

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2013-ENST-0049

EDITE - ED 130

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

TELECOM ParisTech

Spécialité “Informatique et Réseaux”

présentée et soutenue publiquement par

Feng YAN

le 18 Septembre 2013

Homologie Simpliciale et Couverture Radio dans un Réseau de Capteurs

Directeur de thèse: **Philippe MARTINS**

Jury

M. Philippe GODLEWSKI, Professeur, TELECOM ParisTech, France	Président
M. Martin HAENGGI, Professeur, University of Notre Dame, USA	Rapporteur
M. Xavier LAGRANGE, Professeur, TELECOM Bretagne, France	Rapporteur
M. Brian MARK, Professeur, George Mason University, USA	Examinateur
M. Marc LELARGE, Chargés de Recherche, INRIA, France	Examinateur
M. Philippe MARTINS, Professeur, TELECOM ParisTech, France	Directeur de Thèse
M. Laurent DECREUSEFOND, Professeur, TELECOM ParisTech, France	Invité

TELECOM ParisTech
école de l’Institut Mines-Télécom - membre de ParisTech
46 rue Barrault 75013 Paris - (+33) 1 45 81 77 77 - www.telecom-paristech.fr

T H È S E

To my parents, two sisters and wife

Acknowledgements

I would like to express my deepest gratitude to my advisor Professor Philippe Martins. Before meeting him, I had never dreamed of pursuing my Ph.D. degree in France. It was Professor Philippe Martins who gave me such an opportunity. During the past four years, he gave me not only much invaluable guidance in research but also enough freedom and encouragement to explore an unknown world in my subject. He provided a lot of help and suggestions for my future professional career.

Meanwhile, I would like to express my sincere gratitude to my co-advisor Professor Laurent Decreusefond. He taught me a lot in stochastic geometry and algebraic topology and gave me much guidance in research and presentation skills. It was him who made me feel the powerfulness and magic of mathematic tools.

I would like to thank my committee members: Professor Philippe Godlewski, Professor Xavier Lagrange, Professor Martin Haenggi, Professor Brian Mark and Professor Marc Lelarge. Special thanks go to Professor Xavier Lagrange and Professor Martin Haenggi for their very careful comments and reports on my thesis.

I really very appreciate the help of Mrs. Liang Li during my first days in France. I am also very grateful to the members in our 'homology' group: Dr. Eduardo Ferraz, Dr. Anaïs Vergne, Dr. Ian Flint and Amine Benmosbah. Many ideas were inspired from discussions with them.

Particular thanks go to my close friends Dr. Tian Ban, Dr. Zizhu Wang and Wei He. With Tian, we shared our happiness and helped each other to overcome difficulties. Uncountable chats and discussions with Zizhu and Wei broadened my horizon as well as brought much happiness to me.

Finally, I would like to thank my parents and two sisters for their unconditional love, trust, support and encouragement. Special thank is for my wife Ping Yan, without her, I could not enjoy a better life in France.

Abstract

Homology theory provides new and powerful solutions to address the coverage hole problem in wireless sensor networks (WSNs). They are based on two combinatorial objects named Čech complex and Rips complex. Čech complex can detect all coverage holes, but it is very difficult to construct. Rips complex is easy to construct but it may be not accurate in some situations.

In the first part of this thesis, we choose the proportion of the area of holes missed by Rips complex as a metric to evaluate the accuracy of homology based coverage hole detection. Closed form expressions for lower and upper bounds of the proportion are derived. Simulation results are well consistent with the analytical lower and upper bounds, with maximum differences of 0.5% and 3%. In addition, we extend the analysis to the sphere case.

In the second part, we first propose a graph based distributed algorithm to detect non-triangular holes. This algorithm exhibits high complexity. We thus propose another efficient homology based distributed algorithm. This algorithm only requires 1- and 2-hop neighbour nodes information and has the worst case complexity $O(n^3)$ where n is the maximum number of 1-hop neighbour nodes. It can accurately detect the boundary cycles of about 99% coverage holes in about 99% cases.

Résumé

La théorie de l'homologie fournit des solutions nouvelles et efficaces pour régler le problème de trou de couverture dans les réseaux de capteurs sans fil. Ils sont basés sur deux objets combinatoires nommés complexe de Čech et complexe de Rips. Le complexe de Čech peut détecter l'intégralité des trous de couverture, mais il est très difficile à construire. Le complexe de Rips est facile à construire, mais il est imprécis dans certaines situations.

Dans la première partie de cette thèse, nous choisissons la proportion de la surface de trous manqués par le complexe de Rips comme une mesure d'évaluer l'exactitude de la détection de trou de couverture basée sur l'homologie. Des expressions fermées pour les bornes inférieures et supérieures de la proportion sont dérivés. Les résultats de simulation sont bien compatibles avec les bornes inférieure et supérieure d'analyse, avec des différences maximales de 0.5% et 3%. En outre, nous étendons l'analyse au cas de la sphère.

Dans la deuxième partie, nous proposons d'abord un algorithme distribué basé sur les graphes pour détecter les trous non triangulaires. Cet algorithme présente une grande complexité. Nous proposons donc un autre algorithme distribué plus efficace basé sur l'homologie. Cet algorithme ne nécessite que des informations de 1- et 2-saut nœuds voisins et a la complexité $O(n^3)$ où n est le nombre maximum de nœuds voisins à 1 saut. Il peut détecter avec précision les cycles frontières d'environ 99% des trous de couverture dans environ 99% des cas.

Publications

- Journal papers
  - Feng Yan, Philippe Martins and Laurent Decreusefond. *Accuracy of homology based coverage hole detection in wireless sensor networks on sphere*. Submitted to IEEE Transactions on Wireless Communications. http://arxiv.org/abs/1303.0001.

    (in Chapter 4)
  - Feng Yan, Anaïs Vergne, Philippe Martins and Laurent Decreusefond. *Homology-based distributed coverage hole detection in wireless sensor networks*. Submitted to IEEE/ACM Transactions on Networking. http://arxiv.org/abs/1302.7289.

    (in Chapter 3 and 6)
- Conference papers
  - Feng Yan, Philippe Martins and Laurent Decreusefond. *Accuracy of homology based approaches for coverage hole detection in wireless sensor networks*. In IEEE ICC, Ottawa, Canada, 10-15 June, 2012, pp. 497-502.

    (in Chapter 3)
  - Feng Yan, Philippe Martins and Laurent Decreusefond. *Connectivity-based distributed coverage hole detection in wireless sensor networks*. In IEEE GLOBECOM, Houston, TX, USA, 5-9 December, 2011, pp. 1-6.

    (in Chapter 5)

Contents

Acknowledgements iv

Abstract vi

Résumé viii

Publications x

Contents xi

List of Figures xv

List of Tables xix

0 Résumé long en français 1

1 Introduction 33

1.1 Motivations . . . 33

1.2 Objectives and contributions . . . 34

1.2.1 Objectives . . . 34

1.2.2 Contributions . . . 35

1.3 Organization . . . 37

2 Related Work and Mathematical Background 39

2.1 Traditional approaches . . . 39

2.1.1 Location based approaches . . . 39

2.1.2 Range based approaches . . . 40

2.1.3 Graph based approaches . . . 41

2.1.4 Remarks on these approaches . . . 43

2.2 Homology based approaches . . . 44

2.2.1 Mathematical background . . . 45
2.2.2 Homology based approaches . . . 52

3 Accuracy of Homology based Coverage Hole Detection on Plane 55
3.1 Introduction . . . 55
3.2 Models and definitions . . . 56
3.3 Bounds on proportion of triangular holes . . . 58
3.3.1 Preliminary . . . 59
3.3.2 Case $0 < \gamma \leq \sqrt{3}$ . . . 61
3.3.3 Case $\sqrt{3} < \gamma \leq 2$ . . . 61
3.3.4 Case $\gamma > 2$ . . . 68
3.4 Performance evaluation . . . 69
3.4.1 Simulation settings . . . 70
3.4.2 Performance evaluation . . . 70
3.4.3 Discussions on applications . . . 74
3.5 Chapter summary . . . 74

4 Accuracy of Homology based Coverage Hole Detection on Sphere 75
4.1 Introduction . . . 75
4.2 Models and definitions . . . 76
4.3 Bounds on proportion of spherical triangular holes . . . 81
4.3.1 Preliminary . . . 81
4.3.2 Case $0 < R_c \leq R \arccos([3\cos^2(R_s/R) - 1]/2)$ . . . 83
4.3.3 Case $R \arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ . . . 83
4.3.4 Case $R_c > 2R_s$ . . . 91
4.3.5 Case $R \to \infty$ . . . 93
4.4 Performance evaluation . . . 97
4.4.1 Simulation settings . . . 97
4.4.2 Impact of R_s and R_c . . . 97
4.4.3 Impact of R . . . 99
4.4.4 Discussions on applications . . . 101
4.5 Chapter summary . . . 102

5 Graph based Distributed Coverage Hole Detection 103
5.1 Introduction . . . 103
5.2 Models and assumptions . . . 103
5.3 Graph based distributed algorithm . . . 105

5.3.1 Neighbour discovery . . . 107
5.3.2 Boundary nodes discovery . . . 107
5.3.3 Boundary cycles discovery . . . 108
5.3.4 Cycles selection . . . 111
5.4 Simulations and performance evaluation . . . 112
5.4.1 Simulation settings . . . 113
5.4.2 Probability of boundary nodes detection . . . 113
5.4.3 Probability of boundary cycles detection . . . 113
5.5 Chapter summary . . . 116

6 Homology based Distributed Coverage Hole Detection 117
6.1 Introduction . . . 117
6.2 Models and definitions . . . 118
6.3 Homology based distributed algorithm . . . 120
6.3.1 Weight computation . . . 120
6.3.2 Vertex and edge deletion . . . 122
6.3.3 Boundary edge detection . . . 124
6.3.4 Coarse boundary cycles discovery . . . 126
6.3.5 Boundary cycles minimization . . . 127
6.4 Performance evaluation . . . 128
6.4.1 Simulation settings . . . 128
6.4.2 Complexity . . . 128
6.4.3 Comparison with boundary recognition algorithm . . . 129
6.4.4 Comparison with location based algorithm . . . 130
6.5 Chapter summary . . . 132

7 Conclusions and Future Work 133
7.1 Major contributions . . . 133
7.2 Future research directions . . . 134

A Computation of Area $|S^-(r_0, r_1, \varphi_1)|$ 137
A.1 Area $|S^-(r_0, r_1, \varphi_1)|$ in case $\sqrt{3} < \gamma \leq 2$ . . . 137
A.2 Area $|S^-(r_0, r_1, \varphi_1)|$ in case $\gamma > 2$ . . . 139

B Detailed Values of Simulation Results and Bounds on Sphere 143

References 151

List of Figures

1 Un exemple de simplexes 5
2 Un exemple de complexe simplicial 5
3 Illustrations de bord 7
4 Un exemple montrant les relations de $C_k(X), Z_k(X), B_k(X)$ 7
5 (a) Un réseau de capteurs sans fil, (b) le complexe de Čech, (c) le complexe de Rips pour $R_c = 2R_s$, (d) le complexe de Rips pour $R_c = 2.5R_s$ 11
6 La proportion de la superface de trous triangulaires (a) les résultats de simulation et des bornes inférieures, (b) les résultats de simulation et des borne supérieures, (c) les résultats de simulation pour $p_{2d}^{\text{sec}}(\lambda)$. 16
7 La proportion de la surface de trous triangulaires sphériques sous $R = 10R_s$ (a) les résultats de simulation et des bornes inférieures, (b) les résultats de simulation et des bornes supérieures 20
8 La comparaison de la proportion de la surface de trous triangulaires sphériques (a) la comparaison des résultats de simulation, (b) la comparaison des bornes inférieures, (c) la comparaison des bornes supérieures 21
9 Les résultats de simulation pour $p_s^{\text{sec}}(\lambda)$ 22
10 Les procédures du algorithme distribué basé sur les graphes. (a) le complexe de Rips d'un réseau de capteurs sans fil, (b∼c) découverte des nœuds limites, (d∼e) découverte des cycles limites, (f) sélection des cycles 24
11 Diagramme de l'algorithme 26
12 Les procédures de l'algorithme. (a) le complexe de Rips d'un réseau de capteurs, (b) après suppression de sommet, (c∼d) suppression d'arête, (e∼j) détection d'arête frontière, (k) découverte de cycles frontières élémentaires, (l) minimisation de cycles frontières 27

2.1 Examples illustrating that it is not sufficient to only detect boundary nodes in order to discover boundary cycles 44
2.2 An example of simplices 45
2.3 An example of simplicial complex 46
2.4 Illustrations of boundary 47
2.5 An example showing the relations of $C_k(X), Z_k(X), B_k(X)$ 48
2.6 An example of Rips complex falsely detects a coverage hole, assume $R_c = R_s$ 54
2.7 An example of Rips complex missing a coverage hole, assume $R_c = 2R_s$ 54

3.1 (a) A WSN, (b) Čech complex, (c) Rips Complex under $R_c = 2R_s$, (d) Rips Complex under $R_c = 2.5R_s$ 57
3.2 Illustration of O being inside a triangular hole 60
3.3 Illustration of region A^+ in the case $\sqrt{3} < \gamma \leq 2$ 63
3.4 Illustration of regions S^+ and S^- in the case $\sqrt{3} < \gamma \leq 2$ 64
3.5 Illustration of regions S'^+ and S'^- in the case $\sqrt{3} < \gamma \leq 2$ 65
3.6 Illustrations of regions in case $\gamma > 2$. (a) the regions A^+, S^+ and S^-, (b) the regions S'^+ and S'^- 69
3.7 Proportion of the area of triangular holes (a) simulation results and lower bounds, (b) simulation results and upper bounds, (c) simulation results for $p_{2d}^{\text{sec}}(\lambda)$ 71

4.1 0-, 1- and 2-simplex on sphere 77
4.2 (a) A WSN, (b) Čech complex, (c) Rips Complex under $R_c = 2R_s$, (d) Rips Complex under $R_c = 2.5R_s$ 78
4.3 Illustrations of P and spherical triangle $v_0v_1v_2$: (a) P is inside the spherical triangle $v_0v_1v_2$, (b) P is outside the spherical triangle $v_0v_1v_2$ 79
4.4 Illustration of region A^+ in the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ 86
4.5 Illustration of regions S^+ and S^- in the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ 87
4.6 Illustration of regions S'^+ and S'^- in the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ 89
4.7 Illustrations of regions in case $R_c > 2R_s$. (a) the regions A^+, S^+ and S^-, (b) the regions S'^+ and S'^- 92

4.8 Proportion of the area of spherical triangular holes under $R = 10R_s$ (a) simulation results and lower bounds, (b) simulation results and upper bounds . . . 98
4.9 Comparison of the proportion of the area of spherical triangular holes (a) comparison of simulation results, (b) comparison of lower bounds, (c) comparison of upper bounds . . . 100
4.10 Simulation results for $p_s^{\text{sec}}(\lambda)$ . . . 101

