Resilient Wireless Sensor Networks

A thesis submitted in fulfilment of the
requirements for the award of the degree

Doctor of Philosophy

from

University of Technology, Sydney
Faculty of Engineering and Information Technology

by

Ali Rafiei

Bachelor Science of Engineering, 2004
Master Science of Engineering, 2007
Shiraz University

Sydney, Australia
2017

Statement of Authorship/Originality

I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis.

Production Note:
Signature removed prior to publication.

Abstract

With the increase in wireless sensor networks' (WSN) applications as the result of enhancements in sensors' size, battery-life and mobility, sensor nodes have become one of the most ubiquitous and relied-upon electrical appliances in recent years. In harsh and hostile environments, in the absence of centralised supervision, the effects of faults, damages and unbalanced node deployments should be taken into account as they may disturb the operation and quality of service of networks. Coverage holes (CHs) due to the correlated failures and unbalanced deployment of nodes should be considered seriously in a timely manner; otherwise, cascaded failures on the rest of the proximate sensor nodes can jeopardise networks' integrity. Although different distributed topology control (TC) schemes have been devised to address the challenges of node failures and their dynamic behaviours, little work has been directed towards recovering CHs and/or alleviating their undesirable effects especially in Large Scale CHs (LSCH). Thus, devising CH recovery strategies for the swift detection, notification, repair and avoidance of damage events are important to increase the lifetime and resiliency of WSNs and to improve the efficacy and reliability of error-prone and energy-restricted nodes for many applications. In this research, the concepts of resiliency, fault management, network holes, CHs, TC schemes and stages of CH recovery are reviewed. By devising new TC techniques, CHs recovery strategies that partially or wholly repair LSCHs and increase the coverage of WSNs are presented such that a global pattern emerges as a result of nodes' local interactions.

In this study, we propose **(1)** CH detection and boundary node (B-node) selection algorithms, which B-nodes around the damaged area self-select solely based on available 1-hop information extracted from their simple geometrical and statistical features. **(2)** A constraint node movement algorithm based on the idea of virtual chord (v-chords) formed by B-nodes and their neighbours to partially repair CHs. By changing each B-node's v-chord, its movement and connectivity to the rest of network can be controlled in a distributed manner. **(3)** Fuzzy node relocation models based on force-based movement algorithms are suitable to consider the uncertainty governed by nodes' distributed and local interactions and the indefinite choices of movements. **(4)** A model of cooperative CHs recovery in which nodes move towards damaged areas in the form of disjoint spanned trees, which is inspired by nature. Based on nodes' local interactions with their neighbours and their distances to CHs, a set of disjoint trees around the CH spans. **(5)** A hybrid CH recovery strategy that combines sensing power control and physical node relocation using a game theoretic approach for mobile WSNs. **(6)** A sink-based CH recovery via node relocation where moving nodes consider the status of sink nodes.

The proposed node relocation algorithm aims to reduce the distances of moving nodes to the deployed sink nodes while repairing the CHs. The results show that proposed distributed algorithms **(1)-(6)** either outperform or match their counterparts within acceptable ranges.

The significances of proposed algorithms are as follow: Although they are mainly designed base on the available 1-hop knowledge and local interactions of (autonomous) nodes, they result in global behaviours. They can be implemented in harsh and hostile environments in the absence of centralised operators. They are suitable for time-sensitive applications and scenarios with the security concerns that limit the amount of information exchange between nodes. The burden of decision making is spread among nodes.

Acknowledgments

First, I would like to express my deepest appreciations to my supervisor, Dr. Mehran Abolhasan who helped me in my research and who has always been supportive and very patient through my studies despite all the difficulties. I would like to thank my co-supervisor, Dr. Daniel Franklin who gave me valuable feedback and comments during my research. I would also like to thank my previous supervisor Prof. Farzad Safaei from University of Wollongong, Australia for all his feedbacks and continuous supervision even after i transferred to University of Technology, Sydney. I am very grateful for the unique opportunity to work with him. I would like express my gratitude to my alternative supervisor, Prof. Robin Braun for all his kindness and help. I would like to thank Dr. Stephen Smith from Macquarie University for his valuable time and feedback on reading my work and accepting to review the drafts version of my thesis.

I would like to thank my beloved parents, Mohammad and Shahla, for their unlimited support during my studies. Without them it was not possible at all to pass through the hard times. I really appreciate their sacrifice and support for me every single day of my life. I would like to thank my brother and my best friend, Dr. Pedram Rafiei from University of Saskatchewan, Canada who always helped me any possible ways he could from the long distance.

I would like to thanks people who I co-authored a number of papers for the duration of my studies. Special Thanks to Dr. Yashar Maali from University of Technology, Sydney for his help. I also thank my colleagues especially Dr. Abdullah Alsabbagh for his time and creating enjoyable and cooperative environment for me. I would like to express my special thanks to Dr. Wei Ni from Commonwealth Scientific and Industrial Research Organisation (CSIRO) for his invaluable feedback and comments on my work. It was my pleasure to work with him. I would like to thank my former colleague Dr. Banafsheh Lashkari, from Shiraz University for her comments and feedback at the time of preparing the thesis' draft.

I am in debt to anonymous reviewers of my publications for giving feedback and comments which paved the way for new ideas in my research. I would also like to express my appreciation for Australian Research Council (ARC) discovery research grant No. DP0879507 to support my studies and ease the financial burden.

I would like to express my gratitude to Mr. Amir Hesami, Dr. Mohammad Namazi, Dr. Amirreza Niktash, Dr.Reza Memary, Dr. Alireza Banani, Dr. Alireza Mohammadi, Dr. Mehdi Farrokhrooz and Dr. Mostafa Azizpour among many of my friends for their kind hospitality and support.

I would like to express my sincere appreciation for Ms. Phyllis Agius and Mr. Craig Shuard for all their help and advices. I would like to express my gratitude to Ms. Eryani Tjondrowalujo and Ms. Grandia Agathon for their kindness and help with the thesis submission process.

I would like to thank my thesis examiners for their time and for their valuable comments and feedback.

Last but not least, I would like to dedicate my thesis to all the courageous, brave people, and the soldiers of my country which have sacrificed their lives and health to protect my country, my homeland against enemies and made a secure and safe place to grow and prosper. I salute and pray for their souls and respectable families who have suffered drastically from losing their loved ones.

To my beloved and very dear parents, my true friends Mohammad, and Shahla which have supported me in all aspects and were for me every single moment.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ADL	Absolute Degree Loss
AVD	Average Distance from Damaged Area
aM-CSSCMM	M-CSSCMM based on the amplitude
B-B cell	B-Node Voronoi Cell Neighbour with B-node Voronoi Cell
B-D cell	B-Node Voronoi Cell Neighbour with D-node Voronoi Cell
B-node	Boundary Node
BNS-Algorithm	Boundary Node Selection Algorithms
CA	Cellular Automata
CCP	coverage configuration protocol
CH	Coverage Hole
CHR	Coverage Hole Recovery
CHD	Coverage Hole Detour
CI	Confidence Interval
CLAvgB	Cosine Law Average Boundary
CLMinB	Cosine Law Min Boundary
CM	Center of Mass
C-Mobility	Controlled Mobility
CNF-algorithm	Closer Nodes First Algorithm
CP	Candidate Parent

CSSCMM	Combined SS and CM Movement
D-area	Damaged Area
D-event	Damage Event
D-node	Damaged Node
DHSCL	Distributed Homogeneous Synchronous Coverage Learning Algorithm
dM-CSSCMM	M-CSSCMM based on the angle
DN-node(s)	Damaged Neighbour Node(s)
DSN	Distributed Sensor Network
DSSA	Distributed Self-Spreading Algorithm
DS-Tree	Disjoint Spanned Tree
DUCM-Algs.	DN-nodes and UN-Nodes' Centre of Mass Algorithm
EC	Emergent Cooperation
ECE	Efficiency of Consumed Energy
FAM	Fuzzy Angular Movement
FARM	FAM then FRM
FRM	Fuzzy Radial Movement
FRAM	FRM then FAM
FRNAM	FRM and FAM
GPS	Global Positioning System
HBD	Hole Boundary Detection
LSCH	Large Scale Coverage Hole
k-RSC	k-redundant sensing coverage
MANETS	Mobile ad hoc networks
MAS	Multi-Agent System
MaxD-algorithm	Maximum Distance Algorithm
MinD-algorithm	Minimum Distance Algorithm

MB-node(s)	Margin of Boundary node(s)
M-CSSCMM	Modified CSSCMM
MWSNs	Mobile WSNs
N-node	Normal Node
NE	Nash equilibrium
Perc-k-RSC	percentages of k-redundant sensing coverage
PCov	Percentage of Coverage
PP	Potential Parent
PSO	Particle Swarm Optimization
QoS	Quality of Service
RBN	Random Boundary node
RDL	Relative Degree Loss
R_c	Transmission Range
R_s	Sensing Range
RE	Residual Energy
ROI	Region Of Interest
QB	Quantile-based (comparative) Boundary
S-Coverage	Spatial Coverage
SB-Node	Selected Boundary Node
SS	Simple Sink
SSCH	Small Scale Coverage Hole
SSM	Simple Sink Movement
T-Coverage	Temporal Coverage
TC	Topology Control
TS	Takagi and Sugeno
UAV	Unmanned aerial vehicle
U-node	Undamaged Node

UDG	Unit Disk Graph
U-Mobility	Uncontrolled Mobility
UN-node	Undamaged Neighbour Node
V-chord	Virtual Chord
V-Hole	Virtual Hole
V-node	Virtual Node
VA-Algorithm	Voronoi Area Algorithm
VarD-Algorithm	Variance of Distance Algorithm
VD-Algorithm	Voronoi Distance Algorithm
WQB	Weighted quantile-based Boundary
WSN	Wireless Sensor Network

CHAPTER 1

Introduction

Wireless Sensor Networks (WSNs) have an unprecedented rate of growth in popularity over the past decade, due to their diverse range of applications and miniaturisation of electronic circuitry in WSN devices [1–3]. For example, WSNs present solutions for remote health monitoring [4], decentralised environmental monitoring [5–7] for wildfire [8, 9], tsunamis [10], earthquakes [11], volcanoes [12] as well as measuring global temperature and CO_2 levels in air [13].

Therefore, presently there are considerable research activities in the field of wireless sensor networks [1, 2, 14] as they can be used to sense, measure, monitor and track events and phenomena, especially in hostile and hazardous environments where the use of a centralised controller is impractical or impossible. Numerous technological advancements have resulted in compact, low cost devices with significant communication and computation capabilities, transforming sensors from simple measurement and detection devices to intelligent, autonomous systems that are able to process data and make decisions flexibly in response to environmental conditions.

Despite these enhancements, it is not possible to protect all sensor nodes from possible failures and damages due to the nature of their finite battery life, specific applications and environments (e.g. detection/monitoring of forest fire [8]) without greatly increasing network costs and complexity [15, 16]. Therefore, in order to overcome the inherently unreliable complexion of sensors, they are typically redundantly deployed in large numbers across the *region of interest* (ROI) either in a regular geometric pattern or with a uniformly random distribution [17–21]. However, node failures can potentially cause a disturbance to the uniformity of such random deployments or degrade the regularity of predetermined and structured geometric node distributions.

As a result of gradual exhaustion of stored energy, software/hardware faults, physical damage to sensor nodes, suboptimal initial node deployments, physical obstacles and/or uncontrolled node movement [22,23], wireless sensor networks may fail to cover a given area, even though nodes are redundantly deployed to compensate for their limited range and reliability. Therefore, inadequate coverage gives rise to *coverage holes* (CHs) [24] with different sizes and scales. Coverage holes caused by correlated node failures may not only disturb the performance and *quality of service* (QoS) of network in the proximate areas to the en masse failures but may also jeopardise the integrity of the entire network if not properly and promptly addressed. This is because it is likely that CHs cause more domino effects and cascaded failures especially in their immediate vicinity due to local-minima phenomena and increased traffic caused by stateless greedy routing algorithms as data packets travel along the boundaries of coverage holes [24–26].

One main reason to respond and react to dynamic network changes such as network failures, damages, coverage holes, imbalanced deployment and data traffic in a timely manner is to maximise the useful lifetime of the network and make it resilient to unpredictable factors by either preserving energy or maintaining network service to an acceptable level until proper additional repair and reinforcement is supplied from the outside by operators.

Even if they are not primarily designed to deal with coverage holes, there are many different *topology control* (TC) strategies that can be utilised, either alone or in combination to repair or alleviate holes' direct and indirect effects on WSNs [27–34]. Among these solutions, active *node relocation and mobility* seems to be the most promising and has been shown to be able to help maintain the coverage and connectivity of networks if applied properly. However, it is also possible for these strategies to instead lead to newly-formed coverage holes and topological instability [21, 22, 35–40]. Therefore, node mobility is a double-edged sword in that it may either enhance or degrade network performance and quality of service, especially in terms of coverage and connectivity. Mobility may be classified as one of two types: *uncontrollable* mobility (U-mobility) and *controllable* mobility (C-mobility) [22, 23, 41]. In the U-mobility, node movements are affected by the environments (such as flow or waves for sensors on the sea) and are not controllable, while in C-mobility, the movement of nodes are controlled to compensate for gradual drift or holes caused by the nodes' U-mobility.

Recent literature has focused on distributed (as opposed to centralised) node movement schemes in order to cope with the widest possible variety of environmental conditions and to support the requirements of emerging applications despite limited battery life and sensor range. Fully autonomous node movement decision-making based on the node's surroundings (environment and node neighbourhood) appear to be the most popular approach to the distributed automatic repair of coverage holes in wireless sensor networks. However, due to the significant amount of energy required for node mobility and the fact that node mobility is solely for the purpose of improving sensing and communications (the primary functions of each node), design of relocation and movement strategies must give careful consideration to the balance between minimising energy consumption resulting from node movement, while maximising the benefit of that movement in terms of coverage and connectivity [39, 42–44].

Although it is not a complete solution to the problem of repairing coverage holes es, the idea of *cooperation* among nodes [28, 45–48] should be considered as an effective and stable mean to achieve desirable large scale topology control goals in the network while complying with constraints and limitations resulting from autonomous node movement, guided by distributed energy-aware relocation algorithms.

Therefore, the idea of *cooperative recovery* in general, and cooperative movement of mobile nodes in particular, can be defined as *evenly sharing the task of movement and relocation among the nodes in which each node contributes its share to the recovery process of networks*, with less or no additional information exchange among its neighbours. Based on this idea, nodes collaborate with each other within defined ranges by contributing their local movements to improve the global situation [49, 50]. The load of this task should either be evenly divided between the nodes or nodes may autonomously self-select to participate in prospective movements. Subsequently, the available energy and hence the functional lifetime of networks can be increased by such cooperation in the case of cooperative recovery, with limited and controlled mobility of nodes. The challenge and beauty of cooperation becomes most obvious in the case of *emergent cooperation* [49, 50], in which accumulation of blind behaviour and autonomous decision-making of nodes eventually results in global network behaviours, which enable the network to react to topology changes swiftly and flexibly. Design and implementation of the aforementioned cooperative network behaviours in real-time and time-sensitive scenarios and applications is challenging and an open research problem.

The idea of using models that account for the natural uncertainties in information possessed by autonomous nodes and their in-range neighbours, is important. If such models can accurately capture and predict node behaviours given these uncertainties, distributed algorithms can be applied and implemented more efficiently, robustly and with higher performance. For example, devising a model that reduces the amount of movements while increasing coverage in the deployed area by properly considering the uncertainty of virtual forces exerted by autonomous nodes, has shown to enhance the lifetime of the network [51]. Such model also allows the proposed distributed node relocation algorithms to be feasible in real scenarios with available resources of communication and computation [42, 51–54].

1.1 Objectives of the Thesis

The objective of this thesis is to propose and evaluate new models and strategies for making networks resilient to physical damage and node failures, especially coverage holes defined as en masse correlated failures of nodes [55, 56]. Firstly the concept of autonomous node self-selection around damaged areas is introduced to limit the number of nodes involved in cooperative hole recovery. Next, a model of cooperative recovery for coverage holes is proposed using local geometric chords formed between each boundary node and its selected in-range neighbours. Cooperative recovery based on collection of disjoint spanned trees is also introduced in this thesis, in which coverage holes are repaired by the movement of disjoint trees spanned based on the nodes' distance to the damage area through the trees spanned up to their parents. As one solution to improve network resilience, node relocation models based on fuzzy systems are proposed in this thesis. This allows network recovery and node relocation to be performed with a smaller amount of information exchanged between autonomous nodes. Finally, *cooperative emergent* *recovery* strategies for coverage holes are proposed, suitable for those applications and scenarios in which neither safe and secure communication among nodes nor unlimited energy for movement is available.

At present, an objective comparison of alternative techniques for active node relocation is challenging due to a number of factors. Firstly, although the concept of resiliency exists in general, not only in wireless networks but also in wired networks, clear and unambiguous definitions of network-wide resiliency are not yet established in the literature [56–63]. Secondly, as yet there are no widely accepted *standard performance metrics* for hole detection and recovery. Many of the relocation algorithms are geared towards balancing deployment and repairing small-scale coverage holes. Since many of these algorithms are primarily designed to apply to all nodes in the network, their efficacy and respective advantages and disadvantages when restricted to only a subset of nodes have not yet been examined. Therefore, adapting and refining suitable performance metrics to properly compare proposed approaches for detection and recovery of large scale coverage holes via selected node movement, is a key challenge.

1.2 Thesis Organisation

The thesis is organised as follows:

- *Chapter 2* presents a literature review that related work mainly in wireless sensor networks (WSNs), including network resiliency, event and faults, fault management, and various types of network failures such as coverage hole are discussed. Phases of the fault management and the stages of networks' failure recovery are presented. Different topology control schemes such as relocation of nodes are introduced. The concepts of cooperation, emergence and autonomy in the complex and multi-agent systems that can be applied to WSNs' are introduced.

- *Chapter 3* introduces the concept of *boundary node selection*, in which nodes at the proximity of coverage holes (boundary nodes) are self-selected for use in prospective recovery processes. Boundary selection is of great importance because not all boundary nodes in the immediate proximity of the damage event/phenomenon (i.e. the coverage hole) are necessarily required to be considered for possible participation in the prospective recovery processes. Therefore, node selection (similar to node retransmission scheduling) based on different criteria and parameters aims to reduce unnecessary redundancy, while conserving the network resources. Nodes should be carefully selected to be *active* around coverage holes in order to ensure that the overall lifetime of the network is maximised, and to avoid the inadvertent expansion of damaged regions through cascaded failures.

- *Chapter 4* presents an autonomous movement strategy for boundary nodes towards the coverage hole to repair the damaged area. The movement of nodes is fully autonomous and solely based on the knowledge of each node's neighbours. As each node forms a chord with its selected neighbour, called an α-chord, each node moves towards the coverage hole, and at the same time tries to maintain connectivity with its immediate neighbours. These movements result in emergent behaviours which contribute to the repair of the damaged area. By appropriately setting α-chord parameters for each node, the emergent behaviour of the recovery process can be modified into coverage hole avoidance, depending on the specific application.

- *Chapter 5* introduces a simple, fuzzy logic node relocation model in which each node exerts radial and angular forces on its neighbours as a function of their distance. Since in distributed node relocation models, nodes only have local visibility, decisions on the amount of movement for each node would be taken with a degree of uncertainty. Therefore, nodes may oscillate before reaching their final locations, which may unnecessarily exhaust their batteries. Hence, fuzzy movement models should be able to accurately predict future behaviour of neighbours and account for the uncertainties of neighbouring node movements in each iteration. This chapter is divided into three parts: in the first part, fuzzy parameter and membership functions are defined based on the expert knowledge of the scenario; in the second part, the required parameters are obtained by applying *particle swarm optimisation* (PSO) in different zone ranges around nodes in the first iteration; and in the third part, the fuzzy parameters obtained by PSO are modified and tuned with regards to nodes and the status of their in-range neighbours, not only for the first iteration but also for every movement iteration. These parameters are tuned based on the required performance metrics, such as percentage of coverage, uniformity, and average movement, such that these metrics, or a weighted combination of them, reach an optimal point within the assigned area.

- *Chapter 6* proposes a cooperative movement algorithm based on the disjoint spanned trees, such that each disjoint tree moves toward the coverage hole as the nodes in each tree follow their nearest immediate parents. By the sequential movement of the nodes in each tree, the deployment and connectivity of nodes is kept balanced, and at the same time the amount of exchanged information is limited. In this model, nodes are concerned only with their closest parents as they move towards the coverage hole.

- *Chapter 7* proposes a game-theoretic approach to recover the CHs in a distributed manner where sensors have little or no knowledge about other sensors' actions. The key idea is that a potential game between the sensors is formulated, where each mobile sensor in the network only requires local knowledge from its neighbouring nodes and takes CH recovery actions recursively with global convergence. Two actions of physical relocation and sensing range adjustment can be taken at each sensor in the proposed potential game. Appropriate actions can be taken to reduce the CHs in an energy efficient way. Simulation results show that the proposed potential game-theoretic approach is able to significantly increase network lifetime and maintain network coverage in the presence of random damage events when compared to the prior counterpart.

- *Chapter 8* explores how, unlike sporadic node failures, coverage holes emerging from multiple, temporally-correlated node failures can severely affect quality of service in a network and put the integrity of entire wireless sensor networks at risk. Conventional topology control schemes addressing such undesirable topological changes have usually overlooked the status of participating nodes in the recovery process with respect to the deployed sink node(s) in the network. In this chapter, a cooperative coverage hole recovery model is proposed which utilises the simple geometrical procedure of circle inversion. In this model, autonomous nodes consider their distances to the deployed sink node(s) in addition to their local status, while relocating towards the coverage holes. By defining suitable metrics, the performance of our proposed model performance is compared with a force-based approach.

- *Chapter 9* presents a general conclusion and discusses potential future research directions.

1.3 Summary of Contributions and Publications

The principal contributions discussed in this thesis are as follows (with relevant publications):

- Survey on coverage hole recovery
  - A. Rafiei, M. Abolhasan, D. R. Franklin, W. Ni, and F. Safaei, "A survey on coverage hole recovery in wsns," *To be submitted to IEEE Communications Surveys & Tutorials*, 2017.
- Boundary selection algorithms in which nodes autonomously self-select using simple geometrical properties deduced from their local visibility, which allow them to decide whether to participate in the prospective network recovery. Papers relevant to this idea include:
  - A. Rafiei, "Boundary node selection algorithms by simple geometrical properties in wsns," in *Fifth Asia Modelling Symposium (AMS), 2011*, May 2011, pp. 221–226.
  - A. Rafiei, M. Abolhasan, D. Franklin, and F. Safaei, "Boundary node selection algorithms in wsns," in *The 36th IEEE Conference on Local Computer Networks (LCN), 2011*, October 2011, pp. 255–258.
- Node deployment using uncertain model of fuzzy logic:
  - Y. Maali, A. Rafiei, M. Abolhasan, D. R. Franklin, and F. Safaei, "A fuzzy logic node relocation model in wsns," in *6th International Conference on Signal Processing and Communication Systems 12 - 14 December 2012, Gold Coast, Australia*, 2012, pp. 1–6.

- A. Rafiei, Y. Maali, M. Abolhasan, D. R. Franklin, F. Safaei, and S. Smith, "A tuned fuzzy logic relocation model in wsn using particle swarm optimization," in *2013 IEEE 78th Vehicular Technology Conference: VTC2013-Fall 2-5 September 2013, Las Vegas, USA*, September 2013.
- A. Rafiei, Y. Maali, M. Abolhasan, D. Franklin, and S. Smith, "An iteratively tuned fuzzy logic movement model in wsn using particle swarm optimization,"in *2013 7th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2013, pp. 1–7.

- Coverage hole recovery using $\alpha-$chords:
  - A. Rafiei, M. Abolhasan, D. Franklin, and F. Safaei, "Wsns coverage hole partial recovery by nodes' constrained and autonomous movements using virtual *alpha*-chords," in *2012 The Eighth International Conference on Wireless and Mobile Communications (ICWMC)*, 2012, pp. 74–80.

- Recovery of coverage holes using disjoint spanned trees:
  - A. Rafiei, M. Abolhasan, D. R. Franklin, and S. Smith, "A case study for choosing proper relocation algorithms to recover large scale coverage hole(s) in wireless sensor networks," in *1st Workshop on Advances in Real-time Information Networks, 7-10 August, Sydney, Australia.* epress UTS publishing, August 2013.
  - A. Rafiei, M. Abolhasan, D. R. Franklin, F. Safaei, S. Smith, and W. Ni, "Effect of the Number of Participating Nodes on Recovery of WSN Coverage Holes," *To Appear in 27th International Telecommunication Networks and Applications Conference*, November 2017.

  - A. Rafiei, M. Abolhasan, D. R. Franklin, F. Safaei, S. Smith, and W. Ni, “Cooperative Recovery of Coverage Holes in WSNs via Disjoint Spanning Trees,” *To Appear in 11th International Conference on Signal Processing and Communication Systems*, December 2017.

- Recovery of Coverage hole using Hybrid Topology control Schemes:
  - M. Abolhasan, Y. Maali, A. Rafiei, and W. Ni, “Distributed hybrid coverage hole recovery in wireless sensor networks,” in *IEEE Sensors Journal*, vol. 16, no. 23, pp. 8640–8648, Dec 2016.

- Recovery of Coverage hole in addition to considering the sink node:
  - A. Rafiei, Y. Maali, M. Abolhasan, and D. Franklin, “A geometrical sink-based cooperative coverage hole recovery strategy for wsns,” in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec 2015, pp. 1–8.

1.3.1 Contributions not covered in thesis

The following contribution is covered in this thesis, however, it has related material:

- A. A. S. Ali Rafiei, “A network recovery strategy scheme based on network failure scenarios and topologies,” in *International Conference on Communication and Vehicular Technology (ICCVT),Hanoi, Vietnam*, 2010.

CHAPTER 2

Related Work

It cannot be overlooked that different applications of wireless sensor networks (WSNs) to different areas of daily life are emerging. In order to have continuous and reliable services and resilient WSNs, different types of fault management mechanisms many of which mainly stem from those of the wired counterparts, should be properly taken into account. Fault recovery and/or avoidance, as part of fault management mechanisms, can benefit from an arsenal of devised topology control schemes to sustain networks operational to acceptable levels. The migration from centralised to distributed topology control schemes has enabled networks to use a set of fragile, limited resources, and error-prone sensor nodes to be given more autonomy in decision making, less reaction time and lower traffic among nodes. As the result of such migration, global behaviour and cooperation in WSNs fault management emerge. In this chapter, a review of network fault management and recovery of networks under fault mainly due to physical node failure is provided. The concepts of autonomy and cooperation as in WSNs is also presented.

In this chapter, different concepts of network resiliency, resiliency in WSNs, topology control schemes, network faults and events, fault management techniques, and

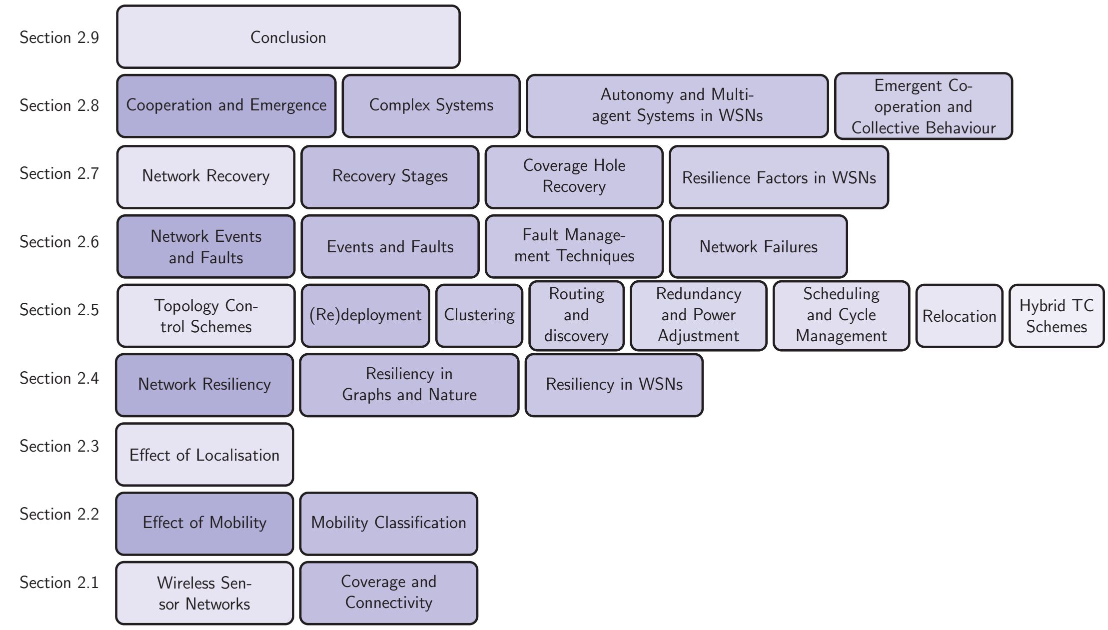


Figure 2.1: Stack Diagram of Sections

different types of network failures, are comprehensively presented. The definition of network recovery and its stages in both wired and wireless networks are also presented.

In order to avoid the unnecessary redundancy of providing a wide spectrum of previous surveys, the focus is more on presenting an overall picture along with required concepts. Finally, possible challenges and future directions are explored. A stack diagram of the chapter's sections is shown in the Figure 2.1.

2.1 Wireless Sensor Networks

Recently, WSNs have attracted the attention of industry and academia as they have become a suitable candidate in sensing, measuring, monitoring and tracking events and different phenomena, especially in the harsh and hostile environments in the absence of central supervision [1,2,14,64–67]. A wide range of improvements in size, battery life, manufacturing costs, communication/computation power in addition to the introduction of new features such as mobility and autonomy into the sensor nodes of different sensing models (i.e. binary, probabilistic, omni-directional, and directional [65,68,69]) has enabled WSNs to be used effectively and flexibly in applications with no infrastructures. Yet far from being fault-free and reliable, sensor nodes are becoming ubiquitous in appliances and devices in the diverse environments [9,10,22,45,65,70–86]. Therefore, in order to efficiently harness the potentials of WSNs, different challenges and constraints with respect to nodes' limitations (e.g. energy consumption, limited bandwidth, heterogeneity and fault tolerance), their environments of deployment, their types (if equipped with different types of sensors) in addition to WSNs' topology changes, scalability, reliability and targeted applications should be addressed [2, 10, 45, 65, 66, 70, 72, 87–91].

Forster et al. [67, 92, 93] considered WSNs' challenges mainly as *topology changes and mobility*, *harsh environments*, *wireless ad hoc medium*, *resource limitation*. Madarasz et al. [94] presented a definition of *environment* as a collection of objects from the group considered as indirect elements that do not belong to the WSN infrastructure, but play an essential role in the behaviour of network. Forster et al. [67] divided sensor nodes into *'passive'* and *'active'* where the former passively sense and measure their environments without affecting it, while the latter are required to actively influence their environments in order to measure them; for example, by emitting light or sound waves. Thus, active sensors' energy consumptions are expected to be much higher than passive sensors. The authors [67] also considered the special case of human as a sensor which used in crowd-sourcing and participatory sensing applications such as a person taking pictures of events or writing a review of a restaurant. Similarly, the authors [67] considered *actuators*, which, instead of sensing or measuring a phenomenon, they manipulate their environment. The concepts of distributed and mobile sensor networks are briefly introduced here.

2.1.1 Distributed WSNs

Distributed systems are defined by Yera Gomez [95] as *'a set of nodes, connected by a network, which appear to its users as a single coherent system'*.
Iyengar et al. [64] defined *a distributed sensor network* (DSN) as

> '*A set of spatially scattered intelligent sensors designed to obtain measurements from the environment, to abstract relevant information from data gathered, and to derive appropriate inferences from the information gained'*.

One of the characteristics of DSN is that they operate over the large areas for long periods of time and with minimal human supervision via stationary and/or mobile sensor nodes, such that their sensors may communicate, yet their topology is unlikely to be fixed or known, and communication bandwidth will be limited (Nicholson et al. [96]).

2.1.2 Mobile WSNs

By adding mobility to sensor nodes and reducing costs, WSNs are able to address some of challenges and be applied in many emerging applications. A survey by Rezazadeh [97] gives an overviews of *mobile wireless sensor networks* (MWSNs) for emerging applications, summarises the challenges of bringing mobility into WSNs and suggests solutions to address problems such as energy consumption and adapting to different environments (for example hostile and dynamic environments). Rezazadeh [97] considered the challenges of static sensor nodes in harsh and dynamic environments, of which some are lack of complete sensing fields coverage after the initial deployment, increased the possibility of the network being partitioned into several isolated subnets (islands); dynamic changes in the face of different obstacles; and node failures due to damage and/or exhaustion of error-prone and battery limited nodes. This is because the nodes' capabilities and flexibilities can be increased by embedding mobility in them to address the aforesaid challenges and enable nodes to perform multiple missions under various circumstances. It should be noted however that, MWSNs lead to emerging issues and problems [97]. The author [97] distinguishes between the *mobile ad hoc networks* (MANETs) and MWSNs in the sense of arbitration and intention of mobilities in the former and the latter networks respectively in which the movements are applied in the controlled manner in MWSNs. Rezazadeh [97] classified different mobile nodes based on their roles in networks, as *Mobile Embedded Sensor* which has no control over its own movements; rather, its motion is directed by some external force; *Mobile Actuated Sensor* which has locomotion abilities to move in the sensing region and can be used to increase the coverage; *Data Mule* which is able to collect data and deliver it to base stations are assumed to recharge automatically; and *Access Point* which positions itself to conserve connectivity as a network access point. Paper [97] also considered mobile base stations as a solution to the problem of surrounding nodes' energy consumptions and failures that are able to increase the channel capacities and reduce the length of paths (i.e. number of hops) to the destinations.

Paper [97] enumerated mobile WSNs challenges as: *localisation*, as it is not easy to keep a record of continuously relocating nodes in deployment area, requiring additional energy and the availability of localisation services; *Routing*, as the routing information should be dynamically updated throughout the path to destination; and *power consumption* as the actual physical movements should be added to the consumed energy in network.

By acknowledging the increase in energy consumption in mobile wireless sensor networks (MWSNs), Selmic et al. [86] listed advantages of of mobility in such networks as **1)** ability of network to wholly/partly reconnect connectivity gaps if some parts of network are disconnected as the result of nodes' battery depletion; **2)** preventing unbalanced battery node depletion and failure as the results of uneven traffic flow toward sink nodes and base stations; **3)** providing multiple paths with presence of the mobile nodes in different parts of network in order to have an increase in channel capacity and integrity of transported data throughout the network. The authors consider MWSNs in three different forms of *planar*, *2-tier* and *3-tier* architectures. In planar, both mobile and stationary nodes communicate with one another. In 2-tier architecture mobile nodes form overlay network responsible for moving data. In 3-tier architecture, data handed from stationary to mobile nodes and then from mobile nodes to access points. Despite the benefits of MWSNs, continuous localisation of mobile nodes is challenging compared to static nodes such as latency and movement-related signal variations (e.g. Doppler effect). This is because, locations of mobile nodes are reliable if as soon as they are measured and should be estimated and error may be propagated due to node movement; otherwise, should be repetitively evaluated with additional computational and communicational resources. According to the authors, there is a trade-off between the localisation latency and its accuracy which distributed localisation algorithms are proposed to reduce latency and address such trade-off.

2.1.3 Coverage and Connectivity

Selmic et al. [86] defined WSNs coverage as the '*ratio of the space covered by the sensor nodes, to the total space of interest*'. Similar to art gallery problem [98] coverage, as important measure of Quality of Service (QoS), represents the quality of sensor nodes on performance of monitoring their designated area in WSNs. Thus coverage should properly managed [86, 99].

Monitoring region of interest via optimal deployment of sensor nodes can address the coverage problem. Selmic et al. [86] considered coverage as,

- *Coverage via Subset Selection.* This coverage assumes that there is redundancy in the deployment of nodes and areas require to be monitored by more than one sensor node. So based on level of desired coverage, minimal subset of sensors is selected and remaining sensor nodes are kept off in order to reduce the nodes power consumptions.
- *Coverage via Homology.* The absence of sensing coverage can be found by using homology via nodes connectivity.

Selmic et al. [86] considered the importance of connectivity in a sense that information can not be transmitted to the designated base stations or gateways. Coverage and connectivity are two of the criteria for the performance of algorithms, and important factors in wireless quality of service [44, 100–102], Wang et al. [38], Miao et al. [102]. Therefore, different algorithms and schemes have been devised to increase coverage and connectivity, reduce consumed energy by taking many of the challenges and natural limitations of WSNs into account [38, 65, 69, 97, 102–111]. Miao et al. [102] present a different coverage categories via a sensor deployment algorithm which improves the coverage of the area/object of interest (AOI/OOI). According to Kott [112], higher connectivity gives a better view of systems' complexity and influences on the systems' robustness and resiliency. Connectivity can be used for deducing the relative locations of nodes (localisation) as in Shang et al. [113]. Younis et al. [110] consider the reachability and connectivity of sensor nodes in the deployed area and role of nodes' connectivity in measurement and forwarding the traffic to sink nodes. Paper especially emphasises on sensor to base stations (BS) connectivity as nodes can be kept reachable to each other in many applications such as disaster management and recovery. With regard the effects of node failure due to external physical damage or hardware malfunction on network, the authors considered proactive and reactive fault-tolerant techniques as solutions to connect and restore the inter-segments connectivity.

Zhu et al. [69] brought survey on networks' static and dynamic coverage preservation, maintenance of connectivity as well as different deployment strategies in WSNs. The authors considered the relationship between coverage and connectivity such that sufficient condition for 1-coverage to imply connectivity is when $R_C \geq 2\dot{R}_S$. In Amitabha et al. [44] the concept of coverage and connectivity, the effects of mobility and node deployment on network coverage are introduced. It should be noted that connectivity and coverage ought to be considered at the same time. However, by existence of coverage, existence of the connectivity with high probability can not be inferred and vice versa.

Bai et al. [114] propose optimal deployment patterns of nodes as defined as *pattern mutation* in which nodes have small relations of communication to the sensing ranges (r_c/r_s) for given k-connectivity and the complete coverage. This approach seems to be beneficial in cases where ranges should be changed with respect to the environment, the relative power of the nodes and the sensor nodes' density.

Selmic et al. [86] defined the concept of *1-coverage* and *k-coverage* with regard the number of nodes covering every point of interest and the absence of 1-coverage as *hole* in the coverage of network. The authors consider hole as *1-hole*, as lack of at least one sensor node to cover the point of interest in the given area.

2.1.3.1 Coverage Categories

Among more early works, Gage [115] divided coverage (by robots) into three classes; Blanket Coverage (*attempt to maximise coverage of the area of the interest*), Barrier Coverage (*attempt to minimise the chance of intruder penetrating*

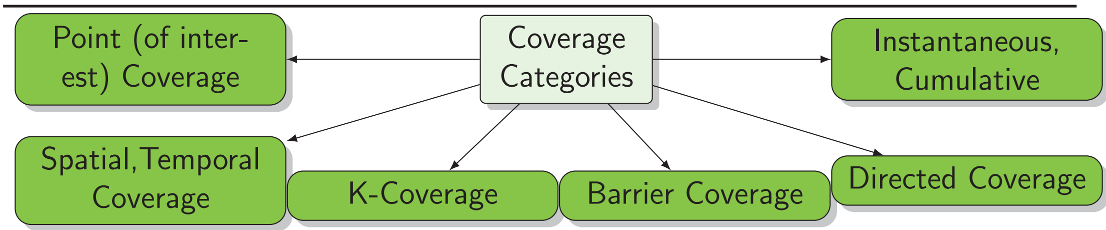


Figure 2.2: Coverage Categories

the barrier), and Sweep Coverage(*equivalent of a barrier moving across the area of interest*). Different types of coverages are briefly listed here, which may overlap with respect to their concepts and applications (Figure 2.2).

- ***Point (of Interest) Coverage*** (Yange et al. [116], Li et al. [117–119], Mia et al. [102] , Li et al. [66]). In such coverage, specific points, called *points of interest* (POI) rather than *regions of interest* (ROI) are covered by sensors. The objective of point coverage is different from conventional coverage in which the aim is to maximise the sensing coverage of a deployment area. Mia and Yun-Qian [102], defined point coverage as covering a set of critical locations or positions which can be monitored more accurately, instead of whole area. In [119], circular-focused coverage in which focused areas (areas of interest) have more priority for coverage, and such priorities decrease as we move further from the given areas of coverage.
- ***Barrier Coverage*** (Tian et al. [120], Tao et al. [121], Mia et al. [102], Liu et al. [122], Kumar et al. [123], Li et al. [66]). The barrier coverage model, formed by lines of sensors throughout the entire monitored area, provides the ability to detect intruders. To have guaranteed detection, proper overlap of outdoor deployed sensors is required. In randomly distributed nodes, dispatching mobile nodes (such as *robust nodes* with additional environmental protections and moving motors) to the locations of failed nodes to address

the vulnerability and increase the survivability of sensors (*regular nodes* lacking protection) in harsh environments exposed to snow and rain seems to be more economically justifiable and applicable than using the nodes with additional protections. In sustaining the barrier coverage of improperly deployed nodes, minimising the overlap sensing ranges and consumed energy, node relocation algorithms are applied.

- ***k-Coverage*** (Xu and Zhu [124], Zhu et al. [69]). K-coverage is the measure of the quality of observation of the physical space and the environment by sensors. In k-coverage, each location is within the range of at least k sensors.
- ***Directed Coverage*** (Bai et al. [125], Guvensan and Yavuz [65]). In directed coverage an intruder penetrating directionally into area enclosed between two boundaries of *source* and *destination* are detected. These boundaries are where the penetrators enter and exit into a network. According to the authors, for the areas of an arbitrary shape, directed coverage is considered to be a general case of barrier coverage. Directed coverage and the intention of an intruder would affect network deployment, network design, and repair and configuration runtime. Limitations of sensors, their costs, the environment and scale of the area of deployment are taken into account to increase the penetration detection in such areas.
- ***Spatial vs Temporal Coverage*** (Liu et al. [126], Miao et al. [102], Ammari et al. [127], Chang et al. [128,129], Leonard et al. [45]). *Spatial coverage* is the area of coverage represented as the fraction of AOI covered by sensors [102]. Points of interest have *temporal coverage* if is monitored and covered for a specific duration of time. Such temporal coverage applies where the number of mobile nodes either is not enough for full coverage, or the coverage of an area of interest for the entire time is not required. Balatin et al. [130] defined *frequency of coverage* as the frequency of each position of interest being covered for a duration of the time by sensor nodes.

- ***Instantaneous vs Cumulative Coverage*** (Miao et al. [102], Zhu et al. [69]). Unlike *instantaneous coverage*, according to the paper by Miao et al. [102], in *cumulative coverage*, the mobile sensors in WSNs where dynamic and distributed nodes have the ability to (autonomously) move and cover different uncovered areas in a timely manner and for the required period.
- ***Target Coverage*** (Miao et al. [102], Guvensan and Yavuz [65]). In *target coverage*, a group of sensors track and inspect targets with higher visibility and greater accuracy. Sensor nodes relocate such that the coverage of targets is enhanced. Miao et al. [102], consider the surveillance of objects as *object detection* (determining the presence or absence of objects or their related information in the AOI by sensing nodes), *object tracking* (determining their place and following the movement of objects), and *object recognition* (in order to conclude the characteristics of objects).

Li et al. [66]) classified coverage into *high, medium* and *low granularity coverage* with regard to the quality of sensed/covered areas. In high granularity coverage, each point of target should be sensed in order to ensure swift response and detection of events, which usually done via and considered as, *blanket coverage*, which increases deployment costs. Medium granularity tries to covers every path crossing, as in barrier coverage, to detect possible intruders. Medium granularity has a delay, and intruders must pass movement path to be reliably detected. In Low granularity coverage, sparse network tolerate delay in detection of event and areas are covered by usually mobile nodes in order to lower cost of deployment and maintenance.

2.2 Effect of Mobility

Random, improper initial node deployment and obstacles lead to inefficient resources utilisation which results in network holes. The aim of different schemes such as node deployment [18], relocation and movements [21,131,132] are to simply compensates all or parts of coverage holes [133].

Integrating mobility into sensor nodes and applying it in controlled manner have provided much flexibility and the ability to address the challenges and issues in WSNs. Yet, at the same time it has introduced new problems and open questions such as topological instability [41, 134, 135]. Playing as the double edge sword [65, 97], the mobility of nodes can affect networks' performance, such as networks' lifetime [51], (MANets) routing [136], the coverage and connectivity of network [137, 138], detection effectiveness [102, 135], etc.

2.2.1 Mobility Classification

Mobility models and patterns can be divided into many aspects within the context of WSNs. By considering a wide spectrum of mobility models and patterns in wireless networks geared for different applications and solutions in the literature [41, 45, 102, 134, 137, 139, 140], some conventional classifications of mobility are as follows (Figure 2.3):

- ***Coordinated, Emergent, and Random Mobility Models*** (Miao et al. [102], Leonard et al. [45]). Fully coordinated mobility models have high efficiency and good controllability, but they are non-scalable and lack robustness. Coordinators can be assigned before or be elected as is done in cluster head nodes, and this process of election can be performed centrally or be distributed. Coordinators also have their own form of hierarchy as in

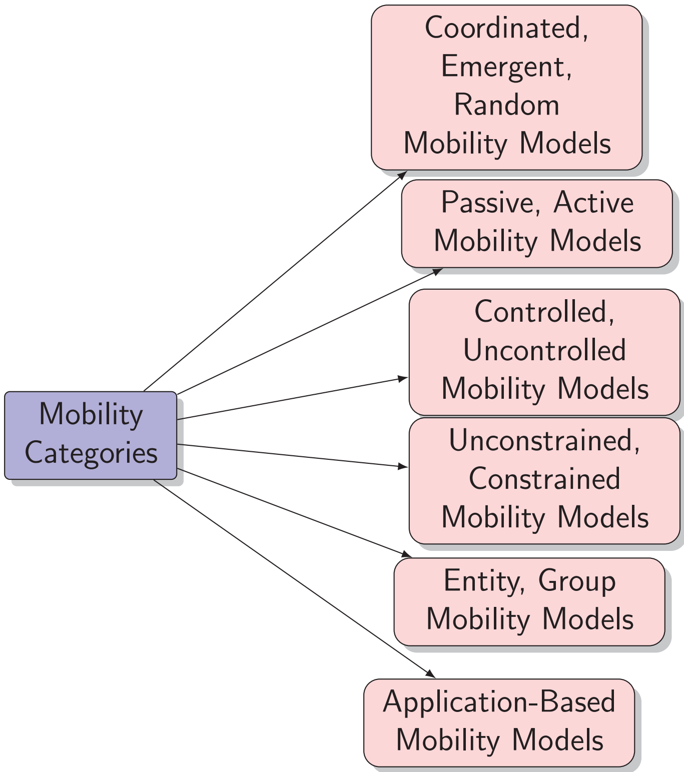


Figure 2.3: Mobility Categories

hierarchical cluster heads. Fully random mobility models are easy to implement and are environmentally adaptable, although they have low efficiency and unpredictable performance. Emergent mobility models have high scalability, flexibility and robustness while having difficult global controllability and unpredicted system status. If nodes move in as a group cooperatively such that nodes reaches/emerge to a common goal or fulfilled a shared task as the result of such movements. A cooperative movement can be but not necessarily group movement. Node can autonomously and independently move in different part of the network and yet realise some operational goals.

- ***Passive and Active Mobility Models*** (Santi [139]). In *passive mobility models*, node movements are triggered by external forces (such as wind, animal movement, ocean currents, etc.) and nodes are not able to have autonomous movements. The author divided further the passive mobility models into sensor nodes, which are embedded in mobile entities, for example people, animals, moving vehicles and so on that move the sensor nodes and deployed nodes which are not fixed and are free to move, such that their movements in the environment are affected by external forces, such as wind and ocean waves. In *active mobility models*, self-propelled motion capabilities contribute to the movements of nodes. Nodes (including sink nodes) can choose the direction, speed and timing of movements. Active mobility models are suited to situations in which coverage or collection of data for initial random deployment of nodes are either maintained or increased by such movements. Better data collection, network lifetime and efficiency of sensor nodes can be achieved by using mobile sink nodes.

- ***Controllable and Uncontrollable Mobility Models*** (Luo et al. [141]). In *uncontrollable mobility* (U-mobility), node movements and locations are affected by the environments such as flow or waves (for sensors on the sea), which are neither easily predictable nor controllable. Such U-mobility would cause (coverage) holes or disconnections between nodes. In *controllable mobility* (C-mobility), the movement and relocation of nodes are controlled to compensate for gradual drift, or to repair failures and holes caused by the nodes' U-mobility.
- ***Unconstrained and Constrained Mobility Models*** (Chellappan et al. [43]). In the *constrained mobility model*, the maximum movement distances of sensors are limited. Such limitations are considered to due nodes' physical characteristics and energy. Limited movements also keep the integrity of network and possible collision as nodes are not relocated large distances. However unlike in the *unconstrained mobility model*, nodes take longer to

reach their objectives, targets or follow their planned paths. The limited movements of nodes can be known either in the centralised or distributed style.

- ***Entity and Group Mobility Models*** (Book by Roy [134], Rezazadeh et al. [97], Hong et al. [142], Miao et al. [102], Olfati Saber [143], Leonard [45], Suzuki et al. [144]). In *entity movement model*, nodes individually move either randomly or deterministically to their targets, or wander around in their given fields. In *group movement model*, nodes move in groups as they are divided into different sets of nodes moving in their fields. Although, nodes move in their groups; they have their own deterministic or random movements. Nodes in the group mobility model configure their mobility behaviours, such as velocity, direction and amount of displacement (in the given iterations), within groups to reach common goals of orientation, velocity and so on. The idea of collective motion in [45] was used to increase the sensing precision in harsh environments such as the ocean. Some of the collective motions are based on flocking theory [143], [145].
- ***Application-Based Mobility Models*** (Aschenbruck et al. [146], Tang et al. [147]). Some mobility models are geared to address specific applications. In [147], group mobility of nodes is applied in military application to maximise the mobility and transmission path privacy by moving the nodes in groups and the cluster in a triangular path shape by accepting and minimising the increase in overheads and distances among nodes using *diamond group mobility* (DGM) for the battlefields and movement of soldiers around the commander.

2.3 Effect of Localisation

Localisation accuracy can affect the performance of large scale sensor network. In order to react and make decision upon the sensed data, the location of sensor nodes becoming more important such as in routing and topology control [86].
Many mobility models implicitly or explicitly assume the presence of GPS or other methods of localisation in their implementation and analyses [101]. In many scenarios the location of nodes should be deduced relatively to their neighbours (Wang et al. [100], [148]). The partition or merging of groups of nodes in each mobility model can have effect over the performance of routing, relocation and recovery algorithms when nodes are relocated to address other issues in networks. Wang et al. [100] provide brief but comprehensive classes of sensor localisation as; proximity-based, where the precision of estimation depends on the number of reference neighbourhoods of a node; *range-based* where localisation is performed with distance information in sparse networks; and *angle-based* which adds the cost of angular measurement to the distance. Franceschini et al. [149] presented three localisation approaches as *time of arrival* (TOA), *received signal strength* (RSS), and *angle of arrival* (AOA). These localisations can be combined together to reduce complexity and increase precision of localisation (Wang et al. [100]).

2.4 Network Resiliency

Resilience as 'rebound' and 'action of springing back', Laprie [150] presented the definition of resilience within different contexts such as ecology, business and industry safety with respect to notations of persistence, stability and diversity. The author considered common definition of resilience as '*the ability of successfully accommodate unforeseen environmental perturbations and disturbances*' and differentiated the fault-tolerant from resilient systems in the sense that the former neglects the unpredicted phenomena system may undergo. With the booming growth of ubiquitous systems in terms of small mobile computers, huge servers and super computers, as author's definition of resilience, he called attention to importance of keeping the ability of such systems in the sense of delivering justifiably-trusted services in the face of continuous changes. Different definition of resilience is as '*The persistence of the avoidance of failures that are unacceptably frequent or severe, when facing changes*'. The author considered the changes to systems with respect to their *nature*, *prospect* and *timing*. Changes with regard to their nature can be functional, environmental or technical (both in hardware and software). Prospect of changes can be foreseen, foreseeable or drastic. Timing of changes can be short, medium, or long terms.

Encompassing a large spectrum of definitions, the concept of *network resiliency* is different with respect to its applications, objectives, addressed challenges, types of networks, and in distributed and complex systems. Due to its importance in network analysis and design, the concept of network resiliency, especially in wireless networks, is briefly introduced.

In one early attempt to provide clear definition, the concept of network resiliency and its probabilistic measure was presented in Najjar and Gaudiot [60]. With respect to the number of nodes and their degrees, the authors considered the problem of network *fault tolerance* with respect to probability of disconnection for regular graph network topologies. In their work, they suggested the probabilistic measure of network fault tolerance as the indicator of network resilience, $NR(p)$ and defined as *the number of failures a network can sustain while remaining connected with a probability* $(1 - p)$ Resiliency of network was presented as the tolerance level of a network facing failures. Analysis was done only for specific regular graphs, the authors, defined degradable level, or graceful degradation as maximum number node failures which a network can tolerate with a given probability.

A survey by Gupta et al. [151] investigated the concept of reliability of a system and suggested that wireless sensor network as *virtual entities* composed of distributed physical sub-entities of sensor nodes which such sub-systems would collaboratively operate with one another. The authors categorised the reliability of WSNs as *reliability of coverage (sensing)*, *reliable packet delivery*, *reliable secure data exchange*, *availability of network*, and *network latency*.

- ***Reliability of coverage (sensing)***. Gupta et al. [151] took into account the effect of (regular basis) failure of nodes within networks in their ability to acquire and collect information and to address such problem with the solution of redundancy in deployments, which results in overlap/multiple coverage of areas of interest, as this solution in contract to minimal coverage, provides the required coverage.
- ***Reliable of packet delivery***. The nature of unreliable and shared mediums of communication is a significant factor in the ability and quality of node communications, such as flow of aggregated information toward the sink node, which is worsened by failure of nodes.
- ***Reliable secure data exchange***. Due to the wireless feature of the medium, the interception of the exchange information may increase leading to compromised networks by either injecting false information or taking control of the network.
- ***Availability of network***. Considered as operational availability which is implication of system reliability and probability of the system to perform its task in a given time. System availability can be considered as the provided services, such as sensing/detecting events and propagating information to the sink node. Therefore, failures reduce the availability and in sequel the lifetime of network.

- ***Network latency***. Receiving information in an integrated and timely manner (with possibly required constraints) may significantly affect decision making in WSNs.

As well as 'end-to-end' or 'hop-by-hop' communication, authors in [151] also classified techniques with regard to re-transmission of packets and redundancy for reliability in packet delivery and events detection of the WSNs.
Network resiliency can be defined with respect to network recovery, survivability and reliability in wired networks (Grover [152], Vasseur et al. [153]).

Considering the broad applications of WSNs in the infrastructure-less environments, Al-Kofahi et al. [154], have taken into account limited resources, interferences, device impairments and harsh environments for nodes and link damages and defined survivability as *the capability of a network to deliver data successfully in a timely manner even in the presence of failures.*

Presenting the concept of resiliency in the context of complex networks, Deffuant et al. [155] tries to model patterns of resilience and viability in complex systems viability in different fields of ecology, sociology and psychology.
Huang et al. [156] distinguished between resilience, robustness, and survivability. Resilience is considered as the ability to recover to original status with the goal of making network fault-tolerant. Robustness is defined as the ability of network to resist attacks, but not necessarily with the implication of restoring from failures such that the robustness of a system makes network difficult to degrade under attack. Survivability makes network very hard to collapse or impair completely.
Work by Kott and Abdelzaher [112] defined the concept of complex systems and their robustness and resiliency. The authors listed different types of failures and presented approaches to render networks robust/resilient to such failures. In order to devise suitable recovery strategies, the authors suggested that proper under-

standing, prediction and evaluation of complex systems, as well as consideration of complex networks' vulnerabilities are important for systems which are governed by connected and distributed paradigms. Such networks have to interact in environments with limited resources and high degrees of uncertainty and failure. The authors distinguished between robustness and resiliency with the definitions below.

- ***Robustness*** *is the degree of ability of systems to tolerate an unexpected internal and external event or change without degradation of their performance.* (Kott et al. [112] deduced from IEEE standard 610.12.1990)
- ***Resiliency*** *is a system's ability to recover or regenerate its performance after an unexpected impact causing of the systems' performance.* (Kott et al. [112])

Huang et al. [156] defined the resiliency of networks as *the ability of network to provide and maintain an acceptable level of service when facing various faults and challenges to its normal operation.* Huang et al. [156] and the references therein have defined resiliency as *the probability of at least having another path within the time interval T, given at least one node on the primary path has failed.*

Westmark [157], and Whitson et al. [158] tried to define the concept of resiliency as *the sensitivity of network service and network service restoration at the time of presence of abnormal/external influences.* Dressler [159] introduced the concept of *self-healing* which is related to network resilience, as *the mechanism allows to detect, localize, and repair failure automatically, primarily distinguished by the cause of failure, i.e. break down, overload, malfunction.* Cholda et al. [57] have given the definition of *quality of resilience* with respect to *frequency and length of service interruption.* In [57] recovery methods and mechanisms were classified based on scenario and scale of recovery, which were to some extent mentioned in [152, 153].

Miranda et al. [160] and Huang et al. [156] defined the resiliency as

The ability to provide and maintain acceptable level of service in the face of faults and challenges to normal operation. Hence, the network must provide and maintain essential service under adverse conditions as well as allow for rapid and full service recovery.

Kott and Abdelzaher [112] placed the dependency of recovery on the extent of the damage and before the point where recovery was impossible. The authors attribute some catastrophic system failures emerging from the nature of high complexity of links or unpredictable interactions which cannot be taken into account in the design of systems' safety precautions. [112] argue that approaches which increase the resiliency and robustness have their own trade-offs, such as dependency. Thus, a proper limit of trade-offs should be reached where some of these trade-offs may co-exist with one another. Some of such the trade-offs are as follows.

- ***Resource vs Resiliency***. Additional resources enhance resiliency.
- ***Performance vs Resiliency***. Network gears for higher performances suffer more from the impacts and are more difficult and time-consuming to be restored to given operation points.
- ***Resiliency to Multiple Types of Disruption***. Reaching proper network resiliency against multiple types of disruption may be more challenging than for one type of impact, such as random failure versus targeted attack.
- ***Resiliency vs Complexity***. More sophisticated mechanisms in a network improves the resiliency with higher cost, but is harder to grasp for the restoration of network's capability and would result in degraded performance in the case of failures of specially the unpredicted types.

Deffuant et al. [155] considers the resiliency of enabling systems to recovery from (strong) perturbations. From different views and definitions, present different stability properties as *constancy* (staying essentially unchanged), *resilience* (returning to the reference state (or dynamics) after a temporary disturbance), *persistence* (through time of an ecological system), *resistance* (staying essentially unchanged despite the presence of disturbances), *elasticity* (speed of return to the reference state (or dynamics)), and *domain of attraction* (the whole of states from which the reference state (or dynamics) can be reached after a temporary disturbance).

Some of definitions of resilience and network resiliency in the literature, that can be considered in WSNs, are presented in Table 2.1.

Some examples of resiliencies inspired by the natures and graphs are provided in the following section.

2.4.1 Resiliency in Graphs and Nature

Similar to other networks and systems, such as biological systems, resiliency is one of the important features in wireless sensor networks (Huang et al. [156]). The authors consider the time it takes for a system to return to equilibrium after perturbation as the measure of resilience in ecology.
Many graphs, due to their structural features, show good resilient behaviour to possible deletion and failures of their nodes and edges, such as random regular graphs [58]. Regular random graphs have given d-degree and d-connectivity with high probability. Although random regular graphs seem to be a good candidate for initial deployment of nodes as a resilient cluster, they don't seem to fit random network models coping with dynamic and random mobility patterns of ad hoc and sensor node networks. Pappas et al. [162, 163] introduced a fairly resilient network model which is inspired by the *mammalian (blood) circulation system*. Though interesting, the given model based on blood circulation systems

Author(s)	**Yrs.**		
Nijjar et al. [60]	1990	**Network Resiliency & Fault Tolerance**	*"The number of failures a network can sustain while remaining connected with a probability $(1-p)$."* **Resiliency of network** was presented as the tolerance level of a network facing failures.
Grover [152], Vasseur et al. [153]	2003	**Network Resiliency**	It can be defined with respect to network recovery, survivability and reliability in wired networks.
Westmark [157] and Whitson et al. [158]	2004 2009	Resiliency	*The sensitivity of network service and network service restoration at the time of presence of abnormal/external influences*
Dressler [159, 161]	2006 2008	**Self-healing**	*The mechanism allows to detect, localise, and repair failure automatically, primarily distinguished by the cause of failure, i.e. break down, overload, malfunction.*
Laprie [150]	2008	**Resilience, Network Resiliency**	'Rebound' and 'action of springing back.' *"The ability of successfully accommodate unforeseen environmental perturbations and disturbances.".* *"The persistence of the avoidance of failures that are unacceptably frequent or severe, when facing changes."*
Cholda et al. [57]	2009	**Resilience**	The definition is with respect to *"A general ability to improve network fault tolerance and, as a result, its reliability."*
Al-Kofahi et al. [154]	2010	**Survivability**	*The capability of a network to deliver data successfully in a timely manner even in the presence of failures.*
Deffuant et al. [155]	2011	**Resiliency**	The resiliency of enabling systems to recovery from (strong) perturbations.
Kott and Abdelzaher [112]	2014	**Robustness Resiliency**	**Robustness** *is the degree of ability of systems to tolerate an unexpected internal and external event or change without degradation of their performance.* **Resiliency** *is a system's ability to recover or regenerate its performance after an unexpected impact causing of the systems' performance.*
Huang et al. [156]	2015	**Resilience Robustness Survivability**	**Resiliency** is considered as the ability to recover to original status with the goal of making network fault-tolerant. *the ability of network to provide and maintain an acceptable level of service when facing various faults and challenges to its normal operation.* **Robustness** is defined as the ability of network to resist attacks, but not necessarily with the implication of restoring from failures such that the robustness of a system makes network difficult to degrade under attack. **Survivability** makes network very hard to collapse or impair completely.
Miranda et al. [160] and Huang et al. [156]	2015 2016	**Resiliency**	*The ability to provide and maintain acceptable level of service in the face of faults and challenges to normal operation. The network must provide and maintain essential service under adverse conditions as well as allow for rapid and full service recovery.*

Table 2.1. Some definitions of Resilience and Network Resiliency in the literature.

in mammals is not always applicable. Each different component (artery, capillary or vein) based on location of failure and damage shows a different level of resiliency and vulnerability. Another example of a resilient pattern in nature, introduced by Jones [164–166], is a biological damage repair mechanism based on chemo-attraction (chemo-repulsion) of (bio) particle-like agent population which collaboratively heals random failures or external damage occurred to them. The extensive work by authors on the subject can be found in [166]. Either analytically [59,62,167,168], or probabilistically [169,170], most of the analysis of network resiliency are geared for cases of random failure of nodes and the cascade failure model, and they would not address holes.

2.4.2 Resiliency in WSNs

Resiliency schemes have been practised in different areas of WSNs such as in (multi-path) routing (Dulman et al. [171]), resilient clustering (Han et al. [172]), encryption and authentications in hierarchical WSNs against faults and malicious attacks (Zhan et al. [173]), proper mobility model for compromised unattended mobile nodes to restore after the attack (Pietro et al. [174,175]), secure communications among nodes against attacks (Miyaji and Omote [176]), resiliency of WSNs within the context of security of geographic forwarding (Abu-Ghazaleh et al. [177]), and redistribution of data and information among the nodes while relaying toward the sink nodes (Valero et al. [178]).

- Multipath routing can be one method of recovering and resisting failures [156]. Considering network failures in the form of either link or node failures, Lee et al. [179] proposed a proactive recovery scheme by constructing disjoint paths resulting in fast recovery when sufficient resources and infrastructure are available. The author also defined *disjointness* and *stretch* of backup path with respect to primary path. Stretch of backup is a fraction of backup path length to primary path length. Disjointness is the measure of how disjoint the primary path is from the backup path by excluding the common links/nodes from both paths. Regarding the nature of node redundancy in WSNs, it is reasonable to consider multiple path routing using partially/totally disjoint backup paths to increase the resiliency of networks [156].

Similar to the classification by Huang et al. [156], Dulman et al. [171] proposed the idea of *multi-path routing*, by sending packets through multiple (disjoint) paths from the given source to destination nodes. They introduced two multi-path routing protocol designs, as *disjoint multi-path* and *braided multipath* routing. Disjoint multi-path consists of primary and alternate paths where the alternate paths are less efficient in terms of delay and energy consumption. Alternate paths are disjoint in the sense that they are not affected by the primary path's failures. The braided multi-path scheme relaxes on the constraints of nodes being disjoint in alternate paths. A braided multi-path can be transformed into disjoint primary and an alternative path scheme (probably consisting of many partial alternative paths). The proposed schemes are applicable to delivering data in environments with high uncertainty and unreliability, and the delays in transmission are decreased with respect to the degree of multi-path. Benefitting from the redundancy of sensor nodes and extra resources such as bandwidth and energy, k multi-paths from source to destination are discovered based on the reputation coefficient of nodes from their previous status of routing traffic. Sent traffic packets are divided into several parts as *sub-packets* and glued together on the assumption that the mobility of nodes and the topological changes are much slower than the rate of discovery and data package transmission, so that the packets can be correctly sent to the proper paths with higher reputations via on-demand routing and without static routing table.

- Han et al. [172] propose a secure dynamic clustering scheme to improve network lifetime and resiliency against node attacks especially cluster-head captures, by providing secure clustering and resilient cluster-head election, as well as providing by data privacy through concealed data aggregation. The proposed scheme, limited memory and information exchange, provide security to prevent wormhole attacks *inside* (i.e. malicious insider) and *outside* attacks.
- Work by Pietro et al. [174, 175] presents a cooperative, low overhead and distributed protocol for intrusion resiliency in unattended mobile sensors by leveraging sensor mobility, enabling the compromised sensors to recover a secure state after compromise in WSNs.
- Listing security-based attacks, such as localisation broadcast manipulation, multiple unicast packet attacks, mobility attacks and subversion of localisation nodes as well as their possible solutions for WSNs, Abu-Ghazaleh et al. [177], considered the resiliency of WSNs within the context of the security of geographic forwarding (where nodes send their packets to the most suitable neighbour closest to the destination) against the attacks of misbehaving nodes of having misleading locations, by introducing address verification algorithms, route authentication, and trust-based route selection as the solution to such attacks.
- Valero et al. [178] propose a scheme of redistribution among nodes with energy-constraints while the information is collected and relayed toward sink node through one or more hops, as information should be stored for possible retrieval.
- Karlof and Wagner [180], by listing security attacks and their countermeasures and solutions, considered secure routing in WSNs.
- Selmic et al. [86] consider the load balancing as a solution to fast depletion

of nodes' energies of the certain parts in network (i.e. nodes proximate to sink node) due to unbalance flow of data in those areas. The authors presented some load balancing schemes through clustering in which less-energy-constrained gateway nodes are considered as the *cluster head* (CE). In Selmic et al. [86], multi-sink multi-path architecture is considered as solution in the form of spatially diverse sinks where data an be sent through multiple disjoint paths toward sink. The authors noted that in this scheme the cost of physical deployment of additional sinks should be taken into account. According to Selmic et al. [86] the routing has effect on the delay, reliability, resource utilisation of WSNs. Fault tolerant routing protocols of multiple non-overlapping paths make network resistant to route failure.

2.5 Topology Control Schemes

Devised *topology control* (TC) schemes [31, 181–183] affect network life time [110, 184, 185] (Network lifetime can be defined as *'the time between the network deployment and the failures as if the effective coverage is less than predefined/threshold value'* [185]) and are expected to have the impact on the resiliency of network (Kott and Abdelzaher [112]). Based on the classification by Labrador et al. [182], two phases in topology control, *topology construction* and *topology maintenance*, are important for power consumption and nodes' lifetime, and would have a definitive effect on the resiliency and health of the network. Kott and Abdelzaher [112] consider the topology modifications as one of the first conditions of the resiliency of networks in order to survive (local) damages. Resiliency has bounds and limited ranges in which the given recovery mechanisms cannot reach their objectives. For example, biological beings are able to recover from localised injuries which are not beyond a threshold. Such limitation may be known and/or estimated from the properties of systems (i.e. possible vulnerabilities and their topologies) and nature of the possible failures.

Different centralised and distributed TC schemes addressing wireless networks' behaviours in harsh and volatile environment have been devised in the literature (For these see surveys by Santi [31] Labrador [182], Sahoo et al. [30], Li and Yang [181], Jian et al. [29], and You et al. [34]) Emerging schemes are also presented for various upcoming scenarios and application as WSNs mature in their different aspects (Younis et al. [110], Aziz et al. [27]). The survey by Li et al. [66] considered topology control techniques mainly for maintaining coverage and connectivity of WSNs. Younis et al. [110], categorises current techniques as *reactive* and *proactive* and classified TC schemes as following (See also Figure 2.4)

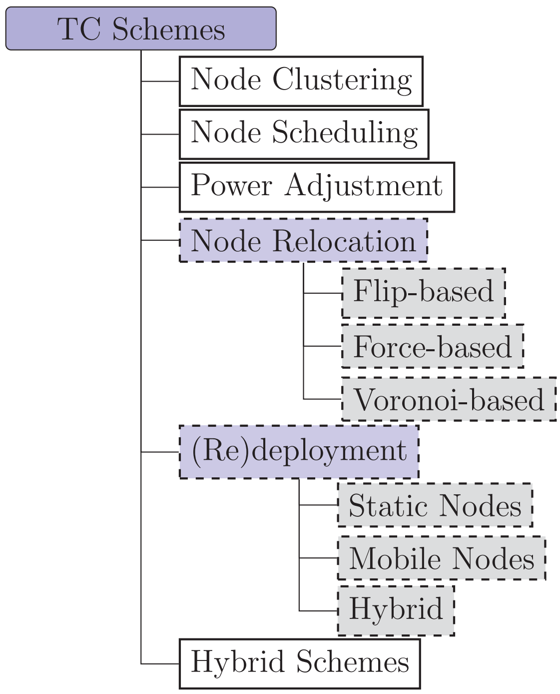


Figure 2.4: Topology Control Scheme Categories

- ***Node discovery***. Detecting nodes, their locations, and sharing this infor-

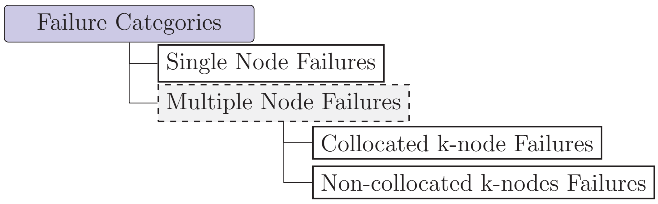


Figure 2.5: Node Failure Categories in WSNs

mation to other parts of networks.

- ***Sleep cycle management***. Scheduled turning on/off of (redundant) nodes to conserve energy and extend network lifetime.
- ***Clustering***. Grouping of nodes with cluster-head nodes to reach higher scalability and energy efficiency as their inter-cluster communications are handled by the primary and/or backup cluster-heads, in the case of primary failure, to reduce traffic and energy consumptions.
- ***Power control***. Due to the relation of longer ranges with higher power consumptions, link-layer techniques can be used in order to tune transmission power and reduce excessive energy and interference as the result of limited and controlled connectivity.
- ***Movement control***. By benefiting from node mobility, it is possible to achieve different performance objectives such as lifetime, delivery latency, higher relay, coverage, etc.

Since different failures of *single node failure*, *non-collocated k-node failures*, and *collocated k-node failures* may create holes in the coverage area and disconnect networks into multiple partitions, recovery from failures should be chosen based on their nature (Figure 2.5). The authors in [110] considered fault-tolerant techniques, such as *reactive/provisioned node placement*, *placement of stationary and mobile network entities* (e.g. relay nodes), and *coordinate multi-node relocations*. Younis et al. [110] considered reaching/maintaining targeted coverage and connectivity as well as conserving the energy of nodes as the primary objective of topology management techniques.

In this thesis, similar to the classifications in [110], topology control techniques are classified as below.

2.5.1 Deployment and Redeployment

Initial deployment and/or deployment of new nodes to compensate for node failures in a network would significantly affect quality of service. (e.g. coverage and connectivity in Hou et al. [186], Gosh and Amitabha et al. [44], Robinson and Knightly [187], Chen et al. [18], and Khoufi et al. [188]), fault-tolerance and operational lifetime of networks, (see Mnasasri et al. [189], Younis and Akkaya [190], Tsai et al. [191] and Hou et al. [186]). Each of these deployments' performance and costs depend on different parameters (Mnasri et al. [189]), such as sensor models (heterogeneous and homogeneous) (Wu et al. [192]), and irregular and more realistic shapes of deployment areas and in the presence of obstacles (Khoufi et al. [188]). According to [44] and [18], different types of node deployments may be suitable and more efficient, depending on the scenario and environment. Lee et al. [193] considered the placement of relay nodes between isolated network partitions as the result of multiple collocated failure, such that these segments can be connected together via 2-vertex-disjoint paths. Different classifications of node deployment are briefly given here:

- ***Mobility of Deployments*** (Chang et al. [194], Younis and Akkaya, [190], Htun et al. [195], Han et al. [89], Leonard et al. [45], Li et al. [66]). Three types of sensor deployments under water are classified in [89] as *static deployment* (either random or regular), *self-adjusted deployment*, and *movement-assisted deployment*. In static deployments, sensors' positions are static after their initial deployment as they are fixed to surface buoys or anchors at the bottom of the ocean. In self-adjusted deployment, nodes adjust their positions to the targeted locations/ requirements. In movement-assisted deployment, under-water mobile sensor nodes move cooperatively with other sensors to boost patrolling and monitoring tasks in the area of interest. Some of the self-adjusted and movement-assisted deployment algorithms may taken into the account waves and currents, and other external factors [45].

A survey by Miranda et al. [160] presented *rapidly deployable networks* (RDNs) that maintain connectivity in unstable environments in the case of failure or disaster. Miranda et al. [160] and the references therein considered *breadcrumb-based* and *mobile robotic backbone* approaches for deployment strategies. In the breadcrumb-based deployment approach, small and inexpensive devices as relays are dropped at regular intervals to maintain connectivity and forward packets. In the mobile robotic backbone approach, autonomous nodes are capable of mobility and self-deployment to adjust their location based on the conditions. The authors considered a deployment scheme as proposed by Pezeshkian et al. [196] in which mobile nodes can be converted to static nodes when some criteria are meet.

To achieve different network design objectives, Chang et al. [194] and Younis and Akkaya [190] classified node deployments as *static* and *dynamic*, in the latter of which, nodes relocated based on the status of networks in the course of being functional.

Htun et al. [195] introduce a hybrid model using static and mobile sensor nodes to repair coverage holes. Mobile nodes play an important role and the authors limited the number of mobile nodes to eight in the healing process of coverage holes. The paper's method mainly divided into initial hybrid sensor node deployments and the covering of hole by using mobile nodes of suitable movement distances.

Li et al. [66], defined three types of networks as *static*, *mobile*, and *hybrid*. The static networks have energy hole problem and reduced lifetime but are easier to deploy as their nodes don't move after being deployed. In mobile networks, sensor nodes can easily relocate to increase network lifetime, but at higher cost and requiring more complex protocol design. Hybrid networks try to harness the benefits of both static and mobile nodes.

Niati et al. [197] considered deployment as *one-time* and *continuous* node deployments during the life of a network wherein continuous deployment nodes are replaced when they fail or their energy levels make them inoperable in a network. Using one-time node deployment, the authors benefitted from spare sensor nodes in addition to deployed nodes, which are not activated and are reserved for future use in their network. Such spare nodes can be used as replacements for failed nodes or play a role as mirror nodes for gateways in cluster-based or high load nodes in flat networks.

- ***Deployment Phases*** (Mnasri et al. [189]). The authors classified deployment into three main phases of *pre-deployment*, *deployment phase* (either manual node placement or dropping from a plane or drone) and *post-deployment* to address node failures, displacement, and fluctuation in radio propagation.
- ***Deployment Control*** (Mnasri et al. [189], Shen et al. [198]). [198] and [189] classify node deployments as *centralised*, *distributed* and *hybrid approaches* (using two or more deployment schemes simultaneously to solve the problem). Based on the way embedded mobility in sensor nodes are controlled and used to reach or maintain targeted performance and planned objectives in the network for various scenarios and applications, [194] classified dynamic deployment of nodes into *centralized* (locations of sensor nodes are decided by

a small number of nodes such as base station or cluster heads), and *distributed* deployment methods (decisions made locally and sometimes autonomously with incomplete information). Hence the decisions on the positions of nodes can be made in a centrally or distributed manner.

- ***Distribution of Deployment*** (Chang et al. [194], Amitabha et al. [44], Chen [18], Robinson and Knightly [187], Senouci et al. [199–201]). The authors in [194], categorised the static deployment further into *deterministic* and *random* with respect to the location of deployed nodes in the network. The authors of [18, 44], considered different types of node deployments such as random, incremental and movement-assisted deployment. In [194] deterministic deployment of nodes (such as grid-based deployment) is based on application and parameters in order to increase performance. In hostile environments, in the absence of a centralised supervisor, and in situations with unpredictable energy consumptions, such static node deployments to fixed locations may not address principal networks' design objectives (e.g. connectivity, coverage, etc.). By accepting a degree of redundancy, random deployment of nodes with given probability distributions or models (e.g. perturbed grid, uniform, Poisson, Gaussian, etc.) can be considered as a solution to cope with challenges more realistically and flexibly for the issues of deterministic deployment of static sensing devices. According to [187], different node deployment schemes such as grid, perturbed grids (nodes displaced with random distances/angles), hexagonal, triangular, square or random (with uniform or poisson distribution) affect network performance such as area coverage. Senouci et al. [199–201] presented different categories and distributions for random node deployments, based on probability density function (PDF) into *simple and compound strategies*, where simple strategies are based on variants of simple diffusion strategy and compound strategies are formed from the repetition of simple diffusion (either continuous or discrete). The authors then compared the performance of the aforesaid different types of node deployments in terms of coverage, connectivity, network fault tolerance and lifespan.

- ***Targeted Deployment*** (Wang et al. [202], Liu et al. [203]). Deployment of new nodes can be more intelligent and targeted than simple uniform deployment. The proposed node deployment model by Wang et al. [202], outperforms random uniform deployment in terms of network lifetime. [202] proposed a model for node deployment in which nodes are added to the areas around sink nodes. This is because nodes close to the sink node should not only send their originating traffic but also relay others' traffic to sinks. Therefore, if nodes are not properly deployed around sinks, their power and energy are exhausted faster, which affects the lifetime of the network. The paper defined network lifetime as *the cumulative active time of the network until the first sensor is out of the power.* Therefore, unlike traditional movement-assisted deployments in order to extend network lifetime, Liu et al. [203] proposed non-uniform node deployment throughout the network aimed at moving nodes closer to areas such as sink nodes, which consume more power.

2.5.2 Node Clustering

Node clustering has been used in WSNs to control network dynamics, hierarchical routing, and reduction of consumed energy in nodes by decreasing the amount of exchanged node traffic. A cluster is defined as *a subset of vertices whose induced graph is connected* (Banergee and Khuller [204]). The clustering model is harnessed for forming hierarchies to apply certain constraints, to control and to compensate for degraded coverage over time (Jiang et al. [205]), as well as for hierarchical routing in WSNs (Banergee and Khuller [204]). To bypass the burden of centralised control, Charalambous et al. [206] presented the *clustering approach* as one of hierarchical and scalable solutions in which sets of nodes are grouped/selected based on different criteria, such as density, distance, energy or network identifier. In this way, inter-cluster information exchange is limited to these cluster heads, and intra-cluster information exchange is also confined within each cluster instead of flooding and broadcasting throughout the network.

2.5.3 Routing and Node Discovery

Routing protocols in ad hoc [207], sensor networks [208], and VANETs [209], although they may not be directly considered as one of topology control schemes in many works, may be classified under the umbrella of TC schemes. This is because, if they are correctly geared for WSNs, they are able preserve nodes' energy and preserve overall network lifetime. Regarding nodes' energy constraints, as an example, Vidhyapriya and Vanathi [210] presented a reactive routing, called *energy aware routing* for more reliable and lower energy operation where suitable routes to destinations are determined based on energy and link status quo of nodes in the network. A survey of WSNs routing techniques and protocols by Al-Karaki et al. [211] presents routing protocol challenges and classifies them mainly as *flat*, *hierarchical*, and *location-based routing* protocols. The lack of node power, physical damage, and environmental interferences is the cause of node failures and blockages in network, according to [211], who consider MAC and routing protocols as the one of solutions for having a fault-tolerant behaviour and forming new links and paths to sink nodes. This consists of transmission power adjustment and/or re-routing traffic along alternative paths of higher energy nodes by accepting proper level of redundancy in the network.

Node discovery also can be considered as a TC scheme in WSNs as the effect of the proper detection of nodes and the perception of each node from its neighbouring nodes (after initial deployment, node relocation, and newly added nodes in the course of network lifetime) cannot be neglected. In large networks, some trade-offs based on application and topology of network should also be taken into account [110]. For example, a trade-off between performance of node discovery and the amount of shared topological information among nodes can be considered in order to preserve resources in terms of energy and bandwidth.

2.5.4 Redundancy and Node Power Adjustment

Redundancy is defined in Burgess et al. [212, 213] as *the duplication of resources to which there are no common dependencies between duplicates, such that they are not affected by the failure of one another.* Such redundancy increases the chance of availability of agents (as possible backups) and promise of the agents for their roles. Having additional economical burden and complexity, *redundancy* (high availability) is used to guarantee reliable communication and address the nature of random spreading of error-prone sensor nodes in harsh environments (Mnasri et al. [189], Silva, [214]). Regarding nodes' distances from one-another as criteria for redundancy, one of two nodes is considered to be the redundant node if the nodes are very close to one-another and within a given threshold (Sekhar et al. [215], Ma et al. [216]). According to Antil and Malik [91], depending on the density of nodes in a network, for higher densities, static nodes are preferable, while in more sparse densities, mobile and hybrid deployments in network can cope better with topological changes. Bettstetter [217] mentioned the effect of excessive numbers of nodes causing a interference, making critical nodes with higher number of degrees more vulnerable to damage or failure.

