \alpha & \text{if } x = \{not\ reliable\} \end{cases} \tag{A.6}$$

Suppose that the *bba m* on Θ represents the source belief about the actual value of Θ. The result of combining the source belief with the *bba* given in A.6 is a new *bba* denoted by m^{α}, defined as:

$$\begin{aligned} m^{\alpha}(A) = &\quad (1-\alpha)m(A) \quad for \ A \subset \Theta \\ m^{\alpha}(\Theta) = &\quad \alpha + (1-\alpha)m(\Theta) \end{aligned}$$

If a priori knowledge on the source reliability is available, discounting factors associated to the sources will be used in the evidence combination step; the evidence received will be discounted using the associated discounting factors.

A.2.3 Decision Making

The typical problem-solving scenario using the evidence theory is to gather bodies of evidence and represent them as belief functions, combine these belief functions with Dempster's rule (or other combination rules), and then select the hypothesis best supported by the combined evidence. *bel* and *pl* are the measures provided by the Dempster-Shafer theory to make the selection. Another measure provided by the transferable belief model is the Pignistic probability.

Pignistic Transformation

When a decision is to be made, a probability function must be constructed from the mass functions to make the optimal decision, i.e., the one that maximizes the expected utility. This transformation is called the pignistic transformation. It is defined by:

$$BetP(A) = \sum_{B \subseteq \Theta} \frac{|A \cap B|}{|B|} \frac{m(B)}{1 - m(\emptyset)}, \ \forall A \subseteq \Theta$$

The meaning and the justification of this transformation are detailed in Smets [105].

Bibliography

[1] A. Ababnah and B. Natarajan. Lqr formulation of sensor deployment for decision fusion based detection. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5, 2010.

[2] A. Ababnah and B. Natarajan. Optimal control-based strategy for sensor deployment. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 41(1):97–104, 2011.

[3] Ahmad Ababnah. *Sensor deployment in detection networks - a control theoretic approach*. PhD thesis, Kansas State University, 2010.

[4] Ahmad Ababnah and Balasubramaniam Natarajan. Optimal sensor deployment for value-fusion based detection. In *GLOBECOM'09*, pages 1–6, 2009.

[5] Ahmad Ababnah and Balasubramaniam Natarajan. Sensor deployment as an optimal control problem. In *Proceedings of 18th International Conference on Computer Communications and Networks*, ICCCN '09, pages 1–5, Washington, DC, USA, 2009.

[6] J. R. Agre, L. P. Clare, G. J. Pottie, and N. P. Romanov. Development platform for self-organizing wireless sensor networks. In *SPIE*, volume 3713, pages 257–268, 1999.

[7] Nadeem Ahmed, Salil S. Kanhere, and Sanjay Jha. The holes problem in wireless sensor networks: a survey. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(2):4–18, April 2005.

[8] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle. Differentiated underwater sensor network deployment. In *EEE/OES, OCEANS-07*, pages 1–6, 2007.

[9] N. Aitsaadi, N. Achir, K. Boussetta, and G. Pujolle. A tabu search wsn deployment method for monitoring geographically irregular distributed events. *Sensors*, 9:1625–1643, 2009.

[10] Nadjib Aitsaadi, Nadjib Achir, Khaled Boussetta, and Guy Pujolle. Artificial potential field approach in wsn deployment: Cost, qom, connectivity, and lifetime constraints. *Comput. Netw.*, 55(1):84–105, January 2011.

[11] K. Akkaya and M. Younis. C2ap: Coverage-aware and connectivity-constrained actor positioning in wireless sensor and actor networks. In *Performance, Computing, and Communications Conference, 2007. IPCCC 2007. IEEE Internationa*, pages 281–288, 2007.

[12] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, March 2002.

[14] Ian F. Akyildiz and Ismail H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367, 2004.

[15] A. Appriou. Formulation et traitement de l'incertain en analyse multisenseurs. In *GRETSI*, pages 951–954, 1991.

[16] Arduino. `http://www.arduino.cc/`.

[17] Astride Aregui and Thierry Deneoux. Constructing consonant belief functions from sample data using confidence sets of pignistic probabilities. *Int. J. Approx. Reasoning*, 49(3):575–594, November 2008.