5.1 Examples of equivalent transformations . . . 105
5.2 Procedures of graph based distributed algorithm. (a) Rips complex of a WSN, (b∼c) boundary nodes discovery, (d∼e) boundary cycles discovery, (f) cycles selection . . . 106
5.3 An example illustrating the process of this algorithm. (a) Rips complex of a WSN, (b∼c) boundary nodes discovery, (d∼h) boundary cycles discovery, (i) cycles selection . . . 112
5.4 (a) Success probability of boundary nodes detection p_b^{succ}, (b) error probability of boundary nodes detection p_b^{err} . . . 114
5.5 (a) Success probability of boundary cycles detection p_c^{succ}, (b) error probability of boundary cycles detection p_c^{err} . . . 115

6.1 Flow chart of the algorithm . . . 120
6.2 Procedures of the boundary detection algorithm. (a) Rips complex of a WSN, (b) after vertex deletion, (c∼d) edge deletion, (e∼j) boundary edge detection, (k) coarse boundary cycles discovery, (l) boundary cycles minimization . . . 121
6.3 Illustration of crossing boundary edges . . . 125
6.4 Illustration of false boundary edges . . . 126
6.5 Special case when some boundary edges can not be detected . . . 126
6.6 (a) Rips complex of a WSN, (b) coarse boundary cycles found by HBA, (c) primary boundary cycles found by BR . . . 131

A.1 Illustration of the region $S^-(r_0, r_1, \varphi_1)$ in the case $\sqrt{3} < \gamma \leq 2$ . . . 137
A.2 For the computation of β_1 . . . 138
A.3 Illustration of two cases for $\beta_0(r0, r1, \varphi_1)$ . . . 140
A.4 Illustration of $R_2(r_0, r_1, \varphi_1, 0)$ . . . 141

List of Tables

1 La complexité de chaque étape de l'algorithme 31
2 Comparaison avec l'algorithme basé sur localisation 32

1.1 Summary of publications 36

2.1 Characteristics of typical traditional approaches for coverage hole detection 43
2.2 Summary of traditional coverage hole detection approaches in WSNs 43
2.3 Characteristics of homology based approaches for coverage hole detection 53

3.1 $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.0, 2.2$ (%) 72
3.2 $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.4, 2.6$ (%) 72
3.3 $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.8, 3.0$ (%) 73
3.4 Simulation results for $p_{2d}^{\text{sec}}(\lambda)$ 73

5.1 Message structure 109
5.2 Number of boundary cycles 116

6.1 Complexity of each step in the algorithm 128
6.2 Number of non-triangular holes 132

B.1 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.0, 2.2$ (%) 143
B.2 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.4, 2.6$ (%) 144
B.3 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.8, 3.0$ (%) 144
B.4 Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 5R_s$ 145
B.5 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.0, 2.2$ (%) 145

B.6 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.4, 2.6$ (%) 146
B.7 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.8, 3.0$ (%) 146
B.8 Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 10R_s$ 147
B.9 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.0, 2.2$ (%) 147
B.10 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.4, 2.6$ (%) 148
B.11 $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.8, 3.0$ (%) 148
B.12 Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 100R_s$ 149

Chapter 0

Résumé long en français

0.1 Introduction

0.1.1 Motivations

Les récents progrès des communications sans fil et des micro-systèmes électro-mécaniques ont permis le développement des réseaux de capteurs [1]. Un réseau de capteurs se compose d'un ensemble de nœuds disposant de capacités de calcul et de transmission réduites. Chacun capteur possède néanmoins des fonctionnalités de mesure et de communication avec des nœuds voisins. Ces capteurs sont déployés dans un domaine à superviser. Plusieurs types de mesures peuvent être effectuées : température, luminosité, intensité sonore, pression .... Les réseaux de capteurs ont beaucoup d'applications, qui peuvent être généralement classés en deux catégories: la supervision et la surveillance [2]. Les applications de surveillance concernent essentiellement les personnes, les animaux et les véhicules [3, 4, 5, 6, 7, 8, 9]. Les applications de supervision sont liées à la l'habitat, l'environnement. [10, 11, 12, 13, 14, 15, 16, 17], l'eSanté [18, 19, 20, 21, 22], la télé-médecine [23, 24, 25, 26] et le contrôle de trafic [27, 28, 29, 30, 31, 32].

La couverture est un problème fondamental dans les réseaux de capteurs. Elle peut être considérée comme une mesure de la qualité de service d'un réseau de capteurs [33]. Basé sur les objectifs de couverture, les auteurs de [34] ont classé la couverture en trois catégories: la couverture d'une zone géographique, la couverture d'un point spécifique et la couverture d'obstacle. Pour qui concerne le problème de la couverture de zone, l'objectif principal du réseau de capteurs est de couvrir l'intégralité des points de la zone. Dans le problème de couverture du point, l'objectif est de couvrir un ensemble de points spécifiques. Le but de la couverture d'obstacle

est de minimiser la probabilité de pénétration non détectée à travers la barrière formée par des capteurs sans fil. Dans cette thèse, nous nous concentrons sur le problème de couverture de la zone.

Dans les applications liées à la couverture de zone, le domaine ciblé doit être être entièrement couvert. Cependant, les trous de couverture peuvent se former pour de nombreuses raisons, telles que le déploiement aléatoire des capteurs, l'épuisement de l'énergie de capteurs ou la destruction de capteurs. Par exemple, dans les applications de surveillance de volcan, le domaine ciblé est généralement hostile ou inaccessible pour les humains. Dans de telles situations, les capteurs doivent être déployées de façon aléatoire par des hélicoptères ou d'autres types de véhicules. Avec le déploiement aléatoire, nœuds de capteurs peuvent se regrouper à un endroit et laisser des trous de couverture à d'autres. En outre, même si dans le déploiement initial, le domaine ciblé est entièrement couvert par des nœuds de capteurs, des trous de couverture peuvent apparaître avec le temps: extinction, pannes, et destruction physique de capteurs. Par conséquent, il est d'une importance primordiale de disposer de mécanismes permettant de détecter et de localiser les trous de couverture. Certains nœuds mobiles peuvent alors être déployés pour rétablir la couverture [35, 36]. Par ailleurs, la connaissance des frontières des trous de couverture est également très utile dans la conception de fonctionnalités des réseaux, telles que le routage point à point et les mécanismes de collecte de données [37].

De nombreuses approches ont été proposées pour la détection de trou de couverture dans les réseaux de capteurs sans fil. Elles peuvent généralement être classés en trois catégories: (i) les approches basées sur la localisation, (ii) les approches basées sur la distance, et (iii) les approches basées sur la connectivité. Les approches basées sur la localisation et la distance peuvent découvrir tous les trous de couverture avec une bonne précision mais nécessitent soit des informations de localisation précises soit des informations de distance précise, ce qui est difficile à obtenir dans de nombreux scénarios. Les approches basées sur la connectivité ont donc reçu une attention considérable ces dernières années. En particulier, les approches basées sur l'homologie ont attiré notre attention. Ghrist et ses collaborateurs ont introduit deux outils combinatoires : le complexe de Čech et le complexe de Rips-Vietoris (qu'on désignera par le terme complexe de Rips dans la suite de ce document). Ces deux objets sont utilisés pour détecter les trous de couverture [38]. Une solution centralisée pour certains scénarios simples a été proposée par ses deux auteurs. Néanmoins, la conception d'un algorithme distribué efficace pour la détection de trous de couvertures, en se basant uniquement sur des informations de

connectivité, demeure une question ouverte.

Conscients des limites du travail existant pour la détection de trous de couverture dans les réseaux de capteurs sans fil, nos travaux analysent plus en détail l'utilité de la théorie d'homologie. Nos travaux ont permis d'élaborer de nouveaux algorithmes distribués plus adaptés au problème de la detection des trous de couverture dans un réseau de capteurs sans fil.

0.1.2 Contributions

Nos contributions sont de deux ordres: (1) la précision de la détection de trous de couverture basée sur l'homologie et (2) des algorithmes distribués pour la détection de trous de couverture.

0.1.2.1 La précision de la détection de trous de couverture basée sur l'homologie

Le complexe de Čech détecte l'intégralité des trous de couverture, mais il est difficile à construire. Le complexe de Rips est facile à construire, mais est imprécis dans certaines situations. Nous avons d'abord analysé la relation entre le complexe de Čech et le complexe de Rips en termes de trous de couverture pour les réseaux de capteurs sans fil sur un domaine ciblé plan. Nous choisissons la proportion de la surface de trous manqués par le complexe de Rips comme une mesure d'exactitude de la détection de trous de couverture basée sur l'homologie. Nous avons montré que cette proportion est liée au ratio entre les rayons de communication et de détection de chaque capteur (noté respectivement par R_c et R_s). Nous avons ensuite analysé cette proportion dans trois cas et pour chaque cas, des expressions fermées pour les bornes inférieure et supérieure ont été dérivés. Les résultats de simulation sont compatibles avec les bornes inférieure et supérieure calculés analytiquement, avec des différences maximales de 0.5% et 3%.

En outre, nous avons étendu l'analyse au cas de la sphère. Les résultats de simulation montrent que le rayon de la sphère a peu d'impact sur la proportion quand il est beaucoup plus grand que les rayons de communication et de détection de chaque capteur.

0.1.2.2 Algorithmes distribués pour la détection de trou de couverture

Nous avons proposé un algorithme distribué basé sur les graphes pour détecter les trous de couverture. L'algorithme se compose de quatre étapes: découverte des

voisins, découverte des nœuds frontières, découverte des cycles frontières, sélection des cycles. Dans l'étape de la découverte des voisins, chaque nœud obtient des informations de tous ses 1- et 2-saut voisins. Dans l'étape de la découverte des nœuds frontières, chaque nœud détermine s'il s'agit d'un nœud frontière ou non en vérifiant l'existence d'un cycle Hamiltonien dans son graphe de voisinage. S'il existe, il est considéré comme un nœud non frontière. Dans le cas contraire, il est supposé être un nœud frontière. Après cela, certains nœuds frontières sont choisis au hasard pour lancer le processus de découverte des cycles frontières et de nombreux cycles seront trouvés. Mais ces cycles trouvés peuvent ne pas être minimum ou certains d'entre eux être liés aux mêmes trous de couverture. Donc, il est nécessaire de faire des choix parmi ces cycles, ce qui se fait dans l'étape finale.

L'algorithme proposé ci-dessus présente une grande complexité. Nous avons donc conçu un algorithme plus efficace basé sur l'homologie. L'idée de base de cet algorithme est la suivante: pour le complexe de Rips d'un réseau de capteurs, nous essayons de supprimer des sommets et des arêtes sans changer l'homologie tout en rendant le complexe de Rips moins dense. Ensuite, il devient plus facile de trouver des cycles frontières. La nouveauté de cet algorithme réside dans le critère de suppression d'un sommet ou d'une arête. A chaque tentative de suppression, l'algorithme vérifie si l'homologie change. Cet algorithme a la complexité $O(n^3)$ où n est le nombre maximum de nœuds voisins à 1 saut et n est fini. Il est distribué et ne nécessite que des informations de 1- et 2-saut nœuds voisins. Il peut détecter avec précision les cycles frontières d'environ 99% des trous de couverture dans environ 99% des cas.

0.2 Contexte mathématique

La théorie de l'homologie fournit des solutions nouvelles et puissantes pour les problèmes d'identification des trous de couverture dans les réseaux de capteurs sans fil. Le principe de la théorie de l'homologie consiste à analyser les propriétés topologiques de certains domaines par des calculs algébriques. Les principaux objets sont connus comme des complexes simpliciaux, qui sont la généralisation du graphe. Le groupe d'homologie est un invariant topologique qui peut faire la distinction entre les espaces topologiques en mesurant le nombre de trous dans cet espace. Ces concepts étant relativement moins connus, il est donc nécessaire d'introduire un certain nombre de concepts fondamentaux. Pour une présentation détaillée, le lecteur pourra se reporter à [39, 40, 41].

0.2.1 Définitions

Etant donné un ensemble de sommets V, un k-simplexe σ est un ensemble non-ordonné $\{v_0, v_1, ..., v_k\} \subseteq V$ où $v_i \neq v_j$ pour tout $i \neq j$, k est la dimension du simplexe. Un 0-simplexe est un sommet, un 1-simplexe est une arête, un 2-simplexe est un triangle avec son intérieur inclus et un 3-simplexe est un tétraèdre avec son intérieur inclus, voir la Figure 1. Tout sous-ensemble de $\{v_0, v_1, ..., v_k\}$ est appelé une face de σ. Un complexe simplicial X est un ensemble de simplexes qui satisfait à deux conditions: (1) toute la face d'un simplexe de X est aussi dans X, (2) l'intersection de deux simplexes σ_1 et σ_2 est une face de σ_1 et $sigma_2$. Un complexe simplicial abstrait est la description purement combinatoire de la notion géométrique d'un complexe simplicial et n'a donc pas besoin de la seconde condition.

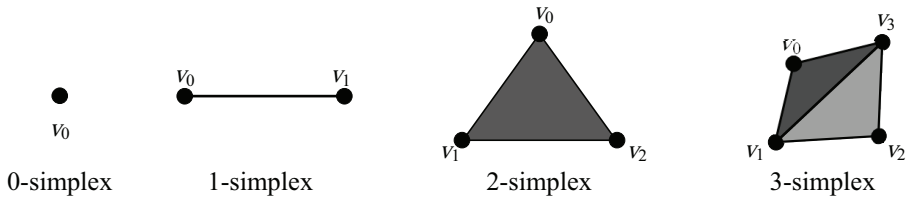


Figure 1: Un exemple de simplexes

La dimension d'un complexe simplicial X est la plus grande dimension de tout simplexe dans X. Un sous-complexe de X est un complexe simplicial $X^{(k)} \subset X$, où k indique la dimension de $X^{(k)}$.

Par exemple, dans le complexe simplicial représenté sur la Figure 2, il contient six 0-simplexes $\{1\}, \{2\}, \ldots, \{6\}$, huit 1-simplexes $\{1,2\}, \{1,6\}, \{2,3\}, \{2,6\}, \{3,4\}, \{3,5\}, \{4,5\}, \{5,6\}$ et un 2-simplexe $\{1,2,6\}$.

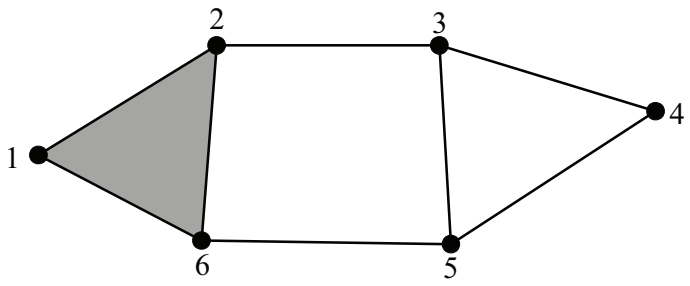


Figure 2: Un exemple de complexe simplicial

Soit X un complexe simplicial abstrait. On peut définir une orientation pour chaque k-simplexe dans X. Notons le k-simplexe $\{v_0, v_1, \cdots, v_k\}$ avec comme ordre

$[v_0, v_1, \cdots, v_k]$. Un changement d'orientation correspond à un changement de signe sur le coefficient comme

$$[v_0, \cdots, v_i, \cdots, v_j, \cdots, v_k] = -[v_0, \cdots, v_j, \cdots, v_i, \cdots, v_k]$$

Ensuite, on peut définir le groupe des chaînes, la différentielle de carré nul, le groupe des cycles et le groupe des bords.