With respect to redundant resources, similar to [152, 153], among survivability mechanisms by Al-Kofahi et al. [154], *protection*, *restoration* and *hybrid* mechanisms are presented below,

- **Protection Mechanism**. Depending on the resource allocations, the protection mechanisms can be further classified into *reactive* and *proactive* protection mechanisms. In reactive protection mechanisms, due to redundancy, some of resources are *shared* among groups of nodes (or paths) in the case of faults and failures. For example, multiple paths as the backup (alternative) paths are shared and not used until a failure is detected on the primary paths. In proactive protection mechanisms, some of resources are *dedicated* to a specific set of nodes (or a path) forming a backup (alternative) path. For example a backup path is dedicated for a given (primary) path in network.
- **Restoration Mechanism**. In these mechanisms, backup resources are neither shared nor dedicated in advance for recovery. Thus, restoration mechanisms use less resources and are slower at time of recovery. This is because after detection of faults, unlike protection mechanisms in general, resources should firstly be discovered and then, if available, be set aside for recovery processes, as the (fault-detecting) nodes have less or no a priori knowledge or assumptions about the available resources.
- **Hybrid Mechanisms**. Combination of protection and restoration mechanisms are used to benefit from the advantage and flexibility of both.

Power adjustment may be another topology scheme that preserves the nodes' energy and achieves the required connectivity (Costa et al. [28], Labrador et al. [182], Santi et al. [31, 218] Sahoo et al. [30]). Power in nodes is consumed in different modes such as *moving*, *transmission* and *sleep* modes (Liu et al. [203]). Younis et al. [110] classified the different *power modes* of sensor node as *idle*, *sleep*, *transmit*, and *receive*. In order to address connectivity constraints, Costa et al. [28]

proposed a cooperative model of topology control in which transmission power levels are tuned to which of the nodes' neighbours reach a certain given degree of connectivity relative to one another. Santi et al. [218] proposed a probabilistic analysis for the range assignments problem in ad hoc networks. Sahoo et al. [30] proposed a distributed algorithm for multi-hop wireless sensor networks which controls the transmission power of nodes. Zhu et al. [69] mentioned the sleep scheduling mechanism and introduced the 'adjustable coverage radius' and the 'radius adaptive mechanism'.

Mahfoudh et al. [219] and Li et al. [66], suggested that power management and control of sensors reduces energy consumption, decreases interference and enhances spacial reuse while ameliorating contention in the MAC-layer medium. Li et al. [66] considered the power control to provide suitable connections among nodes to decrease energy consumption and improve the capacity of WSNs. They argue that most of power control problem has been applied to stationary networks rather than mobile nodes, which can be considered open research. Whether nodes are allowed to have the same or different transmission ranges, the authors listed power control mechanisms respectively as homogeneous or heterogeneous.

2.5.5 Node Scheduling Sleep Cycle Management

Selmic et al. [86] presented *topology Management* as techniques that seek to control the sleep schedules of nodes in a WSN. Scheduling are based on the redundant nature of densely deployed nodes with the high correlation of sensed data especially as nodes are within their close ranges. Scheduling reduces the consumed energy and increases the lifetime of network. However, node scheduling increases the latency of message exchanges and transfers (by the possibility of putting some of shortest-paths to sleep which results in an increase in lengths of average shortest paths between given sources and destinations) and heavily relies on proper synchronisation of which requires a minimum usage of memory and computing power. Non-trivial problem synchronisation plays an important role in scheduling and decision making of sensor in their processes [86].

Node scheduling and sleep/awaking cycle management, in tandem with other topology control schemes, can be applied to WSNs to extend network lifetime (Jiang et al. [205], Cerpa and Estrin [220], Younis et al. [110]). Node scheduling schemes, either centrally or distributedly, aim to manage and configure the redundancy of highly dense networks under energy constraints such that (full area) coverage and (strong) connectivity can be sustained among nodes in network. As a result of sleep cycle management, some redundant nodes can be turned off leading to less transmitted messages and signal interferences, requiring less failed transmission attempts.

The authors in [66] referred to literature to suggest that the largest amount of hardware energy consumption was due to radio communications in regard to the rest of hardware's energy consumption. The survey also liked to emphasise the importance of nodes scheduling sleep cycle management by the significant energy consumption difference between active and sleep state nodes. The authors referred by node scheduling as a *power management mechanism* to increase network lifetime and maintain the network performance at an acceptable level, within the context of network topology.

2.5.6 Node Relocation

By integrating mobility into sensor nodes, different node movement strategies with respect to the nodes' limitations, characteristics and environmental conditions have been devised to address dynamic topological behaviours and challenges such as the coverage problem in WSNs (Wang et al. [38], Sekhar et al. [215], Wu et al. [221],

Jiang et al. [29], Bartolini [222], and Akkaya et al. [223]). Assuming nodes' connectivity, the selection of a proper relocation of currently deployed nodes, outweigh of randomly deploying large number of node in network (Wang [38]).
Regarding the effects of nodes relocation/reposition algorithms, Guvensan and Yavuz [65] divided node repositioning into *physical movement* (i.e. movement of nodes after each step) and *virtual movement* (i.e. movement of nodes to their final destination after the iteration process ends). Virtual movement may save energy if the consumed energy of nodes' actual physical movements exceeds the energy added by increased computations and message exchanges in a network. This is usually the case as traffic message exchanges and communication processes consume less power than that of the actual physical movements of nodes. Selmic et al. [86] classified coverage control into *static* (off-line), dynamic (on-line) where in the former, node deployed in the pre-assigned locations to address a required coverage, while in the latter, mobile nodes relocate according to some specified criteria to reach required coverage in network. Nodes relocations can be controlled by centralised or distributed paradigms respectively by central operator such as base station or the local and autonomous decision-making nodes. The Authors consider one applications of dynamic coverage control in weather monitoring and hurricane tracking/prediction by using close-range sensor nodes, in order to improve precision of measured parameters (i.e. wind speed, precipitation and pressure) of the storm-affected areas which are tracked by satellites.

Defined as *dynamic coverage maintenance* (DCM), Sekhar et al. [215] proposed a constraint movement of mobile sensors to increase coverage and preserve energy in which nodes utilise the 1-hop topology knowledge of their neighbours. The process of node movements is considered as *migration*, where nodes move from their current positions (migration home points) to new positions (migration end points) as their amount of movement distances (migration distances) is limited by some particular points (migration constraint points). If the energy of a migrating nodes is exhausted and reduced to a small amount, node's movement to the new position is defined as 'migration to death' and results to a quick death of the energy-exhausted migrating node.

Wu et al. [221] present a centralised style recovery algorithm by moving (sleeping) sensor nodes toward assigned points called *empty points* such that it can cover the most number of connected orifices (uncovered arc of enclosing circle of node's sensing disk by other nodes). The power saving mechanism is applied by first sending the information and coordinates of the empty point to a centralised (base station) by the pilot sensor in order to move sleeping sensors towards the optimised (minimal) number of empty points in the coverage hole recovery process. The boundary is deduced from the listed orifices and the information about the boundary is broadcast to its neighbours by pilot sensor on the boundaries of coverage holes. Considering the problem as the global optimisation, authors try to minimise the empty points in which nodes should move to cover the hole. Based on the given idea the empty points are assigned for the moving nodes in the way that most number of connected orifices are covered.

Lin et al. [224] present a route recovery approach via controllable movement of a node performed with the help of other nodes. The controllable movement is used to protect active nodes located on the routes, with their neighbouring nodes. As a node moves in its route, its neighbours try to compensate and recover the broken linkage by relocating to the original coordinates of the moving active node. The idea is to obviate the need for pre-establishing or dynamically finding backup routes. The idea may contain fewer recovery overheads but adds a fraction of energy required for controllable movement and may not be suitable for WSNs with constrained batteries and power.

Jiang et al. [29] proposed a topology control mechanism in two methods of *active* and *passive* models where, in the passive model, a path of communications to allow the traffic flow passing is already formed. In such conditions, if the forwarded message can not reach its successor nodes, the grid is considered as the *hole*, which would be cause by one or more node failures. In the active model, each grid area is monitored by the head of grid, 'trusted node which is elected as the grid head where coverage of the grid relies on its presence'. Movement in the passive and active model depends respectively on whether nodes are aware of vacant places (holes) either after or before the moment of path construction (by exchanging the message via grid head and grid head neighbours in the active mode). Using grid head and cluster-based node movement may add complexity to hole recovery mechanisms.

Considering the wide spectrum of relocation algorithms, they can be classified into *Force-based Relocation* (Senouci et al. [225], Joshi and Younis [226], and Heo and Varshney [227]),*Voronoi-Based Relocation* (Bartolini et al. [222], Aziz et al. [228], Yang et al. [229], Wang et al. [21]), and *Flip-based Relocation* Algorithms (Chellappan et al. [132]).

- ***Voronoi-Based Relocation Algorithms***. Voronoi diagrams are one of useful structures in computational geometry [86, 98, 230–233]. To recover coverage holes caused by the absence or displacement of nodes and/or unbalanced deployment, Aziz et al. [228], developed relocation algorithms by using PSO and Voronoi diagrams. The model is aimed at minimising the coverage holes previously resulted from the Voronoi diagrams of sensors whose location information encoded in the particles. A number of Voronoi diagram vertices, as a set of interest points, are evenly distributed on the boundaries of the area of deployment to prevent sensors excessively gathering around specific points or relocating freely out side the boundaries of the region of the interest by using pulling forces. Whether the computation of Voronoi diagrams was performed centrally or in a distributed style was not made explicitly. Wang et al. [39] and Antil and Malik [91] presented three movement-assisted deployment protocols of *Vector-based* (VEC), *Voronoi-based* (VOR) and *MinMax* based on the assumption of movement of nodes from their locations with higher to lower densities using the Voronoi diagrams. Wang et al. [39] controlled oscillation for these movement-assisted deployments, especially for VOR deployments. The node relocation algorithm akin to Voronoi-based algorithm, *VorLag*, suitable for heterogeneous mobile sensor presented by Bartolini et al. [222]. In proposed Vector-based algorithm by Wang et al. [39], nodes relocated from areas of higher to lower densities. By exerted virtual repulsive forces from neighbours, nodes relocate to reach average distances among one another. By such movements nodes are to increase the coverage of their Voronoi cells; otherwise, they move to the midpoint of their target positions. Therefore if nodes' Voronoi cells are known in their movement iterations Voronoi-based node relocation algorithm can increase the coverage. In Voronoi-farthest vertex relocation algorithm, node moves toward its Voronoi cell's vertex as much as to keep its furthest Voronoi cell's vertex within its sensing range if the node detects lack of coverage in its Voronoi Cell.

- ***Flip-Based Relocation Algorithms***. The flip-based relocation algorithm by Chellappan et al. [132] represents a decremented movement of sensor nodes with the limited mobility in order to increase coverage and minimise node movement. The movement unit of sensors to their neighbors is called *flip* and there is limitation of the number of flips. The idea of these flips is to fill the cells of smaller node density. The flips are performed in a cascaded style. In the case that neighbour cells lack enough nodes for flip movements, these flips are performed sequentially to reach a uniform distribution of nodes and cover the holes and voids in the network where the determination of such flips (from the cell to their given neighbour cells) is made by the head nodes elected in each cell. Flips of nodes to their adjacent cells depend on cell resolution, node density and decision of cells' elected head nodes [132, 234, 235].

- **Force-Based Relocation Algorithms**. Many interesting results are deduced from nature as presented in the survey by [53] for wireless networks. For example, electric field lines used as optimal paths between source(s) and destination(s) when sources (sinks) can be imagined as positive (negative) charges. Many solutions to WSN challenges, such as relocation algorithms, are inspired by nature and physical phenomena, such as Newtonian law of gravity and motion (Rashedi et al. [236]), and electromagnetic relations (Toumpis [53]). These can be used to model sensors and their interactions, as each node would exert some pulling/pushing forces on the others (Guo et al. [237], Work by Kelemen et al. [238]). Guo et al. [237] proposed a relocation algorithm for coverage optimisation using three forces of *F(disturb), F(border), and F(inner)* with respect to random nodes' failure, in which nodes contracted (moved) toward a coverage hole for the repair process. The relocation algorithm is applied to all nodes in which, for a sporadic number of abnormalities or multiple coverage holes, movement and redeployment all nodes seems not to be feasible with respect to the number of oscillations and consumed energy. Thus the relocation algorithm may be more efficient if is applied only to boundary nodes or the proximate nodes of areas of interest. Kelemen et al. [238] present distributed control and self-organisation inspired by the amoeboid cell applied to robots. It uses a potential function very similar to a hormone-inspired model, with repulsion and attraction dependent on the distances of cells to one another. In this model the effect of the environment on cells is transformed into oscillation. The authors present another model of group mobility of individual cells in order to lead to movement of a group of cells by local interaction. Boufares et al. [239], for 3-D deployment scenario in WSNs requiring a given coverage, proposed a distributed deployment harnessing force-based node relocation algorithm to compensate for lost coverage and connectivity as result of random deployments. Mobile and autonomous sensor nodes can use the virtual forces for their relocations. In the force-based, potential field and forces are virtually considered such that pairs of nodes exerted repulsive and/or attractive forces which are proportional to distances of one another in order to take into account networks' coverage and connectivity [227, 240]. The nodes pulls and/or push one another based on their distances and the required constraints such as connectivity or expected density. In Selmic et al. [17, 86, 241] and its references there in presented centralised approaches whose virtual forces classified as *repulsive forces from the obstacles*, *attractive forces from the sensing areas of high interest* and *nodes pair-wise forces*. Nodes pair-wise forces are dependent on both pair-wise nodes distance and other design parameters or criteria to keep connectivity of network. Virtual inter-nodal forces of repulsion and attraction via angular/radial pseudo-forces are defined in [35, 227, 241, 242].

2.5.6.1 Sink Effects and their Mobility

The effect of sink nodes and their mobility on networks' quality and health, are considered in this section. Tong and Tavanapong [185] assume a sensor node has no contribution to coverage if its sensing data cannot be delivered to a sink, even if is active and covers some area. Huang et al. [156], considered two neighbour nodes as each other's *upstream* and *downstream* nodes respectively if they were either closer or farther from the the sink node. According to authors, nodes being upstream or downstream should be taken into account for possible topology control such nodes scheduling.

Vlajic et al. [243] presents sink mobility as a solution for balancing traffic loads, reducing node failures and increasing lifetime in WSNs. The author in [243] introduced sink nodes with the superior capabilities than ordinary sensor nodes as the gateway point to multiple sending/receiving systems.

The challenges of many-to-one systems such as *funnelling effect* and *hot-spot* phenomena and their key differences are introduced. In the funnelling effect, nodes at the 1-hop vicinity of sink nodes relay (funnel) the farthest nodes' data and traffic in addition to their own, causing bottleneck of (1-hop) nodes around sink nodes, which results in degraded network performance (i.e. lifetime, total energy consumption, packet delay) due to higher packets dropping and/or retransmission. This effect is due to a noticeable increase in transit traffic, collision and congestion as movement toward the sink node (Chen et al. [244]).

Akin to the funnelling effect, in hot-spot problems, as the result of high node traffic surrounding a sink nodes, nodes' energy is exhausted faster rates, which causes the nodes to die sooner, especially at those sinks' 1-hop neighbours, known as hot-spot nodes in network [243]. Such problems germane to the multi-hop routing (especially in many-to-one transmission) lead to isolation of sink nodes and would render the whole network non-operational. The authors in [243] suggest that bottlenecks and undesirable funnelling effects occur as result of high traffic loads while excessive exhaustion of nodes surrounding sink nodes is irrelevant to traffic intensity (as in hot-spots) and the solutions to these funnelling effects and hot-spots are different, such as using MAC-level scheduling and/or data-compression for the former, and using proper topology control mechanisms (e.g. deployment strategies possibly including mobility) for the latter.

Among the literature, Wang et al. [245] and Santi [139] have considered active mobility of sink nodes and mobile sink nodes as a solution to increase or maintain coverage affected by failures and unbalanced initial random node deployment in WSNs. [243] consider different aspects of applying a mobile sink node and limited its movement to several possible paths of regular geometric form from outer periphery, mid-periphery, diagonal cross, and mid-cross. By having longer network lifetimes due to the presence of mobile sink node in various shapes of trajectory, the authors suggest that sink node mobility may not always be more efficacious than sink node in the real applications (e.g. Zigbee-based WSNs), unless the amount of traffic overhead generated by sink mobility throughout the network is properly controlled.

Response implosion, as one of the important problems in WSNs, is leading bottleneck in the areas of sink nodes when they are inundated by large numbers of nodes responding simultaneously to monitoring requests (Paradis et al. [16]). The authors suggested sampling of densely populated sensor networks, self-orchestrated operation, and diffused computation toward sinks as possible solutions.

Shih et al. and the references therein [246], have introduced failure compensation and recovery mechanisms to maintain the connectivity of nodes and paths to sink nodes when some nodes surrounding the sink nodes fail due to excessive energy consumption. Shih et al. [246] applied a grade diffusion algorithm and genetic algorithms to improve the health of networks aimed at replacing fewer failed sensor nodes and reusing as many routing paths towards sink nodes as possible. The author introduced directed diffusion (DD) and grade diffusion (GD) algorithms. The former algorithm, as a query-driven transmission protocol, aims to reduce the data relay transmission counts for power management where collected data is transmitted if it matches the query from the sink node (the the sink's queries being in the form of attribute-value pairs to be broadcast by other sensor nodes).

The latter algorithm, in addition to creating the sensors' routing, also identifies a set of neighbours for decreasing traffic loads as nodes are able to select neighbours with lighter and/or higher energy for the role of extra operation.
Nakayama et al. [61] proposed mobility models for mobile sink nodes as they traverse through clustered sensor nodes and collect data from the visited clusters with optimised and energy-efficient routes and trajectories which make network more fault-resilient with respect to failed and malfunctioning sensor nodes as the result of attacks.

2.5.7 Hybrid Schemes

In recent years, some topology control schemes have been combined together and considered as *hybrid schemes* to better address the emerging challenges in WSNs (Hua et al. [247], Labrador [182], Santi et al. [183], Ren and Meng [248]).

Node relocation and transmission power control (power-based TC schemes) are amongst the most popular strategies employed to address node failures and CHs [31, 182, 190, 215]. Power adjustment can improve connectivity/bi-directionality of current and/or newly deployed nodes based on locally collected information and decision making procedures [160, 182]. Power-based topology control schemes are generally suitable for small-scale CHs and random node failures [31, 182, 183] and can react faster and more accurately to local and dynamic inter-nodal modifications than physical movements and thus prevents any unnecessary nodes oscillations. However, for large-scale CHs, such topology control scheme is not suitable as the nodes would rapidly drain their batteries if they continuously transmit at a higher power over large distances which results in enlarging the CHs. Physical node relocation can react more flexibly by balancing and/or modifying node distribution and inter-nodal distances especially in the vicinity of damaged areas. However, relocation algorithms can suffer from forming new small CHs due to autonomous and distributed node movement. Improper node movements can exhaust battery supplies and reduce their ability to reach the coverage goal [38, 227, 231, 234]. The node relocation algorithm known as *Distributed Self Spreading Algorithm* (DSSA) [227] addresses coverage problem and uniformity, requires a fairly large number of iterations and a long convergence time. The relocation of mobile sensor nodes also is required to be customised for different terrain conditions (e.g. using hopping sensor nodes instead of wheeled nodes for rugged terrains [249]).

Therefore, applying both node relocation and power adjustment in harmony allows a network recovery process to be faster, more energy efficient and more stable. Recently, *hybrid* topology control schemes, such as [182, 250, 251] have been developed to enable mobile WSNs to adjust both their location and transmission power to modify or maintain optimal network coverage. The joint adjustments are important to reduce the energy requirement for recovering the CHs. As a result, the network lifetime and resilience can be improved.

In general, hybrid topology control requires centralised coordination to keep track of topology changes. However, accurate temporal and geographical knowledge of the CHs are not always available due to the nature of CHs' random occurrences. Repeatedly adjusting the topology and measuring the coverage would be necessary, even in centralised approaches [182]. Obtaining the optimal solution by any centralised algorithm would increase computational complexity and delay due to large numbers of interactive nodes.

Exploiting the concept of hybrid topology control, Zhu et al. [250] formulated an *(constrained) exact potential game* for sensors nodes with a *sensing model* of the directional footprints and finite *angle of views*. Each sensor can take actions of moving or adjusting its angle of view/range based on local information (i.e. its own energy level and neighbouring nodes' actions). However, this model has a limited sensing scope due to its angle of view. Therefore, an omni-directional sensor model is considered so that a wider sensing scope can be provided, and the network can be considered for wide range of applications and scenarios.

In practice, a distributed hybrid topology control is of value [27, 182, 250] where nodes can have limited information about non-neighbouring nodes. In the event that the network is compromised with CHs, each surviving node can autonomously and spontaneously react. Each node measures uncovered areas around itself, interacts with its neighbours, and takes distributed energy-efficient actions to extend the coverage of the remaining surviving network to cover the CHs. The potential benefit of this approach is that nodes can optimise their own energy usages, meanwhile contributing to the overall lifetime of the network. To guarantee the convergence of such a distributed approach is not straightforward, due to the unavailability of the global view on the entire network at each individual sensor node.

2.6 Network Events and Faults

In this section the concept of events in WSNs is briefly defined. In the following sections, among many types of events, they are considered within the context of node failures/faults and different types of holes in network.

2.6.1 Events and Faults

Events are defined in King and Nittel [252] as *'the drift from the predictive model which is characterised by discrete spatio-temporal dimensions.'*, which can be detected by monitoring, sensing the deployed network area, as well as comparing the measured/sensed quantities with predefined thresholds. The event thresholds are considered as *landmarks* to make prospective decisions [253] on status quo of networks. [253] defined the *boundary of event* as a separator of event-space from non-event-space. The boundary of an event contains the location, shape, and size of an event. As in Ammari [254], events can be classified according to the nature and factor of their occurrences, in which nodes either detect or receive notification of given classes of events in a network.

By considering the limitations and constraints of error-prone sensors, WSNs survey of fault management by Paradis and Han [16] suggests monitoring and reacting to physical events. The authors [16] classified the source of faults as: *sensor node failure* due to battery depletion and destruction by external events; capturing, sensing, and communicating incorrect data due to the nature of ad-hoc network; *temporary and permanent links failure* due to external objects, environmental conditions, or nodes moving and reaching out of their ranges; and *congestion*, which causes packet loss.

Mahapatro et al. [255] considered calibration errors, malfunctioning hardware, hostile environments, law battery, and link failure as sources of failure. Mahapatro et al. [255] used definitions of fault and failure of nodes from the literature, which are referred to as *deviation of nodes from their expected behaviour and sudden expected behaviour in the system.*

Yuvaraja et al. [256] divided faults into *node failures* and *network failures*, where node failures are detected by neighbouring nodes monitoring one another as *group detection*. In network failure, topology of network dynamically changes which nodes may not be aware of that due to their local and limited vision.

Selmic et al. [86] consider different levels of abstraction for networks failures. The authors coined fault management in WSNs as *fault-aware WSNs* which can detect, isolate and mitigate faults at different system levels of *sensors (components)*, *nodes*, *network* and *system* levels when faults occurs in those levels. According to the authors such fault management and awareness would increase robustness and reduce the required redundancy of networks but affect the trade-off for higher costs for networks. For example, embedding fault identification algorithms in nodes requires more storage and computing resources.

Regarding the frequency, Selmic et al. [86] classified faults as *high variance* and *low variance*. The authors considered *offset fault* where sensed data values offset by constant value from true measured phenomenon. The authors modelled constant and time-varing (drift fault, shifting from real values with the time) offsets faults, which have slow and fast varying reading differences between sensor node and its neighbouring nodes. A constant offset is modelled as a slow-varying difference between a sensor's measurements and the true value being measured.

Roy et al. [257] and reference therein presented *dumb* nodes which, despite sensing within their ranges, are not capable of exchanging information and communicating with their neighbouring nodes. Such problem is caused by reduction of dumb nodes' transmission ranges due to the environmental circumstances (e.g. temperature variation, rainfall and fog). In the following sub-section, traditional fault-tolerance techniques and fault management strategies in the literature are briefly reviewed.

2.6.2 Fault Management Techniques

Regarding different network events, especially network failures, fault-tolerance techniques in networks are considered by [15, 16, 110, 152, 153, 255] are mainly classified into *pre-event strategies* and *post-event strategies* in which the former are those strategies devised to predict possible failures and pinpoint vulnerable or critical parts of a network along with the possible countermeasures before occurrences of events by using parameters or schemes such as redundancy. The latter are strategies geared for and triggered after occurrences of (unpredicted) events. Pre-event strategies would usually be effective for those events which can be accounted for in the design stage and have designated resources. However, they may not be able to guarantee suitable responses for unpredicted events. Post-event strategies would react more swiftly and interactively to events with the available resources in the network, but they may not guarantee optimal actions or fully address them (e.g. failures).

Due to the catastrophic consequences of having unreliable and faulty sensor nodes, Mahapatro et al. [255] brought attention to the importance of *fault diagnosis* in WSNs. The authors Mahapatro et al. listed the importance of fault diagnosis and the effects of faults in real applications, such as military (for example, battlefield surveillance and attack), environmental (for example, tsunami, forest fires), medical (for example, Body sensor networks), habitat, industrial, agricultural monitoring and target tracking, as fault diagnosis may be critical to warning systems and may help prevent loss of life from dangerous events if network faults exceed defined tolerable levels.

Paradis and Han [16], classified faults-tolerant techniques as *fault prevention, fault detection, fault isolation, fault identification, and fault recovery.*

- ***Fault prevention***. In the design stage this is considered to prevent faults by ensuring desired coverage and connectivity using (constant) monitoring techniques (passive and active monitoring), and enforcing the data delivery paths at different (OSI) layers.
- ***Fault detection***. Depending on types of failure and applications, different symptoms are used to detect and discover the occurrence of faults. Usually, destination nodes are responsible for detecting failure (packet loss).
- ***Fault isolation, Fault identification***. Different types of fault indications and hypotheses can be used. As the nodes 'piggyback' the ID of their

neighbours along with their own information towards sinks, sink nodes are able to have a clue about the topology. In addition to constantly updated fault identification, faults may be isolated and prevented from the further progressing.

- ***Fault recovery***. In this stage, network recover from (reverse) or alleviate effects of faults via different recovery techniques and topology control schemes.

Yuvaraja et al. [256] tabulated different fault management mechanisms such as fault detection, fault recovery and repair. Laprie [150] listed and fault management techniques as *fault forecasting*, *fault tolerance*, *fault removal*, and *fault prevention*. Yu et al. [15] classified fault management architecture as *centralised*, *distributed*, and *hierarchical* approaches. In *centralised* fault management, the decision making processes of nodes rely on either logically or geographically central nodes such as base stations, central controllers, and sink nodes, as the central nodes are assumed to have unlimited energy and/or resources. Centralised approaches have higher accuracy and efficiency, but may not be applicable or scalable for resource-constrained and/or time-sensitive networks due to the high volume of messages to and from centralised nodes, which quickly deplete sensor nodes' energy proximate to central nodes. In *distributed* approaches, local-decision making processes and fault management are based on the nodes having a certain level of decision discretion before they communicate with central nodes; such approaches are scalable but may not reach globally optimal solutions. In *hierarchical* approaches, the burdens of decision making processes and fault management are distributed to different parts of networks as intra and intera-clusters, which are suitable for energy balanced and scalable applications.

Yu et al. [15] divide WSNs' *fault management* processes into, *fault detection*, *diagnosis* and *recovery*. In the fault detection process, by central or self-detection mechanisms, the fault causes are detected. In the diagnosis process, detected faults not only can be identified but also are differentiated from irrelevant false alarms. In the recovery process, structure of network is modified towards preventing faulty nodes to impact further its performance.

Lee et al. [258] proposed detection and isolation of faulty sensor nodes in WSNs. The authors use node sensed data between the neighbours and dissemination of the test result in order to improve the accuracy of detection. Malfunctioning sensor nodes would still be permitted to play the role of relaying nodes for communication in network (if they are fault-free and able to communicate), however they are logically isolated with respect to their sensing operations from network. The authors considered the probability of failure in network and addressed transient faults in communication and sensing via time redundancy and acceptance of performance degradation.

Fault detection can be considered as *passive* or *active*. In the former, nodes use available and received data without request, enquiry or triggering any information exchange for detection purposes. For example, such detection can be performed by comparing the statuses of node's neighbours within a given time period or threshold. Such detection may be considered in cases where the power of signal of node's neighbours suddenly drops or the number of a node's neighbours decreases as the result of failure. In active detection, nodes explicitly send some packets and enquiries to their neighbours asking for confirmation of their presences and receive their responses (e.g. 'hello' or 'ack', 'nack' messages). Passive detection saves network resources and is performed with much less overhead, while active detection have more accurate and up-to-date information at the cost of imposing more load and energy on a network.

Selmic et al. [86] presented different fault detection methods as,

- ***Statistical methods***. In these methods (mathematical) models of healthy data are defined. If behaviours do not match the defined models, they are flagged as faults. Statistical methods though efficient are not able to robustly classify those events and phenomena which are not within their define models.
- ***Fault classification methods***. Unlike the statistical methods, these methods apply mathematical models for the faulty data. Therefore, by examining WSNs' incoming data, behaviours can be mapped to different class of modelled faults. These methods are able only identify faults previously modelled and can not detect faults which do not fit within previously modelled faults.
- ***Nearest neighbour methods***. These methods use the comparison of nodes and their neighbours' data to model faults. Faults are detected If set of nodes with similar features (i.e., the spatially nearest sensors) have significant differences in their readings. In order to benefit from these method, appropriate neighbourhood should be defined.
- ***Clustering-based methods***. These methods consider the clusters of nodes rather sensor in their comparison. The faults are detected if the compared measurements of cluster with its related clusters are not similar. By acknowledging the difficulty of properly forming the clusters, clustering-based reduce amount of computations for fault detection in network.

Similar to [152, 153], in Figure 2.6 depicts recovery stages and fault management. We classified fault management mainly as *Pre-Fault*, *Detection Reconnaissance*, *Maintenance/Repair* and *Post-Fault* phases.

- **Pre-Fault Phase**. In this phase, the focus is on designing and predicting different types of faults which may occur within the life time and operation of network with respect to various parameters such as environments, devices, type of scenarios and applications, required performance and quality of service.

- **Detection Phase**. In this phase, Faults are detected by adjacent nodes and information are forwarded toward nodes' upper layers and other neighbouring nodes. In the *dampening* technique, depending on number of failures within a certain period of time, a *hold-off time* is considered for stabilisation of *flapping* resources whose state change frequently between being 'operational' or 'faulty' [153].
- **Reconnaissance Phase**. In this phase, fault-detecting nodes notifies their neighbouring nodes by forwarding in order to *diagnosed* and *identify* type of failure based on the correlated information.
- **Maintenance and Repair Phase**. In this phase, depending on many factors such as type of faults (i.e. faults' scales and locations, progressive/passive), type and distribution of deployed sensor nodes (i.e. static, mobile and hybrid nodes) and their available resources, WSNs' applications (i.e. real-time, security-sensitive) and environments. Fault can be repaired, avoided, or isolated in order to either restore or maintain network' operations and services to the acceptable levels.
- **Post-Fault Phase**. In this phase, there would be an assessment on the available resources and recovery of the faults (i.e. no, partial or full recovery) to check the possibility of switching back to original resources. The reason is that the original resources such as primary paths may be more ideal than backup paths with respect to length. Such recovery is considered as *revertible* and nodes detect and be notified of repaired faults with a given time to ensure their stable recovery. Although the WSNs' disconnected partitions may be repaired by deploying mobile nodes that act as the relay node [259], revertible recovery in WSNs may not be easily applicable till all the failed nodes are replaced.

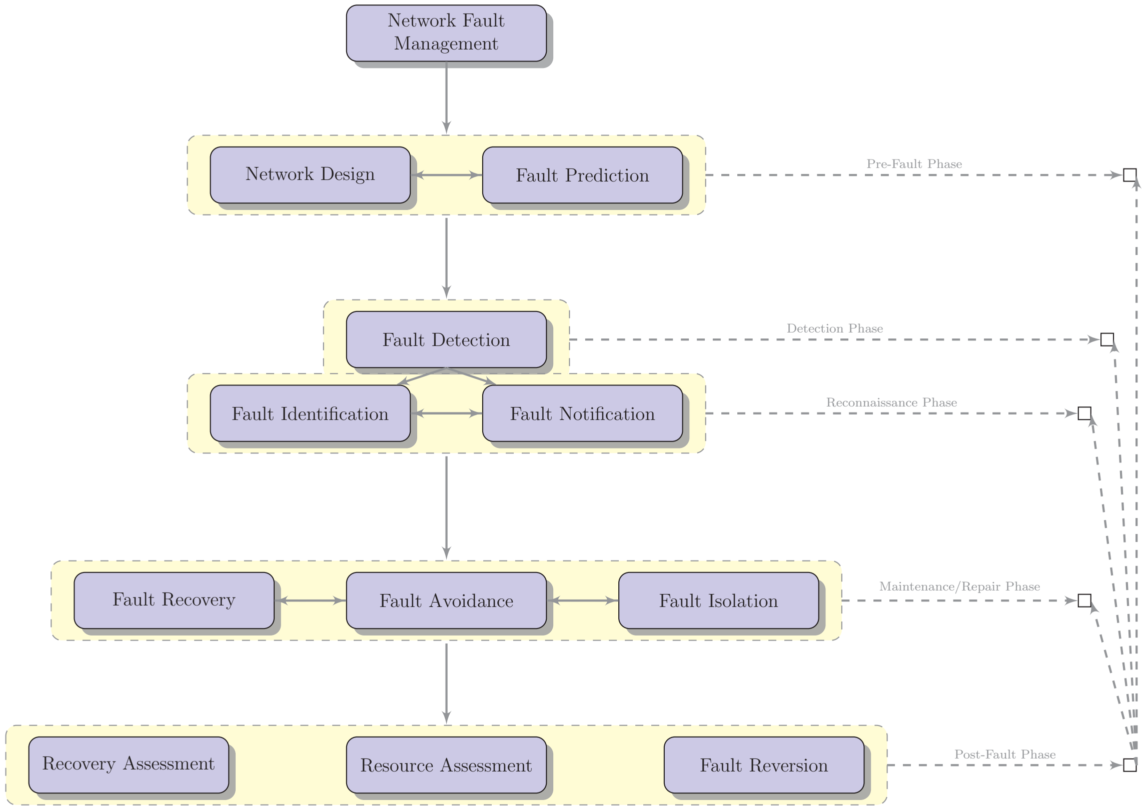


Figure 2.6: Network Failure Management and Recovery Stages

For different types of faults also refer to Selmic et al. [86].

2.6.3 Network Failures

It is generally considered that network failures happen mainly due to *node and link failure* [153], [152]. Al-Karaki et al. [211] also mention *failure* and *blockage of sensor node* as the result of lack of power, physical damage, or environmental interference. Younis et al. [110] consider node failure as *fail-stop* scenarios in which nodes cease to operate as they fails.

Node fault can occur in different OSI layers and levels in WSNs, which should be considered in order to reduce failure and inability of nodes to properly perform their roles and tasks in networks [260,261]. To prevent networks from partitioning, Younis et al. [110] classified fault-tolerance techniques as *precautionary* (fault-tolerance is provisioned and considered at deployment), and *reactive* (through real-time restoration of lost connectivity and/or coverage) at initialisation and during normal operation. The authors in [110] classified node failures as *single node failure*, *non-collocated k-nodes failures*, and *collocated k-node failures*, and considered node relocation, providing provisioned tolerance (e.g. by designating k-spares for each critical node or form a k-connected topology), placement of stationary relays, and using mobile entities (such as relays, data collectors, etc.) as the suggested strategies to address different types of node failures. Precautionary schemes cannot always address collocated node failures. Depending on the choice of either centralised or distributed recovery procedures, the reactive technique considers schemes of relocating mobile nodes to repair damage, static, and mobile deployment of relay nodes to reducing the relay counts and to increase the connectivity. By providing a comparison of the reactive schemes in the paper, authors considered the following factors in these schemes:

- ***Node Selection Criteria***. Nodes are chosen and selected depending on certain factors, i.e., distance to the failed node, node degree, and criticality, before or after the occurrence of failures.
- ***Network and Node State***. These states are required for movement determination, which may or may not be updated and forwarded to proximate nodes. The range of knowledge of node states may vary from 1-hop neighbours to the entire network.
- ***Centralised and Distributed Operation***. Movement of nodes is locally and autonomously determined or centrally supervised by base-stations.
- ***Movement Type***. Node movement may be independent or in accordance

with other nodes such as in cascade mode. Movements may also be constrained due to environmental conditions or a node's limited power and connectivity to the rest of network.

- ***Performance Metrics***. In addition to certain primary objectives such as connectivity/coverage, energy preservation or network lifetime, other secondary objectives may be considered in designing topology control and fault-management techniques.

Younis et al. [110] consider multiple-failures resulted from simultaneous and spatially-independent single failures which have spread over network. Each of these failures can be addressed individually. Other types of multiple-failure are caused by simultaneous and spatially-dependent single failures (correlated failures) as in (coverage) holes or cascaded failures.

Mahapatro et al. [255] brought their own classifications of faults based on the duration, underlying cause and behaviour of failed components. Based on the behaviour of sensor nodes, the authors divided faults into *hard* and *soft*. Nodes with soft faults are still able to function and communicate with networks with the modified behaviour, unlike with hard faults. Nodes' soft faults which result from events within sensor's environment, are expected not to be hard to trace or detect. This is because, the soft faults' indications, effects and causes may rapidly disappear.

Regarding duration, according to [255], faults are considered to be *permanent* or *temporary*, where the latter is further classified into *transient* and *intermittent*, which is usually caused by faulty hardware and software of node. Unlike transient faults, intermittent faults occur frequency and may ultimately lead to permanent faults if the problem persists. Presenting classification in the literature, [255] viewed the faults as either *crash* (battery depletion, faulty transceiver, complete damage or natural phenomena), *omission* (failure or delay in sending messages, relay, due to malicious, natural or due to human), *timing* (natural or due to human causing sensors to respond too soon or too late), *incorrect computation* (nodes unable to send correct measurements even if they have received or sensed correct/genuine data), *fail-stop* (where a node fails due to battery depletion and may alert its 1-hop neighbours of this natural or human-made fault), *authenticated Byzantine* (information may maliciously be modified which affects messages' authenticity), and *Byzantine* (sensor nodes send some messages which compromise security and attack the network).

Mahapatro et al. [255] listed factors influencing fault diagnosis design for WSNs as *resource constraints* (of processor, memory, bandwidth, energy, etc.), random deployment (dropped or deployed sequentially), *dynamic network topology*, *attenuation and signal loss*. The authors also classified the fault diagnosis techniques as *centralised* and *distributed* approaches where, for the former a central supervisor (e.g. a sink node, base station) is present to collect data and propagate decisions throughout the network. In the distributed approach, which entails less overhead and communication, each sensor has local vision of fault states from its immediate neighbours.

Lee et al. [193] divided node failures into 'single node' (due to battery depletion, malfunction, external hazards) and 'simultaneous failure of collocated nodes', where the former are addressed by providing proper bi-connectivity, and the latter would be fixed by redeployment of relay nodes to connect the segments partitioned by failures. The authors suggested that redeployment of relays ameliorates the effect of multiple collocated nodes and connects resource-demanding inter-partitions. The authors propose a 2-vertex-disjoint path by placing relay nodes between autonomous segments (partitions). Mannan et al. [262] considered network faults as *node fault*, *link fault*, or *sink fault*. Link fault may be the result of instability of links between nodes such as interferences or even mobile obstacles, and so on, requiring more appropriate design than replacing failed nodes and sinks.

2.6.3.1 Cascaded Failures

Increasing the complexity of connections between the elements of systems (e.g. number of paths) and increasing the redundancy of system functions may increase resiliency. However, increasing the complexity and redundancy of systems may reduce their robustness if components fail one after another (Kott et al. [112]). Thus, as the elements of systems are not independent and are triggered by one another, robustness against *cascaded resource failures* should be taken into account as possible precondition of network resiliency. Otherwise, the smallest disturbances or failures may lead to cascaded failures, known as *domino effects*. Hence, susceptibility of networks to cascaded failures and the propagation of failures should be considered in network design [263–265].

2.6.3.2 Network Holes

Robinson and Knightly [187] defined *coverage dead spots* where there are little or no probability of connection from the given location. According to Zhang et al. [266] network holes result from the variable and limited sensor nodes' lifetime. Authors of [266], classified holes into, ***energy holes*** *which result from unexpected wear off nodes' energy leading to breaking the process of data forwarding* and ***coverage holes****, blank surveillance areas, which result from disabled nodes due to failure or environmental influences.* Antil and Malik [91] and Gosh et al. [267] considered holes as one challenge which degrades and interrupts the operations of a network. Holes were caused by nodes dying of limited battery, random and improper deployment of nodes, presence of obstacles. Selmic et al. [86] mentioned that detecting coverage hole is major research area which has major impact on the QoS of networks. Proper detection of coverage holes helps to ameliorate their effects such as the minimum phenomena [24,268] caused by greedy routing. Otherwise, nodes' energy exhaustion around holes expanded them further and worsened the network performance. Gosh et al. [267] classified holes as coverage hole, routing, sink and jamming holes.

Ahmad et al. [24] explained the network holes in terms of (1) coverage hole and routing are usually due to physical factors (i.e. inability of hole area nodes to communicate or route traffic), and (2) jamming, sink/black and worm holes are due to security issues and deliberate attacks . Huang and Wu [269] consider nodes' energy constraint/exhaustion, natural disaster, and initial uneven distribution of sensors as the causes of coverage sensing holes. The authors defined a hole as : '*Given a set of sensors and a target area, no coverage hole exists in the target area, if every point in that target area is covered by at least k sensors, where k is the required degree of coverage for a particular application*' [269].

Antil and Malik [91], Jabeur et al. [270] and Selmic et al. [86] has provide taxonomy of sensor node failures as *physical/logical/malicious/semantic* (PLMS) holes:

Semantic holes are relevant to the processing of data.

Logical holes are caused by the cluster-based approach when sensor nodes do not receive support from neighbours in the same clusters.

Physical holes are caused by limited processing, computation capacities, or sensing ranges, defined as *processing holes* (energy exhaustion of nodes resulting in energy holes), and *coverage and routing holes*. Coverage holes are due to random and imbalanced node deployments. Topological and dynamic behavior of networks also causes such holes. Routing holes are due to insufficient ability of sensor nodes to route traffic data through given paths.

Malicious holes are caused by malicious nodes. Such holes may be further categorised as *jamming*, *sink* and *trust* holes. *Jamming holes* result from preventing communication among sensors by using powerful signals in their assigned frequency of sensors nodes. *Sink/black holes* may be due to denial of service (DoS) attacks, in which an attacker introduces fake and 'more interesting' paths and routes to sinks to be used as the next hop for relaying information. *Trust Holes* result from improper or untrue sensor assigned, application dependent weighted values known as *trust values*. Trust values are obtained from different parameters to their neighbours.

Although malicious holes should not be neglected, coverage holes are the most prevalent types of holes in networks. Coverage holes can be considered as a criteria on status of network with respect to degree of coverage and general indicator of network health (Antil and Malik [91]). Selmic et al. [86] noted that clear definition of coverage hole should be brought. Some other classifications of coverage holes are as follow (see also Figure 2.7), Temporal coverage of a region may be suitable where a constant *spatial coverage* is not required or cannot be guaranteed by nodes for the given area - and where the energy consumed by node mobility is not a major limitation over the duration of coverage. Mobile CHs can also be created by mobile obstructions in a particular area [271, 272]. CHs can be alternatively considered as either *permanent* or *transient*; in the former case, CHs and their effects persist in time, while in the latter, CHs would fade away with time.

- ***Convex/Non-Convex CHs*** (LaValle [273], Gazelle and David [274]) The overall shape of coverage holes due to correlated nodes may be convex/non-convex. It may be difficult to address non-convex CHs. However, a solution, non-convex coverage holes are considered as union and overlap of convex shapes.
- ***Stationary (Static)/Mobile CHs*** (Chang et al. [128, 129], Antil and Malik [91]) Coverage holes can be mobile due to group replacement of sensor nodes in a network as these nodes relocate to new positions in uncovered areas, leaving their current positions uncovered. [128, 129] considered the

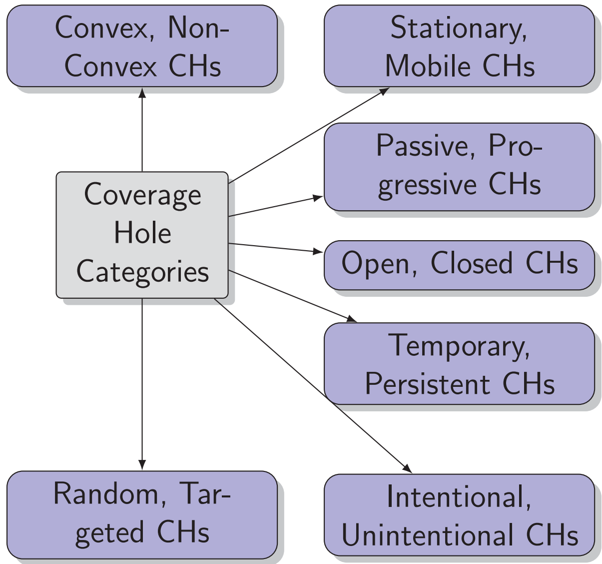


Figure 2.7: Categories of Coverage Holes

movement of mobile nodes equal to the movement of holes in the opposite direction in turn lead to the *virtual movement* of CHs. [91] defined holes formed in static sensor networks as *static holes*. Therefore, as the result of node mobility connection are formed or vanished which resulted in *mobile holes*. On the hand, group node mobility introduces the concept of *temporal coverage*, as nodes have to periodically sweep a given area but do not cover the entire area at once.

- ***Passive/Progressive CHs*** (Antil and Malik [91], Ahmed et al. [24]) If holes expand as the result of nodes' energy exhaustion,it can be considered as progressive, especially in proximity to coverage holes due to the effect of greedy forward routing known as *local minimum phenomenon* [24, 268]. Otherwise, they can be defined as passive coverage holes as their size is unchanged.
- ***Open and Closed CHs*** (Tong and Tavanapong [185]) Regarding the boundaries of coverage holes in general, they can be considered as bound

(closed) and unbound (open) respectively, whether the area of coverage holes is surrounded by boundary nodes or not.

- ***Temporary and Persistent CHs*** (Antil and Malik [91]) regard holes caused by external events (e.g. wildfire, heavy rain) as *persistent holes* if they destroy and permanently damage nodes with the high probability.The paper attributed *temporary holes* to the insufficient wake-up scheduling of sensor nodes in target regions. Static obstacles can also be considered as persistent coverage holes, while moving obstacles that block and disturb proper communications among nodes can be considered as temporary CHs.
- ***Intentional and Unintentional CHs*** (Antil and Malik [91]) in *Unintentional holes*, nodes accidentally lose some of their capabilities while *intentional holes* are formed by intentional scheduling of parts of target fields and collections of nodes to be turned off or put to sleep mode in order to preserve the energy.
- ***Random or Targeted CHs*** feature and occurrences of random coverage holes can not be predicted in advance. However, targeted CHs are those coverage holes which have formed in particular places due to the presence of specific nodes such as sink nodes or by specific attacks to the area.

Coverage holes are modelled by a variety of geometrical shapes. Virtual moving hole (virtual movement of hole in the opposite direction of concerted movements of a set of nodes [128, 129]), different geometrical modelling of coverage holes (in the form of ellipsis [26], gap in rectangular tube of deployed nodes [275], as a grid of small square cells [29], hexagonal shapes of coverage and hole cell [129], in the form single and/or sequential multiple circles with a determined/random radius and locations from used in this work and combinations of convex-shape to form non-convex shaped hole [273,274]) are among some of interesting efforts in addition to the conventional modelling of single and/or random failures in networks. The authors in [128, 129] used hexagonal shapes to model coverage and hole cells. It should be noted that the size, width and height of the holes are important factors on power consumption behaviour and type of hole movement. In [129] hexagonal shapes is used to model the coverage and hole cells. Each cell has 6 directions towards its 6 hexagonal neighbours with which has common edge.

Coverage holes are not always the result of correlated node failures, and may instead be due to voids (e.g. by improper or incomplete node deployments or movement) [24, 276, 277], or static/mobile obstacles [272, 278, 279] in the network.

Regions of nodes' Voronoi cells outside of nodes' sensing range are defined as blind spots in [84], while an ellipse surrounded with a margin of B-nodes [277], a gap in a rectangular region of deployed nodes [84], a set of small cellular squares lacking elected and active node [29] , hexagonal cell shapes [129] are among many proposed model for coverage hole in the literature.

Similar considerations should be taken into account for the different types of obstacles in the network. A fire-spreading model [280, 281] can be considered as an example to model similar aggressive/cascaded mass failure of sensor node due to fire. Dilating circles can be also considered as a model of symmetrical collocated failure due to explosion or gas/chemical expansions. It is also assumed that residual energy of nodes located out of damaged area, would affected by D-event in real scenarios such as explosion and hole expansion.

2.7 Network Recovery

Vasseur et al. [153] defined progression of recovery phases as the *recovery cycle* in networks.

In the following section, different stages of network recovery are briefly reviewed.

2.7.1 Recovery Stages

Network *recovery cycles* can be performed based on sequential actions/stages where, despite possible overlaps, they can be classified into *detection*, *identification*, *notification*, *recovery and repair* (Vasseur [153], Grover [152]). The recovery stages and principles for WSNs [84] could more or less can be emanated from those of wired networks [153], [152], [282]. They may require some modifications. It should be noted that these stages may indistinguishably span across one another. Alternatively parts of stages either are absent or added depending on the nature of the implemented networks, the scenarios, and their applications. For example, this may vary in optical networks, WSNs and even smart grids and power distributions (including hybrid wireless and wired network architecture) (Figure 2.6).
Among the many factors compromising networks' lifetime and health, here the focus is narrowed to recovery of networks from coverage holes caused by physical failures and damages, rather than security and attack (e.g. black holes). For more detail of various types of attacks on different layers and the classification of methods of defences against these kinds of attacks, refer to survey by Lima et al. [283].

2.7.1.1 Detection, Notification and Identification

One of the main reasons for using WSNs is primarily to detect and report *deviations* from normal patterns of sensed and measured data from deployed sensors in given area. Zhang et al. [284] classified detected deviations as *noise and error*, *actual events* and *malicious attacks*. Deviations as anomaly of measurements in WSN, are considered as *outliers*. In addition to detecting such anomalies via adjacent nodes, their status, sizes and scales should be identified. Nodes either independently detect anomalies or are notified by intermediate nodes if they are further from the events. In both cases, exchange of information is performed by mechanism of receive, update and send (forward) of data among nodes. Therefore, nodes have a better perception of the behaviour and dynamics of events. The identification of event/anomalies allows the nodes to make more candid decisions on quality and quantity of prospective recovery processes if deemed necessary.

In Lee et al. [193], the presence and occurrence of collocated failure nodes is detected by failed nodes' 1-hop neighbours in the case of major and sudden degradations in communication traffic. The authors consider these 1-hop neighbour nodes as *border nodes* which forward/notify all of their reachable nodes about failures. At the notification stage, nodes, especially at the margin of events, propagate their perceived information about an event to other nodes via their neighbours. Events notification allows node to make proper decisions to address events in an appropriate and timely manner.

2.7.1.2 Recovery

In the recovery stage, depending on the nature of a network, different strategies may be applied to alleviate the effects of events in the network. Some of these strategies can be considered under the umbrella of *topology control (TC)* schemes such as *range change* (power adjustment), *node relocation*, *node redeployment* (i.e. appending new nodes to the deployed area). Each strategy would be devised to meet the given objectives, in which, by combining strategies, it is possible to achieve trade-offs and multiple objectives. Such hybrid recovery strategies have yet to mature. The ability of a network to heal itself is important in many human inaccessible or unfriendly, or disastrous environments, such as bush fire, earthquakes, and remote areas. A swift network recovery can avoid the interruption of network services (i.e., surveillance and monitoring), and guarantee fast responses in case of emergency events [160, 182, 250, 251].

2.7.2 Coverage Hole Recovery

The process of fighting the (wireless) network holes can be divided into hole detection, hole and boundary localisation, and hole repair. By finding boundary nodes, holes can be located and bypassed. Ordinarily traditional ways of fighting holes would be either by detouring (avoiding holes) and/or using auxiliary mobile nodes which can be sent inside holes to compensate for cases where boundary nodes' ranges are not sufficient to cover them. Each of these methods has benefits and drawbacks [285]. There is a possible alternative way of hole recovery by benefit of hybrid use of both detouring techniques and mobile nodes. Using homogeneous and synchronized sensor nodes, Huang and Wu [269] divided the recovery mechanism into two stages, a *network initiation phase* and a *network maintenance phase*.

- ***The Initiation Phase*** of random (static) node deployments is considered into three steps of *gridding* (partitioning networks into uniform grids of the same length by the assumption of nodes having similar energy, selecting grid head node, and later replacing head node with nodes of the highest residual energy in the grid); *hole detection* (spotting the location and size of holes by sink nodes using the status of each grid, sending the hole repairing request (HRR) from the grid head node if the residual energy plummets under a certain threshold); and *hole recovery* (designating mobile nodes by sink node to recover coverage hole with respect to the virtual force of each pair of mobile nodes and the grid hole).
- ***The Maintenance Phase*** is a period during which network connectivity

for recovery of event (hole grid as the grid whose coverage is lost) is being monitored. In the recovery model, mobile nodes (if available) between sink node and holes are considered for possible movement to repair a hole. The sink node checks if a node's residual energy exceeds the threshold and chooses node with higher virtual forces. Such processes are terminated if the coverage ratio (the ratio of areas being covered in two consecutive iterations) becomes lower than a given threshold. The problem of sink node failure, centralised operation, decision latency, and large coverage holes are still challenging in the grid-based recovery model.

Coordinating between mobile agents themselves and also with boundary nodes is a challenging issue. Coverage hole recovery stages and phases are presented in the following section.

2.7.2.1 CH Detection

Antil and Malik [91] consider the first stage of coverage holes is to detect their boundaries and the immediate nodes around them. The authors mention that hole boundaries can be found by considering the areas of interest such as fire, floods and earthquakes. If repairable, such aforementioned areas usually require a higher density of sensor nodes in order to be recovered. The authors also categorised coverage hole detection algorithms based on the type of used information, computational model and network dynamics. They, along with Fayed et al. [286], classified CH detection as *topological*, *statistical*, and *geometrical* approaches. In [287], statistical, topological, graph theory and geometrical approaches for boundary detection and estimation are presented.

- ***Geographical, Graph, Geometrical approaches*** consider the position of sensor networks in advance by localisation such as via GPS or other methods. Antil and Malik [91] reviewed different algorithms used to detect holes and avoid the local minimum problem around coverage holes by notifying nodes in proximity to coverage holes and further from boundary nodes to flexibly decide and forward traffic around these areas. Graph and geometrical boundary estimation methods are used in both centralised and distributed strategies.

- ***Topological approaches*** use available connectivity information such as degree of nodes as the criteria for detecting holes. By using the concept of Rips complex where holes are classified as non-triangular or triangular and Cech complex, in order to discover the coverage holes, locations and distribution of nodes are not required (Antil and Malik [91], Feng et al. [288], Kanno et al. [289], Selmic et al. [86] and Hanaf et al. [290]). Due to algorithmic complexity, centralised implementation and limited network resources, most topological boundary estimation methods are impractical and remain in the theoretical domain.
- ***Statistical approaches*** apply statistical operations on the collected information from nodes' neighbours, and the final decision of being boundary nodes are made based on Boolean functions. Therefore, some statistical characteristics are due to coverage hole and damage. The locations or density of sensor nodes among the information which is usually required in these approaches. For example, average density is used to detect of holes by comparing actual node degrees with a predefined threshold value [91], and a centrality index [291], in which inner nodes have higher centrality indices than outer nodes. Statistical methods usually work well in dense networks [287].

Geometrical approaches usually require nodes' locations using GPS or other localisation methods, while statistical approaches provide proper performances with the possibly higher cost of computational complexity. Topological approaches add more overhead and complexity to current approaches despite providing more realistic models [91]. Antil and Malik [91] divided algorithms based on the computational model into *centralised*, and *distributed* in which the former algorithms run on one or more nodes in the given locations and holes detected are mostly based on connectivity information. In distributed algorithms hole detection is distributed among nodes, especially boundary nodes around coverage holes. For example, the idea of using attractive and repulsive virtual forces among nodes to keep the computational complexity limited to the neighbours of each nodes by a accepting degree of oscillation among the participating and decision making nodes with local vision [225]. Antil and Malik [91] consider the case of hole detections algorithms triggered by nodes to be either directly or indirectly affected by coverage holes. Corke et al. [292] considered a convex hull around the collocated failed nodes, wherein the local detection by those nodes directly affected by the hole, triggers the detection algorithm. For local detection, initially nodes send pings to get information from their neighbours as the initial connectivity state is required in order to be used later in the nodes neighbours' periodical message transmission and updates. By counting the number of no replies and comparing it with a given threshold, nodes can deduce failures within their ranges. Failed nodes would then be added to the recently computed convex hull of the failed nodes. The author in [292] used path density as a tool to detect coverage holes. As packets travel and are updated via nodes from source to destination, with respect to the concept of *geometric spanners* [293], *path density* is defined as the ratio of straight line distance of source-to-destination to actual distance travelled. The presence of holes in the network can be inferred from the low ratio of path density.

Tong and Tavanapong [185] introduced the *geometrical* scheme of boundary recognition based on the overlap of nodes' sensing ranges. Regarding the relation of transmission and sensing ranges, it is assumed that boundary nodes are joined together which may not always be the case. The authors used the concept of an *orifice* in unit disk graph (UDG) sensor nodes to classify different types of nodes. Orifice is defined as arc of enclosing circle of node's sensing disk, which is not covered by any other sensors. With the union of all network orifices, it is possible to define a contour for holes both inside and outside the boundaries of the network. So, the paper defined the *outer* and *inner boundary of nodes* as the union and connected orifices, which are either subject to the outer area or to the coverage hole. Outer and inner coverage hole boundaries can be distinguished by aggregation and union of the list formed from uncovered arcs (orifices of the sensors). Orifice-based boundary determination schemes rely on determining the regions where node communication ranges intersect and overlap. However, the implicit assumption of connectivity amongst all remaining nodes is not always valid in practice. A survey by Chintalapudi and Govindan [294] divided the networks' zones into *exterior*, *interior* and *edge* sensors with some thickness. The detection approaches are based on the concept of pattern recognition theory. In the *statistical scheme approach*, by using the concept of detection theory, edges are detected with respect to a given threshold. The performance of approach is evaluated in terms of false alarm and miss detection. In the *digital image processing approach* sensors are considered as pixels. Similar to image's pixels, irregularities of sensor nodes are amended with the introduction of *weighted average* of nodes' neighbours values. With the idea of applying the method of pattern recognition theory to pictures' pixels, via the *classifier approach*, nodes are partitioned into different zones. Using partition lines, the exterior, interior and edge nodes are detected relevant to the partitioned zones. Benefiting from local information, communication graphs and connectivity of nodes, Dinh [295] proposed a topological scheme of detecting boundary nodes. The author used 2 and 1-neighbour knowledge so that nodes are able to decide if they are on the boundary. With such knowledge a *neighbour ring* can be formed around nodes. If a neighbour ring is connected around a node, the node is considered not to be a boundary node. The author divided nodes' neighbour boundaries into two subclasses, SB and MB, in which the former are nodes near to one boundary and the latter nodes are near to least two boundaries. The authors has not compared this approach quantitatively but has discussed it verbally and qualitatively with other similar boundary detection and edge recognition approaches.

Kanno et al. [289] introduced solution for determining the number of holes and their locations, which is applied in non-planar communication graphs. The paper mentioned that the existence of hole can also be due to the features of environment (e.g. lakes) or certain occurred phenomena (e.g. fire). The paper tries to pinpoint nodes around holes as hole boundary nodes by using non-planar embedded information. The hole-equivalent planar keeps a record of hole by number, position in a network. This topological method using the concept of simplicial complexes is claimed to be applicable to both coordinate-available and coordinate-free schemes. The authors [289] consider maximal simplicial complexes for reducing the complexity of divide-and-conquer algorithms applied to plane graphs and for decreasing computation time in finding the number of coverage holes and identifying these holes by their corresponding faces in the underlying planar graph.

Using the barycentre and nodes' local information, Khedr et al. [296] presented the distributed and localised algorithms to verify whether the homogeneous nodes are interior or located at the perimeter of given network. From barycentric coordinates, nodes can judge if they are inside or outside of formed triangles from the set of points.

In Fletcher et al. [297] active nodes know about their neighbours by periodic 'hello' messages and locally get the number of arcs within their sensing range perimeters which are not covered by other active nodes. The geometrical approach by McLurkin and Domaine [298] uses a cycle algorithm to detect the boundary of network distributedly based on the number of nodes and their positions, angles formed with respect to their neighbours. The algorithm checks if nodes are exterior or interior boundaries of a network. It uses the cycle algorithm to show whether the node is within the cycle of its neighbours or not. If the node is not within this cycle, then it is either an exterior or interior boundary according to the convexity and concavity of cycles formed by its neighbours, as well as the angle of the node with its neighbours. If the network is dynamic and nodes are mobile the algorithm should be applied frequently by accepting some overhead.

Fayed et al. [286] presented a heuristic-based method for finding boundary/edge nodes (i.e., the edges of network or interior holes) locally by using a *local convex view* (LCV) computed around the given nodes. The algorithms are applied to each of the nodes by computing the convex hull around given node. If the nodes is inside the cycle (the computed 1-hop convex hull convex hull), it is considered not to be a boundary. Zhang et al. [299] used the idea of node clustering boundary estimation rather than using individual distance, angle, and location information of WSNs. The authors defined boundary estimation as finding nodes proximate to boundaries of sensor fields and hole boundaries, which would be generated by improper initial node deployment. The authors classified nodes with respect to their relocations to the cluster head as *cluster interior nodes* and *cluster boundary nodes*, where, in the former nodes resided within the radii of at least two clusters, while in the latter node located within neighbourhood of only one cluster head. Based on the number of nodes and their density, the method's accuracy may vary. Deogun et al. [300] presents a boundary discovery algorithm in networks of heterogeneous nodes that have global unique coordination which nodes are aware of the geometric distance to each of their neighbours. In the authors' model, The authors defined a network as being sufficiently dense that a network's density should be such that each node has at least 3 neighbours. It is also assumed that no communication holes exist inside network among nodes of bidirectional connections. The authors consider a node as *interior*, if it is located inside triangle formed from its neighbours; otherwise, node is *boundary node*. If the node has at least three neighbours, such triangles are formed by using the *choose good neighbours algorithm* (CGN) by finding the three best neighbours (i.e. closest to the given node). The perimeter of the entire network, *network boundary* is an imaginary line resulting from connection of all boundary nodes in the network. Considering the absence of (communication) holes in networks in the paper, the boundary in this case is the perimeter of the entire network which is not realistic. If any holes existed in a network, it is expected that they severely affect the performance of algorithm.

2.7.2.2 CH and Failure Repair

In this stage, sensor nodes are triggered to act such that networks either partially or entirely recover and repair coverage holes and ameliorate the undesirable effect of node failures. For networks of static and mobile nodes, Lee and Younis [301] proposed an algorithm using the minimum Steiner tree to reconnect segments of networks partitioned by failure or physical damage or attack. It tries to move mobile and boundary nodes as to lead the relay nodes of each segment to join the segments together. As the proposed algorithm tries to optimise the number of relays needed to be deployed to join segments and to maintain connectivity in segments, it may move relay nodes in the cascaded style. The paper argues that the optimal count and position of relay nodes for restoring connectivity among the segments of networks is NP-hard and should be addressed heuristically. Detecting failures by perceiving the sudden drops in communication or traffic, or through the un-reachability of certain remote nodes, the authors define *boundary nodes*. Dini et al. [275] present a scenario in which mobile nodes are dispatched to connect the gap between two partitioned sensor networks. The mobile nodes try to connect two parts by keeping connectivity with the rest of network, which is done by using information from the boundary static nodes. It is assumed that the locations of nodes are known to the base station. Mobile sensor nodes are sent by the base station, which centrally detect gap, to form a bridge in the deployed sensors in a tunnel shaped area. The authors propose a *partition detection system* (PDS) on the base station which can detect the network's partitions and their positions. With respect to the place of base station, nodes are considered in *safe partition* (partition of undamaged nodes which entails the base station); *gap* (partition of damaged nodes); and *isolated partition* (partition of undamaged nodes whose nodes are disconnected from the safe partition). Dini et al. [275] assume the centralised method of gap detection and the scenario occurs in the tunnel shape of sensor networks. Despite the presence of proximate static nodes to gap, most of the operations and decision making are done by base station and mobile nodes. If the base station and/or mobile nodes are not at appropriate locations for the location of gap, the required time for the relocation and repair process of mobile nodes are expected to be longer.

Tong and Tavanapong [185] present a movement-assisted deployment called *moving algorithm for non-uniform density* (MAND) geared for a non-uniform node deployment. MAND implementation is based on the idea of Voronoi diagrams. Extended MAND (EMAND) is also proposed to reduce and/or smooth nodes movement towards the desired point efficiently to avoid node blocking the desired area as they move toward.

Shen et al. [198] used a hybrid deployment of homogeneous sensor nodes (mobile nodes either in sleep or working modes) and presented a coverage hole recovery model in which the closest mobile node moves to the (non-overlapped) coverage hole in order to repair and/or maintain the connectivity of the network. In such recovery, newly formed holes in current locations of moving nodes would be filled by other mobile nodes closest to them in the case of enough redundancy. The authors presented an algorithm to help the holes to heal and to maintain network connectivity. The paper's approach seems to be effective for smaller scales of holes, with the assumptions of mobile nodes availability to maintain k-coverage and/or to retain connectivity of network. The algorithm would not be effective for larger scale coverage holes, where such holes have to be decomposed into smaller ones. The authors provided k-coverage in the damaged area by dispatching and adding K' required remaining number of mobile nodes to the damaged area, which may already have had k'' mobile nodes ($k' + k'' = K$ where $k'' \geq 0$) if holes' proximate mobile nodes are available.

An adaptive self-configuration of a swarm of robots in which robots configure themselves into triangular lattices is presented by Lee and Chong [302]. The triangular lattice method is implemented by nodes' local interactions of pair of dynamically selected mobile nodes whose 1-hop information such as their local positions are exchanged with one another. Such local interactions decrease mobile robots' computation and is expected to be flexible to environment conditions. Having the positions, numbers and distances of robots from one another, each robot detects the position of other robots within its sensor boundary, and the triangular configuration is defined by three distinct positions and equilateral configuration. As robots move they select their new neighbours according to the minimum perimeter condition formed with them.

Zhang et al. [303] presented a dynamic proxy tree-based framework to address communication path disconnections from source to destination. Path disconnection is caused by many factors such as nodes' movements and failures. The authors presented the tree-based framework which nodes join, are added to, and move to form a minimum Steiner tree on Euclidean plane from source to destination. Updating the spanned tree from the source to destination was done via two distributed on-line algorithms (shortest path-based and spanning range-based) and one on-line centralised algorithm.

Fletcher et al. [297] attributes the coverage hole and impaired coverages to stochastic node droppings, unpredictable node failures despite the locally redundant node deployments in WSNs. Bestowing mobility to sensor nodes, enables nodes to repair such coverage holes at the same time adding more complexity, hardware design and cost due to local nodes' interactions. In order to design a small group of mobile robots to repair coverage holes so as to avoid cost and complexity increases, the authors under the name proposed algorithms,*Randomised Robot-assisted Relocation of Static Sensors(R3S2)* in which the aforesaid mobile robots move randomly and gather redundant sensors (which are in either passive or active states) and deploy them to coverage holes' locations in a network. In order to gather most of the redundant sensors in field, random movement of mobile nodes is confined within virtual grids where are from the current grid to the least recently visited grids. Two algorithms R3S2 and G-R3S2 are proposed: in R3S2, robots move to region of interest randomly in order to address sensing holes, while in G-R3S2, robot movements are limited in order to reduce randomness, expected traversal times, and coverage repair delays.

Deng et al. [304] consider failures of exhausted energy or destroyed nodes in physical environments, the causes of coverage holes in WSNs. The authors proposed a *best fit node policy* (BFNP) for coverage hole repair in which coverage hole detected by a base station and the closest inactive nodes are dispatched to the centre of the minimal coverage circles of polygons that surround holes as replacements for failed node(s). In terms of quality of coverage, energy consumption and life time of replacing single failed node in network, the proposed policy outperforms the coverage hole patching algorithm (CHPA), that selects points on perpendicular bisectors of hole-boundary edges whose distances to two boundary nodes are equal to the communication radii, and deploys nodes to those points. However, the BFNP requires base station to centrally detect coverage holes which may be far from it. The paper's approach is similar to the idea of Voronoi-based relocation algorithms in repairing the areas of not being covered (using Voronoi-diagrams). The authors benefitted from the redundancy of inactive nodes to repair emerging coverage holes. Although the authors attributed coverage holes to node failure, they did not properly model correlated failures and larger scale coverage holes which, would normally be bigger than a failed node.

Huang and Wu [269] address the hole problem and propose the relocation using the concept of virtual force to prevent further excessive power consumption and elongate network lifetime. Though node deployment can address sensing coverage problems to some extent, it may not always be an appropriate remedy to emerging CHs. Therefore, the authors proposed controlled mobility of mobile nodes (MNs) for the repair and recovery of coverage holes for hybrid node deployments. The authors proposed a grid-based hole recovery mechanism by choosing appropriate nodes in the repair process based on their residual energies.

Ma et al. [137] accounted obstacles, environmental conditions, types of random and/or improper deployment (i.e. dropping from aeroplanes), node failures, and malicious attacks affecting quality of coverage for holes in the network. Using the potential field in the relocation of nodes, such holes can be autonomously repaired. To address the oscillation problem of node relocation, based on Newton's law of motion, the concept of network dynamics is where artificial potential functions are presented for capturing the objectives and the environmental conditions/limits of mobile sensor nodes. The authors consider the deployment problem by applying the concept of potential energy among nodes. The potential energy of mobile sensor nodes is minimised by their movements; these movements should be terminated as they reach their minimum energies. The paper proposed a parallel and distributed algorithm (parallel and distributed network dynamics (PDND)) where nodes relocate with respect to the applied virtual forces obtained from the information they periodically propagate. The authors applied their model for target tracking as sensor nodes were dragged to the target by the attraction force. In the proposed extended version of PDND the deployment time is reduced with a compromise on overall coverage.

Using mobile and static nodes along with idea of Voronoi diagram, Wang et al. [305,306] proposed relocation algorithm in which static nodes obtain coverage over their Voronoi cells and trigger a bidding process to their neighbouring mobile nodes to patch the coverage holes of Voronoi cells. Mobile nodes based on their prices choose the best bid related to distances from the farthest Voronoi vertices and update their prices if they accept the bids from neighbouring static nodes. Bidding process can reduces the possibility of forming newly coverage holes and patching one hole with multiple mobile nodes as the result of inappropriate movements of mobile nodes.

2.7.2.3 Avoidance and Detour Stage

In addition to hole recovery, avoidance and detouring, reinforcement of the overall energy of nodes proximate to coverage holes or damaged areas may be considered as the objectives of increasing networks' resiliency. Avoidance and detour can be performed in the form of either by moving nodes away from the CH, or constructing detours and bypassing traffic around the CH [24, 26, 276, 277, 307]. Needless to say, better hole detection would provide more reliable and clearer views of boundary and communications paths around such regions of interest (ROIs), which results in less performance degradation after the occurrence of events (holes) [137, 277, 307–312] such as the chance of minimum phenomena and cascaded failures around damaged area(s) [24]. Otherwise, incomplete routing and data forwarding strategies may affect whole networks' operational ability. In addition to WSNs, some examples in other fields are provided in this section to support this type of resiliency for WSNs. Shin et al. [311] define problem of hole expansion in WSNs as *hole diffusion* caused by excessive energy consumption and failure of hole boundary nodes. The authors account existing hole's perimeter routing and improper hole's detouring schemes for such boundary node failures.

The idea of coverage hole avoidance by boundary nodes can be similar to the work of Scott et al. [313] and Lee et al. [314–316], in which herds of prey animals try to evade predators; as the 'boundary nodes' (the prey) relocate around (move away from) a 'coverage hole' (a predator). The idea is to alleviate and/or avoid undesirable and detrimental effects of damaged area (fault region) such as minimum phenomena and cascaded failures [24,268,309] (as in aggressive coverage holes such as fire, gas, or steam explosions) on the boundary nodes and rest of network [137]. Lee et al. [314–316] introduce the deformations of prey flock due to predator attack. With a single predator and a single prey flock, predator attacks towards the centre of flock are modelled based on the bias random walk. The authors introduced the stage–wise response and reaction of the prey to the predator based on the size of the flock. Flock responses were classified as *compression, expansion and relaxation* in this sequential manner. The authors presented a flock's motion and escaping behaviours in a predator's attack in terms of *prey-prey* and *prey-predator* interactions with reference to repulsion and/or attraction forces. A predator moves to the centre of a prey-flock with a given angle of attack with respect to the flock's and the predator's directions of movement. The shape of scape of the prey-flock may be different based on many parameters such as predation risk radii, distance of predator to centre of flock, angle of attack, velocity of predator towards prey flock, amount of repulsion/attraction forces among prey-prey and prey-predator and flock size. The authors used three Reynolds laws [145] to model the prey-flock predator behaviours. Having many applications, similar mobile nodes behaviour model can be applied to maintain the integrity and tolerance of WSNs termed *evasion strategy* to (progressive) coverage holes caused by physical attacks and cascaded failures.

Jia et al. [310] presented a *hole avoidance in advance routing protocol* (HAIR) aiming at avoiding holes rather than bypassing them (i.e. Hole Avoiding Re-Routing Protocols). The idea is similar to the marching process of a queue of people passing by a barrier en route when they have not, or when they have limited information about the existence and location of barrier. Thus, if the information is exchanged by the person closest to the barrier toward its successor in the queue, the barrier can be avoided in advance instead of bypassed by long detour. This idea entails that if a given node is marked as a hole node or a hole detecting node, it broadcasts the message to its one-hop nodes, such that each node's neighbours mark it as a hole node. In this way, each node's neighbours may have a list of hole nodes which are within their one-hop distances. In the case that a node is not a hole node or has not detected the hole, it tries to find a neighbour nearest distance to a sink.

Change et al. [308]) proposed a scheme to avoid holes. Sensors bordering holes in their approach actively establish a forbidden region to enable packets to be guided around holes and move along a short path from hole to sink node, thus incurring less energy consumption.

Benefitting from controlled mobility in nodes, Ma et al. [137], inspired by Newtonian Mechanics, applied the potential functions to introduce a dynamic movement algorithm for autonomous sensor nodes. The proposed algorithm is applied to address malicious attacks of jamming node passing through the sensing field by avoiding the jamming hole, then repairing, and filling the area of moving jammer. The algorithm implemented into two phases of *escape* and *reconstruction*. In the escape phase, nodes move to a safe region away from moving jammer and connect

Figure 2.8: Network Resilience Factors and Indicators

to the rest of the network. Then, in the reconstruction phase, nodes relocate and uniformly cover the hole caused by the moving jammer.

2.7.3 Resilience Factors in WSNs

Figure 2.8 depicts different network resilience factors some of which may have overlap with one another. There are many other factors which can affect resiliency of network which should be further studies in more details.

Performance and QoS of network in either presence or absence of fault and events can be consider as an indicator for level of resiliency in network. For example, the duration that a network can provide or maintain its services or its integrity, when facing the faults, can be consider as a measure for the resilience of network. Introducing new performance metrics which can more realistically reflect the resilience of network to different types of events and faults should also not overlooked.

Designing network with respect to different parameters and devising an appropriate failure recovery and fault management, have significant impacts on the resiliency of network. Networks can show different degrees and strengths to the different types of events and faults. For example, network resiliency and behaviours can severely change in the presence of CHs, energy holes or security hole in the network.

Mobility pattern, localisation and routing protocol have importances in either effectuating or nullifying the implemented recovery mechanisms. For example, improper localisation of nodes may reduce the accuracy and performance of recovery mechanism and topology control schemes.

Selecting topology schemes to address events can enhance the resilience of network. For example node relocation can be a solution for large coverage holes; however, it is not an energy efficient scheme to repair random node failure throughout the network.

2.8 Cooperation and Emergence

This section, briefly presents the concepts of complex systems, autonomy of complex systems, collaboration and cooperation behaviour, and emergent cooperation among such systems and nodes.

2.8.1 Complex Systems

Downing [317] considers complex systems to consist of many interacting components or agents that: exhibit *distributed control*, meaning no centralised component predicts or manages the components' interactions; are *Synergistic* such that the sophistication or complexity of the gestalt greatly exceeds the summed complexities of the components; are *emergent self-organising*, meaning global structure arises solely from local interactions; are *autopoietic* where the global structure is maintained by local interactions; are *dissipative*, so that the system is far from equilibrium and yet is stable; and are *adaptive*, meaning the system can adapt its structure and/or behaviour. According to the author, it may not be possible to predict the global behaviour of complex systems from simple analysis and component interactions. The author lists examples of complex systems and their properties of self-organisation and emergent cooperation behaviour such as in rain forests, termite mounds, traffic jams, epidemics, global economies, the Internet, immune systems, migration patterns, stampedes, stock markets and bacterial colonies among many others.

Newman [318] gives a brief introduction to complex networks such as the Internet, social and biological networks, and predicts their characteristics, their failures and epidemic processes.

Wilensky et al. [281] suggested that predicting, controlling and managing complex systems is challenging, and modelling and analysis of complex systems should provide a rough simulation rather than a perfect prediction.

Some of the properties of complex systems are given in the following sections.

2.8.2 Autonomy and Multi-Agent Systems in WSNs

According to Oliehoek and Amato [319] systems such as robots, computers, person, etc. which *'receive information and make decisions about how to act in the world'* can be considered as 'agent' and the *'agent interacts with its environments over sequence of time steps'*. Wilensky et al. [281] introduced the Reynolds Boid model (three roles of coherence, alignment, and avoidance among agents [145]) and agent-based model architecture. The authors defined 'agent' as *an autonomous individual element with properties and actions in a computer simulation*. According to the authors, in *agent-based modelling (ABM)*, the world is modelled by agents, environment, and a description agent-agent and/or agent-environment interactions where some details at the cost of others are neglected. Wilensky et al. [281] considered systems the result of many parts having interactions with one another, leading to an *emergent outcome*. Such emergent outcome is based on interactions and feedbacks between the emergent behaviour and decision-making individuals. The authors distinguish 'emergence' as action of the whole, which is different from just the sum of parts' actions; which such emergent behaviour may affect the decision of individuals.

Nicholson et al. also [96] introduced the multi-agent system paradigm [320, 321] as a solution to model a implementation of networks in which each autonomous nodes make individual and intelligent decisions within their vision based on conditions, constraints in the deployment environment. Though these nodes have their own local goals, they may act towards global goals. Vernon et al. [322] present aspects of agent and robot autonomy and surveyed wireless sensor networks in the context of multi-agent systems (MAS). The paper present the suitability of multi-agent systems concepts to model autonomy and flexibility for distributed and decentralised sensor networks. The authors present a survey of agent technologies applicable to trends of node autonomous behaviour and the challenges of applying MAS technology to nodes.

Boccardo et al. [323] provide an example of human group mobility and social interactions among group members. The authors present groups of characters moving in a virtual world similar to the flock model based on the leader–follower which in each platoon there is a leader that has knowledge about the path or destination and the others have to coordinate their directions based on the leader's decisions. The number of leaders depends on the number of flocks and the degree of scalability. Even in the case of a small number of leaders, a whole group can be guided onto the right path and destination. The similarity of operations of mobile sensor nodes and multi agent systems (MAS) has lead to many solutions which are inspired by physics and biology (Floreano and Mattiussi [324]) in localisation and tracking [325], vehicle collision avoidance control [326], knowledge sharing [327], independent navigation without communication [328] and (distributed) flocking algorithms [143], among other.

2.8.3 Emergent Cooperation and Collective Behaviour

Rubenstein and Wei-Min [334] by assuming knowledge of the gradient map of every point and addition or removal of robots, present a method of self-assembling of robots by forming their shape despite uncontrollable movement. Therefore, by communicating with their neighbours, homogeneous robots are able to show collective behaviour. It is assumed that such robots have a consistent coordinate system and know the total number of robots participating in the collective behaviour;however, in reality such information may not be available on large scales with high numbers of robots.

Canizo et al. [332, 333] provide an introduction to different models of agents' collective motions and behaviours. The authors presented individual-based models of the collective behaviour of swarming based on different factors of repulsion, alignment and attraction and their asymptotic speed with the attraction-repulsion interactions. The work represents a survey of swarming which aims to relate the biological collective motion of particles to its counterparts in physics (such as hydrodynamics and macroscopic motion) is formulated via differential equations and kinetic theory.

According to Sumpter [335], the levels of granularity and modularity in animal and the group systems' collective behaviours, for example ant trails, cockroach aggregations, fish schools, bird migrations, honeybee swarms, web construction by spiders, and locust marching, unlike certain other systems such as protein interactions or ecologies are easily distinguishable while the complexity of animals is higher than proteins and cells, for example which makes it harder to clearly describe the behaviour of individuals. The author defined two approaches to study collective animal behaviour : *functional* and *mechanistic*, where the latter considers animals' interactions to form group level patterns (communication mechanisms), and the former explains the evolution of behaviour through natural selection and reason of individual cooperation to form and persist with collective patterns despite conflicting interest among individuals.

There are many examples of where autonomous agents not only react to one another, but may also react to an external influence forming sort of collective behaviour through which an emergent cooperation can be seen among autonomous agents, as in ant nests, termites, flocks of birds, schools of fish and the Internet (Burgess et al. [212, 213]). According to author, due to vague boundaries, designing system having either *engineered collaborative properties (aiming for direct*

causation) or *emergent collaborative properties (emerging by indirect causation)* sometimes is not straight forward. Each agent based on its limited view plays its role to keep the collective promises of the whole to form a *collective super-agent*. The author in gives consideration to *communication of autonomous agents* with limited views to coordinate their intent and to negotiate objectives/incentives of cooperation which may explicitly or, implicitly propagate among agents.
Forrest [329] defined *emergent computation* when 'a global behaviour emerges from many local interactions. The author in [329] defined emergent computation in which from performing explicit instructions and interactions among collection of agents, an implicit global pattern are obtained. Having reached such a relationship, the emerged global behaviour may reciprocally affect behaviours of local interactions. The author in [329] considered important and overlapping themes of system indicating the emergent computation as self-organisation, collective phenomena and cooperative behaviour.

