NONUNIFORMITY IN WIRELESS SENSOR NETWORKS

by

Rabun Koşar

BS. in Computer Engineering, Boğaziçi University, 2000

MS. in Computer Engineering, Boğaziçi University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2010

to

my mother

Sayme Koşar

ACKNOWLEDGEMENTS

It has been a long journey from the very beginning to the very end. As it is said, it is the journey that matters not the destination. I have been lucky enough to be accompanied by the best company and the list is very long.

First I would like to thank my advisor Prof. Cem Ersoy for everything he has done. He was, is and will always be my role model. I feel very fortunate to know him. I would like to thank professors Fatih Alagöz, Taner Bilgiç, Feza Buzluca and Tuna Tuğcu for being in my thesis jury and for their support throughout the progression of the thesis.

As a member of the computer engineering department, I've had the greatest colleagues. My heartfelt thanks go to İlker Demirkol, he was always near when I faced difficulties, thanks a lot bro. Special thanks to Yunus Dönmez, together with whom I prepared the analytical work in my thesis. I would also like to thank my assistant friends Atay Özgövde, Hande Alemdar, Oya Aran, Itır Karaç, Neşe Alyüz, İsmail Arı, Reyhan Aydoğan, Gül Çalıklı, Dağhan Dinç, Barış Gökçe, Mehmet Gönen, Cem Keskin, Şükrü Kuran, Ali Haydar Özer, Nuri Taşdemir, Metin İnanç, Cem Yüksel, Onur Dikmen, M. Erdem Türsen, Haydar Vural, Serdar Salı, Çiğdem Gündüz, Levent Özgür, M. Okan İrfanoğlu, Olcay Taner Yıldız, Berk Gökberk, Albert Ali Salah, Koray Balcı, Kubilay Atasu, Burak Turhan, Abuzer Yakaryılmaz, Aydın Ulaş, Tamer Demir. Moreover, starting with Onur Büyükceran, I want to thank, in no particular order, to Can Komar, Bilgin Koşucu, Remzi Yavuz, Sinan Işık, Ozan Özen, Evren Önem, Tahir Çelebi, Derya Çavdar, Yunus Emre Kara, Gaye Genç, Pınar Santemiz, Almıla Akdağ Salah, Didem Gözüpek, Suzan Bayhan, Bora Zeytinci, Çetin Meriçli, Tekin Meriçli, Muharrem Derinkök, Sibel Karamaraş, Selin Bakar, Duygu Öztürk, Hatice Emecan, Murat Asker, Kamran Aliyev, Orhan Aliyev, Cem Vurnal, Oğuzcan Hardal, Deniz Yavuz, Yeliz Ayan, Burçak Akbıyık, Selim Erdoğan, Faruk Ordu and Gülsüm Sonal, Jale Yılmaz, Ali Rıza Dikici for being my friends and Ayşe Solak for being a constant source of fun and fights.

That many years would not pass without social activities and hobbies. I would like to thank Emre Camadan, Hamit Erentürk, Onur Ekin Bayıldıran, Salsa Mambo Team, Salsa Dans Vukuat, Aydın Kocamusaoğlu, Pelin Koyun, Onur Kıra, Çiğdem Camgöz, Özhan Araz, Tangoist, Büdans Tango for all the joy and the fun times.

This thesis is written under the influences of Mongo Santamaria, Ray Barretto, Hector Lavoe, Frankie Dante, Eddie Palmieri, Charlie Palmieri, Cal Tjader, Andy Montañez, Tito Puente, Federico Aubele, Gotan Project, Carlos Libedinsky, Koop, Oi Va Voi, Lalo Schifrin, Bajofondo, Francisco Canaro, Juan D'Arienzo (con Alberto Echagüe), Rodolfo Biagi, Osvaldo Pugliese, Miguel Caló and the one and only Ástor Pantaleón Piazzolla.

Special thanks must be reserved for Sinan Kuşdoğan and Cumhur Cebeci for showing me the meaning of good friendship. And I would like to thank my special friends during all those years whose names I cannot mention here. You all will have your special places in my heart.

And... Thank you mom. I know that all the thank you's in the world would never be enough to thank you for all you have done. I love you. This thesis is dedicated to you, my dearest Sayme Koşar.

ABSTRACT

NONUNIFORMITY IN WIRELESS SENSOR NETWORKS

Deemed as suitable for various challenging sensing tasks, wireless sensor networks evolve as candidates for deployment to different environments to perform the given task for as long as possible. Those networks operate redundantly in a distributed manner, requiring little intervention. However, various factors such as the sink location, node deployment characteristics or external impacts such as intentional destructions can severely limit the lifetime of the overall network. The nonuniformity of the networks caused by such factors may result in underutilized network deployments where some parts of the network is still alive yet unable to reach the sink due to disconnections in the intermediate sections.

In this thesis, the effects of deployment nonuniformity is analyzed and methods for mitigating the problems are presented. A sink placement algorithm is presented to find a safe and optimal location for the sink node. Methods to find the bottleneck nodes and sensing holes inside the network are provided. Redeployment based hole mitigation techniques are proposed to prolong the network lifetime under strict quality requiring situations such as border surveillance. An analytical quality metric is also presented to understand the sensing quality under lossy assumptions. The presented methods and models are tested using simulations and the results for different parameter sets are given to see the suitability of each.

ÖZET

KABLOSUZ ALGILAYICI AĞLARDA BİRÖRNEKSİZLİK

Kablosuz algılayıcı ağlar birbirinden farklı ortamlarda verilen algılama görevini uzun süre yapmak konusunda en uygun aday olarak ortaya çıkmaktadırlar. Bu tür ağlar özellikle zorluk derecesi fazla algılama görevleri için uygun görülmektedirler. Ağların en büyük avantajları, dağıtık bir yapıda, asgari dış etmen kullanarak ve artıklık sağlayarak çalışmasıdır. Ancak veri toplama düğümünün yeri, algılayıcı düğümlerin alana yollanma şekli ya da dış saldırılar ağ ömrünü önemli ölçüde azaltmaktadır. Bu tür faktörler yüzünden oluşan birbiçimsiz ağ şekli ağ kaynaklarının yeteri kadar kullanılamamasına sebep olmakta ve ağ içerisinde hala ayakta bulunan ancak ara alanlarda oluşan kesintiler yüzünden toplama düğümüne erişemeyen kısımlar oluşmaktadır.

Bu tez içerisinde birbiçimsiz dağılımın etkileri incelenmekte ve oluşan problemlere uygun çözümler önerilmektedir. Toplama düğümünü uygun ve güvenli bir noktaya koymak için bir düğüm yerleştirme yordamı önerilmiştir. Sıkışma noktalarını ve ağ boşluklarını bulma yöntemleri ve sınır takibi gibi yüksek seviye algılama ihtiyacı olan ortamlara uygun ağ ömrünü uzatan boşluk kapatma yordamı sunulmaktadır. Kayıplı ortamlarda oluşan algılama kalitesini ölçmek için, analitik bir kalite ölçme hesaplaması tez içerisinde önerilmiş ve ayrıntılı olarak incelenmiştir. Sunulan yöntem ve modellerin uygunluğu benzetim yöntemleri ile teste tabi tutulmuş ve sonuçlar tez içerisinde sunulmuştur.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT v
ÖZET vi
LIST OF FIGURES ix
LIST OF TABLES xiii
LIST OF SYMBOLS/ABBREVIATIONS xv
1. INTRODUCTION 1
2. SINK PLACEMENT ON A 3D TERRAIN FOR BORDER SURVEILLANCE 7
2.1. Sink Node Placement 10
2.1.1. Coding 14
2.1.2. Genome Operators 14
2.1.3. Selection 15
2.1.4. Fitness Function 15
2.2. Results 18
2.2.1. Results for Flat World Scenarios 21
2.2.2. Results for Three Dimensional Scenarios 23
3. IDENTIFICATION OF BOTTLENECK AREAS FOR PARTIAL REDEPLOYMENT 29
3.1. Experimental Results 33
4. ENERGY HOLE MITIGATION FOR SURVEILLANCE 46
4.1. Mitigating the Energy Holes 47
4.2. Iterative Sensor Deployment Method 50
4.2.1. Iso-sensing Graph and Deployment Quality 51
4.2.2. Hole Identification 52
4.2.3. Redeployment 55
4.3. Simulation Setup and Results 57
5. ANALYTICAL APPROACH TO DEPLOYMENT QUALITY FOR SURVEILLANCE CONSIDERING HOLES 67
5.1. Related Work 68

5.2. Model Assumptions 70
5.3. Problem Definition 71
5.4. Deployment Quality of a Border Surveillance WSN 74
5.5. Analytical Results 78
5.5.1. The combined effect of area size and sensor count on DQM values 80
5.5.2. The combined effect of area size, jamming area radius and jamming area count on DQM values 81
5.5.3. The combined effect of jamming area radius and jamming area count on DQM values 88
5.5.4. The combined effect of sensor count and jamming area count on DQM values 89
6. Conclusions 92
REFERENCES 95

LIST OF FIGURES

Figure 1.1. Example figure for bottleneck scenario. 4

Figure 2.1. Surveillance line of sight blockage due to terrain obstructions. . . . 11

Figure 2.2. A synthetically generated terrain sample. 11

Figure 2.3. Basic scheme of the proposed approach. 13

Figure 2.4. A sample for the coding scheme. 15

Figure 2.5. Sample network, where the death of an intermediate node disconnects the offspring alive nodes. 18

Figure 2.6. Values for different area shapes. 20

Figure 2.7. Values for different sensing range values. 21

Figure 2.8. Values for different sensor node count values. 22

Figure 2.9. Values for different coverage requirement values. 23

Figure 2.10. Values for different maximum communication range values. 24

Figure 2.11. Synthetically created terrain used in the simulations. 24

Figure 2.12. Values for different area types on the three dimensional terrain. . . 25

Figure 2.13. Values for different communication ranges on the three dimensional terrain. 26

Figure 2.14. Values for different number of sensors on the three dimensional terrain. 26

Figure 2.15. Values for different sensing ranges on the three dimensional terrain. 27

Figure 2.16. Values for different coverage requirement values on the three dimensional terrain. 28

Figure 3.1. Border nodes that have high traffic nodes. 30

Figure 3.2. A sample scenario showing the sensors deployed over different sized rectangular areas of the network. 34

Figure 3.3. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.7. 37

Figure 3.4. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.9. 38

Figure 3.5. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 80 m, coverage requirement value of 0.7. 40

Figure 3.6. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 80 m, coverage requirement value of 0.9. 41

Figure 3.7. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of four, communication range of 60 m, coverage requirement value of 0.7. 42

Figure 3.8. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of four, communication range of 60 m, coverage requirement value of 0.9. 43

Figure 3.9. Effect of node size on network lifetime for message size of 1024 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.7. 45

Figure 4.1. A simple border surveillance scenario where sensors are intentionally destructed. 50

Figure 4.2. A sensor network iso-sensing map example. 53

Figure 4.3. Iso-sensing graph converted to a color image. 54

Figure 4.4. Color image formed after filtering low sensing quality areas, applying connected component analysis and filtering small sized groups in the initial image. 55

Figure 4.5. Final color image formed after redeployment is performed, 'x' sign denotes the sensors deployed in the redeployment phase. 56

Figure 4.6. Energy hole mitigation using different redeployment techniques. . 64

Figure 4.7. Total sensors used vs. lifetime. 65

Figure 4.8. Lifetime comparison. 66

Figure 4.9. WDQM vs. time. 66

Figure 5.1. Graphical representation of the border surveillance intruder detection avoiding jammers problem. 71

Figure 5.2. Graphical representation of the line-set intersection problem. . . . 72

Figure 5.3. Effect of area size and sensor count on analytical and simulated DQM values. Lines are ordered from the greatest sensor count value (900) at the top to the smallest value (500) at the bottom. . 81

Figure 5.4. Effect of area size for the jamming area radius of 50 m on analytical and simulated DQM values. The remaining results for are tabulated in Table 5.3 for the circular scenario and in Table 5.4 for the rectangular scenario. 82

Figure 5.5. Effect of area size for no jamming area case on analytical and simulated DQM values. The remaining results for are tabulated in Table 5.5 for the circular scenario and in Table 5.6 for the rectangular scenario. 85

Figure 5.6. Effect of jamming area radius and jamming area count on analytical and simulated DQM values. Lines are ordered from the smallest jamming area count value (0) at the top to the greatest value (80) at the bottom. 85

Figure 5.7. Effect of high sensor count and jamming area count on analytical and simulated DQM values. 91

LIST OF TABLES

Table 2.1. Hybrid genetic algorithm, GAUSS, steps. 17

Table 2.2. Base genetic algorithm test parameters. 19

Table 2.3. GAUSS simulation test parameters. 20

Table 3.1. BAIA steps. 32

Table 3.2. BAIA test parameters. 35

Table 4.1. IDeA steps. 57

Table 4.2. IDeA steps, distributed version. 58

Table 4.3. Hole mitigation test sensor parameters. 61

Table 4.4. Hole mitigation test network parameters. 62

Table 4.5. Hole mitigation test intruder mobility parameters. 62

Table 5.1. Circular deployment region scenario test parameters. 79

Table 5.2. Rectangular deployment region scenario test parameters. 80

Table 5.3. Analytical and simulated DQM values for the areas with different radii and jamming area radii. Circular deployment region scenario version. 83

Table 5.4. Analytical and simulated DQM values for different area sizes and jamming area radii. Rectangular deployment region scenario. . . . 84

Table 5.5. Analytical and simulated DQM values for the areas with different radii and jamming area counts. Circular deployment region scenario version. 86

Table 5.6. Analytical and simulated DQM values for different area sizes and jamming area counts. Rectangular deployment region scenario version. 87

Table 5.7. Analytical and simulated DQM values for the areas with different sensor counts and different jamming area counts. High number of sensors, circular deployment region scenario version. 90

LIST OF SYMBOLS/ABBREVIATIONS

A	Deployment area
C	A bounded set
G	Intruder path line
h_j	Index of a jammer
H_j	Jamming coverage area of a jammer h_j
I	Color image formed
L	Perimeter of deployment area
L_i	Perimeter of sensing coverage area of a sensor s_i
N_h	Number of jammers
N_s	Number of deployed sensors
p	Distance of the intruder line to the origin
P_a	Probability that a sensor resides outside all of the jamming areas
P_D	Probability that the network detects an intruder
$P_D(k)$	Probability that intruder is detected by at least one sensor
P_i	Probability that intruder is detected by a single sensor s_i
r	Sensing distance threshold in Elfes model
r_e	Measure of uncertainty in sensor detection in Elfes model
R	Radius of circular deployment area
R_h	Radius of jamming area, where all jammers have identical effective area
R_s	Radius of sensing coverage of a sensor, where all sensors have identical coverage
S	Convex deployment area
s_i	Index of a sensor
S_i	Sensing coverage area of a sensor s_i
$w1$	Connectivity filter
$w2$	Small region filter
X	Intruder

β	Sensing parameter based on the sensor characteristics in Elfes model
λ	Sensor parameter based on the sensor characteristic in Elfes model
ϕ	Angle of line perpendicular to intruder line w.r.t. x axis.

BAIA	Bottleneck Area Identification Algorithm
COM	Center Of Mass
DQM	Deployment Quality Measure
GA	Genetic Algorithm
GAUSS	Genetic Algorithm Using Sensor Simulation
IDeA	Iterative Deployment Algorithm
MAC	Medium Access Control
MATLAB	Matrix Laboratory
NDC	Neighborhood Density Control
SWSN	Surveillance Wireless Sensor Network
VSN	Video Sensor Network
WDQM	Watershed Deployment Quality Measure
WSN	Wireless Sensor Network

1. INTRODUCTION

Due to their distributed operation ability, requirement of little maintenance during operation, scalable coverage range and the ability to operate on difficult deployment sites, wireless sensor networks (WSN) are very suitable for sensing different phenomenon over large areas. Composed of many battery powered sensing/communication units, sensors, and mostly one data gathering unit called a sink, these networks are dispersed over the area to perform a given task for as long as possible. Sensors have a goal of monitoring the environment for different types of events like forest fires, border intrusions, chemical gas leakages.

The key steps of the whole life line of a sensor network can be listed as:

- Deployment of the sensors over the area.
- Initial discovery and communication between nodes, network formation.
- Start of environment sensing.
- In case of an event detection, related data is sent to the sink.
- Routing information update (if needed).
- Network announced dead when sensing or communication quality falls below a certain threshold.

Commonly adapted algorithms for routing, communication, location planning are not suitable for WSNs because of the architecture of the sensors, mainly due to lack of high range communication and high energy capacities [1]. Yet, the distributed and automated nature of a WSN makes it tolerant against partial losses of sensing units.

Further developments in microelectronics have enabled the evolution of the sensors to bear devices such as video cameras and made way for video sensor networks (VSN). WSN and VSN appear to be potential candidates for the solutions of inherently difficult tasks such as target tracking and surveillance applications [2]. Such applications generally impose operation restrictions over very large areas and harsh

environments. In order to meet the requirements, sensors must be deployed in large chunks by means of mass deployment methods. Popular choices for mass deployment are using aviation means like planes, helicopter or land means like robots, automobiles or infantry. Especially for harsh environments such as high altitude mountain borders, forests the deployment results in highly nonuniform placement of sensors.

Nonuniform network formation is generally unavoidable for a sensor network. Initial deployment characteristics can result in such a formation. Many–to–one routing from the sensors to the sink can bring nonuniformity. Intermediate nodes that relay the information spend their batteries earlier than others and die out. Loss of intermediate nodes also lead to nonuniformity in the network. External effects such as intentional destruction of nodes, jamming, field obstructions may have similar effects. The most important outcome of nonuniform formations is the decreased lifetime. For a WSN there are different types of lifetime criteria such as the death of first node or last nodes, death of a certain percentage of nodes, inability to cover a percentage of the deployment region, the fall of the sensing quality below a threshold. All are based on the lifetime of the individual nodes. If all the nodes use their batteries for similar time periods, network has the optimal lifetime, however nonuniformity causes some nodes to die earlier.

Deployment and operation of a sensor network is a costly process and requires high amount of initial effort. The autonomous operation ability and longer lifetime of the network is expected to offset the network design effort and cost, so shorter lifetime must be avoided by all means.

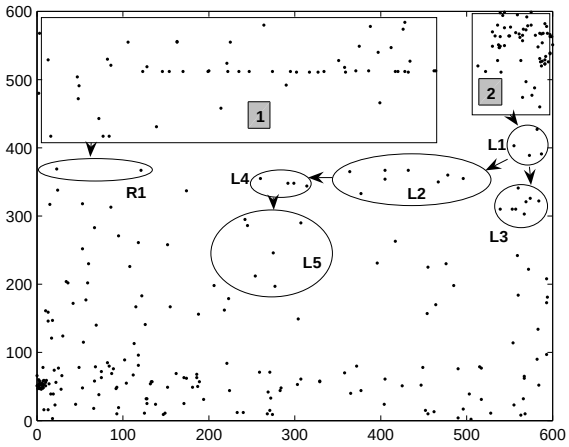
It is possible increase the lifetime of network by optimizing different parameters of the network and mitigating the hole occurrences inside the network. This thesis proposes methods that are based on the optimized placement of sink and redeployment of the sensor nodes to increase the lifetime.

The distance to sink node from a sensor directly affects the energy expenditure of the initiator node and the relaying nodes. Given an already deployed network, changing

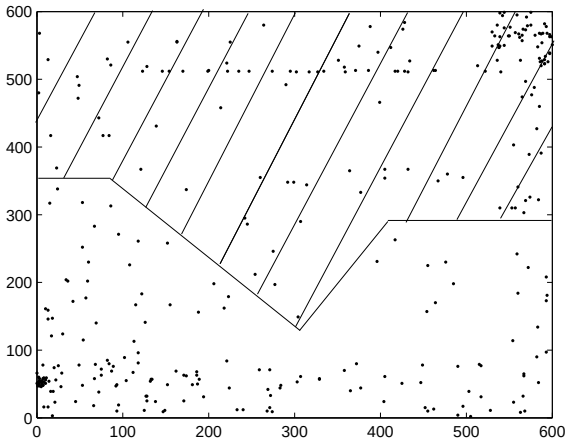
the node locations is not practical at times. However, the location of the sink can be chosen based on some parameters. In this thesis, a method is proposed to find the sink location that provides longer lifetime values and is in a relatively safe location. Compared to two different heuristics the lifetime gains are presented.

As previously stated, node deaths limit the network lifetime although there may be still operational nodes in the network. Due to the inherent redundancy, not all of the dying nodes have direct effect on the network lifetime. Nodes, which disconnect a part of the network when dead, are of more importance from this point of view. Such nodes are critical nodes, since their loss can limit the overall network lifetime. For most scenarios, when the area covered by the network falls below a certain ratio of the initially covered area, the lifetime of the network may be said to have ended. For such lifetime calculations, death of critical nodes result in loss of coverage for the network, which is certainly not desirable under the initially given coverage requirements. Discovery of those nodes in the earlier stages of the network formation make it possible to prolong the network lifetime without too much overhead and with smaller costs.

Figure 1.1 shows examples of bottleneck nodes. In the figure sink is placed on the lower-left section of the field. Simulations on the network have shown that the routing traffic is concentrated on some particular nodes as seen in Figure 1.1a. Region marked as 1 passes all the sensing traffic onto the nodes in the ellipse marked as "R1". For this part of the network, the bottleneck nodes are those within the "R1" region. For region marked as 2, the traffic passes through either "L1, L2, L4, L5" or "L1, L3" paths. For this part of the network, sensors inside the "L1", "L5" and "L3" regions are the bottleneck nodes. Figure 1.1b shows the network status when the simulation terminates, where the shaded area shows the sensors that cannot communicate with the sink. The criterion for the termination is that the area covered by the sensors which can communicate with the sink fall below a certain threshold value. Results have shown that most of the nodes inside the unreachable region still have high battery levels. However, the death of sensors within the presumed bottleneck regions lead to the death of the network since regions one and two are not able to communicate with the sink anymore.



(a) Bottleneck sensors and networks relying on them.



(b) Network connection status at the end of simulation.

Figure 1.1. Example figure for bottleneck scenario.

The load imbalance comes from the fact that some nodes provide the best routing choice and are chosen as the next hop by nearly all of the relaying nodes. Those nodes die out soon to leave the network disconnected and crippled in terms of coverage. Such nodes are found mostly at the perimeter of some local clusters and do not have nearby alternative nodes. We see that those nodes are located near some uncovered regions that form the border areas between clusters.

In the thesis, a method to locate the bottleneck nodes are presented. The method finds the locations that include bottleneck nodes and performs sensor redeployment at

those places to decrease the risk of network deaths caused by them. Based on the observations while designing the algorithm for bottleneck nodes, a novel method to locate the holes within a network is also proposed. The proposed Iterative Deployment Algorithm (IDeA) uses image processing methods to find the probable hole locations for a given deployment. Directed deployments at such locations are performed to remedy the problems caused by the holes. IDeA is extended to mitigate the temporal hole formations and increase the lifetime of the network.

To calculate the sensing quality of a sensor network there are different approaches like calculating the total coverage area, the worst case sensing performance of the network, the number of sensors covering various locations of the network. In this thesis, an analytical method is proposed. The proposed method calculates the sensing probability of a network where the sensing capability may be crippled due to holes of various types.

The contributions in this thesis can be listed as follows:

- A hybrid, simulation-genetic algorithm combined, algorithm is proposed to find a safe location for the sink. The resulting location also increases the network lifetime. The proposed algorithm has three dimensional terrain provisions.
- An algorithm to find the critical bottleneck nodes in a wireless sensor network is proposed.
- Image processing based algorithm to find the holes in a wireless sensor network is proposed. The algorithm proposes a redeployment method to mitigate the holes found.
- An analytical deployment measure is proposed to calculate the sensing quality of wireless sensor networks where some sensors are lost due to holes.

The organization of the thesis is as follows: Chapter 2 provides the details about the sink placement problem with references to the related work in the literature. Then a hybrid genetic algorithm is proposed to place the sink in a border surveillance network. The target location is sought to be both lifetime increasing and safe for sink and the

network. The terrain is three dimensional and obstructions are taken into consideration for the presented algorithm. In Chapter 3, a method to find the bottleneck nodes in the network is proposed. Bottleneck nodes are the sensors that carry relatively higher traffic compared to the other sensors and are prone to die quickly and disconnect the network. To solve the early death problems, sensor redeployment to locations close to the bottleneck nodes are performed. In Chapter 4, the hole problem is explained, the causes and results are detailed. To solve the hole problem, IDeA is proposed and the steps are shown using a demonstrative example. Chapter 5 presents an analytical deployment measure to be used as a quality metric for sensor networks where hole occurrences are unavoidable. Chapter 6 concludes the thesis.

2. SINK PLACEMENT ON A 3D TERRAIN FOR BORDER SURVEILLANCE

There are various challenges associated with the design and operation of WSNs. Due to the node architecture and battery usage, the communication ranges of the sensors are limited. Most nodes cannot reach the sink node directly and routing over other nodes is inevitable. This fact increases the energy burden on the intermediate routing nodes, because the message receiving and submitting operations are very energy consuming and for every operation there is unavoidable energy loss which cannot be lowered without changing the electronic architecture of the nodes. The energy terms for a node is a total of the energy to sense, to receive and to transmit. The transmission energy term depends on the distance and the medium, whereas other terms are fixed costs. Transmission cost depends on the path loss value, a physical property of the medium and is proportional to the path loss exponent and distance. Very long communication distances cause high energy consumptions, yet increasing the hop count and decreasing the distance increases the receiving costs in the network. For three dimensional terrains, obstructions can also cause communication and sensing disruptions.

Given those facts, it is very crucial to optimize the network parameters as much as possible to increase the sensing quality and the network lifetime. Many approaches have been proposed to optimize the information routing from sensors to the sink, optimize the radio communication between sensors, decrease the battery expenditure by sleep schedules. One other approach is to adjust the coordinates of the sink nodes such that the overall energy loss due to the routing is minimal. Such a problem is called the "Sink Location Problem". It can be defined shortly as: "Given a group of sensors, whose locations, sensing and communication ranges, and capacities are known, finding the coordinates of the sink that maximizes the lifetime of the network during which a certain sensing criteria is always met."

This problem is somewhat similar to the "Facility Location Problem", given a group of demand nodes, a place for the source node is found such that the overall communication (or transportation) cost is minimized [3, 4]. Facility location problem specifies that the demand nodes can directly communicate with the source node (e.g., a customer can go directly to a supermarket). Because of this fact, sink location problem differs significantly from the facility location problem. In sensor networks, intermediate nodes lose energy due to data routing operations. This requirement adds one more level of complexity to the problem. Also, sink location should be chosen as safe as possible to decrease the risks associated with the nature of surveillance tasks. A safe location of the sink is preferably a protected and controlled environment to alleviate the risk of attacks on the sink itself.