Definition 0.1. *Pour un complexe simplicial abstrait X, pour chaque entier $k > 0$, le k-ième groupe des chaînes $C_k(X)$ est l'espace vectoriel formé par l'ensemble des k-simplexes orientés de X. Si k est plus grand que la dimension de X, $C_k(X)$ est 0.*

Definition 0.2. *La différentielle de carré nul ∂_k est la transformation linéaire $\partial_k : C_k(X) \to C_{(k-1)}(X)$ qui agit sur les éléments de base de $C_k(X)$ via*

$$\partial_k[v_0, \cdots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k]$$

où la somme est la somme algébrique.

Prenons l'exemple de la Figure 2, en tenant compte de l'orientation, nous supposons qu'il contient six 0-simplexes $[1], [2], \ldots, [6]$, huit 1-simplexes $[1,2], [1,6], [2,3], [2,6], [3,4], [3,5], [4,5], [5,6]$ et un 2-simplexe $[1,2,6]$. Selon la définition 0.2, on peut voir que le bord du 2-simplexe $[1,2,6]$ est la somme de trois 1-simplexes $[1,2], [2,6]$ et $[6,1]$, comme illustré sur la Figure 3(a). Mais le bord de la somme de trois 1-simplexes $[3,4], [4,5]$ et $[5,3]$ est 0, sur la Figure 3(b).

Definition 0.3. *Le k-ième group des cycles de X est $Z_k(X) = \ker \partial_k$.*

Definition 0.4. *Le k-ième group des bords de X est $B_k(X) = \mathrm{im}\partial_{k+1}$.*

Un calcul simple montre que $\partial_k \circ \partial_{k+1} = 0$. Il s'ensuit que $B_k(X) \subset Z_k(X)$. Les relations de $C_k(X), Z_k(X), B_k(X)$ sont montrées dans la Figure 4.

Ensuite, on peut bien définir le groupe d'homologie et sa dimension.

Definition 0.5. *Le k-ième groupe d'homologie de X est l'espace vectoriel quotient*

$$H_k(X) = \frac{Z_k(X)}{B_k(X)}$$

Definition 0.6. *Le k-ième nombre de Betti de X est la dimension de $H_k(X)$:*

$$\beta_k = \dim H_k(X) = \dim Z_k(X) - \dim B_k(X).$$

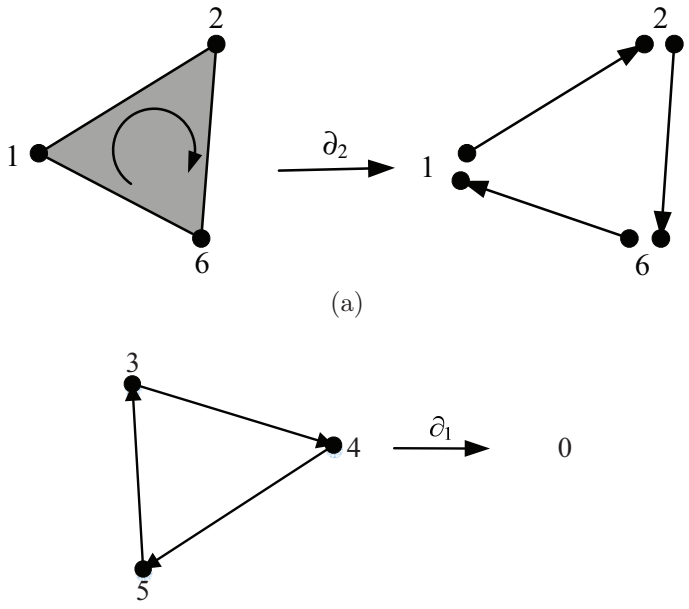


Figure 3: Illustrations de bord

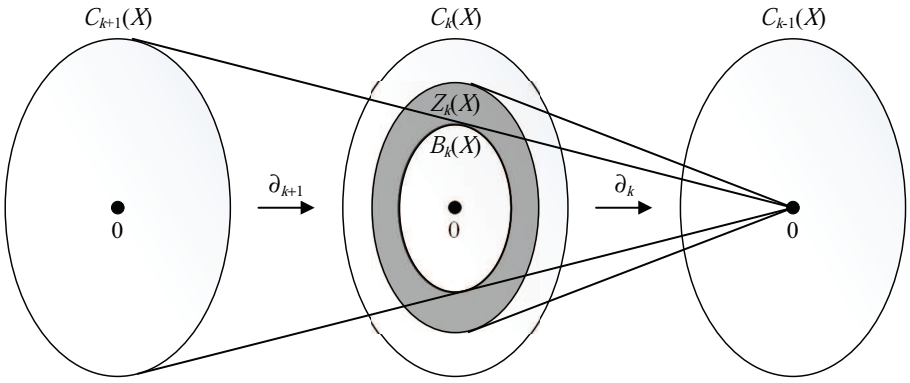


Figure 4: Un exemple montrant les relations de $C_k(X), Z_k(X), B_k(X)$

Les nombres de Betti sont utilisés pour compter le nombre de trous de différentes dimensions dans X. Par exemple, β_0 indique le nombre de trous de dimension 1, qui est le nombre de composantes connexes. Et β_1 compte le nombre de trous dans le plan.

On prend l'exemple dans la Figure 2, les différentielles associées à X sont faciles à obtenir sous forme matricielle:

$$\partial_1 = \begin{array}{c} \\ [1] \\ [2] \\ [3] \\ [4] \\ [5] \\ [6] \end{array} \begin{array}{c} \begin{array}{cccccccc} [1,2] & [1,6] & [2,3] & [2,6] & [3,4] & [3,5] & [4,5] & [5,6] \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\partial_2 = \begin{array}{c} \\ [1,2] \\ [1,6] \\ [2,3] \\ [2,6] \\ [3,4] \\ [3,5] \\ [4,5] \\ [5,6] \end{array} \begin{array}{c} \begin{array}{c} [1,2,6] \end{array} \\ \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

Ensuite on peut obtenir

$$\beta_0 = \dim \ker \partial_0 - \dim \mathrm{im} \partial_1 = 6 - 5 = 1$$
$$\beta_1 = \dim \ker \partial_1 - \dim \mathrm{im} \partial_2 = 3 - 1 = 2$$

0.2.2 Complexes simpliciaux abstraits pour les réseaux

Pour le problème de la couverture dans les réseaux de capteurs sans fil, les deux complexes simpliciaux abstraits plus utiles sont le complexe de Čech et le complexe de Rips.

Le complexe de Čech est défini comme suit [38].

Definition 0.7 (Complexe de Čech)**.** *Etant donné une collection d'ensembles* $\mathcal{U}$*, le complexe de Čech de* $\mathcal{U}$*,* $\check{C}(\mathcal{U})$*, est le complexe simplicial abstrait dont les* k*-simplexes correspondent aux intersections non-vides de* $k+1$ *éléments distincts de* $\mathcal{U}$*.*

Le complexe de Čech capture la topologie de la collection d'ensembles, comme illustré par le théorème suivant.

Theorem 0.1 (Théorème de Čech)**.** *Le complex de Čech complex d'une collection d'ensembles convexes a le type d'homotopie de l'union des ensembles.*

Malheureusement, le complexe de Čech est très difficile à construire. Donc, un autre complexe plus facilement calculable nommé le complexe de Rips est introduit. Il est défini comme suit.

Definition 0.8 (Complexe de Rips)**.** *Soient (M, d) un espace métrique, $\mathcal{V}$ un ensemble fini de points dans M et ϵ un réel positif, le complexe de Rips de $\mathcal{V}$, $\mathcal{R}_\epsilon(\mathcal{V})$, est le complexe simplicial abstrait dont les k-simplexes sont les $(k+1)$-tuples de points de $\mathcal{V}$ qui sont de distance inférieure à ϵ deux à deux.*

Le complexe de Rips peut être construit avec la seule connaissance du graphe de connectivité du réseau et donne une approximation de la couverture par des calculs algébriques simples. Mais le complexe de Rips peut manquer des trous de couverture. En fait, il existe les relations suivantes entre le complexe de Čech et le complexe de Rips [42].

Theorem 0.2. *Soit $\mathcal{V}$ un ensemble fini de points dans $\mathbb{R}^d$ et $\check{C}_\epsilon(\mathcal{V})$ le complexe de Čech de la couverture de $\mathcal{V}$ par des boules de rayon ϵ. Ensuit, il y a*

$$\mathcal{R}_{\epsilon'}(\mathcal{V}) \subset \check{C}_\epsilon(\mathcal{V}) \subset \mathcal{R}_{2\epsilon}(\mathcal{V}) \qquad \textit{whenever} \qquad \frac{\epsilon}{\epsilon'} \geq \sqrt{\frac{d}{2(d+1)}}$$

0.3 Précision de la détection de trou de couverture basée sur l'homologie dans le plan

0.3.1 Introduction

Le complexe de Čech et le complexe de Rips sont deux outils utiles pour la détection de trous de couverture. Le complexe de Čech est assez difficile à construire et le complexe de Rips est facile à construire. Les approches basées sur l'homologie utilisent généralement le complexe de Rips pour détecter les trous de couverture. Néanmoins, le complexe de Rips peut manquer quelques trous de couverture dans certaines situations. Dans ce qui suit, nous choisissons la proportion de la surface de trous manqués par le complexe de Rips comme une mesure de précision de la détection de trous de couverture basée sur l'homologie pour les réseaux de capteurs sans fil dans le plan.

Nous avons d'abord analysé la relation entre le complexe de Čech et le complexe de Rips en termes de trous de couverture. Nous trouvons que leur relation est liée au ratio entre les rayons de la communication et de détection. Les trous manqués

par le complexe de Rips doivent être délimités par un triangle. Nous définissons ainsi le concept de trous triangulaires et de trous non-triangulaire.

Par la suite, nous obtenons des formes fermées pour les bornes inférieures et supérieures de la proportion de la surface de trous triangulaires.

Enfin, des simulations ont été menées et elles ont montré que les résultats de simulation sont bien compatibles avec les bornes analytiques inférieure et supérieure, avec des différences maximales de 0.5% et 3%.

0.3.2 Modèles et définitions

On considère un ensemble de capteurs fixes (appelé aussi nœuds) déployé sur un domaine ciblé plan modélisé par un processus de Poisson d'intensité λ. La propagation radio est supposé isotrope. Chaque capteur contrôle une région à l'intérieur d'un cercle de rayon R_S et peut communiquer avec d'autres capteurs dans un cercle de rayon R_c. Soit $\mathcal{V}$ l'ensemble de emplacements de capteurs dans un réseau de capteurs sans fil et $\mathcal{S} = \{s_v,\, v \in \mathcal{V}\}$ l'ensemble des disques de détection de ces capteurs: pour un emplacement v, $s_v = \{x \in \mathbb{R}^2 : \|x - v\| \leq R_s\}$. Ensuite, selon la définition 0.7, le complexe de Čech du réseau de capteurs sans fil, noté $\check{C}_{R_s}(\mathcal{V})$, peut être construit comme suit: un k-simplexe $[v_0, v_1, \cdots, v_k]$ appartient à $\check{C}_{R_s}(\mathcal{V})$ chaque fois que $\cap_{l=0}^{k} s_{v_l} \neq \emptyset$. De même, selon la définition 0.8 et nous considérons ici l'espace métrique $(\mathbb{R}^2, d)$, le complexe de Rips, noté $\mathcal{R}_{R_c}(\mathcal{V})$, peut être construit comme suit: un k-simplexe $[v_0, v_1, \cdots, v_k]$ appartient à $\mathcal{R}_{R_c}(\mathcal{V})$ chaque fois que $\|v_l - v_m\| \leq R_c$ pour tout $0 \leq l < m \leq k$.

La figure 5 montre un réseau de capteurs, son complexe de Čech et deux complexes de Rips pour deux valeurs différentes de R_c. En fonction du ratio de R_c sur R_s, le complexe de Rips et le complexe de Čech peuvent être proches ou plutôt différents. Dans cet exemple, pour $R_c = 2R_s$, le complexe de Rips voit le trou entouré par des nœuds $2, 3, 5, 6$ comme dans le complexe de Čech tandis qu'il n'est pas visible dans le complexe de Rips pour $R_c = 2.5R_s$. Dans le même temps, le trou de couverture entouré par les nœuds $1, 2, 6$ n'est pas présent dans les deux complexes de Rips.

En fait, le théorème de Čech (Theorem 0.1) indique que tout trou de couverture peut être trouvé dans le complexe de Čech. En outre, selon le Theorem 0.2, soit $d = 2$, $\epsilon = R_s$ et $\epsilon' = R_c$, il y a des relations suivantes entre le complexe de Čech et le complexe de Rips:

$$\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V}) \subset \mathcal{R}_{2R_s}(\mathcal{V}), \text{ whenever } R_c \leq \sqrt{3}R_s. \tag{1}$$

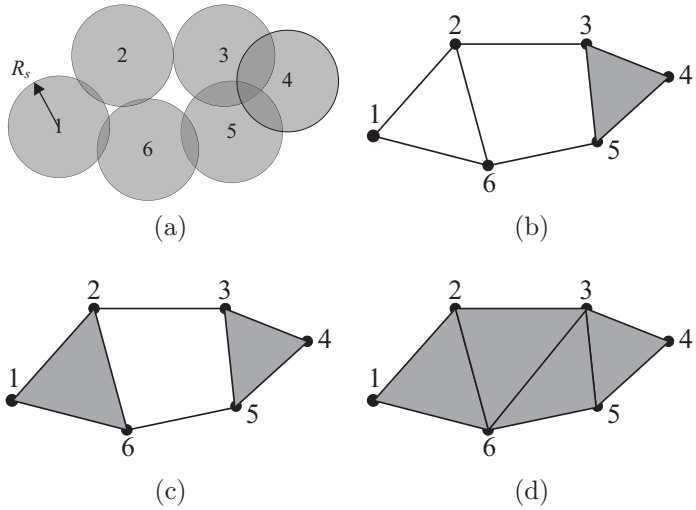


Figure 5: (a) Un réseau de capteurs sans fil, (b) le complexe de Čech, (c) le complexe de Rips pour $R_c = 2R_s$, (d) le complexe de Rips pour $R_c = 2.5R_s$

Selon (1), certaines relations entre le complexe de Čech et le complexe de Rips en termes de trous de couverture peuvent être dérivées comme illustré dans les corollaires suivants. Pour plus de commodité, on définit $\gamma = R_c/R_s$.