Downing [317] defined *emergence intelligence* as the intelligence emerging from the collection of stochastic interactions of agents/complex systems, which eventually are captured and understood but may not be fully predicted. Acknowledging the power and flexibility of emergence of collective intelligence's abilities in many applications, many seminal works have been inspired by nature, life, and biology. Some examples are : Swarm Intelligence (SI) (Engelbrecht [336], Bonabeau [337], Beni [338]); ant colony optimisation (ACO) (Dorigo et al. [339]); artificial bee colony algorithms (ABCs) (Karaboga [340]); glowworm optimisation (Krishnanand and Ghose [341, 342],Liao et al. [343]); and fireworks algorithms (FWA) (Tan et al. [344], Zheng et al. [345]). Please refer to Tan and Zheng et al. [344, 345] for a brief list of evolutionary algorithms (EA) and Xing et al. [346] for a comprehensive list of emerging biology-based computational intelligence (CI) algorithms with their performances. Kulkarni et al. [93] have conducted a survey of computational intelligence in WSNs. The authors listed the computational and memory requirements, flexibility and optimality of CI paradigms and approaches which applied to WSNs.

Some of definitions of the complex systems' and agents' cooperation and collected behaviour in the literature, that can be applied in WSNs, are presented in Table 2.2.

2.9 Conclusion

This chapter, the distributed and mobile sensor networks are briefly introduced. Categories of coverage and mobility in these networks are listed. The concepts of resiliency, especially in WSNs and topology control schemes are briefly presented. The concepts of events, faults, and network failure types have been summarized. Fault management and recovery stages for network's failures in general, and recovery of coverage holes in more specific details are explained. The idea of cooperation, emergence in complex systems formed by autonomous systems/agents, collective behaviour and its implication in WSNs are provided in this chapter. Finally, the open issues and challenges and further directions have been explained.

Author(s)	**Yrs.**		
Forrest [329]	1990	**Emergent Computation**	A global behaviour emerges from many local interactions. From performing explicit instructions and interactions among collection of agents, an implicit global pattern are obtained. The emerged global behaviour may reciprocally affect behaviour of local interactions
Doran et al. [49] and Steels [330]	1994 1997	**Emergent Cooperation**	" Reflex or reactive agents simply act. Therefore, when cooperation occurs it does so without reflection upon possible actions. There is no prediction or predictive planning and therefore no intention."
Cao et al. [331]	1997	**Agent Cooperative Behaviour**	"Given some task specified by a designer; a multiple robot system displays cooperative behaviour if; due to some underlying mechanism .i.e.; the "mechanism of cooperation"/; there is an increase in the total utility of the system. Intuitively, cooperative behaviour entails some type of performance gain over naive collective behavior."
Nicholson et al. [96]	2008	**Emergent Cooperation**	Each autonomous nodes make individual and intelligent decisions within their vision based on conditions, constraints in the deployment environment. Though these nodes have their own local goals, they may act towards global goals.
Boccardo et al. [323]	2009	**Group Mobility and Social Interaction**	Groups of characters moving in a virtual world similar to the flock model based on the leader–follower which in each platoon there is a leader that has knowledge about the path or destination and the others have to coordinate their directions based on the leader's decisions. Whole group can be guided onto the right path and destination.
Canizo et al. [332, 333]	2010	**Collective Motion and Behaviour**	As the result of individual interactions of repulsion, alignment, attraction and their asymptotic speed with the attraction-repulsion interactions, a behaviour of swarming and collective motion is formed.
Burgess et al. [212, 213]	2014	**Collective Super-agent**	Autonomous agents with limited communicate to negotiate their incentives to cooperate. Agents based on their limited views play their roles to keep the collective promises of the whole and act in the form a *collective super-agent.*
Wilensky et al. [281]	2015	**Emergent Outcome**	Systems of many parts having interactions with one another, leading to an emergent outcome that is based on interactions and feedbacks between the emergent behaviour and decision-making individuals.
Downing [317]	2015	**Emergence Intelligence**	Intelligence emerging from the collection of stochastic interactions of agents/complex systems that eventually are captured and understood but may not be fully predicted.

Table 2.2. Some definitions of the complex systems' and agents' collected behaviour and Cooperation in the literature.

CHAPTER 3

Coverage Hole Detection

3.1 Introduction

Wireless sensor networks (WSNs) are increasingly being used to track objects and monitor various phenomena or areas of interest [1, 2]. Physical damage and/or power exhaustion of nodes may lead to coverage holes in WSNs. Automatic identification and repair of network coverage holes have become one of the key research areas concerning WSNs as coverage holes drastically affect networks' topology and operation [24]. Based on the nodes' position relative to the *damaged area* (D-area), nodes can be classified as undamaged or damaged (U-node or D-nodes) [300]. Termed as event-based or triggered coverage holes, they can be directly detected by certain proximate nodes known as *boundary nodes* (B-nodes) due to the changes they instigate on their B-nodes (such as signal or connection loss). Since B-nodes detect the D-events autonomously within their respective ranges, when such an event occurs, the D-areas may become surrounded by a *margin of B-nodes* (MB-nodes) (Figures 3.1 and 3.2).

If all B-nodes take part in the hole-recovery processes, either by increasing their transmission power or by relocating towards the region of interest (ROI), the probability of collision, interference, disconnection and isolation may increase, which can affect a network's performance and *quality of service* (QoS). A number of distributed *boundary node selection algorithms* (BNS-Algorithms) are proposed which may address these issues. These BNS-algorithms allow B-nodes to self-select based on available 1-hop information extracted from their simple geometrical and statistical features. D-nodes can be regarded as virtual nodes if their information and pre damage locations are used by the other nodes to implement BNS-algorithms. By applying the proposed distributed BNS-algorithms, despite little or no message exchange overhead and limited available information, cooperative behaviour among the selected B-nodes emerges [49]. B-nodes selected by the BNS-algorithms are denoted as *selected boundary nodes* (SB-nodes). Selection algorithms are only applied on a limited set of nodes adjacent to the damaged zone (MB-nodes) instead of all nodes in the network. In this chapter, the results show that the performance of the proposed distributed BNS-algorithms approaches that of their centralised counterparts.

3.2 Method and Assumptions

Parameters, scenario and the boundary node selection algorithms are presented in the following sections.

3.2.1 Nodes and Deployment Area

3.2.1.1 Sensor Nodes

Sensor nodes are assumed to be deployed in a two-dimensional (2-D) rectangular area, denoted by $\theta = [x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}]$. Each sensor node i can be modelled

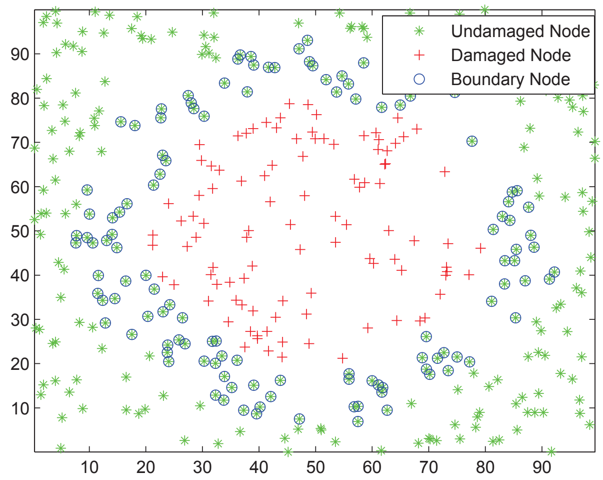


Figure 3.1: Coverage Hole

as a *unit disk graph* (UDG) with a transmission range of R_c^i and sensing range of R_s^i [32,347,348]. Let $\mathcal{V}$ presents the indices of the nodes; where $i \in \mathcal{V} = \{1, 2, \cdots, \mathcal{N}\}$. Each nodes' location is determined by its disk's centre,

$$C_i = \{(x_i, y_i)|x_{min} \leq x_i \leq x_{max}, y_{min} \leq y_i \leq y_{max}\}.$$

It is assumed that all nodes are homogeneous ($R_c^i = R_c$, $\forall i \in \mathcal{V}$) and a pair of nodes are bi-directionally connected if they are within the transmission range of one another. Each node is aware of its own location via GPS or some other localisation methods [101, 349]. The set of neighbours of sensor node i is defined as

$$\mathcal{N}_i = \{j \in \mathcal{V} \backslash \{i\}, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R_c\}.$$

Here, to avoid unnecessary complexity, transmission range (R_c) and sensing range (R_s) are considered to be equal. Sensors may be mobile or static. Whether nodes are fixed or mobile, it is assumed that at the time of observation each node has up-to-date and valid information about itself, its neighbours and its surroundings.

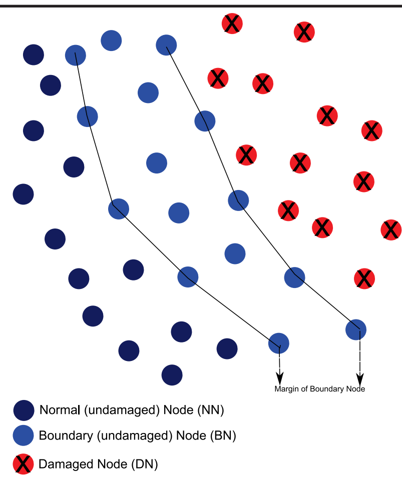


Figure 3.2: Node Types

3.2.2 Coverage Hole

Among different types of network holes in [24,91] the D-area (Coverage hole), k can be considered as a circle of a radius r_{Hole_k} with a centre at (x_{Hole_k}, y_{Hole_k}). These circles represent the occurrence of the events at the location where the failure of nodes triggers and may expand isotropically due to the nature of the physical damage. By combining of multiple CHs of different centres and radii, any arbitrary and more complex holes with either convex and non-convex shapes can be easily approximated [273,274]. It is assumed that B-nodes are aware of their degrees (the number of local one-hop connections) both before and after a D-event, while they are only aware of degrees and distances of their UN-nodes and DN-nodes prior to the D-event. Occurrence of coverage holes in wireless sensor networks are referred as the D-event in this scenario. It should be noted that noise, false alarms, or transient, periodic, and frequent failure of nodes, and link instability are excluded in the aforementioned model.

3.2.3 Nodes Classification

Regarding their locations to damaged areas, deployed nodes are classified as *damaged nodes* (D-nodes) and *undamaged nodes* (U-nodes). D-nodes are a set of failed nodes that reside in the damaged areas. The set of U-nodes may be directly and/or indirectly affected. It is assumed that U-nodes can individually detect damage events if at least one D-node is within their ranges. If U-nodes detect the D-event within their ranges, they are defined as *boundary nodes* (B-nodes); otherwise they are known as *normal nodes* (N-node). The D-event is detected by B-nodes as the these nodes sense any significant changes within their ranges (i.e. signal loss or disconnection due to the failure of their neighbours). D-nodes can also be called *virtual nodes* (V-nodes) because part of B-nodes' decisions are based on the locations and degrees of D-nodes. At the time of observation each node is assumed to have up-to-date and valid information about its 1-hop neighbours and surroundings. Neighbours of each B-node can be either D-nodes or U-nodes depending on their locations relative to the coverage hole. Therefore, immediate neighbours of each B-node can be defined as the *undamaged neighbour nodes* (*UN-nodes*) or *damaged neighbour nodes* (*DN-nodes*). Each B-node maintains sets of UN-nodes and DN-nodes. Before the D-event, nodes have fully up-to-date information regarding their neighbours; however, after the D-event, they cannot determine whether their neighbour has become a B-node or still remains as an N-node. It is assumed that each B-node autonomously makes its selection decision based on the available 1-hop information from its neighbours before the D-event.

Since each B-node autonomously perceives the D-event and its damaged neighbours, a *margin* of B-nodes (*MB-nodes*) are formed around the D-area (please refer to Figures 3.1 and 3.2). In order to harness WSNs' redundancy and also reduce the chance of interference and physical collisions, a set of the B-nodes defined

ALGORITHM 3.1: Distance-based Boundary Node Selection Algorithms

Input:

u_k: Undamaged Node (U-Node) k, d_j: Damaged Node (D-Node) j

b_i: Boundary Node (B-Node) i, N_b:Number of $b_i, i \in \{1, \cdots N_b\}$

D_{b_i}: Degree of B-node $b_i, i \in \{1, \cdots N_b\}$

$N_{b_i^d}$: Number of B-node b_i's DN-nodes, $N_{b_i^u}$: Number of B-node b_i's DN-nodes

$\vec{X}_{b_i}^{d_j}$: Distance vector from B-node b_i to its DN-Node d_j

$\vec{X}_{b_i}^{u_k}$: Distance vector from B-node b_i to its UN-Node u_k

$D_{b_i}^{d_j}$: Degree of DN-Node d_j of B-node b_i

$D_{b_i}^{u_k}$: Degree of UN-Node u_k of B-Node b_i

α: Threshold

Output:

Selected B-nodes (SB-node) $b_{s(i)}$

Algorithm *MinD Boundary Node Selection*:

- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{min}} = min(\vec{X}_{b_i}^{d_j}), \quad \forall N_{b_i^d}$
  - **if** $||\vec{X}_{b_i}^{d_{min}}|| \le \alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

Algorithm *MaxD Boundary Node Selection*:

- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{max}} = max(\vec{X}_{b_i}^{d_j}), \quad \forall N_{b_i^d}$
  - **if** $||\vec{X}_{b_i}^{d_{max}}|| \le \alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

Algorithm *VarD Boundary Node Selection*:

- **foreach** $b_i \in N_b$ **do**
  - Calculate $S_{b_i}^{(d_{Var})} = Variance(\vec{X}_{b_i}^{d_j}), \quad \forall N_{b_i^d}$
  - **if** $||S_{b_i}^{(d_{Var})}|| \le \alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

Algorithm *UN-DN Centres of Mass (DUCM) Boundary Node Selection*:

- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{CM}} = \frac{\sum^{j \in N_{b_i^d}} \left(\vec{X}_{b_i}^{d_j} . D_{b_i}^{d_j}\right)}{\sum_{j \in N_{b_i^d}} \left(D_{b_i}^{d_j}\right)}, \quad \vec{X}_{b_i}^{u_{CM}} = \frac{\sum^{k \in N_{b_i^u}} \left(\vec{X}_{b_i}^{u_k} . D_{b_i}^{u_k}\right)}{\sum_{k \in N_{b_i^u}} \left(D_{b_i}^{u_k}\right)}$
  - **if** $||\vec{X}_{b_i}^{d_{CM}}|| \le ||\vec{X}_{b_i}^{u_{CM}}||.\alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

ALGORITHM 3.2: Degree-based Boundary Node Selection Algorithms

Input:
b_i: Boundary Node (B-Node) i
N_b: Number of $b_i, i \in \{1, \cdots N_b\}$
$D_{b_i}^{Bef}$: Degree of B-node b_i Before D-event
$D_{b_i}^{Aft}$: Degree of B-node b_i After D-event
α: Threshold

Output:
Selected B-nodes (SB-nodes) $b_{s(i)}$

Algorithm *Absoluate Degree Loss (ADL) Boundary Node Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $D_{b_i}^{ADL} = D_{b_i}^{Bef} - D_{b_i}^{Aft}$
  - **if** $D_{b_i}^{ADL} \geq \alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

Algorithm *Relative Degree Loss (RDL) Boundary Node Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $D_{b_i}^{RDL} = \left(\frac{D_{b_i}^{Bef} - D_{b_i}^{Aft}}{D_{b_i}^{Bef}}\right)$
  - **if** $D_{b_i}^{RDL} \geq \alpha$ **then**
    - b_i is SB-node
  - **end**
- **end**

End

as *selected B-nodes* (SB-nodes) are considered for possible participation in the recovery process by moving towards the region of interest (ROI). SB-nodes may be selected by a distributed algorithm or centrally based on some agreed upon criteria and conditions. Nodes in the boundary margin may also classified further as *inner nodes* (IN-nodes) and *outer boundary nodes* (OB-nodes) depending on whether they are selected to participate in recovery process or not. Similar to Section 3.2.4 in Appendix A, a group of (distributed) node selection algorithms, based on

B-nodes' 1-hop geometrical and status of statistical features are presented.

3.2.4 Node Selection Algorithms

The proposed distributed BNS-algorithms are classified as either distributed degree-based or distributed distance-based. Centralised algorithms are also proposed to compare with distributed BNS-algorithms.

3.2.4.1 Distributed distance and degree-based BNS-algorithms

In the *Margin of Boundary* (MB) algorithm, each B-node detects the damage and considers itself as a SB-node. So, a band of boundary nodes forms around the D-area. In *Min/Max Distance* (MinD/MaxD) algorithms, if a B-node minimum and/or maximum distances to its DN-nodes are within a given threshold range, the B-node will consider itself as a selected B-node. The number of selected B-nodes depends on the given threshold values. In the *Variance of Distances* (VarD) algorithm, the variance of distances from each B-node to its respective DN-nodes is compared to a given threshold. In the *DN-nodes* and *UN-nodes Centre of Mass (DUCM)* algorithm, each B-node's distance to the centre of mass (CM) of its DN-nodes ($D_{b(i)}$) and UN-nodes ($U_{b(i)}$) are compared to a given threshold. UN-nodes and DN-nodes' degrees can be used as masses in the DUCM algorithm. B-nodes closer to the D-area are expected to lose a higher number of connections with their neighbours and to have more DN-nodes. Thus, the reduction in degree of a given node following the D-event is a logical criterion for B-node selection. In the proposed *Absolute Degree Loss (ADL)* algorithm, each node compares the change in its degree before and after a D-event with a given threshold. The ADL BNS-algorithm does not always result in accurate B-node selection. For example, a node with a degree of two may suffer a reduction to a degree of one or zero after the D-event, but still may not satisfy the threshold. Therefore, ADL is modified to create the *Relative Degree Loss (RDL)* algorithm, in which the reduction in the degree relative to the initial degree of the node is used as the selection criterion.

3.2.4.2 Centralised BNS-algorithms

In the centralised BNS-algorithms, decisions are based on global knowledge of the network and D-event. To compare the proposed distributed BNS-algorithms with centralised schemes, the *Closer Nodes First (CNF)*, *Voronoi-Area (VA)* and *Voronoi-Distance (VD)* algorithms are proposed. In the CNF algorithm, B-nodes with smaller Euclidean distances to the D-area are selected first. In the VA and VD algorithms, B-node selection is based on the features of the nodes' Voronoi diagrams [232]. In the Voronoi diagram, $Cell_{s(i)}$ and $Cell_{s(j)}$ are neighbours if they share at least one vertex ($VCell_{s(i)} \cap VCell_{s(j)} \neq \phi$). B-node cells are classified according to their neighbour cells: if a B-node cell has a D-node neighbour cell, it is a B-D Voronoi cell; otherwise, it is a B-B Voronoi cell. In the VA algorithm, B-D cells with larger areas are selected first. Similarly, the VD algorithm prefers B-D cells closer to the D-area. If the required number of selected B-nodes is larger than the available number of B-D cells, the selection should also be performed amongst the remaining B-B cells. Thus, in the proposed algorithms, previously selected B-node cells are now considered as D-node cells and their B-B cell neighbours are considered as B-D nodes. To reduce complexity, in VA, instead of Voronoi cells' actual geometric area, their in-circle areas are computed.

We change the role of B-D cells' neighbouring B-B cells to B-D cells by using Voronoi cell neighbourhood relations. By applying VD and VA algorithms to the new B-D cells, based on Voronoi proximity, we can perform stage by stage B-node selection and model failure expansion of the D-area (Algorithms 3.3 and 3.4 and Figures of stage-wise B-nodes selection 3.1). Table 3.1 is shown as an example of

ALGORITHM 3.3: VD Boundary Node Selection

Input:
S : Set of sensor nodes $\{s_1, \cdots, s_n\}$
R_c: Node transmission Range
R_s: Node sensing Range
S_U: Set of undamaged nodes (U-nodes), $S_U \subseteq S$
S_D: Set of damaged nodes (D-nodes), $S_D \subseteq S,\ S_D = S \backslash S_U$
S_B: Set of boundary nodes (B-nodes), $S_B \subseteq S,\ S_B \subseteq S_U$
$dist(x, y)$: Euclidean distance from point x to point y
$s_{i(b)}$: B-node i where $\{S_U | dist(s_i, s_j) \leq R_c,\ s_i \in S_U, \exists s_j \in S_D\}$
$Cell_{s_i}$: Voronoi Cell of sensor node i
$vCell_{s_i}$: Set of Voronoi vertices of $Cell_{s_i}$
$Neigh(s_i, s_j)$: $Cell_{s_i}$ and $Cell_{s_j}$ are neighbours if $vCell_{s_i} \cap vCell_{s_j} \neq \emptyset$
$N^{S_B}_{select}$: Number of selected B-nodes
S^{select}_B: Set of selected boundary nodes
$NumNeigh(A, B)$: Number of $Neigh(s_k, s_l)$ where $s_k \in A, s_l \in B$
$AreaCell_{s_i}$: Area of $Cell_{s_i}$
$Dist(s_i, CH_n)$: Distance from sensor s_i to Coverage Hole n (CH_n)
$sortDSC(.)$: Sort Operator (Descending Order)
$sortASC(.)$: Sort Operator (Ascending Order)

Output:
Selected B-nodes (SB-nodes)

Algorithm *VD Boundary Node Selection*:
    $S^{select}_B == \emptyset$
    while $N^{S_B}_{select} > 0$ **do**
        $X = U_D,\ \ Y = U_B$
        if $N^{S_B}_{select} > NumNeigh(A, B)$ **then**
            $S_B = S_B \backslash \{(s_i, s_j) | Neigh(X, Y)\}$
            $S^{select}_B = S^{select}_B \cup \{(s_i, s_j) | Neigh(X, Y)\}$
            $S_D = S^{select}_B$
            $N^{S_B}_{select} = N^{S_B}_{select} - NumNeigh(X, Y)$
        else
            $S^{select}_B = S^{select}_B \cup$
            $\{(s_i, s_j) | Neigh(X, Y), sortASC(Dist(s_i, CH_n)), i = 1, \cdots, N^{S_B}_{select}\}$
            $N^{S_B}_{select} = N^{S_B}_{select} - |S^{select}_B|$
        end
    end
End

ALGORITHM 3.4: VA Boundary Node Selection

Input:
For the Input Parameters please refer to Algorithm (3.3)

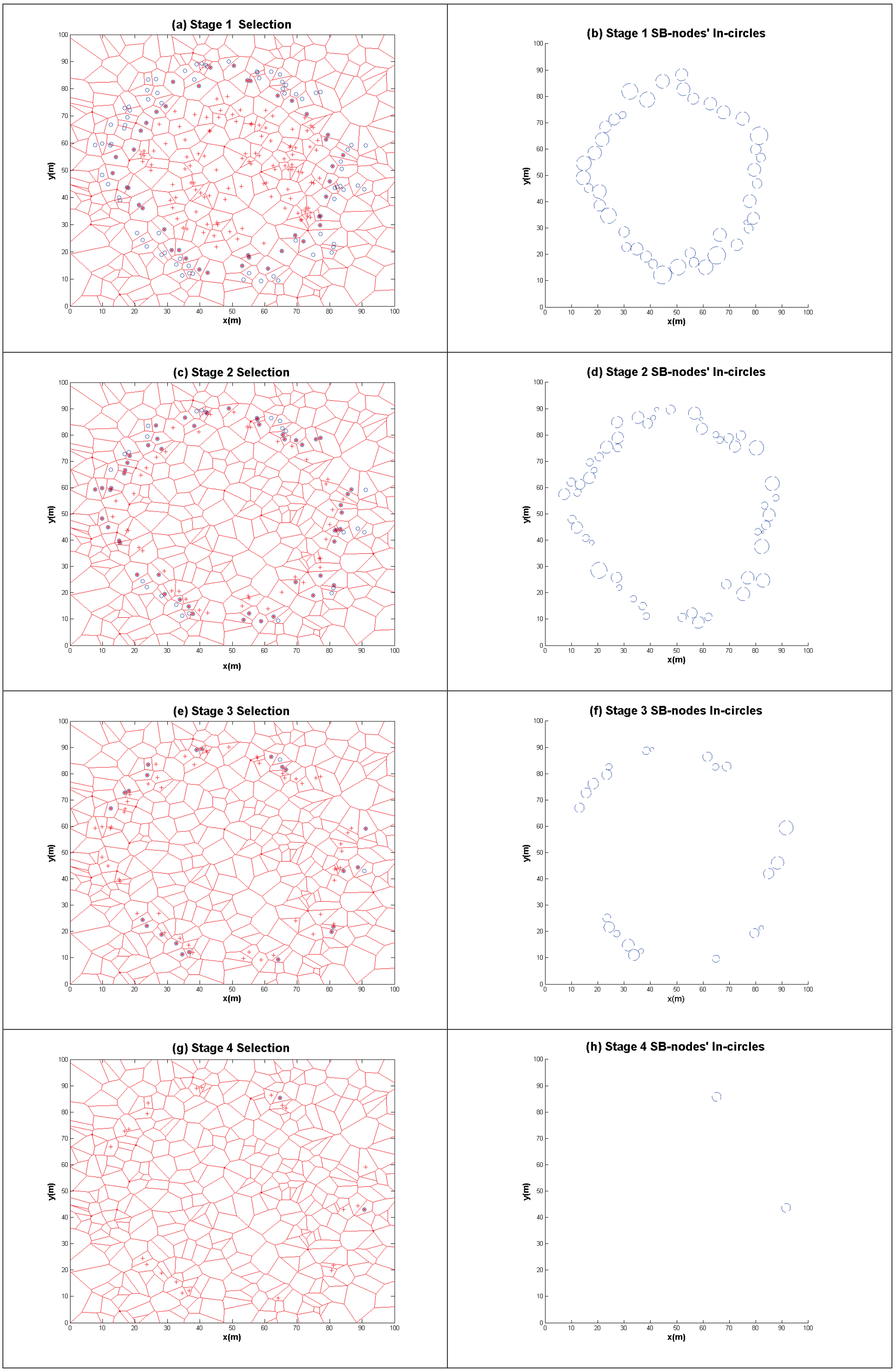
Output:
Selected B-nodes (SB-nodes)

Algorithm *VA Boundary Node Selection*:

$S_B^{select} = \emptyset$
while $N_{select}^{S_B} > 0$ **do**
  $X = U_D, \; Y = U_B$
  if $N_{select}^{S_B} > NumNeigh(A, B)$ **then**
    $S_B = S_B \backslash \{(s_i, s_j) | Neigh(X, Y)\}$
    $S_B^{select} = S_B^{select} \cup \{(s_i, s_j) | Neigh(X, Y)\}$
    $S_D = S_B^{select}$
    $N_{select}^{S_B} = N_{select}^{S_B} - NumNeigh(X, Y)$
  else
    $S_B^{select} = S_B^{select} \cup$
    $\{(s_i, s_j) | Neigh(X, Y), sortASC(Dist(s_i, CH_n)), i = 1, \cdots, N_{select}^{S_B}\}$
    $N_{select}^{S_B} = N_{select}^{S_B} - |S_B^{select}|$
  end
end
End

this stage-wise selection. In Table 3.1, D-nodes, B-nodes and SB-nodes are denoted '+', 'o' and '*' respectively in the left column (a, c, e, g); the right column (b, d, f, h) shows the in-circle of cells in each stage of selection.

Table 3.1. Stage-wise B-node Selection



3.3 Performance Evaluation

3.3.1 Performance Metrics

3.3.1.1 Average distance to Damaged Area

SB-nodes that are closer to a D-area can obtain more accurate measurements of a D-event. Hence, if B-node S_{b_i} is located at (x_i, y_i), the coverage hole with a radius of r_{hole} is located at (x_{hole}, y_{hole}) and $N_{b(i)}$ is the number of selected B-nodes. The *average distance from the damaged area* (AVD) of SB-nodes is defined as :

$$Avg_{Dst} = \frac{1}{N_{b(i)}} \left\{ \sum_{i \in b} [(x_i - x_{hole})^2 + (y_i - y_{hole})^2]^{1/2} - r_{hole} \right\}, \qquad (3.1)$$

Regarding the model of coverage hole, $C_{r_{hole}}(x_{hole}, y_{hole})$, the AVD of SB-nodes ($Avg_{Dst}$) is the average distance of selected B-nodes from the perimeter of coverage hole; Equation 3.1 obtains the average over euclidean distances from SB-nodes to the hole's perimeter.

3.3.1.2 Percentage of Coverage

The deployed 2-D rectangular area ($[x_{min}, x_{max}] \times [y_{min}, y_{max}]$) is discretised into grid cells of z_i=(x_i, y_i). If the coordinates of the grid cell's corner are within a node's range, it is covered by the sensor nodes. Grid cells that are covered simultaneously by k sensors nodes, are defined as grid cells having *k-Coverage* in the area of deployment. The *percentage of coverage* (PCov) is defined as the number of grid cells that are covered by at least one of the sensor nodes (1-covered) over the total number of grid cells in the given deployed area.

Here, as the D-event and BNS-algorithms only affect the rectangular area of $[x_{hole} - r_{hole}, x_{hole} + r_{hole}] \times [y_{hole} - r_{hole}, y_{hole} + r_{hole}]$, PCov is defined as the ratio of covered cells by SB-nodes in the given area to total number of cells in the given area.

3.3.2 Results

Using Matlab, 400 nodes were randomly deployed with the uniform distribution in a 100 $m \times 100$ m region. For the sake of simplicity, all nodes' communication and sensing ranges were considered equal ($R_c = R_s = 15$ m). Sensor nodes may be mobile or static and their locations and 1-hop neighbours' distances are known through GPS or other localisation methods [275, 349, 350]. Figure 3.1 depicts D-area as a circle with radius $r_{hole} = 30$ m located at $(x_{hole}, y_{hole}) = (50, 50)$. The experiment was repeated 100 times with uniform random node distribution for each algorithm.

Since the number of B-nodes varies in each experiment, on the average, the maximum number of selections is the minimum number of B-nodes in 100 samples ($N_b^{exp\{1,\cdots,100\}}$=$min\{N_b^{exp\{1\}}, N_b^{exp\{2\}}, \cdots, N_b^{exp\{100\}}\}$ where $N_b^{exp\{i\}}$ is number of B-nodes in the sample experiment i). The required number of selected B-nodes (SB-nodes) was varied from 1 to $N_b^{exp\{1,\cdots,100\}}$ to observe the AVD and PCov behaviour of different algorithms as the number of SB-nodes increases. As shown in Table 3.3, as the number of selected B-nodes approaches the total number of B-nodes $N_b^{exp\{1,\cdots,100\}}$, AVD and PCov of all BNS-algorithms converge. The algorithms' performances were compared as follows: **1)** Degree-based BNS-algorithms vs. MB and CNF; **2)** Distance-based BNS-algorithms vs. MB and CNF; **3)** Centralized BNS-algorithms vs. MB and CNF; **4)** Degree-based vs. Distance based BNS-algorithms; and **5)** AVD and PCov of Selected Degree-based, Distance-based, and Centralized BNS-algorithms.

Results were obtained with 97.5% confidence intervals. Bounds for BNS-algorithms'

Alg.	Alg. Type	AVD	PCov
RDL	Degree-based	70%-10%	12%-1%
ADL	Degree-based	65%-4%	5%-0%
MaxD	Distance-based	0%-4%	-5%-0%
MinD	Distance-based	71%-10%	11%-0%
DUCM	Distance-based	69%-7%	16 %-1%
VarD	Distance-Based	62%-5%	9%-0%
VA	Centralised	42%-7%	13%-1%
VD	Centralised	45%-8%	12%-1%
CNF	Centralised	78%-13%	30%-4%

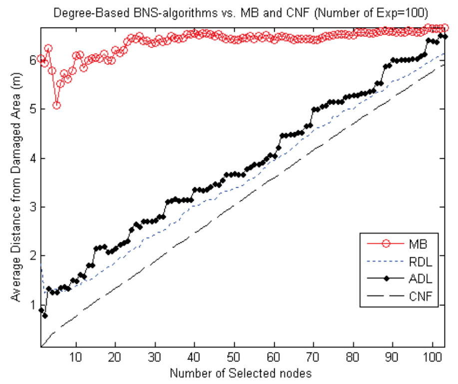
Table 3.2. BNS-Algorithms percentage of AVD,PCov improvement over MB (Number of SB-nodes 20-100)

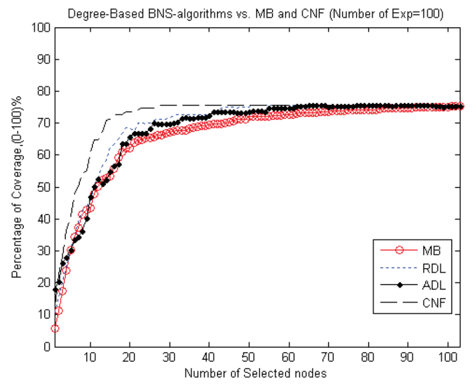
percentage of AVD and PCov improvements over MB selection are shown in Table 3.2 as the number of selected B-nodes increased from 20 to 100. As shown in Table 3.2, all BNS-algorithms show improvement over MB, except for MaxD. The efficacy of BNS-algorithms is higher when fewer B-nodes are selected, although the performance fluctuates significantly. Table 3.2 shows that Degree-based and Distance-based BNS-algorithms perform similarly in terms of both AVD and PCov. However, each of the BNS-algorithms is useful in different circumstances. In the case of unreliable localisation and inaccurate distance measurements, the degree-based BNS-algorithms still allow good performance to be achieved at the cost of somewhat poorer coverage. In general, distributed BNS-algorithms have good performance in comparison to centralised algorithms both in terms of AVD and PCov (Tables 3.2 and 3.3).

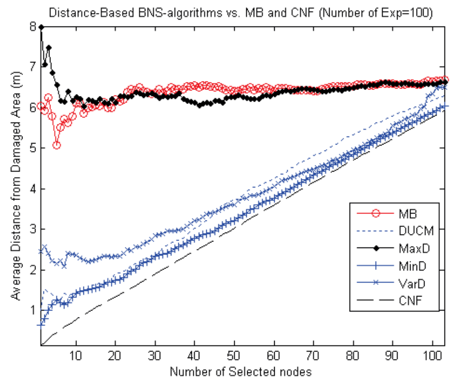
From Table 3.2, as selected B-nodes increase from 20 to 100, it can be seen that,

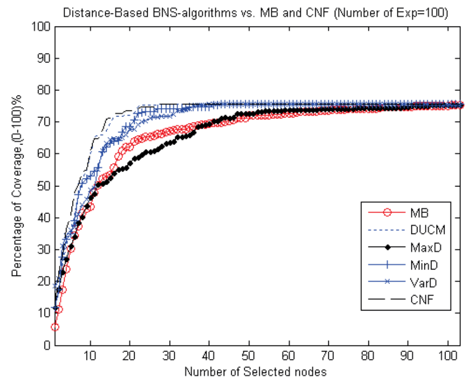
In degree-based BNS-algorithms, RDL outperforms ADL both in AVD and PCov respectively on average by 5.5% and 4%. Thus, B-nodes should consider the relative rather absolute losses of their degrees of connectivity in self-selection process.

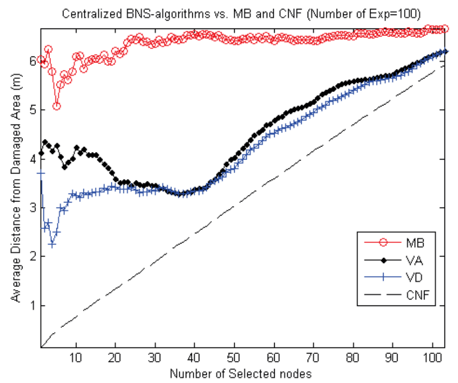
Table 3.3. AVD and PCov of BNS-algorithms

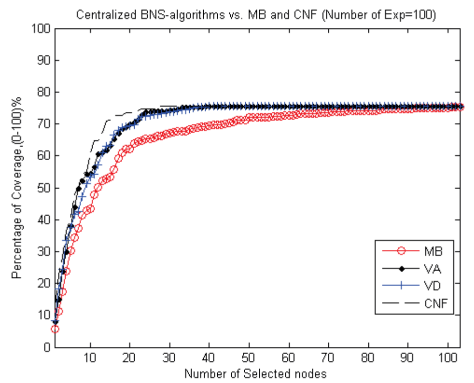


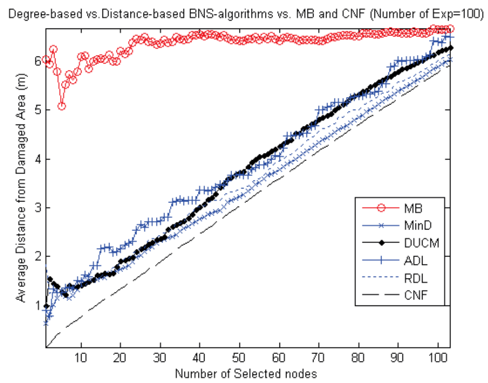


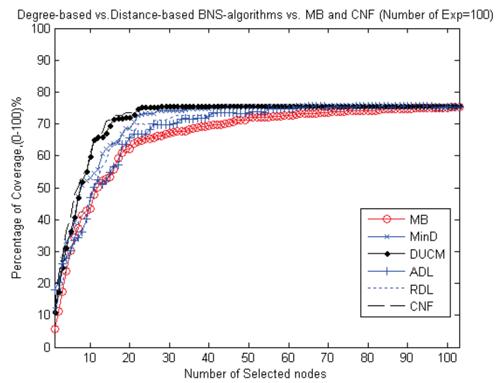












In distance-based BNS-algorithms, MinD algorithm outperforms other algorithms on average by 16% in AVD. DUCM outperforms other selection algorithms on average by 6% in PCov. By excluding MaxD algorithm, MinD outperforms DUCM and VarD algorithms on average by 4.75% in AVD. Similarly, by excluding MaxD, DUCM outperforms MinD and VarD on average by 3.5% in PCov. MaxD algorithm can be considered for the B-nodes in the case of coverage hole avoidance and isolation. By considering the DUCM's PCov, BS-nodes will likely spread more uniformly around the coverage hole and from the margin of boundary nodes.

Regarding degree-based and distance-based algorithms, RDL matches MinD in both AVD and PCov. RDL and DUCM in AVD perform similarly, while DUCM outperforms RDL on average by 2%. Either of these algorithms can be used based on the available information that B-nodes have from their neighbouring nodes after the D-event.

Regarding degree-based and distance-based algorithms with optimal selection algorithm (CNF), RDL is on average within 5.5% and 10.5% of the CNF respectively in AVD and PCov. MinD is on overage within 5% and 11.5% of CNF in AVD and PCov respectively. DUCM algorithm is on average within 7.5% and 8.5% of CNF algorithm in AVD and PCov respectively.

3.4 Conclusion

Distributed algorithms for automatic determination of the boundary of a damaged region of a network based on available 1-hop knowledge inferred from a node statistical and geometrical features are presented. The efficiency of the proposed distributed BNS-algorithms is compared with their centralised counterparts.

CHAPTER 4

Coverage Hole Partial Recovery by Nodes' Constrained and Autonomous Movements Using Virtual α-chords

In this chapter, an emergent cooperative recovery is presented to address coverage holes in real-time for networks, which are deployed in harsh and hostile environments with minimum or no centralised authority. In this recovery model, a global behaviour within the network emerges from the mere local and limited interactions among nodes and their immediate neighbours. Nodes autonomously decide to participate in the recovery process as they detect the coverage hole within their ranges. Nodes' autonomous decisions are based on their local perceptions from the surroundings and their neighbouring nodes. These decisions reduce the network's overall exchanged information especially in cases which communication amongst nodes is either not allowed or kept at certain levels for security. In this recovery model, a set of nodes proximate to coverage holes, known as *boundary nodes*, autonomously decide on the direction and magnitude of their movements towards coverage holes, as node relocation is an effective solution to mitigate the direct and indirect effects of node failures. In order to consider the effect of nodes' physical movements on the their energy consumptions and avoid cascaded node failures, the relocation of boundary nodes is locally controlled via virtual chords, defined as α-chords, that nodes form with their neighbouring nodes. Using the virtual chords in node relocation not only constrains the movement of autonomous boundary nodes, but also results in an emergent cooperative behaviour from the local interactions of boundary nodes in the coverage hole recovery model. The proposed coverage hole recovery model and its performance are presented in this chapter.

4.1 Method and Assumptions

Scenario and deployed sensor nodes, the coverage hole model, and the proposed algorithm are given as follows:

4.1.1 Sensor Node and Coverage Hole Model

The sensor node model and coverage hole described respectively in Sections 3.2.1.1 and 3.2.2. Coordinates of nodes $s_i,\ i = \{1, ..., N\}$ are (x_i, y_i) which are the centres of the circles of the radii equal to the nodes' ranges. The coverage holes are modelled as a circle of radius r_{hole}, with the centre at (x_{hole}, y_{hole}) (Figures 3.1, 4.1, 3.2 and 4.3).

4.1.2 Nodes Classification

Regarding their immediate neighbours and their status to the coverage hole (D-area), nodes can be classified as *damaged nodes* (*D-nodes*), *undamaged nodes* (*U-nodes*) and *boundary nodes* (*B-nodes*), as noted in Section 3.2.3. As each B-node autonomously perceives the D-event and its damaged neighbours, a margin of B-nodes are formed around the D-area (Section 3.2.3 and Figures 4.1 and 4.3). Depending on the nodes' locations relative to the coverage hole, neighbours of each B-node are defined in Section 3.2.3 as the *undamaged neighbour nodes* (*UN-*

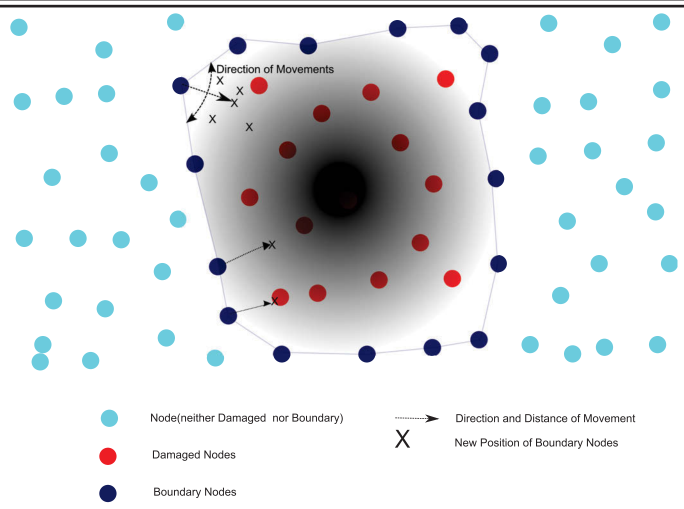


Figure 4.1: Coverage Hole and Node Types

nodes) or *damaged neighbour nodes* (*DN-nodes*). To benefit WSNs' redundancy and reduce the chance of interference and physical collisions, a set of the B-nodes defined as *selected B-nodes* (*SB-nodes*) are considered for possible participation in the recovery process via boundary selection algorithms similar to Chapter 3.

4.1.3 Local Communications Protocol

Each possible moving node's decision is based on its limited knowledge of its 1-hop neighbours. This information includes the status of nodes' 1-hop neighbours before the damage event and the perceived 1-hop neighbours' status change after damage event without any additional message exchanges.

4.1.3.1 Nodes' Status

At the time of the D-event, each B-nodes' distances to both sets of their UN-nodes and DN-nodes, as well as their degrees of connectivity, are used as landmarks in the decision-making processes. Therefore, if a B-node selects a set of its UN-nodes via some selection algorithms, those UN-nodes are considered to be selected

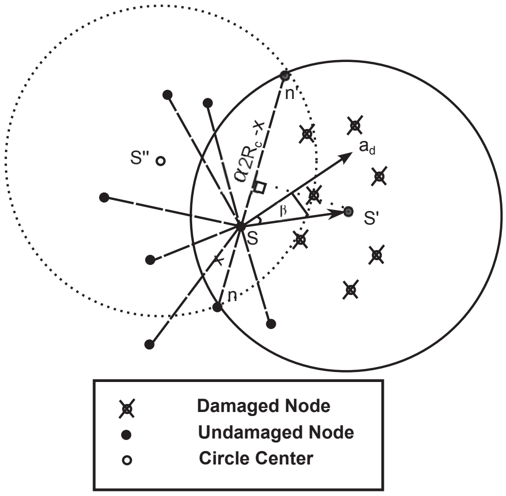


Figure 4.2: B-Node, its real, virtual neighbors and virtual chord

undamaged neighbour nodes (real neighbours).

4.1.3.2 Virtual Neighbours and α-chord

Virtual selected undamaged neighbour nodes are the fictitious B-nodes' neighbours (virtual neighbours) that are connected to a B-node's UN-nodes via line segments. Each line segment $\overline{nn'}$ connects fictitious point n' (i.e. a node's virtual neighbour) to a real point n (i.e. the node's real neighbour) while passing through point s (i.e. the B-node's location) (Figure 4.2). Line segments are considered to be *virtual chords* for pairs of circles where each virtual chord's endpoints (i.e., the node's real and virtual neighbours) lie on the circumference of the two distinct circles with equal radius of R_c. Virtual chords are defined as α-chord with the length of $\alpha \cdot (2 \cdot R_c) \leq 2 \cdot R_c$ where $0 \leq \alpha \leq 1$ (Figure 4.2). One of these circles is considered a *valid circle*, if it is closer to the damaged area. With α-virtual chord node movement, a relocated B-node from s to s' maintains the connectivity with its real and virtual neighbours n and n', provided its real neighbour n is not a moving B-node.

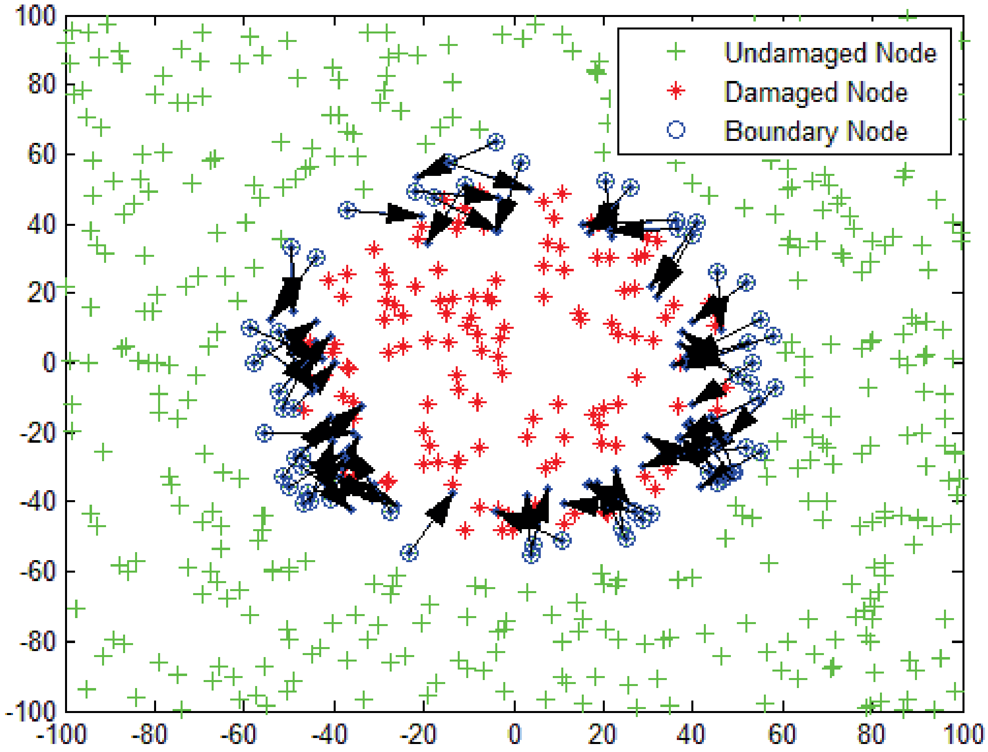
4.1.4 Selection Algorithms

In this section three neighbour node selection algorithms —*closest neighbour*, *random neighbour*, and *β-angle* — are presented (see Algorithm 4.1). Using these algorithms, each B-node selects a node from its neighbours as one of its virtual chord endpoints (Figure 4.2 and Section 4.1.3.2).

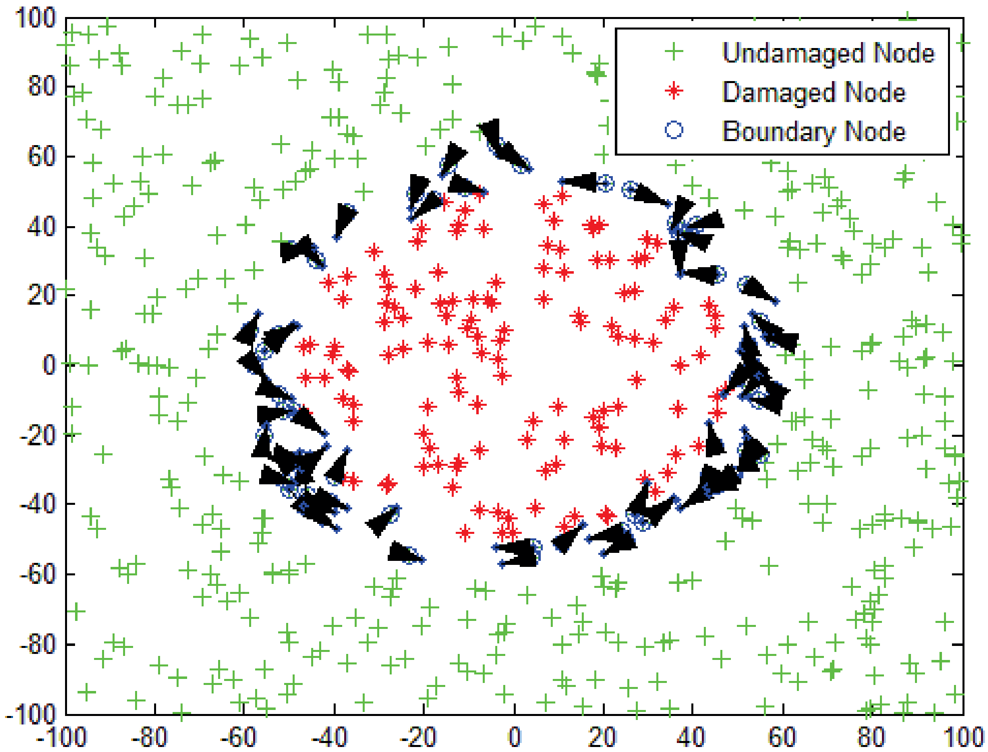
In the closest neighbour algorithm, each B-node selects its closest 1-hop neighbour. In the random select algorithm, each B-node randomly selects one of its 1-hop neighbours. In the β-angle algorithm, for each B-node and the undamaged node in its neighbour set, a set of angles is formed between the normal direction of the virtual chords and distance vector from the B-node to its D-nodes centre of mass (Figure 4.2). The B-node's neighbour whose aforementioned angle is closer to β than any other of B-nodes' 1-hop undamaged neighbours is selected undamaged neighbour and should be unique. If more than one undamaged neighbour can be selected based on these conditions, only one of them is randomly chosen. In finding the centre of mass of the B-nodes' undamaged and damaged neighbours, if the neighbours' degrees of connectivity are taken into account (Algorithm 4.1) they can be considered as weighted, β-angle algorithms with $w = 1$; otherwise they are β-angle with $w = 0$.

4.1.5 Movement Algorithms

In the proposed movement algorithms, each B-node obtains its virtual chord by selecting its real and virtual undamaged neighbour node (Sections 4.1.3.2 and 4.1.4, Algorithm 4.1, and Figure 4.2). The B-node's new location is found from two circles that share the B-node's virtual chord (see Algorithm 4.2 and Figure 4.2). Then, the B-node moves to the new location with probability of p. Movement algorithms can be divided into following states:



(a) $\alpha = 0$



(b) $\alpha = 1$

Figure 4.3: Chord movement algorithm (R_c=15, N=600, β=0)

ALGORITHM 4.1: Nodes' neighbors selection Algorithms

Input:
s_i^b : B-node i $(i = 1, \cdots, m)$, $N_{s_i}^h$: s_i^b's h-hop neighbours
$N_{s_i}^{h_u}$: h-hop U-node neighbours of s_i^b
$N_{s_i}^{h_d}$: h-hop D-node neighbours of s_i^b
$\vec{X}_{s_i}^{s_j^{h_u}}$ distance vector from s_i to s_j (j in $N_{s_i}^{h_u}$)
$\vec{X}_{s_i}^{s_j^{h_d}}$ distance vector from s_i to s_j (j in $N_{s_i}^{h_d}$)
$\beta - angle$: angle parameter
$d_{S_j}^{h_u}$ degree of s_j (j from $N_{s_i}^{h_d}$)
$d_{S_j}^{h_d}$ degree of s_j (j from $N_{s_i}^{h_u}$)

Output: Set of selected h-hop neighbour $s_j^{h_u}$ of s_i^b

case *Closest* if closest neighbour selected
  foreach *B-Node* s_i^b **do**
    Find $N_{s_i}^{h_u}$
    foreach *h-hop UN-nodes* $s_j^{h_u}$ **do**
      Calculate $\vec{X}_{s_i}^{s_j^{h_u}}$
    Calculate $arg_j Min \left(\left| \vec{X}_{s_i}^{s_j^{h_u}} \right| \right)$

case *Random* if Random neighbor Selected
  foreach *B-Node* s_i^b **do**
    Find $N_{s_i}^{h_u}$
    Calculate $arg_j Random \left(N_{s_i}^{h_u} \right)$

case *β-angle* if β-angle is Selected
  foreach *B-Nodes*$_i^b$ **do**
    Find $N_{s_i}^{h_u}$ and $N_{s_i}^{h_d}$
    foreach *h-hop(DN-node* $s_j^{h_d}$, *UN-node* $s_j^{h_u}$) **do**
      Find $d_{S_j}^{h_d}$, $d_{S_j}^{h_u}$
      Calculate $\vec{X}_{s_i}^{s_j^{h_d}}$,$\vec{X}_{s_i}^{s_j^{h_u}}$
    Calculate $\vec{X}_{CM\,si}^{\;h_d} = \frac{\sum (\vec{X}_{s_i}^{s_j^{h_d}}) \cdot d_{S_j}^{h_d}}{\sum d_{S_j}^{h_d}}$
    Calculate $\vec{X}_{CM\,s_i}^{\;h_u} = \frac{\sum (\vec{X}_{s_i}^{s_j^{h_u}}) \cdot d_{S_j}^{h_u}}{\sum d_{S_j}^{h_u}}$
    foreach *h-hop UN-node* $s_j^{h_u}$ **do**
      Calculate $\angle \gamma_{s_i}^{h_u s_j} = \angle \left(\vec{X}_{CM\,si}^{\;h_d}, \vec{X}_{s_i}^{s_j^{h_u}} \right) - \angle \beta$
    Calculate $arg_j Min \left(\left| cos(\angle \gamma_{s_i}^{h_u s_j}) \right| \right)$

ALGORITHM 4.2: Formation of Chord Algorithm

Input:
s_i^b : B-node i $(i = 1, \cdots, m)$, τ : threshold
$s_j^{h_{u_{s_i^b}}}$: Selected h-hop U-node neighbour s_j
α-chord parameter, R_c transmission Range
$N_{s_i}^{h_u}$: h-hop U-node neighbours of s_i^b
$N_{s_i}^{h_d}$: h-hop D-node neighbours of s_i^b

Output:
B-nodes s_i^b's new location (coordinates), ${s'}_i^b(x, y)$

foreach *B-node* s_i^b **do**
  Find s_i^b's current location (coordinates) of $s_i^b(x, y)$
  Find $CM_i^{h_{u(x,y)}}$: s_i^b's h-hop UN-nodes' center of mass
  Find $CM_i^{h_{d(x,y)}}$: s_i^b's h-hop DN-nodes' center of mass
  Calculate $chord - \alpha_i$, virtual node ${s'}_j^{h_{u_{s_i^b}}}$ from α_i and R_c
  Find $C_{\alpha_i}^{(x,y)_{k,k'}}$ (circle center(s)) of $chord_{\alpha_i}$
  foreach $chord_{\alpha_i}$ *and* $C_{\alpha_i}^{(x,y)_{k,k'}}$ **do**
    if $\left\|C_{\alpha_i}^{(x,y)_k} - CM_i^{h_{d(x,y)}}\right\| < \left\|C_{\alpha_i}^{(x,y)_{k'}} - CM_i^{h_{d(x,y)}}\right\|$ **then**
    $C_{Valid_{\alpha_i}}^{(x,y)} = C_{\alpha_i}^{(x,y)_k}$
    else if $\left\|C_{\alpha_i}^{(x,y)_k} - CM_i^{h_{d(x,y)}}\right\| > \left\|C_{\alpha_i}^{(x,y)_{k'}} - CM_i^{h_{d(x,y)}}\right\|$ **then**
    $C_{Valid_{\alpha_i}}^{(x,y)} = C_{\alpha_i}^{(x,y)_{k'}}$
    else if $\left\|C_{\alpha_i}^{(x,y)_k} - CM_i^{h_{d(x,y)}}\right\| = \left\|C_{\alpha_i}^{(x,y)_{k'}} - CM_i^{h_{d(x,y)}}\right\|$ **then**
    Calculate rand $p \sim U[0, 1]$
    if $p > \tau$ **then**
      $C_{Valid_{\alpha_i}}^{(x,y)} = C_{\alpha_i}^{(x,y)_k}$
    else
      $p < \tau$
    $C_{Valid_{\alpha_i}}^{(x,y)} = C_{\alpha_i}^{(x,y)_{k'}}$
  ${s'}_i^b(x, y) = C_{Valid_{\alpha_i}}^{(x,y)}$

- Undamaged neighbour nodes of moving B-nodes are selected based on criteria presented in Algorithm 4.1;
- With regard to the suitable virtual chord parameters α and R_c, the location of B-nodes' virtual neighbours are obtained for each B-node;
- The moving B-nodes' new locations are computed by selecting one of two circles that pass through the endpoints of the virtual chord of each B-node (Algorithm 4.2). The selected circle, which has its centre closer to the damaged area, is defined as the *valid circle* through which the chord is obtained; and
- The B-node then moves to the centre of the valid circle with probability p (uniform distribution) and q otherwise, such that $p + q = 1$. Here it is assumed $p = 1$ in which all B-nodes move towards the coverage hole.

It should be noted that connectivity of B-nodes to their neighbours can not be fully guaranteed because, after the damage event, B-nodes are not able to distinguish if their undamaged neighbours are moving B-nodes or not. For example, Figure 4.3 shows how changing parameter α affects B-nodes' collective movement behaviour in the coverage hole recovery model. β-angle with $\alpha = 0,\ 1$ in Figure 4.3 shows the direction of moving B-nodes towards/around the coverage hole.

4.2 Performance Evaluation

In order to compare the proposed movement model with the two Voronoi-based movement algorithms (VOR and MinMax) [231], different types of performance metrics have been defined in this section. Voronoi-based algorithms are considered as the benchmark and B-nodes were selected similarly to previous chapters.

4.2.1 Performance Metrics

In modelling Voronoi movement algorithms, the problem of nodes with out-of-area and infinite Voronoi vertices are taken into account. The proposed performance metrics in this section are classified as:

- ***Coverage-based metrics***: *Percentage of recovery* is defined as the percentage of recovered networks' coverage after the recovery process. In other words, by using the given movement algorithm, the metric shows what percentage of lost coverage is recovered in the network (see Section 3.3.1.2 for definition of coverage).
- ***Connectivity-based metrics***: *Percentage of connectivity* is defined as the percentage of moving B-nodes that are directly connected to rest of network (those nodes that did not participate in the recovery process) with at least one link over the total number of moving B-nodes. This performance metric shows the effect of movement algorithms on the connectivity of moving nodes and the number of moving B-nodes still directly connected to the rest of network after their movements.
- ***Distance-based metrics***: *Average movement* is defined as the ratio of the total amount of movement to the number of participating nodes in recovery process. Average movement can be used with other metrics to better understand the behaviour of movement algorithms in the coverage hole recovery process.

4.2.2 Results

Using Matlab, $N = 1000$ nodes with communication and sensing range of 15 ($R_c = R_s = 15$ m) were uniformly deployed with random distribution in a

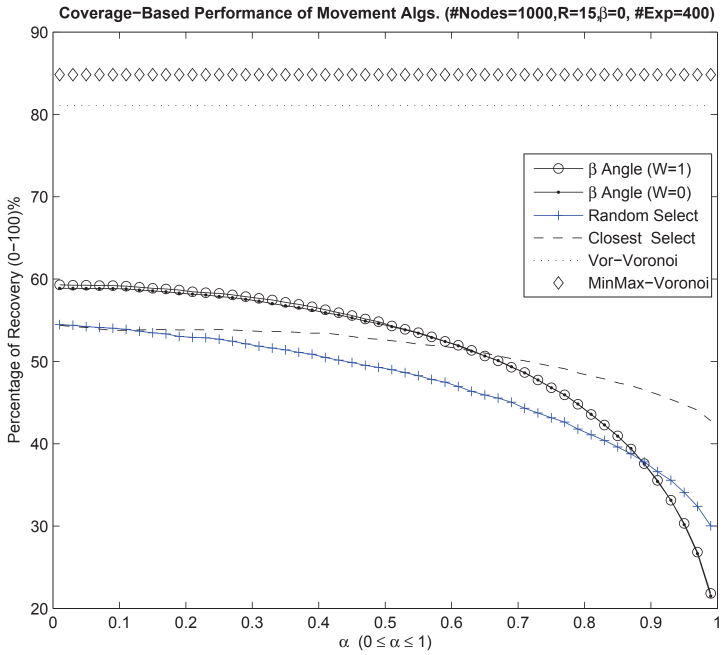


Figure 4.4: Percentage of Recovery

rectangular area of $[-100, 100] \times [-100, 100]$. Similar to the preceding chapter, coverage holes were modelled as circles with a radius $r_{Hole} = 50$ m located at $(x_{Hole}, y_{Hole}) = (0, 0)$. The experiment was repeated $\#Exp = 400$ times for the movement algorithms. Chord parameter (α) was continuously changed from 0 to 1 to examine its effect on the performance and the nodes' collective behaviour with the proposed movement algorithms. Results have confidence intervals 97.5% (Figures 4.4, 4.5, and 4.6). The movement algorithms' performances are also shown in Table 4.1. With regard to Figures 4.4, 4.5, and 4.6, as α changed from 0 to 1, the percentage of recovery and nodes' average movements of virtual chord movement algorithms decreased but at the same time their percentages of connectivity increased, which shows that the B-nodes' collective behaviour and direction of movements shift gradually from moving towards to circulating around coverage hole. Each of these collective mobility behaviours can be used for different

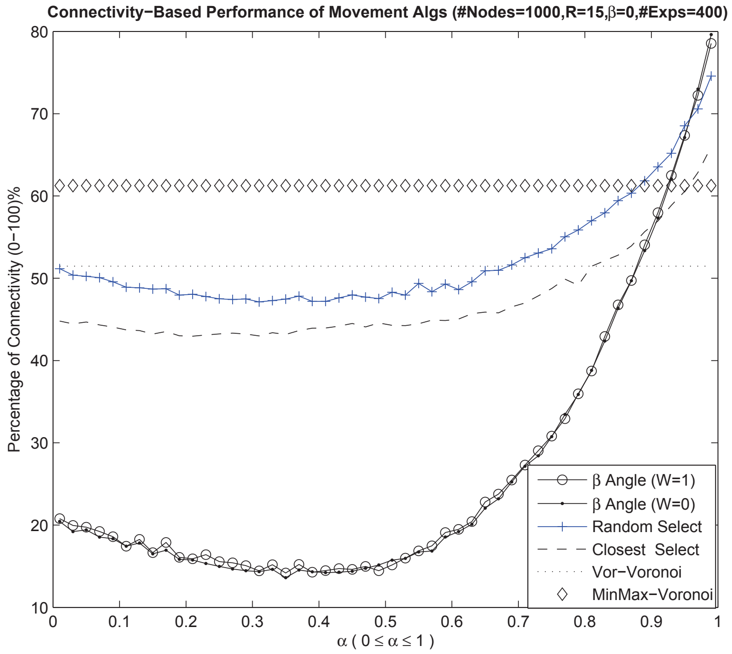


Figure 4.5: Percentage of Connectivity

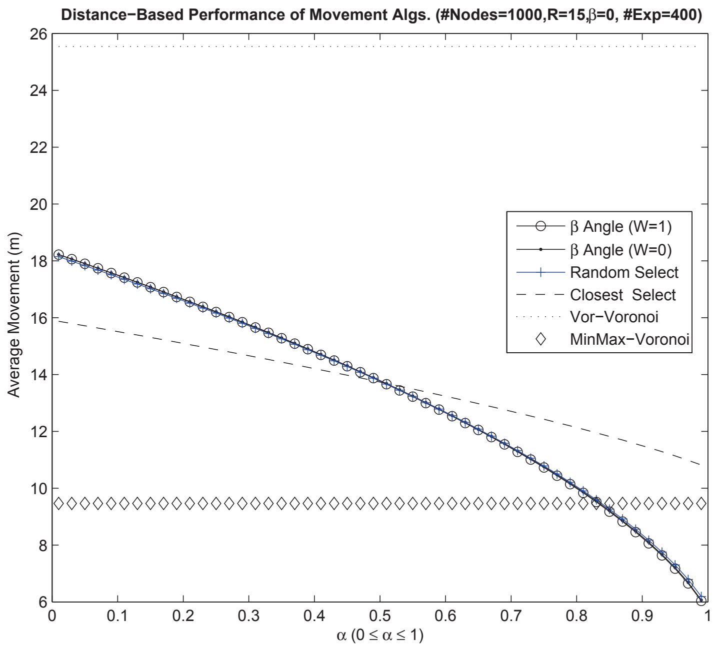


Figure 4.6: Average Movement

Algs.	α	Recovery(%)	Connectivity(%)	Avg. Mov.(m)
β-*angle* ($w = 1$)	0.00	59.3000	20.5821	18.2413
	0.25	58.1561	14.9848	16.1297
	0.50	54.3100	15.7554	13.6721
	0.75	46.3983	32.1165	10.5888
	1.00	21.8333	84.2525	5.6323
β-*angle*($w = 0$)	0.00	58.8752	20.4436	18.2227
	0.25	57.8314	14.4888	16.1230
	0.50	54.2239	15.7458	13.6797
	0.75	46.4474	31.7605	10.6149
	1.00	21.5037	84.0388	5.6850
Closest	0.00	54.3857	43.9619	15.7489
	0.25	53.8395	42.9475	14.7811
	0.50	52.5149	43.9536	13.7172
	0.75	49.2130	48.7582	12.5073
	1.00	42.8042	67.2597	11.0456
Random	0.00	54.4647	52.0741	18.1173
	0.25	52.5196	49.3625	16.0505
	0.50	49.0044	49.2323	13.6488
	0.75	42.9059	55.0499	10.6438
	1.00	30.0398	77.9009	5.8566
Vor	0.00	81.0696	51.3128	25.5432
	0.25	81.0696	51.3128	25.5432
	0.50	81.0696	51.3128	25.5432
	0.75	81.0696	51.3128	25.5432
	1.00	81.0696	51.3128	25.5432
MinMax	0.00	84.8375	61.5663	9.4616
	0.25	84.8375	61.5663	9.4616
	0.50	84.8375	61.5663	9.4616
	0.75	84.8375	61.5663	9.4616
	1.00	84.8375	61.5663	9.4616

Table 4.1. Performances of Movement Algorithms

purposes. In our model, α can be chosen in such a way as to achieve proper percentage of connectivity, percentage of recovery with given amount of nodes' movement. Results from Figures 4.4, 4.5, and 4.6 and Table 4.1, show that although proposed chord movement algorithm is autonomous and requires little or no message exchange, its performance is comparable to Voronoi-based movements. In the real-time scenarios with security and interference concerns, using Voronoi-based algorithms requires global knowledge of the network. So even if higher coverage and connectivity is offered, in these scenarios, they are not practical.

Table 4.1, Figures 4.4, 4.5, 4.6, and 4.3 show that it is possible to have different network recovery behaviours (i.e. from repairing to avoiding coverage holes) based on type of coverage holes in terms of desired coverage, connectivity and nodes' average movements via setting parameter of α in nodes' virtual chords. Characteristics of the proposed model presented in this chapter can be listed here.

- *Light Information Exchange.* Each node exchanges information with its immediate in-range neighbour, which reduces the chance of flooding among the nodes in network.
- *Security.* As nodes act autonomously based on their available, limited and local information (available before occurrence of event), no further information are exchanged among nodes during the recovery process which reduces the possibility of inter-node communications being compromised, intercepted, or interrupted by adversaries.
- *Distributed Recovery.* An autonomous and constrained node movement model based on a node's 1-hop perception provides a rapid recovery mechanism for large-scale coverage holes in real-time and harsh environments. Comparable to its centralised benchmarks, it can be a scalable recovery model in environments lacking appropriate centralised supervision.
- *Connectivity and Movement Control.* Moving nodes' connectivity to the rest of the network to some extent can be maintained via introducing the geometrical concept of α-chords, as nodes move with respect to their neighbouring nodes. By constraining the nodes' movements, their energy can significantly be preserved.
- *Agility and Flexibility.* Although nodes participating in the recovery process act based on their local visions about their surroundings, a movement model based on virtual chords can result in an emergent cooperative behaviour within the network [49] (Figure 4.3). Due to the nodes' local and

autonomous decision making, the recovery process is swift. This speed is desirable for real-time and time-sensitive applications. By setting parameters such as α in a virtual chord movement algorithm, the network is able to behave differently to either repair or avoid coverage holes [277] (Table 4.1 and Figure 4.3). This proposed recovery model is suitable for the applications in which damage sprawls further to proximate nodes and in which an alternative post-action plan for facing such aggressive damage is required. Therefore, this model can flexibly modify the nodes' movements, moving them towards coverage holes to circulating them around holes. The former is suitable for hole recovery. The latter can prevent cascaded node failures and failure expansion around damaged area: the circular node movements can allow moving nodes of higher energies to either enforce or replace nodes with lower residual energies.

4.3 Conclusion

A new autonomous and constrained node movement model is proposed to partially/wholly recover large-scale coverage holes in real-time scenarios with interference and security considerations. In the this chapter, the proposed model of autonomous decision making is based on the available 1-hop knowledge at the time of the damage event. By introducing α-chords, the proposed model also considered the connectivity of moving nodes. When compared with conventional Voronoi-based algorithms, the proposed model provided suitable performance metrics.

CHAPTER 5

Fuzzy Node Relocation Models

5.1 Introduction

The distributed (local) relocation of nodes could address the widespread damages and unbalanced node deployments (i.e. nodes' failure and drifts), as manually adding new sensor nodes in the harsh and hostile environments in the absence of a centralised supervisor is challenging in WSNs. Coping with the unpredictable network topology changes and reducing the burden of centralised relocation paradigms by the distributed movement models comes at the price of oscillations and excessive movements due to nodes' local and limited interactions. If the node movements in the distributed relocation models are not properly addressed, their power will be exhausted. Inspired by the laws of nature and conforming to the uncertain nature and local interactions of nodes, we have proposed the (distributed) force-based fuzzy node movement algorithms (i.e. virtual push-pull forces among the nodes) to steer nodes towards their new locations based on the aggregation of properly exerted virtual forces on the nodes from their neighbours, which may alleviate undesirable oscillations. In this chapter, the proposed fuzzy movement algorithms' parameters are set by using the (human) *expert knowledge* (derived from domain experts based on their experience [351]), and a one-time and iterative *particle swarm optimization* (PSO) with respect to status of nodes and their neighbour nodes. The performance of the proposed models are compared with one another in addition to the *distributed self-spreading algorithm* (DSSA). The results show that the proposed fuzzy movement algorithms either outperform or matches DSSA in one or more performance metric(s).

5.2 Method and Assumption

5.2.1 Nodes and Deployment Area

Nodes and area of deployment are modelled according to Section 3.2.1.1.

5.2.2 Boundary Conditions

In relocation algorithms, the behaviour of moving nodes that have reached the deployment area boundaries (i.e. $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$) with respect to different boundary conditions should be taken into account. Node boundary strategies defined in [352] are adopted as below:

- (B_1): In *Non-Stop At Boundary*, nodes move to their new locations regardless of whether or not their new computed locations are beyond the boundaries of a given area. Therefore, the nodes' movements are not limited by area boundaries. Nodes relocate towards their new locations without limit.
- (B_2): In *Stop At Boundary*, those nodes whose new locations are beyond the boundaries of given area stop at the boundaries as soon as they reach them. Thus, movement of nodes is limited by the given area's boundaries. Nodes stop at boundaries of given area and their movements are limited if their new computed locations are beyond the area boundaries.

- (B_3): In *Wrap Around*, if nodes' new computed locations go beyond the area boundaries, they are wrapped around to other (opposite) sides when they reach to the boundaries of given area, according to toroidal surface.

5.2.3 Fuzzy Logic Parameters

Fuzzy rule-based systems can be used in many different research areas [353, 354]. Fuzzy Takagi and Sugeno (TS) [353] rule-based systems for control problems are briefly described as follows:

$$\begin{aligned} \text{Rule R}_\text{j}:\ & if\ \ x_1\ \text{ is }\ A_{j1}\ \text{and} \cdots \text{and}\ x_n\ \text{ is }\ A_{jn} \\ & then\ \ y_j = a_{0j} + a_{1j}x_1 + \cdots + a_{nj}x_n \end{aligned} \tag{5.1}$$

where $x = (x_1, x_2, ..., x_n)$ is an n-dimensional input, A_{nj} is a fuzzy membership, such as small or large, and y is a non-fuzzy output. Output of the fuzzy rule-base system is calculated from the following equation:

$$y = \frac{\sum_{j=1}^{p} \mu_j(x) \cdot y_j}{\sum_{j=1}^{N} \mu_j(x)}, \tag{5.2}$$

where

$$\mu_j(x) = \mu_{1j}(x) \otimes \mu_{2j}(x) \otimes \cdots \otimes \mu_{nj}(x) \tag{5.3}$$

and p is the total number of rules. Two different fuzzy inference systems are used: *fuzzy pair radial* and *angular forces*.

Fuzzy Pair Radial Force: A pair force system has one input, as distance with the given memberships and one crisp (exact) output, pressure which can take the fuzzy values 'push hard', 'push', 'no action', 'pull' and 'pull hard'. The rules of this system are listed in Table 5.1.

Fuzzy Pair Angular Force: The fuzzy angular force system also has one input as *distance to the target angle* with the given memberships and one crisp output as pressure. As a simple example, consider a node with two neighbours forming a

Table 5.1. Fuzzy Rules

(a) Pair Radial Force System	
Distance	**Pressure**
Very Far	No Action(0)
Far	Pull hard(-1)
Moderate	Pull(-0.5)
Close	Push(0.5)
Too Close	Push Hard(1)

(b) Pair Angular Force System	
Distance	**Pressure**
Very Far	Hard(1)
Far	Medium(0.75)
Moderate	Slow(0.5)
Close	Very Slow(0.25)
Too Close	Nothing (0)

$30°$ degree angle with each other and the given node of the vertex. Since the node has two neighbours, the target angle is $180°$ degrees. Therefore, by considering one of the neighbours as a reference point, the node applies angular force based on the fuzzy model and rule in Table 5.1.

As node's neighbours can exert different angular forces with regards to their locations relative to one another, two angular force strategies can be defined based on how the effective exerted forces are perceived on the node via its neighbours. Angles between the force-exerting node and its neighbours can be considered such that the force-exerting node is at the vertex with angle α ($0 < \alpha \leq 180°$) with each pair set of its neighbours where $\alpha = \angle(n_1\ n_{fv}\ n_2)$ (i.e. n_1 and n_2 are two neighbours of force-exerting node n_{fv}). Angular force strategies are considered as below:

- (A_1): In the *Smallest Angular Movement* strategy, all exerted angular forces are computed from neighbours, then the selected exerted force that causes the smallest node angular movement is chosen.
- (A_2): In the *Closest Neighbour*, the angular force from the closest neighbour is considered as the selected force exerted on the node.

5.3 Fuzzy Logic Relocation Models

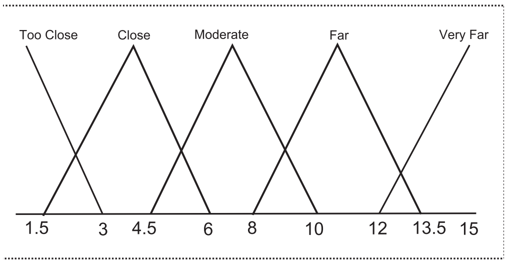
In the following sections, by using the fuzzy inference systems presented in Table 5.1 and the presenting radial pair forces and angular membership functions, three types of Fuzzy Logic Relocation models are introduced in Sections 5.3.1, 5.3.2 and 5.3.3. Section 5.3.1 presents model in which fuzzy parameters are set by *expert knowledge* based on the Table 5.1 and Figure 5.1.

Section 5.3.2 presents the models in which, for the initial iteration, fuzzy parameters are tuned by PSO with respect to the linear weighted combinations of performance metrics in terms of the percentage of coverage, uniformity and average movement (Equation 5.4). The parameter membership functions are defined according to the Tables 5.1 and 5.2 and Figure 5.3.

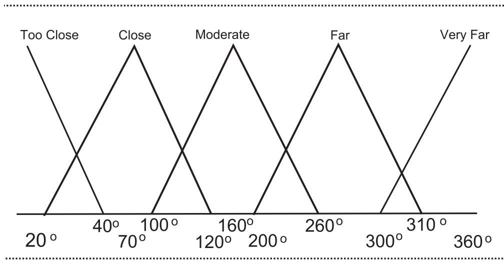
Section 5.3.3 extends the proposed models of Section 5.3.2 in which fuzzy parameters are locally tuned in each node movement iteration to consider the dynamic behaviour of mobile nodes and their neighbouring nodes (refer to Algorithm 5.1 and Figure 5.5).

5.3.1 Expert Knowledge Fuzzy Relocation Model

The fuzzy radial pair force system has one input (inter-nodal distance) with five triangular memberships (Figure 5.1(a)) and one crisp (exact) output as pressure which can take different fuzzy values of 'push hard', 'push', 'no action', 'pull' and 'pull hard'. The rules of this system are listed in Table 5.1. The fuzzy angular force system also has one input (i.e. distance to the target angle) with five triangular memberships and one crisp output as pressure (Figure 5.1(b)). By considering one of the neighbours as a reference point, the node applies angular force based on the fuzzy model and rule in Figure 5.1(b) and Table 5.1.



(a) Fuzzy Radial Memberships



(b) Fuzzy Angular Memberships

Figure 5.1: Memberships

5.3.1.1 Fuzzy Node Movement Algorithms

Depending on the type of applied pair forces, different node movement algorithms can be defined as follows (Figure 5.2),

- In Fuzzy Radial Movement (FRM), each node exerts and mutually senses radial force on/from its neighbours. The amount of node movement is the summation of push/pull virtual forces from its in-range neighbours.
- In Fuzzy Angular Movement (FAM), each node exerts a force on its neighbour based on aforementioned fuzzy decision and angular force strategies.
- In FRM Then FAM (FRAM), FAM is applied to result of FRM in consecutive iterations. The order of applying angular and radial forces is depicted in Figure 5.2
- In FAM Then FRM (FARM), FRM is applied to the result of FAM in consecutive iterations (Figure 5.2).
- In FRM And FAM (FRNAM), node movement is determined when both radial and angular forces are computed on the node from its neighbour in each iteration (Figure 5.2).

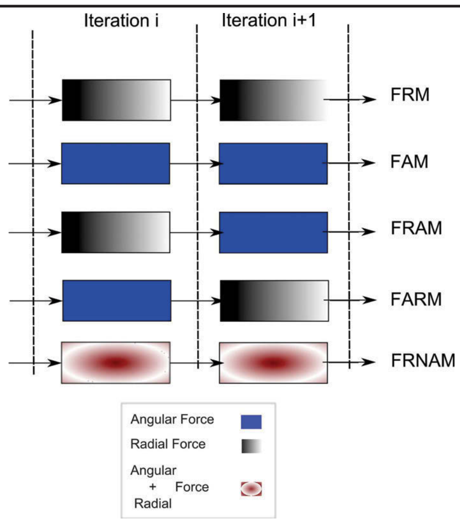


Figure 5.2: Fuzzy Node Movement Algorithms

5.3.2 Tuned Parameter Fuzzy Relocation Model

The circular zone around a node is defined as R_{zone} ($R_{zone} = k \cdot R_c$), which is used to obtain the fuzzy parameters from the nodes' neighbours residing in the given zone via particle swarm optimization (PSO). Both the fuzzy radial pair and fuzzy angular force systems have one input as distance with three Gaussian functions, one z-function and one s-function memberships defined in Table 5.2, and one crisp output, pressure which can take the fuzzy values from the rules listed in Table 5.1. Membership function parameters a, b, c, d, μ, σ are computed using particle swarm optimization. Figure 5.3 shows an example of respectively tuned radial and angular membership functions for angular strategy A_1, boundary condition B_2 and movement strategy $FRAM$. Hence, fuzzy parameters can be tuned using PSO with regard to linear weighted combinations of metrics in terms of the percentage of coverage, uniformity, and average movement equation,

$$F^* = argmax_F\{w_1 \cdot C(F) - w_2 \cdot U(F) - w_3 \cdot M(F)\} \tag{5.4}$$

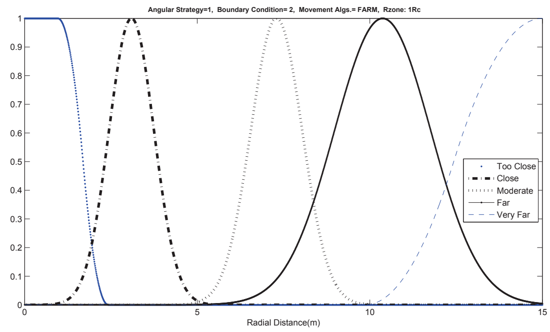
Table 5.2. Membership Functions

z-function
$f_z(x;a,b) = \begin{cases} 1, & x \leq a \\ 1-2\left(\frac{x-a}{b-a}\right)^2, & a \leq x \leq \frac{a+b}{2} \\ 2\left(\frac{x-b}{b-a}\right)^2, & \frac{a+b}{2} \leq x \leq b \\ 0, & x \geq b \end{cases}$
Symmetric Gaussian function
$f_g(x;\sigma,\mu) = e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
s-function
$f_s(x;c,d) = \begin{cases} 0, & x \leq c \\ 2\left(\frac{x-c}{d-c}\right)^2, & c \leq x \leq \frac{c+d}{2} \\ 1-2\left(\frac{x-d}{d-c}\right)^2, & \frac{c+d}{2} \leq x \leq d \\ 1, & x \geq d \end{cases}$

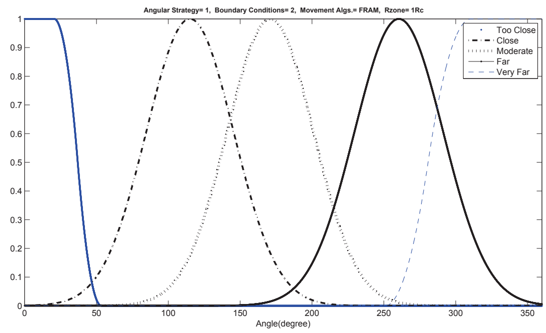
w_1, w_2, w_3 are respectively weights for coverage (C), uniformity (U), and average movement (M). F is a set of fuzzy parameters tuned by PSO with regard to the performance weights. Thus, parameters can be tuned based on one metric or a linear combination of metrics. The negative and positive signs are used where performance metrics should be minimised (i.e. movement, uniformity) or maximised (i.e. coverage) respectively. In order to tune parameters, PSO is applied in two different global and local zone ranges which are as follows:

Considering a *global* range, PSO is applied on all deployed nodes over a whole 2D rectangular field ($[x_{min}, x_{max}] \times [y_{min}, y_{max}]$), while in *local zone-range*, proportion of nodes N_{sel} from a set of deployed nodes N_{total} ($N_{sel} \leq N_{total}$) are randomly selected with uniform distribution. PSO is applied for each selected nodes with a zone-range of R_{zone} ($R_{zone} = k \cdot R_c$) around the selected node by taking account node's neighbours residing within its R_{zone} range.