Literature includes various sink location strategies. Stann and Heidemann propose to place the sinks by hand or only random choice [5]. Their optimization depends on the routing itself, rather than the sink placement, hence the sink coordinates are chosen by hand. Das and Dutta have chosen to place the sink on coordinates that are outcome of a uniform random distribution, which is similar to their choice of the sensor coordinate distribution [6]. Similar choice for the sink placement is made by Intanagonwiwat *et al.* They choose random locations for the sinks and try to optimize the number of sinks [7]. Random placement is well studied in the literature, as seen in the articles by Handsizki *et al.* and by Simon and Farrugia [8, 9].

Sink placement on the edges of the deployment region is another possible choice. Gnawali *et al.* use the corner locations as sink places for their simulations and experiments to find high quality data transmission paths [10]. The same choice of placement is also used by Yu *et al.* and Zhou and Krishnamachari [11, 12]. Li and Cassandras and Xing *et al.* have also chosen the edges, however they use the midpoint of the sides of the deployment region to place the sinks [13, 14]. In [15], authors place the sink at the edges of the network, with the main aim to decrease the energy expenditure.

Choosing the center of the deployment region is another popular choice, which has been employed by Cristescu *et al.*, Ganesan *et al.*, and Maleki and Pedram [16, 17, 18].

A somewhat similar choice is to partition the deployment region into clusters and place the sinks at the center of those clusters instead of the whole region. Such an approach increases the topology awareness of the network, such a scheme is employed by Oyman and Ersoy, Perillo and Heinzelman [19, 20].

Solis and Obraczka have tested a variety of sink placement strategies, placing the sink at corner, random and center coordinates. Then they evaluate the performance of different in–network aggregation algorithms in terms of the trade-offs between energy efficiency, data accuracy, and freshness [21].

A different and interesting approach for the sink placement is by Akkaya and Younis [22]. They choose to make the sink mobile, which differs significantly from the other approaches. Their aim is to obtain an efficient routing using a mobile sink. Mobility of sinks is also exploited by Ye *et al.* to increase the efficiency of the flooding based communication scheme [23]. Kim *et al.* use mobile sinks to receive the information from the nodes directly [24]. The nodes do not route the information, instead opt to wait for the sink to get inside the communication range and directly communicate. Mobility of the sink is also proposed in [25] and [26]. Yet it should be noted that mobility comes with the price of the energy loss caused by frequent network-wide broadcasting. Such broadcasting operation occurs since the network needs to send continuous notifications of sink coordinates to the nodes and the corresponding routing updates [27].

Yang and Lin try to decrease the load on the sink by adapting the high energy nodes around the sink as base stations and distribute the load between the proposed base stations [28]. Poe and Schmitt try to place the sink nodes using genetic algorithms, however their objective is based on the delay between the sink and the nodes with no energy optimization assumptions [29]. Integer programming methods are also used to optimize the sink location [30]. However as the problem size gets bigger, such methods become infeasible as the time they take to complete grows very fast too.

In this chapter, the aim is to find the optimal sink location based on the energy expenditure of the overall network. Optimizing the energy expenditure also increases

the overall network lifetime. Proposed method aims to place the node in a safe location, suitable for especially surveillance wireless sensor networks.

2.1. Sink Node Placement

Sink nodes have been previously defined as the data collection centers to which all the data from the sensors are routed. From a security point of view, the location of the sink should be preferably in a protected and controlled environment to alleviate the risk of attacks on the sink itself. On the other hand, the location should provide shorter routes for the sensor nodes and limit the energy due to data transfers as much as possible. These two constraints often contradict each other because safer locations offer longer routes and lead to more energy loss due to the routing operations. One more complication is the underlying physical terrain, which causes line of sight blockages between the elements of the network. Many approaches in the literature assume that the deployment field is flat and thus nodes are not blocked by the obstructions caused by the undulations of the terrain. For 2D regions, the communication between two sensors and the sensing of a point over the region by a sensor is directly related with the Euclidean distance over (x, y) coordinates. Same operations over 3D regions face another problem, the line-of-sight between two points. The elevation of points in the visual access path between the sensor locations can obstruct the actual communication or sensing path, as shown in Figure 2.1. The differing elevation characteristics of the terrain heavily alters the sensing and communication coverage of WSN. A realistic sink placement approach should take blockages and obstructions into account and aim to minimize the loss caused. Such a modification increases the complexity of the calculations, yet the results becomes much more realistic compared to scenarios with 2D assumptions. Figure 2.2 shows a sample terrain generated synthetically, with a total elevation difference of 50 m.

In WSNs lifetime can be defined in various ways, based on the target scenario on which the network will be applied to. Lifetime can be the time until the first node dies, the last node dies, a portion of sensors die, sensed area range falls below a threshold. We use the total sensing area range as a lifetime limiting metric. From the application

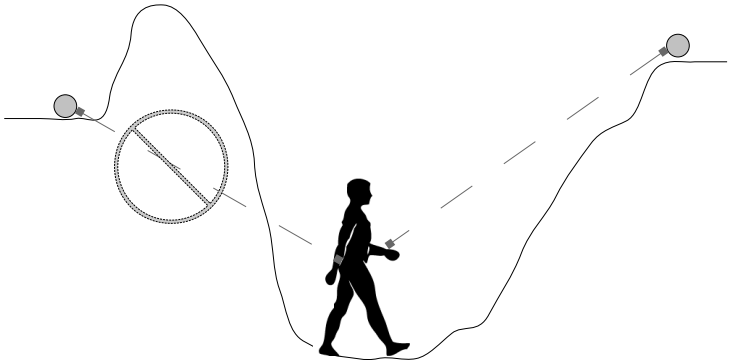


Figure 2.1. Surveillance line of sight blockage due to terrain obstructions.

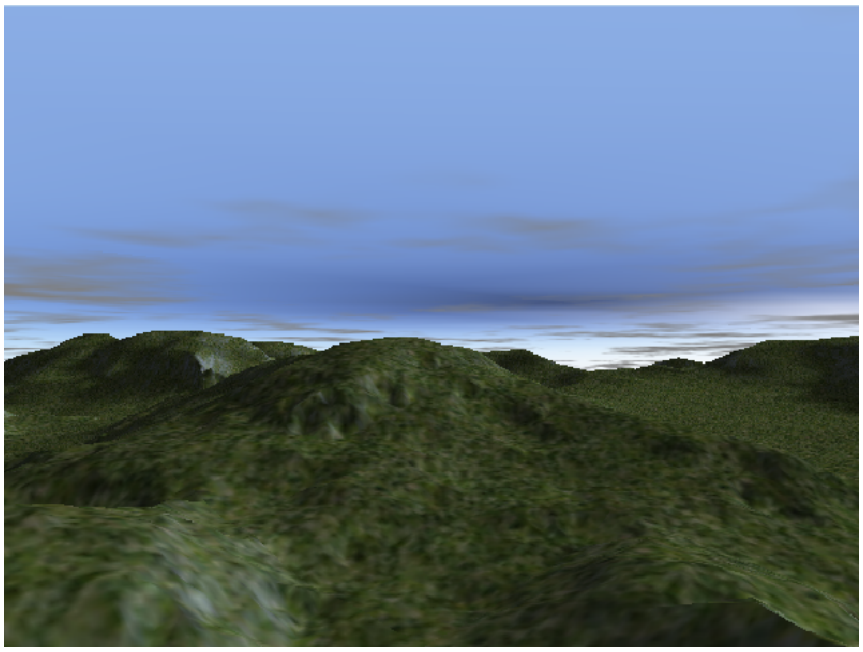


Figure 2.2. A synthetically generated terrain sample.

point of view, death of nodes are not as important as loss of sensing capability over a certain area, due to the fact that WSN provides redundant sensors with whom network can tolerate node losses. In this work, if the sensing coverage over the monitored area falls below a specified percentage of the initial area, then the network is pronounced as dead. This parameter, namely the coverage requirement value, must be chosen according to the testing scenario. Higher values give better sensing coverage, yet limits the lifetime, whereas lower values with longer lifetime values may cause sensing quality problems for especially surveillance and safety oriented networks.

To find the suitable coordinates for the sink in terms of both lifetime and safety, a hybrid version of Genetic Algorithms (GA) are applied. GA is chosen as it provides abstraction from the problem defined and is able to explore the search space in a very good manner. The algorithm is called GAUSS, Genetic Algorithm Using Sensing Simulation. GAs are adaptive heuristic algorithms with ideas mainly based on natural selection of individuals. They explore the search space of an optimization problem in a semi-randomized manner. Algorithms try to locate regions in search space that give better performance, using the information gained from previous steps. The techniques used by the algorithms imitate the selection process of the nature, which results in "survival of the fittest". GAs were initially introduced in 1960s and were primarily used in Artificial Intelligence area. Later this approach has been applied to problems in combinatorial optimization with some very successful results [31, 32].

According to Goldberg, GAs differ from other optimization methods in four ways: [33]

- GAs work with problem coding instead of actual variables, hence they are problem independent.
- The search in the GAs is done between a variety of solutions. Thus, they are less likely to stuck in a local optimum.
- GAs are guided only by the fitness score, rather then problem terrain specific values. GA steps have little information about the problem itself. Such a property makes them applicable to a variety of solutions.

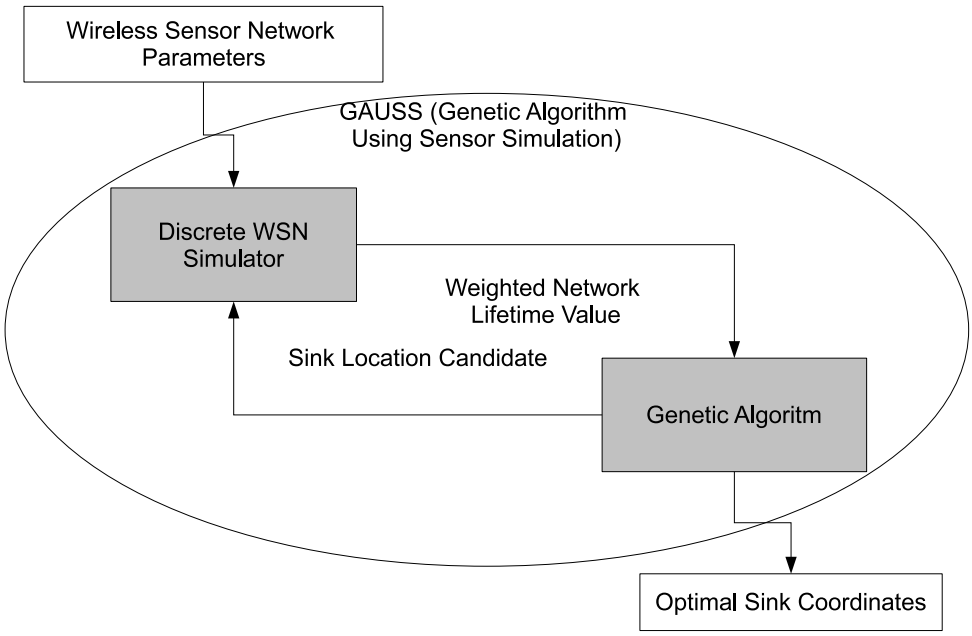


Figure 2.3. Basic scheme of the proposed approach.

- During the search, GAs use stochastic rules rather than deterministic ones.

The proposed approach can be named as a "simulator-in-the-loop" type heuristic. In such an approach, the simulator and the optimizer interact to find the optimal value for the given problem as seen in Figure 2.3. It may be possible to use LP techniques to find the values. However, such techniques can work for small problems and they usually use first node death which is not realistic. GAUSS assumes big networks distributed over large deployment fields. Such a version is very hard to solve using formal optimization methods due to the field size and enormous number of sensor operations. The method tries to simulate the network communication, which, in the case of formal optimization, makes the problem prohibitive in terms of computation costs. A more realistic lifetime metric which includes the effect of networking layers can be used with simulator-in-the-loop. In our approach, the simulator is a discrete event simulator and the optimizer is the GA heuristics. Terrain properties have drastic effects on the performance of the sensor networks. Likewise, the safety restrictions mandate further modifications to the basic approach. To build the routing tree inside the network, line

of sight between the node pairs are calculated. Using those calculations, the neighborhood set of each node is narrowed down to the ones that are directly reachable both in terms of energy limitations and terrain obstructions. Moreover, some portions of the field, although they offer longer lifetime coordinate candidates, are prohibited by the low safety of that location. This problems is solved by the "safety weight" parameter, which assigns different weights to the candidate coordinates according to their distance to the safe side; closer coordinates have higher weights. Safety weight parameter signifies the trade-off between safety and longer lifetime.

2.1.1. Coding

The coding used in the algorithm has a cardinality defined by the length and the width of the area to be covered and sensed, to cover the search space properly. To represent a possible solution, which is a tuple of X and Y coordinates, there are (width×height) candidates for each value in the chromosome. The cardinality of the coding is equal to log_{10}(width×height), in order to cover the whole space.

To represent the two dimensional coordinate tuple, a combing method is used as shown in Figure 2.4. To prevent biases to specific parts of the terrain, X and Y tuples are not combined as XY. While forming the chromosome, the binary values are interleaved. For our purposes binary values present enough details about the coordinates over the terrain. This way, crossover operation is able to preserve a portion of locality values from both X and Y in the new offspring.

2.1.2. Genome Operators

When the algorithm starts, it randomly creates a population with a predefined size. The population size does not change till the algorithm finishes. The crossover method employed is single point sexual crossover. Moreover, single point mutation is employed to increase the diversity within the population.

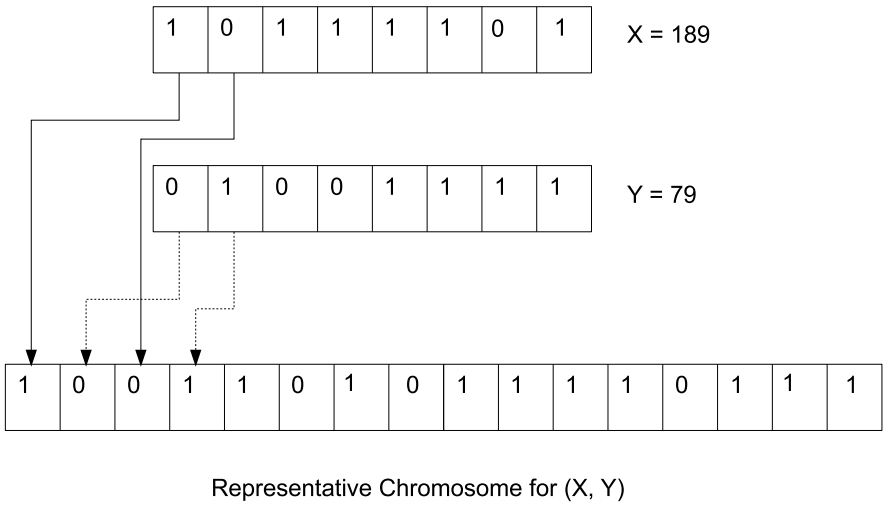


Figure 2.4. A sample for the coding scheme.

2.1.3. Selection

In order to select genomes from the population for mating, the scheme employed is roulette wheel selection. The probability of being selected is directly based on the magnitude of the fitness score. Higher scores increase the likelihood of the individual for selection. The probability is equal to the fitness score of the individual divided by the sum of fitness of the total mating pool. The fitness scaling method used is linear.

2.1.4. Fitness Function

The most important part of the GAs is the fitness function employed. Choosing a suitable one is very crucial to locate the optimal point of the search space. For the sink placement problem, it is difficult to propose a closed form function that can produce a fitness value (lifetime) for the given coordinate tuple. In order to perform this task, a discrete event simulation is embedded for calculating the fitness value of an individual.

The fitness function takes the following parameters as input:

- The candidate coordinate tuple for the sink location.
- The sensor network topology, the location of the sensors.
- The terrain elevation map.
- Sensor communication capabilities and ranges.
- Initial criteria for field coverage.

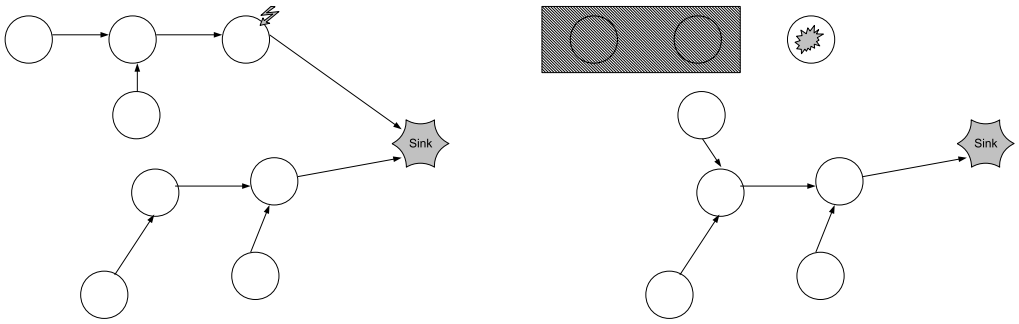
Given those parameters, the simulation continues until the network coverage cannot be met. The time that passes from the beginning till the simulation ending time gives the network lifetime. This network lifetime is the fitness score for the candidate tuple.

During the network simulation, periodic sensing is performed as a data gathering network. During each period, each node performs sensing operation and sends the data to sink node. However, since the communication range for nodes is limited and there are obstructions due to the elevation differences in the deployment area, intermediate nodes are used to relay the information en route to the sink. After the sensing and data gathering operations are completed, the coverage of the network is calculated. It should be noted that, the sensing network is basically a "sink–centered" communication tree. When the total coverage ratio of the network goes below a certain value, the network is effectively dead. The value is the ratio of the coverage area of sensors at that time to the initial area. This point in time of death of the network. Setting this coverage requirement ratio to one means that the network dies when any part of the area which was initially covered is no longer reachable for the sink. One important note is that when an intermediate node dies, the "disconnected" nodes are also considered dead if they cannot find new routes to the sink, even if they have remaining battery, as shown in Figure 2.5.

The pseudo–code for the hybrid GA is given in Table 2.1 [34]. The algorithm begins with selecting a random generation, where each member of population shows a candidate location for the sink. At each generation, crossover and mutation operations

Table 2.1. Hybrid genetic algorithm, GAUSS, steps.

```
randomly place the sensors over the terrain
produce initial population, compute the fitness of each individual
repeat
  repeat
    select two parents for mating with bias towards the fitter individuals
    apply crossover to parents to produce offspring
    apply mutation to the offspring
    compute the fitness of the two offsprings, start the simulation
    loop
      sink is placed in the coordinates specified by the offspring
      calculate the initial coverage of the network
      for all sensors do
        sense the area periodically
        find the reachable nodes within line of sight
        reach the minimum energy spending distance node and send the sensing
        information
        update energy levels
      end for
      calculate new coverage
      if coverage below the coverage requirement value then
        end the simulation and return the total operation time
      else
        continue operation
      end if
    end loop
    place offspring inside the new generation population
  until new generation is full
  new generation ← current generation
until generation is converged
```



(a) Initial network, one sensor dies.

(b) Network with disconnected nodes.

Figure 2.5. Sample network, where the death of an intermediate node disconnects the offspring alive nodes.

are applied. The results, or the offsprings, are submitted to the discrete event simulator to calculate the corresponding fitness score.

For the line of sight calculations inside the simulation, small sized antenna for a sensor node and an average height trespasser are assumed. Since the distance between the sensor and the intruder cannot be more than the sensing range, the earth curvature is assumed to have no effect on the line of sight. The obstruction is caused by the actual elevation differences on the deployment terrain.

2.2. Results

The base values for the parameters of the genetic algorithm are given in Table 2.2. In order to see the effects the generation count, mutation probability and population size are varied. Based on the results, the basic parameters are selected empirically. All of the results presented are taken from the mean of 40 runs with different seeds. Lifetime results for GAUSS values are compared to two heuristics. First heuristic places the sink in the center-of-mass of the X and Y coordinates of the sensors, named as COM deployment. This heuristic aims to minimize the distances between nodes and the sink. On the other hand, second heuristic aims to increase the safety of the sink node by placing it on the secure border. Named as COM deployment at border, it places the sink in the center-of-mass of the X coordinates of the sensors. Y coordinate

Table 2.2. Base genetic algorithm test parameters.

Chromosome Encoding	Binary
Mutation Probability	0.025, ∗0.05, 0.075, 0.01
Crossover Probability	1
Crossover Method	1 Point
Population Size	50,∗100, 150, 200
Generation Count	50, 75, ∗100, 125, 150
Selection	Steady State

∗ These are the default values used in the simulations.

is the secure border.

The lifetime value is the "safety weighted" lifetime defined earlier. In order to emphasize this fact, it is presented as "Nonvulnerable Operation Period Lifetime Value". Such a lifetime value can be interpreted as the ability of the sink to survive against an attack that can originate from the insecure side of the network. The values presented are for very low battery values for practical purposes. Sample simulations have shown that when high battery values are used the results scale up in a similar manner. All the lifetime values are given in minutes.

Discrete event simulation parameters are given in Table 2.3. Values for area shape, sensing and maximum communication range of the sensor, the sensor count deployed in the network and the simulation coverage requirement value, i.e. the required percentage of the network disconnected when network is pronounced as dead, are tested.

Table 2.3. GAUSS simulation test parameters.

Area Shape (width×height)	200×900, 300×600, 450×400, 600×300,∗900×200, 1000×180
Sensing Range (m)	15, ∗20, 25, 30, 35
Node Count	300, 350, ∗400, 450, 500, 550, 600
Coverage Requirement Value	0.7, 0.75, 0.8, 0.85, ∗0.9, 0.95
Communication Range (m)	40, 60, ∗80, 100, 120
Path Loss Exponent	4
Sensor Antenna Height (cm)	10
Average Intruder Height (cm)	150

∗ These are the default values used in the simulations.

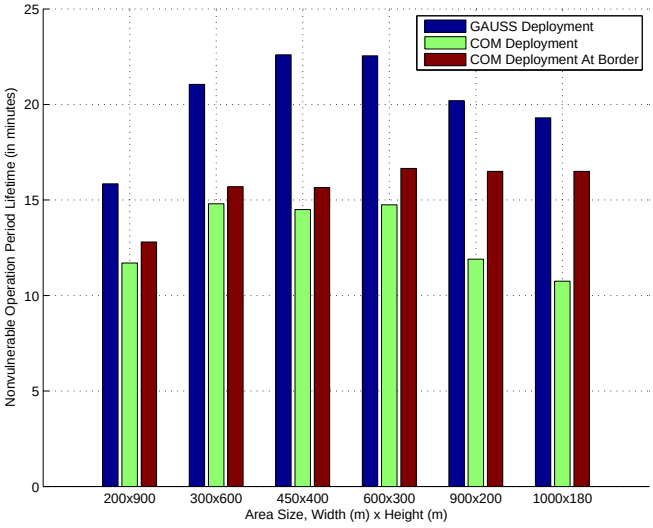


Figure 2.6. Values for different area shapes.

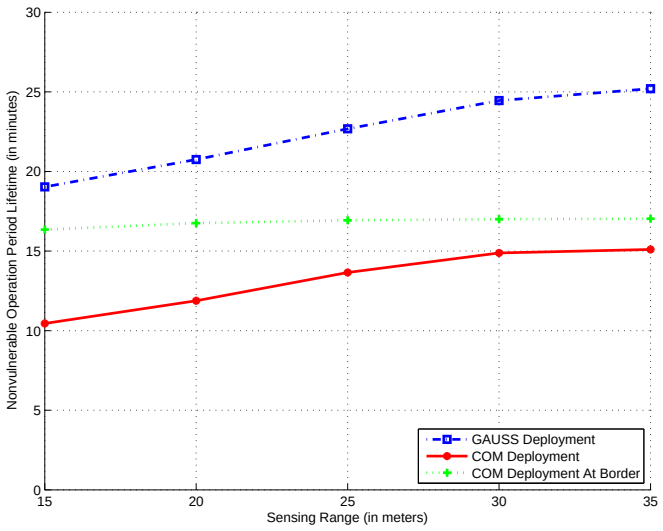


Figure 2.7. Values for different sensing range values.

2.2.1. Results for Flat World Scenarios

Different area shapes with the same total area size are tested to understand the effects on the network lifetime. The results are presented in Figure 2.6. It can be seen that as the network gets thinner, the lifetime values start decreasing. This can be explained by the longer paths to sink and the formation of bottleneck nodes. Such nodes do not have routing alternatives and stand as the only option when acting as intermediate routing points. When the deployment area is relatively larger, the lifetime gain by the algorithm is around 25 per cent. Moreover, since the network gets thinner, sink locations get nearer to the border, which decreases the lifetime value margin between GAUSS and COM deployment at border.

Figure 2.7 shows the lifetime values for different sensing ranges of the sensors. GAUSS performs better as the sensing range value increases. It should be noted that, as the sensing range increases, sensors create more redundancy to cover the deployment region. A direct result is the ability of the network to survive when intermediate nodes start dying, since there is better chance for a nearby alive sensor node to cover those regions. GAUSS performs much better than the COM deployment and COM deployment at the border heuristics.

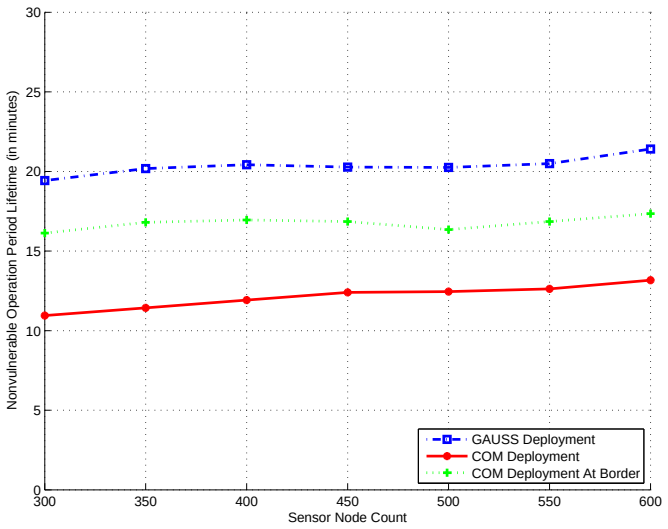


Figure 2.8. Values for different sensor node count values.