Corollary 0.3. *Quand $\gamma \leq \sqrt{3}$, s'il n'y a pas de trou dans le complexe de Rips $\mathcal{R}_{R_c}(\mathcal{V})$, il n'yaura aucun trou dans le complexe de Čech $\check{C}_{R_s}(\mathcal{V})$.*

Corollary 0.4. *Quand $\gamma \geq 2$, s'il y a un trou dans le complexe de Rips $\mathcal{R}_{R_c}(\mathcal{V})$, il y aura un trou dans le complexe de Čech $\check{C}_{R_s}(\mathcal{V})$.*

Corollary 0.5. *Quand $\sqrt{3} < \gamma < 2$, il n'y a pas de relation garantie entre le complexe de Rips $\mathcal{R}_{R_c}(\mathcal{V})$ et le complexe de Čech $\check{C}_{R_s}(\mathcal{V})$ en termes de trous.*

D'après ce qui précède, nous pouvons constater que pour qu'un trou dans un complexe de Čech ne soit pas vu dans un complexe de Rips, il faut et il suffit qu'il soit délimité par un triangle. Basé sur cette observation, une définition formelle de 'trou triangulaire' et de 'trou non-triangulaire' est donnée comme suit.

Definition 0.9 (Trou triangulaire et non-triangulaire)**.** *Pour une paire de complexes $\check{C}_{R_s}(\mathcal{V})$ et $\mathcal{R}_{R_c}(\mathcal{V})$ d'un réseau de capteurs, un trou triangulaire est une région non couverte délimitée par un triangle formé par les trois nœuds v_0, v_1, v_2, où v_0, v_1, v_2 peuvent former un 2-simplexe qui apparaît dans $\mathcal{R}_{R_c}(\mathcal{V})$ mais pas dans $\check{C}_{R_s}(\mathcal{V})$. Tous les autres trous sont non-triangulaires.*

De la définition 0.9, nous pouvons voir dans la Figure 5 que quand $R_c = 2R_s$, il y a seulement un trou triangulaire délimitée par le triangle formé par les nœuds

1, 2 et 6. Lorsque $R_c = 2.5R_s$, il existe deux trous triangulaires supplémentaires, délimitées par des triangles formés par des nœuds 2, 3, 6 et 3, 5, 6 respectivement.

0.3.3 Des bornes sur la proportion de la surface de trous triangulaires

Dans cette section, les conditions dans lesquelles n'importe quel point sur le domaine ciblé est dans un trou triangulaire sont d'abord donnés. Trois cas différents sont pris en compte pour le calcul de la proportion. Pour chaque cas, les bornes supérieures et inférieures de la proportion sont dérivées.

Lemma 0.6. *Pour tout point sur le domaine ciblé, il est à l'intérieur d'un trou triangulaire si et seulement si les deux conditions suivantes sont satisfaites:*

1. *la distance entre le point et son nœud le plus proche est plus grande que R_s.*
2. *le point est à l'intérieur d'un triangle formé par trois nœuds avec la distance par paire inférieure ou égale à R_c.*

Lemma 0.7. *S'il existe un point O qui est à l'intérieur d'un trou triangulaire, alors $R_s < R_c/\sqrt{3}$.*

Lemma 0.8. *Si O est à l'intérieur d'un trou triangulaire et l désigne la distance entre O et son voisin le plus proche, nous avons $R_s < l \leq R_c/\sqrt{3}$.*

Un processus de Poisson homogène est stationnaire, donc sans tenir compte de l'effet frontière [43], tout point a la même probabilité d'être dans un trou triangulaire que l'origine O. Cette probabilité dans une configuration homogène est aussi égale à la proportion de la surface de trous triangulaires.

Nous considérons la probabilité que l'origine O soit à l'intérieur d'un trou triangulaire. Puisque la longueur de chaque arête dans le complexe de Rips doit être au plus R_c, seuls les nœuds dans R_c de l'origine peuvent contribuer au triangle qui délimite un trou triangulaire qui contient l'origine. Par conséquent, nous avons seulement besoin de considérer le processus de Poisson restreint à la la boule fermée $B(O, R_c)$, qui est aussi un processus de Poisson homogène d'intensité λ. On note ce processus Φ. En outre, $T(x, y, z)$ désigne la propriété que l'origine O est à l'intérieur du trou triangulaire délimitée par le triangle avec des points x, y, z en tant que sommets. Quand n_0, n_1, n_2 sont des points du processus Φ, $T(n_0, n_1, n_2)$ est également utilisé pour désigner le cas où le triangle formé par les nœuds n_0, n_1, n_2 délimite un

trou triangulaire qui contient l'origine. En outre, nous utilisons $T'(n_0, n_1, n_2)$ pour désigner le cas où les nœuds n_0, n_1, n_2 ne peuvent pas former un triangle qui délimite un trou triangulaire qui contient l'origine.

Soit $\tau_0 = \tau_0(\Phi)$ le nœud dans le processus Φ qui est plus proche de l'origine. Il y a deux cas à considérer lorsque l'origine à être à l'intérieur d'un trou triangulaire. Le premier cas est que le nœud τ_0 peut contribuer à un triangle qui délimite un trou triangulaire qui contient l'origine. Le deuxième cas est que le nœud τ_0 ne peut pas contribuer à un triangle qui délimite un trou triangulaire qui contient l'origine, mais trois autres nœuds peuvent former un triangle qui délimite un trou triangulaire qui contient l'origine. Ainsi, la probabilité que l'origine soit à l'intérieur d'un trou triangulaire peut être définie comme

$$\begin{aligned} p_{2d}(\lambda) &= \mathrm{P}\{O \text{ is inside a triangular hole}\} \\ &= \mathrm{P}\{\bigcup_{\{n_0,n_1,n_2\}\subseteq\Phi} T(n_0, n_1, n_2)\} \\ &= \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(\tau_0, n_1, n_2)\} + p_{2d}^{\mathrm{sec}}(\lambda) \end{aligned}$$

où

$$p_{2d}^{\mathrm{sec}}(\lambda) = \mathrm{P}\{\bigcup_{\{n_{i1},\cdots,n_{i5}\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(n_{i1}, n_{i2}, n_{i3}) \mid T'(\tau_0, n_{i4}, n_{i5})\}$$

désigne la probabilité que le nœud τ_0 ne puisse contribuer à un triangle qui délimite un trou triangulaire qui contient l'origine, mais trois autres nœuds peuvent former un triangle qui délimite un trou triangulaire qui contient l'origine.

Dans les parties suivantes, nous allons analyser cette probabilité dans trois cas différents.

Theorem 0.9. *Quand* $0 < \gamma \leq \sqrt{3}$, $p_{2d}(\lambda) = 0$.

Theorem 0.10. *Quand* $\sqrt{3} < \gamma \leq 2$, *on a* $p_{2dl}(\lambda) < p_{2d}(\lambda) < p_{2du}(\lambda)$, *où*

$$\begin{aligned} p_{2dl}(\lambda) =& 2\pi\lambda^2 \int_{R_s}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1 \end{aligned} \tag{2}$$

et

$$\begin{aligned} p_{2du}(\lambda) =& 2\pi\lambda^2 \int_{R_s}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1 + p_{2d}^{\mathrm{sec}}(\lambda) \end{aligned}$$

et

$$
\begin{aligned}
&\varphi_l(r_0) = 2\arccos(R_c/(2r_0))\\
&\varphi_u(r_0) = 2\arcsin(R_c/(2r_0)) - 2\arccos(R_c/(2r_0))\\
&R_1(r_0,\varphi_1) = \min(\sqrt{R_c^2 - r_0^2\sin^2\varphi_1} - r_0\cos\varphi_1,\\
&\qquad\qquad \sqrt{R_c^2 - r_0^2\sin^2(\varphi_1+\varphi_l(r_0))} + r_0\cos(\varphi_1+\varphi_l(r_0)))\\
&|S^+(r_0,\varphi_1)| = \int_{\varphi_l(r_0)}^{\varphi_1}\int_{r_0}^{R_1(r_0,\varphi)} r dr d\varphi\\
&|S^-(r_0,r_1,\theta_1)| = \int_{\theta_{2l}}^{-\varphi_l(r_0)}\int_{r_0}^{R_2(r_0,r_1,\theta_1,\theta_2)} r dr d\theta_2\\
&\theta_{2l} = \theta_1 - \arccos\frac{\cos(R_c/R) - \cos\theta_1\cos\theta_0}{\sin\theta_1\sin\theta_0}\\
&R_2(r_0,r_1,\theta_1,\theta_2) = \min(\sqrt{R_c^2 - r_0^2\sin^2\theta_2} - r_0\cos\theta_2,\\
&\qquad\qquad \sqrt{R_c^2 - r_1^2\sin^2(\theta_2-\theta_1)} + r_1\cos(\theta_2-\theta_1))
\end{aligned}
$$

$p_{2d}^{\mathrm{sec}}(\lambda)$ *est obtenu par des simulations.*

Theorem 0.11. *Quand* $\gamma > 2$, *on a* $p_{2dl}(\lambda) < p_{2d}(\lambda) < p_{2du}(\lambda)$, *où*

$$
\begin{aligned}
p_{2dl}(\lambda) =& 2\pi\lambda^2\Big\{\int_{R_s}^{R_c/2} r_0 dr_0 \int_0^{\pi} d\varphi_1 \int_{r_0}^{R_1'(r_0,\varphi_1)} e^{-\lambda\pi r_0^2}\\
&\times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1\\
&+ \int_{R_c/2}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2}\\
&\times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1\Big\}
\end{aligned}
$$

et

$$
\begin{aligned}
p_{2du}(\lambda) =& 2\pi\lambda^2\Big\{\int_{R_s}^{R_c/2} r_0 dr_0 \int_0^{\pi} d\varphi_1 \int_{r_0}^{R_1'(r_0,\varphi_1)} e^{-\lambda\pi r_0^2}\\
&\times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1\\
&+ \int_{R_c/2}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2}\\
&\times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1\Big\} + p_{2d}^{\mathrm{sec}}(\lambda)
\end{aligned}
$$

et

$$
R_1'(r_0,\varphi_1) = \min(\sqrt{R_c^2 - r_0^2\sin^2\varphi_1} - r_0\cos\varphi_1, \sqrt{R_c^2 - r_0^2\sin^2\varphi_1} + r_0\cos\varphi_1)
$$

$p_{2d}^{\text{sec}}(\lambda)$ *est obtenu par des simulations.*

0.3.4 L'évaluation des performances

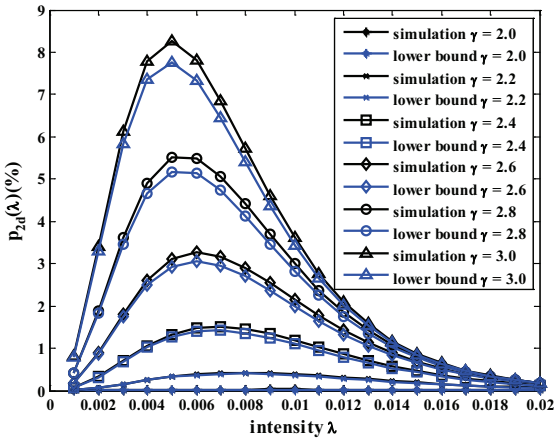
Dans les simulations, un disque centré à l'origine de rayon R_c est considéré. La probabilité que l'origine soit à l'intérieur d'un trou triangulaire est calculée. Les capteurs sont déployés de façon aléatoire dans le disque selon un processus de Poisson homogène d'intensité λ. Le rayon de détection R_s de chaque nœud est 10 mètres et γ est choisi entre 2 et 3 avec un intervalle de 0.2. Ainsi, le rayon de communication R_c varie de 20 à 30 mètres avec un intervalle de 2 mètres. λ est sélectionnée de 0.001 à 0.020 avec un intervalle de 0.001. Pour chaque γ, 10^7 simulations sont exécutés sous chaque λ pour vérifier si l'origine est à l'intérieur d'un trou triangulaire.

La probabilité $p_{2d}(\lambda)$ obtenue par des simulations est présentée avec la borne inférieure et la borne supérieure dans la Figure 6(a) et 6(b) respectivement. Les résultats de simulation pour $p_{2d}^{\text{sec}}(\lambda)$ sont présentés dans la Figure 6(c).

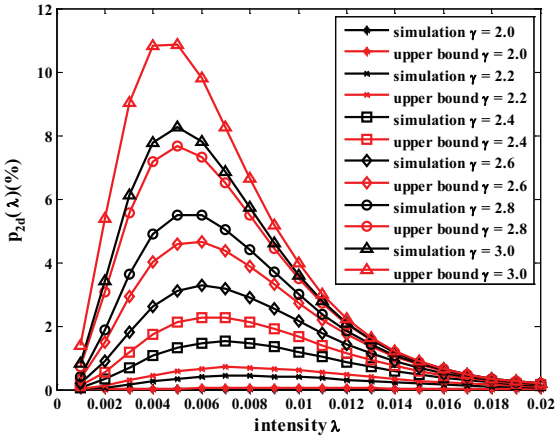
On peut voir que pour une valeur de γ, $p_{2d}(\lambda)$ présente un maximum à une valeur de seuil λ_c de l'intensité. En fait, pour $\lambda \leq \lambda_c$, le nombre de nœuds est faible. Par conséquent, la probabilité que l'origine soit à l'intérieur d'un trou triangulaire est relativement faible aussi. Avec l'augmentation de λ, la connectivité entre les nœuds devient plus forte. Par conséquent, la probabilité que l'origine soit à l'intérieur d'un trou triangulaire augmente. Cependant, quand l'intensité atteint la valeur de seuil, l'origine est recouverte avec une probabilité maximale. $p_{2d}(\lambda)$ diminue pour $\lambda \geq \lambda_c$. Les simulations montrent également que λ_c diminue avec l'augmentation de γ.

D'autre part, on peut voir sur la Figure 6(a) et 6(b) que, pour une intensité fixe λ, $p_{2d}(\lambda)$ augmente avec les augmentations de γ. C'est parce que R_s fixé, plus R_c est important, plus haut la probabilité que chaque triangle contienne un trou de couverture est importante.

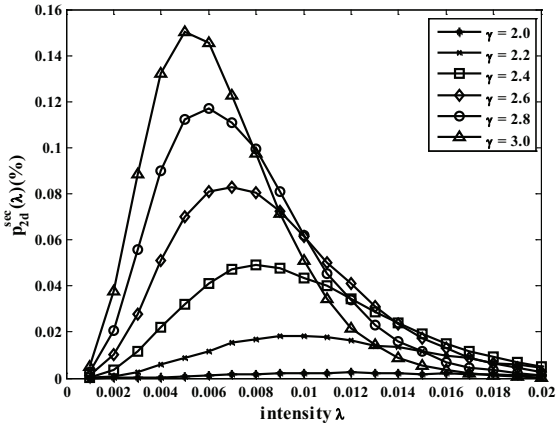
Enfin, on peut voir sur la Figure 6(a) que la probabilité obtenue par des simulations est conforme à la borne inférieure. La différence maximale entre eux est d'environ 0.5%. La Figure 6(b) montre que la probabilité obtenue par simulations est également compatible avec la borne supérieure. La différence maximale est d'environ 3%.