It should be noted that, in both local and global ranges, boundary conditions are



(a) Radial Membership



(b) Angular Membership

Figure 5.3: Radial and Angular Membership function

considered in tuning fuzzy parameters.

5.3.2.1 PSO structures

Constriction coefficient PSO is used similar to [355]. Thus, in this approach the velocity update equation is as follows:

$$\upsilon_{ij(t+1)} = \chi \left[\upsilon_{ij(t)} + \phi_1 \left(y_{ij(t)} - x_{ij(t)}\right) + \phi_2 \left(\hat{y}_{ij(t)} - x_{ij(t)}\right)\right] \tag{5.5}$$

y_{ij} is the particle best and $\hat{y}_{ij}$ is the global best particles and,

$$\chi = \frac{2k}{\left|2 - \phi - \sqrt{\phi(\phi - 4)}\right|} \tag{5.6}$$

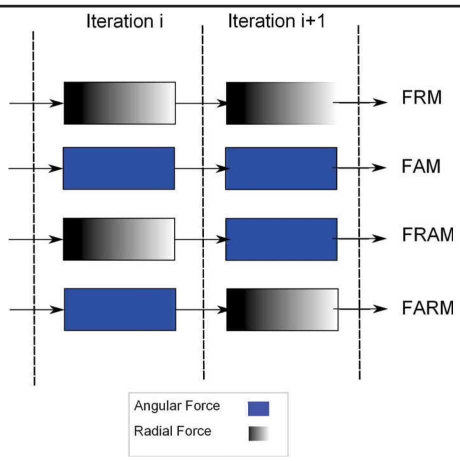


Figure 5.4: Fuzzy Node Movement Algorithms, Figure 5.2

with $\phi = \phi_1 + \phi_2$, $\phi_i = c_i r_i$, for $i = 1, 2$. Equation 5.6 is used under the constraint that $\phi \geq 4$ and k, $r_i \in [0, 1]$. The parameter k in the Equation 5.6 controls the exploration and exploitation. For $k \sim 0$, fast convergence is expected and for $k \sim 1$, we can expect slow convergence with high degree of exploration [355]. Each particle consists of two arrays: one is related to the memberships of the pair force fuzzy systems and another one is related to the memberships of the angular force fuzzy systems. Each fuzzy system has 5 memberships and each membership is specified by its mean and variance; therefore each array has 10 cells.

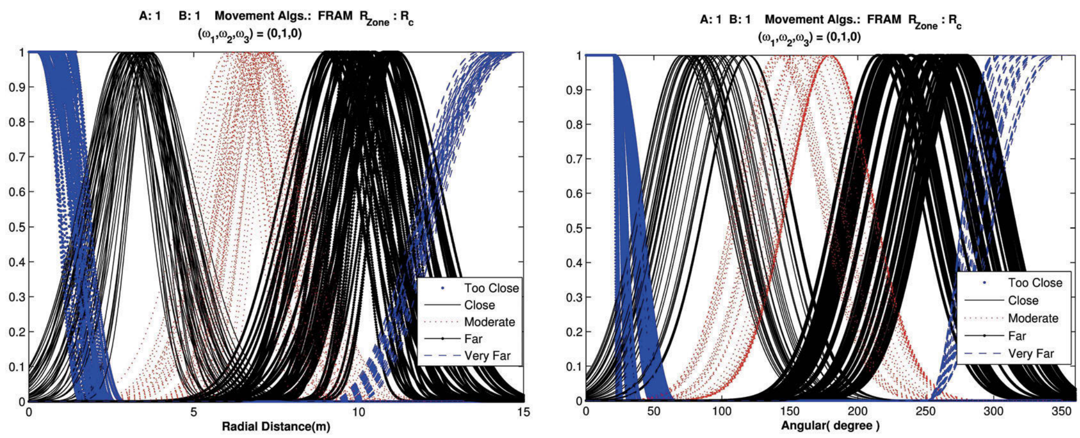
5.3.2.2 Fuzzy Node Movement Algorithms

In the proposed model, the fuzzy node movement algorithms applied are *Fuzzy radial movement* (FRM), *Fuzzy angular Movement (FAM)*, *FRM then FAM (FRAM)*, and *FAM then FRM (FARM)* (Figure 5.4).

5.3.3 Iteratively Tuned Parameter Fuzzy Relocation Model

Fuzzy node relocation model presented modified in a way such that in nodes, autonomously fuzzy parameters are tuned locally at each iteration of movement. Like in Section 5.3.2, the circular zone around the node is defined as a circle with a radius of R_{zone} ($R_{zone} = k \cdot R_c$, $R_{zone} \geq R_c$) where $k = 1$, with the node in the centre of the circle. Node's circular zone is used to obtain the fuzzy parameters from the nodes' neighbours residing in the given zone range by PSO in each iteration according to Algorithm 5.1.

Fuzzy rule-based systems [353, 354] are used in variety of different applications. In this section, the fuzzy rules and fuzzy inference systems of Sections 5.3.1 and 5.3.2 are applied to the sensor nodes. Similar to section 5.3.2, by using particle swarm optimization method in each iteration for each node, membership function parameters $(a, b, c, d, \mu, \sigma)$ (Table 5.2) are computed according to Equation 5.4). Three Boundary strategies of Section 5.2.2 are used in our scenario. Figure 5.5



(a) Radial Membership (b) Angular Membership

Figure 5.5: Radial and Angular Membership functions of Node after 50 Iterations

ALGORITHM 5.1: Iteratively tuned fuzzy logic relocation model

Input:
Set of nodes s_i: $i \in G$, where $G = \{1, \cdots, N_{total}\}$
Set of selected s_i:s_i^j, $j \in H \subseteq G$,$H = \{0, \cdots, N_{Sel}\}$
r_η:r^{th} iteration:$\eta = \{0, \cdots, T_{final} \equiv \text{final iteration} \}$
$Loc^D_{\{s_i(r)\}}$: s_i's location in r_η in D-dimension space
s_i's transmission range R_{c_i} and sensing range R_{s_i}
s_i's zone Range $R_c \leq R_{zone} = k \cdot R_c$ with $k = 1$
t_r : # movement Alg. types in r^{th} iteration, $t = 0, 1$
$\phi^b_{s_i^j}(r_\eta)$: Set of s_i's R_{zone} neighbours $s_l^j(r_\eta)$ in r_η s.t., $\left\| s_i(r_\eta) - s_l^j(r_\eta) \right\| \leq R_{zone}$
$A_{n_{\{1,2\}}}$:Angular Strategy, $B_{m_{\{1,2,3\}}}$:Boundary Condition
$Mov_{p=\{FRM,FAM,FRAM,FARM\}}$:Movement Alg.
$w_{q_{\{Coverage,Uniformity,AverageMovement\}}}$:Metrics Weights
$T_r\left(F_i^j, B_m\right)$ computes $Loc^D_{\{s_i(r)\}}$ for $F_i^j(total)$ & B_m

Output:
$Loc^D_{\{s_i(T_{final})\}}$for iteration of T_{final} and $\forall i \in G$

foreach $r \in \eta$ **do**
  foreach $s_i^j \in H$ **do**
    switch $t_r \in r$ **do**
      case $t_r = 0$ $Mov_{p=\{FRM,FAM\}}$
      case $t_r = 1$ $Mov_{p=\{FRAM,FARM\}}$
    endsw
    forall the $s_l^j(r_\eta) of \phi^b_{s_i^j}(r_\eta)$ **do**
      Calculate $F^* \leftarrow PSO(w_q, A_n, B_m, t_r, R_{zone})$
    end
    foreach $s_l^j(r_\eta)$ *of* $\phi^b_{s_i^j}(r_\eta)$ **do**
      Calculate $f_i^j(t_r) \leftarrow FuzzyForce(F^*, Mov_p)$
    end
    Calculate $F_i^j(total) \leftarrow \bigcup^l_{l \in \phi^b_{s_i^j}(t_r)} \left(f_i^j(t_r)\right)$
  end
  foreach $s_i \in G$ **do**
    Calculate $Loc^D_{\{s_i(r)\}} \leftarrow T_r\left(F_i^j(total), B_m\right)$
  end
end

presents an example of respectively tuned radial and angular membership functions for angular strategy A_1, boundary condition B_1 and movement strategy of FRAM after 50 iterations for a node of $R_{zone} = R_c$ $((\omega_1, \omega_2, \omega_3) = (0, 1, 0))$. The figure shows the variations of membership functions as the parameters are tuned in each iteration.

5.3.3.1 Fuzzy Node Movement Algorithms

Fuzzy node movement algorithms of FRM, FAM, FRAM, and FARM that are presented in Section 5.3.2, are used in this section.

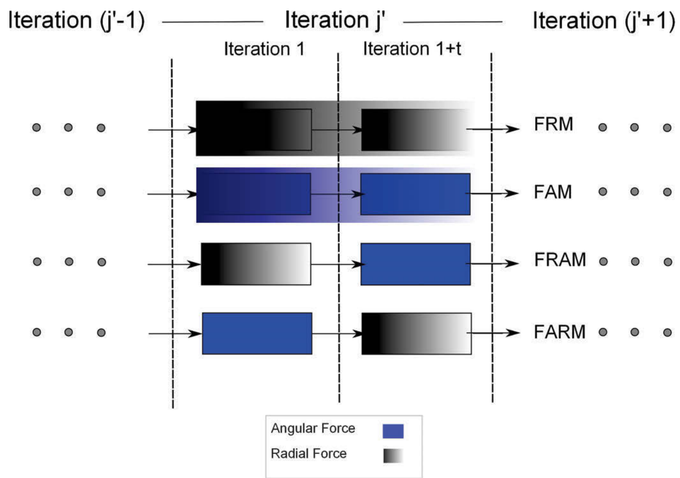


Figure 5.6: Fuzzy Node Movement Algorithms, Figure 5.4

Regarding Sections 5.3.1 and 5.3.2, actual physical movement iteration j' in this model is divided into two sub-iterations t and $1+t$ (here $t = 1$), to represent the number of type change of movement algorithm applied to node in each $j^{'}$th iteration. In our proposed model, by applying FRAM/FARM, each node moves in $1+t$ sub-iterations as the radial/angular movement algorithms and then angular/radial movement algorithms are applied on the given node. t shows number of type change of movement algorithm in the given $j^{'}$th iteration which $t = 0$ for

FRM/FAM as the type of radial/angular algorithm does not change, and $t = 1$ for FRAM and FARM respectively because of the type of radial/angular movement algorithms used in the relocation algorithm does change (Figure 5.6).

In this section, as fuzzy parameters are tuned in each movement iteration $j^{'}$, movement algorithm strategies are presented as *iFRM*, *iFAM*, *iFRAM* and *iFARM* to distinguished them from previous movement algorithm strategies in Section 5.3.2 whose fuzzy parameters were only computed for the initial iteration and were not changed for the rest of movement iterations. The fuzzy logic movement model is shown briefly in the Algorithm 5.1.

5.4 Performance Evaluation

5.4.1 Performance Metrics

The performance metric used in this chapter are as follows:
Percentage of Coverage: represents the efficiency of movement algorithms in terms of how the coverage of given area is improved as nodes relocate (Section 3.3.1.2).

Uniformity: is defined as the average local standard deviation of inter-nodal distances [227].

$$U_i = \left(\frac{\sum_{j=1}^{k_i} (D_{i,j} - M_i)^2}{k_i} \right)^{1/2}, U = \frac{\sum_{i=1}^{N} U_i}{N}, \tag{5.7}$$

where N is the total number of nodes, k_i is the number of neighbours of the ith node, $D_{i,j}$ is the distance between the ith and jth nodes, and M_i is the mean inter-nodal distance between the ith node and its neighbours [227].

Average Movement: is defined as the total movement of nodes in each iteration over the number of nodes in the given iteration. Since movement is related to the amount of consumed energy by each node, the average movement of nodes in each iteration can be considered as a suitable metric.

5.4.2 Results

The results of the proposed model are presented in the following sections.

5.4.2.1 Expert Knowledge Relocation Model

The proposed node movement algorithms were simulated using Matlab. $N = 100$ nodes were uniformly distributed in the rectangular field of $[-100, 100] \times [-100, 100]\ m^2$. Each node had a transmission and sensing range of R_c=R_s=15 m. Each algorithm was simulated 500 times with boundary conditions B_1, B_2, and B_3, and force strategies of A_1 and A_2. In each experiment, the node movement algorithm was run for 500 iterations. The proposed node algorithms are compared with DSSA and tabulated for $iterations$=$1, 100, 200, 300, 400, 500$ in Tables 5.3, 5.4, and 5.5.

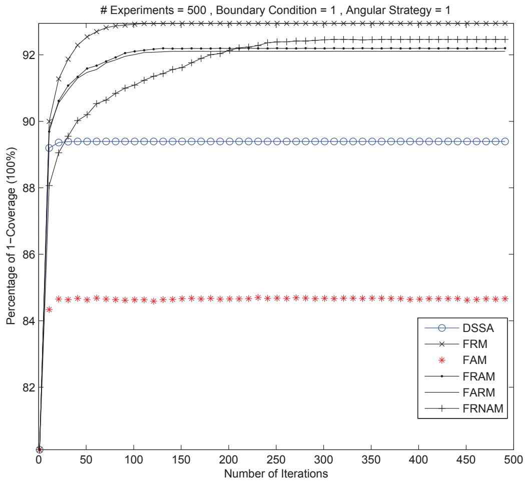
In Table 5.3, the percentage of improvement of fuzzy movement algorithms over DSSA is listed for the given boundary conditions and angular force strategies for the selected iterations points. Similarly, Tables 5.4 and 5.5 compare the performance of fuzzy movement algorithms with DSSA in terms of uniformity and average movement respectively.

The results for angular force strategies A_1 and A_2 are similar. Therefore, only results for A_1 are presented in Figures 5.7, 5.8, and 5.9. Results for A_2 is shown in Appendix C. The results for A_2 are summarised for selected iteration points in Tables 5.3, 5.5 and 5.4. The number of allowed iterations is an important factor for choosing proper movement algorithms. In Figure 5.7(a), FRNAM performs better than FRAM and FARM beyond the 250th iteration, while in Figures 5.7(b) and 5.7(c) FRAM and FARM outperform FRNAM.

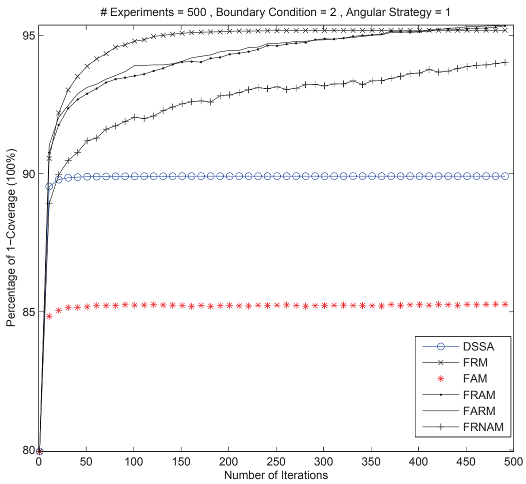
The Figures 5.8(a), 5.8(b), and 5.8(c) show that DSSA outperformed the fuzzy algorithms for the boundary conditions. One main reason is due to the sluggish response and slow node movements that make this algorithm a candidate for network with random and sporadic node failures. The DSSA algorithm is suitable for a network with limited topological changes. With respect to Figures 5.8(a) and 5.8(b), it can be seen that FAM had the worst performance for the boundary conditions of B_1 and B_2 as the number of iterations increased. In Figure 5.8(c), FRNAM had the lowest performance. Except for Figure 5.8(c), most of fuzzy movement algorithms effectively reduced their performance differences with the DSSA as number of movement iterations increased.

Boundary conditions affect the performance of the fuzzy node movement algorithms and as well as DSSA (Figures 5.9(a), 5.9(b) and 5.9(c)). FRM outperformed almost all movement algorithms including DSSA, while FAM's performance was worst. FRAM's and FARM's performances were similar except in terms of average movement (Figures 5.9(b) and 5.9(c)).

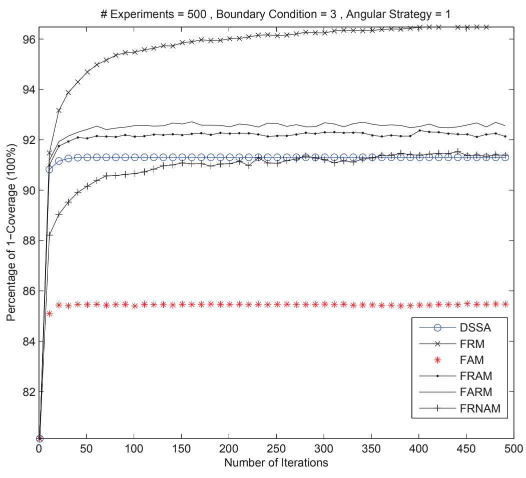
Results up to 500 iterations showed that unlike DSSA which benefits from expected global node density, the proposed models based on the nodes' local visibility outperformed or were comparable to DSSA.



(a) B_1, A_1

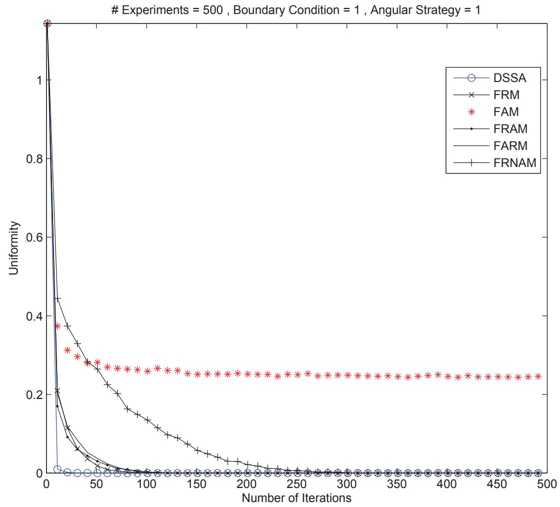


(b) B_2, A_1

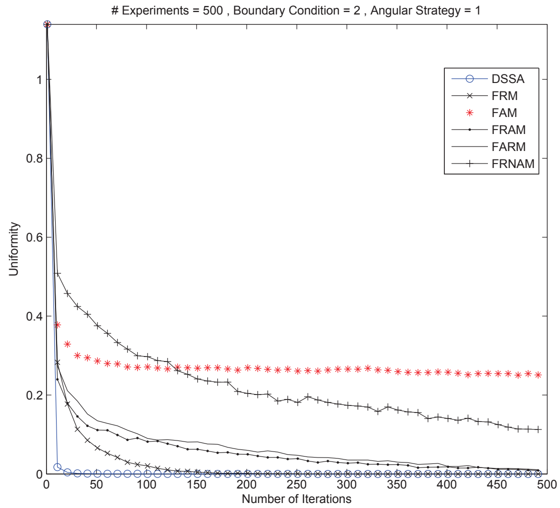


(c) B_3, A_1

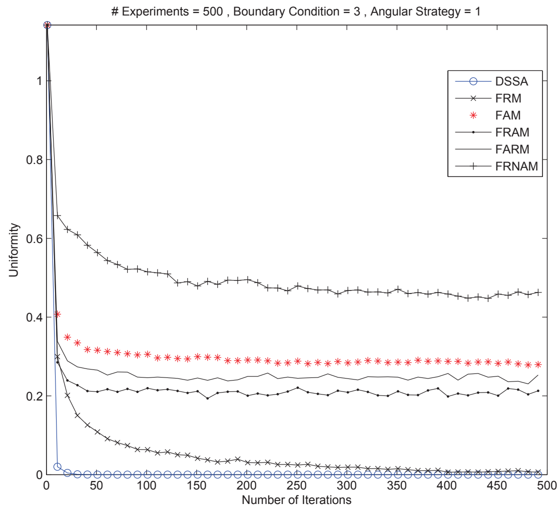
Figure 5.7: Percentage of 1-Coverage (100%) for different boundary conditions with angular force strategy A_1



(a) B_1, A_1

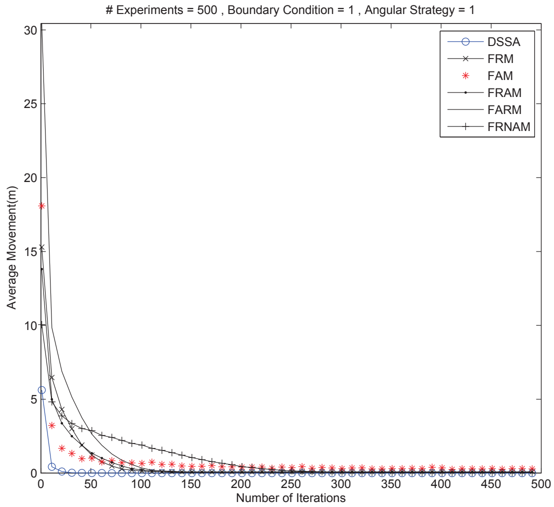


(b) B_2, A_1

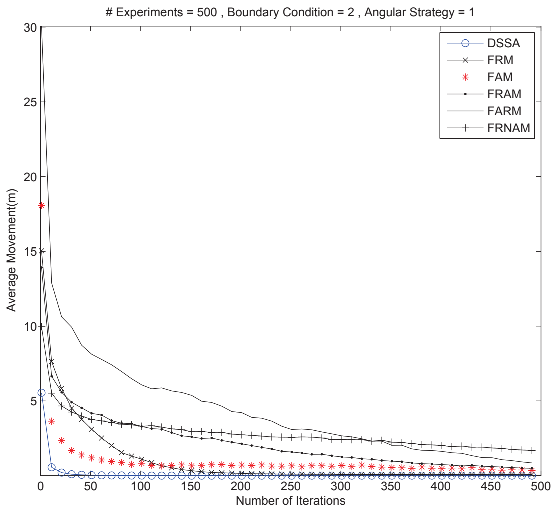


(c) B_3, A_1

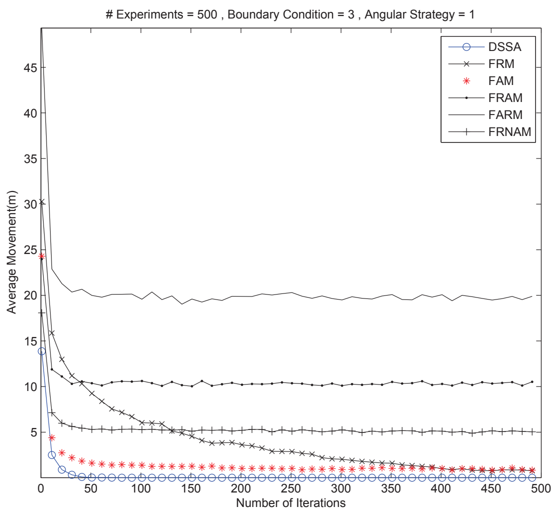
Figure 5.8: Uniformity for different boundary conditions with angular force strategy A_1



(a) B_1, A_1



(b) B_2, A_1



(c) B_3, A_1

Figure 5.9: Average movement for different boundary conditions with angular force strategy A_1

Table 5.3. Percentage of Improvement over DSSA for 1-Coverage (100%)

Algs. \ Iteration		1			100			200			300			400			500		
		B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3
FRM	A_1	00.0000	00.0000	00.0000	3.9440	***5.4069***	4.5806	3.9762	***5.8240***	5.1905	3.9714	***5.8670***	5.4366	3.9732	***5.8704***	5.6209	3.9762	*5.8699*	5.6713
	A_2	00.0000	00.0000	00.0000	3.9147	5.3413	4.5806	3.9465	5.7419	5.1905	3.9422	5.7783	5.4366	3.9433	5.7785	5.6209	3.9465	5.7746	5.6713
FAM	A_1	00.0000	00.0000	00.0000	-5.3432	-5.1933	-6.4914	-5.3086	-5.1905	-6.4143	-5.2898	-5.1997	-6.3937	-5.3169	-5.1781	-6.4062	-5.3061	-5.1843	-6.4032
	A_2	00.0000	00.0000	00.0000	-5.2770	*-5.0180*	-6.3420	-5.3149	*-5.0129*	-6.3650	-5.2993	*-4.9497*	-6.2782	-5.3092	*-4.9684*	-6.2809	-5.3155	*-4.9122*	-6.2848
FRAM	A_1	00.0000	00.0000	00.0000	3.0295	4.0898	0.8966	3.1353	*4.9053*	0.8896	3.1313	*5.5315*	0.9787	3.1294	*5.7820*	1.1464	3.1353	5.9681	1.0423
	A_2	00.0000	00.0000	00.0000	3.0009	*4.1008*	0.9794	3.0579	4.6742	0.9403	3.0550	5.3855	1.1449	3.0626	5.6912	1.2434	3.0579	*5.9924*	1.0547
FARM	A_1	00.0000	00.0000	00.0000	2.9025	*4.3887*	1.3824	3.0333	*5.0614*	1.4049	3.0333	*5.4669*	1.4736	3.0304	5.7871	1.4155	3.0333	***6.0807***	1.3738
	A_2	00.0000	00.0000	00.0000	2.8501	4.1155	1.2232	2.9543	4.8767	1.4779	2.9537	5.3341	1.3267	2.9524	*5.8121*	1.4461	2.9540	5.9485	1.5042
FRNAM	A_1	00.0000	00.0000	00.0000	1.8200	*2.3714*	-0.7079	3.0658	*3.2834*	-0.2614	3.4225	*3.6447*	-0.1676	3.4339	*4.1839*	0.1176	3.4346	*4.6510*	0.0283
	A_2	00.0000	00.0000	00.0000	1.7148	2.0994	-0.7004	3.0337	2.9841	-0.2925	3.4217	3.7592	-0.2445	3.4557	4.0720	0.0229	3.4566	4.4094	-0.0762

Table 5.4. Uniformity (Difference with DSSA)

Algs. \ Iteration		1			100			200			300			400			500		
		B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3
FRM	A_1	0.0000	0.0000	0.0000	0.5646	0.0201	0.0641	0.0000	0.0015	0.0283	0.0000	0.0000	0.0165	0.0000	0.0000	0.0075	0.0000	0.0000	0.0062
	A_2	0.0000	0.0000	0.0000	0.0000	0.0198	0.0641	0.0000	0.0014	0.0283	0.0000	0.0000	0.0165	0.0000	0.0000	0.0075	0.0000	0.0000	0.0062
FAM	A_1	0.0000	0.0000	0.0000	0.2613	0.2708	0.3099	0.2519	0.2680	0.2932	0.2494	0.2645	0.2824	0.2485	0.2562	0.2852	0.2451	0.2535	0.2812
	A_2	0.0000	0.0000	0.0000	0.2524	0.2754	0.2947	0.2511	0.2694	0.2967	0.2436	0.2633	0.2890	0.2414	0.2614	0.2904	0.2413	0.2579	0.2841
FRAM	A_1	0.0000	0.0000	0.0000	0.0035	0.0812	0.2041	0.0000	0.0495	0.2042	0.0000	0.0288	0.2139	0.0000	0.0206	0.2030	0.0000	0.0130	0.2064
	A_2	0.0000	0.0000	0.0000	0.0022	0.0703	0.2037	0.0000	0.0525	0.2039	0.0000	0.0287	0.1946	0.0000	0.0158	0.1968	0.0000	0.0090	0.1991
FARM	A_1	0.0000	0.0000	0.0000	0.0032	0.0936	0.2491	0.0000	0.0588	0.2508	0.0000	0.0401	0.2471	0.0000	0.0206	0.2503	0.0000	0.0109	0.2449
	A_2	0.0000	0.0000	0.0000	0.0046	0.0987	0.2488	0.0000	0.0596	0.2407	0.0000	0.0397	0.2487	0.0000	0.0195	0.2521	0.0000	0.0126	0.2465
FRNAM	A_1	0.0000	0.0000	0.0000	0.1456	0.2936	0.5103	0.0218	0.2052	0.4817	0.0007	0.1754	0.4723	0.0000	0.1411	0.4559	0.0000	0.1066	0.4564
	A_2	0.0000	0.0000	0.0000	0.1481	0.3115	0.5270	0.0214	0.2353	0.4980	0.0012	0.1715	0.4902	0.0000	0.1328	0.4760	0.0000	0.1159	0.4673

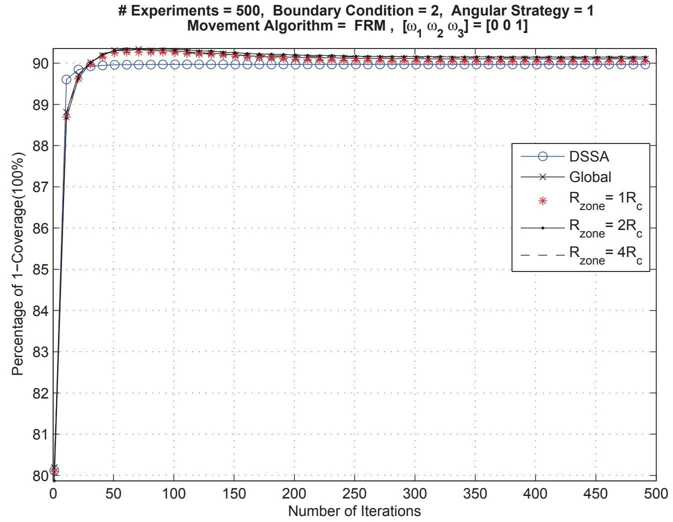
Table 5.5. Average Movement (Difference with DSSA)

Algs. \ Iteration		1			100			200			300			400			500		
		B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3	B_1	B_2	B_3
FRM	A_1	9.6707	9.4966	16.4146	0.1354	1.0872	6.3532	0.0821	0.1471	3.5320	0.0815	0.0828	2.1484	0.0793	0.0730	1.0122	0.0821	0.0795	0.7508
	A_2	9.6651	9.5602	16.4146	0.1360	1.0312	6.3532	0.0826	0.1496	3.5320	0.0826	0.0786	2.1484	0.0798	0.0780	1.0122	0.0826	0.0796	0.7508
FAM	A_1	12.4635	12.5251	10.4157	0.6644	0.8201	1.3858	0.4179	0.6586	1.1130	0.3142	0.6774	0.9247	0.3462	0.4038	0.9579	0.2632	0.3357	0.7634
	A_2	13.5956	13.7317	11.5092	0.4752	0.7497	1.2986	0.3429	0.5892	1.2200	0.2420	0.5994	1.1453	0.1735	0.4843	1.1401	0.1567	0.4603	0.9587
FRAM	A_1	8.1834	8.3727	10.1109	0.2027	3.3538	10.4838	0.0465	2.1139	10.1308	0.0450	1.2737	10.2510	0.0431	0.7926	10.0243	0.0465	0.5051	10.1870
	A_2	8.5897	8.7871	10.5963	0.1855	3.1836	10.6065	0.0540	2.1682	10.6096	0.0543	1.2725	10.4244	0.0550	0.8080	10.3682	0.0540	0.4178	10.1487
FARM	A_1	24.8153	24.5265	35.4379	0.3402	6.0371	20.0483	0.0836	4.1215	19.6631	0.0845	2.7555	19.3872	0.0836	1.6168	19.9004	0.0836	0.8700	19.6739
	A_2	25.2769	25.4411	36.3566	0.4932	6.4395	20.3673	0.1041	4.4381	19.3799	0.0992	2.7364	20.4896	0.0964	1.5043	20.1484	0.0980	1.0196	19.8807
FRNAM	A_1	4.4278	4.4170	4.2079	1.9271	3.3052	5.2964	0.4522	2.7299	5.1444	0.0612	2.4087	5.3287	0.0375	2.0284	5.1386	0.0354	1.5967	5.0356
	A_2	4.4854	4.6074	4.3067	2.0337	3.3332	5.0886	0.4883	2.9437	5.1271	0.0686	2.3784	5.0571	0.0374	2.0097	4.9054	0.0365	1.6857	4.9355

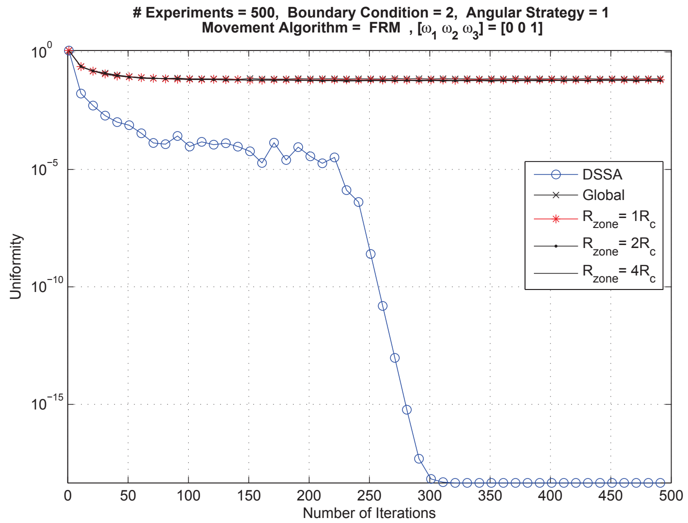
5.4.2.2 Fuzzy Tuned Parameter Relocation Model based on PSO

The proposed model was simulated similar to that in Section 5.4.2.1. In order to locally obtain fuzzy parameters, $N_{sel} = 30$ of total deployed nodes $N_{tot} = 100$ with zone ranges of ($R_{zone} = (1, 2, 4) \cdot R_c$) were selected randomly. The fuzzy parameters were tuned via PSO ($k = 0.5$, $c_1 = 3$, $c_2 = 3$ Equation 5.6) with boundary conditions of B_1, B_2, and B_3 and angular strategies of A_1 and A_2. The membership parameters were also obtained globally in a rectangular field of $[-100, 100] \times [-100, 100]$ m^2. By tuning fuzzy parameters both globally and locally ($R_{zone} = (1, 2, 4) \cdot R_c$) by PSO, relocation algorithms were simulated 500 times using 500 iterations for different boundary conditions and angular strategies. For the sake of brevity, only the results based on tuned fuzzy parameters with $(\omega_1, \omega_2, \omega_3) = (0, 0, 1)$ (Equation 5.4), with zone range R_{zone}=$1 \cdot R_c$ and A_1 and B_2 and movement algorithm of FRM are presented in Figure 5.10. The rest of the results more or less follow the same trends.

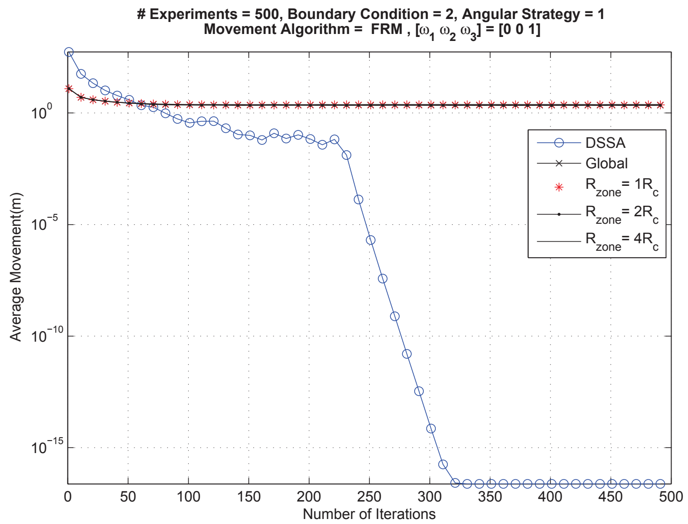
The performance of all of the movement strategies with R_{zone}=R_c and A_1 and B_2 ($N_{sel} = 30$, $N_{tot} = 100$) is compared to DSSA (Figure 5.11). In Figure 5.10, as R_{zone} reduces, performance degrades. Figure 5.10 shows that, even when PSO is applied locally to zone range of $R_{zone} = R_c$ and for 30% of total nodes, performance still is comparable to the case where PSO is applied globally on all the nodes over the whole given area. Figure 5.10 also shows that proposed model either outperform or is comparable to DSSA for different movement strategies, as DSSA benefits from expected global node density. Since for each node tuned parameters were obtained once in the first iteration and did not change in remaining iterations, the proposed model still had acceptable performance when compared to DSSA. Figure 5.11 shows FRAM and FARM had the highest and FAM had the lowest percentage of 1-coverage and FRM has a comparable performance to DSSA.



(a) Percentage of 1-Coverage,(A_1,B_2,FRM)

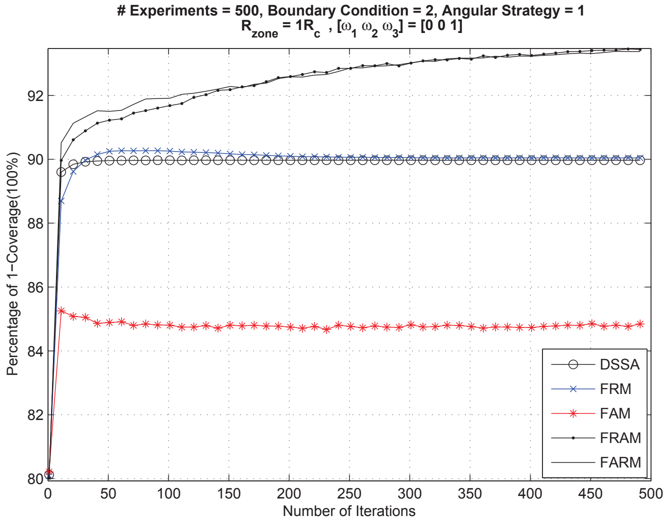


(b) Uniformity,(A_1,B_2,FRM)

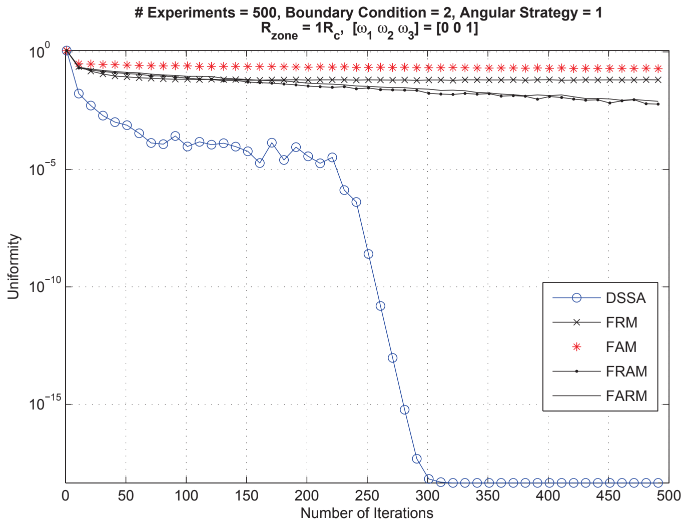


(c) Average Movement,(A_1,B_2,FRM)

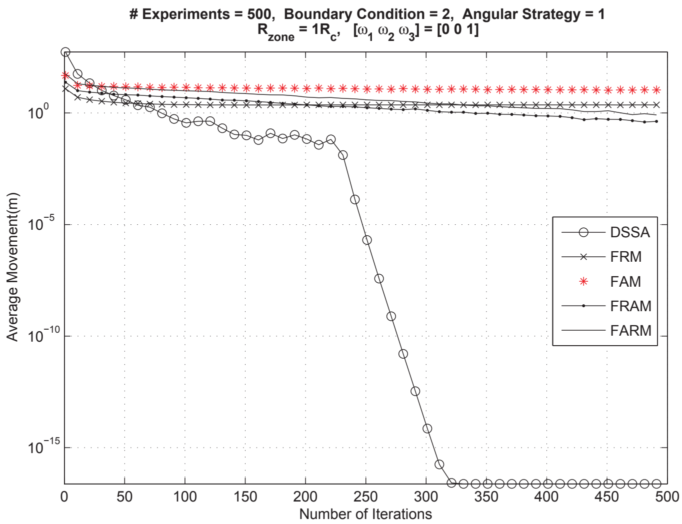
Figure 5.10: Performance Comparison of Relocation Algorithm for globally and locally ($R_{zone} = f1, 2, 4g \cdot R_c$) Tuned fuzzy parameters



(a) Percentage of 1-Coverage



(b) Uniformity



(c) Average Movement

Figure 5.11: Performance of Different Movement Strategies with Boundary condition B_2 and Angular Force Strategy A_1

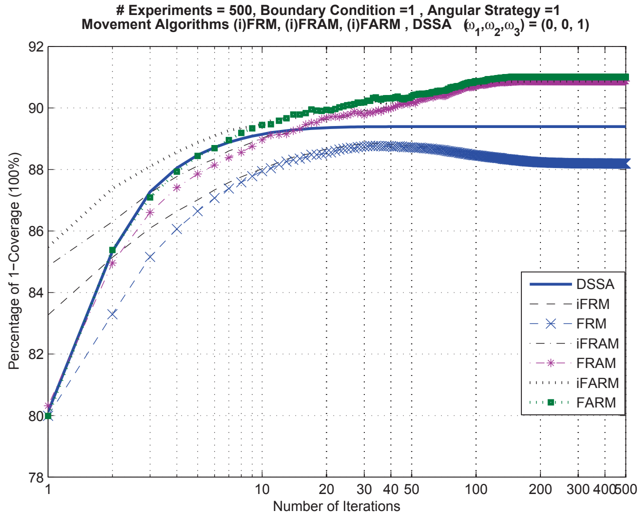
With regard to uniformity and average movement, FRM, FAM, FRAM and FARM had similar performance and are comparable to DSSA. However, depending on different linear combinations of weights (ω_1,ω_2,ω_3) (Equation 5.4), the performance of relocation algorithms with the different movement strategies, FRM, FAM, FRAM and FARM can vary.

In the following section, the proposed node relocation algorithm is extended such that the fuzzy parameters are locally tuned in each of movement iteration with respect to the nodes' status and their neighbouring nodes (Equation 5.4).

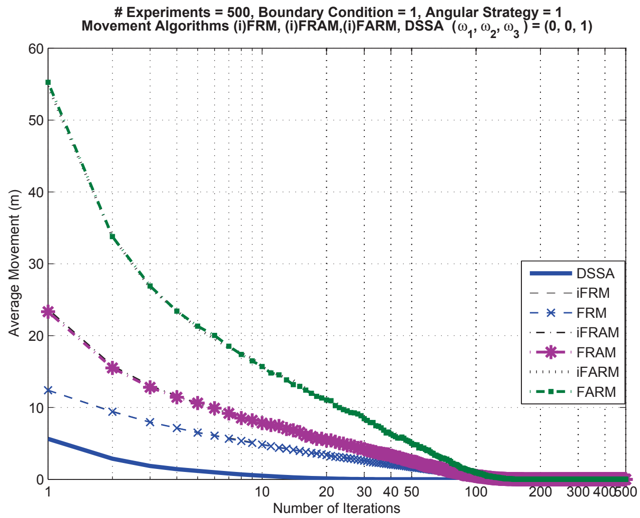
5.4.2.3 Fuzzy Iteratively Tuned Parameter Relocation Model based on PSO

The proposed model is simulated similar to that in Section 5.4.2.1. Similar to Section 5.3.2, by using PSO with boundary conditions of B_1, B_2, B_3 and angular strategies of A_1 and A_2, fuzzy parameters are locally tuned for total deployed nodes $N = 100$ with R_{zone}=R_c in each movement iteration. Nodes are relocated based on the Algorithm 5.1.

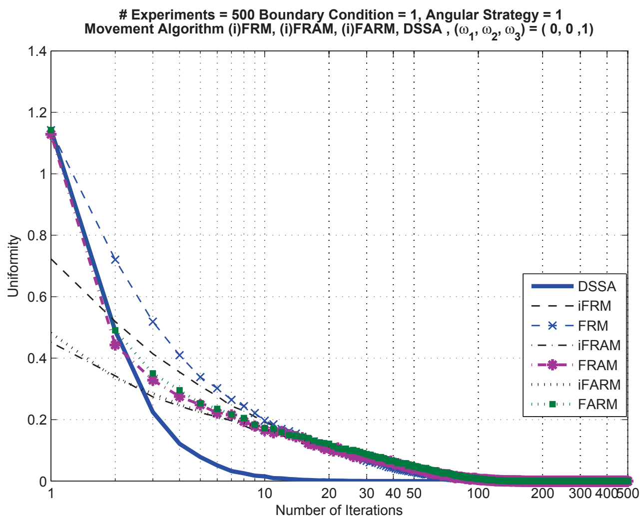
The rest of the results more or less follow the same trends. For the sake of brevity, only the results based on tuned fuzzy parameters with $(\omega_1, \omega_2, \omega_3) = (0, 0, 1)$ (Equation 5.4), with zone range R_{zone}=$1 \cdot R_c$ and A_1 and B_1 and movement algorithm of FRM, FRAM and FARM are presented in Figure 5.12. The FRM and iFRM behaviours are compared to DSSA. Results in the case where fuzzy parameters are tuned based on different weight of performance ($(\omega_1, \omega_2, \omega_3) = (1, 0, 0), (0, 1, 0), (0, 0, 1)$) are shown in Figure 5.13 with B_1 and A_1. In Figure 5.12, the proposed relocation model with different movement algorithms performance is compared to DSSA. It is also compared with the results from section 5.3.2 to see the performance differences if the fuzzy parameters are tuned in each iteration



(a) Percentage of 1-Coverage

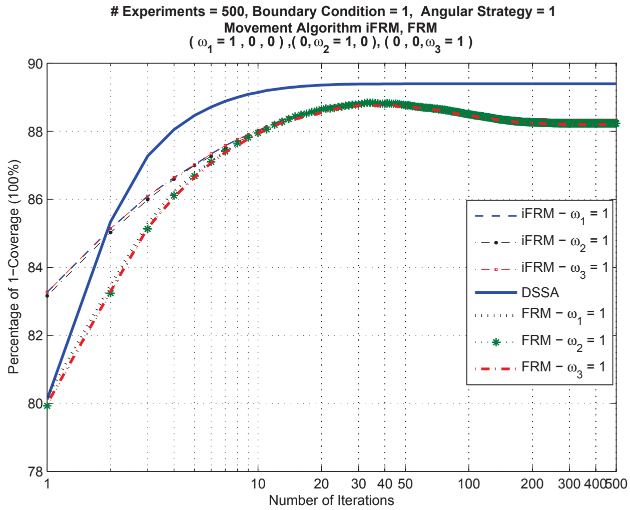


(b) Uniformity

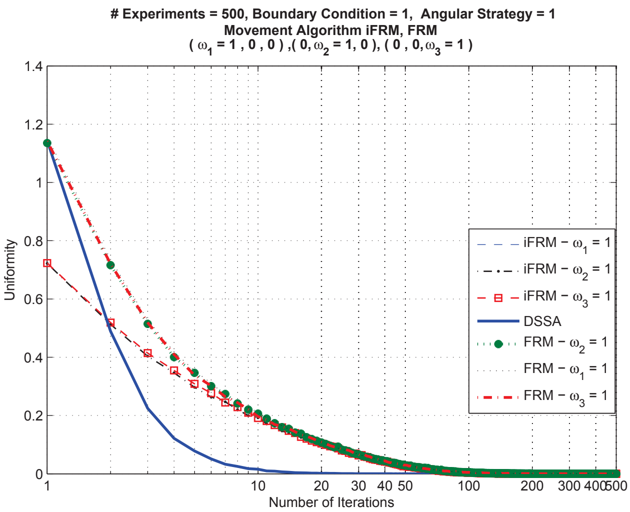


(c) Average Movement

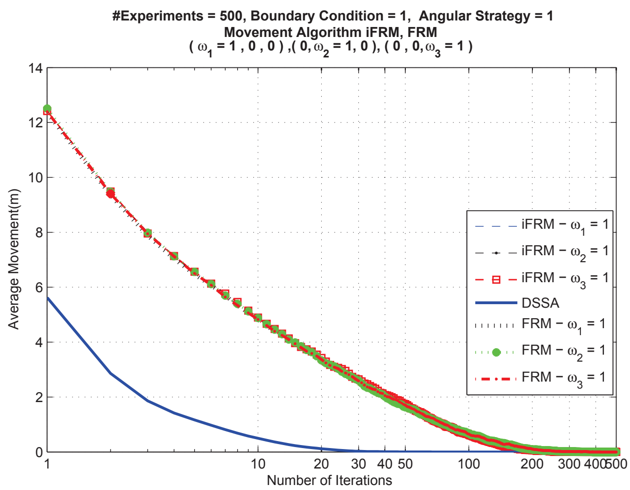
Figure 5.12: Comparison of Performances of Movement Algorithms for $\omega_1 = 0, \omega_2 = 0, \omega_3 = 1$



(a) Percentage of 1-Coverage



(b) Uniformity



(c) [Average Movement

Figure 5.13: Comparison of Performances of Movement Algorithms (FRM) for $(\omega_1, \omega_2, \omega_3)$=$(1, 0, 0), (0, 1, 0), (0, 0, 1)$

instead of initial movement iterations. Figure 5.5 also depicts tuned parameters for angular and radial membership functions of a given node for the 50 iterations. As shown, these parameters change for the different iterations in a given range. With regard to the results in Figures 5.12 and 5.13, following conclusions can be drawn:

- Performances of relocation algorithms (iFRM,iFRAM,iFARM) whose fuzzy parameters are tuned iteratively outperform or are comparable to their legacy algorithms (FRM, FRAM, FARM, Section 5.3.2) in which tuning is only done for the initial iteration. The performance differences decrease in all performance metrics as the number of iterations increase. Therefore, according to Figures 5.12 and 5.13, fuzzy parameters can be iteratively tuned to a certain iteration such as 10th iteration. As the number of iterations increases, the uniformity in node distribution improves and the performance differences decreases. This is because beyond a given number of iterations, changes in the fuzzy parameters are not so significant. Thus, from a given iteration, tuned parameters can be used without further modifications if the distribution of nodes does not abruptly change during iterative nodes relocations.
- It is expected that the proposed relocation model is effective for deployments that have large variations between neighbouring nodes.
- The results and Figure 5.5 show that the tuned parameters of member functions fluctuate and vary within a certain range, which resembles the parameters and behaviour of logic of type-2 fuzzy system [356]. Therefore, instead of iterative parameter tuning, in the first iteration, parameters of type-2 can be tuned and kept for the remaining iterations without further modifications.
- Fuzzy logic relocation models both in iterative and first-time locally-tuned parameters outperform or match the DSSA performances even though DSSA benefits implicitly from expected globally node density in the network.

5.5 Conclusion

In the chapter, node relocation models based on force-based fuzzy node movement algorithms (i.e. virtual push-pull forces among the nodes) were proposed. The fuzzy logic relocation models were shown to be appropriate solutions to address the uncertainty of autonomous nodes among indefinite choices of movements, if proper and justifiable fuzzy parameters and membership functions should be selected and tuned at the beginning or in the nodes' movement iterations. By tuning the fuzzy parameters, node relocation performance improves as the amount of relocations are modified by the updated status of nodes and their neighbours.

The results show that our proposed models either outperform or match DSSA in terms of percentage of coverage, uniformity and average movement, even the tuned parameters are obtained locally within nodes ranges.

CHAPTER 6

Cooperative Recovery of Coverage Holes in WSNs via Disjoint Spanning Trees

6.1 Introduction

Continuous reductions in cost and size, combined with simultaneous growth in mobility and processing power, have increasingly enabled wireless sensor networks (WSNs) [2] to be productively employed in many new applications, especially in hostile environments, such as in distributed and decentralised monitoring of tsunamis, wildfires, earthquakes and volcanoes [9–12]. As new applications and technical demands emerge and the scale of WSNs grows ever larger, new challenges have arisen relating to robustness - particularly where large-scale damage to the network is expected from time to time. Specifically, the emergence of so-called coverage holes (CHs), such as that shown in Figure 6.1, which principally result from the correlated en masse failure of nodes due to external catastrophic events such as natural disasters or a hostile attack, is of particular research interest [24]. CHs not only degrade the quality of service (QoS) and traffic balance across different parts of the network, but they can also cause transient failures in routing protocols, resulting in long-term outages in the remaining network. Geographic greedy routing algorithms are particularly susceptible to this problem, since traffic concentrates along the edge of damaged regions, accelerating energy depletion and exhaustion at those nodes. This in turn would increase the size of the CH and exacerbates the damage [24, 25].

For mobile nodes, it may be possible to relocate some subset of nodes in order to repair or shrink CHs [27, 182, 183]. Distributed relocation algorithms have recently become a very active research topic for networks in which sensor nodes can autonomously respond to catastrophic topology changes through deliberate physical movement, with no or minimal external intervention and control [23, 44, 134, 190, 357]. Various (distributed) node relocation algorithms have been devised for WSNs, including systems that exploit network redundancy [358, 359], diverse node deployment models [190, 360], and controlled movement [23, 38, 41, 134].

Key trends in most recent distributed relocation algorithms are reducing information exchange overheads, granting an increased amount of decision-making autonomy to individual nodes, lowering node response times to events, and encouraging emergent cooperative behaviour [49, 361]. In a network exhibiting such behaviour, each node plays its role by autonomously interacting with its surroundings based on its perception of its in-range neighbours; however, at the macro scale, a cooperative and global response is achieved among nodes in the network. In the emergent cooperative recovery of CHs, nodes autonomously relocate according to local interactions with their in-range neighbours, such that they are able to reduce the size of the CHs with little or no communications or explicit coordination amongst themselves. The essence of the proposed cooperative recovery model is that, by the self-organised movements of a set of *disjoint spanned trees* (DS-Trees) (trees $T_i \cup T_j \neq 0,\ \forall (i, j)$ with maximum depth of h_{max}) [362] constructed around and radiating out from each CH, the CH can be wholly or partially repaired. The

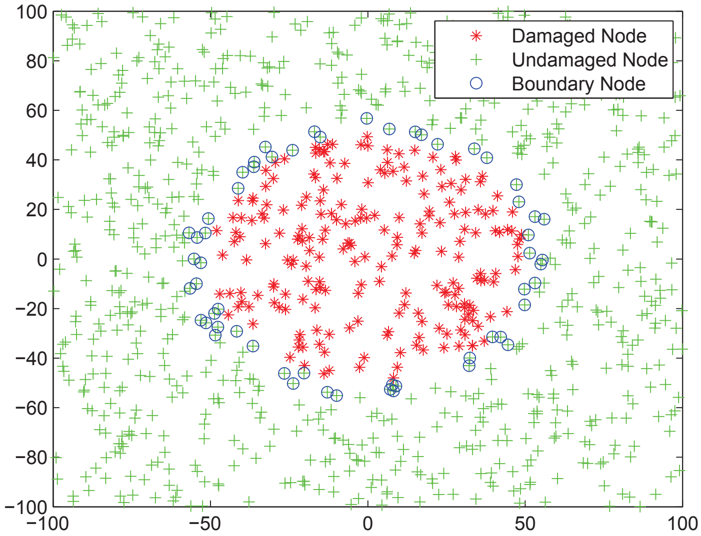
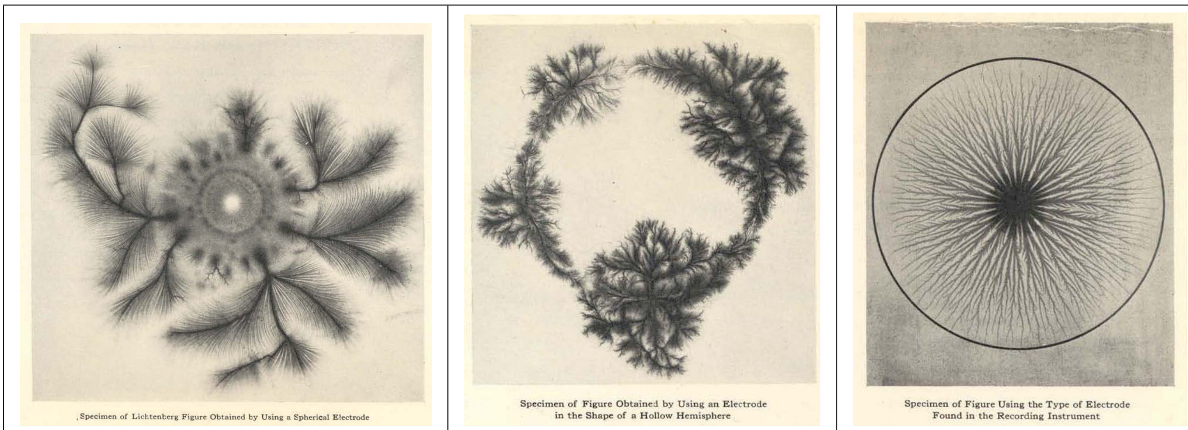


Figure 6.1: Coverage Hole and Types of Nodes

objectives of doing so are as follows:

1. To limit and ameliorate damage-induced congestion [363, 364];
2. To reduce the required amount and range of the distribution of node information during the recovery process; and
3. To decrease the response time to damage events by avoiding the need for centralised coordination and increasing node autonomy.

The proposed idea is inspired by nature, as shown in figures from the work by Lee et al. [365] (Figure 6.2). The circular collections of DS-Trees are formed. The trees are spanned on the basis of the nodes' status in their in-range neighbours' databases, and their distances from CHs via trees that are already spanned from CHs to the in-range neighbours (see Figure 6.3). For the root nodes in DS-Trees, the distance from the CH is defined as the distance to the closest damaged neighbour in the CH (see Figure 6.3). Each node selects its parents from its in-range neighbours



(a) (b) (c)

Figure 6.2: Specimen of lichtenberg Figure obtained by using a spherical electrode [365]

by comparing its distance and its neighbours' distances to the CH via the paths of DS-Trees that have been already spanned from the CH down to the in-range neighbours. Based on the unidirectional parent-child (leader-follower) relation, nodes autonomously decide to join existing DS-Trees by choosing the best potential parent (PP) node; they will subsequently follow their selected parents as their DS-Trees move towards the centre of the CH.

During the parent-selection procedure, each node considers a set of its in-range neighbours as PPs if their cumulative distances from the node to the CH via their associated DS-Trees are less than the node's current perceived distance from the CH. Each node then autonomously selects a candidate parent (CP), which is the PP with the best status amongst the set of PPs. During the spanning process, each node, depending on its own status, decides whether or not to send updated information (such as number of hops or cumulative Euclidean link distance to the CH) to its in-range neighbours.

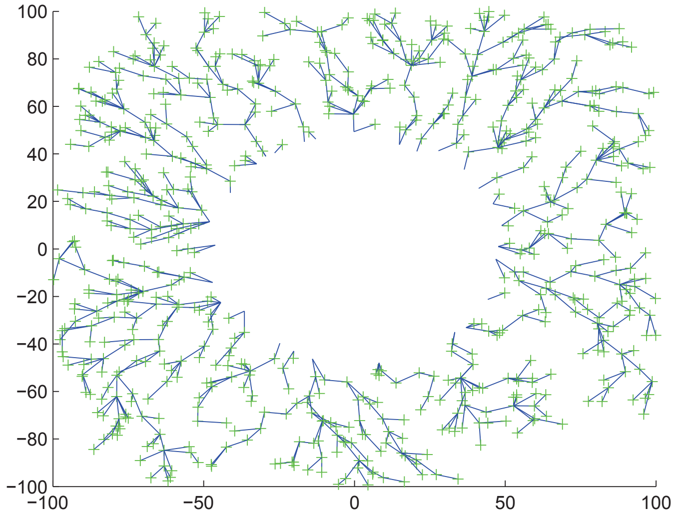


Figure 6.3: Disjoint Spanned Trees (DS-Tree) around CH

The depth of the DS-Trees is limited by *maximum allowed tree depth of notification* (h_{max}). Nodes autonomously decide to forward their updated information after adding their own contribution by comparing their status with those of their in-range neighbours. As nodes follow their candidate parents and autonomously relocate to their new locations, the risk of physical collisions among the nodes reduces; nodes naturally move in a coordinated fashion to close the damaged region of the network.

Based on the location of nodes relative to the damaged area, nodes are classified as either *damaged nodes* (D-nodes) and *undamaged nodes* (U-nodes) similar to Section 3.2.3. U-nodes that are located at the margin of CHs and are able to detect the damage event within their neighbourhood are known as *Boundary-nodes* (B-nodes). B-nodes autonomously notify their immediate neighbours and propagate their knowledge about the event to these nodes. The information forwarded includes nodes' distance from the CH boundary (Euclidean and hop-count) and the potential distance that the node may move towards the CH. This information is thus propagated along the branches of the DS-Trees from the root to the leaves. In the DS-trees, for status updates, event notification and movement decisions, nodes only consider their in-range neighbours - especially candidate parents - and the given depth limit h_{max}. It should be noted that not all DS-Trees are spanned to the given h_{max} due to node location and CH location, geometry and dimensions (Figures 6.1 and 6.3).

The DS-Tree CH recovery model not only limits the amount of information exchange needed between the nodes, but also results in chain-style node movements towards the CHs such that the distance of each moving node in the DS-Tree decreases with increasing distance from the CH. CH Recovery using collective movements of DS-Trees toward the damaged area shows an emergent cooperative behaviour. The performance of the proposed model is compared with two Voronoi-based relocation algorithms and a force-based relocation algorithm [227, 231, 232] in terms of coverage, uniformity and movement.

6.2 Method and Assumptions

6.2.1 Sensor Nodes

Each node is modelled as a unit disk graph (UDG) with a transmission range of R_c and sensing range of R_s [348]. For the sake of simplicity, it is assumed that $R_c = R_s$. A pair of deployed nodes are bidirectionally connected if the centre of one node's disc is covered by the disc of the other node. Nodes are distributed with a 2-D uniform random distribution in a rectangular area of $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. Each node is aware of its own location via GPS or other localisation methods [101, 349, 350].

6.2.2 Coverage Hole

Coverage hole k can be considered as a circle of a radius r_{Hole_k} with centre of $(x_{Hole_k},\ y_{Hole_k})$ (Figure 6.1). This circle represents an occurrence of events in which node failures start at the centre and may expand symmetrically from this location due to some physical damage (e.g. an explosion). By combining multiple CH circles with different centres and radii, any arbitrarily complex CH with a convex or non-convex shape can be approximated [273]. Other alternative forms of CHs are briefly mentioned in [366].

6.2.3 Node Types

In general, a set of nodes of type α is defined as $\phi_\alpha = \{S_\alpha(1), \cdots, S_\alpha(N_\alpha)\}$, where $|\phi_\alpha| = N_\alpha$. Depending on the scale of the damage and the nodes' locations relative to the CH, similar to Section 3.2.3, deployed nodes ($|\phi| = N$) are classified as damaged nodes (D-nodes) $S_d(i)$ ($\phi_d = \{S_d(1), \cdots, S_d(N_d)\}$), which are not operational, and undamaged nodes (U-nodes) $S_u(i)$ ($\phi_u = \{S_u(1), \cdots, S_u(N_u)\}$) where $|\phi| = |\phi_d| + |\phi_u| = N_d + N_u = N$. D-nodes are the set of nodes which reside inside the damaged area. The set of operational nodes in the area of deployment are therefore U-nodes, as described by equation 6.1. It is assumed that nodes can individually detect a damage event if at least one D-node is within their ranges. Thus, the D-nodes and U-nodes are defined as follows:

$$\begin{cases} S_d(i), \quad if \ \ \left\|p_{S_n(i)} - p_{Hole_k}\right\| \leq r_{Hole_k}, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall(i,k) \\ S_u(i), \quad if \ \ \left\|p_{S_n(i)} - p_{Hole_k}\right\| > r_{Hole_k}, \end{cases} \tag{6.1}$$

where $p_{S_\alpha(i)}$ is the location $(x_{S_\alpha(i)}, y_{S_\alpha(i)})$ of node $S_\alpha(i)$ of type α in the given 2D region of deployment. Depending on their location relative to the CH, U-nodes are further classified as *boundary nodes* (B-nodes) $S_b(i)$ ($\phi_b = \{S_b(1), \cdots, S_b(N_b)\}$)

where $\phi_b \subseteq \phi_u$ and *Normal nodes* (N-nodes), $S_n(i)$ ($\phi_n = \{S_n(1), \cdots, S_n(N_n)\}$) such that $\phi_n \subseteq \phi_u$, $\phi_n = \phi_u \backslash \phi_b$, and $|\phi_u| = |\phi_n| + |\phi_b| = N_b + N_n = N_u$ (Figure 6.1). B-nodes are a subset of the set of U-nodes which reside on the margin of the CH as they detect the presence of the damage event within their ranges.

6.2.4 Voronoi Cells

It is assumed that each node is aware of its Voronoi diagram prior to the damage event. After the damage event, similar to classification in Sections 3.2.4.2 and 6.2.3, Voronoi cells are classified into those containing D-nodes, denoted *damaged Voronoi cells* (DV-Cells), $C^v_{S_d(i)}$, and those containing U-nodes, denoted *undamaged Voronoi cells* (UV-Cells), $C^v_{S_u(i)}$. UV-cells can be further classified as *normal Voronoi cells* (NV-cells), $C^v_{S_n(i)}$ and *boundary Voronoi cells* (BV-cells), $C^v_{S_b(i)}$. BV-cells are those UV-cells that have at least one of their Voronoi cell edges ($E(C^v_{S_u(.)})$) in common with the DV-cells ($E(C^v_{S_d(.)})$)(as described in Equation 6.2).

$$\begin{cases} C^v_{S_n(i)} \quad if \ E(C^v_{S_u(i)}) \cap E(C^v_{S_d(j)}) = 0, \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall (i,j) \\ C^v_{S_b(i)}, \quad if \ E(C^v_{S_u(i)}) \cap E(C^v_{S_d(j)}) \neq 0, \end{cases} \tag{6.2}$$

where $E(C^v_{S_u(i)})$ and $E(C^v_{S_d(j)})$ are respectively edges of UV-cell i ($C^v_{S_u(i)}$) and DV-cell j,($C^v_{S_d(j)}$). If a node is in a BV-cell, it is a B-node; however, not all B-nodes necessarily reside in BV-cells. There are the cases in which Voronoi cells of B-nodes do not have common edges with DV-cells due to the density and distribution of node deployment, or the shape of the CH.

For block diagram of the CH boundary detection algorithm, please refer to Figure 6.4.

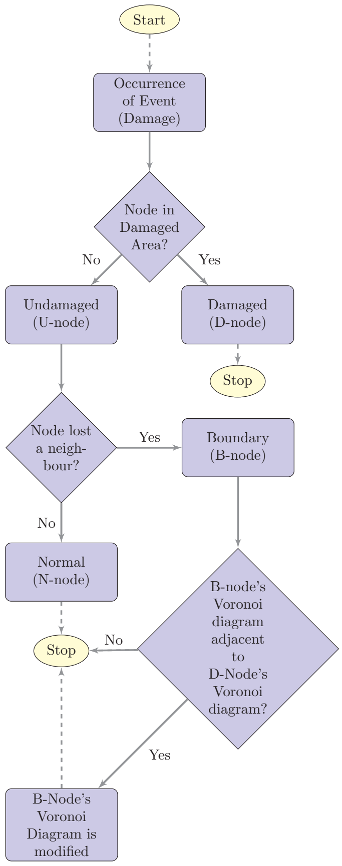


Figure 6.4: Block Diagram of the CH boundary detection algorithm and Node types

6.2.5 Boundary Conditions

The coordinates of the vertices of any Voronoi cells that reside outside of the boundary of the deployment area are clipped to x_{min},x_{Max},y_{min}, y_{Max}. Similarly, if nodes relocate outside of the given deployment area boundary, their updated locations are modified which nodes stop at the boundaries of deployment area.

6.2.6 Disjoint Spanned Tree

6.2.6.1 Tree and Graph

The network structure immediately before and after the damage event can be represented as the graph of $G(V, E)$ and $\acute{G}(V_u, E_u)$ $(V_u = V - V_d,\ E_u = E - E_d)$ where V_d is the set of failed nodes and E_d is the set of disconnected edges. Suppose that the set of sub-graphs $H_1, \cdots, H_n \subseteq \acute{G}$, where $H_i \cap H_j \neq \Phi$, for $\exists(i, j)$ (Φ : *Empty set*) are defined and nodes of $H_i(V_u, E_u)$ are spanned by a tree T_i where $|T_i(v - U)| \leq |H_i(v_u)|$, $|T_i(E_u)| \leq |H_i(E_u)|$.

Definition 1 *Spanned trees T_i and T_j are considered to be* mutually disjoint trees *if they have no common edges or nodes. Thus, the trees are both* node-disjoint *and also* edge-disjoint*.*

Definition 2 *Trees, $\{T_1, \cdots, T_i, \cdots, T_j, \cdots, T_n\}$, are node-disjoint if $V_u(T_i) \cap V_u(T_j) = \Phi,\ \forall(i, j)$ (where $V_u(T_k)$ is the tree T_k's nodes (vertices)).*

Definition 3 *Trees, $\{T_1, \cdots, T_i, \cdots, T_j, \cdots, T_n\}$, are* edge-disjoint *if $E_u(T_i) \cap E_u(T_j) = \Phi,\ \forall(i, j)$ (where $E_u(T_k)$ is the tree T_k's edges). In the proposed model, trees are node and edge disjoint.*

Definition 4 *The* depth *or height of a node in a tree is defined as the length of the path from the given node to its root. Parent nodes have less depth than their children. Nodes are considered to be on the same level (L_i) if they have equal path lengths to the root nodes of their trees [362].*

6.2.6.2 Recovery Stages

In this section, the proposed cooperative recovery of CHs is described. The CH recovery model utilises the proposed *DS-Tree movement algorithm* shown in Algorithm 6.1. In the detection stage, damage events are detected by the CHs' B-nodes. BV-cells are located with respect to the scale of CHs and B-nodes. Those B-nodes

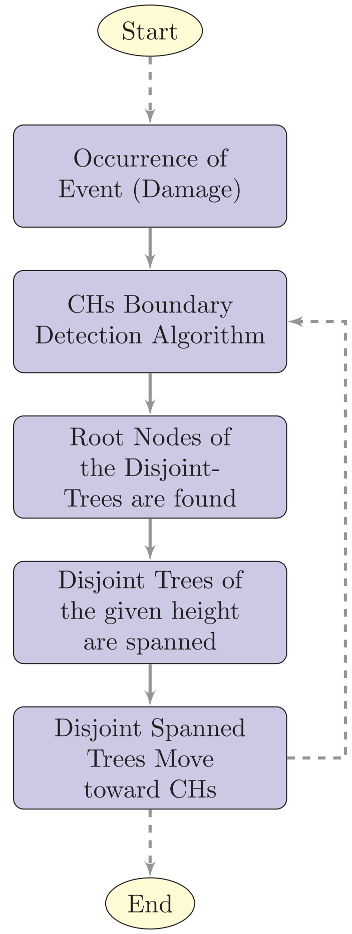


Figure 6.5: Block Diagram for Recovery Stages of CH

whose Voronoi cells have the common edge with DV-Cells are considered to be BV-Cells. Root nodes (R-nodes) of the trees are found according to Algorithm 6.2.

In the CHs' notification and identification phase, DS-trees are formed based on Algorithm 6.3. Finally, the DS-trees move towards the centre of the CH according to Algorithm 6.4. (Please refer to the diagram in Figure 6.5)

ALGORITHM 6.1: DS-Tree Movement Algorithm

Input:
N_α: Total number of type α nodes
Set of U-Nodes $S_u(i)$, $\phi_u = \{S_u(1), \cdots, S_u(N_u)\}$
Set of B-Nodes $S_b(i)$, $\phi_b = \{S_b(1), \cdots, S_b(N_b)\}$
Set of D-Nodes $S_d(i)$, $\phi_d = \{S_d(1), \cdots, S_d(N_d)\}$
$C(S_\alpha(i))^v$: Voronoi cells of Node $S_\alpha(i)$

Output: Movement of DS-Trees T_i, T_j where $T_i \cup T_j \neq 0$ for $\forall(i, j)$

```
if (∃S_α(i) ∈ φ_d) then
    foreach (S_b(i) ∈ φ_b and S_d(j) ∈ φ_d) do
        if C(S_b(i))^v ∩ C(S_d(j))^v ≠ Φ then
            C(S_b(i))^v ← (C(S_b(i))^v)_new
        end
    end
end
Find R-nodes by Alg. 6.2
Find DS-Trees T_i, T_j (T_i ∪ T_j ≠ 0, ∀(i, j)) by Alg. 6.3
Move DS-Trees T_i, T_j (T_i ∪ T_j ≠ 0, ∀(i, j)) by Alg. 6.4
```

6.2.6.3 Root Node Selection (Algorithm 6.2)

As described in Section 6.2.3, nodes are categorised as U-nodes (B-nodes and N-nodes) and D-nodes. B-nodes detect the damage event by observing an abrupt loss of neighbours in their neighbour table, and autonomously consider themselves as the R-nodes according to Algorithm 6.2. Regardless of the type of neighbour, they are defined generally as $\beta-$*neighbours*.

To harness the redundancy of deployed nodes and determine an appropriate number of DS-trees spanning from the CH, a network-wide probability threshold $\tau_{root} \in [0, 1]$ is defined, which represents the probability threshold that a B-node will exclude itself from being a potential R-node. That is, if $\tau_{root} = 0$, all B-nodes self-select as potential R-nodes. If $\tau_{root} = 1$, none of the B-nodes will consider

ALGORITHM 6.2: Root Nodes Selection Algorithm

Input:
Set of U-Nodes $S_u(i)$, $\phi_u = \{S_u(1), \cdots, S_u(N_u)\}$
Set of B-Nodes $S_b(i)$, $\phi_b = \{S_b(1), \cdots, S_b(N_b)\}$
Set of D-Nodes $S_d(i)$, $\phi_d = \{S_d(1), \cdots, S_d(N_d)\}$
$S^{\beta}_{\alpha}(i)$, Set of β-neighbours of $S_{\alpha}(i)$, $\left|S^{\beta}_{\alpha}(i)\right| \geq 0$
$d^{\beta}_{\alpha}(i)$, Set of $S_{\alpha}(i)$ euclidean distances to $S^{\beta}_{\alpha}(i)$
$\overrightarrow{x^{\beta}_{\alpha}}(i)$, Set of Vectors from $S_{\alpha}(i)$ to $S^{\beta}_{\alpha}(i)$
$CM^{\varphi}_{S_{\alpha}(i)}(\overrightarrow{v_{\alpha}}(i), \omega_{\alpha}(i))$, Center of Mass of vector $\overrightarrow{v_{\alpha}}(i)$
& weights of $\omega_{\alpha}(i)$ from $S_{\alpha}(i)$ to set of points $\in \varphi$
$Send_{S_{\alpha}(i)}$, Msg Sent by $S_{\alpha}(i)$ to its neighbours
$Recv_{S_{\alpha}(i)}$, Msg Received by $S_{\alpha}(i)$ from its neighbours
$Parent_{S_{\alpha}(i)}$, Set of $S_{\alpha}(i)$'s Parents in the Tree T_j
$Parent^{*}_{S_{\alpha}(i)}$, Selected Parent from $Parent_{S_{\alpha}(i)}$
$S_{\alpha}(i).Root$ is 1 if Node $S_{\alpha}(i)$ is R-Nodes

Output:
Set of R-Nodes $S_r(i)$, $\phi_r = \{S_r(1), \cdots, S_r(N_r)\}$
when $\alpha \leftarrow b$, and $\exists S_{\alpha} \in \phi_b$ s.t. $S_{\alpha}(i).Root \leftarrow 1$

foreach $S_{\alpha}(i)$ *in* ϕ_b, $C(S_b(i))^v$ **do**
  Calculate rand $p \sim U[0, 1]$
  if $p > \tau_{root}$ **then**
    $S_{\alpha}(i).Root \leftarrow 1$
    $Send_{S_{\alpha}(i)}$
    $Parent^{*}_{S_{\alpha}(i)} \leftarrow CM^{\phi_d}_{S_{\alpha}(i)}(\overrightarrow{x^{d}_{\alpha}}(i), d^{d}_{\alpha}(i))$
  else
    $S_{\alpha}(i).Root \leftarrow 0$
    $Recv_{S_{\alpha}(i)}$
    $Parent^{*}_{S_{\alpha}(i)} \leftarrow \arg_{\gamma} \min d^{\gamma}_{\alpha}$
    $= \{\gamma | \forall \gamma, \beta \in \phi_b \cap S^{\beta}_{\alpha} : d^{\beta}_{\alpha} \geq d^{\gamma}_{\alpha}\}$
  end
end

themselves as potential R-nodes and none of the B-nodes will send information to their in-range neighbours. Thus, by adjusting τ_{root}, the proportion of B-nodes that self-select as R-nodes (and hence the number of DS-Trees) can be tuned.

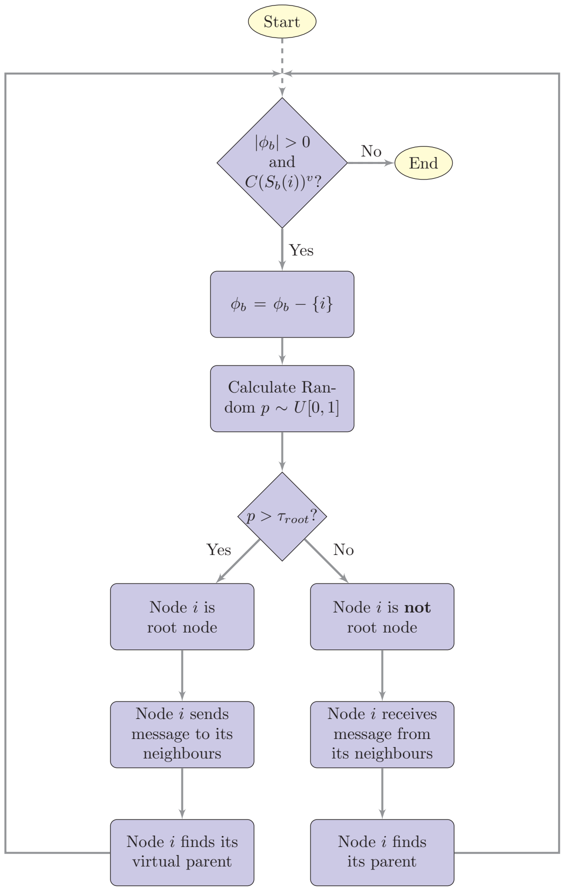


Figure 6.6: Block Diagram of the Root Node Selection Algorithm

If a node autonomously decides that is *not* a potential R-node, it waits to receive information from its in-range undamaged neighbours, to which it adds its own contribution before forwarding the aggregated information to its neighbours. Throughout this chapter, τ_{root} is assumed to be zero (that is, all B-nodes are also potential R-nodes). The threshold determines the probability of an individual node's willingness to act in the role of R-node based on various factors such as the amount of energy remaining in that particular node's battery. Each R-node, $S_r(i)$ considers the *virtual node* obtained from the centre of mass of its D-node neighbours, $CM^{\phi_d}_{S_r(i)}(\overrightarrow{x^d_r}(i), d^d_r(i))$, as its parents (Algorithm 6.2).

ALGORITHM 6.3: Disjoint Spanned Trees Algorithm

Input: Set of U-Nodes $S_u(i)$, $\phi_u = \{S_u(1), \cdots, S_u(N_u)\}$
Set of B-Nodes $S_b(i)$, $\phi_b = \{S_b(1), \cdots, S_b(N_b)\}$
Set of D-Nodes $S_d(i)$, $\phi_d = \{S_d(1), \cdots, S_d(N_d)\}$
$S_\alpha^\beta(i)$, Set of β-neighbours of $S_\alpha(i)$, $\left|S_\alpha^\beta(i)\right| \geq 0$
$d_\alpha^\beta(i)$, Set of $S_\alpha(i)$ euclidean distances to its $S_\alpha^\beta(i)$
$CM_{S_\alpha(i)}^{\varphi}(\overrightarrow{v_\alpha}(i), \omega_\alpha(i))$, Center of Mass of vector $\overrightarrow{v_\alpha}(i)$ & weights of $\omega_\alpha(i)$ from $S_\alpha(i)$ to set of points $\in \varphi$
$Send_{S_\alpha(i)}$, Msg Sent by $S_\alpha(i)$ to $S_\alpha^\beta(i)$
$Recv_{S_\alpha(i)}$, Msg Received by $S_\alpha(i)$ from $S_\alpha^\beta(i)$
$Parent_{S_\alpha(i)}$, Set of $S_\alpha(i)$'s Parents in the Tree T_j
$Parent^*_{S_\alpha(i)}$, Selected Parent from $Parent_{S_\alpha(i)}$
h_{max}, Maximum Height(depth) of Tree
$S_\alpha.d_{Event}$, Perceived Distance of $S_\alpha(i)$ to Event via intermediate nodes
$S_\alpha.h_{Event}$, Perceived Hop-count of $S_\alpha(i)$ to Event via intermediate nodes
$S_\alpha(i).d^*_{Event}$, Min($S_\alpha(i).d_{Event}$),when $\exists k$, s.t. $S_\alpha(i) \in T_k*$
$S_\alpha(i).h^*_{Event}$, Min($S_\alpha(i).h_{Event}$),when $\exists k$, s.t. $S_\alpha(i) \in T_k*$
Output: DS-Trees $T_i, T_j$where $T_i \cap T_j = \Phi$ for $\forall(i,j)$

while $S_\alpha(i).d^*_{Event}$ *and* $S_\alpha(i).h^*_{Event}$ $(\forall i)$ **do**
  foreach $(S_\alpha(i) \in \phi_u)$ **do**
    if $S_\alpha^\beta(i) \neq \Phi$ **then**
      if $S_\alpha.d_{Event} \geq S_j.d_{Event} + d_\alpha^\beta(i)$
      and $S_\alpha.h_{Event} \geq S_j.h_{Event} + 1$
      $(\forall j \in S_\alpha^\beta(i))$
      then
        $Recv_{S_\alpha(i)}$ *from* $S_\alpha^\beta(i)$
        Calculate $p_\alpha(i).d \leftarrow Parent^*_{S_\alpha(i)}.d_{Event}$
        Calculate $p_\alpha(i).h \leftarrow Parent^*_{S_\alpha(i)}.h_{Event}$
        Update the Information
        $S_\alpha.d_{Event} \leftarrow p_\alpha(i).d + d_\alpha^{Parent^*_{S_\alpha(i)}}(i)$
        $S_\alpha.h_{Event} \leftarrow p_\alpha(i).h + 1$
        if $S_\alpha.h_{Event} \leq h_{max}$ **then**
          $Send_{S_\alpha(i)} from S_\alpha^\beta(i)$
        end
      else
        if $S_\alpha.h_{Event} \leq h_{max}$ **then**
          $Send_{S_\alpha(i)}$ *to* $S_\alpha^\beta(i)$
        end
      end
    else
      if $S_\alpha(i) \in \phi_b$ *and* $\exists S_\alpha^\beta \in \phi_d$
      then
        $S_\alpha.h_{Event} \leftarrow 1$
        $S_\alpha.d_{Event} \leftarrow CM_{S_\alpha(i)}^{\phi_d}(\overrightarrow{x_\alpha^d(i)}, d_\alpha^d(i))$
      end
    end
  end
end

6.2.6.4 Damage Event Notification

Knowledge of damage events progressively propagates through the network as nodes exchange information regarding nearby CHs with their in-range neighbours. Nodes compare their current knowledge with data received from their neighbours, and update their model of the network state by adding any new or updated information received from the neighbouring nodes. Nodes then autonomously decide whether or not to forward the state update to their immediate neighbours (as described in Algorithm 6.3). Please refer to Figure 6.6 for Root node selection diagram.