[18] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, U. Kulkarni, S.and Arumugam, M. Nesterenko, A. Vora, and M. Miyashita. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks (Elsevier)*, 46(5):605–634, 2004.

[19] Hossein Ghaffarian Bahareh J. Farahani and Mahmood Fathy. A fuzzy based priority approach in mobile sensor network coverage. *International Journal of Recent Trends in Engineering*, 2(1):138–143, November 2009.

[20] J.A. Barnett. Calculating dempster-shafer plausibility. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):599–602, 1991.

[21] N. Bartolini, T. Calamoneri, E. G. Fusco, A. Massini, and S. Silvestri. Snap and spread: A self-deployment algorithm for mobile sensor networks. In *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, pages 451–456, Berlin, Heidelberg, 2008. Springer-Verlag.

[22] Novella Bartolini, Tiziana Calamoneri, Emanuele Guido Fusco, Annalisa Massini, and Simone Silvestri. Push & pull: autonomous deployment of mobile sensors for a complete coverage. *Wirel. Netw.*, 16(3):607–625, 2010.

[23] M. Cardei, Yinying Yang, and Jie Wu. Non-uniform sensor deployment in mobile wireless sensor networks. In *World of Wireless, Mobile and Multimedia Networks, WoWMoM 2008. International Symposium on a*, pages 1–8, 2008.

[24] B. Carter and R. Ragade. An extensible model for the deployment of non-isotropic sensors. In *IEEE Sensors Applications Symposium*, pages 22–25, 2008.

[25] Alberto Cerpa, Jennifer L. Wong, Miodrag Potkonjak, and Deborah Estrin. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '05, pages 414–425, New York, NY, USA, 2005. ACM.

[26] Z. Chair and P. Varshney. Optimal data fusion in multiple sensor detection systems. *IEEE Trans. Aerosp. Electron. Syst*, AES-22(1):98–101, 1986.

[27] Krishnendu Chakrabarty, S. Sitharama Iyengar, Hairong Qi, and Eungchun Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Trans. Comput.*, 51:1448–1453, December 2002.

[28] Krishnendu Chakrabarty, Sitharama S. Iyengar, Hairong Qi, and Eungchun Cho. Coding theory framework for target location in distributed sensor networks. In *Proceedings of the International Conference on Information Technology: Coding and Computing*, ITCC '01, pages 130–134, Washington, DC, USA, 2001.

[29] Xiangmao Chang, Rui Tan, Guoliang Xing, Zhaohui Yuan, Chenyang Lu, Yixin Chen, and Yixian Yang. Sensor placement algorithms for fusion-based surveillance networks. *IEEE Trans. Parallel Distrib. Syst.*, 22:1407–1414, August 2011.

[30] Sriram Chellappan, Xiaole Bai, Bin Ma, Dong Xuan, and Changqing Xu. Mobility limited flip-based sensor networks deployment. *IEEE Trans. Parallel Distrib. Syst.*, 18:199–211, February 2007.

[31] A. Chen, R.R. Muntz, S. Yuen, I. Locher, S.I. Park, and M.B. Srivastava. A support infrastructure for the smart kindergarten. *IEEE Pervasive Computing*, 1(2):49–57, 2002.

[32] Thomas Clouqueur, Veradej Phipatanasuphorn, Parameswaran Ramanathan, and Kewal K. Saluja. Sensor deployment strategy for target detection. In *Proceedings of*

the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02, pages 42–48, New York, USA, 2002.

[33] Thomas Clouqueur, Veradej Phipatanasuphorn, Parameswaran Ramanathan, and Kewal K. Saluja. Sensor deployment strategy for detection of targets traversing a region. *Mob. Netw. Appl.*, 8:453–461, August 2003.

[34] Thomas Clouqueur, Kewal K. Saluja, and Parameswaran Ramanathan. Fault tolerance in collaborative sensor networks for target detection. *IEEE Transactions on Computers*, 53:320–333, March 2004.

[35] C.A.C. Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Comp. Methods in Applied Mechanics and Engineering*, 191:1245–1287, 2002.

[36] Crossbow. Micaz data sheet. `http://bullseye.xbow.com:81/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf`, 2013.