(a)



(b)



(c)

Figure 6: La proportion de la superface de trous triangulaires (a) les résultats de simulation et des bornes inférieures, (b) les résultats de simulation et des borne supérieures, (c) les résultats de simulation pour $p_{2d}^{\text{sec}}(\lambda)$

0.4 Précision de la détection de trou de couverture basée sur l'homologie sur la sphère

Dans cette section, nous étendons l'analyse sur la précision de la détection de trou de couverture basée sur l'homologie aux réseaux de capteurs sans fil sur la sphère. L'idée est la même que celle pour l'analyse de la précision dans le plan. La différence est que un trou dans un complexe de Čech manqué par un complexe de Rips doit être délimitée par un triangle sphérique. Nous définissons ainsi ces trous comme des trous triangulaires sphériques. On choisit la proportion de la surface de trous triangulaires sphériques comme une métrique pour évaluer la précision de la détection de trou de couverture basée sur l'homologie sur la sphère. Puisque le calcul est similaire, nous donnons simplement les résultats ici.

Supposons les capteurs déployés aléatoirement sur une sphère de rayon R selon un processus de Poisson homogène d'intensité λ. Tous les capteurs ont le même rayon de détection R_s et de communication R_c, $R_s \ll R$, $R_c \ll R$.

Soit $p_s(\lambda)$ désigne la proportion de la surface de trous triangulaires sphériques, nous avons les résultats suivants.

Theorem 0.12. *Quand* $0 < R_c \leq R \arccos([3\cos^2(R_s/R) - 1]/2)$, $p_s(\lambda) = 0$.

Theorem 0.13. *Quand* $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$, *on a* $p_{sl}(\lambda) < p_s(\lambda) < p_{su}(\lambda)$, *où*

$$\begin{aligned} p_{sl}(\lambda) = 2\pi\lambda^2 R^4 \int_{R_s/R}^{\theta_{0u}} \sin\theta_0 d\theta_0 \int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)} d\varphi_{s1} \int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})} \sin\theta_1 \\ \times e^{-\lambda|C(N,R\theta_0)|} e^{-\lambda|S^+(\theta_0,\varphi_{s1})|} (1 - e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|}) d\theta_1 \end{aligned} \tag{3}$$

et

$$\begin{aligned} p_{su}(\lambda) = 2\pi\lambda^2 R^4 \int_{R_s/R}^{\theta_{0u}} \sin\theta_0 d\theta_0 \int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)} d\varphi_{s1} \int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})} \sin\theta_1 \\ \times e^{-\lambda|C(N,R\theta_0)|} e^{-\lambda|S^+(\theta_0,\varphi_{s1})|} (1 - e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|}) d\theta_1 + p_s^{\text{sec}}(\lambda) \end{aligned}$$

et

$$\begin{aligned} \theta_{0u} &= \arccos\sqrt{[1 + 2\cos(R_c/R)]/3} \\ \varphi_m(\theta_0) &= \arccos[(\cos(R_c/R) - \cos^2\theta_0)/\sin^2\theta_0] \\ \theta_{1u}(\theta_0, \varphi_{s1}) &= \min\{\theta_{1u1}(\theta_0, \varphi_{s1}), \theta_{1u2}(\theta_0, \varphi_{s1})\} \end{aligned}$$

$$\theta_{1u1}(\theta_0,\varphi_{s1}) = \arccos\Big[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2\varphi_{s1}}\Big] + \arctan(\cos\varphi_{s1}\tan\theta_0)$$

$$\theta_{1u2}(\theta_0,\varphi_{s1}) = \arccos\Big[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2(\varphi_{s1}-\varphi_m(\theta_0))}\Big] + \arctan(\cos(\varphi_{s1}-\varphi_m(\theta_0))\tan\theta_0)$$

$$|C(N,R\theta_0)| = 2\pi R^2(1-\cos\theta_0)$$

$$|S^+(\theta_0,\varphi_{s1})| = \int_{2\pi-\varphi_m(\theta_0)}^{\varphi_{s1}}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi)} R^2\sin\theta d\theta d\varphi$$

$$|S^-(\theta_0,\theta_1,\varphi_{s1})| = \int_{\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1})}^{\varphi_m(\theta_0)}\int_{\theta_0}^{\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)} R^2\sin\theta_2 d\theta_2 d\varphi_2$$

$$\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1}) = \varphi_{s1} - \arccos\frac{\cos(R_c/R)-\cos\theta_1\cos\theta_0}{\sin\theta_1\sin\theta_0}$$

$$\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2) = \min\{\theta_{1u1}(\theta_0,\varphi_2),\theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)\}$$

$$\theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2) = \arccos\Big[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2(\varphi_2-\varphi_{s1})}\Big] + \arctan(\cos(\varphi_2-\varphi_{s1})\tan\theta_1)$$

$p_s^{\rm sec}(\lambda)$ *est obtenu par des simulations.*

Theorem 0.14. *Quand $R_c > 2R_s$, on a $p_{sl}(\lambda) < p_s(\lambda) < p_{su}(\lambda)$, où*

$$\begin{aligned} p_l(\lambda) = 2\pi\lambda^2R^4\Big\{ &\int_{\frac{R_s}{R}}^{\frac{R_c}{2R}}\sin\theta_0 d\theta_0\int_{\pi}^{2\pi}d\varphi_{s1}\int_{\theta_0}^{\theta'_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 \\ &\times e^{-\lambda|C(N,R\theta_0)|}e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})d\theta_1 \\ &+\int_{R_c/2R}^{\theta_{0u}}\sin\theta_0 d\theta_0\int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)}d\varphi_{s1}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 e^{-\lambda|C(N,R\theta_0)|} \\ &\times e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})d\theta_1\Big\} \end{aligned}$$

et

$$\begin{aligned} p_u(\lambda) = 2\pi\lambda^2R^4\Big\{ &\int_{\frac{R_s}{R}}^{\frac{R_c}{2R}}\sin\theta_0 d\theta_0\int_{\pi}^{2\pi}d\varphi_{s1}\int_{\theta_0}^{\theta'_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 \\ &\times e^{-\lambda|C(N,R\theta_0)|}e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|})d\theta_1 \\ &+\int_{R_c/2R}^{\theta_{0u}}\sin\theta_0 d\theta_0\int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)}d\varphi_{s1}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 e^{-\lambda|C(N,R\theta_0)|} \\ &\times e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|})d\theta_1\Big\} + p_s^{\rm sec}(\lambda) \end{aligned}$$

et

$$\theta'_{1u}(\theta_0, \varphi_{s1}) = \min\{\theta_{1u1}(\theta_0, \varphi_{s1}), \theta'_{1u2}(\theta_0, \varphi_{s1})\}$$
$$\theta'_{1u2}(\theta_0, \varphi_{s1}) = \arccos\left[\cos(R_c/R)/\sqrt{1 - \sin^2\theta_0 \sin^2\varphi_{s1}}\right] - \arctan(\cos\varphi_{s1}\tan\theta_0)$$

$p_s^{\text{sec}}(\lambda)$ *est obtenu par des simulations.*

Theorem 0.15.

$$\lim_{R\to\infty} p_{sl}(\lambda) = p_{2dl}(\lambda)$$

où $p_{sl}(\lambda)$ *et* $p_{2dl}(\lambda)$ *sont présentés dans* (2) *et* (3).

0.4.1 L'évaluation des performances

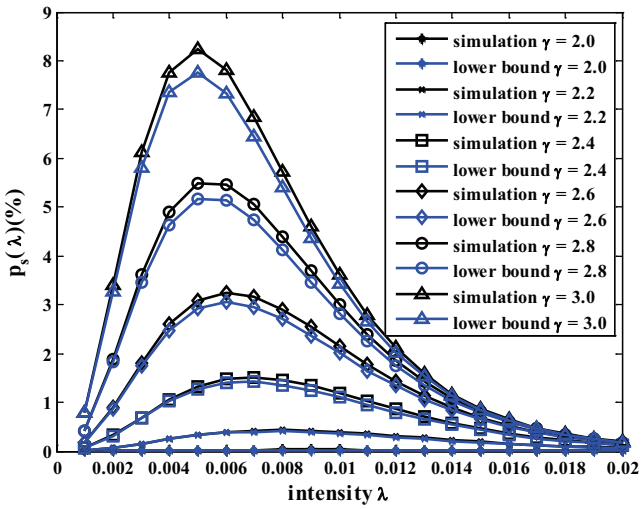
Une sphère centrée à l'origine de rayon R est considérée dans les simulations. La probabilité que le point de coordonnées sphériques $(R, 0, 0)$ soit à l'intérieur d'un trou triangulaire sphérique est calculée. Les capteurs sont déployés aléatoirement sur la sphère selon un processus de Poisson homogène d'intensité λ. Le rayon de détection R_s de chaque nœud est 10 mètres et le rayon de communication R_c est choisi de 20 à 30 mètres avec un intervalle de 2 mètres. Soit $\gamma = R_c/R_s$, alors γ varie de 2 à 3 avec un intervalle de 0.2. En outre, λ est sélectionnée de 0.001 à 0.020 avec un intervalle de 0.001. Pour chaque paire de (λ, γ), 10^7 simulations sont exécutées pour vérifier si le point de coordonnées sphériques $(R, 0, 0)$ est à l'intérieur d'un trou triangulaire sphérique.

0.4.1.1 Impact de R_s et R_c

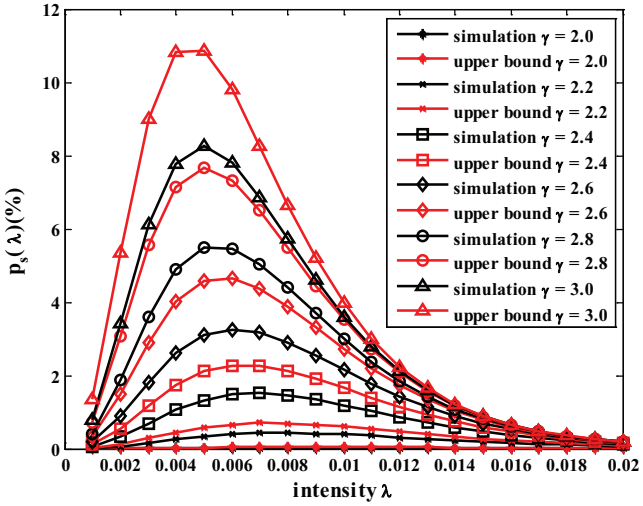
Puisque l'on suppose $R_s \ll R$ et $R_c \ll R$, on choisit $R = 10R_s$ pour analyser l'impact de R_s et R_c sur la probabilité que n'importe quel point soit à l'intérieur d'un trou triangulaire sphérique. Dans cette configuration, la probabilité $p_s(\lambda)$ obtenue par des simulations est présentée avec les bornes inférieure et supérieure dans la Figure 7(a) et 7(b) respectivement. On peut voir que ces résultats sont similaires à ceux de la section 0.3.4.

0.4.1.2 Impact de R

Pour mieux comprendre l'impact de R sur la probabilité que n'importe quel point est à l'intérieur d'un trou triangulaire sphérique, on choisit R à $5R_s, 10R_s$ et $100R_s$. En outre, nous voulons aussi connaître la différence de la probabilité sous les cas



(a)

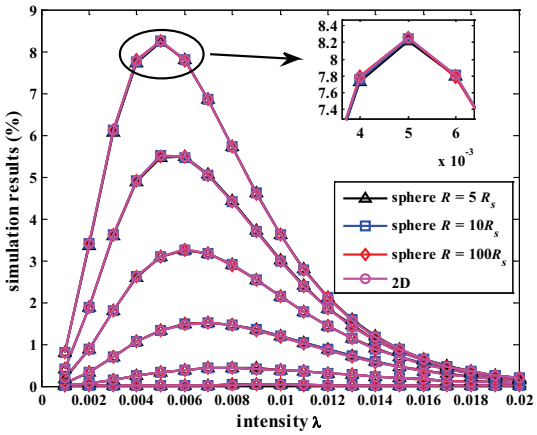


(b)

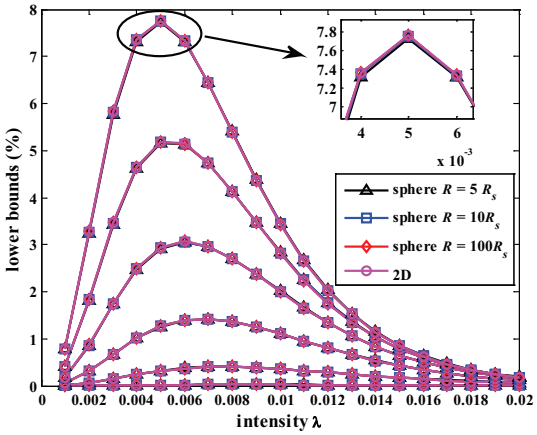
Figure 7: La proportion de la surface de trous triangulaires sphériques sous $R = 10R_s$ (a) les résultats de simulation et des bornes inférieures, (b) les résultats de simulation et des bornes supérieures

sphérique et plan. Par conséquent, les résultats de simulation, des bornes inférieures et supérieures de la probabilité pour les sphères avec des rayons $5R_s, 10R_s, 100R_s$ et le plan sont présentés dans la Figure 8(a), 8(b) et 8(c) respectivement. Les résultats de simulation pour $p_s^{\text{sec}}(\lambda)$ les sphères avec des rayons $5R_s, 10R_s, 100R_s$ et le plan sont présentés dans la Figure 9.

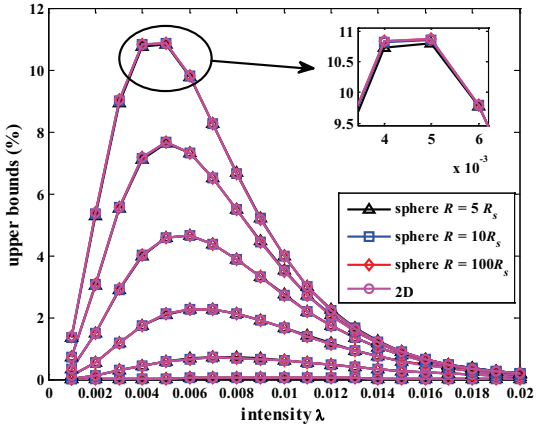
On peut voir sur la Figure 8 que les résultats de simulations, bornes inférieures et supérieures pour les sphères avec des rayons $5R_s, 10R_s, 100R_s$ et le plan sont très proches les uns avec les autres. Il se trouve plus que dans les cas ci-dessus,



(a)



(b)



(c)

Figure 8: La comparaison de la proportion de la surface de trous triangulaires sphériques (a) la comparaison des résultats de simulation, (b) la comparaison des bornes inférieures, (c) la comparaison des bornes supérieures

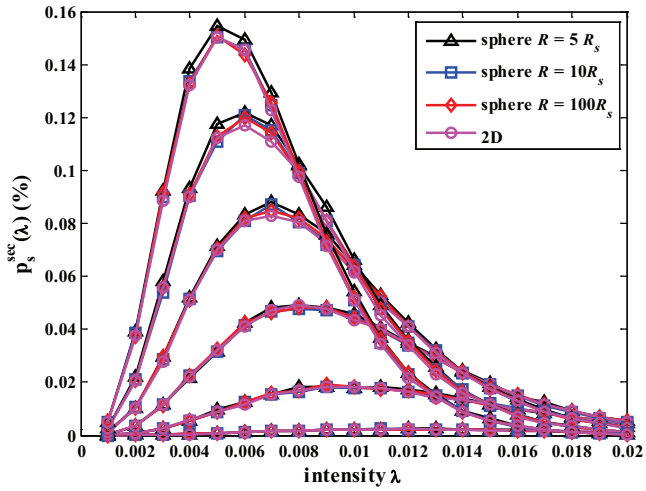


Figure 9: Les résultats de simulation pour $p_s^{\text{sec}}(\lambda)$

les différences maximales entre les résultats de simulation, les bornes inférieures et supérieures sont si petites qu'elles peuvent être négligées. Par conséquent, cela signifie aussi que le rayon de la sphère a peu d'impact sur la probabilité que n'importe quel point sur la sphère soit à l'intérieur d'un trou triangulaire sphérique quand il est beaucoup plus grand que les rayons de communication et de détection.