6.2.6.5 Parent Selection

During data exchange with their in-range neighbours, nodes autonomously select their parents based on a unidirectional parent-child relation. Each node obtains a list of parents from its neighbours with a ranking in terms of hop-counts and distances from the CH (as in Equations 6.3 and 6.4). Each node $S_\alpha(i)$ is aware of its in-range neighbours and listens for their broadcasts.

$$S_\alpha(i).d_{Event}^{New} = min\left(S_\alpha(i).d_{Event}^{Current}, (Parent^*_{S_\alpha(i)}.d_{Event} + d_\alpha^{Parent^*_{S_\alpha(i)}})\right) \tag{6.3}$$

$$S_\alpha(i).h_{Event}^{New} = min\left(S_\alpha(i).h_{Event}^{Current}, (Parent^*_{S_\alpha(i)}.h_{Event} + 1)\right) \tag{6.4}$$

Upon receiving new or updated information from its neighbours, a node updates its database and then autonomously decides to broadcast its knowledge to its neighbours (Equations 6.3 and 6.4).

In the parent-child relation, nodes only move based on the location and movements of their parent nodes; they are not aware of the existence of any potential children and their movements are not influenced by them. Each node autonomously selects its candidate parent from the list of potential parents and joins the tree that spans from the CH down to the node's candidate parent. It then decides to what extent it will follow its candidate parent (discussed in detail in Section 6.2.6.6).

Nodes' autonomous decisions are indirectly affected by their ancestors as information propagates radially from the R-nodes down to the leaf nodes of a given DS-Tree. If a node has more than one candidate parent with exactly the same rank in its list, one will be randomly selected (with equal probability) to become the node's parent. In the course of tree spanning, each node checks to ensure that the maximum allowed tree depth limit h_{max} will not be exceeded before sending its updated status to its neighbours. Unidirectional parent-child relation and h_{max} limit the scope and number of notification messages, and consequently the size of the DS-trees formed around the CHs (Algorithm 6.3), which also limits the number of nodes participating in recovery movements. Please see Figure 6.7 for the block diagram of the DS-Tree algorithms.

6.2.6.6 Disjoint Spanned Tree Movement Model

Two relocation algorithms are proposed, denoted *DS-tree* and *DS-Tree Random* (DS-Tree RD), the latter of which allows node movements to be tuned by a uniformly distributed *random multiplier* of q ($q \sim U[0, 1]$) (Algorithm 6.4). The movement tendency of autonomous nodes in DS-Trees is governed by a per-node probability threshold τ_{act}. If the random variable p ($p \sim U[0, 1]$) exceeds τ_{mov}, the node will self-select for potential movement, subject to having sufficient energy reserves. The movement of each DS-Tree depends on the movement of its R-node. Accordingly, the movement of the given nodes and their children throughout each DS-Tree is then affected iteratively according to Equation 6.5. In order to tune node movements, each node is allowed to autonomously choose either the DS-Tree or DS-Tree RD algorithm at the time of its movement by introducing the probability threshold τ_{mov}. As a result, nodes can randomly select one of the given

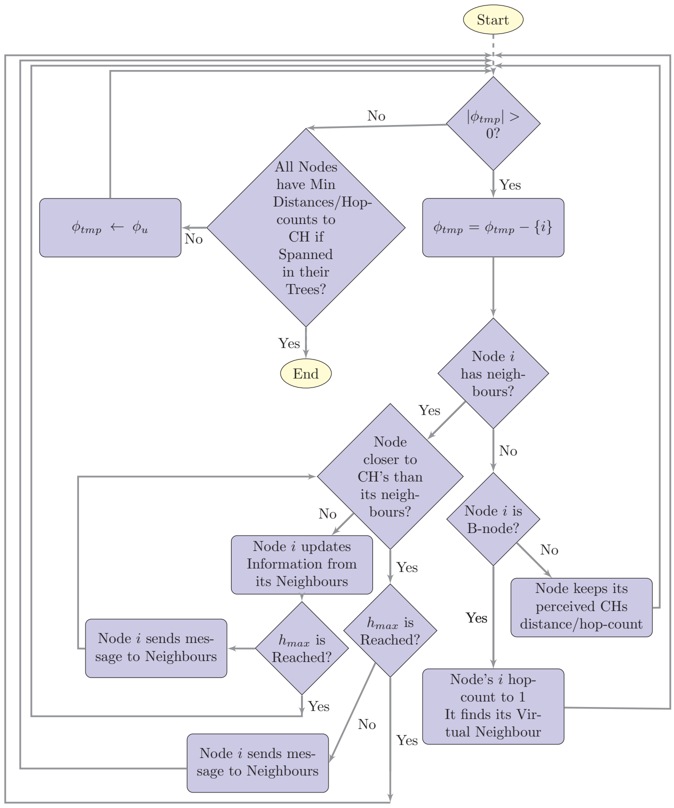


Figure 6.7: Block Diagram of the Disjoint Spanned Trees Algorithm

algorithms depending on whether $q \sim U[0,1]$ exceeds τ_{mov}. By setting $\tau_{mov} = 1$ or $\tau_{mov} = 0$, nodes relocate according to either the DS-tree or DS-Tree RD movement algorithm respectively. The performance of proposed algorithms for $\tau_{mov} = 1$ (strictly DS-tree) and $\tau_{mov} = 0$ (strictly DS-tree RD) are compared in this chapter.

$$\begin{cases} q \cdot \mu_0(S_\alpha(i), j), & if\ S_\alpha(i) \in \phi_r, \\ & \forall(i,j) \\ q \cdot (M((h_{(S_\alpha(i),j)} - 1), k) \cdot h_{max}^{1/h_{(S_\alpha(i),j)}}), & if\ S_\alpha(i) \in \phi_u \backslash \phi_r, \end{cases} \tag{6.5}$$

Although nodes follow their immediate parents in their DS-trees, they are also indirectly affected by their higher-level ancestors. Thus, to gradually reduce such effects and to control the nodes' movements, a depth limit h_{max} is introduced to

ALGORITHM 6.4: Chain Node Movement DS-Tree Alg.

Input:
Set of R-Nodes $S_r(i)$, $\phi_r = \{S_r(1), \cdots, S_r(N_r)\}$
Set of U-Nodes $S_u(i)$, $\phi_u = \{S_u(1), \cdots, S_u(N_u)\}$
Set of DS-Trees T_i, T_j where $T_i \cap T_j = \Phi$ for $\forall(i, j)$
h_{max}, tree depth limit threshold
$h_{(S_\alpha(i),j)}$, hop-count of $S_\alpha(i) \in T_j$ from the given CH
$M(h_{(S_\alpha(i),j)}, k)$, Movement of $S_\alpha(i) \in T_j$ with hop-count of $h_{(S_\alpha(i),j)}$ to CH k
$M(i, 0, k)$, Movement of R-node $S_\alpha(i) \in T_j$ to CH k
$\mu_0(S_\alpha(i), j)$, Initial Movement of R-node $S_\alpha(i) \in T_j$

Output: Movement of DS-Trees T_i, T_j $\forall(i, j)$

```
foreach T_j ∈ χ do
    foreach S_α(i) ∈ φ_u do
        Calculate rand p ~ U[0,1]
        if p ≥ τ_act then
            Calculate rand q ~ U[0,1]
            if S_α(i) ∈ φ_r then
                if q ≥ τ_Mov then
                    M(i,0,k) ← q · μ_0(S_α(i), j)
                else
                    M(i,0,k) ← μ_0(S_α(i), j)
                end
            else
                if q ≥ τ_Mov then
                    M(h_(S_α(i),j), k) ← q · (M((h_(S_α(i),j) − 1), k) · h_max^(1/h_(S_α(i),j)))
                else
                    M(h_(S_α(i),j), k) ← (M((h_(S_α(i),j) − 1), k) · h_max^(1/h_(S_α(i),j)))
                end
            end
        end
    end
end
```

limit the growth of DS-trees from the R-nodes (see Equation 6.5). Depending on the size of the damaged area and the density of nodes, there is a possibility of physical collision among the nodes (especially R-nodes) due to the autonomous movements towards the CH. Therefore, by applying MinMax-Voronoi algorithm to R-nodes, their initial movements $(q \cdot \mu_0(S_\alpha(i), j))$ are limited.

The movements of the remaining nodes are then iteratively tuned along their DS-trees according to Algorithm 6.4. By limiting the nodes' movements around CHs, the risk of disturbance to node distribution and formation of multiple small CHs in the network is also reduced. Iterative chain movement of nodes are attenuated according to Algorithm 6.5 in order to reduce the chance of the formation of small new coverage holes and collision among the nodes in the DS-Trees.

It should be noted that if the proposed algorithms apply to multiple non-overlapped CHs, each DS-tree is associated with only one CH. For example, suppose node $S_\alpha(i)$ is located among a set of CHs $\xi = \{k_1, \cdots, k_l\}$, $(l \geq 2)$ and has respective hop-counts and Euclidean distances of $d_{(S_\alpha(i),\kappa,\xi)}$ and $h_{(S_\alpha(i),\kappa,\xi)}$, $(\kappa \geq 2)$ from them through potential DS-trees T_κ. Then, node $S_\alpha(i)$ joins tree T^*, which has the best score ($\arg_\kappa \min[\{h_{(S_\alpha(i),\kappa,\xi)}\}$ and/or $\{d_{(S_\alpha(i),\kappa,\xi)}\}$ to one of CHs (see Equations 6.3, 6.4 and Algorithm 6.3). For the sake of brevity, only the case of one large scale CH is considered in chapter.

6.3 Performance Evaluation

In the following sections, the performance metrics in terms of percentage of 1-Coverage, uniformity, average movement, [227] and efficiency of movement, and the results are presented.

6.3.1 Performance Metrics

Percentage of Coverage, similar to the definition in Section 3.3.1.2, is considered as a measure of the performance for relocation algorithms as the proportion of the area that the nodes cover, changes due to their movements.

Uniformity, as a measure of node distribution, is defined as the average local standard deviation of inter-nodal distances, as in Section 5.4.1.

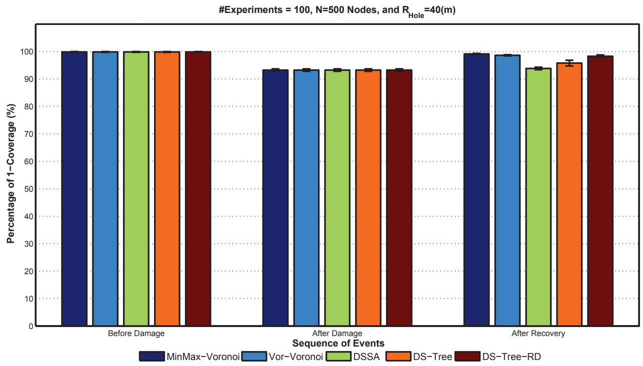
Average Movement, similar to the definition in Section 5.4.1, is used to assess the mobile nodes' consumed energy, as the majority of node's energy is dedicated to its physical movements. Average movement in each movement iteration is presented as a ratio of the total amount of nodes movements to total number of nodes [227].

Efficiency of Movement includes both factors of coverage and movement. Achieving the desired coverage by relocation is done at the price of excessive nodes movements. Thus, considering both performance metrics at once is sensible, should the performance of relocation algorithm be evaluated. Therefore, efficiency of movement in each iteration, is defined as the ratio of coverage change to the average movement of the total nodes in the given iteration (Equation 6.6).

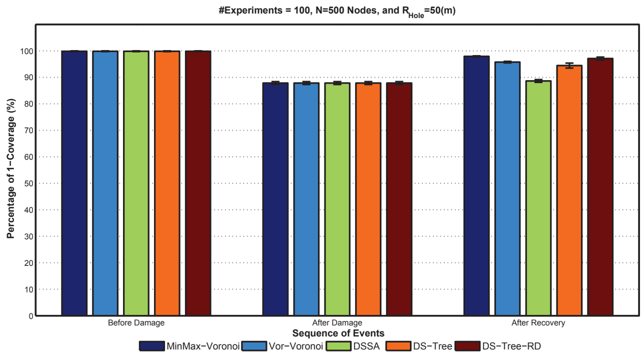
$$M_F = \frac{C_i - C_{i-1}}{M_i} \tag{6.6}$$

6.3.2 Results

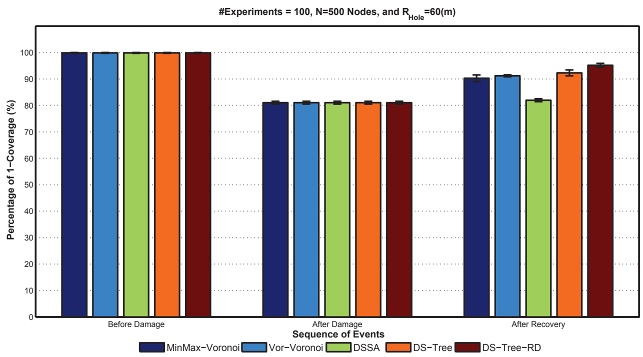
The proposed model were simulated by Matlab. Nodes with the transmission and sensing range of $10m$ were randomly deployed with uniform distribution in a 2-D rectangular area of $[-100, 100] \times [-100, 100]$ m^2. Results are presented in parts one and two (Sections 6.3.2.1 and 6.3.2.2). The course of events are either explicitly labelled by their names in the figures (e.g. Figure 6.8) or by the interval of numbers on the horizontal axis (e.g. Figure 6.13). Time intervals of t=0, and t>0, respectively represent the time of the damage event and after recovery of



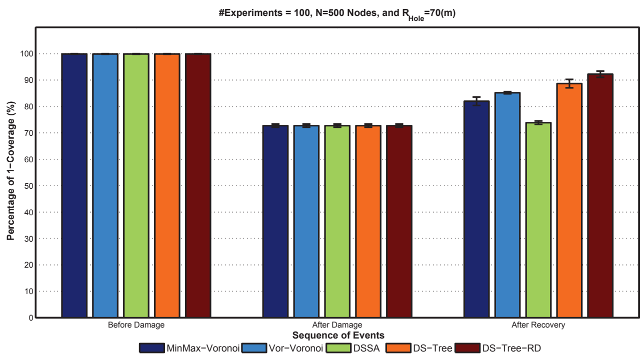
(a) $N = 500$, $R_{Hole} = 40$



(b) $N = 500$, $R_{Hole} = 50$

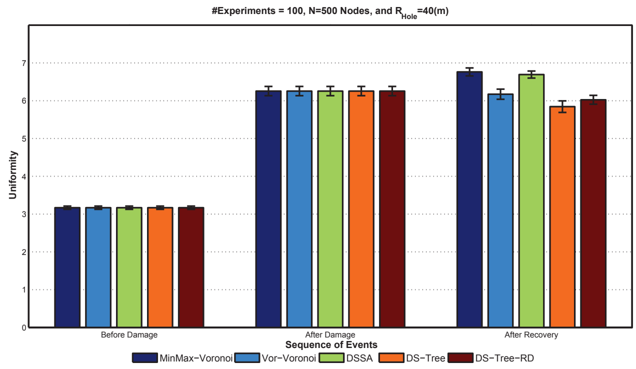


(c) $N = 500$, $R_{Hole} = 60$

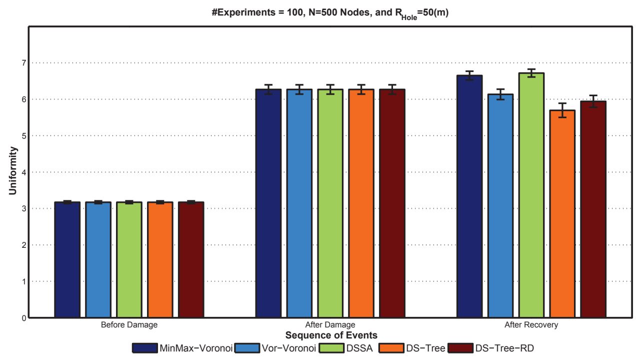


(d) $N = 500$, $R_{Hole} = 70$

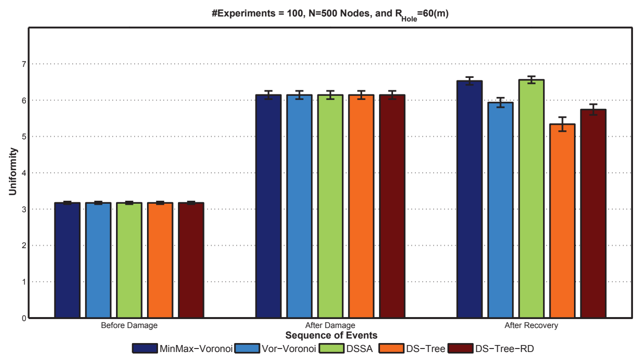
Figure 6.8: Percentage of 1-Coverage of Relocation Algs. for N=500, and radii of R_{Hole}=$(40, 50, 60, 70)m$



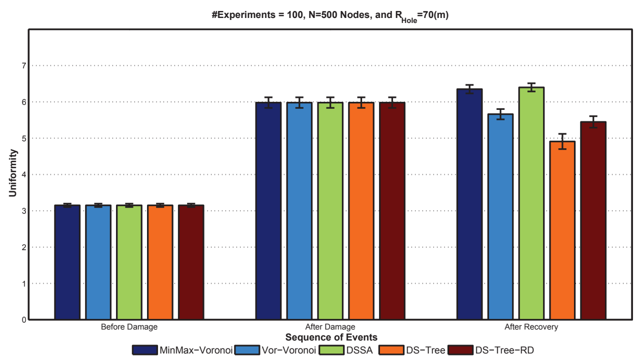
(a) $N = 500,\ R_{Hole} = 40$



(b) $N = 500,\ R_{Hole} = 50$



(c) $N = 500,\ R_{Hole} = 60$



(d) $N = 500,\ R_{Hole} = 70$

Figure 6.9: Uniformity of Relocation Algs. for N=500, and radii of $R_{Hole} = (40, 50, 60, 70)m$

damage.

6.3.2.1 Part One

In this section, the performance of different relocation algorithms in terms of coverage (Section 6.3.2.1.1), uniformity (Section 6.3.2.1.1), average and efficiency of movement (Sections 6.3.2.1.3 and 6.3.2.1.4) through the course of events (before, at, and after the damage event) for a network of N=500 nodes and CHs of R_{Hole}= $(40, 50, 60, 70)m$ are presented in Figures 6.8, 6.9, 6.10 and 6.11 and in Tables 6.1, 6.2, and 6.3. Results regarding network of N=1000 deployed nodes are only presented in Tables 6.1, 6.2, and 6.3 for sake of brevity.

6.3.2.1.1 Coverage

The percentage of 1-coverage of relocation algorithms for the network of N=500 deployed nodes is shown in Figure 6.8 and Table 6.1. According to Figure 6.8 and Table 6.1 for CHs of R_{Hole}=$(40, 50)m$, MinMax-Voronoi and Vor-Voronoi outperformed the rest of the relocation algorithms. MinMax-Voronoi performed better than the Vor-Voronoi algorithm. DS-Tree RD and DS-Tree followed the Voronoi algorithms. DSSA had the worst performance. For CHs R_{Hole}=$(60, 70)m$, Voronoi algorithms performance deteriorated. DS-Tree RD and DS-Tree performances outperform other algorithms. For R_{Hole}=$(60, 70)m$, Vor-Voronoi performs better than MinMax-Voronoi Algorithm. The performance of DSSA is lower than the other algorithms.

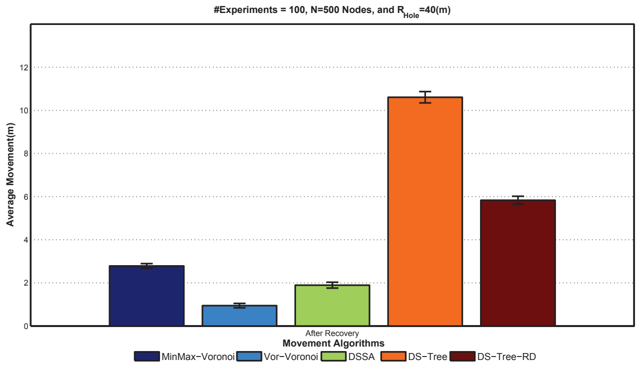
As the size of the CH grew, the coverage performance of MinMax-Voronoi and Vor-Voronoi worsened. However, DS-Tree RD's and DS-Tree's performances did not decrease the same pace as the Voronoi algorithms. DSSA was not able to repair the CHs, especially LSCHs. The performance of DSSA in term of 1-coverage is illustrated in the Appendix B. For the network of N=1000 nodes, the effect of CHs reduces but the performances of different relocation algorithms were similar to the performance of network of N=500 nodes (Table 6.1).

6.3.2.1.2 Uniformity

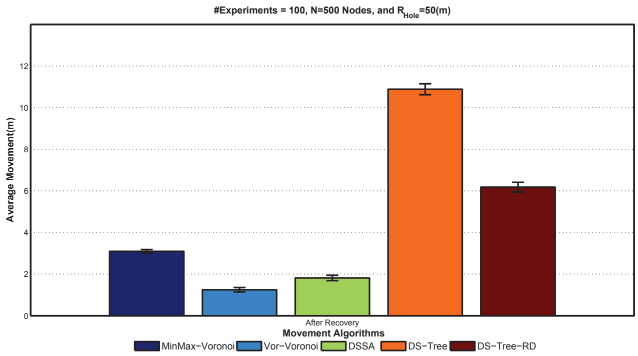
Uniformity after the damage event was calculated based on the U-nodes. According to Figure 6.9 and Table 6.2, with the network of N=500 nodes, MinMax-Voronoi and DSSA had the worst performances for different sizes of CHs (R_{Hole} $= (40, 50, 60, 70)m$). Vor-Voronoi outperformed the DSSA and MinMax-Voronoi algorithms. DS-Tree outperformed the DS-Tree RD. The performances of DS-Tree and DS-Tree RD improved as the CH expanded from $40m$ to $70m$; however, The DSSA and Voronoi algorithms' performances were not affected by different sizes of CH. The performances of the relocation algorithm with network of N=1000 nodes followed the same trends (Table 6.2).

6.3.2.1.3 Average Movement

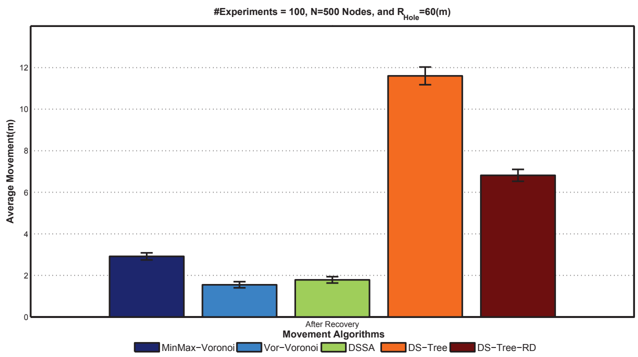
As the results for the network of N=500 nodes in Figure 6.10 and Table 6.3 show, DS-Tree and DS-Tree RD had the worst performances while the Vor-Voronoi and DSSA algorithms outperformed the other algorithms. As in Figure 6.10 and Table 6.3, the average movements of the Voronoi algorithms were not significantly affected by CH size, while the performances of the DS-Tree and DS-Tree RD algorithms depended on sizes of the CH. DSSA's performance did not seem to be much governed by sizes of CH. The behaviour of algorithms for the network of N=1000 nodes, were roughly similar to network of N=500 nodes (Table 6.3).



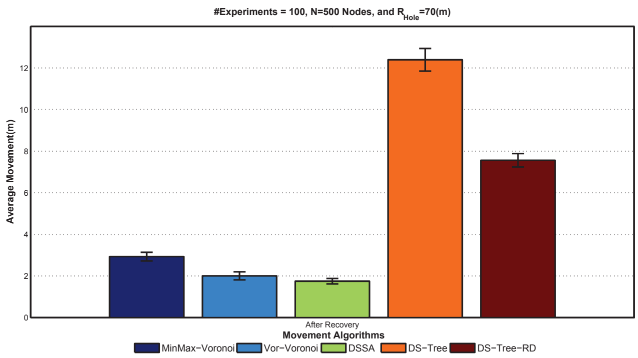
(a) $N = 500$, $R_{Hole} = 40$



(b) $N = 500$, $R_{Hole} = 50$

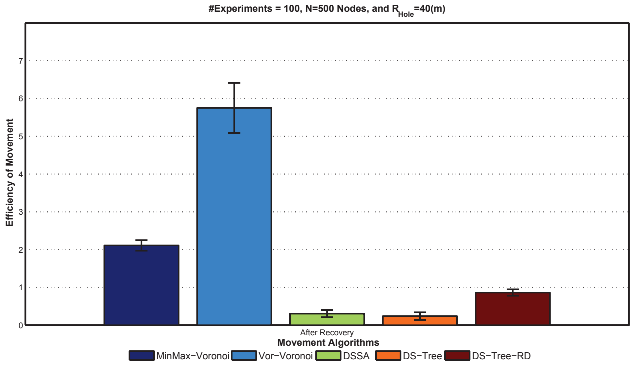


(c) $N = 500$, $R_{Hole} = 60$

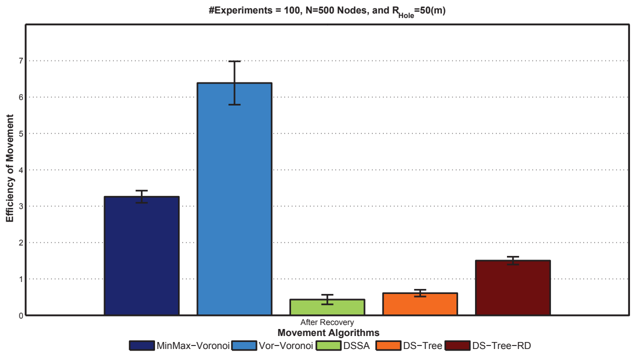


(d) $N = 500$, $R_{Hole} = 70$

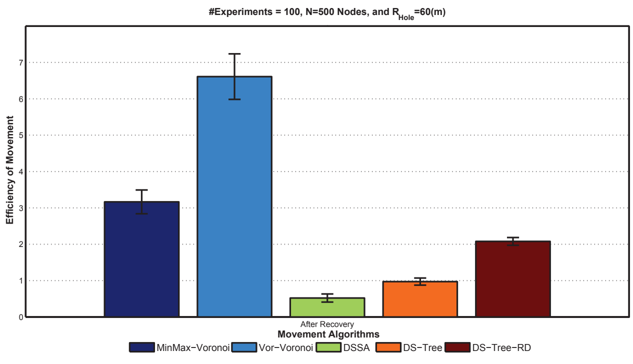
Figure 6.10: Average Movement of Relocation Algs. for N=500, and radii of R_{Hole}=$(40, 50, 60, 70)m$



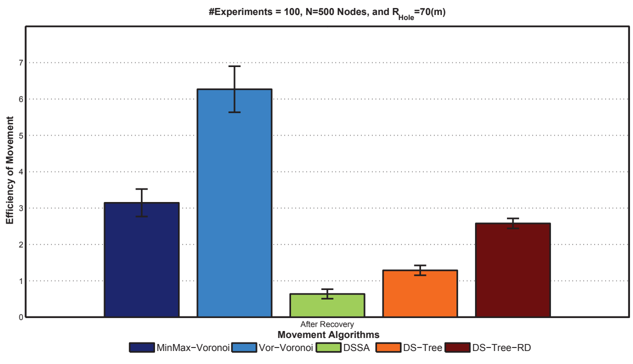
(a) $N = 500,\ R_{Hole} = 40$



(b) $N = 500,\ R_{Hole} = 50$



(c) $N = 500,\ R_{Hole} = 60$



(d) $N = 500,\ R_{Hole} = 70$

Figure 6.11: Efficiency of Movement of Relocation Algs. for N=500, and radii of R_{Hole}=$(40, 50, 60, 70)m$

6.3.2.1.4 Efficiency of Movement

As shown in Figure 6.11 and Table 6.3 for the network of N=500 deployed nodes, Vor-Voronoi and MinMax-Voronoi outperformed the other relocation algorithms, while DSSA had the worst performance among all. The efficiency of movement increases in DS-Tree algorithms as the size of the CH grows. DSSA's performance does not seem to depend on CH sizes, and its efficiency of movement grows very slowly. For the network of N=1000 nodes, the performances of the relocation algorithms show similar behaviours for the given sizes of CH (Table 6.3).

6.3.2.2 Part Two

In this section, the effect of the notification depth and the number of participating nodes (Section 6.3.2.2.1) in the process of CH recovery are examined. Figures 6.13, 6.14, and 6.15 represent the performance of DSSA [227], MinMax-Voronoi and Vor-Voronoi algorithms [231] in terms of percentage of 1-coverage (Section 6.3.2.2.2), uniformity (Section 6.3.2.2.3), and average movement (Section 6.3.2.2.4) for the network of N=500 deployed nodes and CH size of R_{Hole}=50m. The results are obtained based on the different proportions of nodes which are participating in the CH recovery process. Proportion of different node types and participating node selection schemes are shown in Table 6.4 and Section 6.3.2.2.1, respectively.

Moreover, the performance of each given relocation algorithm in terms of the percentage of 1-coverage (Section 6.3.2.2.5), uniformity (Section 6.3.2.2.6), and the average movement (Section 6.3.2.2.7) with respect to the different proportions of participating nodes is shown for the network of N=500 deployed nodes in Figures 6.16, 6.17, 6.18, 6.19, 6.20, and 6.21. Because the results for network of N=1000 nodes with CH sizes of R_{Hole}=$(40, 60, 50, 70)m$ and network of N=500 nodes with CH sizes of R_{Hole}=$(40, 60, 70)m$ have similar trends to the network of N=500

Percentage of Coverage(%)					
No. of Nodes	CH radius	Relocation Alg.	Before Damage	After Damage	After Recovery
$N_{Node} = 500$	$R_{Hole} = 40(m)$	MinMax-Voronoi	99.856 ± 0.150	93.257 ± 0.438	97.923 ± 0.195
		Vor-Voronoi			95.764 ± 0.233
		DSSA			88.637 ± 0.460
		DS-Tree			94.454 ± 1.049
		DS-Tree RD			97.123 ± 0.4934
$N_{Node} = 500$	$R_{Hole} = 50(m)$	MinMax-Voronoi	99.863 ± 0.137	87.852 ± 0.534	97.923 ± 0.194
		Vor-Voronoi			95.764 ± 0.294
		DSSA			88.637 ± 0.513
		DS-Tree			94.454 ± 0.926
		DS-Tree RD			97.123 ± 0.532
$N_{Node} = 500$	$R_{Hole} = 60(m)$	MinMax-Voronoi	99.856 ± 0.147	81.029 ± 0.539	90.272 ± 1.263
		Vor-Voronoi			91.186 ± 0.368
		DSSA			81.956 ± 0.545
		DS-Tree			92.300 ± 0.120
		DS-Tree RD			95.181 ± 0.702
$N_{Node} = 500$	$R_{Hole} = 70(m)$	MinMax-Voronoi	99.887 ± 0.121	72.741 ± 0.605	81.989 ± 1.609
		Vor-Voronoi			85.204 ± 0.427
		DSSA			73.857 ± 0.654
		DS-Tree			88.656 ± 1.593
		DS-Tree RD			92.215 ± 1.196
$N_{Node} = 1000$	$R_{Hole} = 40(m)$	MinMax-Voronoi	99.997 ± 0.013	94.067 ± 0.245	99.444 ± 0.087
		Vor-Voronoi			98.935 ± 0.086
		DSSA			94.484 ± 0.264
		DS-Tree			98.753 ± 0.307
		DS-Tree RD			99.398 ± 0.114
$N_{Node} = 1000$	$R_{Hole} = 50(m)$	MinMax-Voronoi	100.00 ± 0.002	89.068 ± 0.254	98.049 ± 0.089
		Vor-Voronoi			96.291 ± 0.120
		DSSA			89.678 ± 0.208
		DS-Tree			97.887 ± 0.282
		DS-Tree RD			98.449 ± 0.106
$N_{Node} = 1000$	$R_{Hole} = 60(m)$	MinMax-Voronoi	99.999 ± 0.0053	82.541 ± 0.298	93.092 ± 1.091
		Vor-Voronoi			92.104 ± 0.179
		DSSA			83.284 ± 0.344
		DS-Tree			96.377 ± 0.340
		DS-Tree RD			97.070 ± 0.196
$N_{Node} = 1000$	$R_{Hole} = 70(m)$	MinMax-Voronoi	99.993 ± 0.023	74.353 ± 0.307	85.799 ± 1.337
		Vor-Voronoi			86.292 ± 0.207
		DSSA			75.270 ± 0.324
		DS-Tree			94.288 ± 0.356
		DS-Tree RD			95.018 ± 0.272

Table 6.1. Percentage of 1-Coverage of Relocation Algs. Different CH Radius R_{Hole}=$(40, 50, 60, 70)$ m, N=$500, 1000$ Nodes

Uniformity					
No. of Nodes	CH radius	Relocation Alg.	Before Damage	After Damage	After Recovery
$N_{Node} = 500$	$R_{Hole} = 40(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.169 ± 0.043	6.256 ± 0.123	6.762 ± 0.105 6.173 ± 0.134 6.692 ± 0.093 5.843 ± 0.153 6.025 ± 0.117
$N_{Node} = 500$	$R_{Hole} = 50(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.171 ± 0.037	6.268 ± 0.129	6.650 ± 0.120 6.134 ± 0.142 6.718 ± 0.107 5.695 ± 0.194 5.942 ± 0.164
$N_{Node} = 500$	$R_{Hole} = 60(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.171 ± 0.362	6.143 ± 0.113	6.529 ± 0.107 5.935 ± 0.130 6.561 ± 0.096 5.338 ± 0.193 5.740 ± 0.147
$N_{Node} = 500$	$R_{Hole} = 70(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.150 ± 0.046	5.979 ± 0.146	6.350 ± 0.117 5.660 ± 0.142 6.399 ± 0.113 4.909 ± 0.210 5.448 ± 0.158
$N_{Node} = 1000$	$R_{Hole} = 40(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.396 ± 0.0176	6.744 ± 0.076	6.998 ± 0.048 6.728 ± 0.067 6.984 ± 0.056 6.645 ± 0.036 6.707 ± 0.059
$N_{Node} = 1000$	$R_{Hole} = 50(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.394 ± 0.018	6.699 ± 0.060	6.907 ± 0.052 6.647 ± 0.067 6.945 ± 0.045 6.500 ± 0.074 6.622 ± 0.058
$N_{Node} = 1000$	$R_{Hole} = 60(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.390 ± 0.022	6.607 ± 0.057	6.807 ± 0.052 6.520 ± 0.066 6.853 ± 0.053 6.255 ± 0.101 6.480 ± 0.071
$N_{Node} = 1000$	$R_{Hole} = 70(m)$	MinMax-Voronoi Vor-Voronoi DSSA DS-Tree DS-Tree RD	3.394 ± 0.018	6.478 ± 0.066	6.676 ± 0.063 6.340 ± 0.081 6.714 ± 0.051 5.981 ± 0.106 6.284 ± 0.079

Table 6.2. Uniformity of Relocation Algs. Different CH Radius R_{Hole}=$(40, 50, 60, 70)$ m and N=500,N=1000 Nodes

Movement(m)				
No. of Nodes	CH radius	Relocation Alg.	Average Movement	Efficiency of Movement
$N_{Node} = 500$	$R_{Hole} = 40(m)$	MinMax-Voronoi	2.783 ± 0.110	2.111 ± 0.140
		Vor-Voronoi	0.946 ± 0.102	5.751 ± 0.662
		DSSA	1.894 ± 0.136	0.307 ± 0.094
		DS-Tree	10.609 ± 0.263	0.239 ± 0.102
		DS-Tree RD	5.833 ± 0.183	0.865 ± 0.085
$N_{Node} = 500$	$R_{Hole} = 50(m)$	MinMax-Voronoi	3.091 ± 0.087	3.260 ± 0.167
		Vor-Voronoi	1.248 ± 0.106	6.389 ± 0.596
		DSSA	1.814 ± 0.128	0.432 ± 0.132
		DS-Tree	10.884 ± 0.264	0.608 ± 0.093
		DS-Tree RD	6.177 ± 0.233	1.502 ± 0.108
$N_{Node} = 500$	$R_{Hole} = 60(m)$	MinMax-Voronoi	2.916 ± 0.170	3.166 ± 0.327
		Vor-Voronoi	1.550 ± 0.148	6.610 ± 0.626
		DSSA	1.786 ± 0.152	0.520 ± 0.112
		DS-Tree	11.604 ± 0.425	0.973 ± 0.098
		DS-Tree RD	6.816 ± 0.287	2.079 ± 0.107
$N_{Node} = 500$	$R_{Hole} = 70(m)$	MinMax-Voronoi	2.932 ± 0.207	3.146 ± 0.378
		Vor-Voronoi	2.007 ± 0.194	6.268 ± 0.634
		DSSA	1.750 ± 0.129	0.639 ± 0.131
		DS-Tree	12.392 ± 0.545	1.288 ± 0.135
		DS-Tree RD	7.562 ± 0.324	2.578 ± 0.136
$N_{Node} = 1000$	$R_{Hole} = 40(m)$	MinMax-Voronoi	1.967 ± 0.060	2.736 ± 0.140
		Vor-Voronoi	0.483 ± 0.047	10.176 ± 1.074
		DSSA	1.323 ± 0.060	0.313 ± 0.080
		DS-Tree	9.896 ± 0.143	0.473 ± 0.036
		DS-Tree RD	5.439 ± 0.099	0.980 ± 0.041
$N_{Node} = 1000$	$R_{Hole} = 50(m)$	MinMax-Voronoi	2.180 ± 0.064	4.327 ± 0.188
		Vor-Voronoi	0.652 ± 0.063	11.179 ± 1.122
		DSSA	1.320 ± 0.064	0.463 ± 0.116
		DS-Tree	10.004 ± 0.193	0.882 ± 0.037
		DS-Tree RD	6.617 ± 0.135	1.671 ± 0.054
$N_{Node} = 1000$	$R_{Hole} = 60(m)$	MinMax-Voronoi	2.117 ± 0.077	4.980 ± 0.415
		Vor-Voronoi	0.844 ± 0.075	11.432 ± 1.091
		DSSA	1.290 ± 0.022	0.577 ± 0.113
		DS-Tree	10.054 ± 0.281	1.377 ± 0.052
		DS-Tree RD	5.820 ± 0.158	2.482 ± 0.069
$N_{Node} = 1000$	$R_{Hole} = 70(m)$	MinMax-Voronoi	2.195 ± 0.110	5.206 ± 0.508
		Vor-Voronoi	1.130 ± 0.091	10.639 ± 0.876
		DSSA	1.235 ± 0.080	0.744 ± 0.106
		DS-Tree	10.167 ± 0.361	1.963 ± 0.073
		DS-Tree RD	6.160 ± 0.204	3.358 ± 0.109

Table 6.3. Average Efficiency of Movement of Relocation Algs. Different CH Radius R_{Hole}=$(40, 50, 60, 70)$ m N=500,N=1000 Nodes

nodes and CH size of R_{Hole}=50 m, they are not shown.

6.3.2.2.1 Participating Nodes

Different models of participating nodes in the recovery process can be considered according to following selections criteria:

B-nodes that independently detect the damage event (occurrence of CH) within their range.

All U-nodes all functional nodes outside the CH.

Closest x% of U-nodes is the fraction of U-nodes, that it is defined as the percentage of U-nodes which are closest to the damaged area compared to the rest. The distances of U-nodes $S_u(i)$ ($\forall i \in \phi_u$) located in ($x_{S_u(i)}$,$y_{S_u(i)}$) from the CH modelled as a circle with radius of r_{Hole} with the centre of (x_{Hole},y_{Hole}) (Section 6.2.2) are defined as Equation 6.7.

$$d_{S_u(i)}^{Hole_k} = \sqrt{(x_{S_u(i)} - x_{Hole})^2 + (y_{S_u(i)} - y_{Hole})^2} - r_{Hole} \tag{6.7}$$

If these distances are sorted from ascending order as,

$$A_{sort}(S_u(i)) = \{d_{S_u(1)}^{Hole_k} \leq \cdots \leq d_{S_u(N_u)}^{Hole_k}\}, \ (\forall i \in \phi_u) \tag{6.8}$$

N_u is the total number of U-nodes and $Hole_k$ is CH$_k$. Then, closest $x\%$ of U-nodes from CH$_k$ is defined as:

$$x\%(N_U) \equiv \left\lfloor \frac{A_{sort}(U_{s_j})}{N_{U_{s_i}}} \right\rfloor \cdot 100\% \tag{6.9}$$

Tree level-based selects nodes with the same levels in DS-Trees of different heights around the CH (Section 6.2.6). In this scheme, a set of nodes around the CH from the DS-Trees' roots to the given level L are chosen. For example, up to

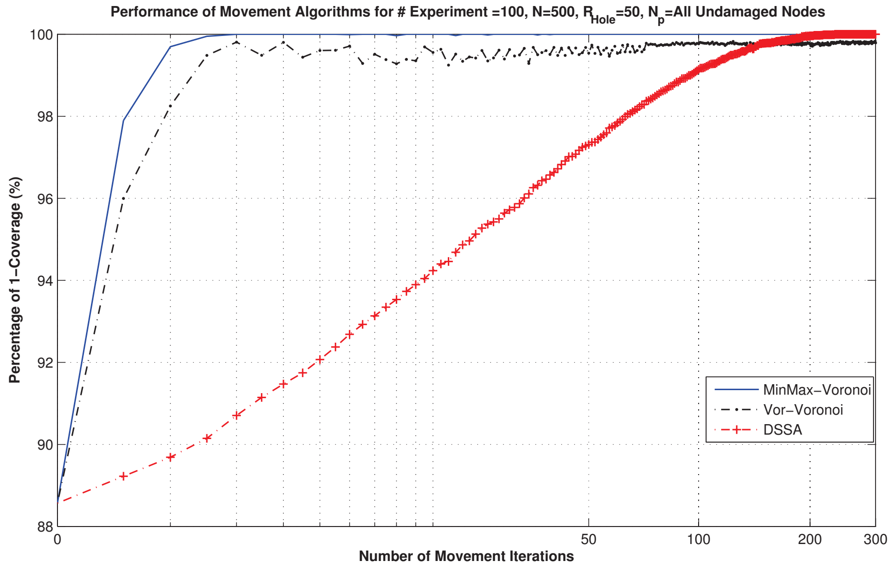
L=4 of each tree, nodes from level 1 (root) to level 4, are selected to participate in the CH recovery. The selection in N_P=100 experiments (Table 6.4) resulted in the average number of participating nodes out of the total number of deployed nodes (N=500).

6.3.2.2.2 Coverage for a Given Node Selection

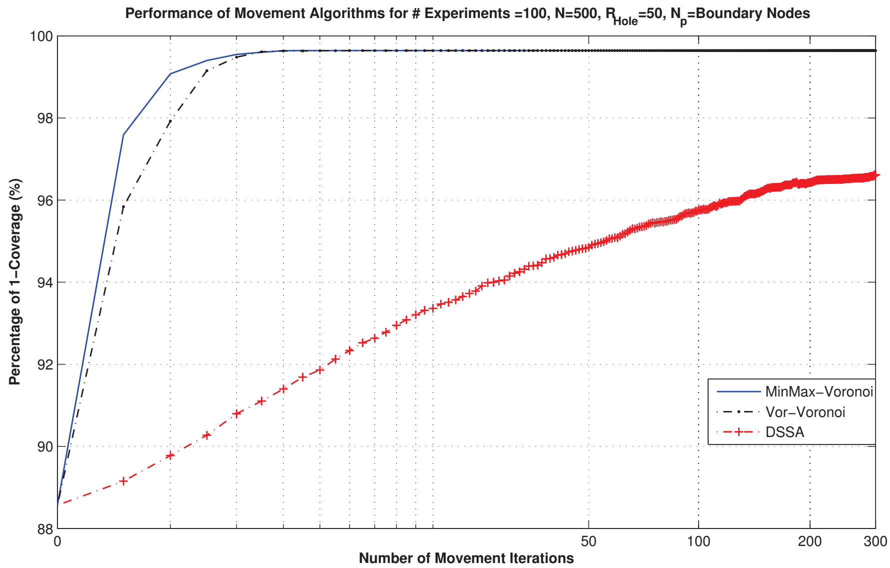
As shown in Figure 6.13, for a network of N=500 deployed nodes, DSSA had the worst performance, while MinMax-Voronoi had the best coverage in different models of node participation. The results indicate that relocation algorithm DSSA slowly reaches full coverage after a large number of movement iterations (around the 200th iteration) (Please refer to Appendix B). In the case that only boundary nodes participated in the recovery, DSSA was not able to fully cover the damaged area. As shown in Figure 6.13 that the behaviours of the MinMax-Voronoi and Vor-Voronoi algorithms became more similar after a small number of iterations, except when all undamaged nodes participate in the relocation. Similarly, if the closest 50% of U-nodes and Tree-based level L=5 participate in recovery, their performance are fairly close to the case where all the U-nodes moves toward CH. It can also be seen that the performance of Tree-based level L=5 had a small difference with the closest 50% of U-Nodes.

6.3.2.2.3 Uniformity of Given Node Selection

Figure 6.14 illustrate that, except for the B-node selection scheme, in other schemes, MinMax-Voronoi and Vor-Voronoi had the worst and best performance respectively. In B-node selection scheme, although all relocation algorithms had close performance, the Voronoi algorithms outperformed the DSSA. For a small number

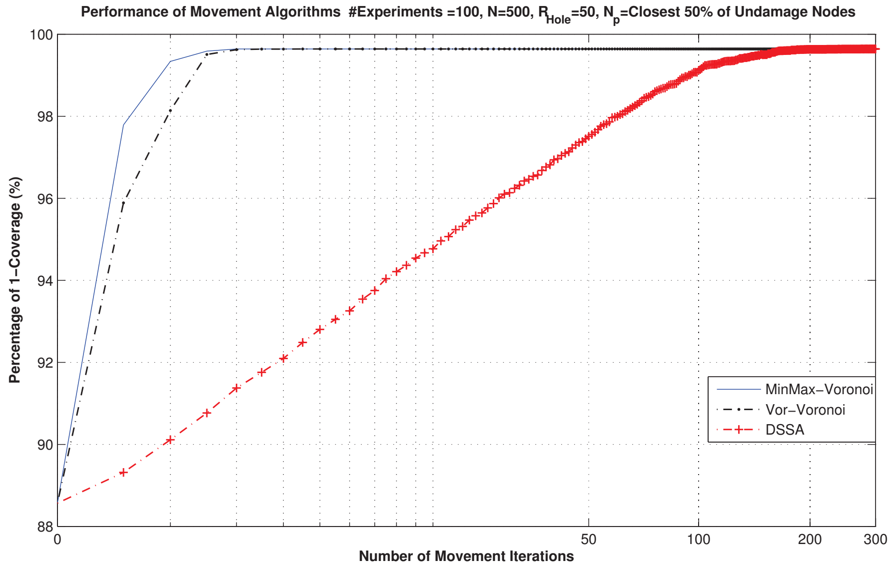


(a) All Undamaged Nodes,$N = 500$ $R_{Hole} = 50$

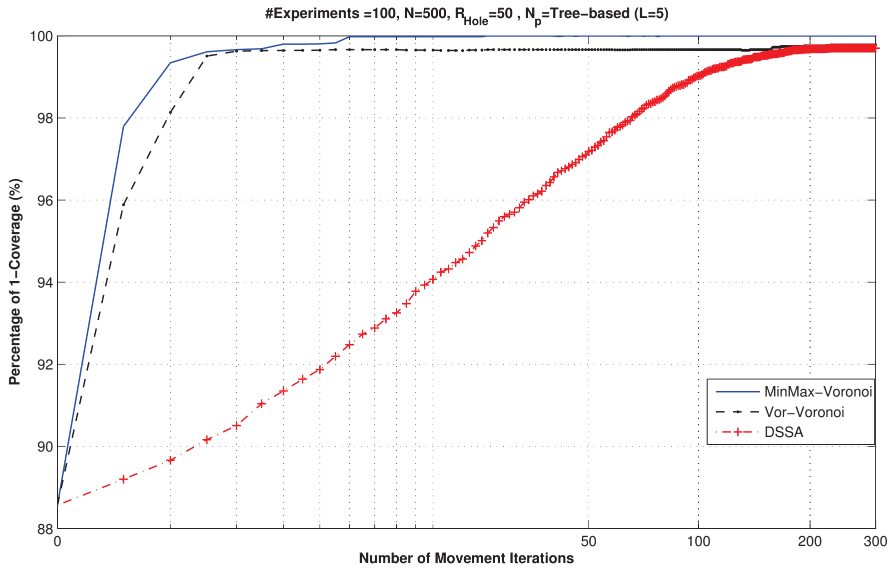


(b) Boundary Nodes,$N = 500$, $R_{Hole} = 50$

Figure 6.12: Percentage of 1-Coverage of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes



(c) Closest 50% of Undamaged Nodes,$N = 500$, $R_{Hole} = 50$



(d) Tree-based, till level $L = 5$, $N = 500$, $R_{Hole} = 50$

Figure 6.13: Percentage of 1-Coverage of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes

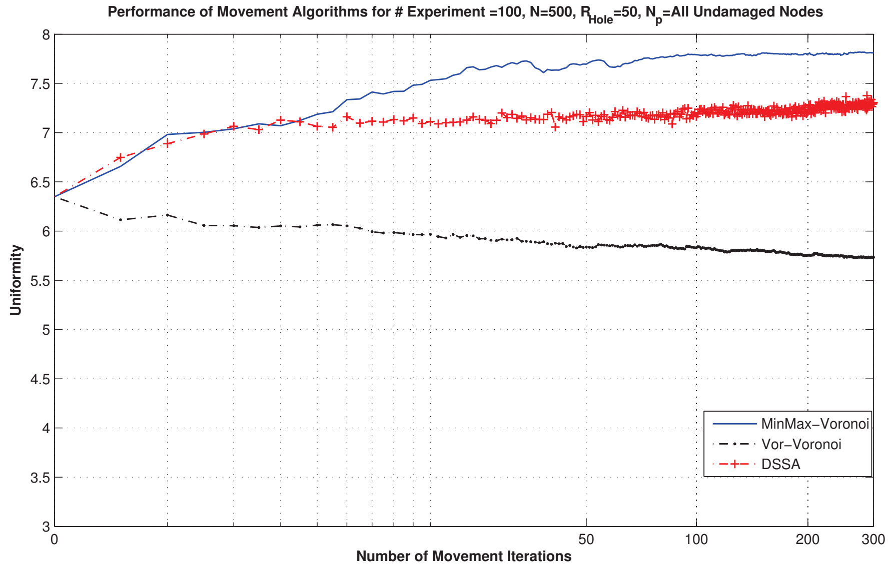
Participating Nodes N_P No. Experiments = 100					
No. Nodes	CH radius	No. B-Nodes	No. U-Nodes	No. D-Nodes	No. Nodes DS-Tree Levels
500	$50(m)$	41.350 ± 4.435	400.960 ± 4.948	99.040 ± 4.948	$L=1$: 41.350 ± 4.457 $L=2$: 96.370 ± 13.380 $L=3$: 62.350 ± 5.110 $L=4$: 70.720 ± 5.784 $L=5$: 76.420 ± 4.753 $L=6$: 52.930 ± 4.776 $L=7$: 26.500 ± 2.443 $L=8$: 12.660 ± 2.430 $L=9$: 1.010 ± 0.266 $L=10$: 0.520 ± 0.502

Table 6.4. Number of Participating Nodes for N=500 Nodes and Coverage Hole Radius, R_{Hole}=50 m

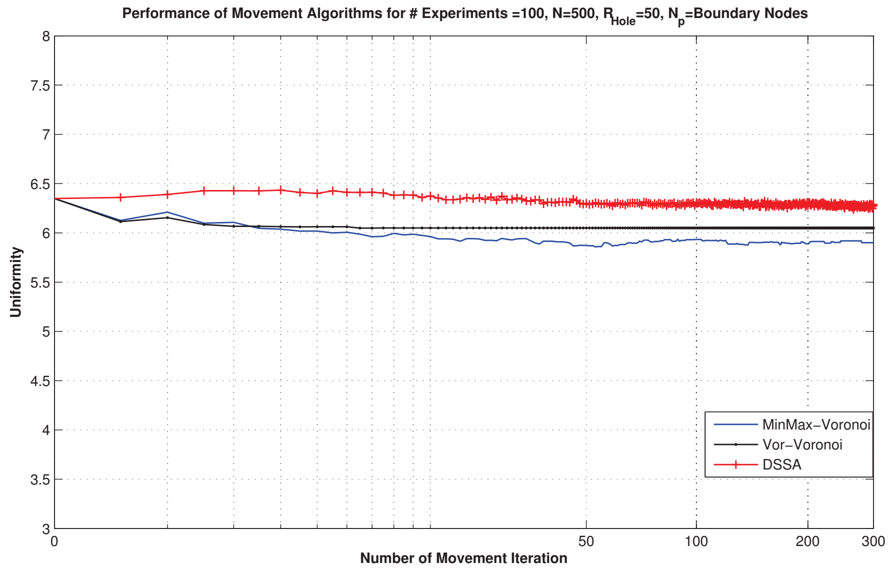
of movement iterations, the performance of MinMax-Voronoi matches the DSSA's performance. For the selection schemes of the closest 50% of U-Nodes and Tree-based L=5, the performance are fairly close to the case where all the U-nodes were applied in the recovery. The results show that it is not necessary to notify and/or move all nodes. Therefore, it is possible to recover even LSCHs, if a certain proportion of nodes around CHs are notified and a suitable node selection scheme is applied for the relocation algorithms.

6.3.2.2.4 Average Movement of of Given Node Selection

As shown in Figure 6.15, for network of N=500 deployed nodes, Vor-Voronoi outperformed other algorithms, while DSSA was the worst. When all U-nodes participated in the recovery, MinMax-Voronoi and Vor-Voronoi fluctuated, where amplitude of oscillation was smaller in the Vor-Voronoi algorithm. If the boundary nodes participated in the recovery process, Vor-Voronoi algorithm showed the best performance, while MinMax-Voronoi and DSSA had close performances. For the

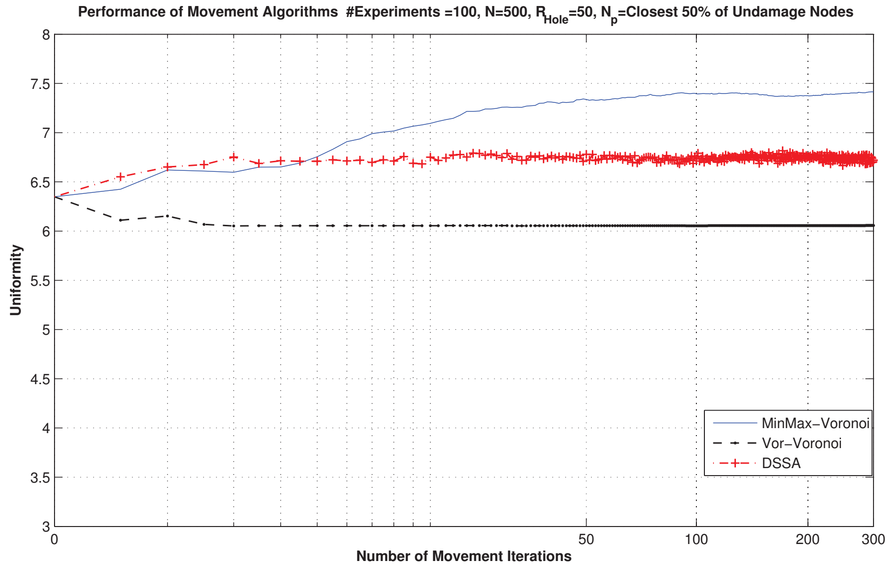


(a) All Undamaged Nodes,$N = 500$,$R_{Hole} = 50$

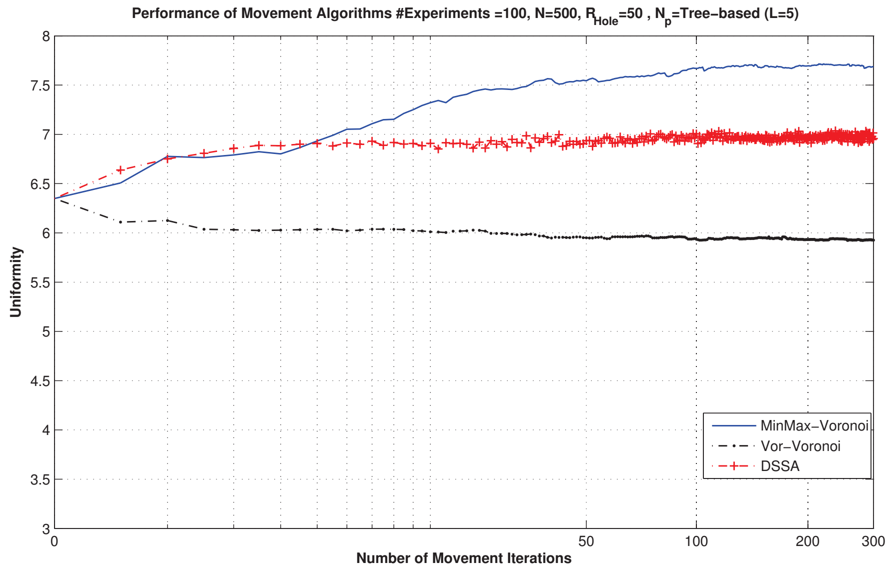


(b) Boundary Nodes,$N = 500$,$R_{Hole} = 50$

Figure 6.14: Uniformity of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes



(c) Closest 50% of Undamaged Nodes,$N = 500$,$R_{Hole} = 50$



(d) Tree-based, till level $L = 5$,$N = 500$,$R_{Hole} = 50$

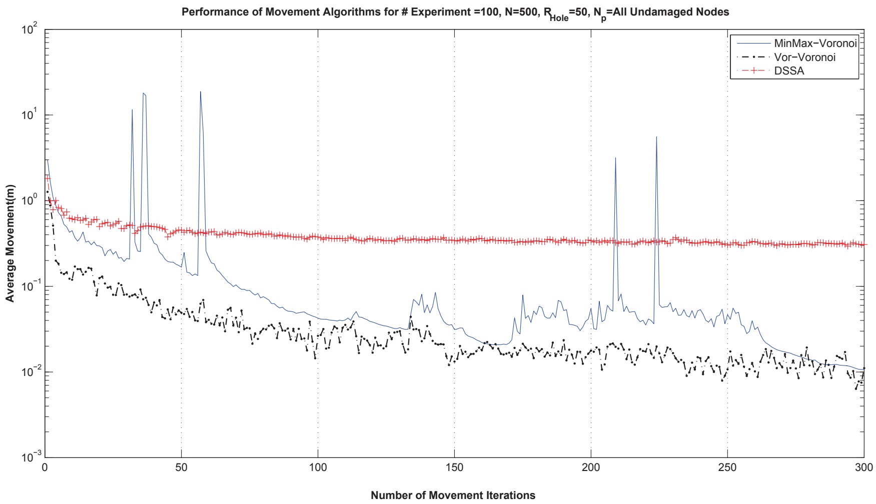
Figure 6.14: Uniformity of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes (cont'd)

case of the closest 50% of U-nodes and Tree-based level L=5 selection schemes, algorithms' performances were fairly close to the case when all the U-nodes were involved in the movement. It implies that, it is not necessary to notify all nodes; the participation of the nodes mainly in the margin of the CHs would be sufficient.

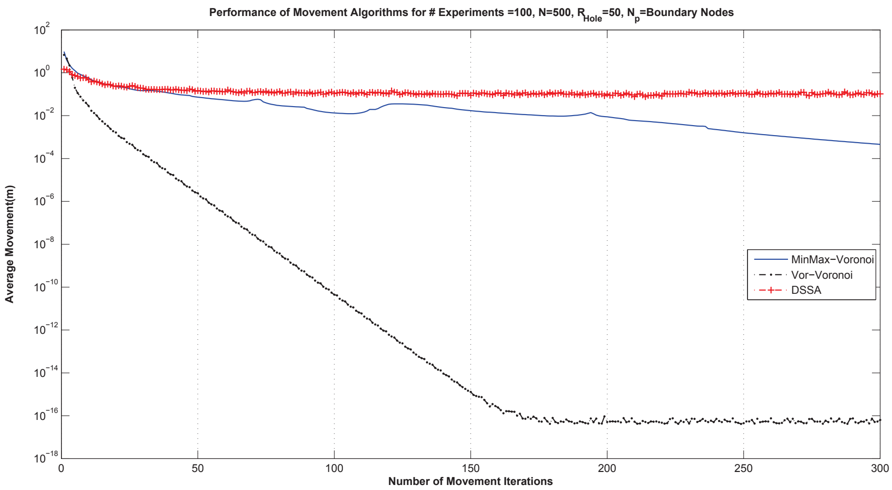
6.3.2.2.5 Coverage for a Given Algorithm

Figure 6.16 shows that, for network of N=500 deployed nodes, the performances of the relocation algorithms with regards to variable participating nodes in which x increased stepwise from 10% to 100% with the steps of 10% in the closest X% of U-nodes selection scheme. The results show that increasing the number of participating nodes would not significantly improve the coverage of damaged area. For DSSA, for lower number of iterations, increasing the number of participating nodes did not have significant effect on the performance of relocation algorithm. The Performance of DSSA in larger numbers of iterations and beyond 40% and 50% participating nodes also did not improve significantly. Thus, it can be seen that effective recovery can be done by about 40% of U-nodes. As shown in Figure 6.16, increasing the participating nodes from 10% to 20% and from 20% to 30% improved the performance of recovery. However, this trend fades as the number of nodes increased from 30% to 40%. Increasing the number of participating nodes beyond 40% closest U-nodes to CH did not seem to have much effect on the performance accordingly.

For large numbers of movement iterations, although increasing the participating nodes from 20% to 30% and from 10% to 20% improved the performance of recovery, the improvement in the former was less than the latter in the network. Figure 6.17 shows the performance in term of the percentage of 1-coverage where reloca-

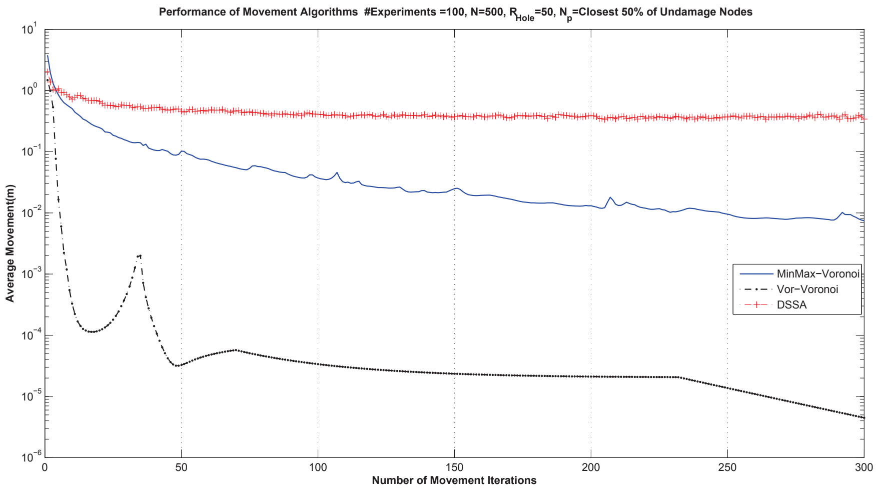


(a) All Undamaged Nodes,$N = 500$,$R_{Hole} = 50$

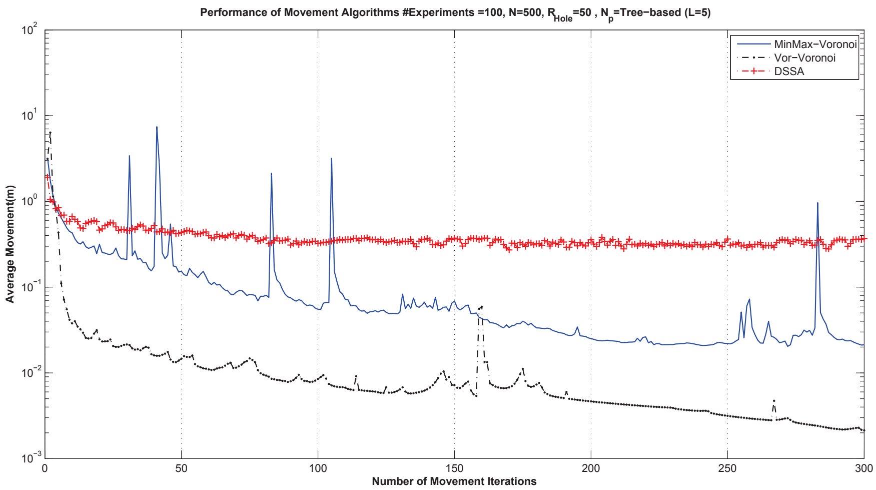


(b) Boundary Nodes,$N = 500$,$R_{Hole} = 50$

Figure 6.15: Average Movement of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes

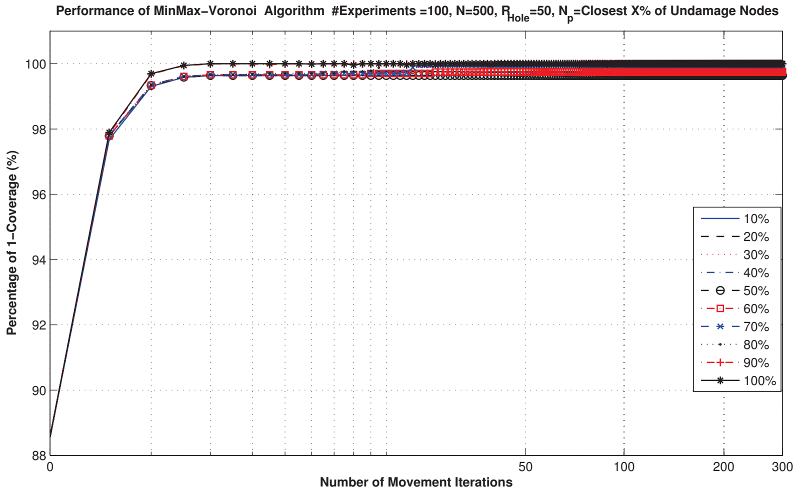


(c) Closest 50% of Undamaged Nodes,$N = 500$,$R_{Hole} = 50$

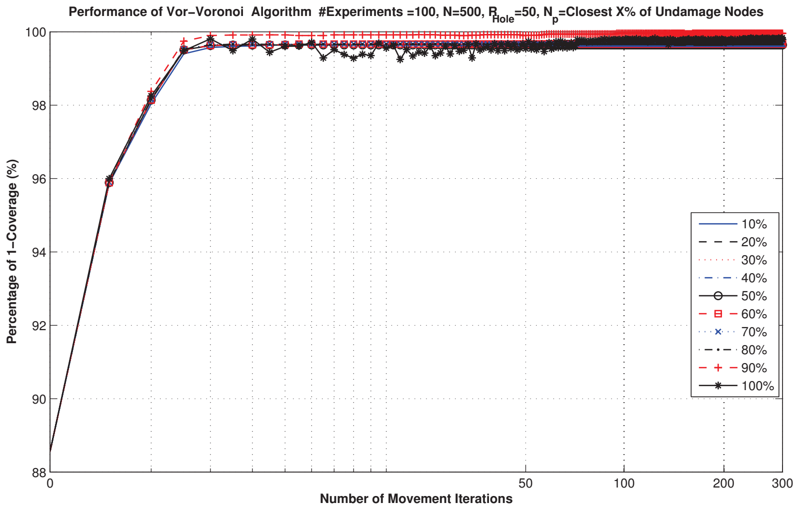


(d) Tree-based, till level $L = 5$,$N = 500$,$R_{Hole} = 50$

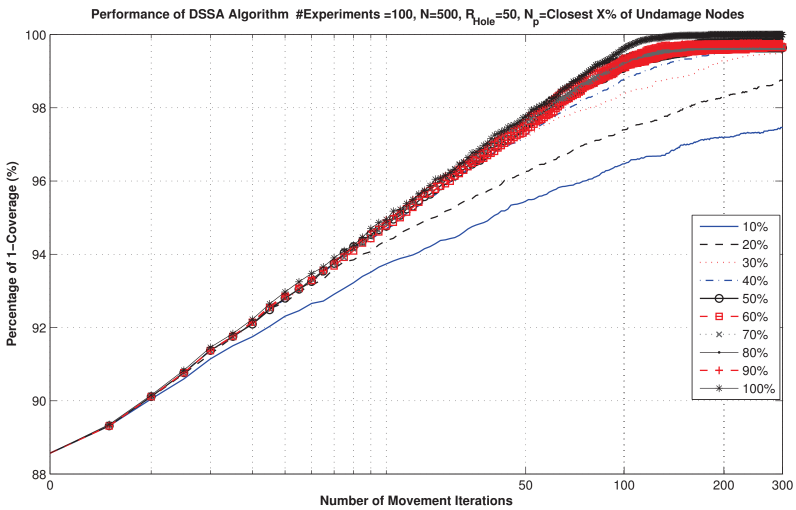
Figure 6.15: Average Movement of Relocation Algs. N=500, R_{Hole}=50 and Different participating Nodes (cont'd)



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$



(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$



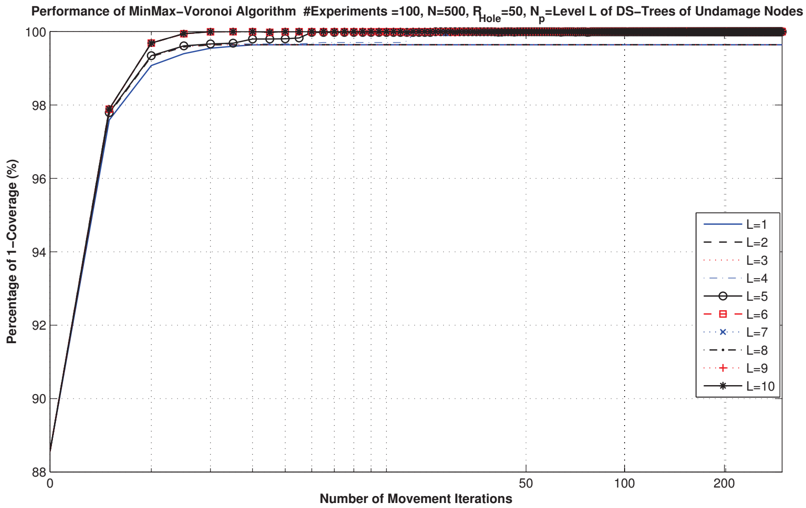
(c) DSSA,$N = 500$,$R_{Hole} = 50$

Figure 6.16: Percentage of 1-Coverage of Relocation Algs. for N=500, R_{Hole}=50 for N_p=Closest $X\%$ of U-nodes

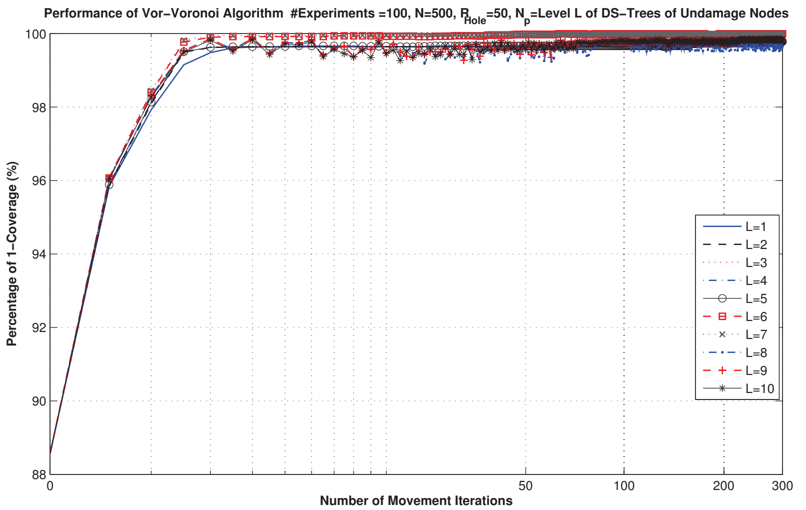
tion algorithms were applied to the variable number participating nodes. These nodes resided in their DS-Trees from the root to the given Level L. L increases from L=1 to L=10 stepwise with the step of 1. Figure 6.17 indicates that increasing the number of participating nodes did not significantly improve the coverage of the damaged area for the MinMax-Voronoi and Vor-Voronoi algorithms. In DSSA, for the lower number of iterations, the variation of the number of participating nodes did not significantly affect the performance, but for the larger number iterations, increasing the number of participating nodes up to L=4 improved its performance. According to Figure 6.17, including nodes from L=1 to L=2 and from L=2 to L=3 of DS-Trees improve the performance. Considering participating nodes beyond level L=4 within the DS-Trees did not seem to contribute significantly to the recovery process. Therefore, efficient recovery can be done when only nodes from DS-Trees' roots to L=4th levels are selected around the damaged area. The main improvement in performance are resulted as the nodes from level L=2 of DS-Trees were appended to the current participating nodes, which is more effective in the larger iterations in DSSA.

6.3.2.2.6 Uniformity for a given Algorithm

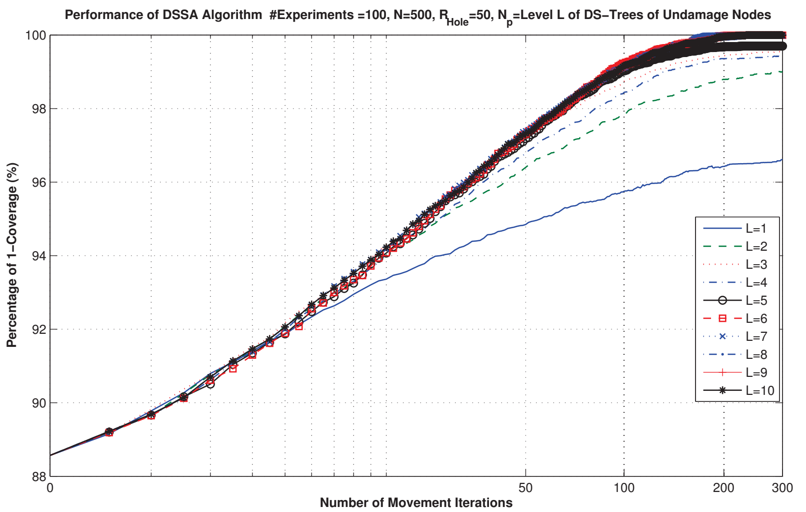
For a network of N=500 deployed nodes, Figure 6.18 shows the uniformity of the given relocation algorithms as the number of participating nodes increased step-wise with step of 10% of the closest nodes around the CH. Figure 6.18 shows that, except for the Vor-Voronoi algorithm, selecting different percentages of nodes affected the performances. If the smaller number of nodes participated in the recovery process, the distribution of the remaining nodes, which are not involved in the recovery would remain untouched in the network, so better uniformity was expected. Using 10% and 20% of a network's nodes did not significantly change



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$



(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$



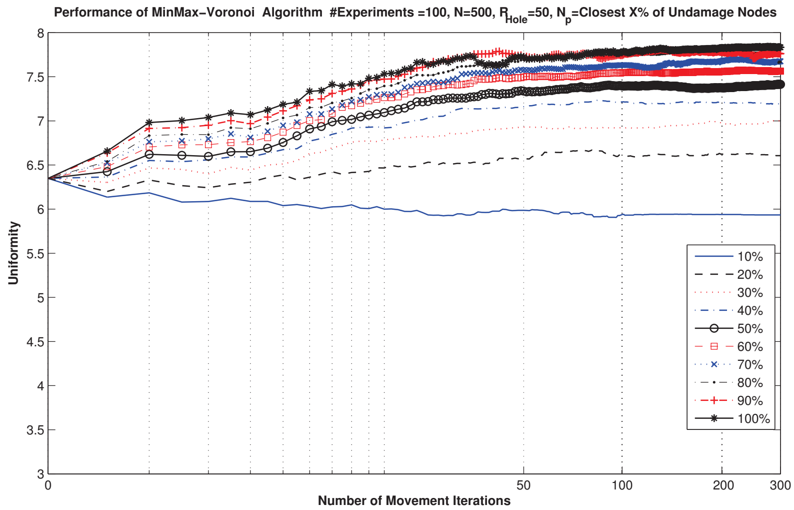
(c) DSSA,$N = 500$,$R_{Hole} = 50$

Figure 6.17: Percentage of 1-Coverage of Relocation for Algs. N=500, R_{Hole}=50, N_p=Level L of DS-Trees of U-nodes

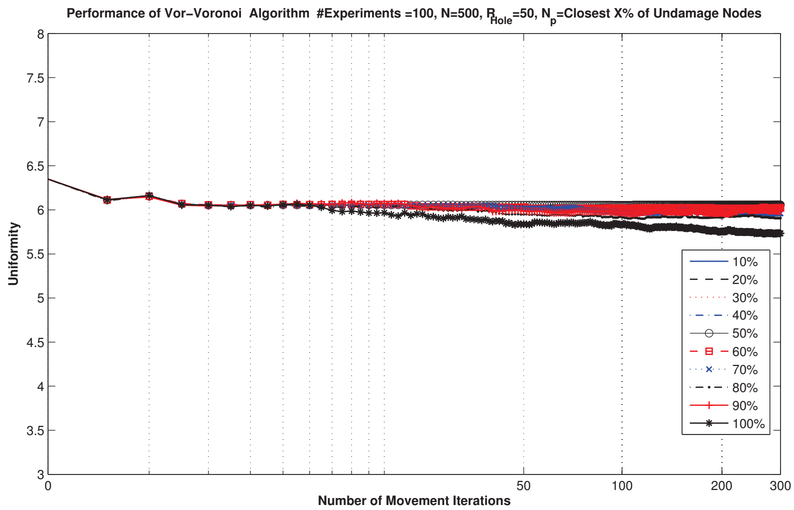
the network's uniformity after it was disturbed by the damage event. The Vor-Voronoi algorithm showed the best performance among the other algorithms. For the network of N=500 deployed nodes, it can be seen from Figure 6.19 that except Vor-Voronoi, including nodes from different levels of DS-Trees affected the performances. The smaller number of participating nodes in tree levels seemed to have better performance as there is less impact on the node arrangements due to the recovery process. Using nodes from levels L=1 and L=2 of DS-Trees, did not seem to worsen the uniformity after the occurrence of the damage event. As in Figure 6.19, Vor-Voronoi showed the best uniformity among all.