[37] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[38] S. S. Dhillon and K. Chakrabarty. Sensor placement for effective coverage and surveillance in distributed sensor networks. In *Proc. IEEE Wireless Communications and Networking (WCNC)*, volume 3, pages 1609–1614, 2003.

[39] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *Proc. Fifth Int Information Fusion Conf*, volume 2, pages 1581–1587, 2002.

[40] Xiaojiang Du, Ming Zhang, Kendall E. Nygard, Sghaier Guizani, and Hwa Chen. Self-healing sensor networks with distributed decision making. *International Journal of Sensor Networks*, 2(5/6):289–298, July 2007.

[41] Marco F. Duarte and Yu Hen Hu. Vehicle classification in distributed sensor networks. *J. Parallel Distrib. Comput.*, 64:826–838, July 2004.

[42] Didier Dubois and Henri Prade. On the combination of evidence in various mathematical frameworks. In J. Flamm and T. Luisi, editors, *Reliability Data Collection and Analysis*, volume 3 of *Eurocourses*, pages 213–241. Springer Netherlands, 1992.

[43] Zied Elouedi, Khaled Mellouli, and Philippe Smets. Assessing sensor reliability for multisensor data fusion within the transferable belief model. *IEEE Transactions on Systems, Man, and Cybernetics.*, 34(1):782–787, February 2004.

[44] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2033–2036, 2001.

[45] Kevin Fall, Kannan Varadhan, and the VINT project. *The ns manual*, January 2009.

[46] Xin Fei, Azzedine Boukerche, and Regina B. Araujo. Irregular sensing range detection model for coverage based protocols in wireless sensor networks. In *Proceedings of the 28th IEEE conference on Global telecommunications*, GLOBECOM'09, pages 5699–5704, Piscataway, NJ, USA, 2009.

[47] Steven Fortune. *Voronoi diagrams and Delaunay triangulations*, pages 377–388. Goodman, Jacob E. and O'Rourke, Joseph, CRC Press, Inc., Boca Raton, FL, USA, 1997.

[48] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, USA, 1st edition, 1989.

[49] P. Gupta and P. R. Kumar. Critical power for asymptotic connectivity in wireless networks. In *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming,*. W. M. McEneaney, G. Yin, and Q. Zhang (Eds.), 1998.

[50] J. Hou H. Zhang. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1(1–2):89–124, 2005.

[51] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J. Stankovic, T. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An integrated sensor network system for energy-efficient surveillance. *ACM Trans. Sen. Netw.*, 2:1–38, February 2006.

[52] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, ICDCS '03, pages 46–55, Washington, DC, USA, 2003.

[53] Nojeong Heo and P.K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, volume 3, pages 1597–1602 vol.3, 2003.

[54] Andrew Howard, Maja J Mataric, and Gaurav S Sukhatme. Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. In *Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems*, pages 299–308, 2002.

[55] M. Iqbal, I. Gondal, and L. Dooley. Dynamic symmetrical topology models for pervasive sensor networks. In *Proceedings of INMIC*, 2004.

[56] M. Ishizuka and M. Aida. Performance study of node placement in sensor networks. In *Distributed Computing Systems Workshops, International Conference on*, volume 5, pages 598–603, Los Alamitos, CA, USA, 2004.

[57] M. Ishizuka and M. Aida. The reliability performance of wireless sensor networks configured by power-law and other forms of stochastic node placement. *IEICE Trans. on Communications*, E87-B(9):2511–2520, September 2004.

[58] Raj Jain. *The Art of Computer Systems Performance Analysis.* John Wiley & Sons, 1991.

[59] Xingyan Jiang, Yuanzhu Chen, and T. Yu. Localized distributed sensor deployment via coevolutionary computation. In *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, pages 785–789, 2008.

[60] J.M. Kahn, R.H. Katx, and K.S.J. Pister. Next century challenges: mobile networking for smart dust. In *MobiCom*, pages 483–492, 1999.

[61] Wei-Chieh Ke, Bing-Hong Liu, and Ming-Jer Tsai. Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete. *IEEE Transactions on Computers*, 56(5):710–715, 2007.

[62] R. Kershner. The number of circles covering a set. *American Journal of Mathematics*, 61(3):665–671, 1939.