0.5 La détection de trou de couverture basée sur les graphs

Dans cette section, on présente un algorithme basé sur les graphes pour trouver cycles frontières de trous de couverture non-triangulaires dans un réseau de capteurs. Dans cet algorithme, on conçoit une méthode pour détecter les nœuds frontières en vérifiant s'il existe un cycle Hamiltonien dans leurs graphes voisins. Après cela, certains nœuds sont choisis au hasard pour lancer le processus de recherche des cycles frontières.

0.5.1 Modèles et hypothèses

On considère un ensemble de capteurs fixes déployés sur un domaine ciblé. Comme d'usage, la propagation radio est supposée isotrope. Chaque capteur contrôle une région à l'intérieur d'un cercle de rayon R_s et peut communiquer avec d'autres capteurs dans un cercle de rayon R_c. En outre, les hypothèses sont posées :

1. Il existe des capteurs situés sur la frontière externe du domaine cible. Ils sont nommés capteurs de frontière et d'autres capteurs sont désignés comme des capteurs internes. Chaque capteur de frontière a deux voisins de clôture.

2. Bien que les capteurs ne soient pas conscients de leurs emplacements, chaque capteur peut savoir s'il s'agit d'un nœud de frontière ou d'un nœud interne en utilisant les mécanismes présentés dans [44] ou [45].

3. Les capteurs internes sont déployés dans le domaine ciblé plan en fonction d'un processus de Poisson homogène d'intensité λ.

4. Chaque capteur a une identité unique. Le réseau a une seule composante connexe.

A partir sur ces hypothèses, nous pouvons utiliser un graphe $G(V, E)$ pour représenter le réseau de capteurs sans fil, où V représente tous les nœuds du réseau de capteurs sans fil. Pour deux nœuds, si ils peuvent communiquer avec l'autre, une arête les connecte sur le graphe. Pour tout nœud $v_i \in V$, son graphe de voisinage est le sous-graphe de $G(V, E)$ induit par tous ses voisins.

0.5.2 Un algorithme distribué basé sur les graphes

Le processus de notre algorithme peut être résumé comme suit.

1. Chaque capteur obtient l'information de ses 1- et 2-saut nœuds voisins et construit un graphe de voisinage. Ensuite, le complexe de Rips du réseau de capteurs peut être construit, comme le montre dans la Figure 10(a).

2. Basé sur le graphe de voisinage, chaque nœud peut déterminer si'l peut être un nœud frontière ou non en vérifiant s'il existe un cycle Hamiltonien dans son graphe de voisinage. S'il existe, le nœud est considéré comme un nœud non-frontière. Dans le cas contraire, il est supposé être un nœud frontière. Les résultats sont montrés dans la Figure 10(b∼c).

3. Lorsque chaque nœud a déterminé s'il s'agit d'un nœud frontière ou pas, il peut diffuser cette information à ses voisins. Ensuite, chaque nœud connaît l'état de ses voisins. Après cela, certains nœuds peuvent initier le processus pour trouver des cycles frontières, comme les nœuds désignés par diamant vert dans la Figure 10(d∼e).

4. Il est possible que certains cycles trouvées à l'étape 3 ne soient pas minimaux pour les cycles liés au même trou. Il est donc nécessaire de réduire et faire des choix parmi tous les cycles trouvés. Le résultat est montré dans la Figure 10(f).

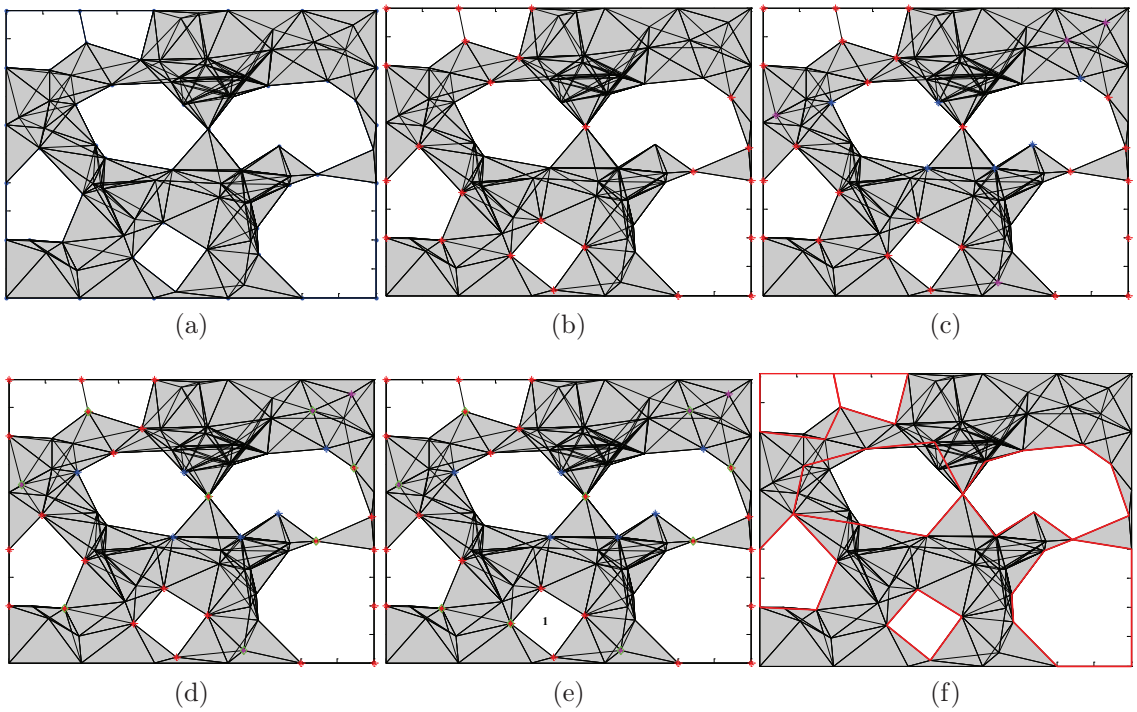


Figure 10: Les procédures du algorithme distribué basé sur les graphes. (a) le complexe de Rips d'un réseau de capteurs sans fil, (b∼c) découverte des nœuds limites, (d∼e) découverte des cycles limites, (f) sélection des cycles

0.6 La détection de trou de couverture basée sur l'homologie

Dans cette section, on présente un algorithme distribué efficace basé sur l'homologie pour détecter le trou de couverture. L'idée de base de cet algorithme est le suivant: pour le complexe de Rips d'un réseau de capteurs, on essaie de supprimer des sommets et des arêtes sans changer l'homologie tout en rendant le complexe de Rips moins dense et presque plan. Ensuite, il est facile de trouver les cycles frontières.

0.6.1 Modèles et définitions

Nous utilisons les mêmes modèles pour les nœuds comme dans la Section 0.5. En outre, nous avons besoin de donner quelques définitions qui seront utilisés dans le processus de cet algorithme.

On dit que un i-simplexe $[v_{i0}, v_{i1}, ..., v_{ii}]$ fait partie d'un j-simplexe $[v_{j0}, v_{j1}, ..., v_{jj}]$ si $[v_{i0}, v_{i1}, ..., v_{ii}] \subset [v_{j0}, v_{j1}, ..., v_{jj}]$. Ainsi, le sommet $[v_0]$ ou $[v_1]$ fait partie de l'arête $[v_0, v_1]$. L'arête $[v_0, v_1]$ fait partie du 2-simplexe $[v_0, v_1, v_2]$. En outre, $E(v)$ désigne tous les arêtes dont le nœud $[v]$ est une partie et $T(v)$ désigne tous les 2-simplexes dont le nœud $[v]$ est une partie.

Definition 0.10 (Indice d'un 2-simplexe)**.** *L'indice d'un 2-simplexe $\triangle$ est la plus grande dimension du simplexe dont le 2-simplexe fait partie, noté $I_\triangle$.*

Definition 0.11 (Poids d'un nœud)**.** *Le poids d'un nœud de clôture est 0. Pour chaque nœud interne v, s'il existe une arête dans $E(v)$ qui ne fait pas partie d'un 2-simplexe, Le poids w_v du nœud v est 0; sinon, le poids est l'indice minimum de tous les 2-simplexes dans $T(v)$, c'est-à-dire $w_v = \min_{\triangle \in T(v)} I_\triangle$.*

Le poids d'un nœud interne est un indicateur de la densité de ses nœuds voisins. Si le poids d'un nœud interne est 0, le nœud doit être sur le bord d'un trou de couverture. Plus le poids est élevé, plus la probabilité que le nœud ne soit pas sur le bord d'un trou de couverture est grande.

On utilise également la définition de graphe simplement connecté comme dans [46]. Soit G un graphe simple avec ensemble de sommets $V(G)$ et d'arêtes $E(G)$. Un cycle C est un sous-graphe de G s'il est connectée et chaque sommet de C est de degré deux. Le longueur d'un cycle C est le nombre de ses arêtes, noté $|E(C)|$. L'espace de cycle $C(G)$ d'un graphe G contient tous les cycles de G. L'ajout de deux cycles C_1 et C_2 est défini comme $C_1 \oplus C_2 = (E(C_1) \cup E(C_2)) \setminus (E(C_1) \cap E(C_2))$. Le sous-espace de cycle du triangle $C_T(G)$ de G est l'ensemble de tous les cycles 3-longueur dans $C(G)$.

Definition 0.12 (Graphe simple connectivité)**.** *Un graphe connexe G est simplement connecté si son espace de cycle $C(G)$ est vide, ou pour chaque cycle C dans $C(G)$, il existe un ensemble de cycles 3-longueur $T_0 \subseteq C_T(G)$ afin que $C = \sum_{T \in T_0} T$.*

Soit X un ensemble de sommets (ou d'arêtes) dans un graphe G, on utilise $G[X]$ pour désigner le sous-graphe induit de sommet (ou d'arête) par X. Les voisins d'un sommet v dans G est noté $N_G(v)$. Le graphe des voisins $\Gamma_G(v)$ d'un sommet v est

noté $G[N_G(v)]$. Le graphe des voisins $\Gamma_G(e)$ d'une arête $e = (u, v)$ est défini comme $G[N_G(u) \cap N_G(v) \cup \{u, v\}] - e$. L'ensemble de voisins d'un k-simplexe $[v_0, v_1, ..., v_k]$ est défini comme $\bigcap_{i=0}^{k} N_G(v_i)$.

Definition 0.13 (Suppression d'un k-simplexe dans un complexe de Rips)**.** *Un k-simplexe $[v_0, v_1, \cdots, v_k]$ est supprimé dans un complexe de Rips $\mathcal{R}(\mathcal{V})$ signifie que le simplexe et tous les simplexes dont le simplexe est une partie sont supprimés à partir de $\mathcal{R}(\mathcal{V})$.*

Basé sur les définitions ci-dessus, on peut donner la définition de la transformation HP (Homologie Préservée).

Definition 0.14 (La transformation HP)**.** *Une transformation HP est une combinaison séquentielle de suppressions de sommets (ou arête) définie comme suit: un sommet (ou une arête) x de G est supprimable si le graphe voisin $\Gamma_G(x)$ (1) a deux ou plusieurs sommets; (2) est connecté et (3) est un graph simple connectivité.*

Theorem 0.16. *Les transformations HP ne modifient pas le nombre de trous de couverture dans le complexe de Rips d'un réseau de capteurs sans fil.*

0.6.2 Un algorithme distribué basé sur l'homologie

L'algorithme contient cinq composants: le calcul de poids, suppression de sommets et d'arêtes, détection d'arêtes frontières, découverte de cycles frontières élémentaires et minimisation des cycles frontières, comme indiqué sur la figure 11.