6.3.2.2.7 Average Movement for a given Algorithm

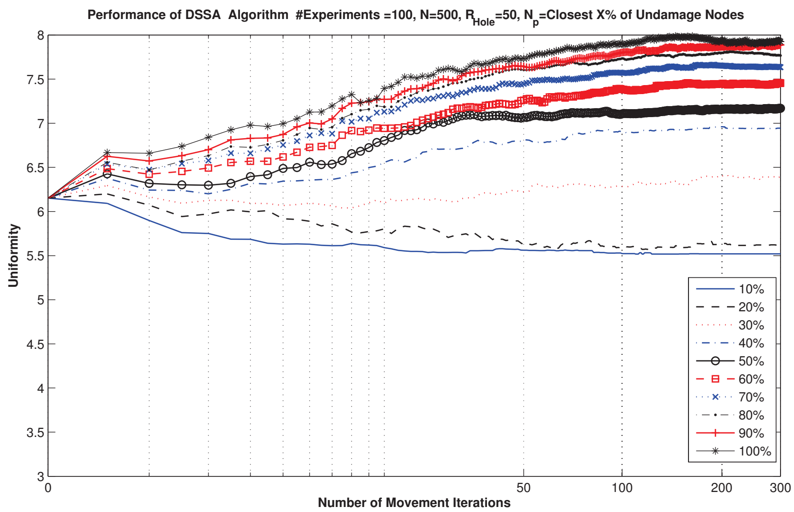
Figure 6.20 presents the relocation algorithms' performance with respect to nodes' average movements for a network of N=500 nodes. Results indicate that the Vor-Voronoi algorithm had the best performance. The MinMax-Voronoi algorithm showed some fluctuations when 90% and 100% of node participate in the recovery. In the Vor-Voronoi algorithm, the performance changed significantly as the percentage of participating nodes increased from 10% to 20% and to 30% accordingly. However, beyond the given percentage, the performance does not seem to change significantly. Thus, the most effective number of participating nodes are 30% of closest U-nodes around the CH. In MinMax-Voronoi and DSSA, this trend is for the first 50% of the closest U-nodes. If more than 50% of the closest U-nodes participate in the recovery process, the performance does not change significantly. Unlike the MinMax-Voronoi and Vor-Voronoi algorithms, the performance of DSSA seemed to change smoothly by adding the closest U-nodes with steps of 10%. As Figure 6.21 presents for network of N=500 nodes, the Vor-Voronoi algorithm had the best performance. The MinMax-Voronoi algorithm showed some fluctuations



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$

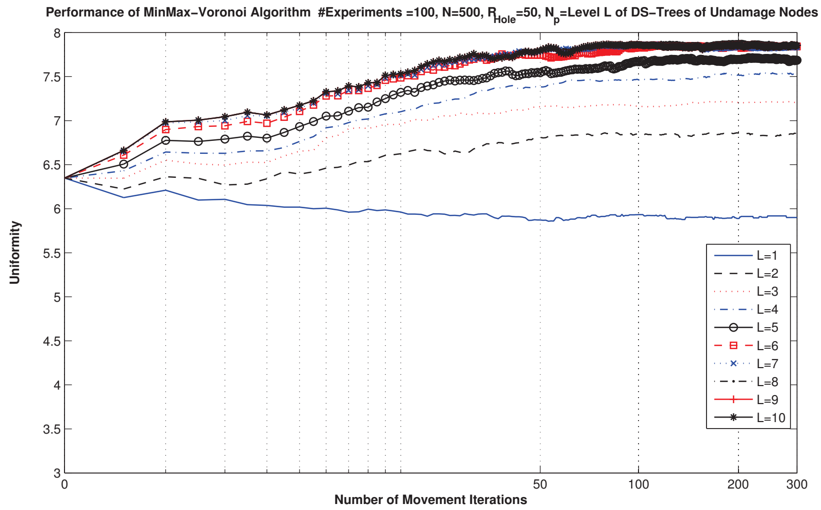


(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$

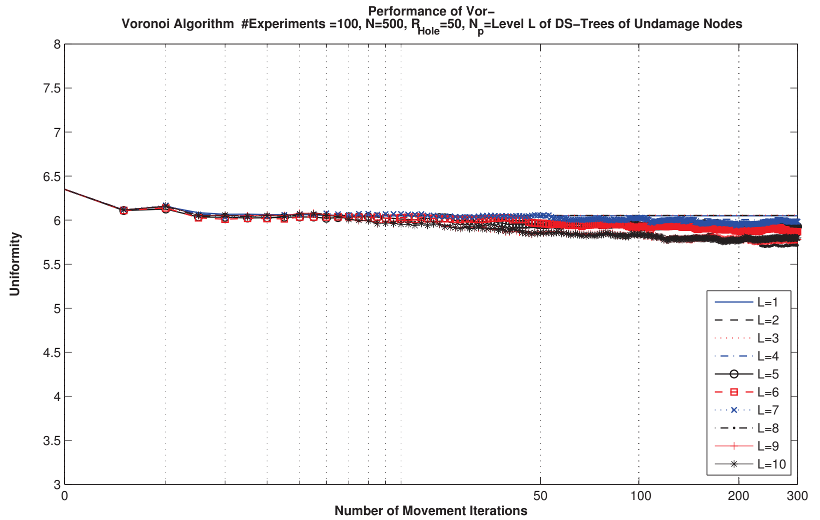


(c) DSSA,$N = 500$,$R_{Hole} = 50$

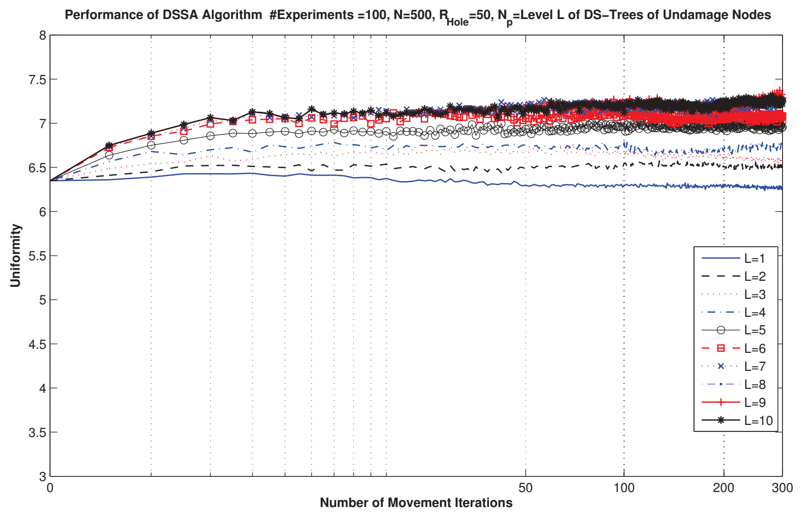
Figure 6.18: Uniformity of Relocation Algs. for N=500,R_{Hole}=50 for N_p=Closest X% of U-nodes



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$

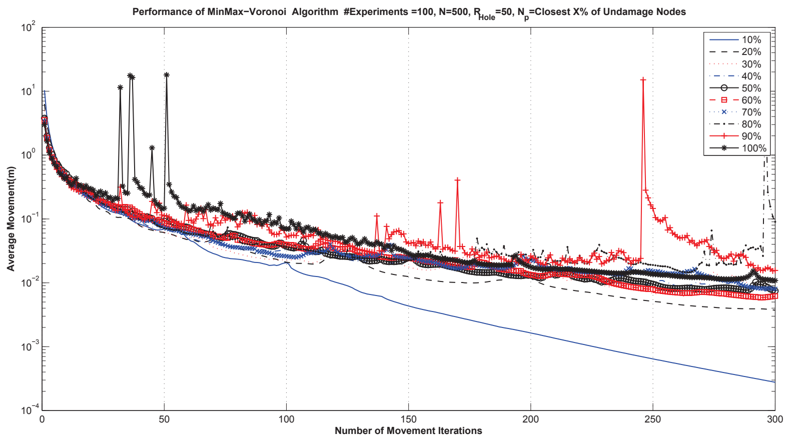


(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$

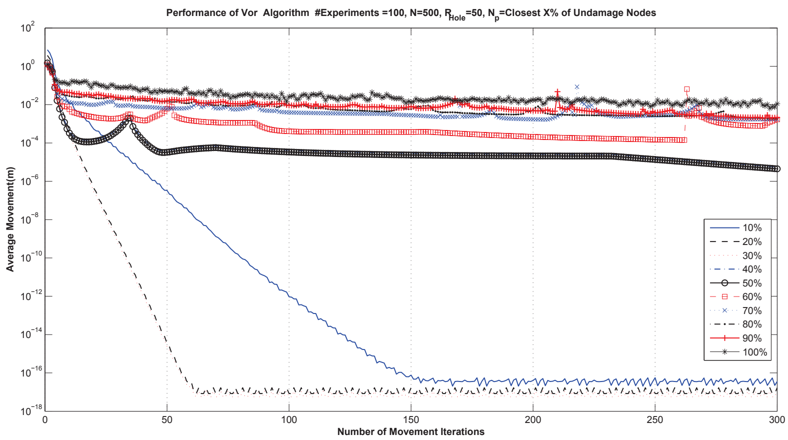


(e) DSSA,$N = 500$,$R_{Hole} = 50$

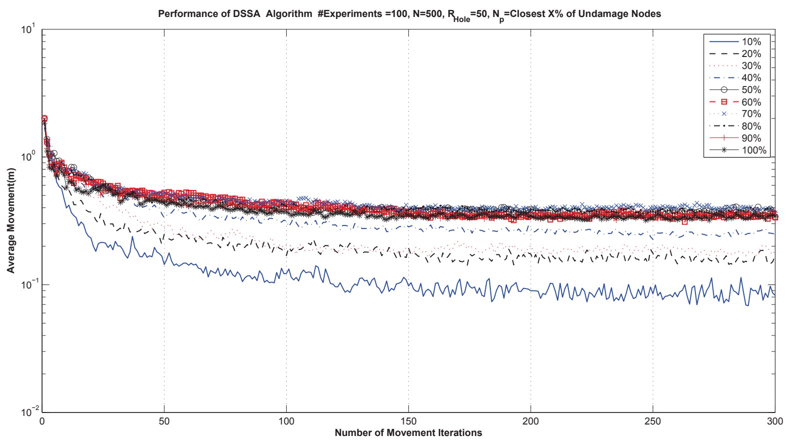
Figure 6.19: Uniformity of Relocation Algs. for N=500, R_{Hole}=50,N_p=Level L of DS-Trees of U-nodes



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$

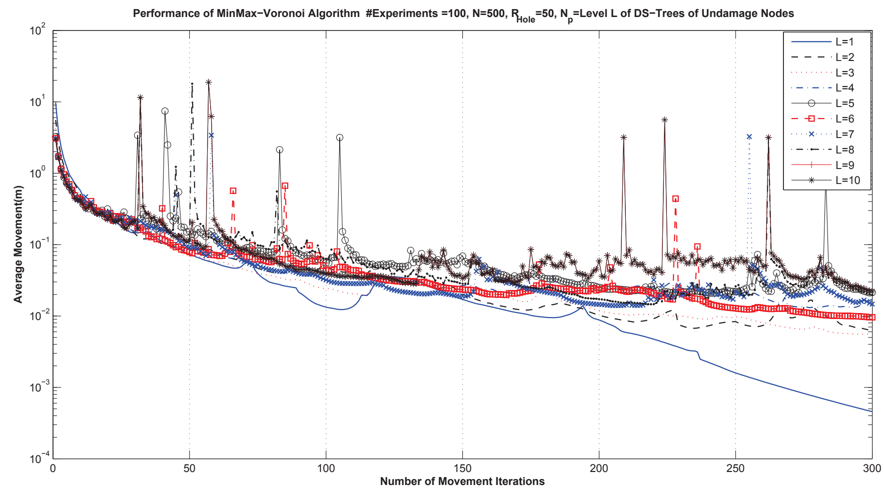


(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$

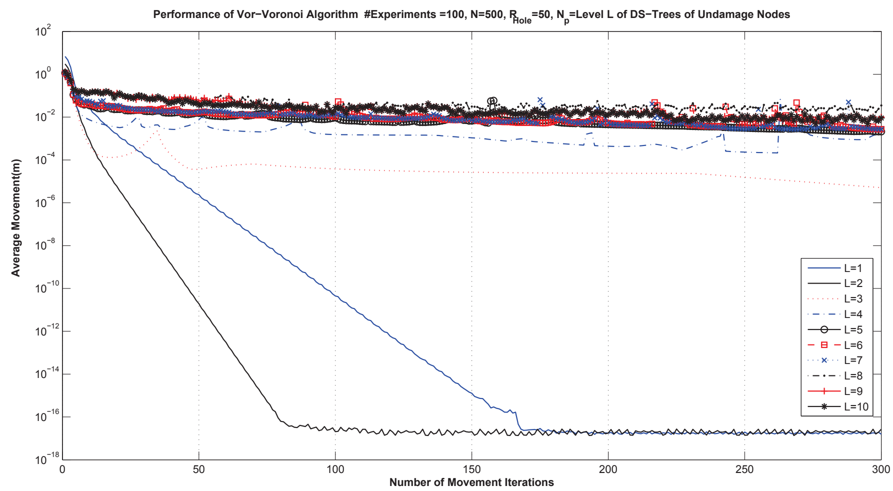


(e) DSSA,$N = 500$,$R_{Hole} = 50$

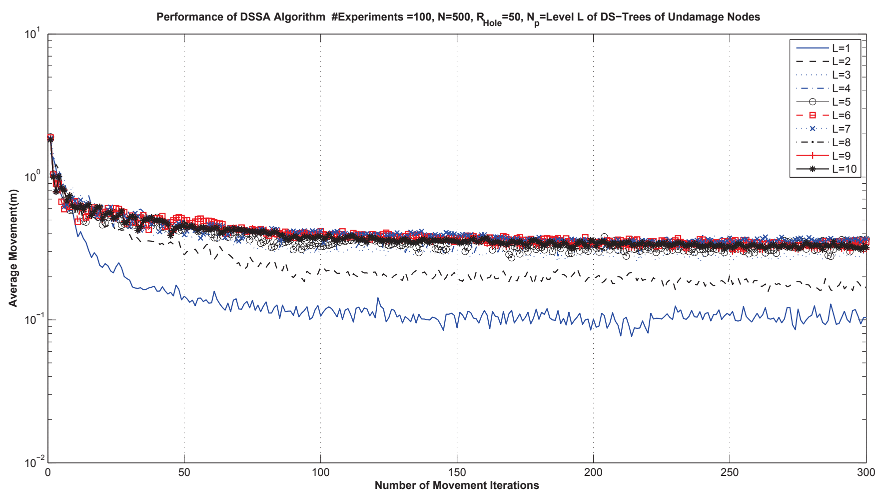
Figure 6.20: Average Movement of Relocation Algs. for N=500, R_{Hole}=50 for N_p=Closest X% of U-nodes



(a) MinMax-Voronoi,$N = 500$,$R_{Hole} = 50$



(b) Vor-Voronoi,$N = 500$,$R_{Hole} = 50$



(c) DSSA,$N = 500$,$R_{Hole} = 50$

Figure 6.21: Average Movement of Relocation Algs. for N=500, R_{Hole}=50, N_p=Level L of DS-Trees of U-nodes

when nodes participated in the recovery from the root to levels of L=4 to L=9 in DS-Trees. In the Vor-Voronoi algorithm the performance changed appreciably as there was an increase in participating nodes from levels of L=1 to L=5 in DS-Trees. Beyond that level, the performance did not significantly change. As shown in Figure 6.21, it seems that the most performance change occurred when nodes from the levels of L=3 to L=4 in the DS-Trees were included in the recovery process. The MinMax-Voronoi algorithm and DSSA have similar behaviour as nodes from levels of L=1 to L=5 in DS-Trees participate in the movement towards the the CH. If nodes are appended to the set of participating nodes beyond the aforesaid levels, the performance did not change significantly. A change of performance was more noticeable in DSSA when DS-Trees' levels of engaged nodes increased from L=1 to L=4. Beyond that level, adding more nodes from the higher levels in DS-Trees did not seem to contribute to the performance.

6.4 Conclusion

In this work, a model of cooperative recovery of CH was in which the damaged area was recovered as the result of node movement in the form of disjoint trees towards the given coverage hole. The given set of disjoint trees around the CH spans the network solely based on the nodes' local and autonomous interactions with their in-range neighbours; however, the independent movements of aforementioned DS-Tree towards the CH lead to an emergent cooperative behaviour in the recovery of large scale CH. In the first part of the chapter, the proposed node relocation algorithm for recovery was compared with one force-based and two Voronoi-based relocation algorithms. It is shown that the performance of proposed model either outperforms or matches these counterparts.

In the second part, the effect of the number participating nodes and the depth of node notification on the recovery of CHs in the given Voronoi-based and force-based node relocation algorithms was investigated. Results suggest that the performances did not seem to change significantly if the number of participating nodes and/or depth of node notification was increased beyond certain degrees. Therefore, by taking the results into account for the recovery of prospective CHs, it is possible to efficiently save energy of autonomous nodes because, in the proposed algorithm, only proper fractions of total nodes are involved/notified in the recovery process and physically moved. Such consideration in large scale would lengthen the lifespan of wireless sensor network.

CHAPTER 7

Distributed Hybrid Recovery of Coverage Hole

7.1 Introduction

Coverage Holes (CHs) can compromise the reliability and functionality of wireless sensor networks (WSNs). The recovery of CHs is challenging, especially in distributed applications where sensors have little knowledge about other sensors' actions. In this chapter, a distributed hybrid CH recovery is proposed such that an appropriate combined action of physical relocation and sensing range adjustment can be taken by each sensor to reduce the CHs in an energy-efficient way. Applying both node relocation and power adjustment in harmony allows a network recovery process to be faster, more energy efficient and more stable. Hybrid topology control schemes, such as several studies shown [182, 250, 251], enable mobile WSNs to adjust both their location and transmission power to modify or maintain optimal network coverage. The joint adjustments are important to reduce the energy requirement for recovery the CHs. As a result, the network lifetime and resilience can be improved.

Using the concept of game theory [367, 368], the proposed approach can combine different TC schemes such as relocation and transmission power control [250]. A new potential game is formulated, where nodes autonomously and efficiently decide on the proper topology control actions (i.e. physically moving or adjusting power). As decisions are based on the status of the nodes and information collected from their neighbours, the proposed CH recovery algorithm can address real-time reactions and coverage requirements, especially for networks deployed in harsh and hostile environments with no plausible centralised control. Exploiting the concept of hybrid topology control, an *(constrained) exact potential game* is formulated for sensors nodes with a sensing model of the directional footprints and finite *angle of views* [250]. Each sensor can take actions of moving or adjusting its angle of view/range based on local information (i.e. its own energy level and neighbouring nodes' actions). However, this model has a limited sensing scope due to its angle of view. Therefore, an omni-directional sensor model is considered so that a wider sensing scope can be provided, and the network can be considered for a wide range of applications and scenarios. The profit of the proposed algorithm is set to be the area exclusively covered by the nodes (see Figure 7.1), which is an effective utility function. In addition to coverage maximisation by Zhu and Martinez [250], our work examines the proposed hybrid topology control model's behaviour and efficiency in the presence of a sequence of randomly distributed *damage events* (i.e. CHs). The efficiency of the proposed algorithm in terms of network coverage and energy consumption are measured.

By proposing a new game theoretic approach for recovering the CHs in a distributed manner, a potential game between the sensors is formulated, such that each mobile sensor in the network only depends on local knowledge of its neighbouring nodes and takes CH recovery actions recursively with global convergence. Compared to the prior counterparts, the simulation results show that the proposed game theoretic approach is able to substantially increase network's lifetime and maintain network coverage in the presence of random damage events.

7.2 Method and Assumptions

Required concepts of game theory are briefly introduced in Section 7.2.1. Most necessary definitions and concepts are mainly from Maschler, Solan, and Zamir; Fudenberg and Levine; and Monderer and Shapley [367,369,370], which should be referred if more details are needed.

The system model and parameters are then presented in Table 7.1).

7.2.1 Game Theory in Brief

Definition 5 *A strategic-form (also in normal form) game is an ordered triple* $\Gamma := \langle V, A, U \rangle$ *consists of three components:*

- *A set* V *finite list (set) of players* $i \in V := \{1, \cdots, N\}$.
- *An Action set* $A := \prod_{i=1}^{N} A_i$ *is the space of all action vectors, where* $s_i \in A_i$ *is the strategy of player* i *and a (multi-player) strategy* $\in A$ *have components* $\{s_1, \cdots, s_N\}$. *Correspondingly, strategies of all players except* i*, presented by* s_{-i}*, and the set of action profiles for all players except* i *presented as* $A_{-i} := \prod_{j \neq i} A_j$.
- *Collection of utility functions* U_i*, where the utility function* $U_i := A \to \Re$ *models player* i*'s preferences over the action profile.*

In game theory, all of the players are assumed to be *rational.* A Player is considered to be rational if he/she seeks an action to maximise his/her own pay-off (utility function). Players' rationality is assumed to be common knowledge. Each utility is determined not only by each player actions but also by all the actions of other players. In game theory, while they follow the global aim, players only choose actions, they believe maximises their utility functions. Such strategy is called the *best response* and is defined as follows,

Definition 6 (Best Response [369]) *strategy $s_i^* \in A_i$ for player i is a best response, given the action of the other players, s_{-i}, if*

$$\forall s_i \in A_i, \qquad U_i(s_i^*, s_{-i}) \geq U_i(s_i, s_{-i}) \tag{7.1}$$

Based on the definition of best response for each player, *Nash Equilibrium* (NE) can be defined for a game as,

Definition 7 (Nash Equilibrium [369, 370]) *of a game is an action profile $s^* \in S$ such that every player is playing a best response to the action choices of the other players such that,*

$$\forall i \in V, \forall s_i \in A_i, \qquad U_i(s_i^*, s^*{}_{-i}) \geq U_i(s_i, s^*{}_{-i}). \tag{7.2}$$

This action named equilibrium since each player cannot obtain higher utility by deviating from the Nash equilibrium and choosing any other actions.

In a specific class of games known as *potential games*, each player's utility function can be aligned with the global aim captured by a potential function $\Phi : A \to \Re$ [370]. Potential games are important classes of games such that, the change in a player's utility caused by a unilateral deviation can be exactly measured by a potential function. The exact definition of potential games is as follows,

Definition 8 (Potential Game [370]) *A game is an exact potential game with the potential function $\phi : A \to \Re$ if, for $\forall i \in V$, and for $\forall s_{-i} \in A_{-i}$ and for every $s_i', s_i \in A$, it holds that,*

$$\phi(s_l', s_{-i}) - \phi(s_i, s_{-i}) := u_i(s_l', s_{-i}) - u_i(s_i, s_{-i}). \tag{7.3}$$

A potential game defined above, requires perfect alignment between the global aim and the players' local utility functions. This means that if a player unilaterally changed his/her action, the change in his/her objective function would be equal to the change in the potential function. Learning algorithms for potential games have been studied comprehensively in the game theory literature [368, 371, 372]. Potential games have some other interesting properties. As an example, for any given game, Nash equilibrium may not exist. However, for potential games, the existence of a Nash equilibrium is guaranteed [370]. In general games, all the actions in A can always be selected by player i. However, in many cases, the available actions to the player i will often be constrained to state of the game as $F_i(s_i, s_{-i}) \in A$. Respectively, the *Constrained Nash Equilibrium* of the game can be defined as follows,

Definition 9 (Constrained Nash Equilibrium [250]) *An action s^* is a constrained Nash equilibrium of the game if $\forall i \in V$ and $\forall s_i \in F_i(s_i, s_{-i})$ it holds that $U_i(s^*, s^*{}_{-i})) \geq U_i(s_i, s^*{}_{-i}))$.*

Similarly, the constrained potential game can be defined as follows,

Definition 10 (Constrained Potential game [250]) *a game is a constrained potential game with potential function ϕ if for $\forall i \in V$, and for $\forall s_{-i} \in A_{-i}$ and for every $\forall s'_i, s_i \in F_i(s'_i, s_i)$,it holds that*

$$\phi(s'_l, s_{-i}) - \phi(s_i, s_{-i}) := u_i(s'_l, s_{-i}) - u_i(s_i, s_{-i}). \tag{7.4}$$

Constraint potential games inherit properties of potential games such as; constrained potential game with assumption of $s_i \in F_i(s'_i, s_i)$ for any $s_i \in A_i$, has at least one constrained Nash equilibrium [250].

7.2.2 Sensor Nodes

Similar to Section 3.2.1.1, nodes are assumed to be deployed in a 2D rectangular area, denoted by $\theta = [x_{\min},\ x_{\max}] \times [y_{\min}, y_{\max}]$. Each sensor node i can be modelled as a Unit Disk Graph (UDG) with a transmission range of R_c^i and sensing range of R_s^i. Let $\mathcal{V}$ presents the indices of the nodes where $i \in \mathcal{V} = \{1, 2, \cdots, \mathcal{N}\}$. Each node's location is determined by its disk's centre,

$$C_i = \{(x_i, y_i)|x_{min} \leq x_i \leq x_{max}, \quad y_{min} \leq y_i \leq y_{max}\}.$$

Nodes' locations and movements are constrained within the boundaries of the given area of deployment. For simplicity, all nodes are assumed to have the same transmission range of $R_c^i = R_c, \ \forall i \in \mathcal{V}$. As their transmission ranges are larger than their sensing ranges in this model, nodes are considered to be connected. In the proposed model, let E_i presents the finite energy of node $i \in \mathcal{V}$.

$$\mathcal{N}_i = \{j \in \mathcal{V} \backslash \{i\}, \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq R_c\}.$$

7.2.3 Coverage Holes

Similar to Section 3.2.2, coverage hole k is considered to be a circle of a radius r_{Hole_k} with the centre of (x_{Hole_k}, y_{Hole_k}).

7.2.4 Node Types

Deployed nodes are classified as either damaged nodes (D-nodes) and undamaged nodes (U-nodes), as discussed in Section 3.2.3.

7.2.5 Proposed Hybrid Recovery Algorithm

In this section, a new game theoretic approach to recover CHs is proposed. The new approach is a potential game, and can be implemented by a new learning-based algorithm with guaranteed global convergence and stability of the proposed potential game.

7.2.6 Coverage Problem Formulation

In this problem, sensors are allowed to simultaneously interact with one another to maximise their coverage areas. Each sensor (selfishly) tries to maximize its utility function based on its (local) vision of the network, environment and also its belief on the actions that other players (e.g. its neighbours) will take. Mutual interactions can be modelled as a *game* in which the sensors are players, and at each time step t, the game is repeated. At step $t > 0$, each sensor $i \in \mathcal{V}$ selects an action $s_i \in A_i$ to maximize its expected utility. A_i is the set of actions that node i can take. Each player's utility is not only determined by its action but also by all (or a subset of) the other players' actions.

The proposed game is a potential game where the change in any player's utility from a unilateral deviation exactly is matched by the change in the global potential function [368]. Potential games seem to be suitable candidates to solve distributed optimization problems when only local information is available but optimizing the global objective is desirable.

Each mobile node i, $i \in \mathcal{N}$, can take a combined action of changing its position and its sensing range $s_i = (m_i, r_i) \in A_i$:

- $m_i = (x_i, y_i) \to (x'_i, y'_i)$ denotes the change of its position, where (x_i, y_i) and (x'_i, y'_i) are the coordinates of the current and the next positions of node i, respectively.
- $r_i, r'_i \in [R_{\min}, R_{\max}]$ presents the change of the sensing range of node i, where r_i and r'_i are the current sensing range and the sensing range that the node is to take at the next time instant. $R_{\min}$ and $R_{\max}$ are the minimum and the maximum sensing ranges of the nodes, respectively.

The *Utility Function* of node i is defined as

$$u_i(s_i, \mathbf{s}_{-i}) = w_1 \times P(s_i, \mathbf{s}_{-i}) - w_2 \times C(s_i), \tag{7.5}$$

where $P(s_i, \mathbf{s}_{-i})$ and $C(s_i)$ denote the profit and cost of strategy s_i at node i, respectively. w_1 and w_2 are the weights associated with the profit and cost to balance these two aspects, and to adjust the speed of the game converging to an equilibrium. The incentive of all players to change their strategy can be expressed via a single global function called the *potential function*.

The utility (pay-off) function determines respectively on the profit a node should gain and cost should be paid when it covers an area. That profit should be high enough to motivate sensor nodes to contribute to CH recovery, while the cost should be sufficient to avoid possible redundant operations and to reduce unnecessary coverage overlaps. So, in order to maximize the sensing coverage of the network, each node aims to reduce the overlapping sensing coverage areas with its neighbour nodes. For this purpose, we define the profit for node i, $P(s_i, s_{-i})$, to be the area covered only by the node (as illustrated by Fig. 7.1), as given by

$$P(s_i, s_{-i}) = |D_i \backslash \bigcup_{j \in \mathcal{N}_i} D_j|. \tag{7.6}$$

Two types of costs, both in the form of energy consumption, are considered. One corresponds to the change of the sensing power of a node, denoted by $C_p(s_i)$. The other corresponds to the change of the node's position, denoted by $C_T(s_i)$ The cost of changing the sensing power of node i can be defined by

$$C_P(s_i) = \begin{cases} e^{|\Delta p_i T|} + c_1, & \Delta p_i > 0; \\ 0, & \Delta p_i = 0; \\ -e^{-|\Delta p_i T|} + c_1, & \Delta p_i < 0, \end{cases} \tag{7.7}$$

where T is the expected interval duration between two consecutive CHs, $|\cdot|$ stands for absolute operation, and the constraint c_1 is introduced to prevent small un-

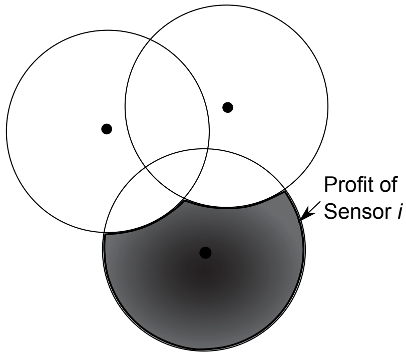


Figure 7.1: Profit of sensor i

necessarily frequent power changes in the network. Note that the exponential function is used to define the cost to discourage excessively large changes in the sensing range of a sensor. This is because the cost grows much faster than (to be more specific, exponentially with) the increase of the sensing range. It should be noted that, in the case of $\Delta p_i < 0$, the cost takes negative values and becomes a reward. This is the incentive for a node to move around to reduce its excessively high sensing power, thereby extending the lifetime of the node.

The cost of changing the position of the node can be defined as

$$C_T(s_i) = \begin{cases} e^{\Delta a_i} + c_2, & |\Delta a_i| \neq 0; \\ 0, & |\Delta a_i| = 0, \end{cases} \tag{7.8}$$

where Δa_i is the distance that node i moves, and the constant c_2 is used to prevent unnecessarily frequent small movements and oscillations. The exponential function is also used to define the cost of changing the location of a sensor to discourage the sensor to move excessively far.

Therefore, the total cost can be given by

$$C(s_i) = k_1 C_P(s_i) + k_2 C_T(s_i), \tag{7.9}$$

where k_1 is the weighting coefficient of the cost for changing the sensing power, and k_2 is the weighting coefficient of the cost for changing the location.

Note that the cost of node i only depends on its strategy s_i, and is independent of the other nodes' strategies $\mathbf{s}_{-i}$. Hence,

$$C(s_i, \mathbf{s}_{-i}) = C(s_i).$$

Theorem 1 *The defined coverage game is a constrained potential game with the following potential function*

$$\phi(\boldsymbol{s}) = w_1 S(\boldsymbol{s}) - w_2 C(\boldsymbol{s}), \tag{7.10}$$

where $\boldsymbol{s} = \{s_i, \mathbf{s}_{-i}\}$ collects the strategies of all the nodes, and $S(\cdot)$ denotes the overall coverage of the sensors.

Proof. 1 *The global aim of the proposed coverage game is to maximise the coverage with the minimum cost. The goal can be defined in the closed form, as given in Equation 7.10. A game is defined to be an **exact potential game** [370], [372] if there is a function $\phi : A \to \mathbb{R}$ such that $\forall \boldsymbol{a}_{-i} \in A_{-i}$, $\forall a_i', a_i'' \in A_i$,*

$$\phi(a_i', \boldsymbol{a}_{-i}) - \phi(a_i'', \boldsymbol{a}_{-i}) = u_i(a_i', \boldsymbol{a}_{-i}) - u_i(a_i'', \boldsymbol{a}_{-i}).$$

In other words, when player i switches from action a_i' to a_i'', the change in the potential function equals to the change in the utility of that player. Following this definition, it can be proved that the proposed game is an exact potential game by varying the action of one of the nodes while keeping those of all the others unchanged.

In the case that node i alone changes its strategy from s_i to s_i', the goal or potential *of the proposed game changes as*

$$\begin{aligned}
&\phi({s'}_i, \mathbf{s}_{-i}) - \phi(s_i, \mathbf{s}_{-i}) \\
&\quad = w_1\Big(S(s'_i, \mathbf{s}_{-i}) - S(s_i, \mathbf{s}_{-i})\Big) \\
&\qquad\qquad - w_2\Big(C({s'}_i, \mathbf{s}_{-i}) - C(s_i, \mathbf{s}_{-i})\Big) \\
&\quad = w_1\Big(S({s'}_i, \mathbf{s}_{-i}) - S(s_i, \mathbf{s}_{-i})\Big) - w_2\Big(C({s'}_i) - C(s_i)\Big).
\end{aligned}$$

In this case, the change of the overall coverage area by shifting from s_i to s'_i is equivalent to the change of the area that is only affected by sensor i (and not by any other neighbouring sensors). The change in the coverage of node i in the overlapped area does not have an impact on the overall coverage, as this area is already covered by at least one other sensor node. Referring to Equation 7.6,

$$S(s'_i, s_{-i}) - S(s_i, s_{-i}) = P(s'_i, s_{-i}) - P(s_i, s_{-i}).$$

As a result, it can be proved that

$$\begin{aligned}
&\phi({s'}_i,\mathbf{s}_{-i}) - \phi(s_i, \mathbf{s}_{-i}) \\
&\quad =w_1\Big(P(s'_i, \mathbf{s}_{-i}) - P(s_i, \mathbf{s}_{-i})\Big) - w_2\Big(C({s'}_i) - C(s_i)\Big) \\
&\quad =u_i({s'}_i, \mathbf{s}_{-i}) - u_i(s_i, \mathbf{s}_{-i}),
\end{aligned}$$

In other words, the incentive of any sensor to change its strategy can be expressed using the single global potential function. Thus, the proposed game is proved to be a potential game. ■

7.2.7 Distributed Payoff-based learning algorithm

The utility of each node depends on the action of its neighbouring nodes. The nodes are unable to access the utility values in advance. Therefore, the action-based learning algorithms, such as the better (or best) reply learning algorithm and adaptive play learning algorithm, cannot be employed to solve this problem [250]. Hence, a distributed learning algorithm that only requires the pay-off received from the previous steps is the most suitable. The players adjust their behaviours based on the observed behaviours of other players. In particular, players know neither

Name of variable	Description
i	Index of Sensor Nodes
N	Maximum Number of Nodes
θ	Deployment area
$\mathcal{V} = \{1, \cdots, N\}$	Set of Sensor Nodes
R_s^i	Sensing Range of Node i
R_c^i	Transmission Range of Node i
E_i	Energy of Node i
$\mathcal{N}_i^{loc}$	Neighbours of Node i
A	Action Set
A_i	Action Set of Node i
$\boldsymbol{s_i}$	Strategy i
D_i	Coverage Area of Node i
m_i	Control Vector for Moving Action of Node i
r_i	Control Vector for Range changing Action of Node i
ϕ	Potential Function
$\epsilon(t, E)$	Exploration rate based on time and energy
$\tau_i(t)$	More successful Node i's action of in last two steps

Table 7.1. Description of Variables

the actions taken by other players nor the structural form of pay-off functions (please refer to Marden et al. [368]).

In this section, a distributed payoff-based learning algorithm, non-trivially extended from Zhu and Martinez's work [250], is proposed to implement the formulated potential game. In the proposed algorithm, for each $t \geq 0, i \in \mathcal{V}$, we can define $\tau_i(t)$ as the more successful actions of sensor i in the last two steps such that:

$$\tau_i(t) = \begin{cases} t, & \text{if } u_i(\boldsymbol{s_i}(t)) \geq u_i(\boldsymbol{s_i}(t-1)); \\ t-1, & \text{Otherwise;} \end{cases} \tag{7.11}$$

The proposed algorithm can be described as follows (Algorithm 7.1):

ALGORITHM 7.1: Proposed Algorithm

1. **Initialization:** At $t = 1$, all sensors keep their initial situations.

2. **Update:** At each time $t \geq 2$, each sensor i updates its state according to the following rules:

    - Sensor i chooses the exploration rate $\epsilon \in (0, 0.5]$ and computes $s_i(\tau_i(t))$.
    - With probability ϵ, sensor i computes utility values of strategies, (m_i^{tp}, p_i^{tp}), (m_i, p_i^{tp}), or (m_i^{tp}, p_i) which are uniformly selected from the set A_i. One of these strategies with highest utility value is selected as temporary strategy, $\boldsymbol{s_i^{tp}}$ of sensor i. Here for example (m_i, p_i^{tp}) means that sensor i, kept its current location but changes its sensing power to p_i^{tp}.
    - With probability $1 - \epsilon$, sensor i does not experiment $\boldsymbol{s_i^{tp}} = \boldsymbol{s_i}(\tau_i(t))$ and keeps the current strategy at time t for time $t + 1$.
    - Sensor i acts based on $\boldsymbol{s_i^{tp}}$.

3. **Repeat:** Sensor i executes step 2 until ending condition is met.

In [250], a similar learning algorithm is proposed for the potential game that is able to find a Nash equilibrium (NE) with an arbitrary high probability by choosing an arbitrarily small and fixed exploration rate of ϵ in advance. Convergence is guaranteed even in situations where the baseline action may not be feasible when the state-dependent constraints potential games are present. In [250], convergence is proven by showing that the time series that comes from $\tau_i(t)$, named as $\{\mathcal{P}_t^{\epsilon}\}$, is a time homogeneous Markov chain, irreducible and aperiodic. Subsequently, with respect to the properties of potential games, for each ϵ, there exists a unique stationary distribution of $\{\mathcal{P}_t^{\epsilon}\}$ as $\mu(\epsilon)$. The existence of limiting distribution of $\mu(\epsilon)$ is proved in [250]. Exactly the same process can be used to prove the convergence of the learning algorithm proposed here because both the proposed model and that in Zhu and Martinez's work [250] are based on the safe experiment dynamics with a small exploration rate of ϵ.

Different from [250], general mobile sensors capable of omni-directionally monitoring ambient environments are considered in this chapter. For example, the sensor can be a radar using a continuous wave, where the sensing range depends on its transmit power. The cost is defined to be the energy consumption required to move a sensor and to change the sensing range of the sensor. Minimizing the total cost can extend the lifetime of the sensor and in turn, that of the entire network. Particularly, adjusting the sensing is encouraged to extend the network lifetime by avoiding excessively frequent sensor movements. In contrast, in [250], a special type of mobile sensor equipped with directional video cameras, is considered, where the cost consists of the energy of moving a sensor as well as the energy of maintaining the sensing coverage area. Changing the sensing range is not encouraged, even when CHs occurs frequently and moving the sensors around frequently can be energy inefficient. In addition, in step 2 of the *Distributed Homogeneous Synchronous Coverage Learning Algorithm* (DHSCL) [250], sensors choose a temporary strategy (m_i^{tp}, p_i^{tp}) with probability ϵ, while here in this chapter, the sensors can choose from three options based on the utility of these strategies.

Also note that this chapter is emphasises on the algorithm design, which lays the foundation for practical implementations. The presented proof of the proposed algorithm being a potential game can guarantee, from a theoretical point-of-view, that the algorithm is convergent with a unique Nash equilibrium. Our extensive simulations also validate the proposed algorithm and corroborate its effectiveness. (see Section 7.3).

7.3 Performance Evaluations

7.3.1 Performance Metrics

In this section, performance metrics are defined in terms of *percentage of Coverage*, *energy consumption*, and *efficiency of consumed energy (ECE)*.

Percentage of Coverage: is defined according to Section 3.3.1.2 as the number of grid cells that are covered by at least one of the sensor nodes over the total number of grid cells in the given deployed area. Grid cells, which are covered simultaneously by k sensors nodes, are defined as grid cells having k-*Coverage* in the area of deployment.

ECE: The ECE is defined as average recovered coverage area over energy consumed during the course of actions by the sensor nodes. It is given by

$$ECE = \left(\frac{\sum_{t} \sum_{i=1}^{N} \Delta C_{ij} \cdot \delta t_j / E_T}{N} \right), \tag{7.12}$$

where E_T is the total consumed energy (power range change in addition to movement). C_{ij} is the coverage change during a given duration of time $t = t_k + \delta t_j$. This is because, some algorithms, such as DSSA [227], neither consume energy nor recover the (large) CHs. So it is sensible to consider the energy and coverage metric simultaneously. Total Energy E_T, is considered as the sum of the consumed energy due to the change of power E_P and the changing location of nodes E_M as follows,

$$E_T = E_P + E_M$$

Name of variables	Value of Variables
Parameters of Network	
$N : Number of Nodes$	$[60, 500]$
θ	$[-100, 100] \times [-100, 100]$ m^2
R_s^i	$U \sim [7, 15]$ m
R_c^i	30 m
Parameters of Experiments	
N_{Trials}	50
Parameters of Coverage Holes	
Number of Damages	5
Damage Events time sequences	1 s, 102 s, 237 s, 445 s, 545 s
R_{Hole}	$U \sim [40,\ 80]$ m
k_1	0.01
k_2	100

Table 7.2. Value of Variables

- *Energy Related to Power Change*: here we only consider the energy consumption after the system is stabilised. The energy consumption is given by

$$E_P = \sum_t \sum_{i=1}^{n} ((4\pi . R_s^i/\lambda)^2 \cdot \delta t_j),$$

- *Energy Related to Movement* is defined as

$$E_M = \sum_t \sum_{i=1}^{N} (Movement(i) \cdot \delta t_j).$$

where R_s^i is the sensing range, i is the node index, and $\lambda = 0.125$.

7.3.2 Results

The proposed algorithm was simulated with Matlab. Sensor nodes were deployed randomly in a 2D-rectangular area θ with a uniform distribution. The transmission range was set to 30 m and the nodes' sensing ranges were initialised randomly between 7 m to 15 m with uniform distributions (see Table 7.2). The proposed algorithm was compared with DSSA [227] and DHSCL [250]. The algorithms' performance was examined in terms of percentage of coverage, consumed energy and efficiency of consumed energy (please refer to Tables 7.5 and 7.4). An exploration rate of $\epsilon = 0.3$ was used for all the experiments. The performance of our proposed model and two benchmarks were examined by different number of nodes, N, ranging from 60 to 500 nodes.

For a fair comparison, we extended and simulated the DHSCL algorithm developed by Zhu and Martinez [250] to our scenario with omni-directional sensing ranges where the cost consists of the energy of moving a sensor and the energy of maintaining the sensing coverage area (as specified in the work by Zhu and Martinez [250]). In this chapter, the cost associated with node movement, and the energy for the sensor to change its sensing range are defined such that the sensor is encouraged to increase its sensing range to avoid unnecessarily frequent small movements especially in the presence of frequently occurring CHs.

For each experiment, 50 trials were conducted in order to have unbiased results. Each experiment consisted of 5 consecutive random failures uniformly distributed in the deployment field. The damage event time sequences were 1 s, 102 s, 237 s, 445 s, and 545 s. After each occurrence of CHs, the algorithms remained active till the overall coverage of 95% was achieved. If such criteria was not met due to current density and energy levels of the operational nodes in the network, then the algorithms ceased to work once they reached to 500 iterations. Both conditions were devised, so the algorithms' *stopping criteria* preserved nodes' residual energy. Table 7.3 shows the average of the required iterations after the occurrence of each successive event for $N = 200$ and 400 nodes in the given algorithms to satisfy the stopping criteria.

Figure 7.2 uses a snapshot example to demonstrate the ability of our proposed game theoretic approach for covering CHs. Specifically, the colours in the figure indicate the number of sensors that can sense a spot. As the number decreases, the colour turns to blue; as it increases, the colour turns to red. Dark blue means that a spot did not have sensing coverage from any sensor. The reducing dark blue CH in the figure shows that our proposed model efficiently recovered and maintained the coverage of the network after consecutive CH events.

Five CHs occurred in a series, as shown in Figures 7.2(b), 7.2(d), 7.2(f), 7.2(h) and 7.2(j). The CH recovery, driven by the proposed potential game, is shown in Figures 7.2(c), 7.2(e), 7.2(g), 7.2(i) and 7.2(k) in response to the five CHs, respectively. As shown in the figures, it can be seen that the sensors moved towards the CHs and adjusted their sensing ranges in order to quickly and effectively cover the CHs. Figure 7.3 presents the algorithms' ability to recover from the sequential CH events (in terms of percentage of coverage) for $N = 400$ nodes. It can be seen that our proposed algorithm outperformed the other CH recovery strategies and maintained the coverage of the network. Furthermore, as nodes fail due to consecutive CHs and energy exhaustion, the advantage of our proposed model become more noticeable with respect to recovered coverage after each CH.

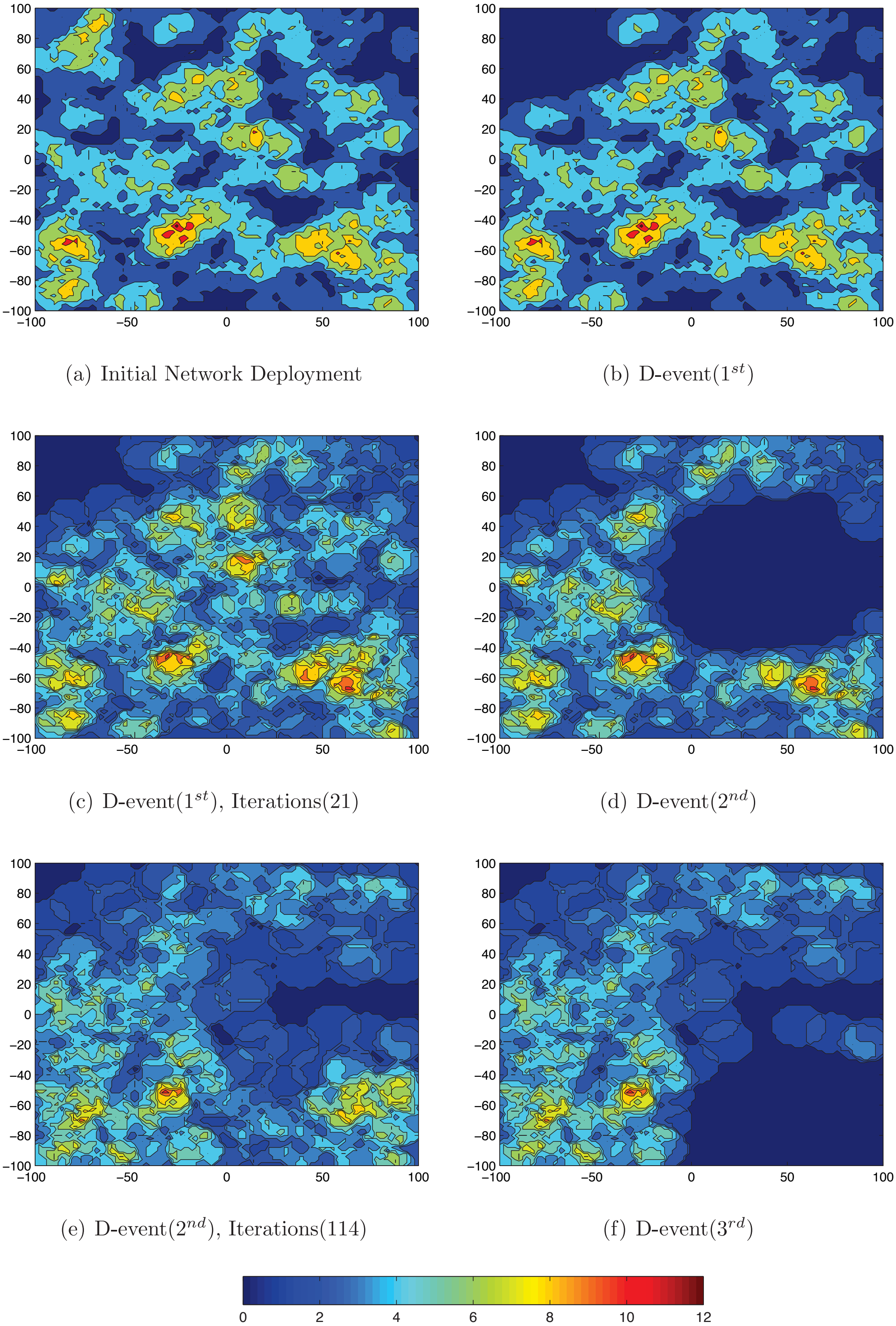
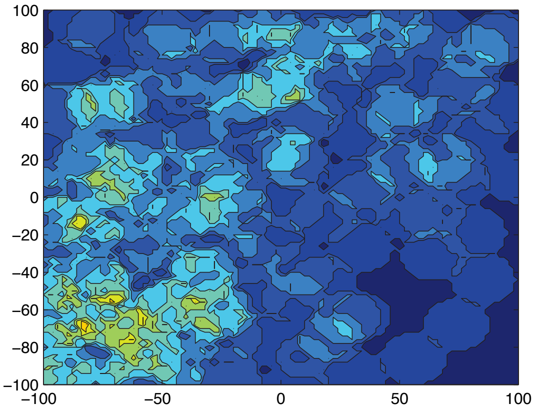
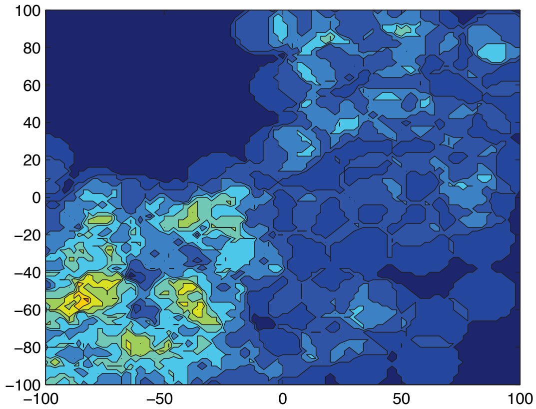


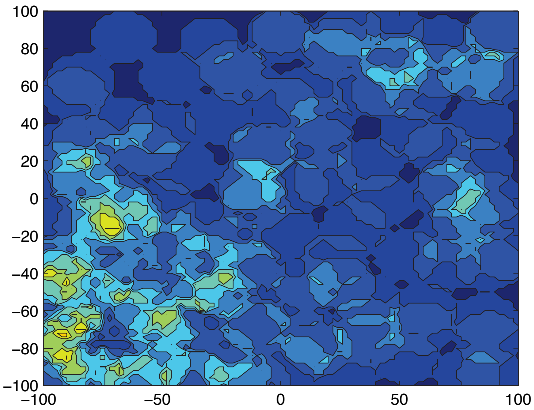
Figure 7.2: k-Coverage of Network $N = 400$ Nodes and Random Consecutive Damage Events (Coverage Holes)



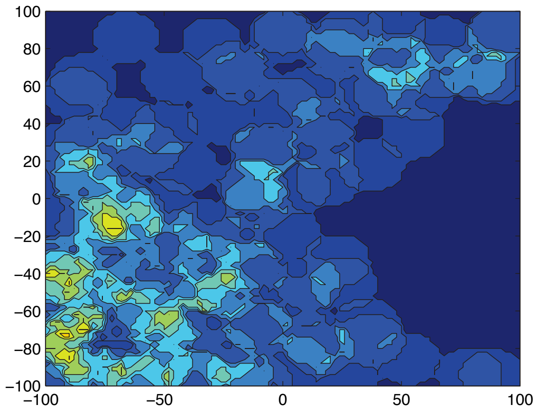
(g) D-event(3^{rd}), Iterations(300)



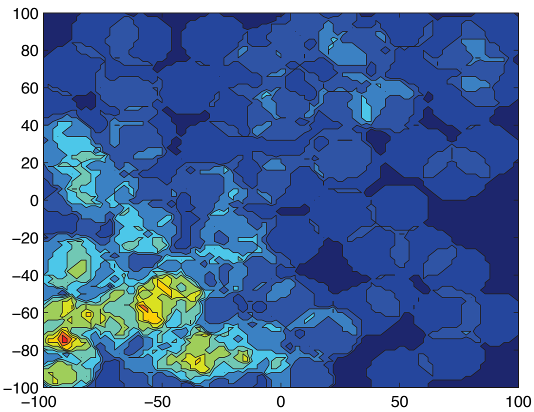
(h) D-event(4^{th})



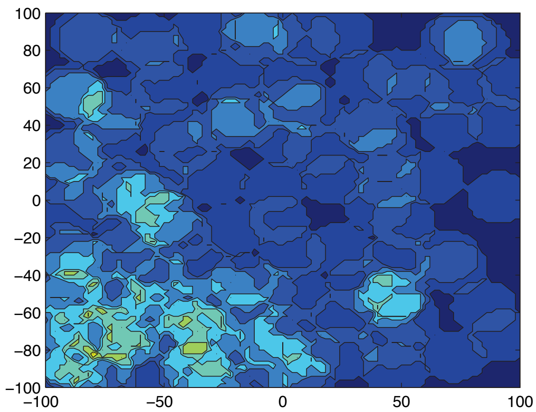
(i) D-event(4^{th}), Iterations(388)



(j) D-event(5^{th})



(k) D-event(5^{th}), Iterations(200)



(l) D-event(5^{th}), Iterations(500)

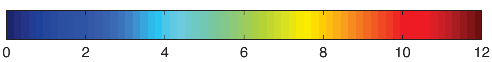


Figure 7.2: k-Coverage of Network $N = 400$ Nodes and Random Consecutive Damage Events (Coverage Holes) (Cont'd.)

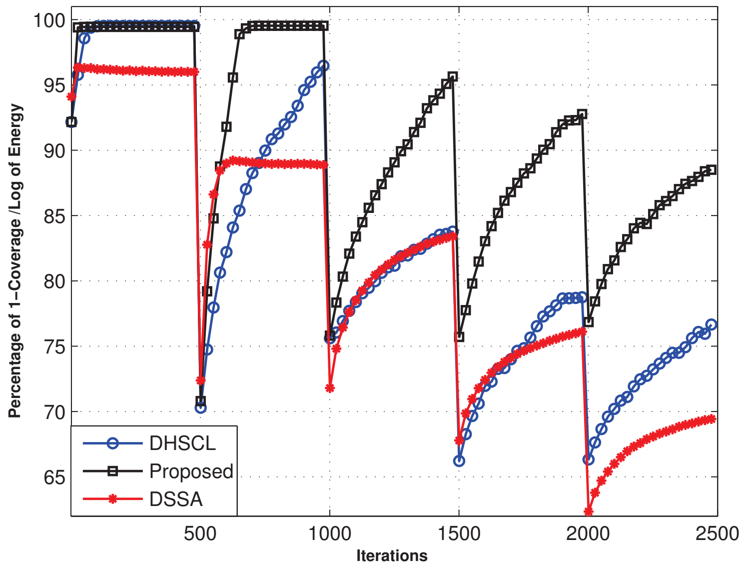


Figure 7.3: Algorithms' Percentage of Coverage vs Processing Time, Iterations $N = 400$ Nodes

Algorithms		Event 1		Event 2		Event 3		Event 4		Event 5	
	# Nodes:	200	400	200	400	200	400	200	400	200	400
DHSCL	# Iterations	488.66	37.76	500.00	444.40	500.00	500.00	500.00	500.00	500.00	500.00
Proposed	# Iterations	45.32	14.42	318.94	136.58	500.00	445.02	500.00	465.06	500.00	489.30
DSSA	# Iterations	500.00	222.46	500.00	500.00	500.00	500.00	500.00	500.00	500.00	500.00

Table 7.3. Value of Stopping Criteria for $N = 200, 400$ Deployed Nodes

Figure 7.4 shows the average percentage of coverage for the DSSA, DHSCL and the proposed algorithm with respect to the number of deployed nodes (60 to 500 nodes). The average coverage was computed based on the average coverage for the entire area during 5 CHs over 50 trials. The figure shows that the proposed algorithm maintained higher coverage.

Table 7.4 shows the efficiency of the proposed approach for $N = 100, \cdots, 500$, as compared to DSSA and DHSCL. The results show that the proposed algorithm

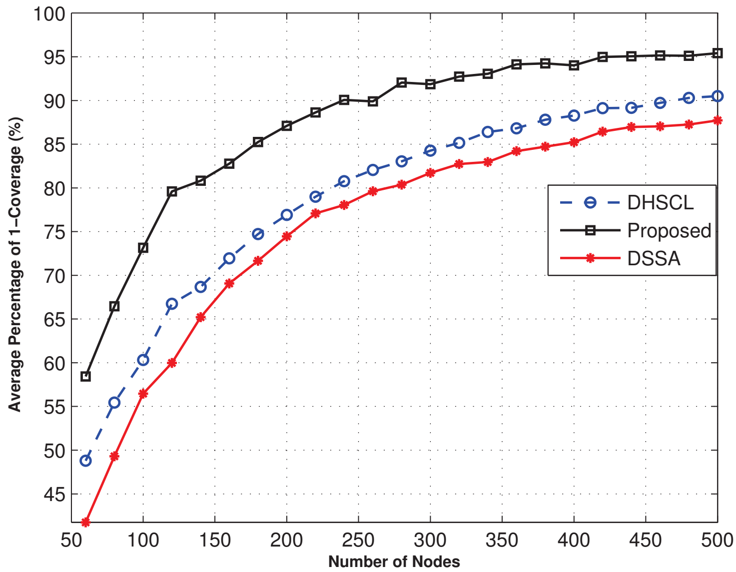


Figure 7.4: Percentage of Avg . Coverage vs Number of Nodes

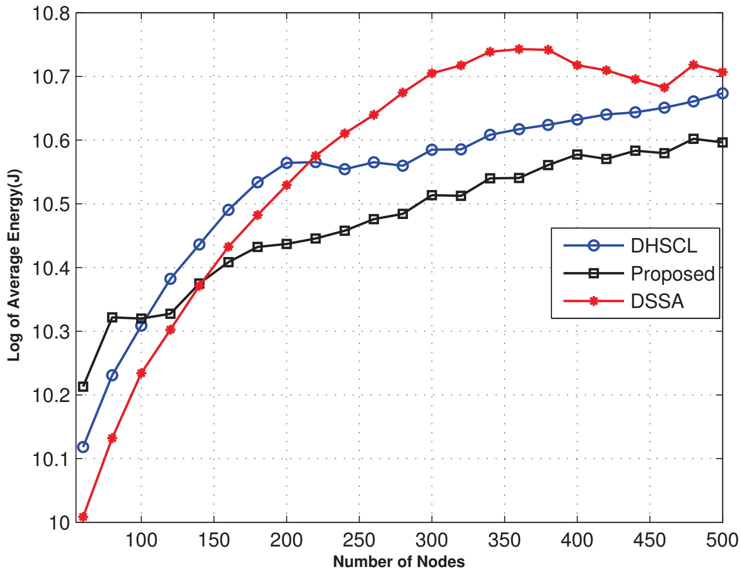


Figure 7.5: Algorithm Consumed Energy vs. Number of Nodes

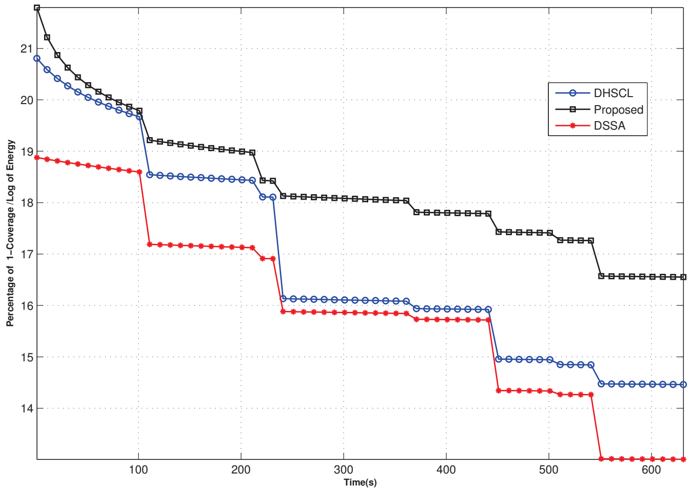


Figure 7.6: Algorithms' Percentage of Coverage/Energy vs Time, $N = 400$ Nodes

outperformed DSSA and DHSCL by up to 30% and 22% for $N = 100$, respectively, and up to 9% and 5.5% for $N = 500$, respectively. Table 7.5 presents the efficiency of our algorithm with respect to the occurrence of CHs.

Figure 7.5 shows the performances of DSSA, DHSCL and the proposed algorithm in term of energy consumption. The average energy was computed based on the total energy consumption during the simulation over 50 independent trials. The results showed that our proposed algorithm consumed slightly more energy for $N < 140$ nodes (see Table 7.4). The additional energy is consumed for more efficient coverage and recovery of the damaged areas after occurrences of CH events. Table 7.5 compares the efficiency of our algorithm with two benchmarks in terms of consumed energy before and after occurrence of the $(1^{st}, \cdots, 5^{th})$ CH events using "$-$" and "$+$" signs accordingly.

As the number of nodes increases, our proposed algorithm continued to outper-

(%) of Improvement of proposed Algorithm			
Performance Metrics	# Nodes	DSSA	DHSCL
Coverage	100	29.58	21.31
	200	16.99	13.25
	300	12.43	9.03
	400	10.30	6.49
	500	8.78	5.42
Energy	100	−0.83	−0.11
	200	0.88	1.20
	300	1.79	0.69
	400	1.31	0.52
	500	1.03	0.72
Energy Consumption Efficiency	100	21.76	15.85
	200	12.88	10.09
	300	9.52	6.95
	400	7.92	5.02
	500	6.77	4.20

Table 7.4. Avg. Coverage and Energy vs. Number of Deployed Nodes

form other benchmarks (see Figure 7.5 and Table 7.4). Table 7.5 shows such trends throughout the events. It should be noted that in Figures 7.4 and 7.5, DSSA did not response swiftly to damage events as sluggish node movements led to slow recovery from the consecutive CHs when compared to DHSCL and the proposed algorithm. Therefore, in DSSA the energy consumption was lower than the hybrid topology control schemes. As the number of nodes reduced due to energy exhaustion and random sequential damage events, the difference in energy consumption between DSSA and the other algorithms also reduced.

Figures 7.6 shows the percentage of improvement after the occurrence each CH event for the proposed algorithm with respect to the benchmarks. As shown in Figure 7.6, our proposed algorithm rendered the network more resilient and efficient than the DSSA and its hybrid counterpart as the number of randomly

(%) of Improvement of proposed Algorithm											
Metrics	Event [1]	# Nodes: 100		# Nodes: 200		# Nodes: 300		# Nodes: 400		# Nodes: 500	
		DSSA[2]	DHSCL[3]	DSSA	DHSCL	DSSA	DHSCL	DSSA	DHSCL	DSSA	DHSCL
Coverage	1_{st}	37.43	26.33	11.75	3.84	5.94	0.04	1.99	0.00	−0.03	0.01
	2_{nd}	43.43	33.68	21.10	17.05	11.27	6.71	7.10	0.89	4.30	0.13
	3_{rd}	47.13	35.79	22.66	22.34	15.99	15.78	12.24	11.40	9.97	9.18
	4_{th}	39.75	31.80	27.14	26.42	21.33	20.07	20.54	16.23	18.47	12.20
	5_{th}	44.65	17.83	32.51	20.86	28.23	18.12	26.80	14.44	25.71	13.54
Energy [4]	1^-_{st}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	1^+_{st}	1.56	2.25	9.27	9.46	12.16	7.13	11.67	4.54	1.86	3.92
	2^-_{nd}	0.77	1.46	5.04	5.25	5.98	2.27	4.16	0.60	0.21	0.28
	2^+_{nd}	−0.37	0.30	2.63	2.89	4.85	3.11	4.29	2.75	3.34	2.34
	3^-_{rd}	−0.82	−0.13	0.94	1.25	2.05	0.98	1.69	0.87	1.44	1.01
	3^+_{rd}	−0.82	−0.13	0.94	1.25	2.04	0.98	1.69	0.87	1.44	1.01
	4^-_{th}	−0.96	−0.30	0.35	0.65	1.08	0.31	0.81	0.27	0.82	0.59
	4^+_{th}	−0.96	−0.33	0.35	0.65	1.08	0.31	0.81	0.27	0.82	0.59
	5^-_{th}	−1.01	−0.33	0.10	0.44	0.67	0.07	0.40	0.04	0.43	0.34
	5^+_{th}	−1.01	−0.33	0.10	0.44	0.67	0.07	0.40	0.04	0.43	0.34
ECE	1^-_{st}	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	1^+_{st}	39.61	29.24	18.62	10.44	20.62	7.73	15.46	4.76	1.86	4.09
	2^-_{nd}	38.50	28.21	13.34	5.54	12.68	2.36	6.41	0.61	0.18	0.29
	2^+_{nd}	42.90	34.09	6.25	2.97	16.94	10.14	11.91	3.72	7.91	2.53
	3^-_{rd}	42.26	33.52	4.44	1.26	13.60	7.76	8.94	1.75	5.82	1.15
	3^+_{rd}	45.93	35.62	1.22	1.26	18.41	16.93	14.17	12.38	11.58	10.29
	4^-_{th}	45.73	35.38	0.61	0.66	17.26	16.14	13.15	11.71	10.89	9.82
	4^+_{th}	38.43	31.41	0.91	0.66	22.26	20.44	21.52	16.55	19.45	12.87
	5^-_{th}	38.35	31.37	0.66	0.44	22.15	20.16	21.02	16.28	18.99	12.59
	5^+_{th}	43.20	17.45	9.74	0.44	29.10	18.21	27.31	14.48	26.26	13.93

[1] Damage Events time sequences (1 s, 102 s, 237 s, 445 s, 545 s)
[2] Distributed Self-Spreading Algorithm(DSSA) [227]
[3] Distributed Homogeneous Synchronous Coverage Learning Algorithm(DHSCL) [250]
[4] Energy unit is in decibel (dB)

Table 7.5. Coverage and Energy vs. Number of Deployed Nodes

distributed damage events and node failures grow.

7.4 Conclusion

This chapter presented a novel hybrid CH recovery algorithm for Mobile WSNs. The proposed algorithm combined sensing power control and physical node relocation using a game theoretic approach. The results from investigating the performance of proposed approach over a number of different nodes densities and CH events via detailed simulation showed that the proposed approach outperformed the other CH recovery algorithms both in terms of percentage of coverage and energy consumption.

As part of future work, in order to improve the technical maturity of the proposed algorithm, some practical constraints, such as limiting the movement of a sensor to certain direction (based on the settings of wheels), could be taken into account. An adequate communication protocols to support the proposed algorithm could also be devised.

CHAPTER 8

Sink-Based Recovery Model

8.1 Introduction

Systematic management of networks' faults in different applications [15, 16] provides reliable and robust quality of service in wireless sensor networks (WSNs) in the case of node failures especially (large scale) Coverage Holes (CHs) with more severe effects WSNs [24, 270]. The integration of mobility into sensor nodes and design of different mobility patterns can provide effective solutions to the problem of network topology dynamism and the formation of damage-induced CHs [38, 41, 43, 134, 215, 231, 357]. Hence distributed node relocation algorithms [38, 215, 227, 357] are a promising class of topology control (TC) schemes in WSNs to reduce the impact of random node failures and CHs in harsh and hostile environments.

In the proposed model, by considering the special role of network sink nodes, the aforementioned node relocation algorithms can be customised to address not only the primary goal of maintaining/extending the network's coverage but also satisfying some of the emerging unique requirements in WSNs. Benefiting from

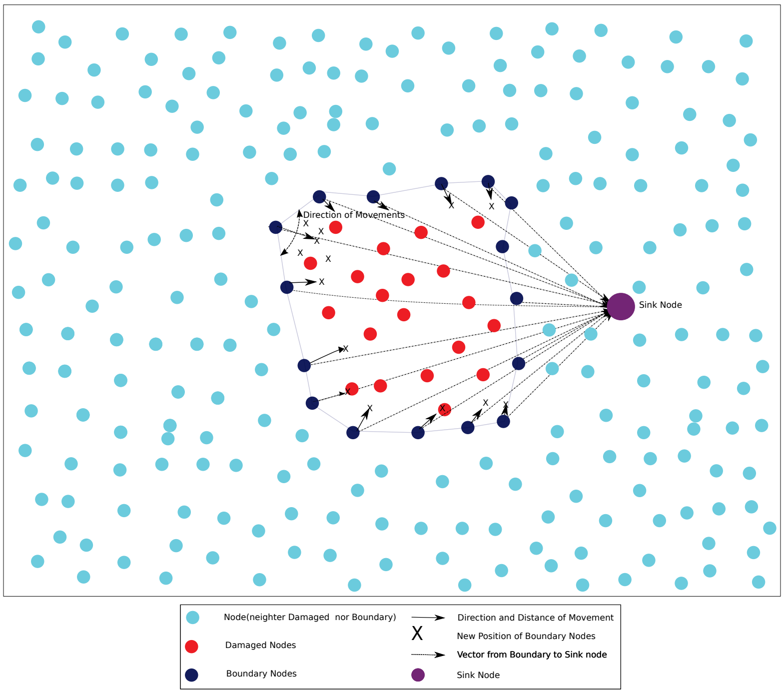


Figure 8.1: Relocation of Boundary nodes in Sink-based CH Recovery Model

the simple geometrical properties of circle inversion (Section 8.2.5 and Figure 8.4) [373, 374], a sink-based CH recovery model is proposed such that nodes try to modify their movements with respect to their ranges and distances to the proximate sink nodes while relocating towards the CHs. The general idea of the proposed recovery is illustrated in Figure 8.1 for one coverage hole in the network.

In this chapter, by using simple geometrical properties of circle inversion, a cooperative CH recovery model is presented. Autonomous nodes in this model use their distances to the deployed sink nodes and their local status in order to relocate toward the coverage holes (damaged areas) in the network. The performance of the proposed model is compared with three node relocation benchmarks.

8.2 Methods and Assumptions

In this section, model parameters are defined and the deployment scenario is described in detail.

8.2.1 Sensor Nodes and Area of Deployment

Homogeneous sensor nodes modelled as the unit disk graph (UDG) [348] are deployed with a 2-D uniform random distribution across an area of $[x_{min}, x_{max}] \times [y_{min}, y_{max}]$. For simplicity, node transmission and sensing ranges of R_c and R_s are considered to be equal. Two nodes are bidirectionally connected if they are separated by a distance less than R_c. Node locations are known from GPS or any other localisation methods [101, 349]. Sink nodes $S_s(k) \in \{S_s(1), \cdots, S_s(N_s)\}$ are deployed in at locations (x_{s_k}, y_{s_k}). Here, it is assumed that one sink node is deployed in the network, though our model is applicable in networks with a higher number of sink nodes. Irrespective of how the sink node is selected in the network, it is assumed that nodes are aware of the deployed sink node's location.

8.2.2 Coverage Holes and Node Types

CHs are a result of correlated node failures, and can be modelled as a union of circles of radius $R_{h(l)}$ centred at locations (x_{h_l}, y_{h_l}) for $l \in \{1, \cdots, N_E\}$ where is N_E is the total number of damage events. Similar to section 3.2.3, nodes can be classified as either undamaged nodes (U-nodes) and damaged nodes (D-nodes), with the former residing outside and the latter residing inside the coverage hole (Figures 8.1 and 8.2). The U-nodes that detect a damage event within their range are defined as boundary nodes (B-nodes). In order to illustrate the different node types, Figure 8.2 shows a coverage hole with $R_h = 15\ m$ and centre (x_{h_l}, y_{h_l}) at

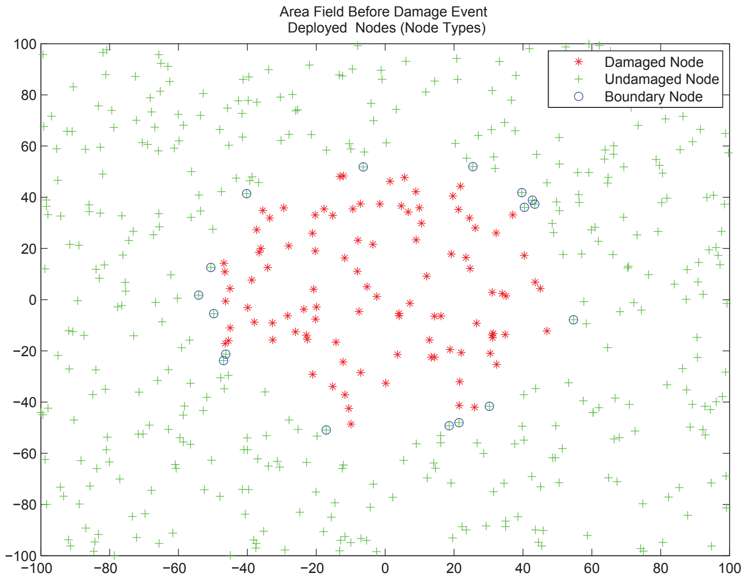


Figure 8.2: Coverage Hole, Node Types $N = 500$ nodes, and $R_c = 15\ m$

(0, 0) in a network of $N = 500$ nodes.

It is assumed that B-nodes detect a damage event if at least one of their neighbours fails (i.e., is in the damaged nodes set) due to the coverage holes. Those U-nodes that are not B-nodes are considered as normal nodes (N-nodes). It is assumed that a sequence of CHs occur with randomly distributed radii and locations throughout the deployed area. The proposed algorithm immediately commences the recovery process following the detection of damage. It is also assumed that B-nodes update their undamaged and damaged neighbours status after the movement and before the next coverage hole event occurs. B-nodes move based on the status of their current neighbours at the time of the damage event. Figure 8.3 schematically depicts a given node with each of the undamaged (blue circle) and damaged (red circle with cross) neighbouring nodes within its range.

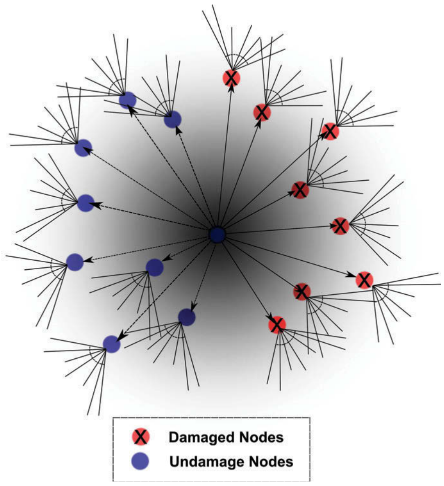


Figure 8.3: Boundary Node and its Undamaged and Damaged Neighbour Nodes

8.2.3 Nodes' Communications Protocol

Before the occurrence of the coverage hole, nodes are aware of their neighbours' degrees and locations in addition to the location of the deployed sink nodes.

It is assumed that, at time of damage event, nodes do not broadcast their information to their neighbours due to security considerations. Therefore, each B-node should autonomously make its decision on the magnitude and direction of its movement. Therefore, after each damage event, B-nodes are not able to distinguish whether their undamaged neighbours are B-nodes or not. Thus, after the formation of a coverage hole, nodes' information about their neighbours is not updated. Figure 8.3 shows a boundary node and its damaged and undamaged neighbours. It should be noted that a given B-node detects occurrence of consecutive damage events as long as it loses at least one additional neighbour after each damage event (CH).

8.2.4 Node Movement Decision

The movement of the autonomous boundary nodes is determined from the following set of parameters:

B-node's neighbours: Those undamaged and damaged neighbours with respective distances of $\vec{x}_{S_b(i)}^{S_{Nu}(l,j)}$ and $\vec{x}_{S_b(i)}^{S_{Nd}(l,j)}$ to node $S_b(i)$ having higher degrees of $d_{S_b(i)}^{S_{Nu}(l,j)}$ and $d_{S_b(i)}^{S_{Nd}(l,j)}$, respectively, are weighted to have a greater influence on the movement vector of $S_b(i)$. The reason is that, if these parameters are properly tuned, each boundary node moves in the direction that will provide maximum coverage to the most distant neighbours of highest degree while maintaining connectivity with its undamaged neighbours, thus maximising its chance of remaining connected to the rest of the network despite its autonomous relocation.

Sink node's status: B-nodes' distances and directions to the deployed sink node S_k, $\vec{x}_{S_b(i)}^{S_s(k)}$ with respect to nodes' ranges are used to modify the relocation of nodes towards the CHs (see Section 8.2.5).

Coverage hole: The locations, number and scales of CHs as well as their distances to sink nodes, may affect the relocations of nodes in the recovery process.

By considering the aforementioned parameters in the movement of B-nodes, the proposed recovery model is introduced as follows:

Centre of mass of B-nodes' neighbours: B-node $S_b(i)$'s centre of mass of neighbouring damaged and undamaged nodes can be weighted such that neighbouring undamaged and damaged nodes with higher degrees should have more importance and therefore a greater effect on the direction and amount of B-nodes' movements. The idea of using the centre of mass of B-nodes' (undamaged and damaged) neighbours is that each B-node moves as much as possible in the direction of the damaged nodes with higher degrees while maintaining its connection to the rest of the network to the greatest extent possible. It is assumed that B-nodes are aware of their neighbours' degrees before the occurrence of each of the consecutive CH events. Node degrees can be used to obtain the centre of mass of B-node' undamaged and damaged neighbouring nodes via Equations 8.1 and 8.2:

$$\vec{M}_{s_b(i)}^{CM_l U} = \frac{\sum_{j=0}^{Nu} \left(d_{S_b(i)}^{S_{Nu}(l,j)} \cdot \vec{x}_{S_b(i)}^{S_{Nu}(j,l)} \right)}{\sum_{j=0}^{Nd} d_{S_b(i)}^{S_{Nu}(l,j)}} \tag{8.1}$$

$$\vec{M}_{s_b(i)}^{CM_l D} = \frac{\sum_{j=0}^{Nd} \left(d_{S_b(i)}^{S_{Nd}(l,j)} \cdot \vec{x}_{S_b(i)}^{S_{Nd}(j,l)} \right)}{\sum_{j=0}^{Nd} d_{S_b(i)}^{S_{Nd}(l,j)}} \tag{8.2}$$

Choosing the proper combinations of undamaged and damaged centres of mass provides an appropriate criteria on the B-nodes' amount and direction of movement towards the damaged areas (CHs). In the case that a B-node loses all of its undamaged and/or damaged neighbouring nodes, its corresponding $\vec{M}_{s_b(i)}^{CM_l U}$ and $\vec{M}_{s_b(i)}^{CM_l D}$ are as follows:

$$\vec{M}_{s_b(i)}^{CM_l U} = \vec{0}, \quad if\ N_u = 0$$
$$\vec{M}_{s_b(i)}^{CM_l D} = \vec{0}, \quad if\ N_d = 0$$

8.2.5 Effect of Sink Node

The amount of movement in the direction of the deployed sink node is computed based on the idea of circle inversion [373,374]. As shown in Figure 8.4, the reverse point P' of point P can be as follows, where the k^2 is defined by Coxeter as the *circle power* [373].

$$\|OP\| \times \|OP'\| = k^2 \tag{8.3}$$

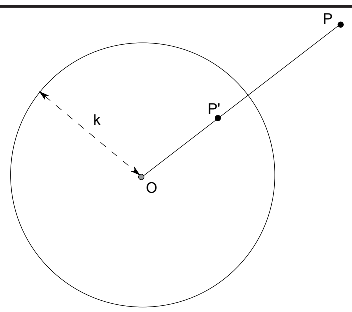


Figure 8.4: Circle Inversion

Using Equation 8.3, if $k = R_{c(S_b(i))}$, $\|OP\|= \|P_{new}(S_b(i)) - P_{cur}(S_b(i))\|$, and $\|OP'\|= \|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\|$ where P_{cur} and P_{new} are the current and new positions of B-node $S_b(i)$ respectively before and after the damage event. Hence, if

$$\|P_{new}(S_b(i)) - P_{cur}(S_b(i))\| \times \|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\| = R^2_{S_b(i)} \tag{8.4}$$

where $\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\|$ is the distance of sink node $S_s(j)$ to the B-node $S_b(i)$ and $R_{S_b(i)}$ is the range of sensor node $S_b(i)$, then the following cases can be considered:

1. $\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\| < R_{S_b(i)}$, as sink node $S_s(k)$ is already within the range of B-node $S_b(i)$. In this case, it is assumed that the movement of B-node $S_b(i)$ in the direction of sink node $S_s(k)$ is not required. So,

$$\|P_{new}(S_b(i)) - P_{cur}(S_b(i))\| = 0, \tag{8.5}$$

   This is the case where the sink node (point P) is within the range of B-node $S_{b(i)}$. In this case, the moving B-node is directed not to move any further toward the sink node in order to reduce the probability of collisions amongst the nodes and to avoid increased probability of interference.

2. $\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\| > R_{S_b(i)}$, the amount of movement node $S_b(i)$ toward the sink node

is give by,

$$\|P_{new}(S_b(i)) - P_{cur}(S_b(i))\| = \frac{R^2_{S_b(i)}}{\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\|} \tag{8.6}$$

3. if $\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\| = R_{S_b(i)}$, then, the amount of movement from the Equation 8.6 is modified by random factor of $\rho_{col} \sim U[0,1]$ with the uniform distribution.

$$\|P_{new}(S_b(i)) - P_{cur}(S_b(i))\| = \frac{R^2_{S_b(i)}}{\|\overrightarrow{x}^{S_s(k)}_{S_b(i)}\|}.\ \rho_{col} \tag{8.7}$$

The random factor ρ_{col} reduces the probability of physical collision of nodes moving towards the deployed sink node.

Therefore, using the aforementioned inverse circle geometrical procedure leads B-nodes to modify the the amount and direction of their movements towards the deployed sensor nodes based on their ranges and their distances to the sink nodes. Based on the properties of circle inversion, B-nodes that are farther from the sink nodes move less, while B-nodes closer to the sink node move more in order to reflect the nodes' importance with respect to their distances to the deployed sink nodes.

8.2.6 Movement Toward CHs

Suppose that the total number of a given B-node's neighbours before the formation of CH l is $N^{U_l}_{S_b(i)}$. Let the number of undamaged and damaged neighbouring nodes of B-node $S_b(i)$ following damage even(CH) l be $N^{U_l}_{S_b(i)}$ and $N^{D_l}_{S_b(i)}$ respectively. The *damage ratio* of boundary node $S_b(i)$ due to CH l is then defined as:

$$\mathcal{E}^{d_l}_{S_b(i)} = \frac{N^{D_l}_{S_b(i)}}{N^{U_l}_{S_b(i)} + N^{D_l}_{S_b(i)}} \tag{8.8}$$

Using the sigmoid function $\sigma(t) = \frac{1}{1+e^{-t}}$ (for the motivations behind using sigmoid function, refer to work by Bishop [375] and Mount [376]), the damage ratio of

B-node $S_b(i)$ can be used to define the following weights:

$$
\begin{aligned}
\omega_{N^{D_l}_{S_b(i)}} &= 1 - \sigma(\{\varepsilon^{d_l}_{S_b(i)}\}^{-1}) \\
\omega_{N^{U_l}_{S_b(i)}} &= 1 - \omega_{N^{D_l}_{S_b(i)}}
\end{aligned}
\tag{8.9}
$$

These weights ($\omega_{N^{U_l}_{S_b(i)}}$ and $\omega_{N^{D_l}_{S_b(i)}}$) are used to combine the two vectors obtained in the direction of centres of mass of $S_b(i)$'s damaged and undamaged neighbours respectively, resulting from the formation of coverage hole l.

8.2.7 Proposed Movement Model

Using the parameters introduced in the previous section, a number of movement models may now be defined.

Simple Sink Movement (SSM) in which each moving B-node $S_b(i)$ solely moves towards the deployed sink node regardless of its status with respect to CHs. Thus, depending on the location of the damage event and the sink node, the recovery of CH may not be primary goal of the movement algorithm. By using the method of circle inversion discussed in Section 8.2.5, B-nodes move towards the sink node. Each B-node's movement amplitude depends on node's own range and its distance to the given deployed sink node. So, for each B-nodes, the movement $\vec{M}{}^{S_k}_{S_b(i)}$ can be defined as

$$
\vec{M}{}^{S_k}_{S_b(i)} = P_{new}(S_b(i)) - P_{cur}(S_b(i)) \tag{8.10}
$$

$$
\vec{M}{}^{Total_l}_{S_b(i)} = \vec{M}{}^{S_k}_{S_b(i)}
$$

In the case of more than one deployed sink node, B-nodes movement can be modified by considering the weighted effect of those sink nodes as,

$$
\vec{M}{}^{Total_l}_{S_b(i)} = \sum_{k=1}^{N_s} \zeta_k . \vec{M}{}^{S_k}_{S_b(i)} \tag{8.11}
$$

where ζ_k is weight of importance of sink node k based on its distance to B-node $S_b(i)$.