[63] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *The International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–10, 2007.

[64] J. Knowles and D. Corne. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages 98–105, 1999.

[65] Andrei Kolgomorov. *Foundations of the Theory of Probability*. Chelsea Publishing Company, 1950.

[66] Cris Koutsougeras, Yi Liu, and Rong Zheng. Event-driven sensor deployment using self-organizing maps. *International Journal of Sensor Networks*, 3(3):142–151, May 2008.

[67] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 126–137, New York, NY, USA, 2003. ACM.

[68] Philip Levis, Sam Madden, Joseph Polastre, Robert Szewczyk, Kamin Whitehouse, Alec Woo, David Gay, Jason Hill, Matt Welsh, Eric Brewer, David Culler, W. Weber, J. Rabaey, and E. Aarts. *TinyOS: An Operating System for Wireless Sensor Networks*. Ambient Intelligence. Springer-Verlag, 2004.

[69] Mo Li, Weifang Cheng, Kebin Liu, Yuan He, Xiangyang Li, and Xiangke Liao. Sweep coverage with mobile sensors. *Mobile Computing, IEEE Transactions on*, 10(11):1534–1545, 2011.

[70] Shijian Li, Congfu Xu, Weike Pan, and Yunhe Pan. Sensor deployment optimization for detecting maneuvering targets. In *Information Fusion, 2005 8th International Conference on*, volume 2, pages 1629–1635, Philadelphia, PA, USA, 2005.

[71] Y. Li, C. Vu, C. Ai, G. Chen, and Y. Zhao. Transforming complete coverage algorithms to partial coverage algorithms for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22:695–703, 2011.

[72] Arnaud Liefooghe, Laetitia Jourdan, Thomas Legrand, Jérémie Humeau, and El-Ghazali Talbi. Paradiseo-moeo: A software framework for evolutionary multi-objective optimization. In CarlosA. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan, editors, *Advances in Multi-Objective Nature Inspired Computing*, volume 272 of *Studies in Computational Intelligence*, pages 87–117. Springer Berlin Heidelberg, 2010.

[73] K. Lorincz, D.J. Malan, T.R. Fulford-Jones, A. Nawoj, A. Clavel, V. Shnayder, G. Mainland, M. Welsh, and S. Moulton. Sensor networks for emergency response: Challenges and opportunities. *IEEE Pervasive Computing*, 3(4):16–23, 2004.

[74] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 88–97, 2002.

[75] S. Megerian, F. Koushanfar, G. Qu, G. Veltri, and M. Potkonjak. Exposure in wireless sensor networks: Theory and practical solutions. *Journal of Wireless Networks ACM Kluwer Academic Publishers*, 8(5):443–454, 2002.

[76] S. Meguerdichian, F Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1380–1387, 2001.

[77] D. Mercier, T. Denoeux, and M.-H. Masson. General correction mechanisms for weakening or reinforcing belief functions. In *9th International Conference on Information Fusion*, pages 1–7, Florence, Italy, July 2006.

[78] H. Nikookar and H. Hashemi. Statistical modeling of signal amplitude fading of indoor radio propagation channels. In *Universal Personal Communications, 1993. Personal Communications: Gateway to the 21st Century. Conference Record., 2nd International Conference on*, volume 1, pages 84–88 vol.1, 1993.

[79] E. Onur, C. Ersoy, H. Delic, and . Akarun. Surveillance wireless sensor networks: Deployment quality analysis. *Network, IEEE*, 21(6):48–53, November-December 2007.

[80] J. ORourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

[81] Celal Ozturk, Dervis Karaboga, and Beyza Gorkemli. Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm. *Sensors*, 11(6):6056–6065, 2011.

[82] Li Peng, Desheng Wang, and Long Zhao. An algorithm based on market competition for wireless sensor network connectivity and coverage. In *Information and Automation, 2008. ICIA 2008. International Conference on*, pages 379–383, 2008.

[83] M. Petrova, J. Riihijarvi, P. Mahonen, and S. Labella. Performance study of ieee 802.15.4 using measurements and simulations. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 1, pages 487–492, 2006.

[84] Bastien Pietropaoli, Michele Dominici, and Frédéric Weis. Belief inference with timed evidence : Methodology and application using sensors in smart home. In *The 2nd International Conference on Belief Functions*, pages 409–416, 9-11 May 2012.