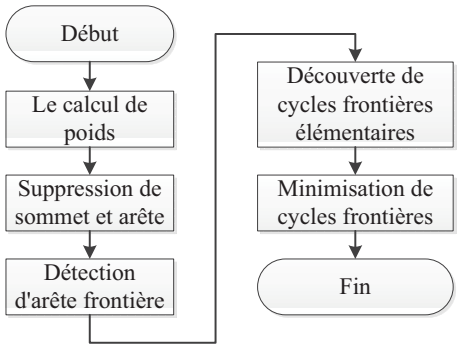


Figure 11: Diagramme de l'algorithme

Un exemple est utilisé pour illustrer les procédures de cet algorithme dans la Figure 12. Pour un réseau de capteurs sans fil avec quelques trous de couverture, le complexe de Rips du réseau de capteurs sans fil est d'abord construit, illustré à

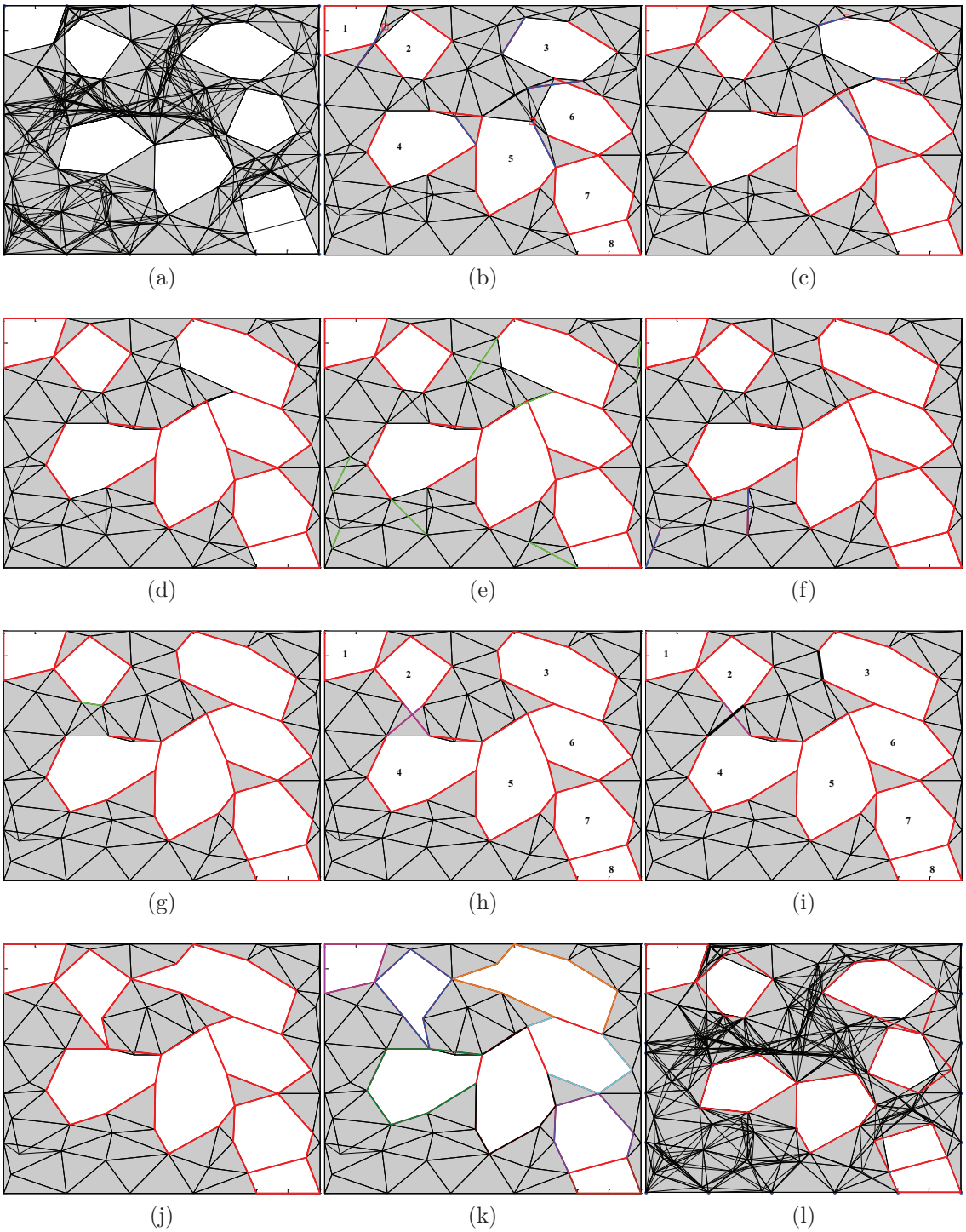


Figure 12: Les procédures de l'algorithme. (a) le complexe de Rips d'un réseau de capteurs, (b) après suppression de sommet, (c∼d) suppression d'arête, (e∼j) détection d'arête frontière, (k) découverte de cycles frontières élémentaires, (l) minimisation de cycles frontières

la Figure 12(a), alors l'algorithme vise a découvrir des cycles frontières minimales de tous les trous. Dans la phase de calcul de poids, chaque nœud calcule son poids indépendamment selon la définition 0.11. Après l'obtention du poids, chaque nœud détermine s'il peut être supprimé ou non selon certaines règles définies ci-après. Figure 12(b) montre le résultat de la suppression de sommet. En outre, un nœud particulier décide si une certaine arête peut être supprimée ou non. Figure 12(b∼c) montre le processus de cette suppression d'arête spéciale. Après le deuxième phase, de nombreuses arêtes frontières peuvent être trouvées, comme la ligne rouge illustrée à la Figure 12(d). Mais il est possible que d'autres arêtes frontières n'aient pas été trouvés. Ensuite, dans le troisième phase, tous ou presque toutes les arêtes frontières seront trouvées après la suppression de certains arêtes, voir la Figure 12(e∼j). Par la suite, les cycles frontières élémentaires peuvent être facilement découverts, comme le montre dans la Figure 12(k). Il est possible que les cycles frontières trouvés ne soient pas minimaux. Dans ce cas, les cycles frontières ils seront minimisés dans la dernière phase de l'algorithme comme indiqué sur la Figure 12(l).

0.6.2.1 Calcul de poids

Dans ce composant, chaque nœud calcule son poids. Le poids de nœud de clôture est 0. Pour tout nœud interne, théoriquement, il doit construire tous les simplexes dont il fait partie. Comme on considère les réseaux de capteurs sur un domaine ciblé plan, chaque nœud interne a juste besoin de construire tous ses 1-simplexes et 2-simplexes et leurs voisins. Pout tout nœud v, soit $E(v)$ l'ensemble de ses 1-simplexes et $T(v)$ l'ensemble de ses 2-simplexes. Pour tout $e \in E(v)$, soit $n(e)$ l'ensemble de voisins de e. Pour tout $t \in T(v)$, soit $n(t)$ l'ensemble de voisin de t. Ensuite, le poids de nœud v peut être calculé comme dans l'algorithme 1.

Algorithm 1 Calcul de poids (pour nœud interne v)

Begin
  if $\exists e \in E(v), n(e)$ est vide **then**
    $w_v = 0$
  else if $\exists t \in T(v), n(t)$ est vide **then**
    $w_v = 2$
  else
    $w_v = 3$
  end if
END

0.6.2.2 Suppression de sommet et arête

Dans ce composant, on effectue un nombre maximal de suppressions de sommets sans changer le nombre de trous de couverture dans le réseau de capteurs original et on supprime également des arêtes spéciales si ces arêtes existent. Pour la suppression de sommets, on ne considère que les nœuds internes, les nœuds de clôture ne seront jamais supprimés. Dans la partie suivante, on utilise sommet et nœud de manière interchangeable.

(1) Suppression de sommet

Si le poids d'un sommet est inférieur à 3, il ne doit jamais être supprimé. Sinon, le sommet continue à vérifier s'il est supprimable ou non en fonction de la transformation HP. Après la vérification, le sommet diffuse un message indiquant qu'il peut être supprimé ou non. Après avoir reçu le statut de tous ses voisins, chaque sommet supprimable continue à vérifier s'il doit être supprimé. On peut trouver que le poids de chaque sommet supprimable doit être 3. On suppose que le sommet avec une identité inférieure a la priorité doit être supprimé en premier. Donc, chaque sommet supprimable a juste besoin de vérifier si son identité est la plus basse parmi tous ses voisins supprimables. Si c'est le cas, il devrait être supprimé. Sinon, il ne doit pas être supprimé. L'algorithme 2 donne la procédure détaillée pour la suppression de sommet. La procédure de suppression de sommet se termine jusqu'à ce que aucun sommet ne puisse être supprimé dans le complexe de Rips. La Figure 12(b) donne le résultat initial après la suppression de sommet.

Algorithm 2 Suppression de sommet (pour nœud interne v)

Begin
  if $w_v < 3$ **then**
    nœud v ne peut pas supprimé
  else if nœud v n'est pas supprimable en fonction de la transformation HP **then**
    nœud v ne peut pas supprimé
  else if l'identité de nœud v est la plus bas parmi tous ses voisins supprimables. **then**
    nœud v est supprimé
  end if
END

(2) Suppression d'arête

Après la suppression de sommets, on trouve une chose intéressante. Les Arêtes n'ayant pas de voisins doivent être sur le bord de trous, comme l'arête commune de trous de couverture 7 et 8 dans la Figure 12(b). L'arête ayant un seul voisin se

trouve sur le bord d'un trou de couverture avec une grande probabilité, comme les arêtes rouges sur la Figure 12(b). Mais il est possible qu'il existe certaines de ces arêtes spéciales qui ne se trouvent pas sur le bord, comme l'arête bleu entre les trous de couverture 1 et 2 dans la Figure 12(b). Nous essayons de supprimer ces arêtes spéciales. Puisque l'arête a un seul voisin, la suppression de l'arête ne créera pas un nouveau trou.

On appelle arêtes frontière des arêtes ayant au plus un voisin. Ensuite, nous concevons une règle pour suppresser les arêtes frontières spéciales. Si un sommet v n'a qu'une seul arête frontière vx et vx a un seul voisin y et la suppression de vx ne fera pas v avoir une nouvelle arête frontière, alors vx peut être supprimé. Cela peut être facilement réalisé en vérifiant si vy a plus de deux voisins. La Figure 12(d) montre le résultat après la suppression d'arête.

0.6.2.3 Détection d'arête frontière

Après la suppression de sommets et d'arêtes, nous pouvons trouver que presque toutes les arêtes frontières se situent sur le bord de trous. On peut également voir que certaines arêtes situées sur le bord n'ont pas été trouvés. Dans ce composant, nous essayons de trouver les arêtes autant que possible. Dans tous les cas, ces arêtes ont deux ou plusieurs voisins. Si on définit les nœuds ayant une ou plusieurs arêtes frontières comme nœuds frontières et d'autres nœuds comme nœuds non-frontières, puis on essaie de supprimer certaines arêtes reliant les nœuds non-frontières et les nœuds frontières en fonction de la transformation HP, telles que les arêtes vertes illustrées sur la Figure 12(e). Après cela, il est possible que certaines nouvelles arêtes frontières soient reconnues et des arêtes spéciales illustrées dans la Section 0.6.2.2 sont également identifiés, comme les arêtes bleues présentées dans la Figure 12(f). Dans ce cas, on peut toujours utiliser la règle dans la Section 0.6.2.2 pour les supprimer. Il est encore possible que certaines arêtes se trouvant sur le bord n'aient pas été découvertes. Ce cas se produit généralement lorsque certains nœuds frontières sont des voisins et les arêtes qui les relient ont plus d'un voisin. Dans ce cas, on supprime aléatoirement certaines de ces arêtes en fonction de la transformation HP, comme les arêtes vertes sur la Figure 12(g). De cette façon, presque toutes les arêtes frontières peuvent être trouvées.

0.6.2.4 Découverte de cycles frontières élémentaires

Après la détection des arêtes frontières , il est facile de découvrir les cycles frontières élémentaires. Nous pouvons choisir au hasard des nœuds qui ont deux arêtes frontières comme initiateurs. Chaque nœud lance un processus pour trouver le cycle frontière élémentaire en envoyant un message le long de l'un de ses arêtes frontières. Quand il reçoit le message en retour le long de l'autre arête frontière, il découvre un cycle frontière élémentaire. De cette façon, tous les cycles frontières élémentaires peuvent être trouvés, comme les cycles indiqués par des couleurs différentes sur la Figure 12(k).

0.6.2.5 Minimisation de cycles frontières

Il est possible que certains cycles frontières élémentaires trouvés ne soient pas minimum, nous avons donc besoin de minimiser ces cycles. Ceci peut être réalisé en vérifiant s'il existe un chemin plus court entre deux nœuds quelconques dans le cycle. De cette façon, nous pouvons obtenir presque la plupart des cycles minimum entourant les trous de couverture.

0.6.3 L'évaluation des performances

On choisit une région carrée de 100×100 m^2 comme le domaine ciblé. Le rayon de détection R_s de chaque nœud est 10 mètres. Le rayon de communication R_c est 20 mètres et ainsi $\gamma = 2$. Il y a des capteurs de frontière le long des arêtes du carré avec 20 mètres de distance entre voisins. D'autres capteurs internes sont déployés de façon aléatoire dans la région basée sur un processus Poisson homogène d'intensité λ.

0.6.3.1 Complexité

La complexité de calcul de chaque étape de l'algorithme est illustré dans Table 1.

Table 1: La complexité de chaque étape de l'algorithme

Etape	Complexité
Le calcul de poids	$O(n^2)$
Suppression de sommet et arête	$O(n^3)$
Détection d'arête frontière	$O(n^3)$
Découverte de cycles frontières élémentaires	$O(1)$
Minimisation de cycles frontières	$O(1)$

0.6.3.2 Comparaison avec l'algorithme basé sur localisation

Afin d'évaluer la précision de notre algorithme HBA, on le compare à l'algorithme basé sur localisation LBA proposée dans [47]. Soit λ 0.008, 0.010 et 0.012 respectivement. Pour chaque intensité, 1000 simulations sont effectuées. Les résultats de simulation montrent que lorsque λ est 0.008, il y a neuf fois parmi les 1000 fois où notre algorithme ne peut pas trouver tous les trous de couverture non-triangulaires. Dans chacun des neuf fois, un seul trou de couverture est manqué. Il y a 7363 trous non-triangulaires au total et 7354 ceux-ci trouvés par notre algorithme. Lorsque λ est 0.010 et 0.012, une seule fois parmi les 1000 fois où notre algorithme ne peut pas trouver tous les trous de couverture. Et à ce moment, un seul trou de couverture est manqué. Lorsque λ est 0.010, il y a 6114 trous non-triangulaires au total et 6113 ceux-ci trouvés par notre algorithme. Lorsque λ est 0.012, il y a 4613 trous non-triangulaires au total et 4612 ceux-ci trouvés. Les résultats sont présentés dans le Table 2. Tous ces résultats montrent que notre algorithme peut trouver environ 99% des trous de couverture dans environ 99% des cas.

Table 2: Comparaison avec l'algorithme basé sur localisation

λ	nombre de trous non-triangulaires trouvés par HBA	nombre de trous non-triangulaires manqués par HBA	nombre de trous non-triangulaires trouvés par LBA
0.008	7354	9	7363
0.010	6113	1	6114
0.012	4612	1	4613

Chapter 1

Introduction

1.1 Motivations

Recent advancements in wireless communications and Micro-Electro-Mechanical System (MEMS) have enabled the development of wireless sensor networks (WSNs) [1]. A WSN consists of a number of tiny sensor nodes each capable of sensing, data processing and communicating with neighbouring nodes. These sensor nodes are deployed in the target field to collectively monitor physical phenomena, such as heat, light, sound, pressure, motion. WSNs have lots of applications, which can be generally classified into two categories: tracking and monitoring [2]. Tracking applications include tracking humans, animals and vehicles [3, 4, 5, 6, 7, 8, 9]. Monitoring applications include habitat and environmental monitoring [10, 11, 12, 13, 14, 15, 16, 17], structural health monitoring [18, 19, 20, 21, 22], health-care monitoring [23, 24, 25, 26] and traffic monitoring [27, 28, 29, 30, 31, 32].

Coverage is a fundamental problem in WSNs. It can be considered as the measure of quality of service of a sensor network [33]. Based on the coverage objectives, the authors in [34] classified coverage into three types: area coverage, point coverage and barrier coverage. As for the area coverage problem, the main objective of the sensor network is to cover an area. While in the point coverage problem, the objective is to cover a set of points. The aim of barrier coverage is to minimize the probability of undetected penetration through the barrier formed by wireless sensor networks. In this thesis, we focus on area coverage problem.