Number of sensors deployed in the region are also varied and the results are presented in Figure 2.8. Results indicate that there is a stable lifetime gain using the presented heuristic compared to the other alternatives. We have seen that the lifetime gain by employing more sensors is very marginal after a threshold value. Such a phenomenon is a result of the bottleneck node occurrences that constrain the lifetime of the network. Bottleneck nodes carry most of the traffic in a particular region of the network. High traffic patterns kill those nodes early and disconnect the network bringing the end. For such cases, increasing the network sensor count does not necessarily increase lifetime. Bottleneck availability voids the assumed increase in the lifetime due to new sensors. Moreover, if the sensors are deployed "behind" the bottleneck nodes, they may further increase the traffic load on them. The result is earlier death and decreased network lifetime. Such facts must be carefully analyzed before increasing the node count in network.

Results with different coverage requirement values are tested and are given in Figure 2.9. It is clear that the most important parameter is the coverage requirement value. High values increases the sensing reliability of the network, however severely decreases the lifetime. With high values, the overall network is highly sensitive to particular sensor losses and does not present overall operational redundancy. Such

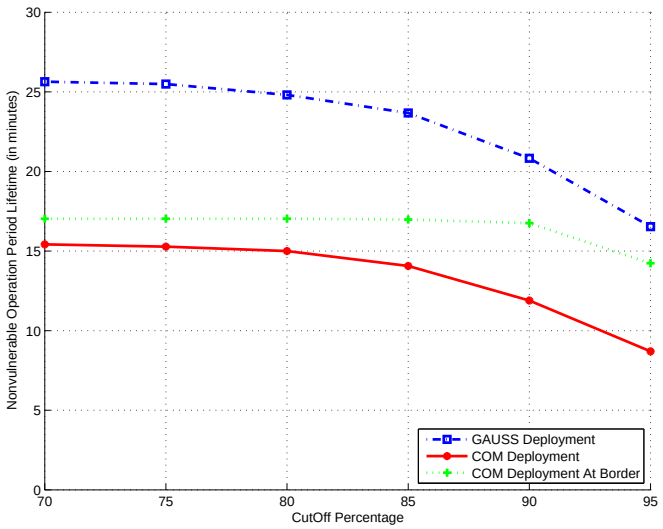


Figure 2.9. Values for different coverage requirement values.

an important parameter must be tuned according to the specific requirement of each deployment. However, the results indicate the possibility of further optimization.

Higher maximum communication range of the sensor decreases the possibility of occurrence of bottleneck nodes by reaching out to more sensors while routing. Results for different communication range values are presented in Figure 2.10. However, when this parameter is increased it causes higher battery depletions as the energy spent while communication is related to the distance between communicating nodes. And the results show that a certain threshold value is sufficient to find an optimal lifetime value. In our cases, 80 m is observed to be the sufficient range value.

2.2.2. Results for Three Dimensional Scenarios

Figure 2.11 shows the sample terrain used to test GAUSS with three dimensional deployment fields. Up to 100 m elevation differences are possible over the given terrain and local elevation differences and obstructions do cause line of sight blockages between sensors.

Figure 2.12 shows the values for different area shapes where the total area size

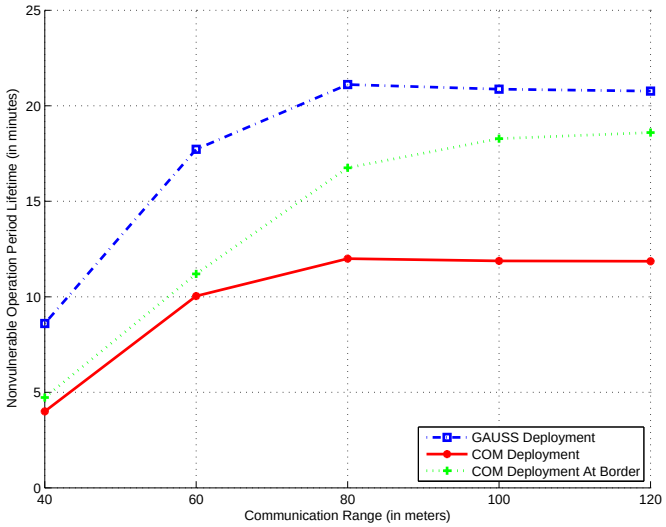


Figure 2.10. Values for different maximum communication range values.

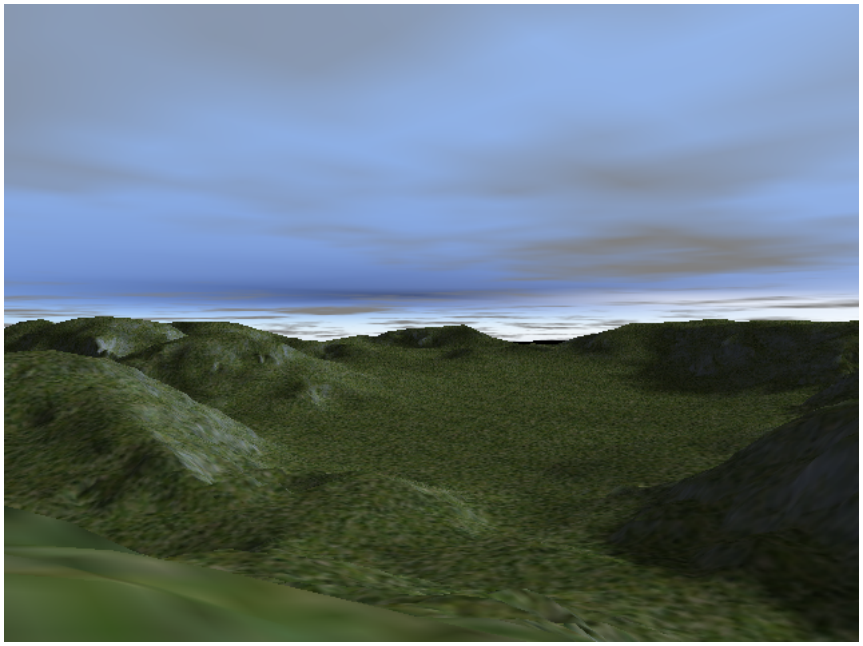


Figure 2.11. Synthetically created terrain used in the simulations.

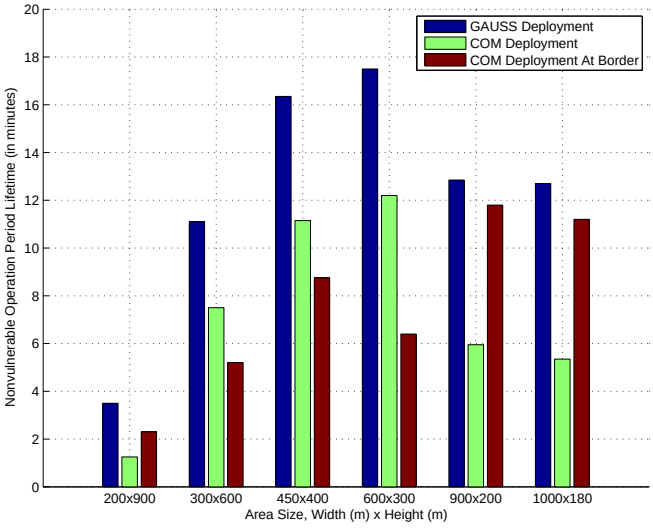


Figure 2.12. Values for different area types on the three dimensional terrain.

is kept constant. As expected the lifetime gets better as the area shape becomes more like a square, therefore decrease the mean distance to the sensor nodes. For areas that have very high width, lifetime values are severely limited because of the increased distance between sink and nodes and less number of routing alternatives. Yet it should be noted that area shape is generally imposed by the environmental factors and is not possible to modify.

Figure 2.13 shows the values for different communication ranges over the given three dimensional terrain. The figure shows that small communication ranges cause very small lifetime values. Small communication range and the terrain obstruction results in very few reachable neighbors for sensors and the death of a couple of nodes disconnect a part of the network and end the network lifetime. The results for GAUSS show up to 20 per cent lifetime increase compared to COM deployment at border and up to 120 per cent compared to the COM deployment.

Our tests with different sensor counts are shown in Figure 2.14. A steady increase in the lifetime is possible as the number of sensors increases. However, doubling the node count provides only 50 per cent lifetime increase, which may bring the cost issues into mind. GAUSS provides 10 per cent lifetime gains compared to the COM

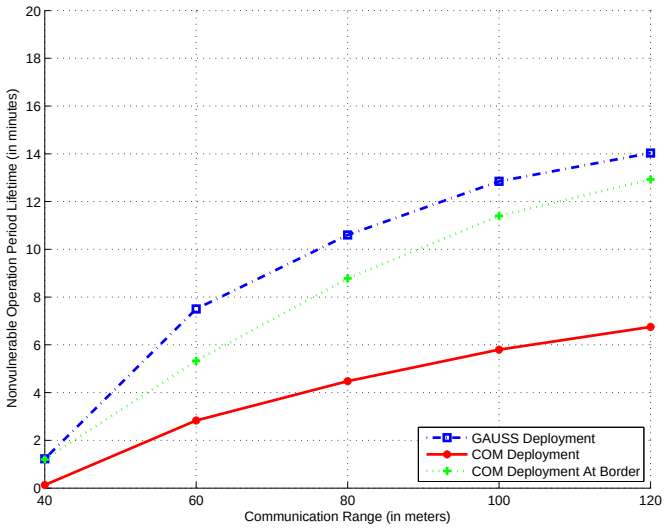


Figure 2.13. Values for different communication ranges on the three dimensional terrain.

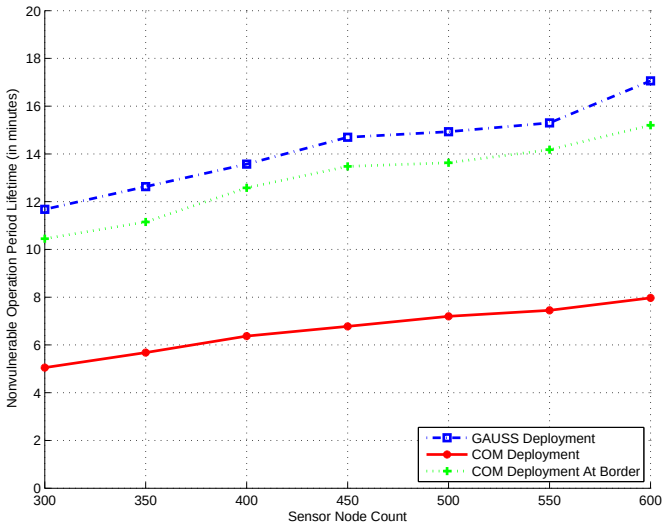


Figure 2.14. Values for different number of sensors on the three dimensional terrain.

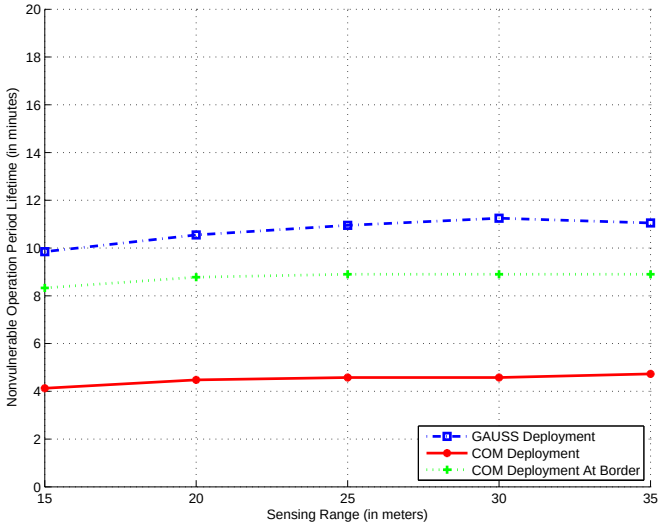


Figure 2.15. Values for different sensing ranges on the three dimensional terrain.

deployment at border heuristic.

Figure 2.15 shows the nonvulnerable operation period lifetime values for different sensing range of the deployed sensors. In our cases, the effects of the increased sensing range is weakened by the terrain obstruction. Our results showed that the sensing range is not effective in the given terrain types. Using the proposed GAUSS method, it is possible to have 20 per cent lifetime increases on the average compared to the COM deployment at border and more than 100 per cent lifetime gains compared to the COM deployment.

The most effective parameter in the three dimensional version is the coverage requirement value. Border surveillance type applications are required to provide a very high level of operation quality. In border surveillance wireless sensor networks, operation quality consists of the overall sensing and reporting quality of the network. The coverage requirement value sets the minimum acceptable level of quality. If it is set too high, for an especially sensing environments with many obstructions, a five per cent increase in the requirement may cause up to 35 per cent lifetime decreases. The results of tests on coverage requirement value over the test terrain is presented in Figure 2.16. On the average, around 25 per cent lifetime gains are possible using

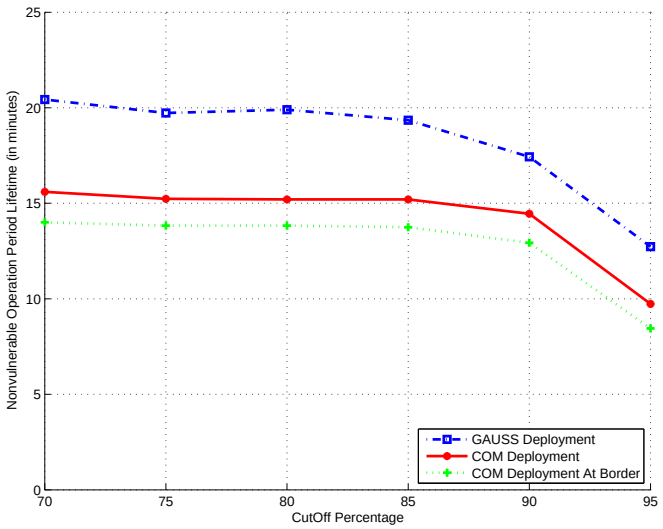


Figure 2.16. Values for different coverage requirement values on the three dimensional terrain.

GAUSS, compared to COM deployment at border. However, a coverage requirement value increase from 90 per cent to 95 per cent drastically decreases the lifetime values for all of the tested heuristics. The value should be chosen to suit the specific needs of the underlying scenario and the available resources that is to be used in the network. In order to understand the overall performance of the GAUSS, we also performed exhaustive search on small sized problems with different parameters to compare against a typical GAUSS run. The results showed that, for the cases studies, GAUSS was able to find the best lifetime value candidate, confirmed by the exhaustive search results for different sets.

3. IDENTIFICATION OF BOTTLENECK AREAS FOR PARTIAL REDEPLOYMENT

Nonuniform deployment of sensors over a terrain may be the result of the underlying physical constraints, or changing sensing or communication requirements over the terrain. For such nonuniform deployments, the communication traffic follows an uneven pattern, creating considerably high loads on some particular nodes. The areas that include highly loaded nodes become the bottleneck areas soon after the network becomes operational, decreasing the lifetime of the network. In the previous chapter, we proposed a method to place the sink in a relatively energy optimal location. Yet, even the best location can offer a limited lifetime gain due to those bottleneck nodes. Prolonging the lifetime of the overall network is possible by identifying the bottleneck areas at the earliest stage possible and deploying new sensors over those areas.

Our approach to locate the bottleneck nodes and the regions is to decompose the network into subnetworks. The number of the subnetworks formed is not fixed and decided by the algorithm adaptively based on the network topology and some other parameters. Clustering approaches can be employed to perform the decomposition of the network to smaller subnetworks [1],[35]–[42]. Once the clustering is completed, the nodes that form the border between the subnetworks are the bottleneck candidate nodes. However, most clustering algorithms take mainly the intra–cluster node distances into consideration, therefore the perimeter nodes have little effect on the overall cluster formation. In the proposed method, the choice is to incorporate the perimeter nodes heavily into the cluster formation steps.

Our approach is based on breadth–first search starting from a chosen starting node and enlarging the cluster using the nearest unclustered nodes. Such a ring enlargement approach for clustering can also be seen in [43]. While designing the algorithm, observations from previous simulations performed on sensor networks are used, which are given below:

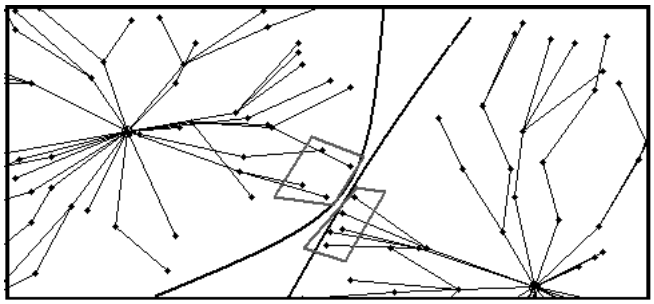


Figure 3.1. Border nodes that have high traffic nodes.

- Clusters that are rather far away from the network tend to become more isolated and contain bottleneck nodes.
- Clusters must be compact in terms of intra–cluster node distance and node neighborhood values.
- Once clusters are formed, the traffic passing over the uncovered areas between the clusters cause high loads on the nodes that stand at the perimeter of the clusters, an example is shown Figure 3.1. The load on the nodes inside the polygons is much higher than the other nodes in the clusters formed.

The algorithm progressively tries to form compact clusters whose borders do not intersect. Clustering is based on a geometric approach, since priority is given to nodes that are nearer to the cluster center of mass. The following decisions are the key issues for the algorithm.

- Start clustering from nodes that have the maximum distance to the center of mass of the network.
- Nodes that are candidates to be added to the cluster, must have at least "minimum–connection–count" number of connections to the nodes inside the cluster.
- Candidate nodes must be within the "link–limit" distance to the neighboring cluster nodes (the cluster nodes that can actually communicate with this node).
- Clusters should choose the next node, that will be added, according to its distance to the cluster center–of-mass, nearer ones are added first.

The decisions are set to form clusters that suit our initial observations. The starting node is not the cluster head, but serves as the first step of the algorithm which results in formation of separated clusters. The second decision is valuable for increasing the number of intra–cluster connections as much as possible. Combined with the third decision, they provide more compact clusters in terms of both distance and connection count between the cluster elements. The fourth decision parameter tries to form the perimeter of the cluster at each step progressively, by employing a breadth first approach. The algorithm avoids going deep into the network and tries to do a ring enlargement approach, with the aim of keeping the cluster diameter short.

The pseudo–code of finding the bottleneck area identification algorithm (BAIA) is given in Table 3.1. Initially, location of the network center (center–of–mass) is found. The nodes that is farthest away from this location is the starting point for BAIA. Starting with the farthest node, the cluster embarks on the enlargement process. Our enlargement method prefers keeping the cluster diameter short, by trying to find the nodes that have as many connections to the already clustered nodes as possible. Moreover, a lower bound is set to the number of connections that cluster candidate nodes must have to the nodes inside the cluster. The lower bound helps avoiding the problems with nodes that have small distance to the cluster center, but do not have enough connections. It should be noted that the nodes with insufficient connections are our future bottleneck node candidates, which should not be included in the middle sections of the cluster.

When the clusters are formed, BAIA forms the convex polygons surrounding the clusters. If those polygons intersect the clusters are said to be ineffective to locate the redeployment areas. We discard the already formed polygons, decrease the link–limit parameter and the algorithm starts over again. Decreasing the link–limit leads to new clusters which enforce smaller intra cluster distances and results in formation of geometrically smaller clusters. This operation continues until the cluster related polygons do not intersect. It is also important to note that, at this step of the algorithm, convex polygons serves as a basis to check whether the clusters are separated or not.

Table 3.1. BAIA steps.

```
1: link–limit ← communication range
2: set minimum–connection–count
3: repeat
4:    while there are unclustered nodes do
5:       find the center of mass for the unclustered nodes
6:       find the node that is farthest away from the center–of–mass
7:       set this node as the starting node
8:       add starting node to the queue
9:       while queue has nodes do
10:         take out the next node from the queue
11:         add the node to the cluster
12:         find its free neighbors within link–limit and having minimum–connection–
            count connections to the cluster nodes
13:         sort the neighbors according to their Euclidian distances to the cluster
            center–of–mass
14:         add the neighbors to the queue in sorted order
15:      end while
16:   end while
17:   for all clusters do
18:      find the perimeter nodes
19:      form perimeter convex polygon for the cluster
20:   end for
21:   if the clusters NOT intersect then
22:      clusters are formed
23:   else
24:      decrease link–limit
25:   end if
26: until clusters are formed
```

When BAIA stops, there are different number of clusters, whose inter–cluster areas are the redeployment region candidates. Redeployment over those areas tend to increase the uniformity of the node distribution. The formation of bottleneck and critical nodes is a direct result of the topological uniformity/nonuniformity of the network. For nonuniform topologies, critical nodes appear at the earliest steps. Redeployment at border areas is a heuristic to reform the network as a rather more uniform one, hence the risk of critical node formation is postponed to the later periods of the network life. Polygon intersection heuristic is a simple and effective way to locate the border areas. The results presented in the next section show that the heuristic performs well for different scenarios. BAIA assures that when the algorithm stops, the bottleneck regions are left between the polygons. This way, the initial clusters are included inside the final ones. At first glance, such a decision may seem overkill, however it has the advantage that a relatively larger area of nodes become neighbors with redundant nodes hence bottleneck formation risk in that area is discarded. For other purposes, bottleneck region can be reduced using a second pass of the algorithm for only that region.

For BAIA, the main design idea is to create an algorithm able to work with different communication schemes. In order to develop such an independent approach, routing operations to the sink are not taken into consideration. BAIA assumes that bottleneck regions always require sensors to cover the respective area at least in terms of sensing, whether they are close to the sink or not. Moreover, since the general traffic pattern that directs the communication to the sink, ideally sink is placed in a sensor crowded environment. Such decisions do not affect, or mitigate bottleneck formations. Under those circumstances, BAIA does not make any assumptions about the sink and the communication traffic.

3.1. Experimental Results

BAIA is tested using scenarios with nonuniformly placed sensors. For each scenario, the network is divided into different sized rectangles, as seen in Figure 3.2. This approach models the idea of simulating a region spanning different farm areas. Each

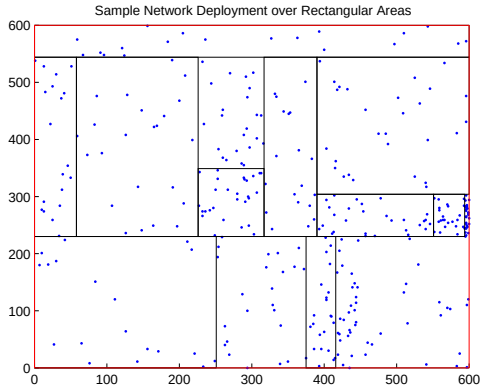


Figure 3.2. A sample scenario showing the sensors deployed over different sized rectangular areas of the network.

rectangle is randomly assigned to one priority level of three levels. The priority level sets the number of nodes that the particular rectangular area receives. A farmer may be willing to receive a higher level quality service (with more number of sensors) and, thus, is ready to pay more, whereas another one is content with inferior services (with less number of sensors) for less price. This tradeoff between quality and price is modeled with the priority values. In such a scenario by having different density sub–regions, our aim is to create an overall distribution that does not have a uniformity property. By forming networks with nonuniformly placed sensors, where many bottleneck nodes and isolated areas exist, the aim is to create scenarios where nonuniformity becomes the most important lifetime limiting factor.

The basic experiment parameters are the sensing range, communication range and the path loss exponent, which are set to have realistic values as listed in Table 3.2. However, the initial battery energies are kept at comparatively very low values to keep the simulations short. Since the sensing operation is done periodically, starting with low battery energy levels does not affect the relative lifetime measurements other than scaling them down. In order to demonstrate the amount of scaling, 10 scenarios with initial 0.1J and 0.01J sensor energy levels are used. It is observed that the lifetime of each network also scales down with 1/10 ratio.

Table 3.2. BAIA test parameters.

Maximum Communication Range (m)	60, 80
Maximum Sensing Range (m)	20
Path Loss Exponent	2, 4
Sensing	Periodic
Initial Energy	0.01 J
Simulation Termination	Coverage below 70, 90 per cent of initial area
Area Size (m^2)	400×400, 500×500, 600×600
Initial Network Node Count	250, 300, 350 or 300, 400, 500
Redeployment Percentage	10
Packet Size (bytes)	512, 1024
Sink Location	Network center, GAUSS

The sensor network simulator counts the number of sensing operations done in the network until the overall covered area falls below a certain percentage of the initially covered area. The nodes that can perform sensing must also be able to reach the sink. While calculating the coverage percentage, areas which are covered by nodes that are still alive but not able to send the sensing information to sink are not taken into consideration.

The packet size parameter is related to the energy expenditure in the nodes. The energy term is divided into two sections, the energy lost in the node electronics for sensing and processing operations and the energy lost for sending the prepared packets over the wireless channel. Changing packet sizes changes the energy expenditure. A similar parameter is the path loss exponent, which is related to the energy loss due to the transmission medium.

To analyze the effect of different sink locations, two different sink location meth-

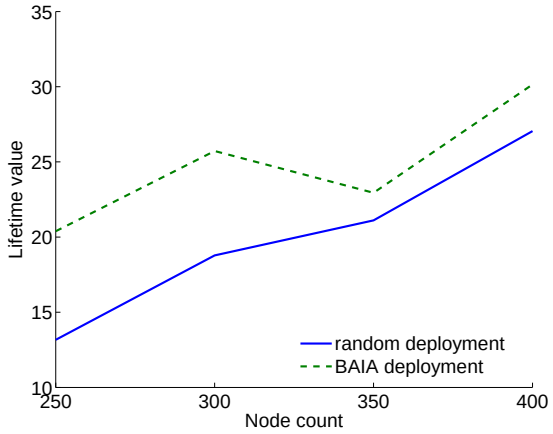
ods are employed. One method is to choose the center of mass of the network as the sink. The other method is the GAUSS presented previously.

Since our traffic is periodic, lifetime in simulations is given in terms of the number of sensing operations. The results can be easily converted to time by multiplying with the interval between two sensing operations. Also, the values given are the average of 10 different scenarios, where for each of the 10 scenarios the rectangle sizes, locations and priorities differ. The number of rectangles in a network is set to 15. For each scenario, two deployment approaches are compared. In one approach, the 10 per cent deployment is done randomly over the area, bottleneck location and deployment using our algorithm is the second approach. The overall lifetime values for two cases are compared in the results.