[85] Guang ping Qi, Ping Song, and Ke jie Li. Blackboard mechanism based ant colony theory for dynamic deployment of mobile sensor networks. *Journal of Bionic Engineering*, 5(3):197–203, 2008.

[86] S. Poduri and G. Sukhatme. Constrained coverage for mobile sensor networks. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 1, pages 165–171, 2004.

[87] D. Pompili, T. Melodia, and I. F. Akyildiz. Deployment analysis in underwater acoustic wireless sensor networks. In *the Proceedings of the ACM International Workshop on UnderWater Networks (WUWNet)*, Los Angeles, CA, September 2006.

[88] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[89] C. S. Sahin, E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo. Self-deployment of mobile agents in manets for military applications. In *Army Science Conference*, pages 1–8, 2008.

[90] Ruhul Sarker and CarlosA. Coello Coello. Assessment methodologies for multiobjective evolutionary algorithms. In *Evolutionary Optimization*, volume 48 of *International Series in Operations Research & Management Science*, pages 177–195. Springer US, 2003.

[91] M.R. Senouci, A. Mellouk, and A. Aissani. An analysis of intrinsic properties of stochastic node placement in sensor networks. In *IEEE Global Telecommunications Conf. GLOBECOM'12*, pages 494–499, Houston, Texas, USA, December 2012.

[92] M.R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani. Uncertainty-aware sensor network deployment. In *IEEE Global Telecommunications Conf. GLOBECOM'11*, pages 1–5, Houston, Texas, USA, December 2011.

[93] M.R. Senouci, A. Mellouk, L. Oukhellou, and A. Aissani. Efficient uncertainty-aware deployment algorithms for wireless sensor networks. In *IEEE Wireless Communications and Networking Conf. WCNC'2012*, pages 2163–2167, Paris, France, April 2012.

[94] Mustapha Reda Senouci, Mellouk Abdelhamid, and Khaled Assnoune. Localized movement-assisted sensor deployment algorithm for hole detection and healing. *IEEE Transactions on Parallel and Distributed Systems*, 99(PrePrints):1, 2013.

[95] Mustapha Reda Senouci, Abdelhamid Mellouk, Hadj Senouci, and Amar Aissani. Performance evaluation of network lifetime spatial-temporal distribution for wsn routing protocols. *Journal of Network and Computer Applications*, 35(4):1317–1328, 2012.

[96] Kari Sentz and Scott Ferson. Combination of evidence in dempster-shafer theory. Technical Report SAND 2002-0835, Sandia National Laboratories, April 2002.

[97] C. Sergiou and V. Vassiliou. Energy utilization of htap under specific node placements in wireless sensor networks. In *EW2010*, pages 482–487, Lucca, Italy, April 2010.

[98] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[99] Haining Shu and Qilian Liang. Fuzzy optimization for distributed sensor deployment. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1903–1908, 2005.

[100] G.T. Sibley, M.H. Rahimi, and G. Sukhatme. Robomote: a tiny mobile robot platform for large-scale ad-hoc sensor networks. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 2, pages 1143–1148, 2002.

[101] G. Simon, M. Maróti, Ákos L., G. Balogh, B. Kusy, A. Nádas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based counter sniper system. In *ACM the 2nd International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–12, 2004.

[102] A. Sinha and B. Pal. Stensor: A novel stochastic algorithm for placement of sensors in a rectangular grid. In *paper-presentation competition "Eureka" in Kshitij, IIT Kharagpur*, 2007.

[103] P. Smets. Belief functions. In D.Dubois P. Smets, E.H. Mamdani and H. Prade, editors, *Non-Standard Logics for Automated Reasoning*, pages 253–286. Academic Press, London, 1988.

[104] Ph. Smets. The combination of evidence in the transferable belief model. *IEEE Pattern analysis and Machine Intelligence*, 12:447–458, 1990.

[105] Ph. Smets. Decision making in the TBM: the necessity of the pignistic transformation. *Int. J. Approximate Reasoning*, 38:133–147, 2005.

[106] Ph. Smets and R. Kennes. The transferable belief model. *Artificial Intelligence*, 66:191–234, 1994.