The relocation of $S_b(i)$ for coverage hole l due to the overall effect of centre of mass of $S_b(i)$'s neighbours of different types, $\overset{\rightarrow CM_l}{M}_{S_b(i)}$, can be defined as a function of f(Equation 8.12). The function f considers the effect of the centres of mass of B-nodes' undamaged and damaged neighbours (obtained in Section 8.2.4 and Equations 8.1 and 8.2) for CH of l.

$$\overset{\rightarrow CM_l}{M}_{S_b(i)} = f(\overset{\rightarrow CM_l U}{M}_{S_b(i)}, \overset{\rightarrow CM_l D}{M}_{S_b(i)}) \tag{8.12}$$

where $\overset{\rightarrow CM_l U}{M}_{S_b(i)}$ and $\overset{\rightarrow CM_l D}{M}_{S_b(i)}$ are centres of masses of undamaged neighbour and damaged neighbours computed. Using weights in Equation 8.9, f is defined as a linear combination of centres of mass $\overset{\rightarrow CM_l U}{M}_{S_b(i)}$ and $\overset{\rightarrow CM_l D}{M}_{S_b(i)}$ as

$$\overset{\rightarrow CM_l}{M}_{S_b(i)} = \omega_{N^{D_l}_{S_b(i)}} . \overset{\rightarrow CM_l U}{M}_{S_b(i)} + \omega_{N^{D_l}_{S_b(i)}} . \overset{\rightarrow CM_l D}{M}_{S_b(i)} . \tag{8.13}$$

Therefore, function f can be used to tune the effect of B-nodes' centres of mass on their relocations towards CHs similar to Equations 8.9 and 8.13.

If each B-node solely moves towards the CHs regardless of the status of the deployed sink nodes, the amount and directions of nodes' movements can be governed by the centres of mass of B-nodes' undamaged and damaged neighbours according to Equation 8.13. So in this case, primary objective of moving B-nodes is to repair of CHs rather to modify the B-nodes' distances to the sink nodes in the network. In this case, the relocation of each B-node is obtained as,

$$\overset{\rightarrow Total_l}{M}_{S_b(i)} = \overset{\rightarrow CM_l}{M}_{S_b(i)} \tag{8.14}$$

Combined SS and CM Movement (CSSCMM): In this algorithm, the movement amplitude and direction of each B-node depends on both the neighbouring nodes' *centres of mass weight* (calculated as $\lambda_{cm} = 1 - \lambda_s$) and the sink node weight $\lambda_s \in [0,1]$, defined as the *sink move weight*. Using these weights, the total movements are linear combinations of Equations 8.10 and 8.12:

$$\overset{\rightarrow Cmb_l}{M}_{S_b(i)} = \lambda_s \cdot \overset{\rightarrow CM_l}{M}_{S_b(i)} + \lambda_{cm} \cdot \overset{\rightarrow S_k}{M}_{S_b(i)} \tag{8.15}$$

where $\overset{\rightarrow Cmb_l}{M}_{S_b(i)}$ is B-node $S_b(i)$'s combined movement vector due to CH l.

$$\overset{\rightarrow Total_l}{M}_{S_b(i)} = \overset{\rightarrow Cmb_l}{M}_{S_b(i)}$$

Modified CSSCMM (M-CSSCMM): In this algorithm, the amount of movement of participating nodes is tuned with regard to the status of B-nodes around the CH and the sink node, as defined in Equations 8.16 to 8.19,

$$\alpha_{\langle cm,s\rangle} = \begin{cases} \arccos\left[\dfrac{\overset{\rightarrow CM_l}{M}_{S_b(i)} \cdot \overset{\rightarrow S_k}{M}_{S_b(i)}}{\|\overset{\rightarrow CM_l}{M}_{S_b(i)}\| \|\overset{\rightarrow S_k}{M}_{S_b(i)}\|}\right], & \text{if } \|\overset{\rightarrow CM_l}{M}_{S_b(i)}\|, \|\overset{\rightarrow s_k}{M}_{S_b(i)}\| \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{8.16}$$

where $\alpha_{\langle cm,s\rangle}$ is angle between the moving vectors obtained from Equations 8.10 and 8.12.

$$\beta^l_{\langle cm,s\rangle} = \alpha_{\langle cm,s\rangle} \cdot \left(\frac{\theta_\tau}{\pi/2}\right)^{-1} \tag{8.17}$$

where θ_τ is defined as *threshold angle* and $\beta^l_{\langle cm,s\rangle}$ is define as *modification angle*.

$$\gamma^l_{\langle cm,s\rangle} = cos^2(\beta^l_{\langle cm,s\rangle}) \tag{8.18}$$

and the attenuated movement is obtained using the Equation 8.15,

$$\overset{\rightarrow Total_l}{M}_{s_b(i)} = \begin{cases} \gamma^l_{\langle cm,s\rangle} \cdot \overset{\rightarrow CM_l}{M}_{s_b(i)}, & \text{if } (\beta^l_{\langle cm,s\rangle} < 1), \\ 0, & \text{otherwise} \end{cases} \tag{8.19}$$

The idea of M-CSSCMM is that B-nodes movement be calculated flexibly according to the angle between the two movement vectors that each B-node forms with its neighbours centres of mass and the deployed sink node. In Equation 8.19, If $\gamma^l_{\langle cm,s\rangle} = 1$, movement of B-nodes is tuned solely based on the angle of $\alpha_{\langle cm,s\rangle}$. This movement algorithm is denoted as dM-CSSCMM. If the condition of $\beta^l_{\langle cm,s\rangle} < 1$ is relaxed, the movement algorithm is denoted as aM-CSSCMM in which only amplitudes of movements are attenuated.

It should be noted that appropriate selection algorithms of boundary nodes similar to the ones presented in Chapter 3 and Appendix A can be used, if required, to reduce the probability of physical collisions, interferences and movements among autonomous boundary nodes.

8.2.8 Condition at Border of Deployment area

As the result of relocation algorithms, if nodes move beyond the borders of the deployment area, their locations are limited to the borders of x_{min}, x_{max}, y_{min}, and y_{max}.

8.3 Performance Evaluation

In this section, suitable performance metrics and benchmarks are introduced, and simulation results are presented.

8.3.1 Performance Metrics

The following performance metrics are used for evaluating the proposed models.

8.3.1.1 Percentage of Coverage

The 2D rectangular deployment area ($[x_{min}, x_{max}] \times [y_{min}, y_{max}]$) is divided into grid cells. Grid cells are covered by sensor nodes if their coordinates z_i=(x_i, y_i) resided within the nodes' ranges. *Percentage of 1-coverage* is the number of grid cells that are covered by at least one sensor nodes over the total number of grid cells in the given deployed area.

8.3.1.2 B-Nodes to Sink Nodes Distances

In the recovery of consecutive random CHs, it is desirable that the moving B-nodes maintain or reduce their distances to the deployed sink nodes in the network. Thus, as the result of the proposed recovery model, distances of the set of moving B-nodes to the deployed sink node(s) are expected to monotonically decrease as the given nodes relocate towards the CHs. The movement algorithms' performance are evaluated in terms of the cumulative mean of maximum, minimum and average distances of B-nodes to the deployed sink nodes in the course of recovery of consecutive random CHs in the network.

8.3.1.3 Sparsity of Cut

As in Trevisan' study [377], it is mentioned that property of *clique* can be used as the standard of network reliability, as the clique is considered to be the most reliable network layout if a fraction of vertices are disconnected from the graph. Therefore, the expansion and sparsity cut parameters that Trevisan defined [377] can be used to measure the reliability of the network with respect to clique (as a criteria).

Definition 11 *[377], Let $G = (V, E)$ be a graph and let $(S, V - S)$ be a partition* *of the vertices (a cut). Then the sparsity of the cut is*

$$\phi(S) := \frac{E(S, V-S)}{|E|} . \left(\frac{|S|.|V-S|}{|V|^2/2} \right)^{-1} \tag{8.20}$$

where $E(S, V-S)$ is the number of edges in E that have one endpoint in S and one end point in $V-S$.

Sparsity can be used to measure of how the set of B-nodes are connected to the rest of U-nodes that is defined as the normal nodes (N-nodes) (see Section 8.2.2). A higher sparsity cut requires a larger number of link disconnections to cause a partitioning of the network. The performance of movement algorithms are compared by the cumulative mean of sparsity throughout the consecutive damage events and recovery.

8.3.2 Benchmark Movement Algorithms

The proposed movement algorithms are compared with the distributed self spreading algorithm (DSSA) [227] (a force-based movement algorithm) and Vor-voronoi and MinMax-voronoi (two Voronoi-based movement algorithms) [231]. The performance of limited version of Vor-Voronoi and MinMax-Voronoi movement algorithms (Vor-voronoi (L) and MinMax-voronoi (L), respectively) are also compared with the proposed movement algorithms. Nodes' movements in Vor-voronoi(L) and MinMax-voronoi (L) are limited when they move beyond their ranges.

8.3.3 Results

Using Matlab, N=1000 nodes with communication and sensing ranges of 15 m ($R_c = R_s = 15\ m$) were deployed with a uniformly distributed 2D random distribution in a rectangular deployment area of $[-100, 100] \times [-100, 100]$. Each coverage hole is modelled as a circle with random radius $r_{Hole}(U_r \sim [0, 50])$ and random centre of (x_{Hole}, y_{Hole}) with the uniform distribution ($U_x \sim [-100, 100]$, $U_y \sim [-100, 100]$). To properly compare the proposed movement algorithms with the benchmarks, we used a randomly generated radii of $r_{Hole}^{\{1st,\cdots,5th\}} = [47.3, 48.9, 35.2$ $,41.8, 42.5]$ m and randomly generated locations of $x_{Hole}^{\{1st,\cdots,5th\}} = [23.5, -59, 41.7$ $,12.7, -67.3]$ and $y_{Hole}^{\{1st,\cdots,5th\}} = [47, 31, 76.2, -18.5, -42]$ to model the 5 consecutive CHs with respect to their order of occurrences (resulted in large scale CHs). The performance of the movement algorithms in terms of sparsity, percentage of coverage and boundary-sink inter-nodal distances were examined for different parameters of $\lambda_s = \{0.9, 0.5, 0.1\}$ ($\lambda_{cm} = 1 - \lambda_s$), sink node at the locations of $(\pm 100, \pm 100)$. Due to similar performance patterns for the given parameters and for sake of brevity, only results for $\lambda_s = 0.9$ with the sink node at the location of $(100, 100)$ are presented in this chapter. The experiment was repeated $\#Exp = 50$ times for the movement algorithms.

It should be noted that in this chapter that the movement algorithms' performances were considered for the disk-shaped coverage holes of random radii and centres (with uniform distribution) which were generated sequentially and their recoveries began as soon as they were detected. Therefore, efficiency of movement algorithms for other damage patterns and CHs may be different, which could be investigated in further studies.

Performance of movement algorithms were compared in terms of network's sparsity (Table 8.1 and Figure 8.5), percentage of coverage (Table 8.2 and Figure 8.6), and B-nodes to sink node distances (Table 8.3 and Figures 8.7, 8.8, and 8.9). To compare different movement algorithms from Tables 8.1, 8.2, and 8.3, performance was calculated based on the efficacy of each algorithm, which is applied immediately after the occurrence of each damage event (i.e., 2nd row from each movement algorithms in the tables are used to compare their performance).

Algs.	Sparsity				
	1st	2nd	3rd	4th	5th
SSM	1.7125	1.4266	1.2806	1.1905	1.1182
	1.7120	1.4410	1.2974	1.2049	1.1336
CSSCMM	1.8860	1.4527	1.2880	1.1550	1.0466
	1.7082	1.3169	1.1675	1.0471	0.9495
M-CSSCMM	1.8969	1.4754	1.3304	1.1994	1.0973
	1.7995	1.4137	1.2814	1.1572	1.0611
aM-CSSCMM	1.8897	1.4673	1.3223	1.1915	1.0893
	1.7831	1.3969	1.2653	1.1417	1.0460
dM-CSSMM	1.8201	1.4368	1.3084	1.1829	1.0824
	1.6980	1.3584	1.2463	1.1290	1.0364
DSSA	1.8035	1.4604	1.3845	1.2655	1.1686
	1.8324	1.4799	1.4012	1.2803	1.1816
Vor-Voronoi	1.7352	1.4247	1.3859	1.2862	1.1972
	2.0631	1.6582	1.5712	1.4493	1.3359
Vor-Voronoi(L)	1.8448	1.4733	1.4025	1.2761	1.1767
	2.1112	1.6597	1.5399	1.3968	1.2816
MinMax-Voronoi	1.7053	1.4147	1.3902	1.2842	1.1880
	1.9868	1.5892	1.5042	1.3844	1.2706
MinMax-Voronoi(L)	1.7275	1.4455	1.4056	1.2982	1.1994
	1.9957	1.6054	1.5191	1.4026	1.2865

Table 8.1. Sparsity of Network with Sink Nodes located at $(100, 100)$ and 5 Consecutive CHs

From Table 8.1 and Figure 8.5, it can be seen that the Voronoi-based movement algorithms and DSSA outperform our proposed algorithms by 20% and 10%, respectively. Among the proposed movement algorithms, only SSM improved network sparsity cut. Vor-Voronoi movement algorithms had the best performance. DSS outperformed SSM by about 5%.

Results from Table 8.2 and Figure 8.6 show that the proposed movement algorithms either slightly outperformed or matched DSSA. Voronoi-based algorithms outperformed the proposed movement algorithms and DSSA with average of 20%−24% during the recovery of consecutive CHs.

Algs.	Percentage of Coverage(%)					
	Initial	1st	2nd	3rd	4th	5th
SSM	100	90.62 90.65	78.45 78.46	76.27 76.19	66.94 67.08	57.85 58.15
CSSCMM	100	90.65 90.82	78.59 78.81	76.08 76.32	67.46 67.62	58.35 58.42
M-CSSCMM	100	90.63 90.79	78.57 78.68	75.88 75.97	66.92 66.99	57.64 57.68
aM-CSSCMM	100	90.63 90.77	78.55 78.64	75.86 75.94	66.89 66.96	57.55 57.52
dM-CSSMM	100	90.56 90.74	78.63 78.77	76.02 76.13	66.93 67.03	57.93 57.99
DSSA	100	90.55 90.68	78.47 78.70	75.88 76.17	67.31 67.48	58.52 58.70
Vor-Voronoi	100	90.80 99.65	87.92 97.31	91.54 98.24	90.19 97.00	87.60 96.00
Vor-Voronoi(L)	100	90.62 97.15	86.00 94.64	89.09 95.94	87.62 95.33	86.01 93.10
MinMax-Voronoi	100	90.63 98.76	87.36 96.96	91.28 97.66	90.11 97.83	89.27 96.38
MinMax-Voronoi(L)	100	90.63 98.76	87.33 96.89	91.28 97.75	90.22 97.53	88.83 96.16

Table 8.2. Percentage of Coverage of Network, with Sink Nodes located at $(100, 100)$ and 5 Consecutive CHs

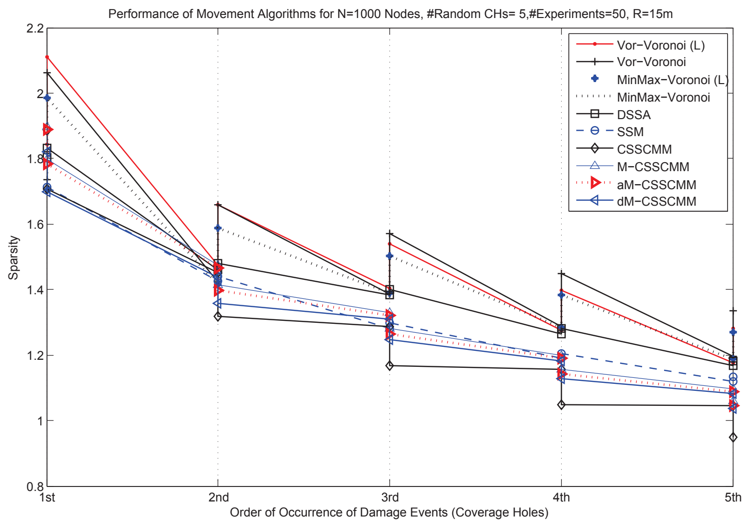


Figure 8.5: Sparsity of Network

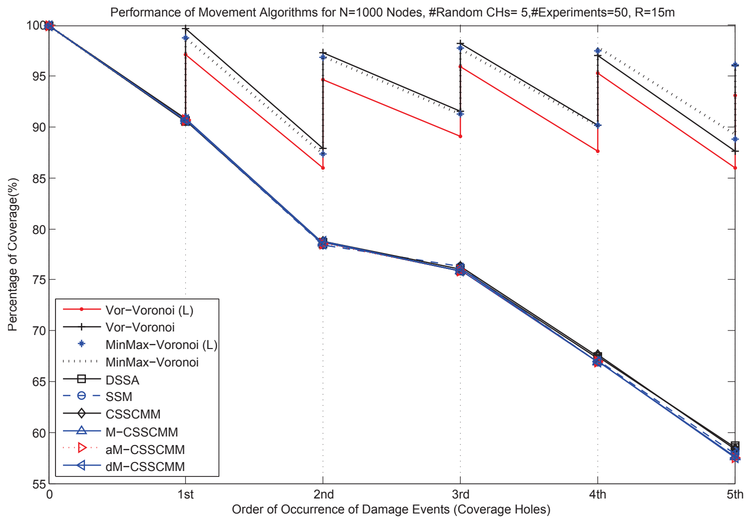


Figure 8.6: Percentage of Coverage

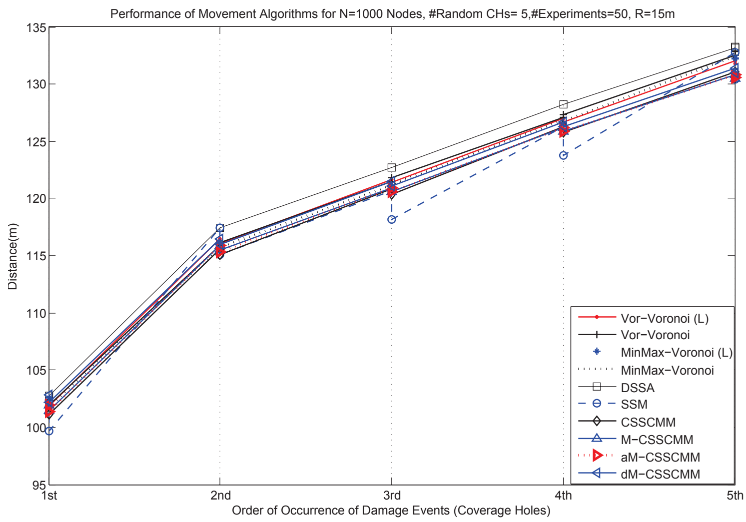


Figure 8.7: Avg Distances CHs' Boundary Nodes to Sink Node

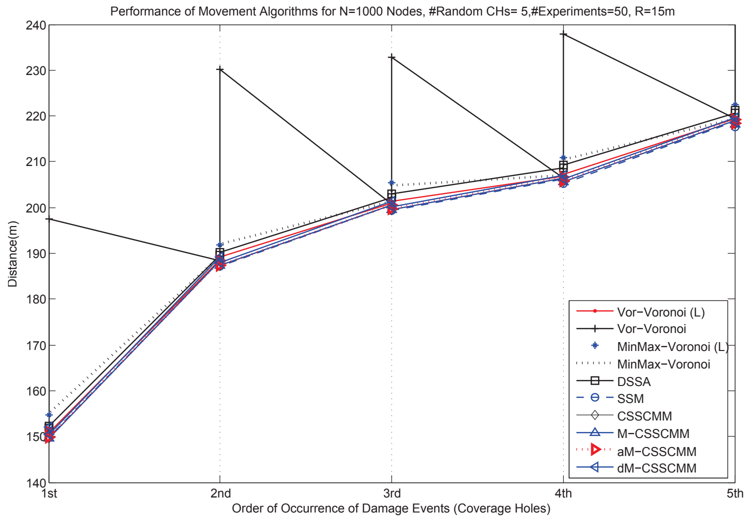


Figure 8.8: Max Distances CHs' Boundary Nodes to Sink Node

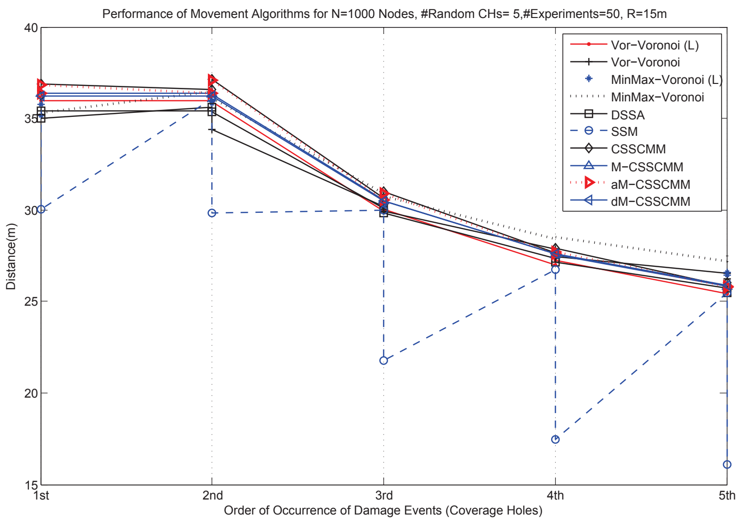


Figure 8.9: Min Distances CHs' Boundary Nodes to Sink Node

Algs.	Average Distance					Maximum Distances					Minimum Distance				
	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th	1st	2nd	3rd	4th	5th
SSM	102.35	117.47	120.69	126.25	132.73	151.30	188.47	200.53	206.31	218.73	36.25	36.09	29.96	26.74	25.48
	99.69	115.07	118.19	123.79	130.39	149.81	187.23	199.37	205.19	217.66	30.02	29.83	21.75	17.44	16.07
CSSCMM	101.83	115.60	120.87	126.28	131.03	151.14	188.40	200.55	206.47	219.10	36.30	36.58	30.63	27.65	25.84
	101.15	115.07	120.38	125.80	130.59	149.69	187.26	199.59	205.64	218.26	36.86	37.12	30.98	27.89	26.02
M-CSSCMM	101.98	115.86	120.95	126.21	130.84	151.12	188.40	200.60	206.56	219.19	36.39	36.37	30.52	27.59	25.84
	101.46	115.47	120.61	125.88	130.53	149.75	187.37	199.73	205.81	218.41	36.35	36.33	30.49	27.57	25.82
aM-CSSCMM	101.94	115.81	121.01	126.30	130.96	151.13	188.41	200.61	206.57	219.16	36.39	36.62	30.69	27.64	25.79
	101.35	115.36	120.59	125.89	130.58	149.75	187.38	199.74	205.81	218.39	36.85	37.06	30.87	27.71	25.80
dM-CSSMM	102.83	116.46	121.50	126.69	131.36	151.98	189.17	201.26	207.14	219.56	36.21	36.21	30.49	27.62	25.90
	102.21	115.99	121.08	126.29	130.99	150.49	187.97	200.23	206.21	218.61	36.21	36.21	30.49	27.62	25.89
DSSA	102.73	117.42	122.67	128.19	133.21	151.80	189.53	202.27	208.65	220.71	35.41	35.60	30.10	27.35	25.70
	102.75	117.45	122.72	128.21	133.23	152.35	190.22	202.93	209.23	221.29	35.02	29.84	17.52	27.12	25.48
Vor-Voronoi	102.31	115.87	121.51	127.06	132.54	150.49	188.43	200.74	206.51	219.44	35.44	35.44	30.19	27.88	26.56
	101.94	116.14	121.83	127.32	132.89	197.52	230.11	232.78	237.96	244.44	35.44	34.41	29.92	27.46	26.25
Vor-Voronoi(L)	102.59	116.44	121.68	126.98	132.03	150.95	188.50	200.88	206.93	219.46	35.96	35.97	29.94	27.01	25.42
	101.85	116.06	121.40	126.68	131.79	150.93	189.13	201.35	207.16	219.47	35.98	35.94	30.03	27.24	25.64
MinMax-Voronoi	102.32	115.81	121.19	126.81	132.49	151.85	189.34	201.35	206.99	219.40	36.21	36.49	30.90	28.40	27.19
	101.76	115.76	121.25	126.79	132.45	155.05	191.96	204.87	210.57	222.32	35.32	36.11	30.74	28.50	27.53
MinMax-Voronoi(L)	102.55	189.02	120.98	126.51	132.20	151.59	226.44	201.23	206.91	219.28	35.74	35.96	30.33	27.70	26.44
	101.88	115.95	121.12	126.59	132.31	154.61	191.83	205.35	210.90	222.44	35.13	35.80	30.15	27.78	26.61

Table 8.3. Distances to Sink Nodes located at $(100, 100)$ and 5 Consecutive CHs

As shown in Table 8.3 and Figure 8.7, the proposed movement algorithms slightly outperform and/or matches the Voronoi-based algorithms in terms of average boundary to sink node distance during the course of the recovery process. DSSA has the worst performance among the other algorithms. SSM marginally has the best performance.

From Table 8.3 and Figure 8.8, regarding the maximum distance of boundary to sink node, the proposed algorithms marginally outperform the DSSA and Voronoi-based algorithms. SSM and Vor-Voronoi have the best and worst performances, respectively as the SSM outperforms the Voronoi-based algorithms by an average of 17% in the course of recovery from consecutive CHs. The performance improvement in SSM progressively decreases from 25% to 10% as the number of coverage holes increase from 1 to 5.

From Table 8.3 and Figure 8.9, performance of SSM's minimum distances boundary to sink node is noticeable. SSM outperforms other movement algorithms which varies from 15% to 40% as the number of CHs increase from 1 to 5. SSM outperforms Voronoi-based algorithms by an average of 25%. The performance differences increase with the occurrence of more coverage holes. The other proposed movement algorithms marginally outperform DSSA and Voronoi-based algorithms between 1% and 5%.

8.4 Conclusion

This chapter presented a new approach to the recovery of CHs via node relocation algorithms in WSNs. In the proposed cooperative recovery model, the status of deployed sink nodes are considered by autonomous moving nodes in the recovery process. The proposed distributed node relocation algorithms aimed to address additional design goals, such as reducing the moving nodes' distances to the deployed sink nodes. The approach presented in this chapter can be considered as an initial exploratory effort to partially address the issues that arise due to the gradual and undesirable impact of sink nodes' disconnections with their surrounding nodes.

The performance of the proposed movement algorithms used in the CH recovery model were compared with a range of suitable benchmarks previously introduced in this thesis (DSSA as a representative of force-based algorithms together with two Voronoi-based movement algorithms), including percentage of coverage, boundary to sink node distances and sparsity cut during the course of recovery from the consecutive random and large scale CHs in the network. Results show that there are trade-offs between targeted design goals for each relocation algorithm. Depending on the performance metric, the proposal model outperform, underperform or matches its benchmarks. The proposed model presents new approach by considering the importance of sink nodes into for prospective node relocation algorithms. Therefore, further performance improvements and novel node relocation algorithms still hold a great potential.

CHAPTER 9

Concluding Remarks

9.1 Conclusions

We reviewed concepts of resiliency in general and in networks, topology control schemes, networks' events, and faults and fault management techniques in WSNs in addition to stages of network recovery in this thesis. By presenting a thorough classifications of networks' holes, we focused on the coverage holes (CHs) as they are most prevalent holes in WSNs and can severely disrupt networks' operation and integrity if they are unattended. Large scale coverage holes (LSCHs) have more deteriorating effects on WSNs. Therefore, the resiliency of networks was investigated with respect to the CHs that resulted from correlated node failures in WSNs. Different stages of network recovery from CHs were considered based on idea of an emergent cooperation among nodes, through which a global pattern would be reached in a network that swiftly reacts to the dynamic topology changes solely based on local interactions and knowledge of autonomous nodes. Devising such models with the proper level of complexity to address the requirements of given applications is a non-trivial task that requires further study.

In order to surmount the emerging undesirable changes in topology and dynamic behaviours of networks, different topology control (TC) schemes (e.g. node relocation) were introduced [66,110,181,182,250,251]. Different (distributed) node relocation algorithms were proposed. Distributed node relocation algorithms are suitable for the recovery of CHs in many real-time and security-sensitive applications where extensive message exchanges among the nodes are not practical. By preventing unnecessary broadcasts and using available 1-hop information, the nodes' energy can be preserved and the overall lifetime of the network extended as the burden of decision making is spread among the (autonomous) nodes rather than delegated back to a centralised supervisor. We considered the benefit and performance of hybrid strategies in TC of network because hybrid approaches can combine the strengths and avoid the weaknesses of individual approaches. However, such flexibility comes at the price of new challenges that need to be addressed if their the benefits are to be realised. For example, the problem of power control has been mainly considered in tandem with stationary rather than mobile nodes [66]. If it is applied in conjunction with other topology control schemes, new opportunities and challenges arise.

Coverage holes modelled mainly in the shapes of circles with given radii and locations, are used to represent omni-directional correlated node failures, such as explosions. Other shapes of coverage hole can be formed by combinations of simpler convex shapes [273].

This work showed that topology control schemes (e.g. node relocation) performance would change if the scale of the CHs changes. Therefore, although a topology scheme is known to be efficient in addressing changes in networks' topology, it may not be considered as a general solution for other CH shapes and/or scales.

In the proposed CHs detection model, the presence of CHs is detected as soon as nodes, known as boundary nodes (B-nodes) perceive the damage event within their ranges. Based on available 1-hop knowledge inferred from the B-nodes' statistical and geometrical features, (distributed) B-node selection algorithms (BNS-algorithm) are proposed, in which B-nodes autonomously self-select to participate in the potential recovery processes. With nodes' local information and their autonomous decisions, BNS-algorithms can reach a global pattern in WSNs. Benefiting from the redundancy of deployed nodes, BNS-algorithms reduce the number of B-nodes that are selected for required levels of service in the margin of B-nodes, such as coverage. The significance of BNS-algorithms is in the time-sensitive applications which may have the security concerns in hostile and deserted environments without the need for the centralised supervision.

A constrained node movement model in which autonomous B-nodes partially/wholly recover LSCHs was presented. B-nodes make autonomous decisions based on their available 1-hop information of immediate neighbouring nodes and use the concept of α-chord to either move towards or circulate around the damaged areas (CHs). Autonomous B-nodes movements, connectivities and the network's global behaviour as the result of B-nodes' local interactions can be easily controlled by the B-nodes' α-chords. This model is an example that, with B-nodes' local interactions, an emergent cooperation can be achieved so that B-nodes swiftly are able to partially recover the CHs when damage events occur within their ranges. The significance of such node movement model is for the time-sensitive applications where centralised control and recovery are neither feasible nor encouraged, possibly due to the security concerns.

In this work, the proposed fuzzy node relocation models based on force-based fuzzy node movement algorithms are suitable to consider the uncertainty governed by distributed and local interactions of moving nodes and the indefinite choices of movements. Properly considering the uncertainty among interacting nodes would reduce the likelihood of collision, isolation and oscillations of nodes which improve the network's connectivity. In the proposed fuzzy node relocation models, the problem of choosing appropriate fuzzy parameters to achieve higher performances was addressed by tuning these parameters via either expert knowledge or particle swarm optimisation (PSO) (in the initial and each movement iteration). Each mobile node is able to locally tune its parameters solely based on its neighbours' distances within its ranges.

Inspired by nature, a model of cooperative recovery of CHs in which nodes move towards damaged areas in the form of disjoint spanned trees (DS-Trees) was proposed. Based on nodes' local and autonomous interactions with their neighbouring nodes and their distances to CHs, a set of disjoint trees around the CH spans as they are notified about the CHs. As a result of independent movements of DS-Trees towards CH, a cooperative behaviour in the recovery of LSCHs emerges. Nodes' movements change proportionally with respect to their DS-Trees' depths and their movements are controlled to reduce the chance of collisions among nodes. Nodes are notified about their surrounding CHs with given levels as they are spanned in their disjoint trees. The effects of different number of participating nodes and the depths of nodes notified around CHs in the recovery process (i.e. node relocation) of damaged area were investigated for Voronoi-based and force-based node relocation algorithms. From the results, increasing the number of participating nodes and/or the depth of notified nodes beyond certain degrees, did not seem to improve the performance of node relocation algorithms used for the CHs recovery significantly, especially with respect to percentage of coverage. Therefore, if proper fractions of nodes are notified and participate in recovery process, a significant amount of autonomous nodes' energies can be preserved, which would increase the mobile WSNs' lifetime.

In this work, by devising hybrid topology control scheme, a CH recovery algorithm was proposed for Mobile WSNs (MWSNs). The proposed algorithm combined sensing power control and physical node relocation using a game theoretic approach. With respect to simulation results over number of different node densities and CH events, the proposed approach outperformed other CH recovery algorithms in terms of percentage of coverage and energy consumption. The significance of proposed algorithm is that it reduces the required energy for recovering the CHs which increase network's lifetime and resiliency. By applying a new game theoretic approach to repair CHs in a distributed manner, nodes decide only based on local information of their neighbouring nodes and autonomously take CH recovery actions such that a global behaviour emerges, which can maintain network coverage in the presence of random damage events.

A new approach of CH recovery via node relocation was presented. We suggested that moving autonomous nodes consider the status of sink nodes in the recovery process. Significance of this work as an initial exploratory effort, is that the proposed node relocation algorithm aimed to reduce the distances of moving nodes to the deployed sink nodes while repairing the CHs in order to account for the gradual and undesirable impact of sink nodes' disconnections with surrounding nodes. The results of comparing the proposed algorithm with its suitable benchmarks showed that there are trade-offs between targeted design goals for each relocation algorithm. The proposed model present a new approach by considering the importance of sink nodes for prospective node relocation algorithms. Therefore, there is still a great potential for further performance improvements and novel node relocation algorithms.

9.2 Future Research Suggestions

Several interesting opportunities and research directions have been identified in the course of this thesis. Some of them are listed as follows:

- **Node Selection Algorithms**. Different BNS-algorithms based on simple but interesting geometrical and statistical properties of B-nodes and their neighbours could be proposed. These algorithms could be devised such that a sensible global behaviour and emergent cooperation could be achieved with no or the least possible message exchanges among nodes in network. Such lightweight BNS-algorithms are effective in many time-sensitive and/or security-oriented applications.
- **Relocation Algorithms vs. CHs**. Recovery capability and efficiency of conventional and/or newly devised relocation algorithms for different types of CHs with respect to their scales, numbers and distributions should be investigated.
- **CHs' Direct/Indirect Effects**. Depending on their locations, CHs of different scales/shapes not only directly affect nodes in the immediate vicinities, but also result in indirect changes in more distant parts of network. Such indirect effects are significant and should not be neglected in the design and recovery mechanism of networks. Therefore, by having a clear understanding of the indirect/direct effects of CHs, it could possible to design networks that are more resilient and robust to undesirable events, such as minimum phenomena, funnel effects, and to devise more realistic recovery strategies and topology control schemes of higher performances in response to dynamic topological changes.
- **Nano Scale and Molecular Networks**. Due to their different environ-

ments and communication nano robots, molecular communications and micro bio-robots [378–383], many of approaches to cope with the topological changes and fault management mechanisms should be revised.

- **Realistic Fault Model**. In order to more effectively examine the impact of different faults on the lifetime and performance of the networks and hence devise more targeted recovery mechanisms and fault management strategies, it is suggested that these faults be modelled more realistically with respect to their behaviours (i.e. aggressive, transient, temporary node failure, cascaded failures, etc.), the geometry, the distribution, the frequency of occurrences (i.e. simultaneous or sequential, single or multiple, random or collocate, etc.) and the features of network environments (i.e. air, ocean, terrain, underground, etc.) and alignments of nodes. This is important for network design and fault-tolerance mechanisms to either recover the failure or isolate [258] it from rest of the network by accepting some performance degradation.
- **Other Categories of Network Holes**. Although CHs are considered as the most prevalent holes in WSNs [24, 86, 91], the recovery models should be designed such that networks can be resilient to multiple categories of holes (e.g. wormholes attacks) and can simultaneously meet multiple objectives, such coverages, energy and security.
- **Hybrid/Heterogeneous Networks**. New trends of using a hybrid network has brought new possibilities to increase the networks' robustness and resiliency, and increase the speed of adaptability to networks' dynamic behaviours in volatile environments. Different types of mobile robots are used in disaster recovery and critical mission management [384]. In addition to dropping sensors from the air [385, 386], for instance, unmanned aerial vehicles (UAVs) [87, 387] can play an important role in mobile sensor networks. UAVs as the mobile nodes, can flexibly cope with harsh and isolated

environments, and autonomously act as a relay to compensate for the disconnection of ground based-terminals and nodes and repair the failures in WSNs [388–392]. Similarly, autonomous underwater vehicles (AUVs) can be used to mend underwater sensor node failures [393, 394]. Combining the capabilities of ground, aerial, and/or underwater vehicles to collaboratively operate and move in different environments could warrant attention. As an example, the combination of unmanned ground vehicle (UGV) and UAV as UGV-UAV would cooperatively increases the mobility, navigation, coordination [395] and monitoring [396], which could be used to fill the gaps and failed nodes in the harsh environments with rough terrain and obstacles (e.g. rivers). In hostile and harsh environments, wheeled or hopping mobile nodes [249,384,397,398] may not be able to form and repair the network without the aid of airborne nodes. Similarly, a combination of UAVs and AUVs could cooperate to have better performance, such as coordination for ocean exploration [399]. However, such a hybrid UAV-AVU deployment have its own challenges and issues, which require careful examinations [400]. Hybrid robots, which can act partly as UAV and/or UGV based on the environment, have also been garnering interest from researchers recently.

- **Centralised and Distributed Recovery Schemes**. Although many distributed and centralised coverage hole recovery algorithms have been devised, it is not quite clear what is the optimal amount of exchanged messages and overhead among the nodes in order to perform desirable tasks. Distributed approaches may suffer from slower convergence in the network and could cause oscillations among the interacting nodes. Distributed approaches may not able to solve the problems as optimally as their centralised counterparts. On the other hand, with centralised approaches, the delay as the result of propagation to/from central supervisor may increase beyond acceptable levels. Due to the nature of scenarios with security or real-time concerns, using centralised approaches may neither be encouraged nor feasible. For example in WSNs' CH recovery, in many applications, swiftness is more important than accuracy. They may not able to tolerate the time required for the notifications or coordination of nodes in centralised recovery paradigms. Thus, it is preferable that coverage of area be partially repaired and quality of service maintained to a certain acceptable level as soon as damages occur, rather than to have a full recovery. In this case, a reasonable trade-off between recovery response time and accuracy should be considered.

- **Nature and Bio-inspired Recovery**. As nature tends toward optimal or near-optimal solutions [53, 324], (e.g. bio-inspired clustering [206]), interesting recovery strategies may be inferred from nature, such as Glowworm [341, 342], slime mold [401], game theory [402], PSO [403] or cellular automata [404–406]. In the recovery model, nature can inspire many ideas, as many bio-inspired models can be used to model the behaviour of sensor nodes redundantly deployed in a given field that only have control and vision within their neighbourhood ranges. For example, by defining different neighbours in a cellular automata model, the level of abstraction and computation can be kept local and minimum, and an emergent cooperation among sensor nodes could be reached in the recovery of coverage holes [404–406].
- **Real-time Large Scale CHs Detection**. Autonomous nodes with limited visibilities would detect the presence of large coverage holes, but it would be challenging to grasp the scale and shape of these holes without significant message exchanges among the nodes in time sensitive and real-time applications. Swiftly detecting and determining the scope and scale of node failures, especially in the form of large, distributed coverage holes with high accuracy should be considered in more detail.

- **K-Connectivity and K-Coverage**. New recovery models based on the distributed and local interactions of autonomous nodes can be defined to satisfy some degrees of coverage and connectivity requirements. This can be useful in many applications in which nodes are deployed in an isolated environments with minimum or no centralised control. Recovery and restoration via node relocation to achieve k-connectivity and/or k-coverage, especially in 3D environments and networks such as *underwater wireless sensor networks* (UWSNs) efficiently is a challenging task [110] and warrants further study.
- **Energy Scavenging and Harvesting**. Technical advancements, more effective power utilisation and ambient harvesting in (mobile) sensor nodes in WSNs [407–410] in the near future is expected to offer autonomous nodes more flexibility and longevity during the topology control schemes and recovery processes. Further studies would be useful.
- **Boundary Reinforcements and CHs Avoidance**. Node relocation algorithms are mainly developed to address coverage problems and unbalanced deployments in WSNs. In many cases, addressing undesirable effects such as minimum phenomenon around the coverage holes has higher priority than repairing the lost coverages (of the damaged area) and/or node relocation towards the damaged area is not possible (e.g. fire, explosion) [86, 411, 412]. Therefore, based on the application and types of coverage holes, new relocation algorithms could help avoid damaged areas and/or reinforce the nodes at the CHs boundaries. With the newly devised (distributed) node relocation algorithms, autonomous nodes should also move such that an emergent cooperative behaviour can be reached. The avoidance and/or reinforcement of boundary nodes around the CHs (i.e. damaged areas) could be achieved by flexible behaviours such as circulating and/or moving away from the given areas, in addition to possibly increasing the node density around the dam-

aged areas. By using such node relocation algorithms, progressive damages and failures could be prevented and/or isolated in WSNs [277, 310, 411].

- **Appropriate Performance Metrics**. The efficiency of newly devised algorithms can not be acknowledged and seen with their counterparts without using suitable performance metrics in WSNs. With a wide range of performance metrics tailored for different layers of communications networks [413, 414], it is important to either choose or introduce appropriate metrics that correctly reflect the advantages of the newly proposed algorithms over their counterparts, especially if those algorithms are application-specific.
- **Decision Making Under Uncertainty**. Sensor nodes of limited computation and communication power with the local visibility, inevitably suffer from a degree of uncertainty in their decision-makings and responses to events such as node failures. When devising TC schemes and recovery models, in order to cope with the uncertainties relevant to the states of nodes' immediate neighbours, it may be reasonable to harness the models and schemes that encompass such uncertainties per se to address the challenges resulted from nodes' limited information about network status [354, 356, 415–419].
- **Multi-Objective TC Schemes**. Huang et al. [156] pointed out the effect of considering other factors in devising topology control schemes such as proximate nodes around sink nodes for node scheduling. In this thesis, an example of such TC schemes that considers sink nodes while applying node relocation algorithms in WSNs is presented. If their challenges are addressed, TC schemes that consider multiple objectives and trade-offs (e.g. coverage and energy consumptions) can be more effective (Huang et al. [156]). Hybrid TC schemes seem to be able to address of some of these challenges. Therefore, hybrid TCs are interesting and open to further examination and research. One such has been presented in this thesis.

- **Fault Isolation and Avoidance**. By idea of forming an optimal route as in the travelling salesperson problem, Xiangrong et al. [420], considered a mobile node as a scanner to find and isolate faulty nodes by not allowing them to connect to the normal nodes based on the importance and priority of nodes' tree levels, in which nodes are formed, in the deployment area and within their communication regions from base station, and periodically selected/visited all monitoring station as well as static nodes. It is suggested the effect of faulty node isolation in increasing the resilience of network. Even though it is expected that recovery mechanisms and fault management may be different based on the type of faults, it is interesting to note that, even for specific types of fault that may have different features, in the recovery mechanism may be different as well. For example, in the recovery of networks from failures, whether they are random or co-located, the intensity and scale is different. Results from thesis show that, depending on the scale of coverage holes, one relocation algorithm outperforms and/or becomes unsuitable and slow and vice versa. Recovery from random and multiple sporadic, scattered node failures can be achieved with different topology control schemes in the case that failures are correlated and/or co-located in the form of coverage holes.
- **Autonomy and Emergent Cooperation**. Benefitting from light overhead information, swift response, distributed control over agents in complex systems, it is desirable but challenging to have a global behaviour from local interaction of agents of limited vision [361, 421, 422]. Many models of such designs should be exhaustively examined [281, 423, 424]; this should be performed with different parameters that are empirically and patiently tuned. This is because, with a small change in certain parameters relating to the agents, the environment, and their reaction rules, a drastic change

in the global behaviour of complex systems may be observed [425]. For a better understanding of such sharp transitions and drastic changes to a global behaviour as the result of small modifications in individual behaviours/parameters, many models, such as percolation, flocking, forest fire spreading and 'heroes and cowards', are examined and visually shown in Netlogo (Netlogo is a multi-agent, programmable modelling environment) [280, 281]. For example, in the fire spreading forest model, modification and small changes in one or more of the model's parameters (i.e. density of trees, probability of spread, wind directions and possibility of sparks across long distances) would drastically change the global behaviour of the model [280, 281]. The level and detail of the agent (node) abstract model is another key factor to consider. Furthermore, design differentiation between intentional cooperation and emergent cooperation in complex systems is crucial [159]. It should be noted that devising the distributed systems (nodes) to show emergent cooperation is not straightforward or trivial, and there is no guarantee that micro limited behaviour of agents (nodes) will result in the expected and desirable macro and global behaviour in response to predictable or unpredictable events [212, 213, 317].

Bibliography

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[3] M. Obaidat and S. Misra, *Principles of Wireless Sensor Networks.* Cambridge University Press, 2014.

[4] S. Junnila, H. Kailanto, J. Merilahti, A.-M. Vainio, A. Vehkaoja, M. Zakrzewski, and J. Hyttinen, "Wireless, multipurpose in-home health monitoring platform: Two case trials," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 447–455, March 2010.

[5] M. Dunbabin and L. Marques, "Robots for environmental monitoring: Significant advancements and applications," *Robotics Automation Magazine, IEEE*, vol. 19, no. 1, pp. 24 –39, march 2012.

[6] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring," *IEEE Sensors Journal*, vol. 11, no. 1, pp. 45–55, Jan 2011.

[7] M. Srbinovska, C. Gavrovski, V. Dimcev, A. Krkoleva, and V. Borozan, "Environmental parameters monitoring in precision agriculture using wireless sensor networks," *Journal of Cleaner Production*, vol. 88, no. Supplement C, pp. 297 – 307, 2015.

[8] A. Bayo, D. Antolín, N. Medrano, B. Calvo, and S. Celma, "Early detection and monitoring of forest fire with a wireless sensor network system," *Procedia Engineering*, vol. 5, pp. 248 – 251, 2010.

[9] Y. Li, Z. Wang, and Y. Song, "Wireless sensor network design for wildfire monitoring," in *The Sixth World Congress on Intelligent Control and Automation, WCICA 2006*, vol. 1, 2006, pp. 109 –113.

[10] S. Bhima, A. Gogada, and R. Garimella, "A tsunami warning system employing level controlled gossiping in wireless sensor networks," in *Proceedings of the 4th international conference on Distributed computing and internet technology*, ser. ICDCIT'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 306–313.

[11] M. Suzuki, S. Saruwatari, N. Kurata, and H. Morikawa, "A high-density earthquake monitoring system using wireless sensor networks," in *Proceedings of the 5th international conference on Embedded networked sensor systems*, ser. SenSys 2007. New York, NY, USA: ACM, 2007, pp. 373–374.

[12] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh, "Deploying a wireless sensor network on an active volcano," *Internet Computing, IEEE*,

vol. 10, no. 2, pp. 18–25, March-April 2006.

[13] H. Yang, Y. Qin, G. Feng, and H. Ci, “Online monitoring of geological CO_2 storage and leakage based on wireless sensor networks,” *IEEE Sensors Journal*, vol. 13, no. 2, pp. 556–562, Feb 2013.

[14] H. Qi, S. Iyengar, and K. Chakrabarty, “Distributed sensor networks-a review of recent research,” *Journal of the Franklin Institute*, vol. 338, no. 6, pp. 655 – 668, 2001.

[15] M. Yu, H. Mokhtar, and M. Merabti, “Fault management in wireless sensor networks,” *Wireless Communications, IEEE*, vol. 14, no. 6, pp. 13 –19, December 2007.

[16] L. Paradis and Q. Han, “A survey of fault management in wireless sensor networks,” *J. Netw. Syst. Manage.*, vol. 15, no. 2, pp. 171–190, June 2007.

[17] Y. Zou and K. Chakrabarty, “Sensor deployment and target localization in distributed sensor networks,” *ACM Trans. Embed. Comput. Syst.*, vol. 3, pp. 61–91, February 2004.

[18] J. Chen, E. Shen, and Y. Sun, “The deployment algorithms in wireless sensor net works: A survey,” *Information Technology Journal*, vol. 8, no. 3, pp. 293–301, 2009.

[19] A. Howard, M. J. Mataric, and G. S. Sukhatme, “Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem,” in *In Proceedings of the 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02)*, Fukuoka, Japan, June 25-27 2002, pp. 299–308.

[20] S. Kushwaha, V. Kumar, and S. Jain, “Node architectures and its deployment in wireless sensor networks: A survey,” in *High Performance Architecture and Grid Computing*, ser. Communications in Computer and Information Science, A. Mantri, S. Nandi, G. Kumar, and S. Kumar, Eds. Springer Berlin Heidelberg, 2011, vol. 169, pp. 515–526.

[21] G. Wang, G. Cao, and T. La Porta, “Movement-assisted sensor deployment,” in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 4, March, pp. 2469–2479 vol.4.

[22] J. Luo, D. Wang, and Q. Zhang, “Double mobility: coverage of the sea surface with mobile sensor networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 13, pp. 52–55, June 2009.

[23] ——, “On the double mobility problem for water surface coverage with mobile sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, p. 1, 2011.

[24] N. Ahmed, S. S. Kanhere, and S. Jha, “The holes problem in wireless sensor networks: a survey,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 2, pp. 4–18, 2005.

[25] B. Karp and H. T. Kung, “Gpsr: greedy perimeter stateless routing for wireless networks,” in *Proceedings of the 6th annual international conference on Mobile computing and networking*, ser. MobiCom ’00. New York, NY, USA: ACM, 2000, pp. 243–254.

[26] F. Yu, Y. Choi, S. Park, Y. Tian, and S.-H. Kim, “A hole geometric modeling in wireless sensor networks,” in *2007 International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007.*, September 2007, pp. 2432 –2435.

[27] A. Aziz, Y. Sekercioglu, P. Fitzpatrick, and M. Ivanovich, “A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks,” *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1 –24, 2012.

[28] P. Costa, M. Cesana, S. Brambilla, and L. Casartelli, “A cooperative approach for topology control in wireless sensor networks,” *Pervasive and Mobile Computing*, vol. 5, no. 5, pp. 526 – 541, 2009.

[29] Z. Jiang, J. Wu, A. Agah, and B. Lu, “Topology control for secured coverage in wireless sensor networks,” in *IEEE International Conference on Mobile Adhoc and Sensor Systems, (MASS) 2007*, October 2007, pp. 1–6.

[30] P. K. Sahoo, J.-P. Sheu, and K.-Y. Hsieh, “Power control based topology construction for

the distributed wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2774 – 2785, 2007.

[31] P. Santi, *Topology Control in Wireless Ad Hoc and Sensor Networks*. John Wiley & Sons, 2005.

[32] Y. Wang, "Topology control for wireless sensor networks," in *Wireless Sensor Networks and Applications*, ser. Signals and Communication Technology, Y. Li, M. T. Thai, and W. Wu, Eds. Springer US, 2008, pp. 113–147.

[33] N. Xu, A. Huang, T.-W. Hou, and H.-H. Chen, "Coverage and connectivity guaranteed topology control algorithm for cluster-based wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 12, no. 1, pp. 23–32, 2012.

[34] B. You, G. Chen, and W. Guo, "Topology control in wireless sensor networks based on discrete particle swarm optimization," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009.*, vol. 1, November 2009, pp. 269–273.

[35] N. Bartolini, T. Calamoneri, E. G. Fusco, A. Massini, and S. Silvestri, "Push & pull: autonomous deployment of mobile sensors for a complete coverage," *Wirel. Netw.*, vol. 16, no. 3, pp. 607–625, April 2010.

[36] A. Casteigts, J. Albert, S. Chaumette, A. Nayak, and I. Stojmenovic, "Biconnecting a network of mobile robots using virtual angular forces," *Computer Communications*, 2011.

[37] C.-Y. Chang, H.-R. Chang, H.-J. Liu, and S.-W. Chang, "On providing temporal full-coverage by applying energy-efficient hole-movement strategies for mobile wsns," in *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, 2007, pp. 2778–2783.

[38] B. Wang, H. B. Lim, and D. Ma, "A survey of movement strategies for improving network coverage in wireless sensor networks," *Computer Communications*, vol. 32, no. 13-14, pp. 1427–1436, 2009.

[39] G. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 6, pp. 640–652, June 2006.

[40] C. Xiao, Y. Peng, and M. Yu, "The deployment method and movement control strategy in mobile wireless sensor networks," in *International Symposium on Computer Science and Computational Technology, ISCSCT 2008*, vol. 2, December 2008, pp. 520–523.

[41] B. Liu, P. Brass, O. Dousse, P. Nain, and D. Towsley, "Mobility improves coverage of sensor networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, ser. MobiHoc 2005. New York, NY, USA: ACM, 2005, pp. 300–308.

[42] M. Garetto, M. Gribaudo, C.-F. Chiasserini, and E. Leonardi, "Sensor deployment and relocation: A unified scheme," *Journal of Computer Science and Technology*, vol. 23, no. 3, pp. 400–412, 2008.

[43] S. Chellappan, W. Gu, X. Bai, D. Xuan, B. Ma, and K. Zhang, "Deploying wireless sensor networks under limited mobility constraints," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, pp. 1142 –1157, October 2007.

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

[45] N. Leonard, D. Paley, F. Lekien, R. Sepulchre, D. Fratantoni, and R. Davis, "Collective motion, sensor networks, and ocean sampling," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 48–74, Jan 2007.

[46] X. Li, R. Falcon, A. Nayak, and I. Stojmenovic, "Servicing wireless sensor networks by mobile robots," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 147–154, 2012.

[47] N. Patwari, J. Ash, S. Kyperountas, A. Hero, R. Moses, and N. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *Signal Processing Magazine,*

IEEE, vol. 22, no. 4, pp. 54–69, July.

[48] N. Patwari, I. Hero, AlfredO., and J. Costa, "Learning sensor location from signal strength and connectivity," in *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*, ser. Advances in Information Security, R. Poovendran, S. Roy, and C. Wang, Eds. Springer US, 2007, vol. 30, pp. 57–81.

[49] J. E. Doran, S. Franklin, N. R. Jennings, and T. J. Norman, "On cooperation in multi-agent systems," *Knowl. Eng. Rev.*, vol. 12, pp. 309–314, September 1997.

[50] N. Ito, T. Esaki, and N. Ishii, "A cooperative agent model by forming a group," in *2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02.*, vol. 2. IEEE, December 2002, pp. 1260–1265 vol.2.

[51] Y. Yoo and D. Agrawal, "Mobile sensor relocation to prolong the lifetime of wireless sensor networks," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, May 2008, pp. 193 –197.

[52] M. D. Coles, D. Azzi, B. P. Haynes, and A. Hewitt, "A bayesian network approach to a biologically inspired motion strategy for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1217 – 1228, 2009.

[53] S. Toumpis, "Mother nature knows best: A survey of recent results on wireless networks based on analogies with physics," *Computer Networks*, vol. 52, no. 2, pp. 360 – 383, 2008.

[54] G. Wang, G. Cao, T. La Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 4, march 2005, pp. 2302 –2312 vol. 4.

[55] P. Smith, D. Hutchison, J. Sterbenz, M. Schöller, A. Fessi, M. Karaliopoulos, C. Lac, and B. Plattner, "Network resilience: a systematic approach," *Communications Magazine, IEEE*, vol. 49, no. 7, pp. 88 –97, july 2011.

[56] J. P. Sterbenz, D. Hutchison, E. K. Çetinkaya, A. Jabbar, J. P. Rohrer, M. Schöller, and P. Smith, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Computer Networks*, vol. 54, no. 8, pp. 1245 – 1265, 2010.

[57] P. Cholda, J. Tapolcai, T. Cinkler, K. Wajda, and A. Jajszczyk, "Quality of resilience as a network reliability characterization tool," *Network, IEEE*, vol. 23, no. 2, pp. 11–19, March 2009.

[58] S. Dimitrov, P. Krishnan, C. Mallows, J. Meloche, and S. Yajnik, "r3: Resilient random regular graphs," in *Distributed Computing*, ser. Lecture Notes in Computer Science, G. Taubenfeld, Ed. Springer Berlin Heidelberg, 2008, vol. 5218, pp. 137–151.

[59] A. Gutfraind, "Constructing networks for cascade resilience," *arXiv preprint arXiv:0906.0786*, 2009.

[60] W. Najjar and J.-L. Gaudiot, "Network resilience: a measure of network fault tolerance," *IEEE Transactions on Computers*, vol. 39, no. 2, pp. 174–181, February 1990.

[61] H. Nakayama, N. Ansari, A. Jamalipour, and N. Kato, "Fault-resilient sensing in wireless sensor networks," *Computer Communications*, vol. 30, no. 11–12, pp. 2375 – 2384, 2007.

[62] G. Klau and R. Weiskircher, "Robustness and resilience," in *Network Analysis*, ser. Lecture Notes in Computer Science, U. Brandes and T. Erlebach, Eds. Springer Berlin Heidelberg, 2005, vol. 3418, pp. 417–437.

[63] V. Pappasa, D. C. Vermaa, A. Swamib, and B. J. Koa, "On the resiliency of wireless sensor networks with circulation graphs," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6981, 2008, p. 5.

[64] S. S. Iyengar, K. G. Boroojeni, and N. Balakrishnan, *Mathematical Theories of Distributed Sensor Networks*. Springer Publishing Company, Incorporated, 2014.

[65] M. A. Guvensan and A. G. Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1238 – 1255, 2011.

[66] M. Li, Z. Li, and A. Vasilakos, "A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2538–2557, December 2013.

[67] Anna Förster, *Introduction to Wireless Sensor Networks*, 1st ed. Wiley-IEEE Press, July 2016.

[68] X. Fan, Z. Zhang, and H. Wang, "The probabilistic sense model for coverage hole elimination in wsn," in *2014 33rd Chinese Control Conference (CCC)*, July 2014, pp. 422–427.

[69] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619 – 632, 2012.

[70] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, 2005.

[71] E. Alba and G. Molina, "Optimal wireless sensor network layout with metaheuristics: Solving a large scale instance," in *Large-Scale Scientific Computing*, I. Lirkov, S. Margenov, and J. Waśniewski, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 527–535.

[72] J. Allred, A. B. Hasan, S. Panichsakul, W. Pisano, P. Gray, J. Huang, R. Han, D. Lawrence, and K. Mohseni, "Sensorflock: an airborne wireless sensor network of micro-air vehicles," in *Proceedings of the 5th international conference on Embedded networked sensor systems*, ser. SenSys 2007. New York, NY, USA: ACM, 2007, pp. 117–129.

[73] T. Arici and Y. Altunbasak, "Adaptive sensing for environment monitoring using wireless sensor networks," in *Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE*, vol. 4, March 2004, pp. 2347–2352 Vol.4.

[74] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, "Brimon: a sensor network system for railway bridge monitoring," in *Proceedings of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08. New York, NY, USA: ACM, 2008, pp. 2–14.

[75] M. Durisic, Z. Tafa, G. Dimic, and V. Milutinovic, "A survey of military applications of wireless sensor networks," in *2012 Mediterranean Conference on Embedded Computing (MECO)*, june 2012, pp. 196–199.

[76] J. K. Hart and K. Martinez, "Environmental sensor networks: A revolution in the earth system science?" *Earth-Science Reviews*, vol. 78, no. 3, pp. 177 – 191, 2006.

[77] M. Jiang, Z. Guo, F. Hong, Y. Ma, and H. Luo, "Oceansense: A practical wireless sensor network on the surface of the sea," in *IEEE International Conference on Pervasive Computing and Communications, 2009. PerCom 2009*, March 2009, pp. 1–5.

[78] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, "Health monitoring of civil infrastructures using wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, ser. IPSN 2007. New York, NY, USA: ACM, 2007, pp. 254–263.

[79] M. Li and Y. Liu, "Underground structure monitoring with wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, ser. IPSN 2007. New York, NY, USA: ACM, 2007, pp. 69–78.

[80] P. Neumann, S. Asadi, A. Lilienthal, M. Bartholmai, and J. Schiller, "Autonomous gas-sensitive microdrone: Wind vector estimation and gas distribution mapping," *Robotics Automation Magazine, IEEE*, vol. 19, no. 1, pp. 50 –61, march 2012.

[81] S. Nittel, "A survey of geosensor networks: advances in dynamic environmental monitoring," *Sensors*, vol. 9, no. 7, pp. 5664–5678, 2009.

[82] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09.

New York, NY, USA: ACM, 2009, pp. 305–318.

[83] K. Zhang, Q. Zhang, T. Jiang, and P. Guo, "A navigation system based on a sensor network without exit and locations," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 311–312.

[84] L. Jin, J. Jia, G. Chang, and X. Wang, "Restoration of coverage blind spots in wireless sensor networks based on ant colony algorithm," in *Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation*, ser. GEC '09. New York, NY, USA: ACM, 2009, pp. 847–850.

[85] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal wsn deployment models for air pollution monitoring," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2017.

[86] R. R. Selmic, V. V. Phoha, and A. Serwadda, *Wireless Sensor Networks: Security, Coverage, and Localization.* Springer, 2016, vol. 69.

[87] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles.* Springer, 2014.

[88] M. Z. A. Bhuiyan, G. Wang, J. Cao, and J. Wu, "Deploying wireless sensor networks with fault-tolerance for structural health monitoring," *IEEE Transactions on Computers*, vol. 64, no. 2, pp. 382–395, February 2015.

[89] G. Han, C. Zhang, L. Shu, N. Sun, and Q. Li, "A survey on deployment algorithms in underwater acoustic sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, 2013.

[90] J. Partan, J. Kurose, and B. N. Levine, "A survey of practical issues in underwater networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 23–33, October 2007.

[91] P. Antil and A. Malik, "Hole detection for quantifying connectivity in wireless sensor networks: A survey," *Journal of Computer Networks and Communications*, vol. 2014, 2014.

[92] A. Förster, "Teaching networks how to learn," Ph.D. dissertation, Università della Svizzera italiana, 2009.

[93] R. V. Kulkarni, A. Forster, and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 68–96, First 2011.

[94] L. Madarász and J. Živčák, *Aspects of Computational Intelligence: Theory and Applications: Revised and Selected Papers of the 15th IEEE International Conference on Intelligent Engineering Systems 2011, INES 2011.* Springer Science & Business Media, 2012, vol. 2.

[95] A. Yera Gomez, "Task distribution framework using zookeeper: using hierarchical state machines to design and implement zookeeper recipes," Master's thesis, School of Information and Communication Technology (ICT), KTH Royal Institute of Technology, 2013.

[96] D. Nicholson, S. Ramchurn, and A. Rogers, "Information-based control of decentralised sensor networks," in *Defence Industry Applications of Autonomous Agents and Multi-Agent Systems*, ser. Whitestein Series in Software Agent Technologies and Autonomic Computing, M. Pěchouček, S. Thompson, and H. Voos, Eds. Birkhäuser Basel, 2008, pp. 15–32.

[97] J. Rezazadeh, "Mobile wireles sensor networks overview," *International Journal of Computer Communications and Networks (IJCCN)*, vol. 2, no. 1, 2012.

[98] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–17.

[99] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2001, pp. 1380–1387 vol.3.

[100] J. Wang, R. Ghosh, and S. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

[101] Y. Liu and Z. Yang, *Location, localization, and localizability: location-awareness technology for wireless networks*. Springer, 2010.

[102] Y.-Q. Miao, "A study of mobility models in mobile surveillance systems," Ph.D. dissertation, Electrical and Computer Engineering, University of Waterloo, 2010.

[103] A. T. Murray, K. Kim, J. W. Davis, R. Machiraju, and R. Parent, "Coverage optimization to support security monitoring," *Computers, Environment and Urban Systems*, vol. 31, no. 2, pp. 133 – 147, 2007.

[104] S. J. Habib, "Modeling and simulating coverage in sensor networks," *Computer Communications*, vol. 30, no. 5, pp. 1029 – 1035, 2007.

[105] C. Costanzo, V. Loscrí, E. Natalizio, and T. Razafindralambo, "Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network," *Comput. Commun.*, vol. 35, no. 9, pp. 1047–1055, May 2012.

[106] W. Xiaoling, S. Lei, Y. Jie, X. Hui, J. Cho, and S. Lee, "Swarm based sensor deployment optimization in ad hoc sensor networks," in *Embedded Software and Systems*, ser. Lecture Notes in Computer Science, L. Yang, X. Zhou, W. Zhao, Z. Wu, Y. Zhu, and M. Lin, Eds. Springer Berlin Heidelberg, 2005, vol. 3820, pp. 533–541.

[107] X. Lu, Y. Ding, and K. Hao, "Immune clonal selection algorithm for target coverage of wireless sensor networks," *International Journal of Modelling, Identification and Control*, vol. 12, no. 1-2, pp. 119–124, 2011.

[108] C. Chibaya and S. Bangay, "Flock inspired area coverage using wireless boid-like sensor agents," in *Tenth International Conference on Computer Modeling and Simulation, 2008. UKSIM 2008.*, April 2008, pp. 144–149.

[109] H. Zhao, Q. Zhang, L. Zhang, and Y. Wang, "A novel sensor deployment approach using fruit fly optimization algorithm in wireless sensor networks," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1, August 2015, pp. 1292–1297.

[110] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Computer Networks*, vol. 58, no. 0, pp. 254 – 283, 2014.

[111] X. Wang, S. Wang, and J. Ma, "Dynamic deployment optimization in wireless sensor networks," in *Intelligent Control and Automation*, ser. Lecture Notes in Control and Information Sciences, D.-S. Huang, K. Li, and G. Irwin, Eds. Springer Berlin Heidelberg, 2006, vol. 344, pp. 182–187.

[112] A. Kott and T. Abdelzaher, "Resiliency and robustness of complex systems and networks," *Adaptive, Dynamic, and Resilient Systems*, p. 67, 2014.

[113] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc '03. New York, NY, USA: ACM, 2003, pp. 201–212.

[114] X. Bai, Z. Yun, D. Xuan, W. Jia, and W. Zhao, "Pattern mutation in wireless sensor deployment," in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.

[115] D. W. Gage, "Command control for many-robot systems," DTIC Document, Tech. Rep., 1992.

[116] S. Yang, F. Dai, M. Cardei, J. Wu, and F. Patterson, "On connected multiple point

coverage in wireless sensor networks," *International Journal of Wireless Information Networks*, vol. 13, no. 4, pp. 289–301, 2006.

[117] X. Li, N. Mitton, I. Ryl, and D. Simplot, "Localized sensor self-deployment with coverage guarantee in complex environment," in *Ad-Hoc, Mobile and Wireless Networks*, ser. Lecture Notes in Computer Science, P. Ruiz and J. Garcia-Luna-Aceves, Eds. Springer Berlin Heidelberg, 2009, vol. 5793, pp. 138–151.

[118] X. Li, H. Frey, N. Santoro, and I. Stojmenovic, "Focused-coverage by mobile sensor networks," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, MASS '09.*, October 2009, pp. 466–475.

[119] ——, "Localized sensor self-deployment for guaranteed coverage radius maximization," in *2009 IEEE International Conference on Communications, ICC '09.*, June 2009, pp. 1–5.

[120] J. Tian, X. Liang, and G. Wang, "Deployment and reallocation in mobile survivability-heterogeneous wireless sensor networks for barrier coverage," *Ad Hoc Networks*, pp. –, 2015.

[121] D. Tao and T.-Y. Wu, "A survey on barrier coverage problem in directional sensor networks," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 876–885, February 2015.

[122] T. Liu, H. Lin, C. Wang, K. Peng, D. Wang, T. Deng, and H. Jiang, "Chain-based barrier coverage in wsns: toward identifying and repairing weak zones," *Wireless Networks*, pp. 1–14, 2015.

[123] S. Kumar, T. H. Lai, and A. Arora, "Barrier coverage with wireless sensors," in *Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '05. New York, NY, USA: ACM, 2005, pp. 284–298.

[124] Y. Chen, C. Li, J. Yu, H. Zhu, and Y. Sun, "k-perimeter coverage evaluation and deployment in wireless sensor networks," in *Wireless Algorithms, Systems, and Applications*, ser. Lecture Notes in Computer Science, K. Xu and H. Zhu, Eds. Springer International Publishing, 2015, vol. 9204, pp. 50–59.

[125] X. Bai, L. Ding, J. Teng, S. Chellappan, C. Xu, and D. Xuan, "Directed coverage in wireless sensor networks: Concept and quality," in *IEEE 6th International Conference on Mobile Ad hoc and Sensor Systems, 2009. MASS '09.*, October 2009, pp. 476–485.

[126] C. Liu and G. Cao, "Spatial-temporal coverage optimization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 4, pp. 465–478, April 2011.

[127] H. M. Ammari, *The Art of Wireless Sensor Networks: Volume 2 Advanced Topics and Applications*. Springer Publishing Company, Incorporated, 2014.

[128] C.-Y. Chang, H.-j. Liu, and H.-r. Chang, "Energy-balanced strategies for hole movement in mobile wireless sensor networks," , 2006.

[129] C.-Y. Chang, W.-C. Chu, C.-Y. Lin, and C.-F. Cheng, "Energy-balanced hole-movement mechanism for temporal full-coverage in mobile wsns," in *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, ser. IWCMC '10. New York, NY, USA: ACM, 2010, pp. 89–93.

[130] M. A. Batalin and G. S. Sukhatme, "Multi-robot dynamic coverage of a planar bounded environment," DTIC Document, Tech. Rep., 2003.

[131] G. Wang, G. Cao, and T. La Porta, "Proxy-based sensor deployment for mobile sensor networks," in *2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems,*, October 2004, pp. 493–502.

[132] S. Chellappan, X. Bai, B. Ma, D. Xuan, and C. Xu, "Mobility limited flip-based sensor networks deployment," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 18, no. 2, pp. 199–211, February 2007.

[133] B. Wang, *Coverage Control in Sensor Networks*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[134] R. R. Roy, *Handbook of mobile ad hoc networks for mobility models.* New York, NY: Springer, 2011.
[135] W. Wang, V. Srinivasan, and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proc. ACM MobiCom*, vol. 7, 2007, pp. 39–50.
[136] J. Li, Y. Zhou, and L. Lamont, "Analysis of the impact of swarm mobility on performance of routing protocols in manets," in *2009 IEEE 28th International Performance Computing and Communications Conference (IPCCC)*, December 2009, pp. 193–199.
[137] K. Ma, Y. Zhang, and W. Trappe, "Managing the mobility of a mobile sensor network using network dynamics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 106–120, January 2008.
[138] F. Bai, N. Sadagopan, and A. Helmy, "The {IMPORTANT} framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks," *Ad Hoc Networks*, vol. 1, no. 4, pp. 383 – 403, 2003.
[139] P. Santi, *Mobility models for next generation wireless networks: ad hoc, vehicular and mesh networks.* John Wiley & Sons, 2012.
[140] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, 2002.
[141] J. Luo, D. Wang, and Q. Zhang, "Double mobility: Coverage of the sea surface with mobile sensor networks," in *INFOCOM 2009, IEEE*, April 2009, pp. 118–126.
[142] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, ser. MSWiM '99. New York, NY, USA: ACM, 1999, pp. 53–60.
[143] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, March 2006.
[144] K. Suzuki and T. Ikegami, "Shapes and self-movement in protocell systems," *Artificial Life*, vol. 15, no. 1, pp. 59–70, January 2009.
[145] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 25–34, August 1987.
[146] N. Aschenbruck, E. Gerhards-Padilla, and P. Martini, "Modeling mobility in disaster area scenarios," *Performance Evaluation*, vol. 66, no. 12, pp. 773 – 790, 2009.
[147] L. Tang, S. Vrbsky, and X. Hong, "Collaborated camouflaging mobility for mobile privacy," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, November 2008, pp. 1–5.
[148] J. F. Sanford, M. Potkonjak, and S. Slijepcevic, *Localization in wireless networks: Foundations and applications.* Springer Science & Business Media, 2012.
[149] F. Franceschini, M. Galetto, D. Maisano, and L. Mastrogiacomo, "A review of localization algorithms for distributed wireless sensor networks in manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 22, no. 7, pp. 698–716, 2009.
[150] J.-C. Laprie, "From dependability to resilience," in *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks.* Citeseer, 2008, pp. G8–G9.
[151] S. Gupta, S. Verma, and R. K. Abrol, "Towards achieving reliability in wireless sensor networks–a survey," *International Journal of Control and Automation*, vol. 8, no. 5, pp. 417–440, 2015.
[152] W. D. Grover, *Mesh-based survivable transport networks: options and strategies for optical, MPLS, SONET and ATM networking.* Upper Saddle River, NJ, USA: Prentice Hall Professional, 2003.
[153] J. Vasseur, M. Pickavet, and P. Demeester, *Network recovery.* San Francisco, CA, USA:

Elsevier, 2004.

[154] O. M. Al-Kofahi and A. E. Kamal, "Survivability strategies in multihop wireless networks [accepted from open call]," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 71–80, October 2010.

[155] G. Deffuant and N. Gilbert, *Viability and resilience of complex systems: concepts, methods and case studies from ecology and society.* Springer, 2011.

[156] Y. Huang, J.-F. Martínez, J. Sendra, and L. López, "Resilientwireless sensor networks using topology control: A review," *Sensors*, vol. 15, no. 10, p. 24735, 2015.

[157] V. Westmark, "A definition for information system survivability," in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004.*, January 2004, pp. 10 pp.–.

[158] J. C. Whitson and J. E. Ramirez-Marquez, "Resiliency as a component importance measure in network reliability," *Reliability Engineering & System Safety*, vol. 94, no. 10, pp. 1685 – 1693, 2009.

[159] F. Dressler *et al.*, "Self-organization in ad hoc networks: Overview and classification," *University of Erlangen, Dept. of Computer Science*, vol. 7, pp. 1–12, 2006.

[160] K. Miranda, A. Molinaro, and T. Razafindralambo, "A survey on rapidly deployable solutions for post-disaster networks," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 117–123, April 2016.

[161] F. Dressler, *Self-organization in sensor and actor networks.* John Wiley & Sons, 2008.

[162] V. Pappas, D. C. Verma, A. Swami, and B. J. Ko, "On the resiliency of wireless sensor networks with circulation graphs," in *SPIE Defense and Security Symposium.* International Society for Optics and Photonics, 2008, pp. 698 107–698 107.

[163] V. Pappas, D. Verma, B.-J. Ko, and A. Swami, "A circulatory system approach for wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 4, pp. 706 – 724, 2009.

[164] J. Jones, "Characteristics of pattern formation and evolution in approximations of physarum transport networks," *Artificial life*, vol. 16, no. 2, pp. 127–153, 2010.

[165] J. Jones and A. Adamatzky, "Slime mould inspired generalised voronoi diagrams with repulsive fields," *arXiv preprint arXiv:1503.06973*, 2015.

[166] J. Jones, *From Pattern Formation to Material Computation: Multi-agent Modelling of Physarum polycephalum.* Springer, 2015, vol. 15.

[167] J. Ash and D. Newth, "Optimizing complex networks for resilience against cascading failure," *Physica A: Statistical Mechanics and its Applications*, vol. 380, pp. 673 – 683, 2007.

[168] R. Criado, J. Pello, M. Romance, and M. a. Vela-Pérez, "(ψ, p, q)-vulnerabilities: A unified approach to network robustness," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 19, no. 1, p. 013133, 2009.

[169] Z. Kong and E. Yeh, "Percolation processes and wireless network resilience," in *Information Theory and Applications Workshop, 2008*, January 2008, pp. 461–470.

[170] Z. Kong and E. M. Yeh, "Percolation processes and wireless network resilience to degree-dependent and cascading node failures," *CoRR*, vol. abs/0902.4447, 2009.

[171] S. O. Dulman, T. Nieberg, J. Wu, and P. J. M. Havinga, "Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks," http://eprints.eemcs.utwente.nl/5834/, Centre for Telematics and Information Technology University of Twente, Enschede, Technical Report TR-CTIT-03-14, April 2003.

[172] Y.-J. Han, M.-W. Park, and T.-M. Chung, "Secdeach: Secure and resilient dynamic clustering protocol preserving data privacy in wsns," in *Computational Science and Its Applications – ICCSA 2010*, ser. Lecture Notes in Computer Science, D. Taniar, O. Gervasi, B. Murgante, E. Pardede, and B. Apduhan, Eds. Springer Berlin Heidelberg, 2010, vol. 6018, pp. 142–157.

[173] G. Zhan, W. Shi, and J. Deng, "Sensortrust: A resilient trust model for wsns," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 411–412.

[174] R. Di Pietro, G. Oligeri, C. Soriente, and G. Tsudik, "Intrusion-resilience in mobile unattended wsns," in *INFOCOM, 2010 Proceedings IEEE*, March 2010, pp. 1–9.

[175] ——, "United we stand: Intrusion resilience in mobile unattended wsns," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1456–1468, July 2013.

[176] A. Miyaji and K. Omote, "Self-healing schemes suitable for various wsns," in *Internet and Distributed Computing Systems*, ser. Lecture Notes in Computer Science, M. Pathan, G. Wei, and G. Fortino, Eds. Springer Berlin Heidelberg, 2013, vol. 8223, pp. 92–105.

[177] N. Abu-Ghazaleh, K.-D. Kang, and K. Liu, "Towards resilient geographic routing in wsns," in *Proceedings of the 1st ACM International Workshop on Quality of Service &Amp; Security in Wireless and Mobile Networks*, ser. Q2SWinet '05. New York, NY, USA: ACM, 2005, pp. 71–78.

[178] M. Valero, M. Xu, N. Mancuso, W.-Z. Song, and R. Beyah, "Edr2: A sink failure resilient approach for wsns," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 616–621.

[179] Y. O. Lee and A. N. Reddy, "Constructing disjoint paths for failure recovery and multipath routing," *Computer Networks*, vol. 56, no. 2, pp. 719 – 730, 2012.

[180] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2–3, pp. 293 – 315, 2003.

[181] M. Li and B. Yang, "A survey on topology issues in wireless sensor network." in *ICWN*, 2006, p. 503.

[182] M. Labrador and P. Wightman, *Topology Control in Wireless Sensor Networks: with a companion simulation tool for teaching and research.* Springer, 2009.

[183] P. Santi, "Topology control in wireless ad hoc and sensor networks," *ACM Comput. Surv.*, vol. 37, no. 2, pp. 164–194, jun 2005.

[184] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 5:1–5:39, February 2009.

[185] B. Tong and W. Tavanapong, "On discovering sensing coverage holes in large-scale sensor networks," *Computer Science, Iowa State University., Tech. Rep. TR*, pp. 06–03, 2006.

[186] Y.-T. Hou, T.-C. Lee, B.-C. Jeng, and C.-M. Chen, "Optimal coverage deployment for wireless sensor networks," in *The 8th International Conference on Advanced Communication Technology, 2006. ICACT 2006.*, vol. 1, February 2006, pp. 5 pp.–527.

[187] J. Robinson and E. Knightly, "A performance study of deployment factors in wireless mesh networks," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, May 2007, pp. 2054–2062.

[188] I. Khoufi, P. Minet, A. Laouiti, and E. Livolant, "A simple method for the deployment of wireless sensors to ensure full coverage of an irregular area with obstacles," in *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '14. New York, NY, USA: ACM, 2014, pp. 203–210.

[189] S. Mnasri, N. Nasri, and T. Val, "The deployment in the wireless sensor networks: Methodologies, recent works and applications," in *International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN 2014)*, November 2014.

[190] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621 – 655, 2008.

[191] C.-W. Tsai, P.-W. Tsai, J.-S. Pan, and H.-C. Chao, "Metaheuristics for the deployment problem of wsn: A review," *Microprocessors and Microsystems*, pp. –, 2015.

[192] C.-H. Wu and Y.-C. Chung, *Advances in Grid and Pervasive Computing: Second International Conference, GPC 2007, Paris, France, May 2-4, 2007. Proceedings.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, ch. Heterogeneous Wireless Sensor Network Deployment and Topology Control Based on Irregular Sensor Model, pp. 78–88.

[193] S. Lee, M. Younis, and M. Lee, "Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance," *Ad Hoc Networks*, vol. 24, Part A, pp. 1 – 19, 2015.

[194] R.-S. Chang and S.-H. Wang, "Deployment strategies for wireless sensor networks," *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice: From Principle to Practice*, p. 20, 2010.

[195] A. M. Htun, M. S. Maw, and I. Sasase, "Reduced complexity on mobile sensor deployment and coverage hole healing by using adaptive threshold distance in hybrid wireless sensor networks," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, September 2014, pp. 1547–1552.

[196] N. Pezeshkian, H. G. Nguyen, and A. Burmeister, "Unmanned ground vehicle radio relay deployment system for non-line-of-sight operations," DTIC Document, Tech. Rep., 2007.

[197] R. Niati, N. Yazdani, and M. Nourani, "Deployment of spare nodes in wireless sensor networks," in *2006 IFIP International Conference on Wireless and Optical Communications Networks*, 2006, pp. 5 pp.–5.

[198] Y. Shen, D. Nguyen, and M. Thai, "Adaptive approximation algorithms for hole healing in hybrid wireless sensor networks," in *INFOCOM, 2013 Proceedings IEEE*, April 2013, pp. 1178–1186.

[199] M. R. Senouci, A. Mellouk, and A. Aissani, "Random deployment of wireless sensor networks: a survey and approach," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1-3, pp. 133–146, 2014.

[200] M. R. Senouci, D. Bouguettouche, F. Souilah, and A. Mellouk, "Static wireless sensor networks deployment using an improved binary pso," *International Journal of Communication Systems*, vol. 29, no. 5, pp. 1026–1041, 2016.

[201] M. R. Senouci and A. Mellouk, *Deploying Wireless Sensor Networks: Theory and Practice.* Elsevier, April 2016.

[202] D. Wang, Y. Cheng, Y. Wang, and D. P. Agrawal, "Lifetime enhancement of wireless sensor networks by differentiable node density deployment," in *2006 IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS),*, October 2006, pp. 546 –549.