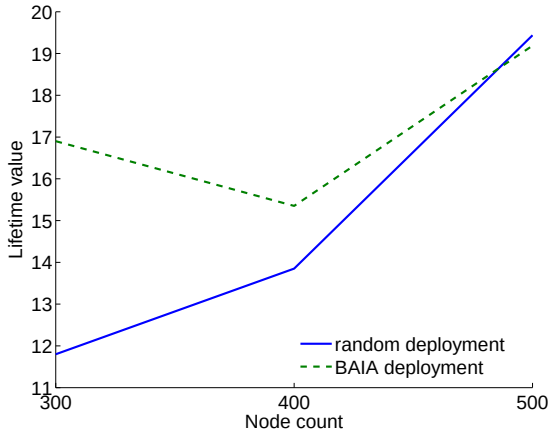
Figure 3.3 shows the results for 10 different scenarios over area sizes 400×400 m^2, 500×500 m^2 and 600×600 m^2. Lifetime values tend to increase with increasing node count and decreasing area size, as expected. However, the nonuniformity of the deployment can decrease the lifetime values, as seen in Figures 3.3a and 3.3b, where the node count is increased from 350 to 400 but the rectangular area differences cause lifetime value loss. The test cases prepared for the BAIA introduces high level of nonuniformity for the network. The rectangle parameters are kept constant for different sensor counts. Keeping rectangle parameters as same leads to bottleneck area formation again around similar regions. For such regions, increasing the node count deploys more sensors inside the rectangles, yet does not affect the bottlenecks i.e. the borders between rectangles. Hence although the node count increases, the bottleneck regions stay the same, which leads to earlier network death.

Another observation is the increasing effect of the nonuniformity with the increasing area size. Increase in lifetime values are rather monotonic for the area of 600×600 m^2 as seen in Figure 3.3c. For these set of scenarios, the average lifetime increase on the network by using our algorithm is 24 per cent.

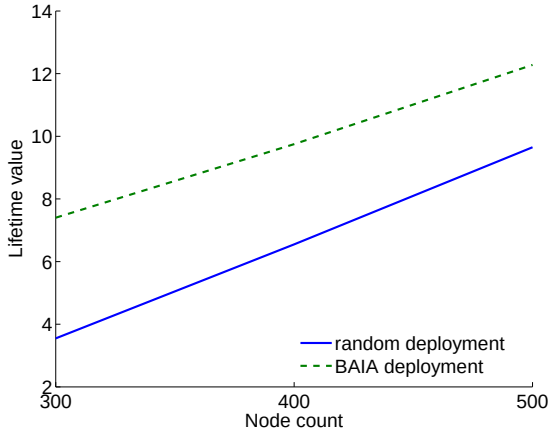
When restricting the coverage requirement value to a higher number of 0.9, the



(a) Area size of 400×400 m^2.

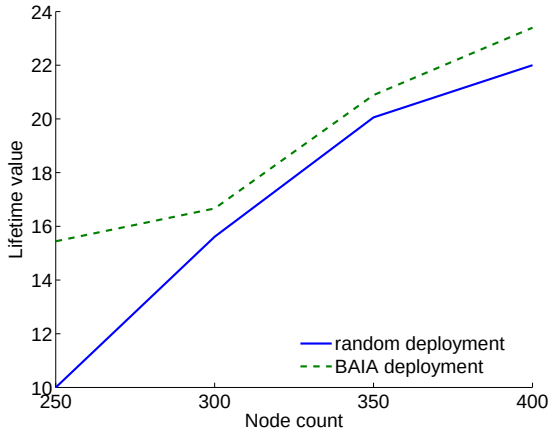


(b) Area size of 500×500 m^2.

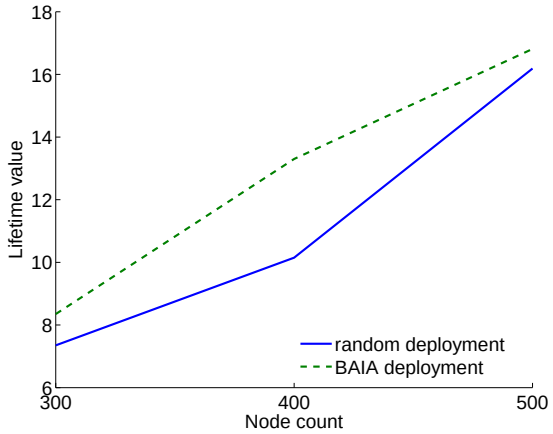


(c) Area size of 600×600 m^2.

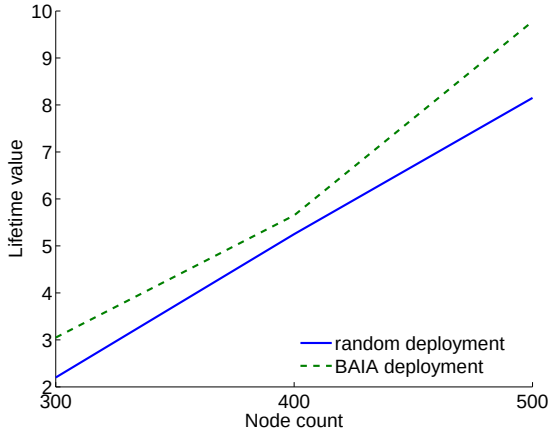
Figure 3.3. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.7.



(a) Area size of 400×400 m^2.



(b) Area size of 500×500 m^2.



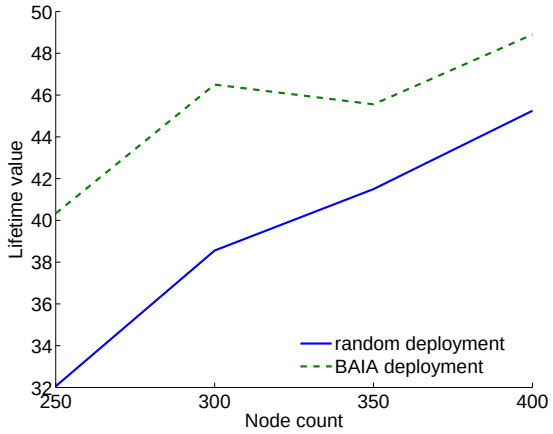
(c) Area size of 600×600 m^2.

Figure 3.4. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.9.

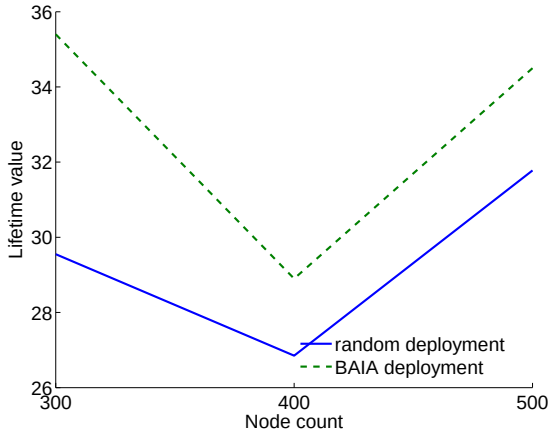
lifetime values for all scenarios fall by a ratio of 30 per cent, as seen in Figure 3.4. The effects of tighter restrictions are more drastic in sparser networks like 250 node deployment over 600×600 m^2, where the network is barely able to meet the initial coverage requirements. For this set, the algorithm increases the lifetime by 20 per cent on the average. The results indicate that the algorithm is able to perform properly under tight constraints too and increase the overall network lifetime.

In order to see the performance of the algorithm over changing communication ranges, the algorithm is tested with a communication range of 80 m. Compared to the 60 m scenarios, the lifetime values are higher by a ratio of 84 per cent. Such an increase is due to the decrease of the path length/hop count. With smaller hop counts, the energy loss on the intermediate nodes decrease, which in turn prolongs the overall network life. The algorithm increases the network lifetime by 16 per cent when coverage requirement value is set to 0.7, see Figure 3.5 and 11 per cent when coverage requirement value is set to 0.9, see Figure 3.6. The lifetime values are still directly affected by the rectangular areas and the nonuniform deployment, since the lifetime values are still not monotonic. The deployments over 400×400 m^2 as seen in Figures 3.5a and 3.6a show smoother lifetime changes, as a results of high density deployments, masking the effects of nonuniformity and bottleneck node formation to a certain degree.

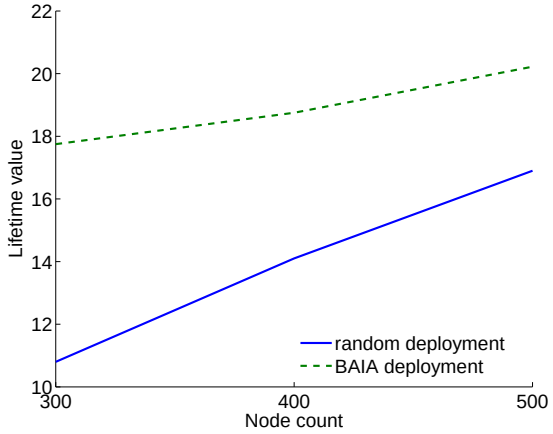
To see the performance of the proposed algorithm for multi-path fading environments, the algorithm is tested with path loss exponent set to four. The values are shown in Figure 3.7 and Figure 3.8. As expected, the overall lifetime values show a decline of 10 per cent on the average. This relatively small percentage value is a result of the low communication range of 60 m, which limits the loss due to the path loss exponent. However, as the values are increased to 80 m or 100 m, the relative lifetime loss increases to 30 per cent and 46 per cent respectively, as higher communication ranges increases the energy loss due to the environment and related the path loss exponent. Our algorithm, for this case, provides an average lifetime increase of 22 per cent. It should be noted that, the lifetime value changes for path loss exponent values two and four look similar. This shows us that the bottleneck formation due to the nonuniform



(a) Area size of 400×400 m^2.

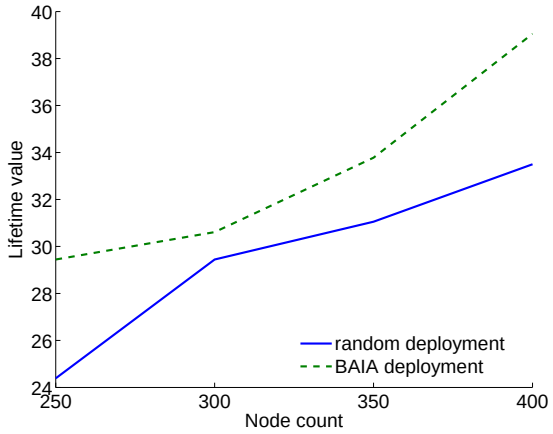


(b) Area size of 500×500 m^2.

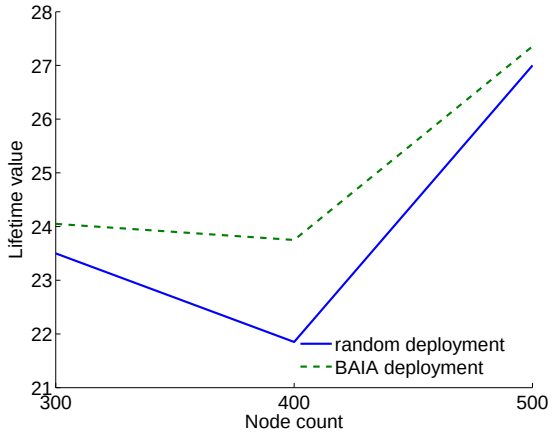


(c) Area size of 600×600 m^2.

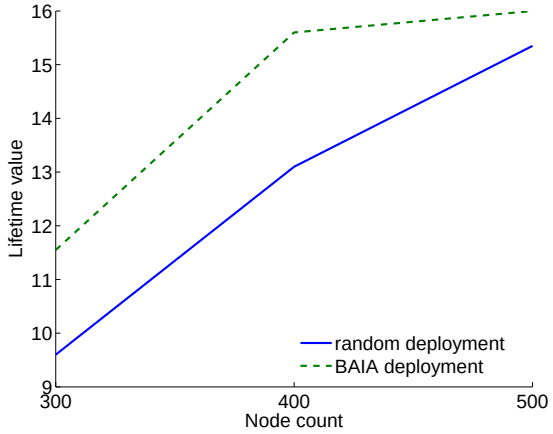
Figure 3.5. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 80 m, coverage requirement value of 0.7.



(a) Area size of 400×400 m^2.

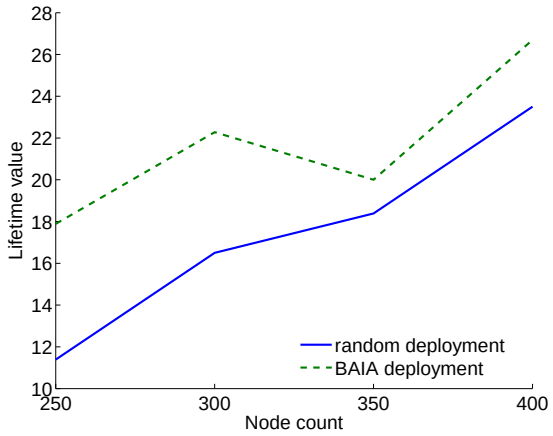


(b) Area size of 500×500 m^2.

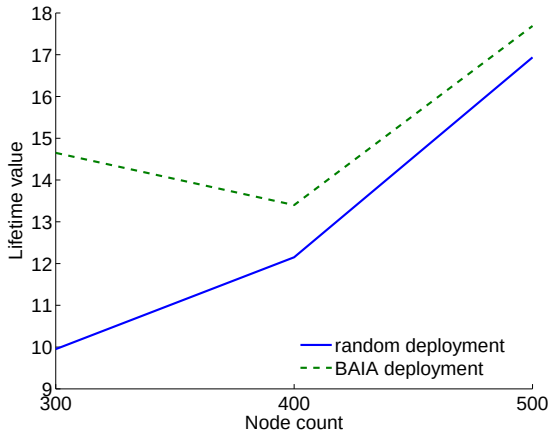


(c) Area size of 600×600 m^2.

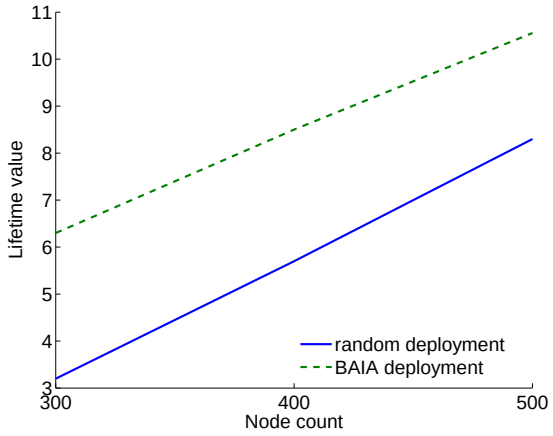
Figure 3.6. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of two, communication range of 80 m, coverage requirement value of 0.9.



(a) Area size of 400×400 m^2.

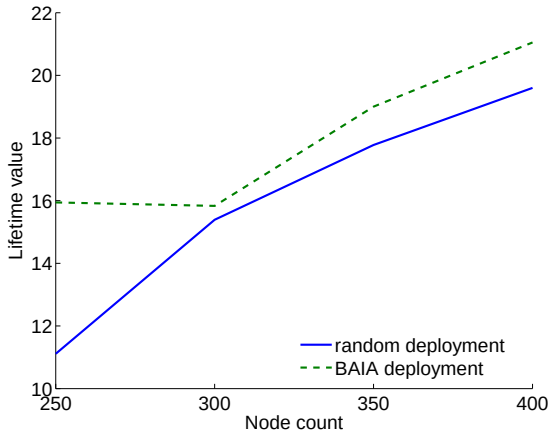


(b) Area size of 500×500 m^2.

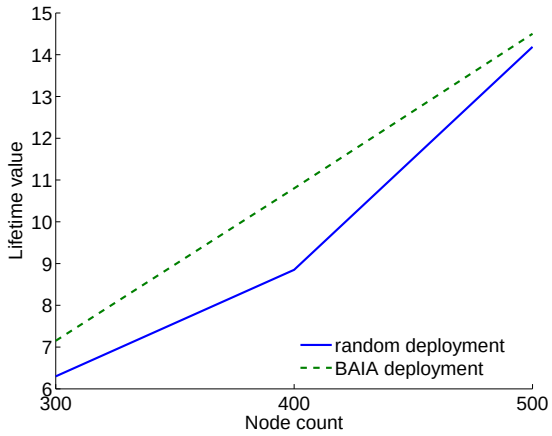


(c) Area size of 600×600 m^2.

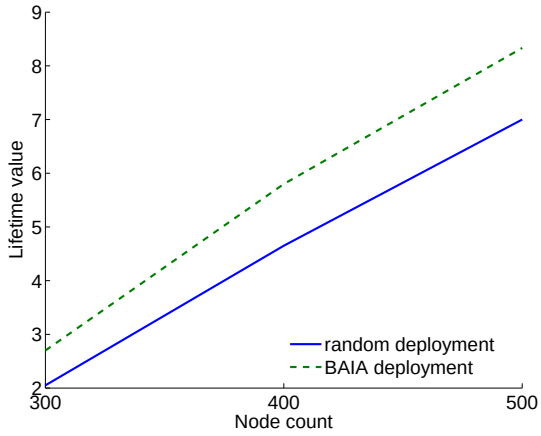
Figure 3.7. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of four, communication range of 60 m, coverage requirement value of 0.7.



(a) Area size of 400×400 m^2.



(b) Area size of 500×500 m^2.



(c) Area size of 600×600 m^2.

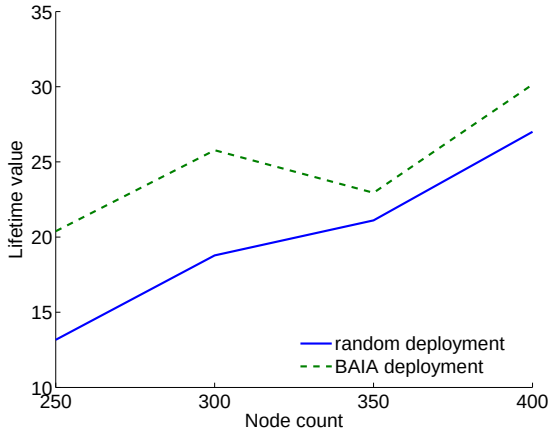
Figure 3.8. Effect of node size on network lifetime for message size of 512 bytes, path loss exponent of four, communication range of 60 m, coverage requirement value of 0.9.

deployment is highly lifetime limiting for both values. The network deaths are mainly caused by bottlenecks rather than higher energy losses due to the path loss changes. Yet by small redeployments, it is possible to increase the network lifetime considerably.

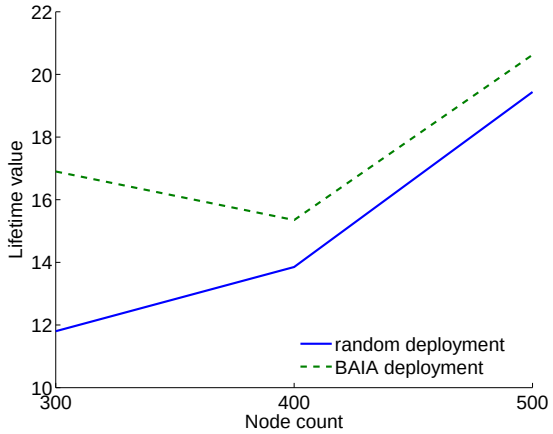
Figure 3.9 shows the lifetime results, when the size of the packet sent between the nodes are increased to 1024 bytes. We expect this change to have similar effects as changing the path loss, since they both increase the energy lost during transmitting a packet to the destination. The results are indeed similar to those with path loss value changed. In this scenario, our algorithm provides an average of 25 per cent network lifetime increase, performing better with especially normal density deployments like 350 nodes over 500×500 m^2 area or, 450 nodes over 600×600 m^2 area.

The proposed BAIA algorithm has a progressive and greedy approach. At each progression step, the link connection parameter is reduced and a greedy cluster formation starts. This approach assures that the algorithm will produce the possible bottleneck areas in one of the intermediate states. However, the results will not be necessarily actual causes of deaths for the network. The link connection parameter reduction also shows that the overall network is actually a compact one, hence other factors such as routing, event locality becomes more dominant in deciding the lifetime.

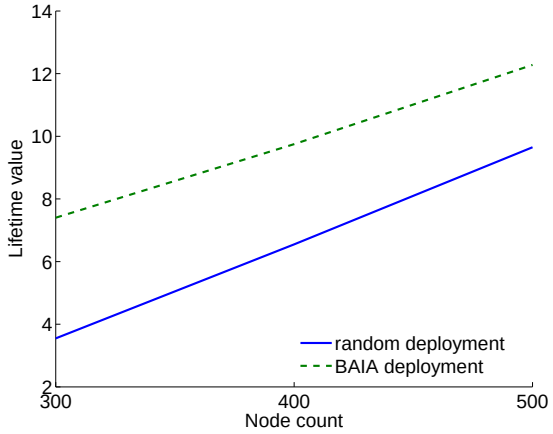
Computational experiments show that when redeployment is applied over a given network, BAIA gives considerably better results than redeploying randomly, especially for networks with nonuniformly placed sensors. The algorithm tries to increase the uniformity of the sensor location distribution, which in turn distributes the traffic over the network more evenly. In any case, BAIA is expected to increase the lifetime 20 per cent on the average compared to the random deployment.



(a) Area size of 400×400 m^2.



(b) Area size of 500×500 m^2.



(c) Area size of 600×600 m^2.

Figure 3.9. Effect of node size on network lifetime for message size of 1024 bytes, path loss exponent of two, communication range of 60 m, coverage requirement value of 0.7.

4. ENERGY HOLE MITIGATION FOR SURVEILLANCE

A highly important problem faced in border surveillance applications and more broadly in many-to-one WSNs is the uneven energy consumption [44]. The large number of sensors reporting to a single sink exposes the sensors around the sink to higher traffic loads. Larger radio communication on those nodes causes rapid consumption of energy, as their battery capacities are limited [45]. Consequently, the nodes around the sink begin to die, rendering sink as unreachable to other sensors in the network. Moreover, as explained in the previous chapter some nodes in the network become bottlenecks whose early deaths cause unreachable areas. In surveillance cases, some regions of the network can be blocked out due to intentional destructions sensor or jamming of the region. The resulting phenomenon is named as the energy hole problem in wireless sensor networks [46, 47]. Mitigation of the energy holes is crucial to extend the network lifetime in surveillance applications of WSNs.

Throughout the thesis, the terms mitigating and delaying of the holes will be used interchangeably. This is due to the temporal properties of the network. Mitigating a hole using one of the proposed methods at any step in a region of the network locally delays holes for the overall network, since a hole is inevitable at another region of the network in the future steps. Hence both terms point to the same approach.

There are three major problems related to a border surveillance WSN. The first problem is the network partitioning, where a section is disconnected from the main network. The second problem is breach path formation through which the intruder may pass undetected. The third problem emerges in cases when, although the sensing quality of the network may be above the required level, the sink may become unreachable to the network after the nodes around the sink die under high traffic load. Such formations mark the end of the effective lifetime of the network, as an intruder can easily pass through the network, without being sensed or sink being unaware of its presence. Due to the nature of border surveillance operations, this is not a desired situation which must be avoided.

In this chapter, redeployment based approaches are proposed to mitigate the energy hole problem in SWSNs and analyze their performance. To see the performance of each mitigation strategy, the variation of the deployment quality of the network over time for different network configurations are measured. Performance of each technique is evaluated using realistic MAC and routing layers in OPNET discrete event simulator [48]. All approaches analyzed increase the network lifetime without effecting the sensing quality of the network.

4.1. Mitigating the Energy Holes

There are two dominant approaches for energy conservation in a given WSN. The first method aims to reduce communication costs by keeping the transceiver of a node off as long as possible without affecting the network functionality which is called as *duty cycling*. The second one tries to reduce communication costs by transmitting only useful data to the destination which is also called as *data reduction* or *in-network processing*. There exist various proposed algorithms [49] and protocols for each of the approaches most of which are based on some common and not always correct assumptions such as the *unit disk model* and the *flat world model* as described in [50]. Most of the approaches minimize only the energy consumption without considering network performance or sensing quality.

Energy holes cause nonuniform coverage and decrease the sensing quality of the network. They can be avoided by balancing energy consumption in the network or maintaining uniform coverage. In Kosar *et al.*, we proposed a sensor redeployment method that can be used to mitigate sensing holes [51]. Initially only a portion of the available sensors is deployed and the rest is spared. Later on, sensors are redeployed over poorly covered regions in order to maximize the deployment quality of the network. Networks using the proposed redeployment technique achieve better sensing quality than networks using the same total number of sensors at once. However, the effects of redeployment on the network lifetime are not considered.

In some cases, hole formation may be the result of intentional node destruction.

Arifler propose an information theoretic approach for detecting systematic node destructions [52]. A self deployment technique for mobile sensors based on potential fields to achieve uniform coverage through the network is also proposed [53]. This approach has been shown to provide good coverage and can be used to maintain sensing quality above critical values but is not applicable in the case of static sensors. Chen *et al.* propose a virtual force algorithm based sensor deployment [54]. A distributed algorithm based on the Voronoi diagrams for sensor deployment that produces better coverage is proposed by Wang *et al.* [55]. However, both approaches assume mobility of sensors which may be impractical for majority of the applications. Kim *et al.* propose a fuzzy system based method for deciding sensor redeployment [56]. Their method requires special cluster head sensor nodes for deciding on the deployment site and number of sensors. This method may be useful for energy depleting nodes, yet initial sensing holes due to external factors cannot be covered by this method. Binary integer programming approach is also proposed for effective sensor placement when there exist various sensor types with different sensing quality and cost [57]. Authors also suggest the usage of a probabilistic approach in cases when calculation of effective positions is not feasible. Chiang and Byrd propose an algorithm for density control in which sensors decide whether to participate in the network by using the incoming packet information [58]. Olariu and Stojmenovic analyze the uneven energy consumption observed in many-to-one sensor networks [59]. They show that for energy consumption models having path loss value of two, there is no routing strategy that can avoid the energy hole creation around the sink.

When hole formation cannot be avoided, routing protocols try to forward packets along hole boundaries causing hole diffusion. A geometric modeling for the hole problem which prevents packets from traveling along hole boundaries is presented by Yu and friends [60]. Instead of bypassing encountered holes, Jia *et al.* propose avoiding holes in advance by using neighbor feedback [61]. Fang *et al.* find the hole boundaries using a distributed algorithm to enable formation of new routes [62]. This method, albeit useful for route formation, does not offer a solution for mitigating the hole problem. The holes defined by the authors are not actual sensing/communication voids but rather the nodes that does not offer routing improvements.

An algorithm that make use of specific beacon nodes to detect communication holes is also proposed by Funke [63]. However, the algorithm proposed does not present a solution for locating sensing holes. A hole detection method based on the largest empty circle problem is proposed [64]. Their approach finds local holes as circles for sensor deployment. However, to mitigate the holes, multiple iterations are required and the number of iterations is not known a priori and depend on the performance of the intermediate steps.