[107] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline. Pipenet: A wireless sensor network for pipeline monitoring. In *The International Conference on Information Processing in Sensor Networks (IPSN)*, pages 1–10, 2007.

[108] Yulai Suen. A genetic-algorithm based mobile sensor network deployment algorithm. Technical report, Department of Electrical and Computer Engineering, University of Texas at Austin, 2008.

[109] El-Ghazali Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

[110] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia. Exploiting data fusion to improve the coverage of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 20(2):450–462, 2011.

[111] Di Tian and Nicolas D. Georganas. Connectivity maintenance and coverage preservation in wireless sensor networks. *Ad Hoc Networks*, 3(6):744–761, November 2005.

[112] Yu-Chee Tseng, You-Chiun Wang, Kai-Yang Cheng, and Yao-Yu Hsieh. imouse: An integrated mobile surveillance and wireless sensor system. *Computer*, 40(6):60–66, 2007.

[113] Ricquebourg V., Delafosse M., Delahoche L., and Marhic B.and Jolly-Desodt A.M.and Menga D. Fault detection by combining redundant sensors : a conflict approach within the tbm framework. In Stanford University, editor, *COGIS 2007, COGnitive systems with Interactive Sensors*, California, USA, 26 - 28 November 2007.

[114] P. Varshney. *Distributed Detection and Data Fusion*. Springer-Verlag, 1996.

[115] V. Vassiliou and C. Sergiou. Performance study of node placement for congestion control in wireless sensor networks. In *3rd International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–8, December 2009.

[116] L. Vieira, M. Vieira, L. Beatriz, A. Loureiro, D. Junior, and A. Fernandes. Efficient incremental sensor network deployment algorithm. In *Brazilian Symposium on Computer Networks*, 2004.

[117] Wang W., Srinivasan V., Chua K.C., and Wang B. Energy-efficient coverage for target detection in wireless sensor networks. In *The 6th International Conference on Information Processing in Sensor Networks (IPSN)*, pages 313–322, 2007.

[118] Benjamin W. Wah, Yixin Chen, and Tao Wang. Simulated annealing with asymptotic convergence for nonlinear constrained optimization. *J. of Global Optimization*, 39(1):1–37, September 2007.

[119] P. Wan and C. Yi. Coverage by randomly deployed wireless sensor networks. *IEEE/ACM Trans. Netw.*, 14(SI):2658–2669, June 2006.

[120] B. Wang. *Coverage Control in Sensor Networks*. Springer, New York, 2010.

[121] Bang Wang, Hock Beng Lim, and Di Ma. A survey of movement strategies for improving network coverage in wireless sensor networks. *Computer Communications*, 32(13-14):1427–1436, 2009.

[122] D. Wang, B. Xie, and D.P Agrawal. Coverage and lifetime optimization of wireless sensor networks with gaussian distribution. *IEEE Transactions on Mobile Computing*, 7(12):1444–1458, December 2008.

[123] Guiling Wang, Guohong Cao, T. La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2302–2312 vol. 4, 2005.

[124] Guiling Wang, Guohong Cao, and Tom La Porta. Movement-assisted sensor deployment. *Mobile Computing, IEEE Transactions on*, 5(6):640–652, 2006.

[125] Guiling (Grace) Wang, Guohong Cao, Piotr Berman, and Thomas F. La Porta. Bidding protocols for deploying mobile sensors. *IEEE Transactions on Mobile Computing*, 6(5):563–576, 2007.

[126] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *Proceedings of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 28–39, New York, NY, USA, 2003. ACM.

[127] Xue Wang, Sheng Wang, and Junjie Ma. Dynamic deployment optimization in wireless sensor networks. In De-Shuang Huang, Kang Li, and GeorgeWilliam Irwin, editors, *Intelligent Control and Automation*, volume 344 of *Lecture Notes in Control and Information Sciences*, pages 182–187. Springer Berlin Heidelberg, 2006.

[128] Xueqing Wang, Yongtian Yang, and Yibing Song. Redundant movement-assisted sensor deployment based on virtual rhomb grid in wireless sensor networks. In *Mechatronics and Automation, Proceedings of the 2006 IEEE International Conference on*, pages 775–779, 2006.