[203] C.-H. Liu, K.-F. Ssu, and W.-T. Wang, "A moving algorithm for non-uniform deployment in mobile sensor networks," *International Journal of Autonomous and Adaptive Communications Systems*, vol. 4, no. 3, pp. 271–290, 2011.

[204] S. Banergee and S. Khuller, "A clustering scheme for hierarchical routing in wireless networks," Technical Report CS-TR, Tech. Rep., 2000.

[205] J.-A. Jiang, C.-P. Chen, C.-L. Chuang, T.-S. Lin, C.-L. Tseng, E.-C. Yang, and Y.-C. Wang, "Cocma: Energy-efficient coverage control in cluster-based wireless sensor networks using a memetic algorithm," *Sensors*, vol. 9, no. 6, pp. 4918–4940, 2009.

[206] C. Charalambous and S. Cui, "A bio-inspired distributed clustering algorithm for wireless sensor networks," in *Proceedings of the 4th Annual International Conference on Wireless Internet*, ser. WICON '08. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 6:1–6:8.

[207] C. Liu and J. Kaiser, "A survey of mobile ad hoc network routing protocols," Universität Ulm, Fakultät für Informatik., Tech. Rep., 2003.

[208] K. Akkaya and M. Younis, "A survey on routing protocols for wireless sensor networks," *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[209] F. Li and Y. Wang, “Routing in vehicular ad hoc networks: A survey,” *Vehicular Technology Magazine, IEEE*, vol. 2, no. 2, pp. 12–22, June 2007.

[210] R. Vidhyapriya and P. Vanathi, “Energy aware routing for wireless sensor networks,” in *2007 International Conference on Signal Processing, Communications and Networking, ICSCN ’07.*, February 2007, pp. 545–550.

[211] J. Al-Karaki and A. Kamal, “Routing techniques in wireless sensor networks: a survey,” *Wireless Communications, IEEE*, vol. 11, no. 6, pp. 6–28, December 2004.

[212] M. Burgess, *Thinking in Promises*, 1st ed. O’Reilly Media, 2015.

[213] J. A. Bergstra and M. Burgess, *Promise Theory: Principles and Applications*, 1st ed. χtAxis Press, 2014.

[214] A. R. d. Silva, M. Moghaddam, and M. Liu, “Design of low data-rate environmental monitoring applications,” in *The Art of Wireless Sensor Networks*, ser. Signals and Communication Technology, H. M. Ammari, Ed. Springer Berlin Heidelberg, 2014, pp. 51–94.

[215] A. Sekhar, B. Manoj, and C. Murthy, “Dynamic coverage maintenance algorithms for sensor networks with limited mobility,” in *Third IEEE International Conference on Pervasive Computing and Communications, PerCom 2005*, March 2005, pp. 51–60.

[216] X. Wu, Y. Niu, L. Shu, J. Cho, Y. Lee, and S. Lee, “Relay shift based self-deployment for mobility limited sensor networks,” in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, J. Ma, H. Jin, L. Yang, and J.-P. Tsai, Eds. Springer Berlin Heidelberg, 2006, vol. 4159, pp. 556–564.

[217] C. Bettstetter, “On the connectivity of ad hoc networks,” *The computer journal*, vol. 47, no. 4, pp. 432–447, 2004.

[218] P. Santi, D. M. Blough, and F. Vainstein, “A probabilistic analysis for the range assignment problem in ad hoc networks,” in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, ser. MobiHoc ’01. New York, NY, USA: ACM, 2001, pp. 212–220.

[219] S. Mahfoudh and P. Minet, “Survey of energy efficient strategies in wireless ad hoc and sensor networks,” in *2008 Seventh International Conference on Networking, ICN 2008.*, April 2008, pp. 1–7.

[220] A. Cerpa and D. Estrin, “Ascent: adaptive self-configuring sensor networks topologies,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 3, pp. 272–285, July 2004.

[221] R. Wu, J. He, T. Li, and H. Shi, “Energy-efficient coverage hole self-repair in mobile sensor networks,” in *2009 International Conference on New Trends in Information and Service Science, NISS ’09.*, June 2009, pp. 1297–1302.

[222] N. Bartolini, T. Calamoneri, T. La Porta, and S. Silvestri, “Autonomous deployment of heterogeneous mobile sensors,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 753–766, June 2011.

[223] K. Akkaya, A. Thimmapuram, F. Senel, and S. Uludag, “Distributed recovery of actor failures in wireless sensor and actor networks,” in *2008 IEEE Wireless Communications and Networking Conference*, March 2008, pp. 2480–2485.

[224] J.-W. Lin and Y.-T. Chen, “Local replacement for route recovery on a collaborative mobile ad hoc network,” *Wireless Communications and Mobile Computing*, vol. 11, no. 10, pp. 1341–1357, 2011.

[225] M. Senouci, A. Mellouk, and K. Assnoune, “Localized movement-assisted sensor deployment algorithm for holedetection and healing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1267–1277, May 2014.

[226] Y. K. Joshi and M. Younis, “Autonomous recovery from multi-node failure in wireless sensor network,” in *Global Communications Conference (GLOBECOM), 2012 IEEE*, De-

cember 2012, pp. 652–657.

[227] N. Heo and P. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003.*, vol. 3, March 2003, pp. 1597 –1602 vol.3.

[228] N. Aziz, A. Mohemmed, and M. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram," in *International Conference on Networking, Sensing and Control, 2009. ICNSC '09.*, March 2009, pp. 602–607.

[229] J. Yang, Y. Mao, Q. Yu, and S. Leng, "Researches on coverage holes recovery algorithm in wsn," in *2013 International Conference on Communications, Circuits and Systems (ICC-CAS),*, vol. 2, November 2013, pp. 78–83.

[230] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009, vol. 501.

[231] M. Argany, M. Mostafavi, and F. Karimipour, "Voronoi-based approaches for geosensor networks coverage determination and optimisation: A survey," in *2010 International Symposium on Voronoi Diagrams in Science and Engineering (ISVD)*, June 2010, pp. 115–123.

[232] F. Aurenhammer, "Voronoi diagrams a survey of a fundamental geometric data structure," *ACM Comput. Surv.*, vol. 23, pp. 345–405, September 1991.

[233] H. Edelsbrunner, *A short course in computational geometry and topology*. Springer, 2014.

[234] T. W. Sung and C. S. Yang, "A cell-based sensor deployment strategy with improved coverage for mobility assisted hybrid wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 5, no. 3, pp. 189–198, March 2010.

[235] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14–15, pp. 2826 – 2841, 2007.

[236] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232 – 2248, 2009.

[237] P. Guo, G. Zhu, and L. Fang, "An adaptive coverage algorithm for large-scale mobile sensor networks," in *Ubiquitous Intelligence and Computing*, ser. Lecture Notes in Computer Science, J. Ma, H. Jin, L. Yang, and J.-P. Tsai, Eds. Springer Berlin Heidelberg, 2006, vol. 4159, pp. 468–477.

[238] N. Takahashi, T. Nagai, H. Yokoi, and Y. Kakazu, "Control system of flexible structure multi-cell robot using amoeboid self-organization mode," in *Advances in Artificial Life*, ser. Lecture Notes in Computer Science, J. Kelemen and P. Sosík, Eds. Springer Berlin Heidelberg, 2001, vol. 2159, pp. 563–572.

[239] N. Boufares, I. Khoufi, P. Minet, L. Saidane, and Y. Ben Saied, "Three dimensional mobile wireless sensor networks redeployment based on virtual forces," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2015 International*, Aug 2015, pp. 563–568.

[240] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA 2004*, vol. 1, April-1 May 2004, pp. 165 – 171 Vol.1.

[241] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 2, March-3 April 2003, pp. 1293–1303 vol.2.

[242] A. Casteigts, J. Albert, S. Chaumette, A. Nayak, and I. Stojmenovic and, "Biconnecting a network of mobile robots using virtual angular forces," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, September 2010, pp. 1 –5.

[243] N. Vlajic, D. Stevanovic, and G. Spanogiannopoulos, "Sink mobility in wireless sensor

networks: from theory to practice," *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice: From Principle to Practice*, p. 371, 2010.

[244] Y. Chen, M. Li, L. Shu, L. Wang, and T. Hara, "A proportional fairness backoff scheme for funnelling effect in wireless sensor networks," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 6, pp. 585–597, 2012.

[245] Z. Wang, S. Basagni, E. Melachrinoudis, and C. Petrioli, "Exploiting sink mobility for maximizing sensor networks lifetime," in *Proceedings of the 38th Annual Hawaii International Conference on System Sciences, 2005. HICSS '05.*, January 2005, pp. 287a–287a.

[246] H. C. Shih, J. H. Ho, B. Y. Liao, and J. S. Pan, "Fault node recovery algorithm for a wireless sensor network," *IEEE Sensors Journal*, vol. 13, no. 7, pp. 2683–2689, July 2013.

[247] Q.-S. Hua and F. C. Lau, "Joint link scheduling and topology control for wireless sensor networks with sinr constraints," *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice*, pp. 184–208, 2010.

[248] H. Ren and M. Q. H. Meng, "Game-theoretic modeling of joint topology control and power scheduling for wireless sensor networks," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 610–625, October 2009.

[249] M. Kim and M. Mutka, "On relocation of hopping sensors for balanced migration distribution of sensors," in *Computational Science and Its Applications ICCSA 2009*, ser. Lecture Notes in Computer Science, O. Gervasi, D. Taniar, B. Murgante, A. Lagan, Y. Mun, and M. Gavrilova, Eds. Springer Berlin / Heidelberg, 2009, vol. 5593.

[250] M. Zhu and S. Martínez, "Distributed coverage games for energy-aware mobile sensor networks," *SIAM Journal on Control and Optimization*, vol. 51, no. 1, pp. 1–27, 2013.

[251] S. Geethapriya and A. Jawahar, "Performance evaluation of hybrid topology control in wsn," in *2013 International Conference on Communications and Signal Processing (ICCSP),*, April 2013, pp. 9–13.

[252] K. King and S. Nittel, "Efficient data collection and event boundary detection in wireless sensor networks using tiny models," in *Geographic Information Science*, ser. Lecture Notes in Computer Science, S. Fabrikant, T. Reichenbacher, M. van Kreveld, and C. Schlieder, Eds. Springer Berlin Heidelberg, 2010, vol. 6292, pp. 100–114.

[253] G. Jin and S. Nittel, "Ned: An efficient noise-tolerant event and event boundary detection algorithm in wireless sensor networks," in *2006. 7th International Conference on Mobile Data Management, 2006. MDM*, May 2006, pp. 153–153.

[254] H. M. Ammari, *The Art of Wireless Sensor Networks: Volume 1 Fundamentals.* Springer Publishing Company, Incorporated, 2013.

[255] A. Mahapatro and P. M. Khilar, "Fault diagnosis in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 4, pp. 2000–2026, Fourth 2013.

[256] M. Yuvaraja and M. Sabrigiriraj, "Fault detection and recovery scheme for routing and lifetime enhancement in wsn," *Wireless Networks*, pp. 1–11, 2015.

[257] A. Roy, S. Misra, P. Kar, and A. Mondal, "Topology control for self-adaptation in wireless sensor networks with temporary connection impairment," *ACM Trans. Auton. Adapt. Syst.*, vol. 11, no. 4, pp. 21:1–21:34, Jan. 2017.

[258] M.-H. Lee and Y.-H. Choi, "Fault detection of wireless sensor networks," *Computer Communications*, vol. 31, no. 14, pp. 3469 – 3475, 2008.

[259] V. Sreejith, K. R. Anupama, L. J. Gudino, and R. Suriyadeepan, "Partition discovery and connectivity restoration in wsn using mobile relays," in *Proceedings of the 2015 International Conference on Distributed Computing and Networking*, ser. ICDCN '15. New York, NY, USA: ACM, 2015, pp. 36:1–36:9.

[260] H. Liu, A. Nayak, and I. Stojmenović, *Fault-Tolerant Algorithms/Protocols in Wireless*

Sensor Networks. London: Springer London, 2009, pp. 261–291.

[261] F. Koushanfar, M. Potkonjak, and A. Sangiovanni-Vincentelli, "Fault tolerance in wireless sensor networks," *Handbook of Sensor Networks*, 2004.

[262] M. Mannan and S. B. Rana, "Fault tolerance in wireless sensor network," *International Journal of Current Engineering and Technology*, vol. 5, no. 3, pp. 1785–1788, 2015.

[263] M. Korkali, J. G. Veneman, B. F. Tivnan, and P. D. Hines, "Reducing cascading failure risk by increasing infrastructure network interdependency," *arXiv preprint arXiv:1410.6836*, 2014.

[264] G. Fu, R. Dawson, M. Khoury, and S. Bullock, "Interdependent networks: vulnerability analysis and strategies to limit cascading failure," *The European Physical Journal B*, vol. 87, no. 7, 2014.

[265] D. J. Watts, "A simple model of global cascades on random networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 9, pp. 5766–5771, 2002.

[266] Y. Zhang, X. Zhang, W. Fu, Z. Wang, and H. Liu, "Hdre: Coverage hole detection with residual energy in wireless sensor networks," *Journal of Communications and Networks*, vol. 16, no. 5, pp. 493–501, October 2014.

[267] P. Ghosh, J. Gao, A. Gasparri, and B. Krishnamachari, "Distributed hole detection algorithms for wireless sensor networks," in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, October 2014, pp. 257–261.

[268] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing routing holes in sensor networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, March 2004, pp. 2458–2468 vol.4.

[269] K.-F. Huang and S.-J. Wu, "Grid-based hole recovery mechanism using virtual force in hybrid wsns," *Int. J. Distrib. Sen. Netw.*, vol. 2015, pp. 46:46–46:46, jan 2015.

[270] N. Jabeur, N. Sahli, and I. M. Khan, "Survey on sensor holes: A cause-effect-solution perspective," *Procedia Computer Science*, vol. 19, no. 0, pp. 1074 – 1080, 2013.

[271] M. Kouakou, K. Yasumoto, S. Yamamoto, and M. Ito, "Cost-efficient sensor deployment in indoor space with obstacles," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2012, pp. 1–9.

[272] I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, "Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks," in *Proceedings of the 39th annual Symposium on Simulation*, ser. ANSS '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 50–60.

[273] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006.

[274] B. Chazelle and D. Dobkin, "Decomposing a polygon into its convex parts," in *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, ser. STOC '79. New York, NY, USA: ACM, 1979, pp. 38–48.

[275] G. Dini, M. Pelagatti, and I. Savino, "An algorithm for reconnecting wireless sensor network partitions," in *Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, R. Verdone, Ed. Springer Berlin Heidelberg, 2008, vol. 4913, pp. 253–267.

[276] Q. Fang, J. Gao, and L. J. Guibas, "Locating and bypassing holes in sensor networks," *Mob. Netw. Appl.*, vol. 11, no. 2, pp. 187–200, April 2006.

[277] F. Yu, E. lee, Y. Choi, S. Park, D. Lee, Y. tian, and S.-H. Kim, "A modeling for hole problem in wireless sensor networks," in *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, ser. IWCMC 2007. New York, NY, USA: ACM, 2007, pp. 370–375.

[278] F. Reichenbach, R. Salomon, and D. Timmermann, "Distributed obstacle localization in large wireless sensor networks," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, ser. IWCMC '06. New York, NY, USA:

ACM, 2006, pp. 1317–1322.

[279] K.-Y. Hsieh and J.-P. Sheu, “Hole detection and boundary recognition in wireless sensor networks,” in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2009, pp. 72–76.

[280] U. Wilensky, {*NetLogo*}. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999.

[281] U. Wilensky and W. Rand, *An introduction to agent-based modeling*. MIT Press, Cambridge, 2015.

[282] A. A. S. Ali Rafiei, “A network recovery strategy scheme based on network failure scenarios and topologies,” in *International Conference on Communication and Vehicular Technology (ICCVT),Hanoi, Vietnam*, 2010.

[283] M. N. Lima, A. L. dos Santos, and G. Pujolle, “A survey of survivability in mobile ad hoc networks,” *IEEE Communications Surveys Tutorials*, vol. 11, no. 1, pp. 66–77, First 2009.

[284] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 159–170, 2010.

[285] N. Vlajic and N. Moniz, “Self-healing wireless sensor networks: Results that may surprise,” in *Globecom Workshops, 2007 IEEE*, nov. 2007, pp. 1 –6.

[286] M. Fayed and H. T. Mouftah, “Localised convex hulls to identify boundary nodes in sensor networks,” *International Journal of Sensor Networks*, vol. 5, no. 2, pp. 112–125, January 2009.

[287] L. Sitanayah, A. Datta, and R. Cardell-Oliver, “Heuristic algorithm for finding boundary cycles in location-free low density wireless sensor networks,” *Computer Networks*, vol. 54, no. 10, pp. 1630 – 1645, 2010.

[288] F. Yan, P. Martins, and L. Decreusefond, “Connectivity-based distributed coverage hole detection in wireless sensor networks,” in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, December 2011, pp. 1–6.

[289] J. Kanno, J. Buchart, R. Selmic, and V. Phoha, “Detecting coverage holes in wireless sensor networks,” in *2009 17th Mediterranean Conference on Control and Automation, MED ’09.*, June 2009, pp. 452–457.

[290] A. Hanaf, A. Goupil, M. Colas, and G. Gelle, “Distributed consensus algorithm and its application to detect coverage hole in sensor networks,” in *2015 27th International Conference on Microelectronics (ICM)*, December 2015, pp. 51–54.

[291] S. Fekete, A. Kröller, D. Pfisterer, S. Fischer, and C. Buschmann, “Neighborhood-based topology recognition in sensor networks,” in *Algorithmic Aspects of Wireless Sensor Networks*, ser. Lecture Notes in Computer Science, S. Nikoletseas and J. Rolim, Eds. Springer Berlin Heidelberg, 2004, vol. 3121, pp. 123–136.

[292] P. Corke, R. Peterson, and D. Rus, “Finding holes in sensor networks,” in *the IEEE Workshop on Omniscient Space: Robot Control*, 2007, pp. 1–8.

[293] G. Narasimhan and M. Smid, *Geometric spanner networks*. Cambridge University Press, 2007.

[294] K. K. Chintalapudi and R. Govindan, “Localized edge detection in sensor fields,” *Ad Hoc Networks*, vol. 1, no. 2–3, pp. 273 – 291, 2003.

[295] T. L. Dinh, “Topological boundary detection in wireless sensor networks,” *Journal of Information Processing Systems*, vol. 5, no. 3, pp. 145–150, 2009.

[296] A. M. Khedr, W. Osamy, and D. P. Agrawal, “Perimeter discovery in wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 69, no. 11, pp. 922 – 929, 2009.

[297] G. Fletcher, X. Li, A. Nayak, and I. Stojmenovic, “Randomized robot-assisted relocation of

sensors for coverage repair in wireless sensor networks," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, September 2010, pp. 1–5.

[298] J. McLurkin and E. Demaine, "A distributed boundary detection algorithm for multi-robot systems," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009.*, October 2009, pp. 4791–4798.

[299] Q. Zhang, Y. Sun, and X. Li, "Cluster-based boundary estimation method in sensor networks," in *2007 IET Conference on Wireless, Mobile and Sensor Networks (CCWMSN07)*, Dec 2007, pp. 418–421.

[300] J. Deogun, S. Das, H. Hamza, and S. Goddard, "An algorithm for boundary discovery in wireless sensor networks," in *High Performance Computing – HiPC 2005*, ser. Lecture Notes in Computer Science, D. Bader, M. Parashar, V. Sridhar, and V. Prasanna, Eds. Springer Berlin Heidelberg, 2005, vol. 3769, pp. 343–352.

[301] S. Lee and M. Younis, "Recovery from multiple simultaneous failures in wireless sensor networks using minimum steiner tree," *Journal of Parallel and Distributed Computing*, vol. 70, no. 5, pp. 525 – 536, 2010.

[302] G. Lee and N. Y. Chong, "Self-configurable mobile robot swarms with hole repair capability," in *2008. IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'08*, September 2008, pp. 1403–1408.

[303] W. Zhang, G. Cao, and T. La Porta, "Dynamic proxy tree-based data dissemination schemes for wireless sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 583–595, 2007.

[304] X.-H. Deng, C.-G. Xu, F.-Y. Zhao, and Y. Liu, "Repair policies of coverage holes based dynamic node activation in wireless sensor networks," in *2010 IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC),*, December 2010, pp. 368–371.

[305] G. Wang, G. Cao, and T. LaPorta, "A bidding protocol for deploying mobile sensors," in *11th IEEE International Conference on Network Protocols, 2003. Proceedings.*, November 2003, pp. 315–324.

[306] G. Wang, G. Cao, P. Berman, and T. La Porta, "Bidding protocols for deploying mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 6, no. 5, pp. 563 –576, May 2007.

[307] S. Yang, C.-K. Yeo, and B.-S. Lee, "Toward reliable data delivery for highly dynamic mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 111–124, 2012.

[308] C.-Y. Chang, C.-Y. Chang, Y.-C. Chen, and S.-C. Lee, "Active route-guiding protocols for resisting obstacles in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 9, pp. 4425–4442, November 2010.

[309] A. Abdulla, H. Nishiyama, N. Ansari, and N. Kato, "Energy-aware routing for wireless sensor networks," in *The Art of Wireless Sensor Networks*, ser. Signals and Communication Technology, H. M. Ammari, Ed. Springer Berlin Heidelberg, 2014, pp. 201–234.

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

[311] I. Shin, N. Pham, and H. Choo, "Virtual convex polygon based hole boundary detection and time delay based hole detour scheme in wsns," in *Human Interface and the Management of Information. Designing Information Environments*, ser. Lecture Notes in Computer Science, M. J. Smith and G. Salvendy, Eds. Springer Berlin Heidelberg, 2009, vol. 5617, pp. 619–627.

[312] Z. Guping and W. Yu, "Advance detour strategy for geographic routing in wireless sensor networks," in *International Forum on Information Technology and Applications, 2009.*

IFITA '09., vol. 1, May 2009, pp. 296–299.

[313] W. Scott and N. Leonard, "Dynamics of pursuit and evasion in a heterogeneous herd," in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC),*, December 2014, pp. 2920–2925.

[314] S. Lee, J. Park, T. Chon, and H. Pak, "Prey-flock deformation under a predator's attack," *Journal-Korean Physical Society*, vol. 48, p. S236, 2006.

[315] S.-H. Lee, "Predator's attack-induced phase-like transition in prey flock," *Physics Letters A*, vol. 357, no. 4–5, pp. 270 – 274, 2006.

[316] S.-H. Lee, H. Pak, and T.-S. Chon, "Dynamics of prey-flock escaping behavior in response to predator's attack," *Journal of Theoretical Biology*, vol. 240, no. 2, pp. 250 – 259, 2006.

[317] K. L. Downing, *Intelligence Emerging: Adaptivity and Search in Evolving Neural Systems.* MIT Press, 2015.

[318] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[319] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs.* Springer, 2016.

[320] V. Lesser, C. L. Ortiz Jr, and M. Tambe, *Distributed sensor networks: A multiagent perspective.* Springer Science & Business Media, 2013, vol. 9.

[321] M. Vinyals, J. A. Rodriguez-Aguilar, and J. Cerquides, "A survey on sensor networks from a multiagent perspective," *The Computer Journal*, p. bxq018, 2010.

[322] D. Vernon, *Artificial cognitive systems: A primer.* MIT Press, 2014.

[323] A. Boccardo, R. De Chiara, and V. Scarano, "Massive battle: coordinated movement of autonomous agents," in *Proc. of the Workshop on 3D Advanced Media In Gaming And Simulation (3AMIGAS)*, 2009.

[324] D. Floreano and C. Mattiussi, *Bio-inspired artificial intelligence: theories, methods, and technologies.* MIT press, 2008.

[325] F. Gechter, V. Chevrier, and F. Charpillet, "A reactive agent-based problem-solving model: Application to localization and tracking," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 189–222, December 2006.

[326] S. Yang, F. Gechter, and A. Koukam, "Application of reactive multi-agent system to vehicle collision avoidance," in *2008 20th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '08.*, vol. 1, November, pp. 197–204.

[327] E. Urrea, C. Şafak Şahin, İbrahim Hökelek, M. Ümit Uyar, M. Conner, G. Bertoli, and C. Pizzo, "Bio-inspired topology control for knowledge sharing mobile agents," *Ad Hoc Networks*, vol. 7, no. 4, pp. 677 – 689, 2009.

[328] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.*, May, pp. 1928–1935.

[329] S. Forrest, "Emergent computation: Self-organizing, collective, and cooperative phenomena in natural and artificial computing networks," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1, pp. 1 – 11, 1990.

[330] L. Steels, "Cooperation between distributed agents through self-organisation," in *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, July 1990, pp. 8–14 supl.

[331] Y. U. Cao, A. S. Fukunaga, and A. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Auton. Robots*, vol. 4, no. 1, pp. 7–27, mar 1997.

[332] J. A. Cañizo Rincón, J. Rosado Linares, and J. A. Carrillo de la Plata, "Collective behavior of animals: Swarming and complex patterns," in *Arbor*, vol. 186, 2010, pp. 1035–1049.

[333] J. Carrillo, M. Fornasier, G. Toscani, and F. Vecil, "Particle, kinetic, and hydrodynamic

models of swarming," in *Mathematical Modeling of Collective Behavior in Socio-Economic and Life Sciences*, ser. Modeling and Simulation in Science, Engineering and Technology, G. Naldi, L. Pareschi, and G. Toscani, Eds. Birkhäuser Boston, 2010, pp. 297–336.

[334] M. Rubenstein and W.-M. Shen, "Scalable self-assembly and self-repair in a collective of robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, October 2009, pp. 1484–1489.

[335] D. J. Sumpter, *Collective animal behavior.* Princeton University Press, 2010.

[336] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence.* John Wiley & Sons, 2006.

[337] E. Bonabeau, M. Dorigo, and G. Theraulaz, *From Natural to Artificial Swarm Intelligence.* Oxford University Press, 1999.

[338] G. Beni, "From swarm intelligence to swarm robotics," in *Swarm robotics.* Springer, 2005, pp. 1–9.

[339] M. Dorigo and M. Birattari, "Ant colony optimization," in *Encyclopedia of Machine Learning*, C. Sammut and G. Webb, Eds. Springer US, 2010, pp. 36–39.

[340] D. Karaboga, "Artificial bee colony algorithm," *scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

[341] K. Krishnanand and D. Ghose, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions," *Swarm Intelligence*, vol. 3, no. 2, pp. 87–124, 2009.

[342] K. N. Kaipa and D. Ghose, *Glowworm Swarm Optimization: Theory, Algorithms, and Applications.* Springer, 2017, vol. 698.

[343] W.-H. Liao, Y. Kao, and Y.-S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12 180 – 12 188, 2011.

[344] Y. Tan, *Fireworks Algorithm: A Novel Swarm Intelligence Optimization Method.* Springer, 2015.

[345] S. Zheng, J. Li, A. Janecek, and Y. Tan, "A cooperative framework for fireworks algorithm," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 14, no. 1, pp. 27–41, January 2017.

[346] B. Xing and W.-J. Gao, "Emerging biology-based ci algorithms," in *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, ser. Intelligent Systems Reference Library. Springer International Publishing, 2014, vol. 62, pp. 217–317.

[347] B. N. Clark, C. J. Colbourn, and D. S. Johnson, "Unit disk graphs," *Discrete mathematics*, vol. 86, no. 1-3, pp. 165–177, 1990.

[348] S. Schmid and R. Wattenhofer, "Algorithmic models for sensor networks," in *20th International Parallel and Distributed Processing Symposium, IPDPS 2006*, April 2006.

[349] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile entity localization and tracking in GPS-less environments*, ser. MELT'09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 235–254.

[350] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529 – 2553, 2007.

[351] S. Sivanandam, S. Sumathi, S. Deepa *et al.*, *Introduction to fuzzy logic using MATLAB.* Springer, 2007, vol. 1.

[352] C. Bettstetter, "Mobility modeling in wireless networks: categorization, smooth movement, and border effects," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 3, pp. 55–66, Jul 2001.

[353] H. Ying, "An analytical study on structure, stability and design of general nonlinear takagi-sugeno fuzzy control systems," *Automatica*, vol. 34, no. 12, pp. 1617 – 1623, 1998.

[354] H. Ishibuchi and T. Nakashima, "Effect of rule weights in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems*, vol. 9, no. 4, pp. 506 –515, Aug 2001.

[355] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.*, vol. 3, 1999, pp. 3 vol. (xxxvii+2348).

[356] J. Mendel, R. John, and F. Liu, "Interval type-2 fuzzy logic systems made simple," *Fuzzy Systems, IEEE Transactions on*, vol. 14, no. 6, pp. 808–821, 2006.

[357] X. Bai, S. Li, and J. Xu, "Mobile sensor deployment optimization for k-coverage in wireless sensor networks with a limited mobility model," *IETE Technical Review*, vol. 27, no. 2, p. 124, 2010.

[358] Y. Gao, K. Wu, and F. Li, "Analysis on the redundancy of wireless sensor networks," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, ser. WSNA '03. New York, NY, USA: ACM, 2003, pp. 108–114.

[359] B. Cărbunar, A. Grama, J. Vitek, and O. Cărbunar, "Redundancy and coverage detection in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 94–128, February 2006.

[360] W. W. V. Srinivasan and K.-C. Chua, "Trade-offs between mobility and density for coverage in wireless sensor networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, ser. MobiCom 2007. New York, NY, USA: ACM, 2007, pp. 39–50.

[361] M. B. Jan A Bergstra, *Promise Theory*, 1st ed. XtAxis Press, Oslo, Norway: XtAxis Press, Febraury 2014.

[362] R. Diestel, *Graph Theory, 4th Edition*, 3rd ed., ser. Graduate texts in mathematics. Springer, 2012, vol. 173.

[363] S. M. Hedetniemi, S. T. Hedetniemi, and A. L. Liestman, "A survey of gossiping and broadcasting in communication networks," *Networks*, vol. 18, no. 4, pp. 319–349, 1988.

[364] L. Orecchia, A. Panconesi, C. Petrioli, and A. Vitaletti, "Localized techniques for broadcasting in wireless sensor networks," in *Proceedings of the 2004 joint workshop on Foundations of mobile computing*, ser. DIALM-POMC '04. New York, NY, USA: ACM, 2004, pp. 41–51.

[365] E. S. Lee and C. M. Foust, "The measurement of surge voltages on transmission lines due to lightning," *Journal of the A.I.E.E.*, vol. 46, no. 2, pp. 149–158, February 1927.

[366] M. Huang, S. Chen, and Y. Wang, "Minimum cost localization problem in wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 387 – 399, 2011.

[367] M. Maschler, E. Solan, and S. Zamir, *Game theory. Translated from the Hebrew by Ziv Hellman and edited by Mike Borns.* Cambridge University Press, 2013.

[368] J. Marden, H. Young, G. Arslan, and J. Shamma, "Payoff-based dynamics for multiplayer weakly acyclic games," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 373–396, 2009.

[369] D. Fudenberg and D. K. Levine, *The theory of learning in games*. MIT press, 1998, vol. 2.

[370] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124 – 143, 1996.

[371] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[372] Q. D. L, Y. H. Chew, and B.-H. Soong, *Potential Game Theory: Applications in Radio Resource Allocation*, 1st ed. Springer Publishing Company, Incorporated, 2016.

[373] H. Coxeter, "Inversion in a circle and inversion of lines and circles," *Introduction to Geometry, 2nd ed. John Wiley and Sons, New York*, pp. 77–83, 1969.

[374] E. W. Weisstein, "Inversion-from mathworld–a wolfram web resource."

[375] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statis-*

tics). Springer-Verlag New York, Inc., 2006.

[376] J. Mount, "The equivalence of logistic regression and maximum entropy models," *www.win-vector.com/dfiles/LogisticRegressionMaxEnt.pdf*, 2011.

[377] L. Trevisan, "Cs359g: Graph partitioning and expanders, lecture 1," Course at Stanford University, http://www.eecs.berkeley.edu/ luca/cs359g/index.html, January 2011.

[378] T. Nakano, A. W. Eckford, and T. Haraguchi, *Molecular communication*. Cambridge University Press, 2013.

[379] B. Atakan, *Molecular Communications and Nanonetworks*. Springer, 2014.

[380] S. F. Bush, *Nanoscale Communication Networks*. Artech House, 2010.

[381] M. Kim and E. Steager, *Microbiorobotics: biologically inspired microscale robotic systems*. William Andrew, 2012.

[382] J. Schiller, *Nanorobots and Microrobots Exciting Tools of Future*. CreateSpace Independent Publishing Platform, 2013.

[383] Y. Chen, T. Nakano, P. Kosmas, C. Yuen, A. V. Vasilakos, and M. Asvial, "Green touchable nanorobotic sensor networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 136–142, November 2016.

[384] R. R. Murphy, *Disaster robotics*. MIT Press, 2014.

[385] Y. Taniguchi, T. Kitani, and K. Leibnitz, "An airdrop deployment method for sensor nodes with coordinated gliding and falling," *Proceedings of Workshop on Sensor Networks for Earth and Space Science Applications (ESSA 2009)*, pp. 41–48, 2009.

[386] ——, "A uniform airdrop deployment method for large-scale wireless sensor networks," *International Journal of Sensor Networks*, vol. 9, no. 3-4, pp. 182–191, 2011.

[387] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized control of unmanned aerial vehicle collaborative sensing missions," in *2007 American Control Conference*, July 2007, pp. 4672–4677.

[388] I. Guvenc, W. Saad, M. Bennis, C. Wietfeld, M. Ding, and L. Pike, "Wireless communications, networking, and positioning with unmanned aerial vehicles [guest editorial]," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 24–25, May 2016.

[389] E. P. de Freitas, T. Heimfarth, I. F. Netto, C. E. Lino, C. E. Pereira, A. M. Ferreira, F. R. Wagner, and T. Larsson, "Uav relay network to support wsn connectivity," in *International Congress on Ultra Modern Telecommunications and Control Systems*, October 2010, pp. 309–314.

[390] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *2004 IEEE International Conference on Robotics and Automation, Proceedings. ICRA '04.*, vol. 4, April 2004, pp. 3602–3608 Vol.4.

[391] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, May 2016.

[392] P. Zhan, K. Yu, and A. L. Swindlehurst, "Wireless relay communications with unmanned aerial vehicles: Performance and optimization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 3, pp. 2068–2085, July 2011.

[393] J. Yuh, "Design and control of autonomous underwater robots: A survey," *Autonomous Robots*, vol. 8, no. 1, pp. 7–24, 2000.

[394] S. Wadoo and P. Kachroo, *Autonomous underwater vehicles: modeling, control design and simulation*. CRC Press, 2011.

[395] M. Garzón, J. Valente, D. Zapata, and A. Barrientos, "An aerial-ground robotic system for navigation and obstacle mapping in large outdoor areas," *Sensors*, vol. 13, no. 1, pp. 1247–1267, 2013.

[396] M. A. Hsieh, A. Cowley, J. F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C. J. Taylor, Y. Endo, R. C. Arkin, B. Jung, D. F. Wolf, G. S. Sukhatme, and D. C. MacKenzie, "Adaptive teams of autonomous aerial and ground robots for situational awareness," *Journal of Field Robotics*, vol. 24, no. 11-12, pp. 991–1014, 2007.

[397] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to autonomous mobile robots*. MIT press, 2011.

[398] Y. Gao, *Contemporary Planetary Robotics: An Approach Toward Autonomous Systems*. John Wiley & Sons, 2016.

[399] P. B. Sujit, J. Sousa, and F. L. Pereira, "Uav and auvs coordination for ocean exploration," in *OCEANS 2009 - EUROPE*, May 2009, pp. 1–7.

[400] M. Faria, J. Pinto, F. Py, J. Fortuna, H. Dias, R. Martins, F. Leira, T. A. Johansen, J. Sousa, and K. Rajan, "Coordinating uavs and auvs for oceanographic field experiments: Challenges and lessons learned," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 6606–6611.

[401] K. Li, K. Thomas, L. Rossi, and C.-C. Shen, "Slime mold inspired protocol for wireless sensor networks," in *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO '08.*, October, pp. 319–328.

[402] R. Machado and S. Tekinay, "A survey of game-theoretic approaches in wireless sensor networks," *Computer Networks*, vol. 52, no. 16, pp. 3047 – 3061, 2008.

[403] X. Fan, Z. Zhang, X. Lin, and H. Wang, "Coverage hole elimination based on sensor intelligent redeployment in wsn," in *2014 IEEE 4th Annual International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER),*, June 2014, pp. 336–339.

[404] S. Choudhury, K. Salomaa, and S. G. Akl, "Cellular automaton-based algorithms for the dispersion of mobile wireless sensor networks," *International Journal of Parallel, Emergent and Distributed Systems*, vol. 29, no. 2, pp. 147–177, 2014.

[405] S. Choudhury, S. G. Akl, and K. Salomaa, "Energy efficient cellular automaton based algorithms for mobile wireless sensor networks," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2012, pp. 2341–2346.

[406] S. Choudhury, *Cellular Automata and Wireless Sensor Networks*. Cham: Springer International Publishing, 2017, pp. 321–335.

[407] Y. K. Tan, *Energy harvesting autonomous sensor systems: design, analysis, and practical implementation*. CRC Press, 2013.

[408] M. T. Penella-López and M. Gasulla-Forner, *Powering autonomous sensors: an integral approach with focus on solar and RF energy harvesting*. Springer Science & Business Media, 2011.

[409] Z. A. Eu, H.-P. Tan, and W. K. Seah, "Opportunistic routing in wireless sensor networks powered by ambient energy harvesting," *Computer Networks*, vol. 54, no. 17, pp. 2943 – 2966, 2010.

[410] M. Stojcev, M. Kosanovic, and L. Golubovic, "Power management and energy harvesting techniques for wireless sensor nodes," in *2009 9th International Conference on Telecommunication in Modern Satellite, Cable, and Broadcasting Services, TELSIKS '09.*, October 2009, pp. 65 –72.

[411] L. Shu, M. Mukherjee, and X. Wu, "Toxic gas boundary area detection in large-scale petrochemical plants with industrial wireless sensor networks," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 22–28, October 2016.

[412] L. H. Encinas, S. H. White, A. M. del Rey, and G. R. Sánchez, "Modelling forest fire spread using hexagonal cellular automata," *Applied Mathematical Modelling*, vol. 31, no. 6, pp. 1213 – 1227, 2007.

[413] K. Wehrle, M. Gnes, and J. Gross, *Modeling and Tools for Network Simulation*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[414] A.-S. K. Pathan, M. M. Monowar, and S. Khan, *Simulation Technologies in Networking and Communications: Selecting the Best Tool for the Test.* CRC Press, 2014.

[415] H. Shu, Q. Liang, and J. Gao, "Distributed sensor networks deployment using fuzzy logic systems," *International Journal of Wireless Information Networks*, vol. 14, pp. 163–173, 2007.

[416] I. Gupta, D. Riordan, and S. Sampalli, "Cluster-head election using fuzzy logic for wireless sensor networks," in *3rd Annual Communication Networks and Services Research Conference (CNSR'05)*, May 2005, pp. 255–260.

[417] D. Izadi, J. Abawajy, and S. Ghanavati, "An alternative node deployment scheme for wsns," *Sensors Journal, IEEE*, vol. 15, no. 2, pp. 667–675, February 2015.

[418] P. Nayak and A. Devulapalli, "A fuzzy logic-based clustering algorithm for wsn to extend the network lifetime," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 137–144, January 2016.

[419] J. M. Mendel, *Uncertain rule-based fuzzy logic systems: introduction and new directions.* Prentice Hall PTR Upper Saddle River, 2001.

[420] X. Xiao, D. Hu, D. Guo, and K. Liu, "If: isolating fault nodes with mobile scanner," *International Journal of Communication Systems*, vol. 27, no. 4, pp. 592–605, 2014.

[421] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, November 2005.

[422] M. Burgess, "An approach to understanding policy based on autonomy and voluntary cooperation," in *Ambient Networks*, ser. Lecture Notes in Computer Science, J. Schönwälder and J. Serrat, Eds. Springer Berlin Heidelberg, 2005, vol. 3775, pp. 97–108.

[423] M. Buckland, *Programming game AI by example.* Jones & Bartlett Learning, 2004.

[424] I. Millington and J. Funge, *Artificial intelligence for games.* CRC Press, 2009.

[425] G. Grimmett, "What is percolation?" in *Percolation*, ser. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 1999, vol. 321, pp. 1–31.

[426] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration for energy conservation in sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 1, pp. 36–72, August 2005.

APPENDIX A

Boundary Node Selection Algorithms

A.1 Introduction

Catastrophic events can result in large holes in the coverage of WSNs. Repair of such damage requires accurate identification of the proper subset of nodes that form the boundaries of damaged regions (B-nodes). Some of these nodes, defined as *outer boundary nodes*, may be directed to reduce the size of damaged regions by either increasing nodes' sensing/transmission ranges and/or physically moving nodes towards the damaged regions. In order to address the burden of centralised decision making in WSNs, decisions of participating in the hole recovery process can autonomously be made by nodes based on their limited and local information about their surroundings and neighbouring nodes. In the following section a number of node selection algorithms are proposed based on which nodes determine whether or not they belong to a set of outer boundary nodes. The efficacy of each algorithm is examined by using a number of performance metrics.

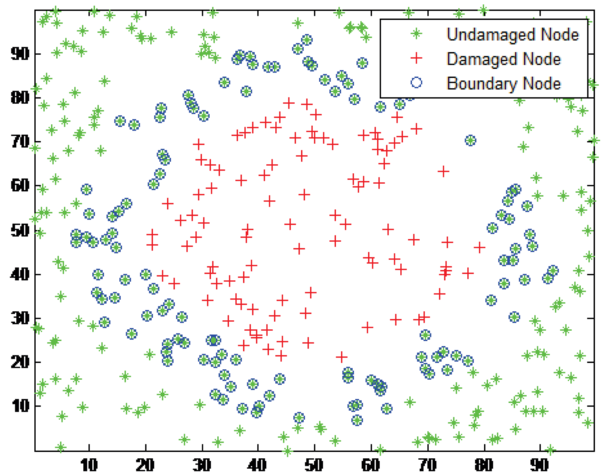


Figure A.1: Coverage Hole

A.2 Method and Assumptions

A.2.1 Sensor and Network Hole Models

Similar to Section 3.2.1, sensor nodes, damaged area (coverage hole) and network's area of deployment are modelled.

A.2.2 Node classification

Similar to Section 3.2.3, after a damage event (D-event), based on the nodes' distances to the region of interest (ROI) (i.e. the coverage hole), they are classified mainly as *undamaged nodes* (U-nodes) and *damaged nodes* (D-nodes) (Figure A.1). D-nodes can also be called *virtual nodes* (V-nodes) as the information about their locations and degrees is used by their neighbouring B-nodes to enhance the precision of hole detection and selection algorithms.

U-nodes that detect the D-event within their ranges are considered as *boundary nodes* (B-nodes); otherwise, they are known as *normal nodes* (N-node). B-nodes can form a *margin of B-nodes* (MB-nodes) around the damaged area as in Figure A.1. In this chapter, B-nodes in the margin that are closer to the D-event and are selected to participate in recovery process (SB-nodes as in Section (3.2.4.2)) are defined as *outer boundary nodes* (OB-nodes).

A.2.3 Visibility, information and model of decision

Considering its status, each B-node autonomously decides whether it is an OB-node or not. B-nodes use their available 1-hop information (degrees and locations of their neighbour nodes) in order to make decision. They are aware of their damaged neighbouring nodes due to signal loss. After the D-event, B-nodes are not able to determine whether their undamaged neighbours nodes (UDN-node) are B-nodes or N-nodes.

A.2.4 Outer Boundary Estimation/Selection Algorithms

The goal of node selection is to efficiently increase the number of B-nodes engaged in the recovery mechanism in order to alleviate problems such as the redundant coverage of boundary areas within B-nodes' ranges and the interference around D-areas. The proper selection can be achievable via simple criteria and decisions of autonomous B-nodes. Based on its available local knowledge of neighbours before and after the D-event, each B-node can autonomously decide to consider itself an OB-node. Similar to Section 3.2.4, in this chapter, different boundary node selection algorithms (BNS-algorithms) based on the statistical and geometrical feature of B-nodes are introduced. BNS-algorithms (Algorithms A.1 and A.2) are listed below:

ALGORITHM A.1: Boundary Selection Algorithms

Input:
u_k: Undamaged Node (U-Node) k
d_j: Damaged Node (D-Node) j
b_i: Boundary Node (B-Node) i
N_b: Number of B-nodes, $b_i, i \in \{1, \cdots N_b\}$
D_{b_i}: Degree of B-node $b_i, i \in \{1, \cdots N_b\}$
$N_{b_i^d}$: Number of B-node b_i's DN-nodes
$N_{b_i^u}$: Number of B-node b_i's UN-nodes
$\vec{X}_{b_i}^{d_j}$: Distance vector from B-node b_i to its DN-Node d_j
$\vec{X}_{b_i}^{u_k}$: Distance vector from B-node b_i to its UN-Node u_k
$D_{b_i}^{d_j}$: Degree of DN-Node d_j of B-node b_i
$D_{b_i}^{u_k}$: Degree of UN-Node u_k of B-node b_i
$CosineLaw(a, b)$: $(a^2 + b^2 - 2.a.b.cos(atan(a/b)))^{1/2}$
α: Threshold

Output:
Selected (Outer) B-nodes (OB-node) $b_{s(i)}$

Algorithm *CLMinB Outer Boundary Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{min}} = min(||\vec{X}_{b_i}^{d_j}||), \quad \forall N_{b_i^d}$
  - Calculate $S_{b_i}^{(\mathbf{p}_{min}, u_k)} = CosineLaw\Big(||\vec{X}_{b_i}^{d_{min}}||, ||\vec{X}_{b_i}^{u_k}||\Big), \quad \forall N_{b_i^u}$
  - **if** $min(S_{b_i}^{(\mathbf{p}_{min}, u_k)}) > ||\vec{X}_{b_i}^{d_{min}}||$ **then**
    - b_i is OB-node
  - **else**
    - u_k is OB-node
  - **end**
- **end**

End

Algorithm *CLAvgB Outer Boundary Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{Avg}} = \sum_{j \in N_{b_i^d}} \left(\frac{\vec{X}_{b_i}^{d_j}}{N_{b_i^d}} \right)$
  - Calculate $S_{b_i}^{(\mathbf{p}_{Avg}, u_k)} = CosineLaw\Big(||\vec{X}_{b_i}^{d_{Avg}}||, ||\vec{X}_{b_i}^{u_k}||\Big), \quad \forall N_{b_i^u}$
  - **if** $min(S_{b_i}^{(\mathbf{p}_{Avg}, u_k)}) > ||\vec{X}_{b_i}^{d_{min}}||$ **then**
    - b_i is OB-node
  - **else**
    - u_k is OB-node
  - **end**
- **end**

End

ALGORITHM A.2: Boundary Node Selection Algorithms (Alg. A.1 Contd.)

Input:
For the Input Parameter please refer to Algorithm (A.1)

Output:
Selected (outer) B-nodes (OB-node) $b_{s(i)}$

Algorithm *CLwb Outer Boundary Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{CM}} = \dfrac{\sum^{j \in N_{b_i^d}} \left(\vec{X}_{b_i}^{d_j} . D_{b_i}^{d_j} \right)}{\sum_{j \in N_{b_i^d}} \left(D_{b_i}^{d_j} \right)}$
  - Calculate $S_{b_i}^{(\mathbf{p}_{CM}, u_k)} = CosineLaw\Big(||\vec{X}_{b_i}^{d_{CM}}||, ||\vec{X}_{b_i}^{u_k}|| \Big), \quad \forall N_{b_i^u}$
  - **if** $min(S_{b_i}^{(\mathbf{p}_{CM}, u_k)}) > ||\vec{X}_{b_i}^{d_{min}}||$ **then**
    - b_i is OB-node
  - **else**
    - u_k is OB-node
  - **end**
- **end**

End

Algorithm *QB Outer Boundary Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{Avg}} = \sum^{j \in N_{b_i^d}} \left(\dfrac{\vec{X}_{b_i}^{d_j}}{N_{b_i^d}} \right)$
  - Calculate $S_{b_i}^{(\mathbf{p}_{\tau}, u_k)} = Quantile\Big(\vec{X}_{b_i}^{d_j}, \alpha \Big), \quad \forall N_{b_i^d}$
  - **if** $S_{b_i}^{(\mathbf{p}_{\tau}, u_k)} > ||\vec{X}_{b_i}^{d_{Avg}}||$ **then**
    - b_i is OB-node
  - **else**
    - u_k is OB-node
  - **end**
- **end**

End

Algorithm *WQB Outer Boundary Selection*:
- **foreach** $b_i \in N_b$ **do**
  - Calculate $\vec{X}_{b_i}^{d_{Wgt}} = \dfrac{\sum^{j \in N_{b_i^d}} \left(\vec{X}_{b_i}^{d_j} . D_{b_i}^{d_j} \right)}{\sum_{j \in N_{b_i^d}} \left(D_{b_i}^{d_j} \right)}$
  - Calculate $S_{b_i}^{(\mathbf{p}_{\tau}, u_k)} = Quantile\Big(\vec{X}_{b_i}^{d_j}, \alpha \Big), \quad \forall N_{b_i^d}$
  - **if** $S_{b_i}^{(\mathbf{p}_{\tau}, u_k)} > ||\vec{X}_{b_i}^{d_{Wgt}}||$ **then**
    - b_i is OB-node
  - **else**
    - u_k is OB-node
  - **end**
- **end**

End

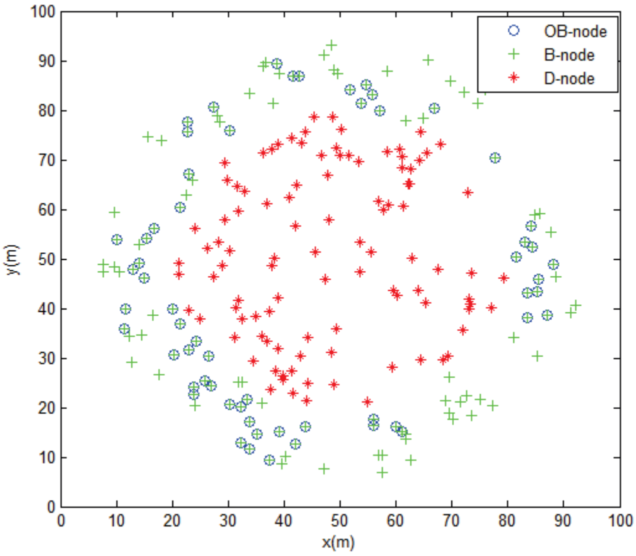


Figure A.2: QB (0.5) OB-Node Selection Algorithm applied on B-nodes

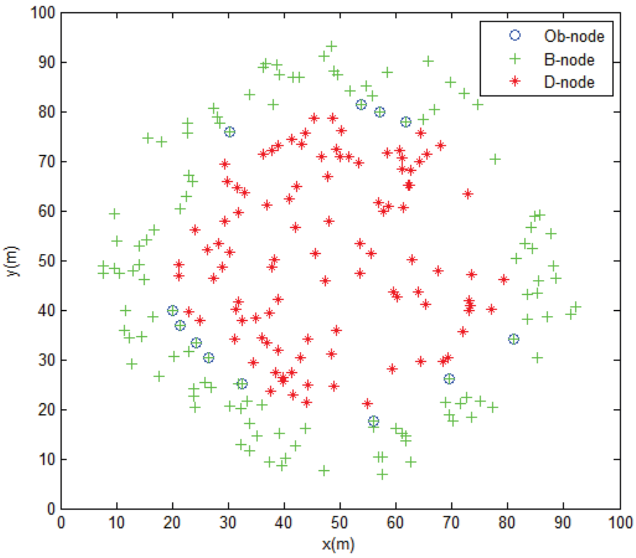


Figure A.3: Cosine Law Boundary Min Distance, Selection Algorithm

- ***Margin of Boundary Nodes (MBN)***. Each B-node detects the damage and considers itself as an OB-node. So, a band of boundary nodes forms around the D-area.
- ***Random Boundary Node (RBN)***. Each B-node randomly considers itself as an OB-node with a given probability of p (here $p = 0.5$). This is a simple network node sampling applied only to MB-nodes instead of all the nodes in network.
- ***Quantile-Based Boundary (QB)***. Each B-node compares its α-quantile (here median) and average distances to all its D-node neighbours (DN-nodes) in order to decide on its status. One reason for using quantile is to show the central tendency of distances of B-nodes to their DN-nodes, which cannot be shown by their simple averages (Figure A.2).
- ***Weighted Quantile-Based Boundary (WQB)***. Similar to QB, WQB algorithm considers the DN-nodes' degree of B-nodes as a weight. The degrees of D-nodes are considered in order to account for the importance and effect of DN-nodes of a given B-node in the decision-making process.
- ***Cosine Law Min Boundary (CLMinB)***. Each B-node estimates the distances of its U-node neighbours (UN-nodes) to its closest DN-node by using the cosine law. Then, if the B-node is closer to its closest DN-node than all of its UN-nodes, it considers itself an OB-node (Figure A.3).
- ***Cosine Law Average Boundary (CLAvgB)***. Each B-node tries to obtain the average distance vector to its DN-nodes. Then, similar to the CLMinB algorithm, it checks to see if one of its UN-nodes is closer to DN-node average point (vector's tail) or not. If not, the given B-node considers itself an OB-node.
- ***Cosine Law weighted Boundary (CLwB)***. Each B-node tries to obtain

the centre of mass vector to its DN-nodes (degree as the mass). Then, similar to CLMinB algorithm, it checks to see if one of its UN-nodes is closer to the DN-nodes' centre of mass (vector's tail) or not. If not, the given B-node considers itself an OB-node.

Figures A.2 and A.3 respectively depict the selected boundary nodes by QB(0.5) and CLMinB) selection algorithms.

A.3 Performance Evaluation

A.3.1 Performance Metrics

Average distance from the damaged area : The average distance to D-area (AVD) for OB-nodes and a given coverage hole modelled in section 3.2.2, is defined according to Section 3.3.1. Since B-nodes closer to ROI would be able to sense and collect more precise information, it is more desirable to select B-nodes closer to the D-area as as OB-nodes. One reasonable way to show that these OB-nodes are rather distributed evenly around the damaged area would be the histogram of angles of OB-nodes at the centre of coverage hole,

$$\theta_{OB_i}^{Hole} = tan^{-1}\left(\frac{y_{OB_i} - y_{hole}}{x_{OB_i} - x_{hole}}\right) \tag{A.1}$$

Therefore, if the distribution of OB-nodes' angles are rather uniform in the histogram, it shows that the selected nodes are evenly selected around the damaged area. As an example, the angle distribution of OB-nodes in radian with range of $(-\pi, \pi]$ for 3 BNS-algorithms is shown in Figure A.4.

k-redundant Sensing Coverage (k-RSC) : In their proposed *coverage configuration protocol* (CCP), Xing et al. [426] defined *degree of sensing coverage* and *average degree of sensing coverage* as a performance metric to activate a proper

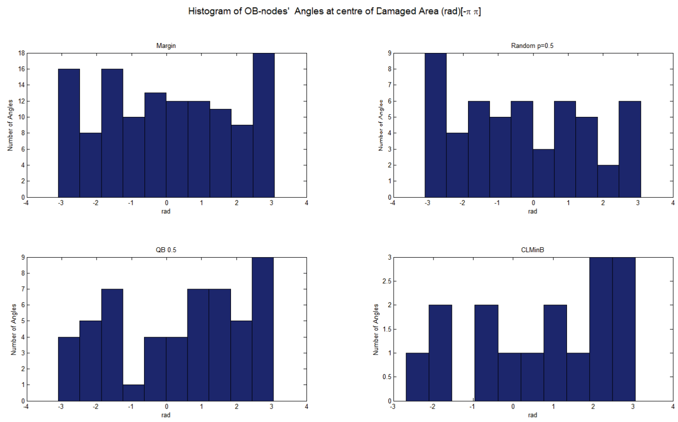


Figure A.4: 5 Distribution of Angles (histogram) OB-nodes at (x_{hole}, y_{hole})

number of nodes and reduce redundant sensing coverages. By slightly modifying their metrics, percentages of k-redundant sensing coverage as (Perc-k-RSC) can be defined as,

$$Perc_{k-RSC} = \left[\frac{Number - Covered\ Cells\ \{n = k+1, \ldots, N_{node}\}}{Total\ Number\ Cells\ in\ the\ area} \right] \times 100, \quad \text{(A.2)}$$

Where N_{node} is number of sensor nodes. Since before and after damage event in this model, other parts of the network's coverage do not change, only the rectangular area of $[x_{hole} - r_{hole}, x_{hole} + r_{hole}]$, $[y_{hole} - r_{hole}, y_{hole} + r_{hole}]$ is considered here and the given area is divided into grid cells. Similar to Section 3.3.1.2, the coverage of grid cells is measured by the number of nodes covering the cells. By this metric, it can be seen how coverage redundancies change as the number of selected nodes are reduced in BNS-algorithms (Figures A.6 and A.5). For example, Perc-1-RSC, shows the percentage of a given area that is covered by at least more than two nodes.

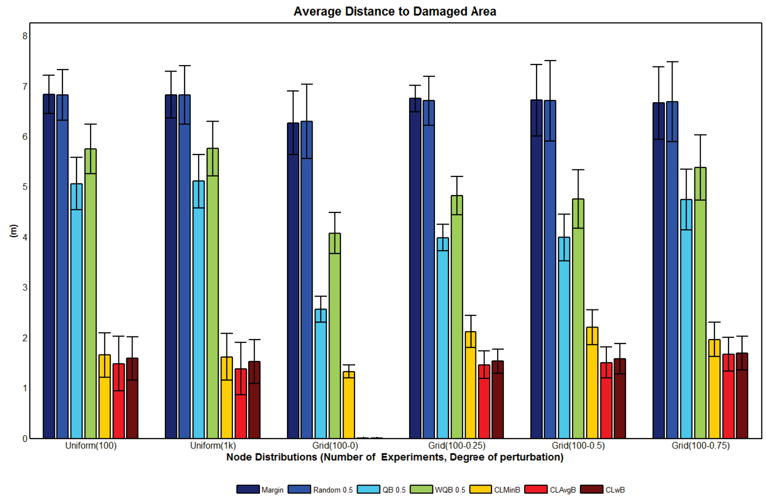


Figure A.5: Average distances from damage area of BNS-algorithms

A.3.2 Results

Similar to Section 3.3.2, $N_{node} = 400$ nodes within a area of $[0\ 100]^2\ m^2$ were deployed in a random uniform distribution, grid and randomly perturbed grid models (random perturbations of 0.25, 0.5, and 0.75). The experiment was done 1000 and 100 times for random uniform distribution and 100 times for grid and perturbed grid distributions in the area of deployment. Nodes had the transmission and sensing radius ranges of $R_c = R_s = 15\ m$. The damaged area was modelled with the circle of radius $r_{hole} = 30\ m$ centred at (x_{hole}, y_{hole})=$(50, 50)$.

The sample results of BNS-algorithms are shown in Figures A.2 and A.3. Performance results are shown with error bars (Figures A.5, A.6 and A.7) that contain the value and their standard deviation (SD). As shown in Figure A.5 that the AVD of OB-nodes in random uniform distribution reduced from $6.82 - 6.83\ m$ to $5 - 5.7\ m$ and $1.4 - 1.6\ m$ when QB and CL algorithms were applied respectively. The number of selected B-nodes and Perc-k-RSC of the roposed BNS-algorithms

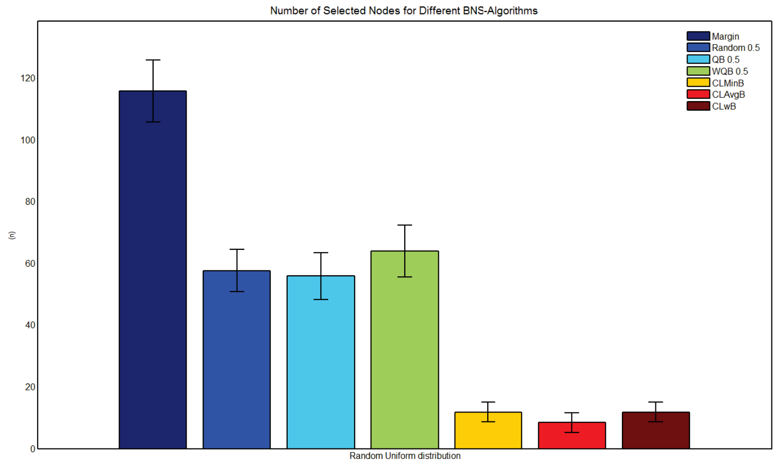


Figure A.6: Number of Selected B-nodes of BNS-algorithms

are shown in Figures A.6 and A.7, respectively.

Perc-k-RSC for $k = 1,\ 2,\ 3$ presents the percentage of redundant sensing coverage when grid cells are covered by at least $2,\ 3,\ 4$ nodes respectively. It can be seen that redundancy in coverage was reduced faster in CLMinB, CLwB, CLAvgB than with the other BNS-algorithms. Thus, these algorithms are useful for the applications where 1 and 2-coverage is required to be maintained around the damaged area and at the same time the lifetime and energy of the B-nodes are important. As Figure A.6 shows, the number of B-nodes with CLMinB, CLavgB, and CLwB BNS-algorithms were reduced 10 times while they are halved in the others. Those algorithms that halved the number of selected B-nodes are suitable for applications with a moderate degree of sensing coverages. Therefore, as shown in Figures A.6 and A.7, the number of selected nodes reduced to half and the required degree of sensing coverage were kept to some acceptable levels. Therefore, depending on the application, different BNS-algorithms can be applied to MB-nodes. Hence, BNS-algorithms can reduce interference and redundancy in coverage/connectivity

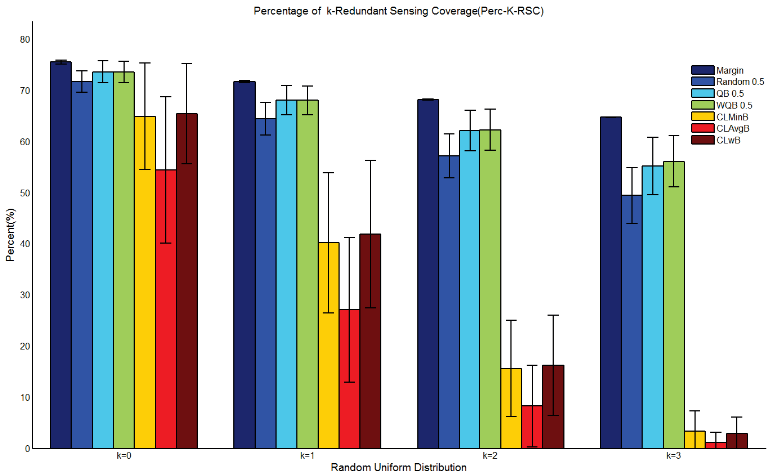


Figure A.7: Perc-k-RSC of BNS-algorithms

and increase the lifetime of margin of boundary area (MB-area), which in turn may avoid or delay further failure expansion in MB-area around the coverage hole.

A.4 Conclusion

In this chapter, different BNS-algorithms based on the statistical and geometrical feature of B-nodes were introduced. The proposed BNS-algorithms used B-nodes' available 1-hop information and can be considered as one of solutions for reducing the number of participating nodes in the recovery process and maintaining an acceptable level of network performance, such as coverage around the damaged area. This is because, the redundancy of participating B-nodes in the potential recovery process of damaged area are expected to cause interference and collision/disconnection when such B-nodes relocate to damaged area. For the proposed BNS-algorithms, a trade-off between the required number of B-nodes and network quality of service, such as coverage, can be expected.

APPENDIX B

Performance of DSSA for Large Scale CH

B.1 Introduction

Large scale coverage holes disturb normal operation of networks and may severely affect their integrity. As a feasible and economical topology control scheme, distributed node relocation can be considered as an effective solution to repair node failures in WSNs especially where sensor nodes are deployed in harsh and isolated environments without central supervision. Devised relocation algorithms may effectively meet their design goals; however, they may not be responsive to unpredicted events such as networks' coverage holes. In this chapter, one of such case is demonstrated. DSSA seems to be suitable for balancing the deployed nodes and for repairing small coverage holes. However, as it is shown, DSSA is not able to fully repair large scale coverage holes, even if all the nodes participate in the recovery process and relocate with sufficient number of iterations.

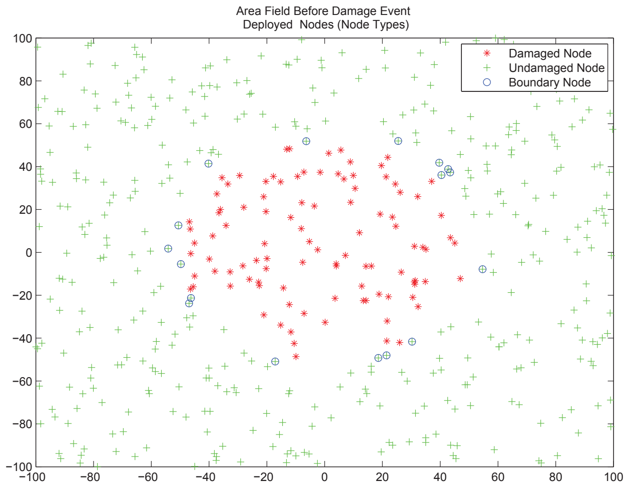
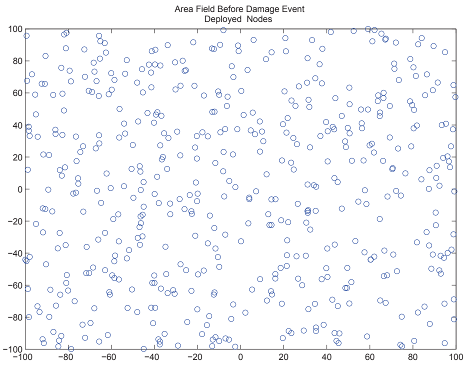


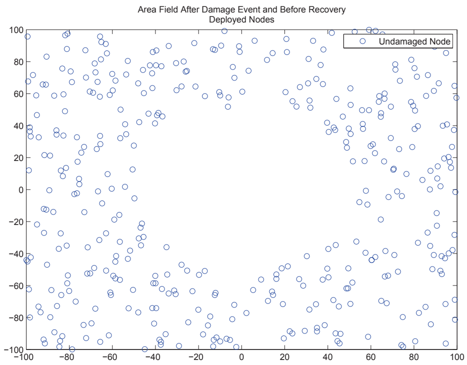
Figure B.1: Network Hole and Node Types

B.2 Method and Assumptions

In our scenario, the deployed sensor nodes, area of deployment and coverage hole were modelled according to Section 3.2.1.1. Similar to Section 3.2.3, nodes are classified as *damaged* and *undamaged* nodes (D-nodes and U-nodes) with respect to their locations and the damaged area (coverage holes). U-nodes that detect the *damage event* within their ranges are further defined as *boundary nodes* (B-nodes) as in Section 3.2.3 (Figure B.1). *Distributed Self-Spreading Algorithm (DSSA)* [227], as one of force-based relocation algorithms with the promising performance, was applied to either B-nodes or all U-nodes. Their performances in term of coverage and the ability to repair the damaged area is depicted in Figures B.2 and B.3. Nodes stop at the boundaries of area of deployment if they reach them (Section 5.2.2).

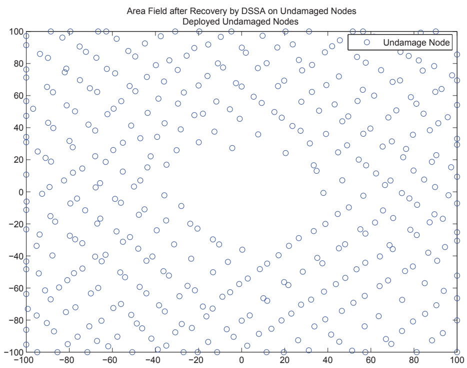


(a) Initial Network Deployment



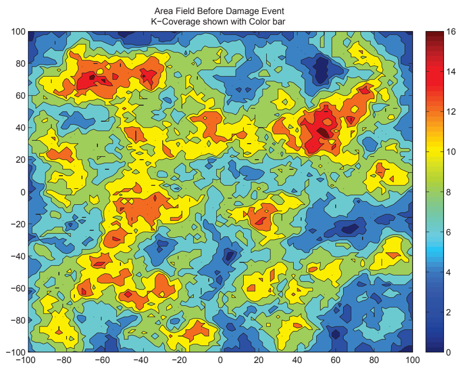
(b) Network Deployment After Coverage Hole/Before Recovery

(c) Network Deployment After Recovery using Boundary Nodes

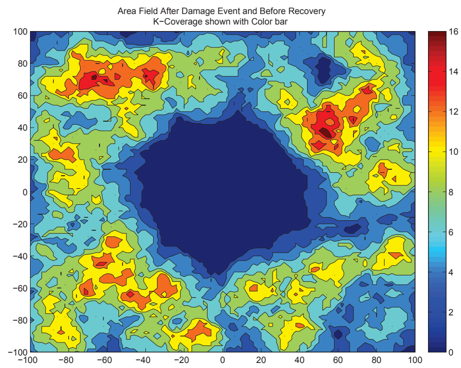


(d) Network Deployment After Recovery using Undamaged Nodes

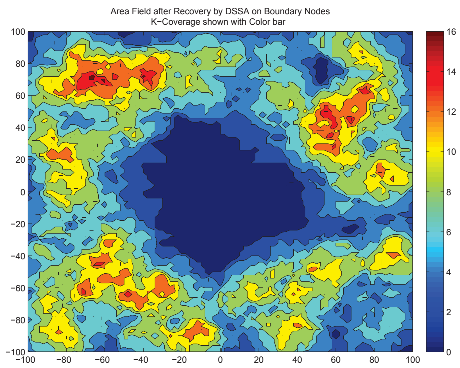
Figure B.2: Network Deployment Stages in the Recovery, Performed by DSSA on Boundary/Undamaged Nodes



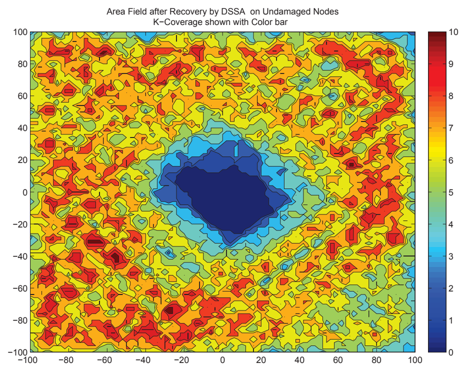
(a) Initial Network Coverage



(b) Network Coverage After Coverage Hole/Before Recovery



(c) Network Coverage After Recovery using Boundary Nodes



(d) Network Coverage After Recovery using Undamaged Nodes

Figure B.3: Network Coverage Stages in the Recovery, Performed by DSSA on Boundary/Undamaged Nodes

B.3 Performance Evaluation

B.3.1 Results

N=500 nodes with transmission and sensing range of R_c=R_s=15 m deployed in the rectangular 2-D area of $[-100, 100] \times [-100, 100]$ m^2. By considering the boundary condition used in Section 5.2.2, DSSA was applied to B-nodes and U-nodes in the area of deployment. Figures B.2 and B.3 show network deployment and coverage status before and after occurrence/recovery of a coverage hole as DSSA algorithm applied to B- and U-nodes for 50 iterations. It should be noted that the most recovery resulted form the first 10 iterations. As the number of iterations increased, the DSSA performance did not improve accordingly. Therefore according to Figures B.2 and B.3, although DSSA is an efficient relocation algorithm for aligning unbalanced deployments or repairing small coverage holes, it is a not proper choice for recovering large scale coverage holes.

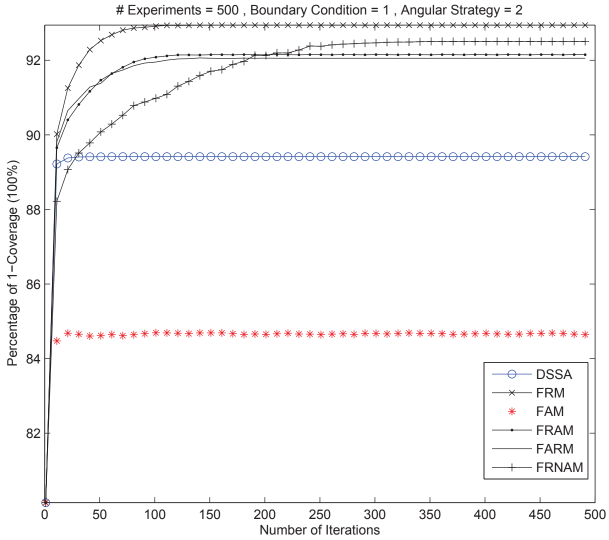
B.4 Conclusion

In this chapter, as a force-based relocation algorithms, DSSA was chosen for the recovery of a large coverage hole. DSSA recovery capability and efficiency in term of coverage was demonstrated as it was applied to either boundary or all undamaged nodes. As shown, DSSA is not a proper choice for recovery of a large scale coverage holes even if all of the network's nodes participate in the recovery process and/or the number of relocation iterations are increased sufficiently.

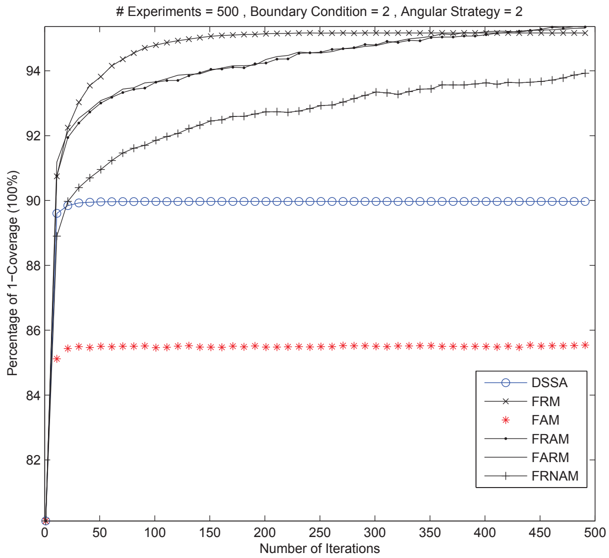
APPENDIX C

Fuzzy Logic Movement Model Figures

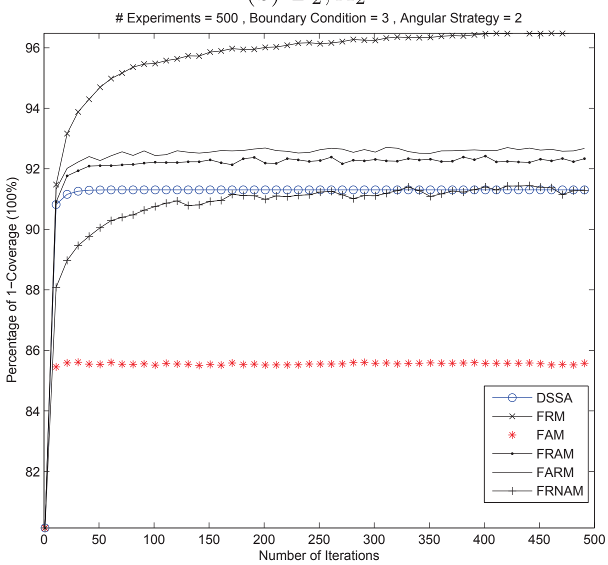
In this section result for angular force strategies A_2 are presented in Figures C.1, C.2, and C.3 for the fuzzy logic node relocation model based on the expert knowledge.



(a) B_1, A_2

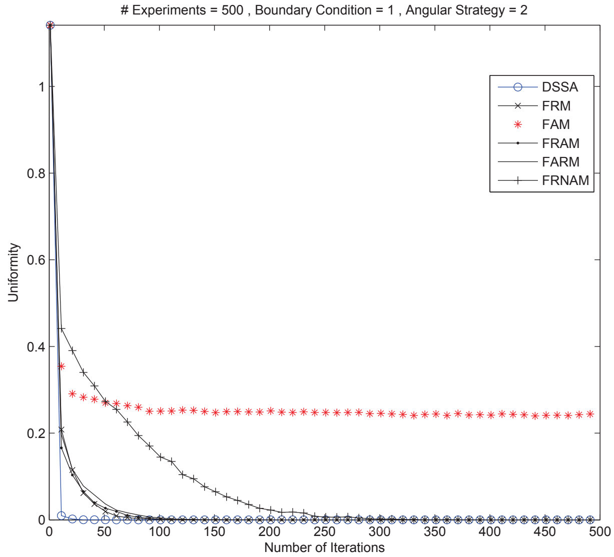


(b) B_2, A_2

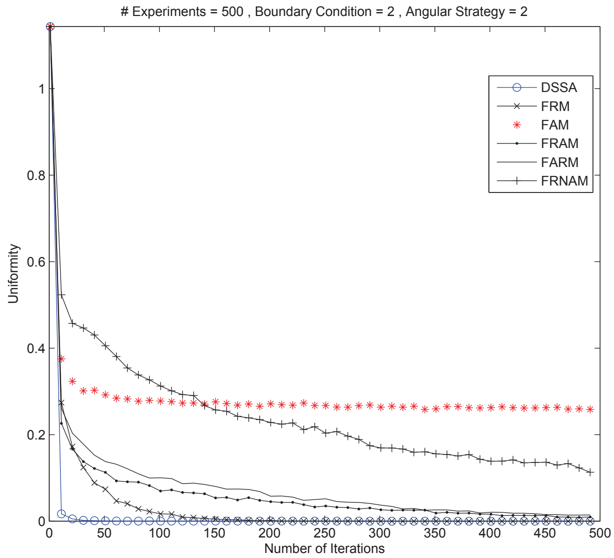


(c) B_3, A_2

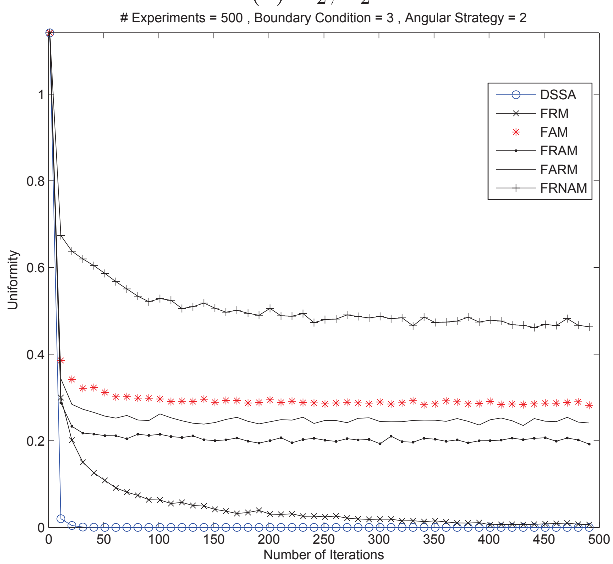
Figure C.1: Percentage of 1-Coverage (100%) for different boundary conditions with angular force strategy A_2



(a) B_1, A_2

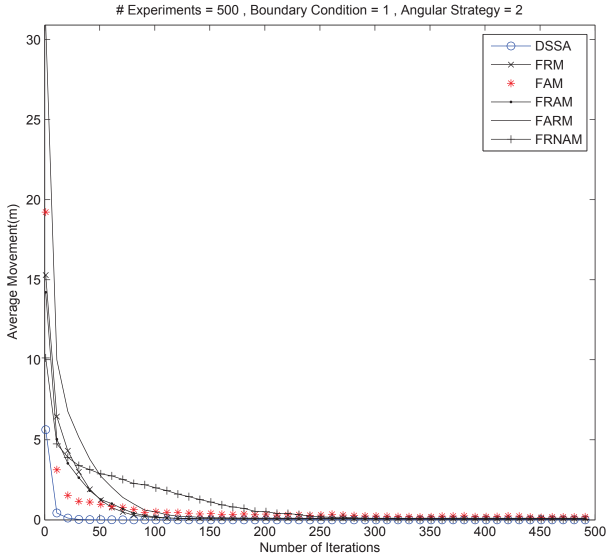


(b) B_2, A_2

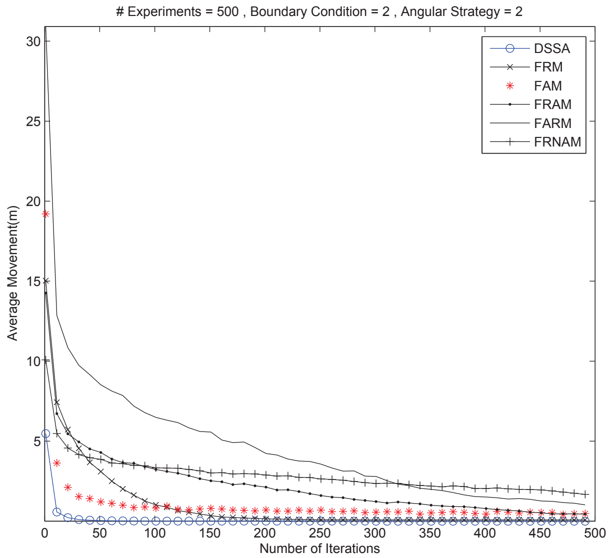


(c) B_3, A_2

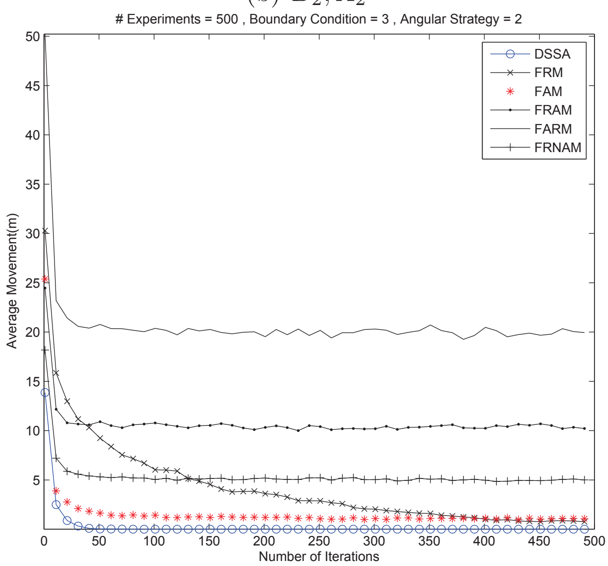
Figure C.2: Uniformity for different boundary conditions with angular force strategy A_2



(a) B_1, A_2



(b) B_2, A_2



(c) B_3, A_2

Figure C.3: Average movement for different boundary conditions with angular force strategy A_2