For sensor redeployment, there are different approaches employed. Yang and Cardei divide the deployment region into circles with the sink at the center and performing the redeployment such that the sensor density increases with the decreasing distance to the sink [65]. A similar redeployment method where sensors are deployed iteratively by either uniform distribution or with a specific distribution that enables more sensor deployment on the regions around the sink is proposed by Chatzigiannakis et al [66]. Wu *et al.* propose a somewhat similar method [67]. To mitigate the hole problem, sensors are deployed to circular strips, called coronas, around the sink. Each strip has different node densities that lead to a different type of nonuniformly deployed sensor network. However, this approach is ideally suited to sensor deployment regions that are square or circular. For different type of regions like rectangular narrow strips, meandering regions like river beds sink centered deployment approaches is not be able to avoid hole formations.

In this section, a deployment quality measure for the worst case sensing performance of the network is used. Such a strong metric is especially suitable for SWSNs. We assume probabilistic sensing model. Our approach aims to locate and mitigate the holes caused by outside effects such as destructions and the ones that form due to battery depletions over time. We aim to keep the sensing quality above threshold both in spatial and temporal domains making use of strong quality metrics and mitigation approaches.

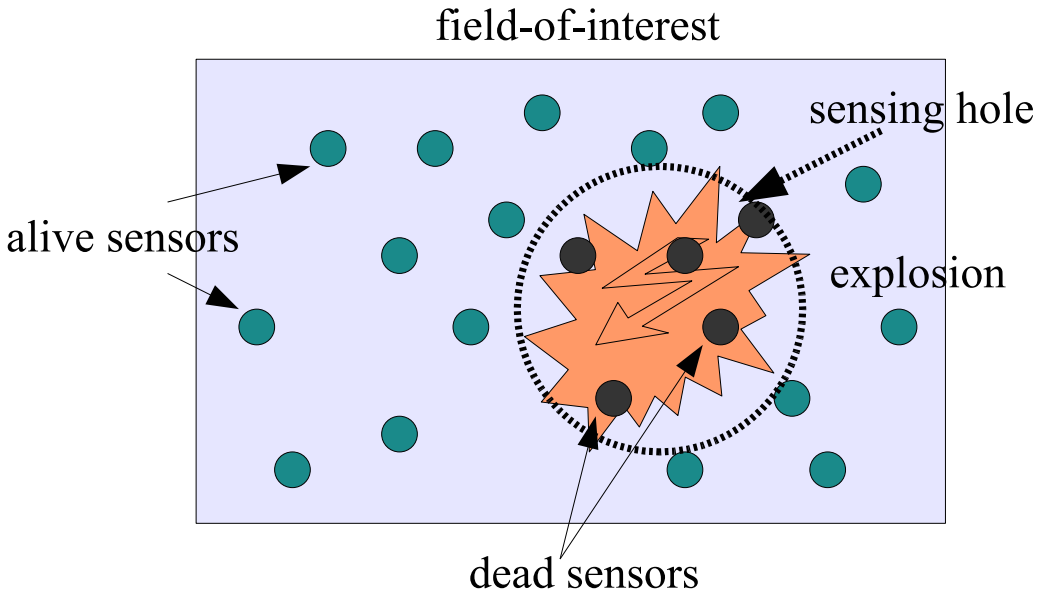


Figure 4.1. A simple border surveillance scenario where sensors are intentionally destructed.

4.2. Iterative Sensor Deployment Method

The base method in this work makes use of redeployment approach to decrease the adverse effects of holes. The method proposed is the iterative deployment algorithm (IDeA) where a portion of the sensors are reserved and the remaining are deployed in the initial phase. Based on the observed sensing quality, further redeployment iterations can be made to the holes using the spare sensors. The proposed algorithm uses fast image processing methods to identify probable sensing holes. For quality measurements, it uses a robust deployment quality metric to understand the overall sensing quality. The method aims to differentiate small and relatively low effect holes from the larger and sensing quality limiting ones.

In border surveillance scenarios, some set of sensors are deployed to the FoI to detect unauthorized intrusions. For a required security level, the number of sensors to be deployed can be determined as described in [68]. It is assumed that after deployment, some set of sensors are destroyed as seen in Figure 4.1 which cause hole formations. After the destruction, what is the deployment quality, which parts of the

network become useless or what should be done to mitigate this problem are the several questions to be answered. Redeployment of a set of spare sensors can be a solution. First, the holes must be located and isolated, afterwards redeployment is possible to those regions.

4.2.1. Iso-sensing Graph and Deployment Quality

To analyze the sensing quality of a WSN, the deployment quality measure proposed in [68] is utilized. To apply this method, the sensor detections are modeled as defined by Elfes [69]. Given a sensor k, the probability of detecting a target on grid point (i, j) is

$$p_{ijk} = \begin{cases} 0 & \text{if } r + r_e \leq d_{ijk}, \\ e^{-\lambda \alpha^{\beta}} & \text{if } r_e > |r - d_{ijk}|, \\ 1 & \text{if } r - r_e \geq d_{ijk}, \end{cases} \tag{4.1}$$

where $r_e < r$ are the thresholds for the sensing distance, d_{ijk} is the distance between the sensor k and the target and $\alpha = d - r + r_e$. The characteristics of different sensors can be reflected with parameters λ and β. Additionally, r_e ($r_e < r$) is a measure of uncertainty in sensor detection. If sensor-to-target distance is smaller than $r - r_e$, the target is absolutely detected. If the sensor-to-target distance is larger than $r + r_e$ the target cannot be detected.

The individual decisions of a subset of sensors may be highly correlated, particularly if the deployment is dense. If a sensor detects a target, it is highly probable that another sensor which is located in similar proximity as the first one also detects the same target, assuming homogeneous environmental factors. The performance of the sensor with the best detection capability is more valuable. We model the field as an eight-connected grid and under correlated-detections assumption, the detection probability for a grid point (i, j) is

$$p_{ij} = \max_{1 \leq k \leq R} p_{ijk}, \tag{4.2}$$

where R is the total number of sensors deployed in the region.

Modeling the field as a two-dimensional grid and adding the detection probability as the third dimension, a three-dimensional surface which is referred to as the iso-sensing graph is obtained [68]. The graph resembles the contour lines of a topographic map, where the altitude can be mapped to the detection probability and the grid points can be mapped to the two-dimensional locations. Therefore, it is named as iso-sensing graph.

A sample iso-sensing graph is shown in Figure 4.2. To evaluate the sensing quality of the network for the given model, the watershed deployment quality measure (WDQM) presented in [68] is used. The WDQM model is primarily based on watershed segmentation [70]. Given the iso-sensing graph, it can be assumed that at the minima points of the topography there are water pumps through which water is continuously pumped. As water rises up, dams are built to avoid merging of water sources from different valves. Finally, the dams called the watershed contours, segment the topographic map into different partitions. From the security view-point, the breach paths follow the watershed contours. The WDQM is the minimum of the maximum detection probabilities of these contours and gives an insight about the security level provided by the network. Consequently, WDQM is a strong measure to calculate the worst case performance of the network which is especially suitable for surveillance networks.

4.2.2. Hole Identification

In this section, ways to identify the locations of the sensing holes using the iso-sensing graph is explained. A topography analysis based approach, which explicitly makes use of morphology based image processing operations is used. Once the sensor positions are known, the iso-sensing graph is calculated as the initial step. The iso-sensing graph provides the necessary information to locate the sensing holes. The iso-sensing graph is used to differentiate the areas with high or low sensing quality. Highly sensed parts of the graph are the higher planes on the topography and are used as auxiliary means to locate the holes.

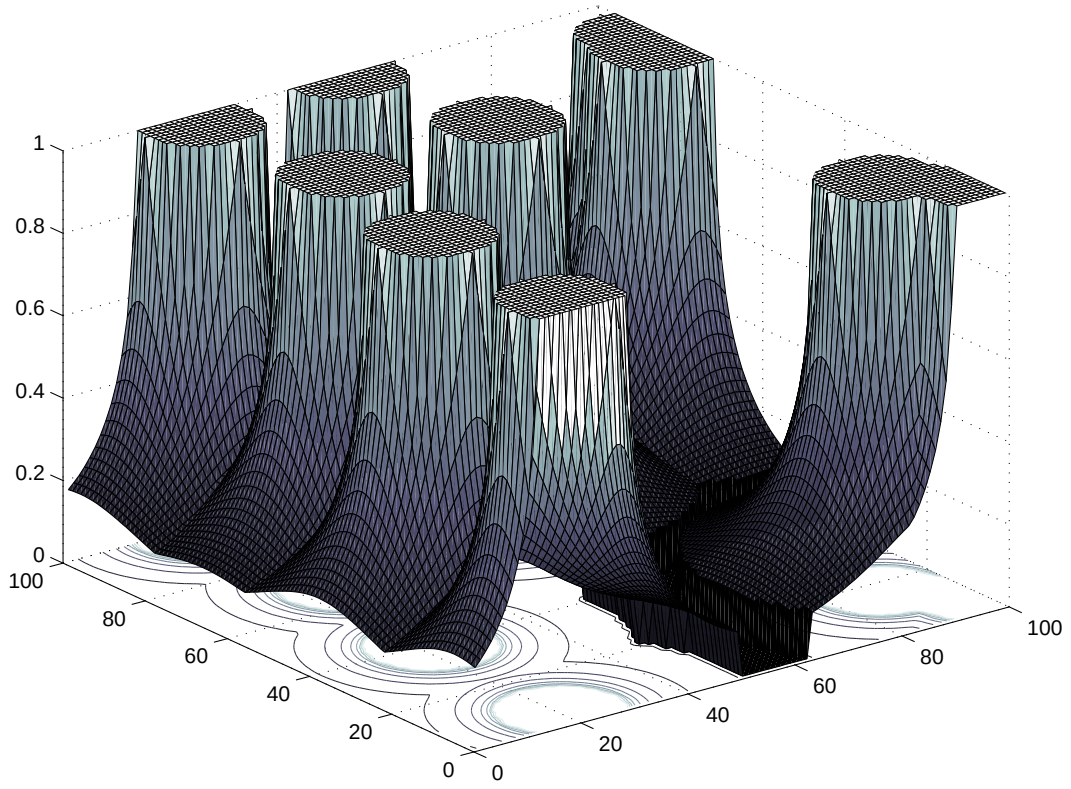


Figure 4.2. A sensor network iso-sensing map example.

Given the iso-sensing graph, image processing algorithms are employed to detect the candidate areas. In the initial step, the graph is converted into a color image. The $2D$ coordinates, denoted by the $R \times 2$ matrix $\overline{x}$, of the map are used as the pixel coordinates of the image and the sensing quality of grid points are used as the color intensity value at the corresponding pixel, $h = H(\overline{x})$. An example for the resulting color image is given in Figure 4.3.

To reveal the highly sensed parts, the initial step of the algorithm is to discard segments that have low sensing quality values. For this purpose, the filter,

$$w_1(\overline{x}) = \begin{cases} H(\overline{x}) & \text{if } H(\overline{x}) \geq \frac{R*\pi*c^2}{l*h}, \\ 0 & \text{otherwise,} \end{cases} \tag{4.3}$$

is applied to the color image. Here, c is the sensing range of each sensor, l and h are the length and the height of the covered area respectively, R is the total number

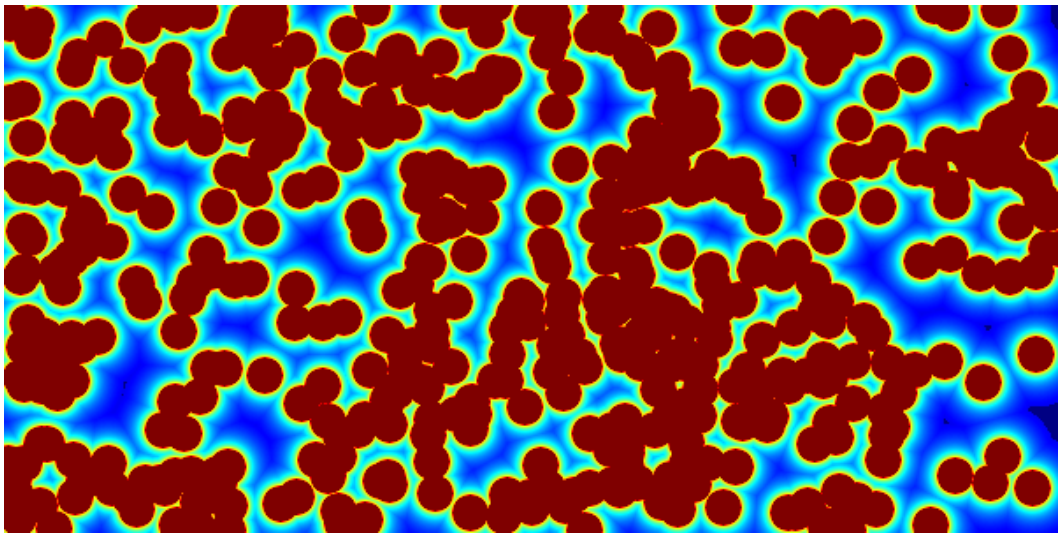


Figure 4.3. Iso-sensing graph converted to a color image.

of sensors deployed. This filter isolates the grid points that have a sensing quality above average, which in turn allows for connected component analysis to differentiate those highly sensed groups [71]. The component labeling step groups and labels pixels with similar intensity values and are connected to each other. The particular approach used is the eight-connectivity connected component labeling algorithm [71]. The eight-connectivity measure groups grid points whose all eight neighbors have higher than average sensing quality values.

After this step, there may be small area segments left behind. These are insignificant compared to the main segments and to filter them, a second filter,

$$w_2(\overline{x}) = \begin{cases} H(\overline{x}) & \text{if } (\overline{x}) \in G_i \text{ and } A(G_i) \geq l * h * 0.01, \\ 0 & \text{otherwise,} \end{cases} \tag{4.4}$$

is applied where G_i denotes component group indexed by i and A is the area of the corresponding group. By using the filter w_2, the regions smaller than one per cent of the total area are discarded. With values below, the filter does not efficiently ignore the small sized hole areas. Redeployment over such small sized areas does not effect the overall sensing quality and the running time of the further steps is increased with the increasing number of areas. However, increasing the cut off value increases the risk of

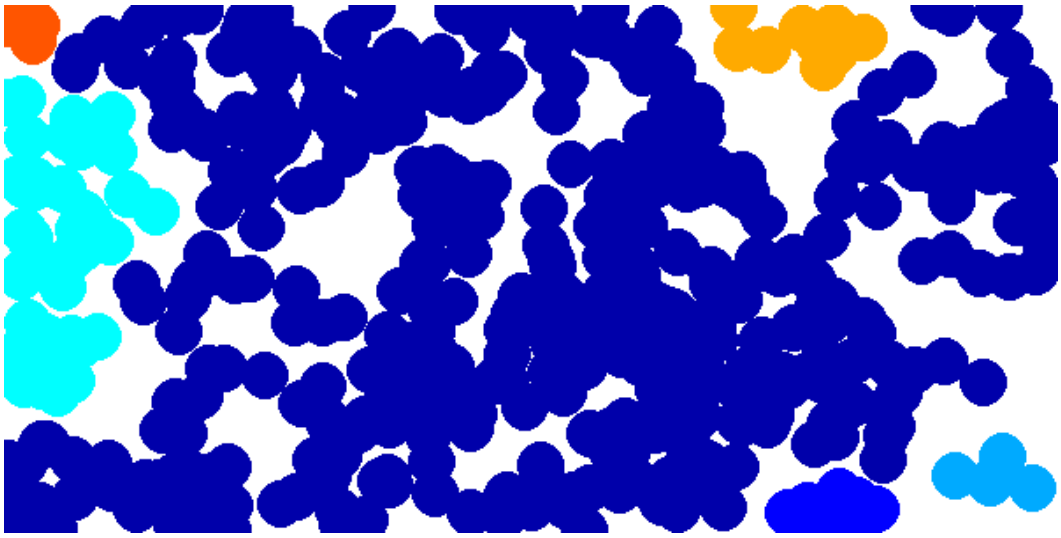


Figure 4.4. Color image formed after filtering low sensing quality areas, applying connected component analysis and filtering small sized groups in the initial image.

discarding big size holes. Based on the average of 20 different scenarios, 20 repetitions for each scenario, one per cent is found as the optimal cut off value.

As a result of the filter operation, comparatively much more important groups with high sensing values are left behind in the image. An example image after w_2 filter applied is given in Figure 4.4. In the images, the colors are used to differentiate between the eight-connected sensing groups and do not provide information about the sensing quality of the group or any other metric.

4.2.3. Redeployment

At this point, the parts of the network that have high sensing quality values are isolated. Redeployment must be done in between these areas which are the topographic valleys between the higher planes of the iso-sensing map.

In the corresponding color image, redeployment candidate areas lie between the connected components. To locate those areas in the algorithm, a polygon that wraps two components is drawn. However, deployment is just made to the parts of the polygon

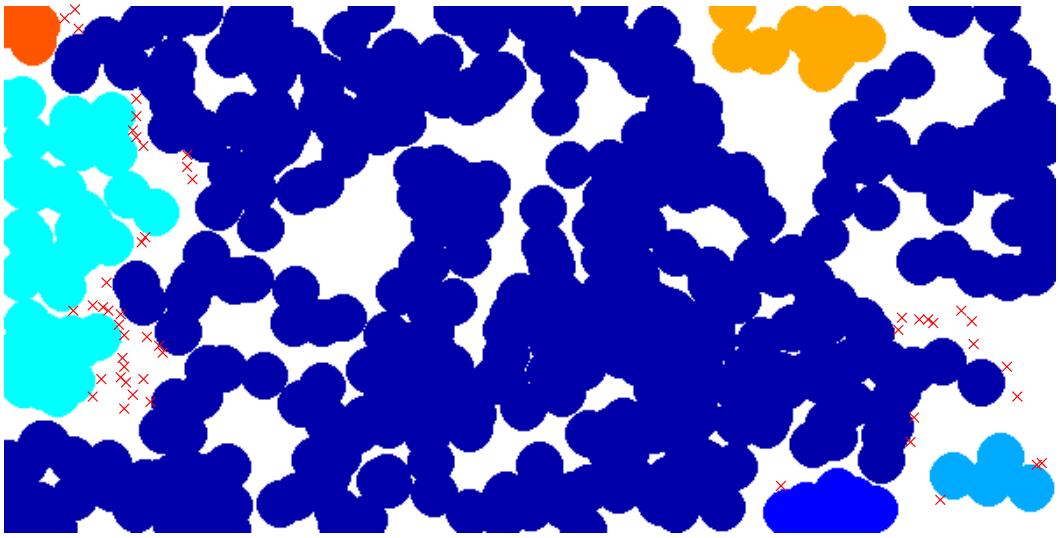


Figure 4.5. Final color image formed after redeployment is performed, 'x' sign denotes the sensors deployed in the redeployment phase.

not covered by those two connected components. In this manner, the area surrounded by the polygon perimeter and component borders is a future sensing void candidate and is a redeployment site. Once all such regions are found, the redeployment is performed and the algorithm stops. An example final image is shown in Figure 4.5.

The overall pseudo code of the steps of IDeA is presented in Table 4.1. It should be noted that the algorithm presented here is a centralized version. Yet, the algorithm can also be run by the cluster heads inside the network in a distributed manner. The distributed version is presented in Table 4.2. The initial cluster formation is not within the scope of this work and is assumed to be done using a suitable clustering algorithm. The algorithm given assumes only the sink can initiate the redeployment procedure. For the cases where cluster heads are able to start redeployment, steps 11 and 12 can be omitted. Given the local alive node coordinates, cluster heads can use the algorithm to locate the sensing holes and initiate the redeployment procedure locally. In any case, it should be noted that redeployment procedure is can be costly if performed frequently and should be coordinated by the sink to accommodate the increasing costs. The distributed version of the algorithm can used by the cluster heads to modify the routing scheme in the cluster to decrease the load on sensors around the holes to balance

Table 4.1. IDeA steps.

1: R sensors given
2: $R * p$ sensors separated away, p is the redeployment percentage
3: $R * (1 - p)$ sensors deployed over the terrain
Require: $(x_i, y_i), \forall i = 1..(R * (1 - p))$ known
4: S (Iso-sensing graph) is formed
5: $I = H(S)$, form the color image
6: $I' = w_1(I)$, apply connectivity filter
7: $I'' = w_2(I')$, apply small region filter
Require: G_i (highly sensed area groups) are known
8: $R_k = \text{Poly}(G_i, G_j) \setminus (G_i, G_j)$, find the deployment regions between
9: Deploy $R * p$ sensors over R_k regions, each regions gets sensor count proportional to its area size

the high energy load due to the avalanche effect on them. Similarly, sleep schedules of the sensors located close to the holes can be modified to help decrease the effects of holes and increase the lifetime of the border nodes.

4.3. Simulation Setup and Results

It this chapter the network lifetime is defined as the time from the initial network deployment until the time when the WDQM value drops below the acceptable level or the sink itself becomes unreachable. The WSN is modeled in OPNET as a rectangular field with many sensors, one sink and random intruders [48]. The model contains the wireless sensor, sink, intruder, mobility configuration and network configuration components. The wireless sensor is composed of three layers: the intruder detector, the routing layer (using Min-Hop routing) and the MAC layer (using S-MAC). The sink is similar to the wireless sensor except that it has an infinite amount of energy. The intruder is modeled as a mobile object that can be detected by the sensors. The mobility parameters of the intruder such as speed, interarrival period, simultaneous intruder count and direction update frequency are managed by the mobility configuration.

Table 4.2. IDeA steps, distributed version.

1: R sensors given.

2: $R * p$ sensors separated away. {p is the redeployment percentage.}

3: $R * (1 - p)$ sensors deployed over the terrain.

Require: K clusters are formed.

4: **for all** $k_m, \forall m = 1..K$ **do**

Require: Cluster head knows the node coordinates in the cluster.

5: S_k {Form iso-sensing graph.}

6: $I_k = H(S_k)$ {Form the color image.}

7: $I'_k = w_1(I_k)$ {Apply connectivity filter}

8: $I''_k = w_2(I'_k)$ {Apply small region filter.}

Require: G_k are known {Highly sensed area groups.}

9: $D_{km} = \text{Poly}(G_k i, G_k j) \setminus (G_k i, G_k j)$ {Find the deployment regions between.}

10: **end for**

11: Cluster heads send deployment region information to sink.

12: $D_k = \bigcup_m D_{km}$ {The deployment region information is combined at the cluster.}

13: Deploy $R * p$ sensors over D_k regions, each regions gets sensor count proportional to its area size.

Common network configuration parameters are managed by the network configuration component.

In the simulations, S-MAC is chosen as the MAC layer due to its energy efficiency [72]. Dynamic Min-Hop routing is chosen as the routing layer due to its simplicity and ability to balance the network load in the long run [73]. In the initial scenario, the deployment site is modeled as a 900×100 m^2 rectangle and 550 sensors are deployed in the field. The field resembles a strip border where surveillance is performed by sensors. There exists only one sink located at the center of the field. The sensors are uniformly randomly deployed and not mobile. The intruder starts moving through the network starting from the upper boundary at a constant speed of two meters per second in a forward random direction and changes its direction randomly every five seconds. Intruder is not allowed to make backward movements and it is reflected from lateral borders in order to avoid going out of the field. When a sensor detects the presence of an intruder, it sends a packet towards the sink to report the intrusion. Intruder interarrival period is uniformly distributed between 500 and 700 seconds. All wireless sensors have a sensing range of 20 m and the maximum communication range of 40 m. The radio circuitry is modeled similar to the Chipcon CC1000 radio chip [74]. The sensors consume 1 μA in the idle state and 0.2 μA in the sleep state. The receiver consumes 7.4 mA and the transmitter consumes between 5.3 mA and 26.7 mA. The channel data rate is 38.4 kbps. S-MAC layer has a duty cycle of two percent. All the sensors have a sensing interval of 10 sec and use the Elfes's probabilistic detection model with parameters $\beta = 0.9$, $\lambda = 0.1$, $r = 20$ m, $r_e = 5$ m [69]. Sensors are equipped with a 3 V Li-Ion Cell that can supply 2100 mAh.

The system is configured to check the network health status (WDQM) after every sensor death and the network is considered dead if the WQDM value drops below the minimum value of 0.75 or the sink becomes unreachable. The redeployment process is initiated if the WDQM of the monitored network drops below the threshold value of 0.8 and further redeployment is possible.

Typical parameters used for the sensor model such as the energy model and

detector parameters are shown in Table 4.3. Other simulation parameters used for S-MAC layer, WDQM calculations and energy hole mitigation strategies are shown in Table 4.4. Parameters used for intruder mobility modeling are shown in Table 4.5. Each of the simulation scenarios repeated are 25 times and the averages are considered. No intruder was able to cross the border without being detected in any of the scenarios.

We study the effect of the number of sensors to the network lifetime by keeping the network dimensions fixed and simulate five different scenarios with 450, 550, 650, 750 and 850 nodes. Each scenario is simulated using four different configurations: first without any strategy for energy hole mitigation, shown as None, then using different versions of the IDeA approach:

- Random: In the Random redeployment strategy, IDeA step eight is modified. Sensor can be deployed all over the polygons formed, instead of the regions between the connected components. In this version the routing capability of the network is maximized.
- MaxDQM: In MaxDQM redeployment strategy, plain IDeA is used which maximizes the current DQM value of the network.
- Hybrid: In Hybrid redeployment, a balance between Random and MaxDQM is used. At redeployment step of IDeA, half of the sensors are placed over the whole polygon and the other half is deployed to the regions between connected component. This version aims to increase both the deployment quality and the routing capability of the network.

In all approaches 80 per cent of the available sensors are initially deployed. Figure 4.6 shows the effectiveness of each of the approaches in covering the emerging holes. In the figure, nodes that are living and connected to the other parts of the network are shown by $\bigcirc$ marks. Nodes that have consumed all their energy are shown by $\triangle$ marks, and $\square$ marks represent nodes that have not consumed all their energy yet but are not connected to the network. From the sink point of view, unreachable alive nodes are called dead as well. MaxDQM redeployment deploys spare sensors in areas which need urgent care, while Random redeployment deploys spare sensors randomly over

Table 4.3. Hole mitigation test sensor parameters.

Data	
Data Packet Size (b)	1024
Buffer Size (packets)	10
Data Reporting Period (s)	1800
Data Rate (bps)	38,400
Max TX Distance (m)	40
Path Loss Exponent	2
Energy	
Initial Energy (J)	100
TX Current (max) (mA)	26.7
TX Current (min) (mA)	5.3
RX Current (mA)	7.4
IDLE Current (mA)	0.001
SLEEP Current (mA)	0.0002
Transceiver Voltage (V)	3
Detection	
Detection Model	ELFES
r	20
r_e	5
β	0.9
λ	0.1
Sensing Interval (s)	10

Table 4.4. Hole mitigation test network parameters.