[129] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng. Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *Proc. First Int Wireless Internet Conf*, pages 114–121, 2005.

[130] You-Chiun Wang and Yu-Chee Tseng. Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage. *Parallel and Distributed Systems, IEEE Transactions on*, 19(9):1280–1294, 2008.

[131] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing*, 10(2):18–25, 2006.

[132] T. Wong, T. Tsuchiya, and T. Kikuno. A self-organizing technique for sensor placement in wireless micro-sensor networks. In *Advanced Information Networking and Applications, 2004. AINA 2004. 18th International Conference on*, volume 1, pages 78–83, 2004.

[133] X. Wu, G. Chen, and S.K. Das. Avoiding energy holes in wireless sensor networks with nonuniform node distribution. *IEEE Trans. Parallel Distrib. Syst.*, 19(5):710–720, May 2008.

[134] G. Xing, R. Tan, B. Liu, J. Wang, X. Jia, and C. Yi. Data fusion improves the coverage of wireless sensor networks. In *MobiCom '09*, pages 157–168, New York, USA, 2009.

[135] N. Xu, S. Rangwala, K.K. Chintalapaudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *CM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 13–24, 2004.

[136] R. R. Yager, J. Kacprzyk, and M. Fedrizzi, editors. *Advances in the Dempster-Shafer Theory of Evidence*. Wiley, New York, 1994.

[137] G. Yang and D. Qiao. Barrier information coverage with wireless sensors. In *IEEE Infocom*, pages 918–926, 2009.

[138] G.Z. Yang. *Body Sensor Networks*. Springer, Berlin, 2006.

[139] Minghua Yang, Yuanda Cao, Li Tan, and Jiong Yu. An enhanced self-deployment algorithm in mobile sensor network. In *International Seminar on Future Information Technology and Management Engineering, FITME '08*, pages 573–576, 2008.

[140] Shuhui Yang, Minglu Li, and Jie Wu. Scan-based movement-assisted sensor deployment methods in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 18(8):1108–1121, 2007.

[141] Shuhui Yang, Jie Wu, and Fei Dai. Localized movement-assisted sensor deployment in wireless sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 753–758, 2006.

[142] Z. Yong and W. Li. A sensor deployment algorithm for mobile wireless sensor networks. In *Proceedings of the 2nd annual international conference on Chinese control and decision conference*, pages 4642–4647, Guilin, China, 2009.

[143] M. Younis and K. Akkaya. Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Netw.*, 6:621–655, June 2008.

[144] O. Younis and Sonia Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, 2004.

[145] Z. Yun, X. Bai, D. Xuan, T. H. Lai, and W. Jia. Optimal deployment patterns for full coverage and k-connectivity ($k \leq 6$) wireless sensor networks. *IEEE/ACM Trans. Netw.*, 18(3):934–947, 2010.

[146] L.A. Zadeh. Fuzzy sets. *Information Control*, 8:338–353, 1965.

[147] L.A. Zadeh. Fuzzy sets as a basis for theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[148] H. Zhang and J. Hou. On deriving the upper bound of α-lifetime for large sensor networks. In *MobiHoc*, pages 121–132, 2004.

[149] J. Zhang, T. Yan, and S. H. Son. Deployment strategies for differentiated detection in wireless sensor networks. In *3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON*, volume 1, pages 316–325, 2006.

[150] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In Agoston Eiben, Thomas Bäck, Marc Schoenauer, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer Berlin Heidelberg, 1998.

[151] Y. Zou and K. Chakrabarty. Sensor deployment and target localization based on virtual forces. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1293–1303, 2003.

[152] Y. Zou and K. Chakrabarty. Uncertainty-aware sensor deployment algorithms for surveillance applications. In *Proc. IEEE Global Telecommunications Conf. GLOBECOM '03*, volume 5, pages 2972–2976, 2003.

[153] Y. Zou and K. Chakrabarty. Uncertainty-aware and coverage-oriented deployment for sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):788 – 798, 2004.

[154] Y. Zou and K. Chakrabarty. A distributed coverage- and connectivity-centric technique for selecting active nodes in wireless sensor networks. *IEEE Transactions on Computers*, 54(8):978–991, 2005.