Network	
Dimensions X×Y (m^2)	900×100
Number of Sensors	450/ 550/650/750/850
Routing Type	Min-Hop
MAC Type	S-MAC
S-MAC	
Setup Duration (s)	10
Duty Cycle Percentage	2
WDQM	
WDQM_Trigger_Limit	0.8
MIN_WDQM	0.75
Monitor Period (s)	2
Redeployment	
Redeployment Type	Random/MaxDQM/Hybrid
Initial Deployment Prc.	0.8

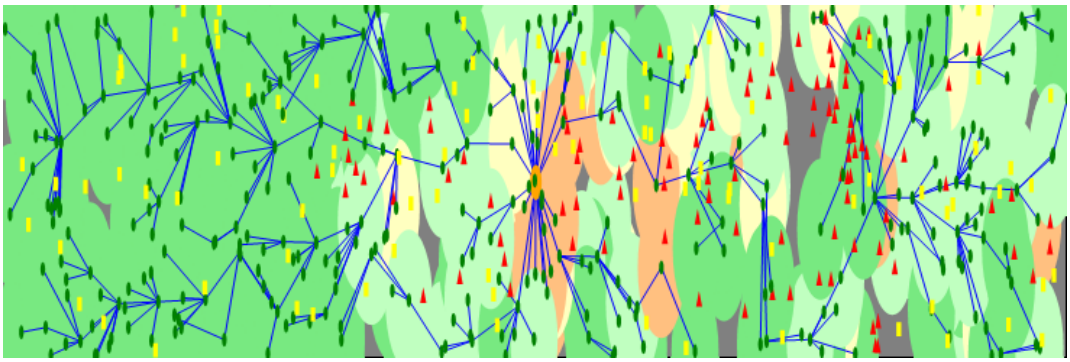
Table 4.5. Hole mitigation test intruder mobility parameters.

Intruder	
Speed (m/s)	2
Direction Update Frequency (s)	5
Intruder Interarrival	uniform(500-700)
Forward Movement Angle (deg)	uniform(0-180)
Target Batch	1

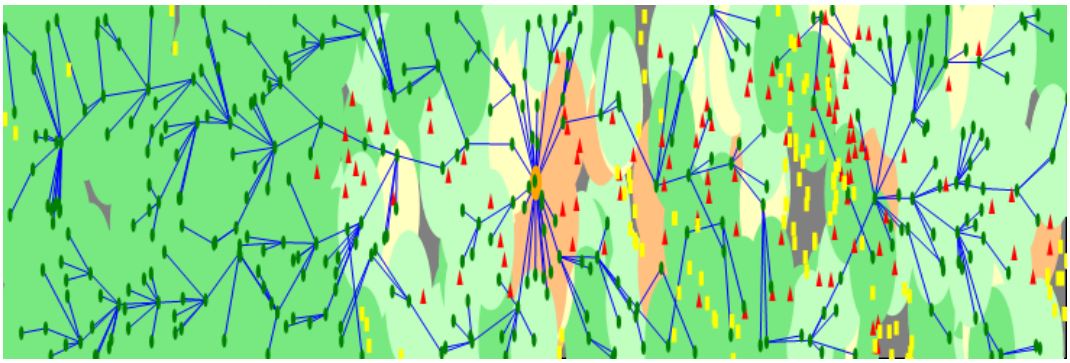
the network. Deploying spare sensors in areas which need urgent care improves sensing quality of the network. On the other hand, random sensor deployment does not guarantee any improvement in areas which need urgent care but increases the overall network robustness. Hybrid redeployment covers holes that need immediate care and also improves the overall network coverage.

As seen in Figure 4.7, as the total number of deployed sensors increases, the network lifetime increases. For small number of sensors, the MaxDQM redeployment strategy gives better results since it uses the spare sensors more effectively. When the number of sensors is high, Random and MaxDQM redeployment strategies have similar results since the amount of spare sensors is larger and the created holes are covered anyway. Random redeployment gives similar results to the default scenario for small number of sensors, but better results than the default scenario for larger number of sensors since it avoids overhearing problems by using smaller number of sensors initially and increases network's overall resilience by redeploying sensors randomly. The Hybrid redeployment strategy gives always better results since it combines the ability of MaxDQM to cover holes effectively and the ability of Random redeployment to increase network's overall robustness.

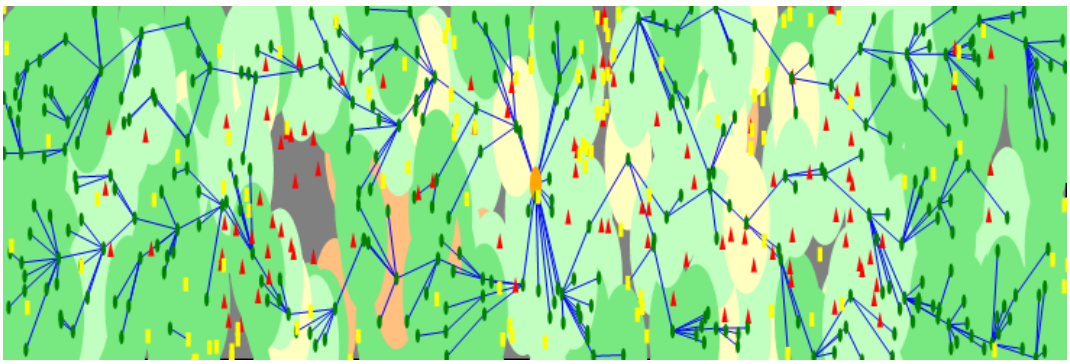
To understand the state of the network at a specific instant, it is possible to look at different indicators such as the WDQM measure or the number of connected sensors. Different hole mitigation strategies are simulated as long as the surveillance quality is above the given threshold. The strategies that are used are detailed in our manuscript [75]. Aggregation strategy combines correlated data from local sensors to decrease the number of packets in the network. Neighborhood density control (NDC) controls the number of awake sensors in a locality to decrease the overall traffic caused. The effect of each mitigation strategy on the network lifetime is studied. It is noticed that in the Random redeployment scenario, sensors have the shortest lifetime while in the aggregation scenario sensors have the longest lifetime as shown in Figure 4.8. Random and the MaxDQM redeployment strategies initially behave similarly, but after the redeployment step the MaxDQM redeployment scenario lifetime gets longer due to more effective deployment of spare sensors. We notice that the NDC strategy gives



(a) Random redeployment.



(b) MaxDQM redeployment.



(c) HYBRID redeployment.

Figure 4.6. Energy hole mitigation using different redeployment techniques.

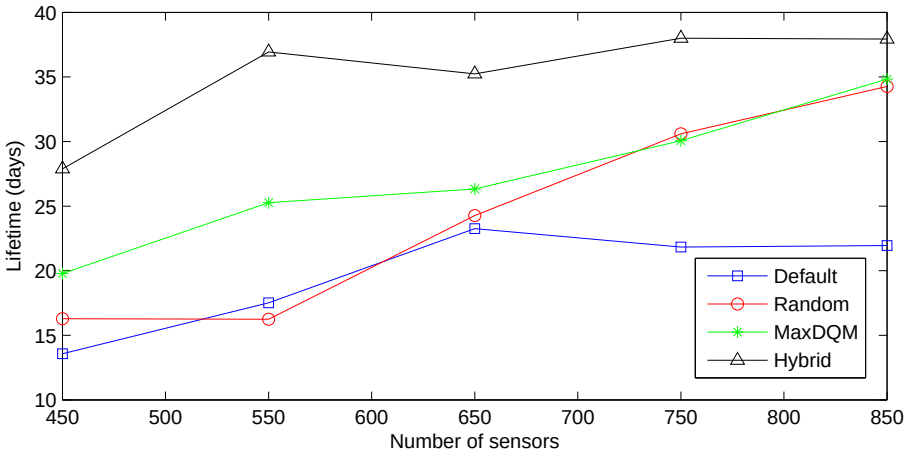


Figure 4.7. Total sensors used vs. lifetime.

longer lifetime than Random and MaxDQM redeployment strategies. NDC strategy starts with 245 nodes and spare nodes are activated whenever an active node is exhausted. Hybrid redeployment strategy has similar lifetime to NDC strategy. Using the proposed methodologies, it is possible to increase the lifetime of the network more than 100 per cent. The methodology to be used should be chosen carefully based on the requirements like sustainability, cost and availability of the network, the sensing quality, the number of sensors. Yet, in any case, network lifetime based on the worst case scenario is increased compared to the default case.

The variation of the WDQM with time is shown in Figure 4.9. We notice that Aggregation gives longer lifetime and higher WDQM values. Default scenario gives higher WDQM value than Redeployment scenarios but has shorter lifetime. Redeployment approaches give lower WDQM values since they use smaller initial number of sensors but give a longer lifetime than that of the default scenario. NDC strategy has lower initial WDQM value due to the smaller number of sensors initially activated, but gives longer lifetime than redeployment strategies due to the minimization of overhearing costs and resilience to failures. Hybrid redeployment gives higher WDQM values and longer lifetime compared to NDC and other redeployment strategies.

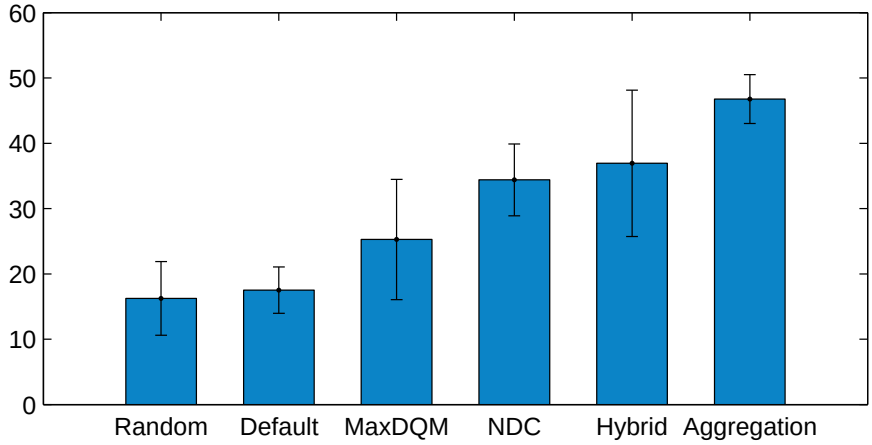


Figure 4.8. Lifetime comparison.

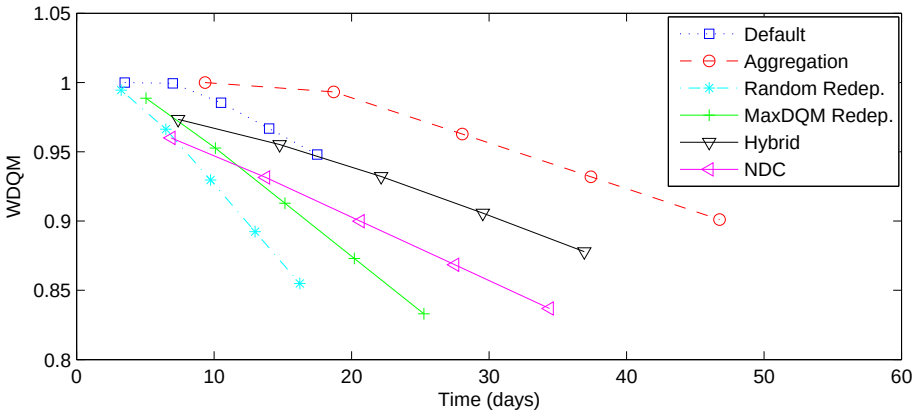


Figure 4.9. WDQM vs. time.

5. ANALYTICAL APPROACH TO DEPLOYMENT QUALITY FOR SURVEILLANCE CONSIDERING HOLES

Analytical approaches to provide reliable deployment quality values must take hole formations into account for realistic scenarios. By providing a closed-form DQM with loss assumptions it is able to give assistance to the WSN designers and supervisors in the following situations:

- A fast and precise calculation of sensing quality is required for a robust operation of an initially deployed surveillance wireless sensor network (SWSN) [77].
- At any point in the lifetime of a network, critical decisions should be taken on the sensor redeployment, relocation, sleep and wake commands [78].
- Online monitoring of the sensing quality, which affects the reliable detection of intruders, is required to estimate the current network performance [79].
- The network parameters should be determined carefully to obtain an adequate level of sensing capability in the network [80].

These major factors that guides the SWSN engineers in designing and maintenance of the network are the main motivations to propose a deployment quality metric with sensor loss provision. The network engineers can utilize the given metric to find a solution to these situations in the following ways:

- SWSNs bear mission and time critical tasks, which requires a minimum level of deployment quality. The DQM monitoring approaches based on simulations require unacceptably long periods, whereas our analytical approach provides real-time results.
- The critical decisions about the network operation depend mostly on the sensing quality metrics for SWSNs. Simulation based decisions induces long processing times extending the total operation delays. Our analytical method provides a low-cost calculation of the DQM value in terms of processing time.

- The detection reliability of the network can be estimated by an online calculation of the DQM value using our analysis.
- Our analysis can be used as a tool to calculate the network parameters such as the required sensor node count and node sensing range to monitor a region, when certain parameters are provided as an input by the network engineers.

The online calculation of the DQM value is established using the prediction of current network topology acquired by the feedbacks of the sensors in the form of ALIVE messages. Any successful delivery of ALIVE messages from an individual sensor within a given time period indicates that the sensor is alive. The remaining sensors are assumed to be inaccessible and constitute the energy holes/jammed regions in the network. At the end of the time period, a model of the current network topology is constructed based on the information gathered from the sensors.

5.1. Related Work

In their work, Lazos *et al.* provide analytical results for target detection in a convex deployment site [81]. They assume that the sensors have convex coverage areas and are uniformly placed in the deployment site. They use the presented results to compare the random and heuristic based sensor placement methods. They also assume that the sensors may have heterogeneous sensing capabilities. Cao *et al.* present closed form results for target detection under given network parameters [82]. They assume uniform sensor distribution, with fixed sensing range. Clouqueur *et al.* present algorithms for collaborative target detection [83]. In this work, the faulty detecting sensors are aimed to be excluded from the detection probability, by using a decision fusion. Dousse *et al.* present a model to calculate the delay for an intruder to be detected by a sensor that is connected to the network [84]. They assume a fixed sensing range of sensors. Wang *et al.* present an analytical model to measure the event detection latency using a probabilistic approach [85]. The model incorporates detection models involving one or more sensor detection. The model is used to evaluate the coverage performance of a WSN. Ren *et al.* try to calculate the detection probability of the network based on the operation schedule of the sensors [86]. All of the works

presented have the implicit assumption that the area of interest is homogeneously covered. On the contrary, our work assumes that there are holes within the sensing field caused by the destruction of sensors, energy depletion or jammers, that render some sensors unusable. As a result, the overall sensor distribution is no longer homogeneous. Wang *et al.* propose a scheme for analytical coverage calculation of heterogeneous wireless sensor network [87]. However, the heterogeneity of the network is due to the type of sensor nodes, which have different sensing capabilities. In our work, the deployment and the holes inside the network cause the heterogeneity. In a different approach, Saipulla *et al.* concentrate on deployment of sensors as a line and barrier coverage for surveillance. The aim is to analyze the deployment performance of a tripwire like sensor network deployment [88]. In our work, barrier coverage is not analyzed, instead the coverage of the total intrusion area is assumed.

The overall sensing performance of the WSN is analyzed in [89]. Authors present a fusion based detection model for wireless sensor networks. Their distributed approach tries to calculate the detection probability locally using a hypothesis testing on the number of detections in the vicinity and the system level detection performance is approximated analytically using the central limit theorem. However, this approach requires a dense network to sustain sufficient number of detections to perform the hypothesis test.

The subject of jammers is studied in [90], which presents algorithms to detect the existence of radio interference jamming inside the network. Ngai *et al.* approach the sinkhole attacks, where attacker tries to alter the data obtained by the sink by attacking the nodes surrounding [91]. They present an intrusion detection algorithm that tries to overcome such attacks by statistical and geographic information based operations. The malicious nodes can be excluded from the network using this algorithm. Li *et al.* present a model for controllable jamming attacks and possible solutions to such attacks [92]. Cagalj *et al.* propose probabilistic communication wormholes out of the jammed region inside the network [93]. Their approach tries to decrease event reporting delay caused by jammers. Jamming and other possible attacks on WSNs are modeled in [94]. They also present a routing model to stand against attacks on the

network. Li and Hunter present a distributed algorithm to detect and recover holes in the network [95].

As a summary, our work presents a deployment quality model for heterogeneous network formations due to the sensor losses caused by energy depletion, destructions and jammers. The proposed solution is designed for the border surveillance applications, where a target passes through the network. The detection is the task of the sensors over the total area. Hence, no barrier coverage is assumed. Overcoming the sensor losses is not in the scope of this work. On the contrary, it is aimed to present a fast calculation scheme to understand the overall sensing quality in existence of sensor losses at a given snapshot of border surveillance wireless sensor network.

5.2. Model Assumptions

This section presents the model assumptions used in the analytical derivation of the DQM. In this analysis, a convex 2-D deployment site, S, of perimeter L sensed by N_s sensors is assumed. Each sensor s_i is assumed to have convex sensing coverage of S_i with perimeter L_i, uniformly and independently distributed within S. In addition, there are N_h jammers where each jammer h_j have a jamming coverage area of H_j, uniformly and independently distributed within S.

The energy holes caused by node losses in the network may occur due to several factors such as the destruction and energy depletion of sensors and jammer attacks. The area of these energy holes are determined as follows. Each sensor is assumed to transmit ALIVE messages periodically to the sink, which infers that the source node is alive upon the reception of a message. The remaining sensors are assumed to be inaccessible and constitute the energy holes/jammed regions in the network. Since the area of any type of energy holes is determined similarly, for simplicity, the energy holes are mentioned as the jamming areas hereinafter.

In addition, it is assumed that the trajectories of the mobile targets are straight lines, crossing the deployment area equiprobably along the area border. Although this

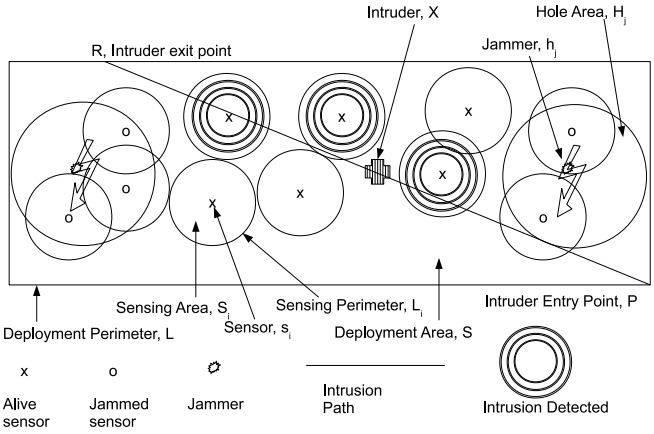


Figure 5.1. Graphical representation of the border surveillance intruder detection avoiding jammers problem.

assumption brings about a restriction on the variety of possible trajectories, the linear trajectory provides the shortest path between arbitrary entry and exit points in the deployment area, minimizing the time required for detection. The detection probability for a linear trajectory provides the worst case probability among all trajectories with the same entry and exit points. In addition, the parameterization of line trajectories can easily be integrated in our analytical calculation and the physical interpretation of the network model.

Finally, the detection model is assumed to be the binary detection model where a target is assumed to be detected if it enters the sensing coverage of a sensor.

5.3. Problem Definition

In this section a formal definition of the SWSN intruder detection problem in existence of jammers/energy holes is provided. Moreover, two additional problems will be provided which will be mapped to each other to derive a solution for the geometric domain utilizing metrics and tools from Integral Geometry and Geometric Probability [96].

The following definition is the formal statement of our problem:

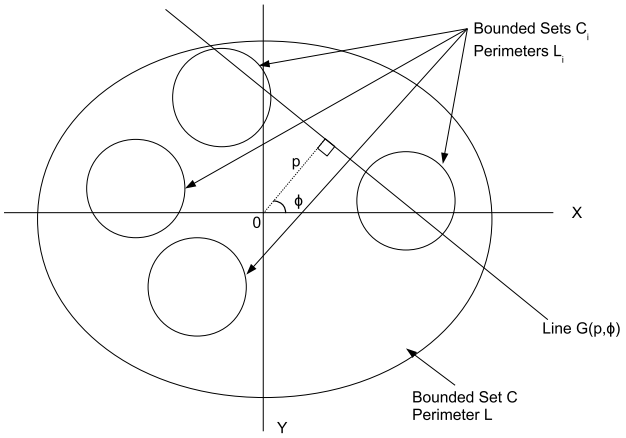


Figure 5.2. Graphical representation of the line-set intersection problem.

Problem 1. *Border surveillance intruder detection avoiding jammers problem: We have a deployment area, S, of perimeter L sensed by N_s sensors where each sensor s_i has a sensing coverage S_i with perimeter L_i, uniformly and independently distributed within S. In addition, there are N_h jammers where each jammer h_j have a jamming coverage area of H_j, uniformly and independently distributed within S. The sensors residing in the jamming area are assumed to be dead. What is the probability $P_D(k)$ that an intruder X randomly crossing S is detected by at least one sensor, not residing in any of the jamming areas?*

A graphical representation of Problem 1 is given in Figure 5.1. The probability $P_D(k)$ is defined as the deployment quality measure (DQM) of such a sensor network. The solution of this problem requires the calculation of the probability of detection for a set of sensors in the network. In order to provide a solution for this problem, two additional problems are introduced. The first one, Problem 2, is the simplified version of Problem 1 which does not include jammers. The second one, Problem 3, corresponds to the geometric interpretation of such a sensor network. We will provide a mapping between these two problems to show that they are equivalent, i.e., both can be reduced to each other.

Problem 2. *Border surveillance intruder detection problem: We have a deployment area, S, of perimeter L sensed by N_s sensors where each sensor s_i has a sensing cov-*

erage S_i with perimeter L_i, uniformly and independently distributed deployed within S. What is the probability $P_D(k)$ that an intruder X randomly crossing S is detected by at least one sensor?

Problem 3. *Line-set intersection problem [81]: We have a bounded set C of perimeter length L and N_s sets C_i with perimeter length L_i, uniformly and independently distributed inside C. What is the probability $P_D(k)$ that a random line G intersecting C, also intersects at least one of the sets C_i, $i = 1 \ldots N_s$?*

An illustration of the line-set intersection problem is given in Figure 5.2. Our analytical study relies on the fact that these two problems can be reduced to each other with a bijection between both domains. The following lemma presents this bijection.

Lemma 1. *Border surveillance intruder detection problem and the line-set intersection problem are equivalent, i.e., can be reduced to each other with a bijective mapping.*

Proof. We provide the following mapping of intruder detection problem to line-set intersection problem proposed in [81] for a more general case of these two problems. We map the deployment area, S, to a bounded set C, which is a collection of points in the plane with perimeter length L. The sensing area, S_i of sensor s_i is mapped to a bounded set C_i with perimeter length L_i, uniformly and independently distributed in C. We map the trajectory of the intruder X to a straight line $G(p, \phi)$ in the plane, defined by p as the shortest distance of G to the origin O of a coordinate system and ϕ as the angle of the line perpendicular to G with respect to the x axis. This provides a mapping between the mobile target detection problem for a stochastic sensor network and the line-set intersection problem. Hence, the conclusion is that both problems are equivalent. □

The mapping stated in Lemma 1 between Problem 2 and Problem 3, provides a bijection between the physical sensor network domain and the geometric domain. In the next section, a solution for the probability of detection of a target by a single sensor in the geometric domain will be provided since it is previously shown that Problem 2

and Problem 3 can be reduced to each other. Finally, a closed-form solution for the probability of detection in Problem 1 will be provided.

5.4. Deployment Quality of a Border Surveillance WSN

The probability of target detection P_D can be evaluated using the frequency count of lines intersecting geometric shapes. As we consider the set of all possible linear trajectories intersecting the deployment area, P_D is equal to the quotient of the number of lines that intersect any of the sensing areas, over the number of lines that intersect the deployment area. However, the set of lines in the plane intersecting a set is uncountable.

Hence, an analytical solution to the Line-Set intersection problem using the line measure will derived. Let $G(p, \phi)$ be a straight line in any plane. Here, the density of the line is formulated as

$$dG = dp \wedge d\phi \tag{5.1}$$

The measure $m(G)$ of a set of lines $G(p, \phi)$ is the integral of the density of the line, which is in the differential form, over the set. Hence,

$$m(G) = \int dp \wedge d\phi \tag{5.2}$$

The measure of the set of lines that pass over a bounded convex set, C, defined by the support function $p = p(\phi)$ of the convex set, is given as [96]:

$$m(G : G \cap C \neq 0) = \int_{G \cap C \neq 0} dp \wedge d\phi = \int_0^{2\pi} p d\phi = L \tag{5.3}$$

where, L is the perimeter of C. Equation 5.3 can be reinterpreted that the measure of lines intersecting a convex set is equal to the perimeter of that convex set. By

Equation 5.3, the perimeters can in turn be used to calculate the intersection of a random line with a convex set C_i within C, which will be used for the proof of the following lemma.

Lemma 2. *Let S be the deployment area of a sensor network and let s_i be any of the sensors deployed in the area. The probability that an intruder randomly passing through S is detected by the single sensor, s_i is equal to*

$$P_i = \frac{L_i}{L}, \tag{5.4}$$

where L_i is the perimeter of the sensor coverage area and L is the perimeter of the deployment area.

Proof. The probability that an intruder randomly passing through S is detected by a single sensor, s_i, is derived as follows. By the mapping of the intruder detection problem to line-set intersection problem provided in Lemma 1, this probability is equivalent to the probability that a line G intersecting C, also intersects C_i. In terms of the measures, this probability is equal to the ratio of the measure of the set of lines that intersect both C and C_i to the measure of the set of lines that intersect C.

$$\begin{aligned} P_i &= pr[G \cap C_i \cap C \neq 0 | G \cap C \neq 0] \\ &= \frac{m(G : G \cap C_i \cap C \neq 0)}{m(G : G \cap C \neq 0)} \end{aligned} \tag{5.5}$$

Since C_i is known to be inside the convex set C, Equation 5.5 can be rewritten as

$$P_i = \frac{m(G : G \cap C_i \neq 0)}{m(G : G \cap C \neq 0)} \tag{5.6}$$

Using Equation 5.3, Equation 5.6 is equal to

$$P_i = \frac{L_i}{L} \tag{5.7}$$

□

Thus, using the mapping provided in Lemma 1, the probability that an intruder randomly passing through S is detected by a single sensor, s_i, is found in terms of L, the perimeter of deployment area and L_i, the perimeter of the sensing coverage of a sensor. Now, it is needed to derive the effect of jammers on the probability of detection in the sensor network, i.e., derive a solution for the *border surveillance intruder detection avoiding jammers problem* (Problem 1).

Theorem 1. *The DQM of any sensor network with N_s sensors deployed in the area and N_h jamming regions is*

$$P_D = \sum_{k=0}^{N_s} \sum_{v=1}^{\left|Z_{N_s,k}\right|} \left(1 - \prod_{i=1}^{k} \left(1 - \frac{L_{\mathbf{z_{k,v}}(i)}}{L}\right)\right) P_a^k \left(1 - P_a\right)^{N_s - k} \tag{5.8}$$

and

$$P_a = \prod_{j=1}^{N_h} \left(1 - \frac{A_j}{A}\right) \tag{5.9}$$

where $Z_{N_s,k}$ denotes the set of all $\binom{N_s}{k}$ vectors, $\mathbf{z_{k,v}}$, containing ordered and distinct k elements of vector $[1, \ldots, N_s]$. A denotes the area of the deployment area, S and A_j, j = 1,...,N_h denotes the area of the jamming region, H_j.

The necessary derivations and the related proofs can found in detail in our manuscript [97].

If sensors have sensing areas with perimeters of equal length (not necessarily identical shapes) and the areas of each jamming region is the same, Equation 5.8 can

be simplified to the following form.

Corollary 1. *When all sensors deployed in the network have sensing areas of equal perimeters L_s, the DQM of the network is equal to:*

$$P_D = 1 - \left(1 - \frac{L_s}{L} \cdot P_a\right)^{N_s} \tag{5.10}$$

Moreover, if all jamming regions have the same area, A_h, then the probability that a sensor is alive (P_a) is equal to:

$$P_a = \left(1 - \frac{A_h}{A}\right)^{N_h} \tag{5.11}$$

In all formulations, we only require the sets to be convex and bounded. However for simplicity, we assume circular sensing coverage, jamming coverage and deployment area.

Corollary 2. *If the sensors have identical circular sensing coverage and jamming regions are circular and identical, then the DQM of a sensor network with a circular deployment area is equal to:*

$$P_D = 1 - \left(1 - \frac{R_s}{R} \cdot P_a\right)^{N_s} \tag{5.12}$$

and

$$P_a = \left(1 - \left(\frac{R_h}{R}\right)^2\right)^{N_h} \tag{5.13}$$

where R_s is the radius of the sensing coverage of a sensor, R_h is the radius of the jamming coverage and R is the radius of the deployment area.

5.5. Analytical Results

In this section, a comparison between the analytical and the simulation results are provided to present the validity of the analytical estimation. In order to observe the effects of individual parameters in the model, the following assumptions are made to minimize the effect of external parameters on our results.

- A binary sensing model is assumed, where for a given sensor s located at z_s, probability of detection $P_D(s)$ of a point x is:

$$P_D(s) = \begin{cases} 1, & \textit{if } d(x, z_s) < R_s \\ 0, & \textit{otherwise} \end{cases} \tag{5.14}$$

- A circular sensing range is assumed, on a 2-D flat world deployment region.
- Intruders follow a straight line while they are trespassing the detection area.
- Communication operations between the sensors are involved only in the determination of the hole sizes.
- Sensing equipments of sensors are awake and able to detect the intruders at all times.
- Holes are circular inside the deployment region, due to jamming or destruction of sensors. In either case, the sensors residing in a hole are inoperable.

Our simulations are designed to include different scenarios based on various sets of the parameters involved in the analytical model and are implemented in MATLAB [98]. We evaluated the effects of these parameters for 100 random network deployments and 100 random jamming area placements per simulation test case, where each deployment is tested with each placement. For each (deployment, placement) pair, 1000 random intrusion paths are generated and the probability of detection is evaluated using the frequency count of lines passing through the sensors that are alive, i.e., outside the jamming areas as seen in Figure 5.1. The results presented in each graph are the mean values of the results obtained for all (deployment, placement) pairs.

Table 5.1. Circular deployment region scenario test parameters.

Area Radius (m)	2000, 3000, 4000, $*$5000, 6000, 7000, 8000
Jamming Area Radius (m)	50, 100, 150, $*$200, 250, 300, 350
Jamming Area Count	0, 10, $*$20, 30, 40, 50, 60, 70, 80
Sensor Count	$*$500, 600, 700, 800, 900

$*$ These are the default values used in the simulations.

Area shape has a very drastic effect on the performance of wireless sensor networks due to inherent limitations induced by the shape. Hence, it is aimed to present the effects of the DQM parameters on the probability of detection for two types of deployment sites in our scenarios: circular region and rectangular region. In our first scenario, sensors are deployed over a circular region. Circular region is introduced to model a circular deployment site such as a basin or a junction point, on which such surveillance tasks can be performed. The parameter set for the circular deployment scenario is given in Table 5.1. More extensive tests with sensing radius changes and lower sensor counts are also performed and the results can be found in our manuscript [97].

Our second scenario is designed with the assumption of rectangular deployment areas to observe their effects on the DQM values. Rectangular region is introduced to model a border that forms a long strip along a river bank or by a mountain. Security and surveillance tasks are more generally performed on such deployment sites all over the world. Hence, the model is tested on a similar, border-like deployment site to perform more realistic simulations. The parameter set for the rectangular deployment scenario is given in Table 5.2.

In both scenario types, the values indicated with the $*$ sign are used as the default values for fixed parameters in the simulations. Sensing ranges and jamming ranges are assumed to be circular areas. In addition, the jamming area radii are uniformly distributed in the range $[0.8 \times value,\ 1.2 \times value]$, where $value$ is the jamming area radius parameter value used in the simulation. That is, in order to be more realistic, the jamming area radii are not exactly equal to, but are close to the parameter value

Table 5.2. Rectangular deployment region scenario test parameters.

Region Width (m)	2000
Region Height (m)	4000, 6000, 8000, 10000, 12000, ∗14000, 16000
Jamming Area Radius (m)	50, 100, 150, ∗200, 250, 300, 350
Jamming Area Count	0, 10, ∗20, 30, 40, 50, 60, 70, 80
Sensor Count	∗500, 600, 700, 800, 900

∗ These are the default values used in the simulations.

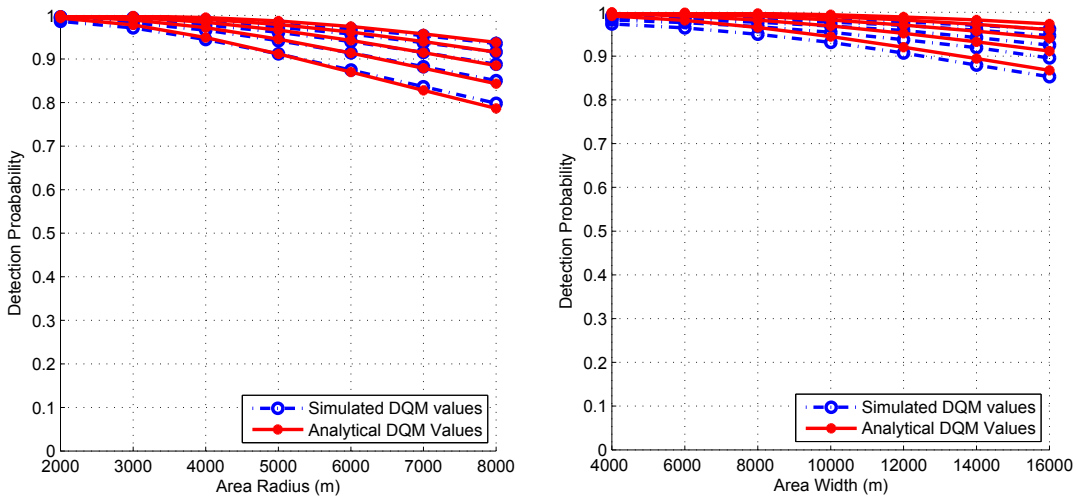
used in the simulation.

In the following subsections, the joint effects of the simulation parameters on a deployed sensor network are investigated, together with a comparison of the simulation and analytical results. To emphasize the effects of these parameters on the DQM values, some reference DQM values are provided in the feasible operation range of a sample SWSN, which requires the DQM value to be above 90 per cent.

5.5.1. The combined effect of area size and sensor count on DQM values

In this test case, the area size and the sensor count parameters are changed according to the values given in the parameter sets, while other parameters are fixed to observe the combined effect of these parameters on probability of detection. Sensor deployment quality values for these test cases are presented in Figure 5.3 for different sensor counts. In the circular scenario, the gap between the analytical and simulated DQM values is at most 1.1 per cent for different sensor counts, whereas the gap is at most 1.8 per cent in the rectangular scenario. Hence, the narrow gap between analytical and simulation results indicate that the provided DQM is a close estimate of the simulated detection probability in the given cases.

The area size is observed to be inversely proportional to the probability of detection, as indicated in the DQM formulation. Under the given assumptions and default



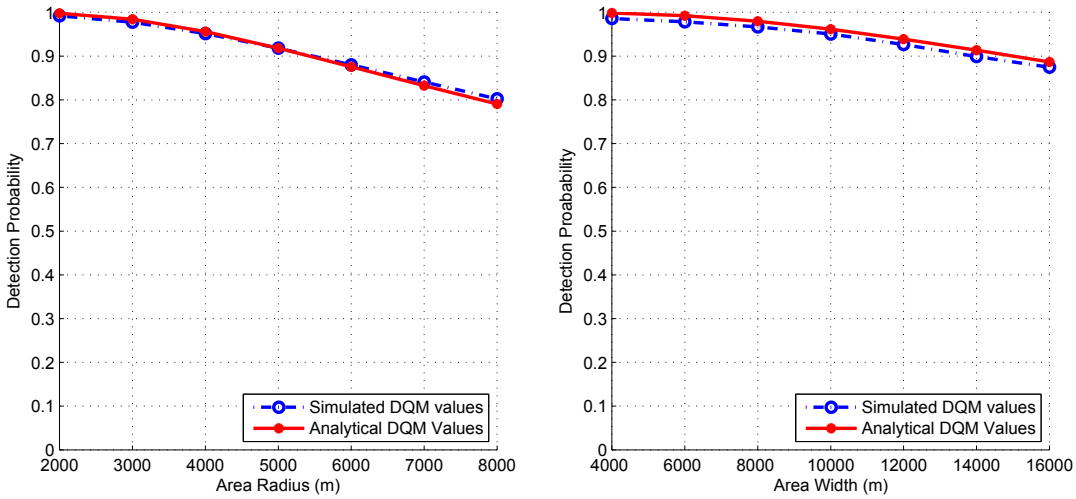
(a) Circular deployment region scenario. (b) Rectangular deployment region scenario.

Figure 5.3. Effect of area size and sensor count on analytical and simulated DQM values. Lines are ordered from the greatest sensor count value (900) at the top to the smallest value (500) at the bottom.

parameter values, the feasible range of operation, where the DQM value is above 90 per cent, requires the sensor count to be above 800 in a circular region with a 8000 m radius and above 600 in a 2000×14000 m^2 rectangular region. Similarly, for a sensor count of 400, a circular area up to 4000 m radius and a rectangular area up to 1000 m width can be monitored for a feasible DQM.

5.5.2. The combined effect of area size, jamming area radius and jamming area count on DQM values

In order to see the combined effect of the area size and the jamming area radius, a separate test case is designed. In this test, the area size and the jamming area radius parameters are changed according to the values given in the parameter sets, while other parameters are fixed. The results for various jamming area radii are observed to be overlapping. Hence, the results for the jamming area radius of 50 m is presented in Figure 5.4 and all results are tabulated in Table 5.3 for the circular scenario and in Table 5.4 for the rectangular scenario. In the circular scenario, the gap between the analytical and simulated DQM values is at most 1.7 per cent, whereas the gap is at



(a) Circular deployment region scenario. (b) Rectangular deployment region scenario.

Figure 5.4. Effect of area size for the jamming area radius of 50 m on analytical and simulated DQM values. The remaining results for are tabulated in Table 5.3 for the circular scenario and in Table 5.4 for the rectangular scenario.

most 3.2 per cent in the rectangular scenario.

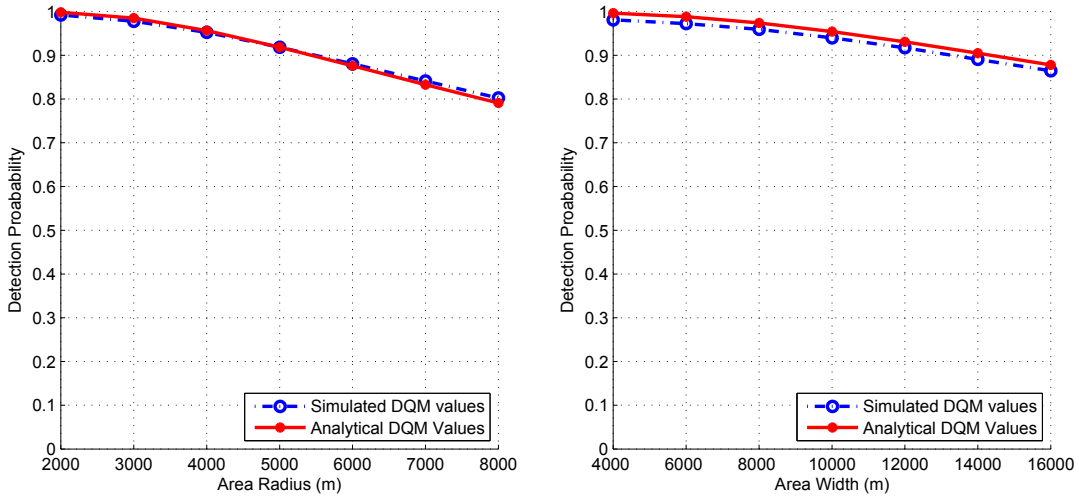
To provide a more detailed analysis of the combined effect of jamming areas and area size on the probability of detection, another test case is designed, in which the area size and the jamming area count parameters are changed according to the values given in the parameter sets, while other parameters are fixed. Similar to the jamming area radius case, the results for various jamming area counts are also observed to be overlapping. Hence, the results when there are no jamming areas is presented in Figure 5.5 and all results are tabulated in Table 5.5 for the circular scenario and in Table 5.6 for the rectangular scenario. In the circular scenario, the gap between the analytical and simulated DQM values is at most 1.6 per cent, whereas the gap is at most 2.7 per cent in the rectangular scenario. These results show that the DQM values can be utilized as a close estimate of the detection probability of the simulated networks for the given test cases. In addition, we observe that an increase in the jamming area radius and jamming area count have similar effects on the sensing performance of the network.

Table 5.3. Analytical and simulated DQM values for the areas with different radii and jamming area radii. Circular deployment region scenario version.

Jamming	**2000 m**		**3000 m**		**4000 m**		**5000 m**		**6000 m**		**7000 m**		**8000 m**		**9000 m**	
Area Rad.	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**
50 m	0.992	0.998	0.977	0.984	0.952	0.956	0.918	0.918	0.881	0.876	0.841	0.833	0.802	0.791	0.766	0.751
100 m	0.991	0.998	0.976	0.983	0.951	0.955	0.917	0.917	0.879	0.875	0.840	0.832	0.802	0.790	0.765	0.750
150 m	0.990	0.996	0.975	0.981	0.948	0.953	0.915	0.915	0.877	0.873	0.839	0.830	0.800	0.789	0.763	0.749
200 m	0.987	0.994	0.971	0.978	0.946	0.949	0.912	0.912	0.875	0.870	0.836	0.828	0.798	0.787	0.762	0.748
250 m	0.981	0.990	0.966	0.974	0.941	0.945	0.909	0.908	0.871	0.867	0.834	0.825	0.797	0.785	0.761	0.746
300 m	0.969	0.981	0.960	0.967	0.935	0.939	0.903	0.903	0.869	0.863	0.830	0.822	0.794	0.782	0.758	0.743
350 m	0.948	0.966	0.949	0.958	0.928	0.932	0.898	0.897	0.862	0.858	0.826	0.818	0.791	0.778	0.755	0.741

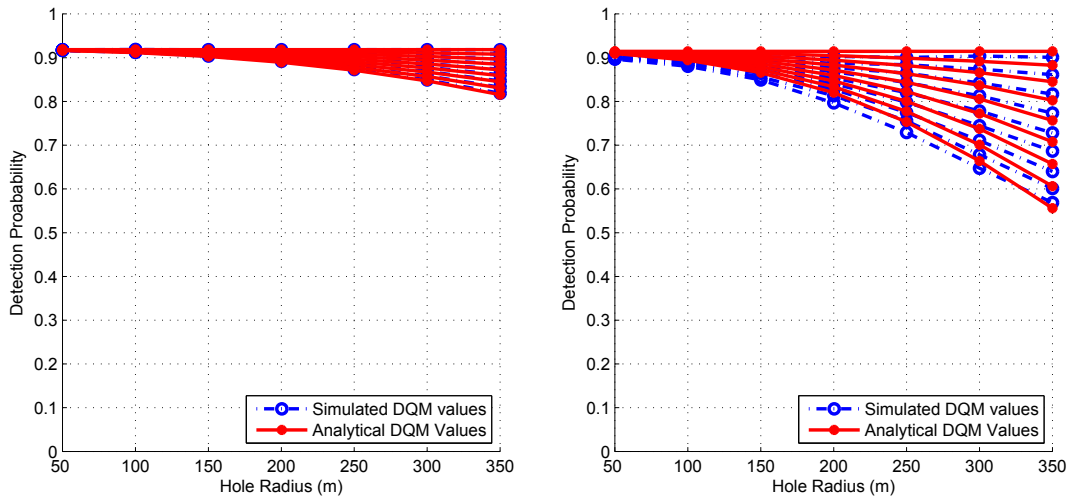
Table 5.4. Analytical and simulated DQM values for different area sizes and jamming area radii. Rectangular deployment region scenario.

Jamming	**8 km^2**		**12 km^2**		**16 km^2**		**20 km^2**		**24 km^2**		**28 km^2**		**32 km^2**	
Area Rad.	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**
50 m	0.9859	0.9984	0.9785	0.9923	0.9667	0.9798	0.9506	0.9615	0.9264	0.9389	0.8989	0.9134	0.8745	0.8865
100 m	0.9832	0.9977	0.9760	0.9907	0.9637	0.9774	0.9474	0.9585	0.9221	0.9354	0.8975	0.9098	0.8675	0.8828
150 m	0.9816	0.9960	0.9730	0.9875	0.9590	0.9729	0.9396	0.9530	0.9158	0.9294	0.8902	0.9036	0.8646	0.8765
200 m	0.9724	0.9917	0.9638	0.9815	0.9502	0.9655	0.9302	0.9446	0.9059	0.9205	0.8797	0.8944	0.8528	0.8674
250 m	0.9562	0.9816	0.9513	0.9710	0.9350	0.9539	0.9140	0.9324	0.8893	0.9080	0.8632	0.8820	0.8357	0.8552
300 m	0.9279	0.9591	0.9271	0.9529	0.9121	0.9365	0.8931	0.9152	0.8693	0.8912	0.8461	0.8657	0.8170	0.8396
350 m	0.8814	0.9135	0.8892	0.9230	0.8795	0.9110	0.8650	0.8918	0.8420	0.8692	0.8184	0.8450	0.7948	0.8202



(a) Circular deployment region scenario. (b) Rectangular deployment region scenario.

Figure 5.5. Effect of area size for no jamming area case on analytical and simulated DQM values. The remaining results for are tabulated in Table 5.5 for the circular scenario and in Table 5.6 for the rectangular scenario.



(a) Circular deployment region scenario. (b) Rectangular deployment region scenario.

Figure 5.6. Effect of jamming area radius and jamming area count on analytical and simulated DQM values. Lines are ordered from the smallest jamming area count value (0) at the top to the greatest value (80) at the bottom.

Table 5.5. Analytical and simulated DQM values for the areas with different radii and jamming area counts. Circular deployment region scenario version.

Jamming	**2000 m**		**3000 m**		**4000 m**		**5000 m**		**6000 m**		**7000 m**		**8000 m**	
Area Cnt.	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**
0	0.9920	0.9981	0.9778	0.9848	0.9521	0.9565	0.9184	0.9184	0.8805	0.8760	0.8411	0.8329	0.8024	0.7909
10	0.9915	0.9978	0.9773	0.9840	0.9511	0.9556	0.9175	0.9176	0.8799	0.8753	0.8405	0.8322	0.8017	0.7904
20	0.9911	0.9975	0.9761	0.9833	0.9507	0.9548	0.9167	0.9168	0.8789	0.8746	0.8399	0.8316	0.8016	0.7899
30	0.9904	0.9971	0.9757	0.9825	0.9497	0.9539	0.9160	0.9159	0.8780	0.8739	0.8395	0.8310	0.8014	0.7894
40	0.9900	0.9966	0.9751	0.9817	0.9487	0.9530	0.9149	0.9151	0.8776	0.8731	0.8388	0.8304	0.8003	0.7888
50	0.9892	0.9961	0.9739	0.9809	0.9476	0.9521	0.9141	0.9143	0.8770	0.8724	0.8380	0.8298	0.7998	0.7883
60	0.9887	0.9955	0.9732	0.9800	0.9472	0.9512	0.9136	0.9134	0.8760	0.8717	0.8374	0.8292	0.7996	0.7878
70	0.9877	0.9949	0.9723	0.9791	0.9460	0.9502	0.9126	0.9126	0.8757	0.8709	0.8368	0.8286	0.7989	0.7873
80	0.9865	0.9942	0.9712	0.9782	0.9451	0.9493	0.9118	0.9117	0.8746	0.8702	0.8366	0.8279	0.7982	0.7868

Table 5.6. Analytical and simulated DQM values for different area sizes and jamming area counts. Rectangular deployment region scenario version.

Jamming	**8 km^2**		**12 km^2**		**16 km^2**		**20 km^2**		**24 km^2**		**28 km^2**		**32 km^2**	
Area Cnt.	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**
0	0.9854	0.9986	0.9788	0.9928	0.9673	0.9806	0.9502	0.9625	0.9290	0.9400	0.9011	0.9146	0.8745	0.8877
10	0.9810	0.9964	0.9723	0.9882	0.9593	0.9738	0.9397	0.9541	0.9171	0.9307	0.8904	0.9048	0.8645	0.8778
20	0.9734	0.9917	0.9635	0.9815	0.9499	0.9655	0.9292	0.9446	0.9057	0.9205	0.8782	0.8944	0.8527	0.8674
30	0.9604	0.9832	0.9529	0.9725	0.9386	0.9554	0.9185	0.9339	0.8946	0.9095	0.8671	0.8834	0.8386	0.8567
40	0.9430	0.9694	0.9401	0.9605	0.9262	0.9435	0.9028	0.9219	0.8797	0.8976	0.8538	0.8718	0.8258	0.8455
50	0.9210	0.9489	0.9240	0.9454	0.9097	0.9297	0.8908	0.9087	0.8680	0.8849	0.8399	0.8597	0.8157	0.8339
60	0.8940	0.9210	0.9046	0.9270	0.8919	0.9139	0.8749	0.8943	0.8523	0.8714	0.8285	0.8470	0.8035	0.8220
70	0.8641	0.8853	0.8796	0.9051	0.8757	0.8963	0.8590	0.8787	0.8358	0.8572	0.8134	0.8338	0.7877	0.8097
80	0.8213	0.8424	0.8568	0.8799	0.8559	0.8768	0.8414	0.8620	0.8213	0.8422	0.7999	0.8201	0.7744	0.7971

The jamming area radius and the jamming area count parameters are observed to be inversely proportional to the probability of detection, as indicated in the DQM formulation. To illustrate the joint effect of the area size and the jamming area on the DQM values, sample values in the feasible operation range of a typical SWSN are provided. Under the given assumptions and default parameter values, the feasible range of operation requires the jamming area radius to be below 300 m in both a circular region with a 5000 m radius and a 2000×10000 m^2 rectangular region. We can observe that for a deployment area radius above 6000 m, a reliable network operation with sufficient DQM cannot be achieved for the given jamming area counts. When the size of the jamming area radii are known to be approximately 200 m (considering the current test case values), a circular deployment area up to 5000 m radius and a rectangular deployment area up to 12000 m width is observed to be monitored properly for a feasible DQM (i.e., $P_D > 0.9$).

Similarly, when the jamming area radius is set to be 200 m, all of the tested jamming area count values satisfy the feasibility constraint of the DQM in a circular region with a 5000 m radius. The number of jamming areas is required to be below 50 in a 2000×10000 m^2 rectangular region. The simulation results show that for a deployment area radius above 6000 m, a reliable network operation with sufficient DQM cannot be achieved for the given jamming area counts. When the number of the jamming areas in a deployment site is 30, a circular deployment area up to 5000 m radius and a rectangular deployment area up to 12000 m width is observed to be monitored properly for a feasible DQM (i.e., $P_D > 0.9$).

5.5.3. The combined effect of jamming area radius and jamming area count on DQM values

In the previous section, we observed that the jamming area radius and the jamming area count parameters have similar effects on the DQM value. In order to analyze these effects in detail, a separate test case involving only the properties of the jamming areas is designed. In this test case, the jamming area radius and the jamming area count parameters are changed according to the values given in the parameter sets,

while other parameters are fixed. The DQM values for these test cases are presented in Figure 5.6. In the circular scenario, the gap between the analytical and simulated DQM values is at most 0.3 per cent, whereas the gap is at most 2.9 per cent in the rectangular scenario. These results show that the DQM values can be utilized as a close estimate of the detection probability of the simulated networks for the given test cases.

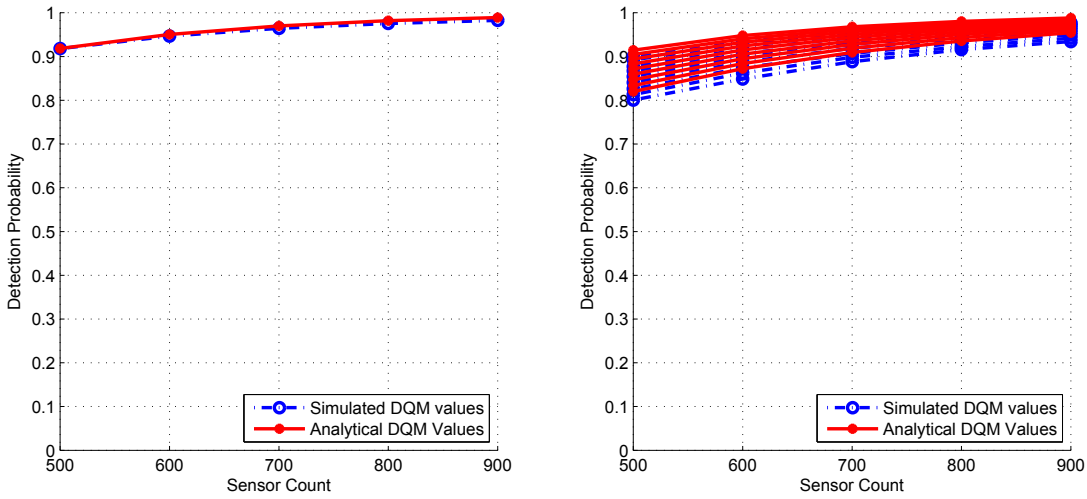
In this section, we justify our observations about the effect of the jamming area on the DQM value. The jamming area radius and count parameters are observed to be inversely proportional to the probability of detection. For low jamming area values, deployment quality loss is not very high, however the effect is clear when jamming areas occupy more than 20 per cent of the network. Under the given assumptions and default parameter values, the feasible range of operation, where the DQM value is above 90 per cent, requires the jamming area count to be below 50 in the circular scenario and below 10 in the rectangular scenario, where the jamming are radius is set to be 200 m. Similarly, if the number of jamming areas is given as 30, the jamming area radius is required to be below 250 m for the circular scenario and below 100 m for the rectangular scenario for a feasible DQM (i.e., $P_D > 0.9$).

5.5.4. The combined effect of sensor count and jamming area count on DQM values

In this section, we designed a separate test case to present the joint effect of the sensors and jamming areas on the DQM value. In this test case, the sensor count and the jamming area count parameters are changed according to the values given in the parameter sets, while other parameters are fixed to observe the combined effect of these parameters on the probability of detection. Figure 5.7 presents the sensor deployment quality values for the dense networks, where the sensor count is between 500 and 900 nodes. The results for various jamming area counts are observed to be overlapping in the circular scenario. Hence, the results, when there are no jamming areas, is presented in Figure 5.7.a. The related results are tabulated in Table 5.7.

Table 5.7. Analytical and simulated DQM values for the areas with different sensor counts and different jamming area counts. High number of sensors, circular deployment region scenario version.

Jamming	**500**		**600**		**700**		**800**		**900**	
Area Cnt.	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**	**Sim.**	**An.**
0	0.9181	0.9184	0.9466	0.9506	0.9646	0.9701	0.9754	0.9819	0.9827	0.9890
10	0.9154	0.9151	0.9448	0.9482	0.9631	0.9683	0.9746	0.9807	0.9817	0.9882
20	0.9119	0.9117	0.9424	0.9457	0.9615	0.9666	0.9733	0.9794	0.9810	0.9873
30	0.9085	0.9082	0.9398	0.9431	0.9597	0.9647	0.9721	0.9781	0.9804	0.9864
40	0.9049	0.9047	0.9376	0.9404	0.9574	0.9628	0.9707	0.9767	0.9790	0.9855
50	0.9016	0.9010	0.9350	0.9377	0.9560	0.9608	0.9694	0.9753	0.9780	0.9844
60	0.8980	0.8973	0.9323	0.9349	0.9541	0.9587	0.9677	0.9738	0.9769	0.9834
70	0.8947	0.8935	0.9295	0.9320	0.9520	0.9565	0.9665	0.9722	0.9759	0.9823
80	0.8912	0.8897	0.9269	0.9290	0.9493	0.9543	0.9648	0.9706	0.9745	0.9811



(a) Circular deployment region scenario for no jamming area case. The remaining results for are tabulated in Table 5.7.

(b) Rectangular deployment region scenario. Lines are ordered from the smallest jamming area count value (0) at the top to the greatest value (80) at the bottom.

Figure 5.7. Effect of high sensor count and jamming area count on analytical and simulated DQM values.

In the circular scenario, the gap between the analytical and simulated DQM values is at most 0.7 per cent for different sensor counts, whereas the gap is at most 2.1 per cent in the rectangular scenario. These results show that the DQM values can be utilized as a close estimate of the detection probability of the simulated networks for the given test cases.

As we figure out from the previous test cases, the sensor count parameter is observed to be proportional, whereas the jamming area count parameter is observed to be inversely proportional to the probability of detection. In this test case, we observe the same behavior of the DQM values for the given parameters. To meet the required sensing quality, the jamming area count is required to be below 50 for the circular scenario and below 100 for the rectangular scenario, if the sensor count is set to be 500. Similarly, for a jamming area count of 60, the sensor count is required to be above 600 for the circular scenario and above 700 for the rectangular scenario for a feasible DQM (i.e., $P_D > 0.9$).

6. Conclusions

Nonuniform deployment of sensors over a terrain may be the result of the underlying physical constraints, the deployment method, sensing or communication requirements over the terrain, sensor losses or the external factor such as jammers, destructions. Under nonuniformity criteria, it is of the highest importance to keep the network alive for as long as possible. In this thesis, we proposed methods to prolong the network lifetime by optimal placements and critical sensor deployment at required locations and times. Also an analytical method to understand the sensing quality under lossy situations is proposed.

For especially border surveillance networks, sink placement is a crucial operation for a WSN. The sink location must be both safe and lifetime optimal. These two fact can present a trade-off. Safe locations increases the routing paths from sensors to the sink and locations that provide shorter routes may be in the relatively unsafe points of the deployment region. The deployment regions are mostly not flat and sensors are obstructed by the terrain. Given such restrictions, a hybrid GA based heuristic, namely GAUSS is presented to find the suitable and relatively better sink locations. The heuristic makes use of discrete event simulation to find the fitness value of the candidates. The fitness value is a weighted lifetime metric that favors safer locations. Results are presented for deployments in a flat terrain and a 3D terrain. Results for different cases are presented to show the efficiency of GAUSS. The results show that GAUSS does present better results than placing the sink at center of mass coordinates or at the borderline. Based on the parameters, lifetime gains between 20 per cent and 100 per cent are possible using GAUSS compared to the other heuristics.

Another issue for nonuniform deployment is the formation of bottleneck traffic patterns. For nonuniform deployments, the communication traffic follows an uneven pattern, creating considerably high loads on some particular nodes. The areas that include the highly loaded nodes become the bottleneck areas soon after the network becomes operational, decreasing the lifetime of the network. Prolonging the lifetime

of the overall network is possible by identification of bottleneck areas at the earliest stage possible and deploying new sensors over those areas. In this thesis, BAIA, a geometric clustering method, is proposed to locate the bottleneck areas and perform small redeployments over those areas to increase the lifetime values of the networks. The experiments with different scenarios show that the algorithm increases the lifetime considerably on the average.

Similar to bottleneck nodes, holes are inherent in sensor networks with adverse impacts. For especially SWSN, holes soon cripple and render the whole network unusable. In this thesis, a redeployment based scheme that uses image processing operations, namely IDeA, is proposed to delay the hole formations as much as possible. Simulation results showed that it possible to have more than 100 per cent lifetime gains for border surveillance networks using the methods presented.

The critical sensor network applications such as the border surveillance and target detection applications require a certain level of sensing capability, which is indicated by the deployment quality of the network, to maintain a feasible network operation. However, various factors such as the existence of jammers and sensor deaths, combined with the physical properties of the deployed sensor network affect this quality. An analytical and closed form formulation for the deployment quality, which takes the loss of sensors into consideration, is highly usable regarding the real life sensor applications of the industry and the simulation of sensor applications in the research community. In this thesis, a metric for this deployment quality is proposed, which is analogous to the probability of detection in the network. A formulation for this quality metric which integrates the stated factors is presented. The proposed formulation can be used for different types of border surveillance scenarios, where one or more physical properties may differ from one to another. Being in closed form, it is scalable and suitable for different types of sensor network operations.

The effects of network properties are extensively analyzed running various parameter test cases, to conclude that the proposed DQM provides realistic results, with a deviation of ± 3.2 per cent with the simulation results. It is observed that the an-

alytical results reflect the effects of the sensing coverage loss due to jamming, sensor node death and node destructions on the deployment quality.

As future work, we plan to extend IDeA to 3D terrains to make it more realistic and robust for real use in hole mitigation. Moreover, the analytical deployment metric that we proposed assumes binary sensing model for simplicity. We plan to modify the formulation to provide a high quality metric when the sensing model is probabilistic.

REFERENCES

1. Al-Karaki, J. N. and A. E. Kamal, "Routing Techniques In Wireless Sensor Networks: A Survey", *IEEE Wireless Communications*, December, 2004.

2. Akyildiz, I., T. Melodia and K. Chowdhury, "A survey on wireless multimedia sensor networks", *Computer Networks*, 51 (4), pp. 921–960, 2007.

3. Balinksi, M. L., "On finding integer solutions to linear programs.", *Proc. IBM Scientific Computing Symp. on Combinatorial Problems*, pp. 225–248, IBM, 1966.

4. Drezner, Z. and H. W. Hamacher, "Facility Location: Applications and Theory", Springer, 2002.

5. Stann, F. and J. Heidemann, "BARD: Bayesian-Assisted Resource Discovery In Sensor Networks", *Proc. of the IEEE INFOCOM 2005*, Miami, Florida, USA, March, 2005.

6. Das, A. and D. Dutta, "Data acquisition in multiple-sink sensor networks", *Proc. of 2nd Int. Conf. on Embedded Networked Sensor Systems*, Baltimore, MD, USA, 2004.

7. Intanagonwiwat, C., D. Estrin, R. Govindan and J. Heidemann, "Impact of Network Density on Data Aggregation in Wireless Sensor Networks", *Technical Report*, 01-750, University of Southern California Computer Science Department, November, 2001.

8. Handziski, V., A. Köpke, H. Karl, C. Frank and W. Drytkiewicz, "Improving the Energy Efficiency of Directed Diffusion Using passive Clustering", *Proc. of 1st European Workshop on Wireless Sensor Networks*, Volume 2920 of LNCS, Berlin, Germany, January, 2004.

9. Simon, R. and E. Farrugia, "Topology transparent support for sensor networks", *Proc. of 1st European Workshop on Wireless Sensor Networks*, Volume 2920 of LNCS, Berlin, Germany, January, 2004.

10. Gnawali, O., M. Yarvis, J. Heidemann and R. Govindan, "Interaction of Retransmission, Blacklisting and Routing Metrics for Reliability in Sensor Network Routing", *Proc. of the First IEEE Conference on Sensor and Adhoc Communication and Networks*, pp. 34–43, Santa Clara, California, USA, 2004.

11. Yu, Y., B. Krishnamachari and V.K. Prasanna, "Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks", *IEEE INFOCOM*, March, 2004.

12. Zhou, C. and B. Krishnamachari, "Localized Topology Generation Mechanisms for Self-Configuring Sensor Networks", *IEEE Globecom*, December, 2003.

13. Li, W. and C.G. Cassandras, "A Minimum-Power Wireless Sensor Network Self-Deployment Scheme", *IEEE Wireless Communications and Networking Conference*, 2004.

14. Xing, G., Chenyang L., Ying Z., Qingfeng H. and R. Pless, "Minimum Power Configuration in Wireless Sensor Networks", *Proc. of the 6th ACM International Symposium on Mobile ad hoc Networking and Computing*, 2005.

15. Eu, Z.-A., H.-p. Tan and W. K. G. Seah, "Routing and Relay Node Placement in Wireless Sensor Networks powered by Ambient Energy Harvesting", *IEEE WCNC*, April, 2009.

16. Cristescu, R., B. Beferull-Lozano, M. Vetterli and R. Wattenhofer, "Network Correlated Data Gathering with Explicit Communication: NP-Completeness and Algorithms", *IEEE/ACM Transactions on Networking*, 2004.

17. Ganesan, D., R. Cristescu and B. Beferull-Lozano, "Power-efficient Sensor Placement and Transmission Structure for Data Gathering Under Distortion Con-

straints", *3rd International Symposium on Information processing in Sensor Networks*, IPSN 2004, April, 2004.

18. Maleki, M. and M. Pedram, "QoM and Lifetime-Constrained Random Deployment of Sensor Networks for Minimum Energy Consumption", *Proc. of IPSN* 2005.

19. Oyman, E.I. and C. Ersoy, "Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks", *Proc. of the IEEE International Conference on Communications*, Paris, France, 2004.

20. Perillo, M. and W. Heinzelman, "DAPR: A Protocol for Wireless Sensor Networks Utilizing an Application-based Routing Cost", *Proc. of the IEEE Wireless Communications and Networking Conference 2004*, March, 2004.

21. Solis, I. and K. Obraczka, "In-network aggregation trade-offs for data collection in wireless sensor networks", *INRG Technical Report*, 102, 2003.

22. Akkaya, K. and M. Younis, "Energy–Aware Routing to a Mobile Gateway in Wireless Sensor Networks", in *Proc. of the IEEE Globecom Wireless Ad Hoc and Sensor Networks Workshop*, Dallas, TX, November, 2004.

23. Ye, F., H. Luo, J. Cheng, S. Lu and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks", *MOBICOM*, August, 2002.

24. Kim, H.S., T. Abdelzaher and W. H. Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks", *ACM SenSys*, Los Angeles, CA, November, 2003.

25. Marta, M. and M. Cardei, "Using Sink Mobility to Increase Wireless Sensor Networks Lifetime", *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Newport Beach, CA, USA, 2008.

26. Saad, L. B. and B. Tourancheau, "Multiple Mobile Sinks Positioning in Wireless Sensor Networks for Buildings", *SENSORCOMM, 2009 Third International Con-*

ference on Sensor Technologies and Applications, Greece, June 18-23, 2009.

27. Wu, X. and G. Chen, "Dual-Sink: Using Mobile and Static Sinks for Lifetime Improvement in Wireless Sensor Networks", *IEEE ICCCN 2007*, Honolulu, Hawaii, USA, August, 2007.

28. Yang, C.-H. and L.-W. Lin, "Adaptive Base-Station Placement Approach to Alleviate Sink Hotspots in Wireless Sensor Networks" *The 20-th International Conference on Information Management*, Taipei, Taiwan, 2009.

29. Poe, W. Y. and J. B. Schmitt, "Placing Multiple Sinks in Time-Sensitive Wireless Sensor Networks using a Genetic Algorithm", *14th GI/ITG Conference - Measurement, Modelling and Evalutation of Computer and Communication Systems*, Dortmund, Germany, 2008.

30. Guney, E., İ. K. Altinel, N. Aras and C. Ersoy, "A Variable Neighborhood Search Heuristic for Point Coverage, Sink Location and Data Routing in Wireless Sensor Networks", in *Proc. of the 2009 Second International Conference on Communication Theory, Reliability, and Quality of Service*, pp. 81-86, Colmar, France, 2009.

31. Holland, J.H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, 1975.

32. Telfar, G., *Generally Applicable Heuristics for Global Optimisation: An Investigation for Algorithm Performance for the Euclidian Traveling Salesman problem*, MS. Thesis, Victoria University of Wellington, Wellington, 1994.

33. Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison–Wesley Publishing Company Inc, 1989.

34. Beasley, D., D.R. Bull and R.R. Martin,"An Overview of Genetic Algorithms: Part 1. Fundamentals", *University Computing*, Vol. 15, pp. 58–69, 1993.

35. Bandyopadhyay, S. and Coyle, E.J., "An Energy Efficient Hierarchical Clustering

Algorithm For Wireless Sensor Networks", *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, (INFOCOM 2003)*, Vol. 3, pp. 1713–1723, 2003.

36. Chan, H. and A. Perrig, "ACE: An Emergent Algorithm for Highly Uniform Cluster Formation", *European Workshop on Sensor Networks*, pp. 154–171, 2004.

37. Chen, Y., A. Liestman and J. Liu, "Clustering Algorithms for Ad Hoc Wireless Networks", *Ad Hoc and Sensor Networks*, 2004.

38. Flake, G. W., R. E. Tarjan and K. Tsioutsiouliklis, "Graph Clustering and Minimum Cut Trees", *Internet Mathematics*, Vol. 1, No. 4, pp. 385–408, 2003.

39. Jennings, E., L. Motyckova and D. Carr, "Evaluating Graph Theoretic Clustering Algorithms for Reliable Multicasting", *Proc. of IEEE GLOBECOM 2001*, San Antonio, 2001.

40. Motyckova, L. and D. Carr, "Locality Issues in Reliable Multicasting", *Proc. of the 6th APCC*, Seoul, South Korea, 2000.

41. Wang, K., S. Abu Ayyash, T.D.C. Little, and P. Basu, "Attribute-Based Clustering for Information Dissemination in Wireless Sensor Networks", in *Proc. 2nd Annual IEEE Communications Society Conf. on Sensor and Ad Hoc Communications and Networks (SECON 2005)*, Santa Clara, CA, September 2005.

42. Younis, O. and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad-hoc Sensor Networks", *IEEE Transactions on Mobile Computing*, Vol. 3, No. 4, pp. 366–379, 2004.

43. Krishnan, R. and D. Starobinski, "Efficient Clustering Algorithms for Self-Organizing Wireless Sensor Networks", *Journal of Ad-Hoc Networks*, Vol. 4, No. 1, pp. 36–59, January, 2006.

44. Li, J. and P. Mohapatra, "An analytical model for the energy hole problem in

many-to-one sensor networks", *IEEE Vehicular Technology Conference 2005 (VTC)*, vol. 4, 2005.

45. Pottie, G. and W. Kaiser, "Wireless integrated network sensors", *Communications of the ACM*, vol. 43, no. 5, pp. 51–58, 2000.

46. Ahmed, N., S. S. Kanhere and S. Jha, "The holes problem in wireless sensor networks: a survey", *Mobile Computing and Communications Review*, vol. 9, no. 2, pp. 4–18, 2005.

47. Li, M. and B. Yang, "A survey on topology issues in wireless sensor networks", in *ICWN*, Las Vegas, Nevada, USA, June 2006.

48. OPNET, *Opnet Technologies INC*, 2009.

49. Ganesan, D., A. Cerpa, W. Ye, Y. Yu, J. Zhao and D. Estrin, "Networking issues in wireless sensor networks", *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, pp. 799–814, 2004.

50. Ali, M., A. Dunkels, K. Römer, K. Langendoen, J. Polastre and Z. Uzmi, "Medium access control issues in sensor networks", *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 33–36, 2006.

51. Kosar, R., E. Onur and C. Ersoy, "Redeployment based sensing hole mitigation in wireless sensor networks", in *IEEE WCNC*, April, 2009.

52. Arifler, D., "Information theoretic approach to detecting systematic node destructions in wireless sensor networks", *IEEE Transactions on Wireless Communications*, vol. 7, no. 11, part 2, pp. 4730–4738, 2008.

53. Howard, A., M. Mataric and G. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem", *Distributed Autonomous Robotic Systems*, vol. 5, pp. 299–308, 2002.

54. Chen, J., S. Li and Y. Sun, "Novel deployment schemes for mobile sensor networks", *Sensors*, vol. 7, pp. 2907–2919, 2007.

55. Wang, Y., J. Gao and J. S. B. Mitchell, "Boundary recognition in sensor networks by topological methods", in *Proc. of ACM MobiCom*, 2006.

56. Kim, B., J. Kim and T. Cho, "Advanced redeployment point determining method in sensor networks", *International Journal of Computer Science and Network Security*, vol. 7, no. 11, pp. 21–25, November 2007.

57. Altınel, K. I., N. Aras, E. Güney and C. Ersoy, "Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks", *Computer Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

58. Chiang, M. and G. Byrd, "Neighborhood-aware density control in wireless sensor networks", *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08*, pp. 122–129, 2008.

59. Olariu, S. and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting", *IEEE INFOCOM*, 2006.

60. Yu, F., E. Lee, Y. Choi, S. Park, D. Lee, T. Ye and S.-H. Kim, "A modeling for hole problem in wireless sensor networks", in *IWCMC '07: Proc. of the 2007 International Conference on Wireless Communications and Mobile Computing*, pp. 370–375, New York, NY, 2007.

61. Jia, W., T. Wang, G. Wang and M. Guo, "Hole avoiding in advance routing in wireless sensor networks", *IEEE Wireless Communications and Networking Conference, 2007 (WCNC)*, pp. 3519–3523, 2007.

62. Fang, Q., J. Gao and L. Guibas, "Locating and bypassing routing holes in sensor networks", *Mobile Networks and Applications*, vol. 11, pp. 187–200, 2006.

63. Funke, S., "Topological hole detection in wireless sensor networks and its applications", *DIALM-POMC*, 2005.

64. Vieira, L. F. M., M. A. M. Vieira, L. R. Beatriz, A. A. F. Loureiro, D. C. da S. Junior and A. O. Fernandes, "Efficient incremental sensor network deployment algorithm", *SBRC Brazilian Symposium on Computer Networks*, 2004.

65. Yang, Y. and M. Cardei, "Movement-assisted sensor redeployment scheme for network lifetime increase", *The 10th ACM/IEEE Intl. Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, October, 2007.

66. Chatzigiannakis, I., A. Kinalis and S. Nikoletseas, "Adaptive energy management for incremental deployment of heterogeneous wireless sensors", *Theory of Computing Systems*, July, 2007.

67. Wu, X., G. Chen and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution", *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 5, pp. 710–720, May, 2008.

68. Onur, E., C. Ersoy, H. Delic and L. Akarun, "Coverage in sensor networks when obstacles are present", in *Proc. of the IEEE ICC*, June, 2006.

69. Elfes, A., "Occupancy grids: A stochastic spatial representation for active robot perception", in *Proc. of the Sixth Conference on Uncertainty in AI*, pp. 60–70, 1990.

70. Vincent, L. and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, June 1991.

71. Gonzalez R. and R. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992.

72. Ye, W., J. Heidemann and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks", in *Proc. of the IEEE Infocom*, vol. 3, 2002.

73. Kwon, S. and N. Shroff, "Paradox of shortest path routing for large multi-hop wireless networks", in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pp. 1001–1009, 2007.

74. Chipcon Datasheet, Chipcon AS, http://www.snm.ethz.ch/pub/uploads/Projects /CC1000 datasheet.pdf, 04 2002.

75. Kosar, R., I. Bojaxhiu, E. Onur and C. Ersoy, "Energy Hole Mitigation Techniques for Surveillance Wireless Sensor Networks", *Submitted to IEEE Transactions on Vehicular Technology*.

76. Yick, J., B. Mukherjee and D. Ghosal, "Wireless sensor network survey", *Computer Networks*, 52 (12) pp. 2292–2330, 2008.

77. Guo, S., T. He, M. Mokbel, J. A. Stankovic and T. Abdelzaher, "On accurate and efficient statistical counting in sensor-based surveillance systems", in *Proc. of the 5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'08)*, pp. 24–35, Atlanta, Georgia, 2008.

78. Lu, G., B. Krishnamachari and C. S. Raghavendra, "An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks", *Wireless Communications and Mobile Computing*, 7 (7), pp. 863–875, 2007.

79. Oh, S., P. Chen, M. Manzo and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance", in *Proc. of the 2006 IEEE International Conference on Robotics and Automation (ICRA 2006)*, pp. 3128–3133, 2006.

80. Bokareva, T., W. Hu, S. Kanhere, B. Ristic, N. Gordon, T. Bessell, M. Rutten and S. Jha, "Wireless sensor networks for battlefield surveillance", in *Proc. of the Land Warfare Conference*, Brisbane, Queensland, Australia, 2006.

81. Lazos, L., R. Poovendran and J. A. Ritcey, "Analytic evaluation of target detection in heterogeneous wireless sensor networks", *ACM Transactions on Sensor Networks*,

5 (2), pp. 1–38, 2009.

82. Cao, Q., T. Yan, J. A. Stankovic and T. F. Abdelzaher, "Analysis of target detection performance for wireless sensor networks", in *Proc. of the Distributed Computing in Sensor Systems*, LNCS, Vol. 3560, pp. 276–292, Springer, Marina del Rey, CA, USA, 2005.

83. Clouqueur, T., K. K. Saluja and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection", *IEEE Transactions on Computers*, 53 (3), pp. 320–333, 2004.

84. Dousse, O., C. Tavoularis and P. Thiran, "Delay of intrusion detection in wireless sensor networks", in *Proc. of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'07)*, pp. 155–165, Florence, Italy, 2006.

85. Wang, Y.-C., K.-Y. Cheng and Y.-C. Tseng, "Using event detection latency to evaluate the coverage of a wireless sensor network", *Computer Communications*, 30 (14-15), pp. 2699–2707, 2007.

86. Ren, S., Q. Li, H. Wang, X. Chen and X. Zhang, "Analyzing object detection quality under probabilistic coverage in sensor networks", in *Proc. of the International Workshop on Quality of Service (IWQoS 2005)*, LNCS, Vol. 3552, pp. 107–122, 2005.

87. Wang, Y., X. Wang, D. P. Agrawal and A. A. Minai, "Impact of heterogeneity on coverage and broadcast reachability in wireless sensor networks", in *Proc. of the 2006 International Conference on Computer Communications and Networks (ICCCN'06)*, pp. 63–67, Arlington, VA, 2006.

88. Saipulla, A., C. Westphal, B. Liu and J. Wang, "Barrier coverage of line-based deployed wireless sensor networks", in *Proc. of the 28th IEEE Conference on Computer Communications (INFOCOM 2009)*, pp. 127–135, Rio de Janeiro, Brazil, 2009.

89. Niu, R., P. K. Varshney and Q. Cheng, "Distributed detection in a large wireless sensor network", *Information Fusion*, 7 (4), pp. 380–394, 2006.

90. Xu, W., W. Trappe, Y. Zhang and T. Wood, "The feasability of launching and detecting jamming attacks in wireless networks", in *Proc. of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC'05)*, pp. 46–57, Urbana-Champaign, IL, USA, 2005.

91. Ngai, E. C. H., J. Liu and M. R. Lyu, "An efficient intruder detection algorithm against sinkhole attacks in wireless sensor networks", *Computer Communications*, 30 (11-12), pp. 2353–2364, 2007.

92. Li, M., I. Koutsopoulos and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks", in *Proc. of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, pp. 1307–1315, 2007.

93. Cagalj, M., S. Capkun and J.-P. Hubaux, "Wormhole-based antijamming techniques in sensor networks", *IEEE Transactions On Mobile Computing*, 6 (1), pp. 100–114, 2007.

94. Chen, X., K. Makki, K. Yen and N. Pissinou, "Attack distribution modeling and its applications in sensor network security", *EURASIP Journal on Wireless Communications and Networking*, 2008, pp. 1–11, 2008.

95. Li, X. and D. K. Hunter, "Distributed coordinate-free hole recovery", in *Proc. of 13th IEEE International Conference on Communications Workshops (ICC Workshops '08)*, pp. 189–194, 2008.

96. Santalo, L. A., "Integral Geometry and Geometric Probability", Addison-Wesley Publishing, 1976.

97. Dönmez, M.Y., R. Kosar and C. Ersoy, "An Analytical Approach to De-

ployment Quality of Surveillance Wireless Sensor Networks Considering the Effect of Jammers and Energy Holes", *Computer Networks* (to appear), http://dx.doi.org/10.1016/j.comnet.2010.07.007, 2010.

98. MATLAB, *The MathWorks Inc.*, 2009.