In applications related to area coverage, the target field is usually required to be fully covered. However, coverage holes may be formed due to many reasons, such as random deployment, energy depletion or destruction of sensors. For example, in the volcano monitoring applications, the target field is usually hostile or unapproachable

for human beings, under such situations, sensor nodes have to be deployed randomly by helicopters or other kinds of vehicles. With random deployment, sensor nodes may cluster at some place while leaving coverage holes at some other places. Furthermore, even if in the initial deployment, the target field is fully covered by sensor nodes. With time goes by, some nodes may deplete their battery power more quickly than others, which may cause a coverage hole. In addition, some nodes may be destroyed by a natural disaster. Therefore, it is of paramount importance to detect and localize coverage holes. On one hand, it can facilitate the full coverage of target field. After coverage holes are localized, some moving nodes can de deployed in the coverage hole areas to patch them [35, 36]. On the other hand, knowing coverage hole boundaries is also very helpful in the design of basic networking functionalities, such as point-to-point routing and data gathering mechanisms [37].

Many approaches have been proposed for coverage hole detection in WSNs. They can be generally classified into three categories: (i) location based approaches, (ii) range based approaches, and (iii) connectivity based approaches. Location based and range based approaches can discover all coverage holes with good accuracy but require either precise location information or accurate distance information, which is difficult to obtain in many scenarios. So connectivity based approaches received considerable attention in recent years. Particularly, homology based approaches attracted our attention. Ghrist and his collaborators introduced two combinatorial tools, Čech complex and Vietoris-Rips complex (we abbreviate it to Rips complex in this thesis), to detect coverage holes [38]. They proposed a centralized solution for some simple scenarios, but designing an efficient distributed algorithm for coverage hole detection with only connectivity information is still an open issue.

Realizing the limitations of existing work for coverage hole detection in WSNs, we try to further analyse the usefulness of homology theory for coverage hole detection and design some distributed algorithms to detect coverage holes in WSNs.

1.2 Objectives and contributions

1.2.1 Objectives

The main objectives are to design algorithms for coverage hole detection in WSNs, which have the following properties.

- *Connectivity based.* Location based and range based approaches require either precise location information of nodes or accurate distance information between

neighbouring nodes. Such requirement is either expensive or impractical in many scenarios, which substantially limits their applicability. So we aim to design connectivity based approaches.

- *Distributed.* Usually, there is no central unit in a WSN to control all the nodes. So it is required to design distributed algorithms, especially for large scale sensor networks.

- *Efficient.* Since sensor nodes have only limited capabilities, the complexity of the proposed algorithms should be low as much as possible so that sensor nodes can implement them efficiently, which can extend the lifetime of the sensor networks.

- *Can discover boundary cycles.* A boundary cycle of a coverage hole is a cycle connecting all the nodes surrounding the coverage hole. After discovering boundary cycles of coverage holes, we can deploy more nodes in these regions to cover them. It is thus important to detect boundary cycles. Many of the above presented approaches only detect boundary nodes, which may be not sufficient to discover the boundary cycles of coverage holes. For location based approaches, it is possible to find exact boundary cycles after detecting boundary nodes, as presented in [47]. But for range based approaches, detecting boundary nodes is not sufficient to discover the exact boundary cycles in some cases, which will be illustrated in Section 2.1.4. Therefore, we aim to design algorithms which can discover boundary cycles.

1.2.2 Contributions

Our contributions are twofold: (1) accuracy of homology based coverage hole detection and (2) distributed algorithms for coverage hole detection. These works have been published in some conferences or submitted to some journals, as summarized in Table 1.1.

1.2.2.1 Accuracy of homology based coverage hole detection

Čech complex can fully capture coverage holes but it is difficult to construct. Rips complex is easy to construct but may miss some holes. We first analysed the relationship between Čech complex and Rips complex in terms of coverage holes for WSNs on a planar target field. Then we chose the proportion of the area of holes missed by Rips complex as a metric to evaluate the accuracy of homology based

Table 1.1: Summary of publications

Titles	Contributions
IEEE GLOBECOM 2011	Graph based algorithm, see Chapter 5
IEEE ICC 2012	Accuracy of homology based coverage hole detection on plane, see Chapter 3
IEEE Transactions on Wireless Communications	Accuracy of homology based coverage hole detection on sphere, see Chapter 4
IEEE/ACM Transactions on Networking	Accuracy of homology based coverage hole detection on plane and homology based algorithm, see Chapter 3 and 6

coverage hole detection. We showed that such proportion is related to the ratio between communication and sensing radii of each sensor (denoted respectively by R_c and R_s). We then analysed this proportion in three cases and for each case, closed form expressions for lower and upper bounds were derived. Simulations results are well consistent with the analytical lower and upper bounds, with maximum differences of 0.5% and 3%. This part of work has been published in ICC 2012 [48].

In addition, we extended the analysis to the sphere case. Simulation results show that the radius of sphere has little impact on the proportion when it is much larger than communication and sensing radii of each sensor. This part of work has been submitted to IEEE Transactions on Wireless Communications [49].

1.2.2.2 Distributed algorithms for coverage hole detection

We first proposed a graph based distributed algorithm to detect coverage holes. The algorithm consists of four steps: neighbour discovery, boundary nodes discovery, boundary cycles discovery, cycles selection. In the step of neighbour discovery, each node obtains all its 1- and 2-hop neighbours information. In boundary nodes discovery step, each node determines whether it is a boundary node or not by checking the existence of a Hamiltonian cycle in its neighbour graph. If there exists, it is considered to be a non-boundary node. Otherwise, it is assumed to be a boundary node. After that, some boundary nodes are randomly selected to initiate the boundary cycles discovery process and many cycles will be found. But these found cycles may not be minimum or some of them bound the same coverage holes. So it is required to make choices among these cycles, which is done in the final step. This part of work has been published in GLOBECOM 2011 [50].

The above proposed algorithm exhibits high complexity. So we designed a more efficient homology based algorithm. The basic idea of this algorithm is that for the

Rips complex of a WSN, we try to delete some vertices and edges without changing the homology while making the Rips complex more sparse and nearly planar. Then it is easier to find boundary cycles. The novelty of this algorithm lies in the rule we proposed to decide for each vertex or edge whether its deletion can change the homology or not. This algorithm has the worst case complexity $O(n^3)$ where n is the maximum number of 1-hop neighbour nodes and n is finite. It is distributed and requires only 1- and 2-hop neighbour nodes information. It can accurately detect the boundary cycles of about 99% coverage holes in about 99% cases. This part of work has been submitted to IEEE/ACM transactions on networking [51].

1.3 Organization

The rest of this thesis is organised as follows. Chapter 2 gives a survey about the coverage hole detection approaches and introduces homology theory briefly. Chapter 3 analyses the accuracy of homology based coverage hole detection for WSNs on a planar target field. It first identifies the relation between Čech complex and Rips complex on plane in terms of coverage holes. Then the lower and upper bounds of the accuracy are derived. Chapter 4 extends the ideas in Chapter 3 to WSNs on the sphere. Chapter 5 introduces a graph based distributed coverage hole detection algorithm. Chapter 6 further presents an efficient homology based distributed algorithm for coverage hole detection. Finally, Chapter 7 provides concluding remarks and some possible future research directions.

Chapter 2

Related Work and Mathematical Background

Many approaches have been proposed for coverage hole detection in WSNs. They can be generally classified into three categories: (1)location based approaches; (2) range based approaches and (3) connectivity based approaches. The connectivity based approaches can be further classified into graph based and homology based according to the tools they adopt. The former two types of approaches attract less attention since they require either precise location information of nodes or accurate distance information between neighbouring nodes which is very difficult to obtain in many scenarios. Connectivity based approaches attract a great deal of attention as they only need connectivity information which is easy to obtain. In this category, homology based approaches attract particular attention due to its powerful tools for coverage hole detection, which is also our interest.

In this chapter, we consider location based, range based and graph based approaches as traditional approaches and first give a survey about them and describe their pros and cons in Section 2.1. Then we give a brief introduction to homology theory before presenting homology based approaches in Section 2.2.

2.1 Traditional approaches

2.1.1 Location based approaches

The location based approaches can further be classified into two cases according to the boundary node detection methods they adopted.

2.1.1.1 Polygon based approaches

In [35, 52, 53, 54], the Voronoi diagram was used to detect boundary nodes. The Voronoi diagram of a collection of nodes partitions the space into polygons called Voronoi polygons. Every point in a given polygon is closer to the node in this polygon than to any other node. So if some portion of a Voronoi polygon is not covered by the node inside this polygon, it will not be covered by any other node, which implies a coverage hole. However, it is known from computational geometry that the Voronoi polygon of boundary nodes can not be locally computed in general [55]. Realising such a problem, the authors in [56, 57] proposed to use localized Voronoi polygons for boundary node detection. In the scheme, each node constructs its localized Voronoi polygon. If its localized Voronoi polygon is infinite or it is finite but with some vertices uncovered by the node, then the node must be a boundary node.

2.1.1.2 Perimeter based approaches

Different from polygon based approaches, perimeter based approaches detect boundary nodes by checking whether the perimeter of the node's sensing disk is covered by its neighbours or not. In [58], it is proved that a sensor node does not border a coverage hole if its sensing border is entirely covered by the sensing ranges of its neighbours. Another boundary node detection approach proposed in [59, 60] simplifies the previous border checking approach by only checking intersection points on the sensing border. A point is called an intersection point between nodes u and v if it is an intersection point of the sensing borders of u and v. A node is a boundary node if and only if there exists at least one intersection point which is not covered by any other neighbours. Based on that criterion, some other algorithms were proposed in [47, 61] to discover boundary nodes. Furthermore, the authors also proposed a distributed algorithm to discover the exact boundary cycles of coverage holes in [47].

2.1.2 Range based approaches

The range based approaches attempt to identify boundary nodes based on relative distance between neighbouring nodes. They also follow the ideas of either polygon based or perimeter based approaches. In [62, 63], a localized Voronoi polygon based boundary node detection algorithm was proposed, which is similar as that in [56, 57]. The difference lies in that the localized Voronoi polygon is constructed using location

information of nodes in [56, 57] while it is constructed based on directional and distance information between neighbouring nodes in [62, 63].

In [44], the author followed the idea of perimeter based approaches. He proposed a coverage verification algorithm based on distances between neighbouring nodes. In the algorithm, each node first calculates a set of segments. Each segment is a part of its sensing border that is covered by one of its intersecting neighbours. After that, the node verifies whether its entire sensing border is covered by the set of segments. If yes, it implies the node is not a boundary node. Otherwise, it is a boundary node. Recently, in [64], the same author proposed another simpler algorithm for coverage verification. He assumed that the transmission radius of each sensor is at least four times larger than its sensing radius. The assumption implies that for any node, every pair of its intersecting neighbours are also neighbours of each other. The algorithm uses this property to determine the relative locations of all the intersecting neighbours of any node and uses it to verify coverage. But if the assumption is not satisfied, the algorithm will not work. In addition, a more general scheme was proposed in [65] for verifying k-coverage of a d-dimensional target field for an arbitrary positive integer k and $d \in \{1, 2, 3\}$. The scheme transforms the k-coverage verification problem in d dimension to a number of coverage verification problems in $(d-1)$ dimension by dimension reduction technique. It then uses the same ideas as that in [44, 64] to verify k-coverage in one dimension.

2.1.3 Graph based approaches

Graph based approaches usually detect coverage holes based on graph theory. In [66, 67], a set of active nodes is selected using some distance information. Then for each active node, it checks whether there exists a 3MeSH ring in its neighbour graph. A 3MeSH ring is defined as a closed polygon formed by nodes in neighbour graph, which can not be triangulated by these nodes. If there exists a 3MeSH ring, then the node is a non-boundary node, otherwise, it is a boundary node. After all boundary nodes are detected, boundary cycles are then discovered through signaling protocols. This approach is not purely based on connectivity information, it needs some distance information. In addition, it can detect holes with up to ten edges with proper complexity, but for holes with more than ten edges, the complexity will be higher.

More recently, some distributed algorithms were proposed to detect topological hole [68, 69] or to recognise boundary in sensor networks by using only connectivity

information [37, 45, 46, 70, 71]. These algorithms may be useful for coverage hole detection. In [68, 69], Funke et al proposed a simple distributed algorithm to identify nodes near the boundary of the target field and holes. The basic idea is to construct iso-contours based on hop count from a root node and identify where the contours are broken. Following their ideas, two boundary detection algorithms were proposed in [72] and [73]. In [70], Kröller et al proposed to recognise boundary by searching for some specific structures called flower and augmenting cycle. The success of their algorithm critically depends on the identification of at least one flower structure, which may not always be the case especially in a sparse network [37]. In [74], Saukh et al introduced the concept of patterns and proposed a fully distributed algorithm for boundary recognition based on that concepts. Their approach is applicable for both dense and sparse deployments. In [75], a distributed algorithm based solely on connectivity information was proposed to detect holes and boundaries. The basic idea is to use multidimensional scaling (MDS) [76] to compute virtual coordinates of nodes, and then use a geometrical method to detect boundary nodes. But the running time of MDS is high. In [37], Wang et al exploited a special structure called "cut" in the shortest path tree to detect boundary nodes and connect them to boundary cycles. As their basic objective is to identify the global topology of the underlying environment where sensors are deployed, some holes may be neglected by their algorithm. In order to identify holes with any size, Dong et al [46] proposed a fine grained boundary recognition approach. The basic idea is to first extract the skeleton of the sensor network, and then to identify primary boundary cycles, finally to minimize these boundary cycles. As their algorithm involves many communications in the whole network, the control overhead is significant.

Fekete et al [77] proposed a statistical method to detect boundary nodes based on the assumption that nodes on the boundaries have much less average degrees than nodes in the interior of the network. Another statistical approach was proposed in [78] by computing the restricted stress centrality of a node. Nodes in the interior tend to have a higher centrality than nodes on the boundary. Similarly, Li et al [79] exploited two centrality measures in graph theory, called betweenness and closeness, to identify boundary nodes. It is observed that boundary nodes usually have lower betweenness or higher closeness than their neighbouring nodes. These statistical approaches usually require very high node density.

In summary, these boundary recognition approaches suffer from one or more following disadvantages if used for coverage hole detection. (1) They only detect coarse boundary nodes and do not consider boundary cycles. (2) They can only

detect large coverage holes. (3) The complexity is high. (4) High density of nodes is required.

2.1.4 Remarks on these approaches

To summarize, Table 2.1 gives some typical traditional approaches and their characteristics and Table 2.2 presents the pros and cons of the traditional coverage hole detection approaches.

Table 2.1: Characteristics of typical traditional approaches for coverage hole detection

References	Category	Centralized /Distributed	Complexity	Discover boundary cycles
Fang et al [52]	Location based	Centralized	NA	No
Zhang et al [57]	Location based	Distributed	NA	No
Huang et al [58]	Location based	Distributed	$O(n \log n)$	No
Tong et al [47]	Location based	Distributed	NA	Yes
Zhang et al [62]	Range based	Distributed	NA	No
Bejerano [44]	Range based	Distributed	$O(n^3)$	No
Li et al [66]	Graph based	Distributed	$O(n^3)$	Yes
Wang et al[37]	Graph based	Distributed	NA	Yes
Dong et al[46]	Graph based	Distributed	NA	Yes

Table 2.2: Summary of traditional coverage hole detection approaches in WSNs

Approaches	pros	cons
Location based	accurately detect boundary nodes and boundary cycles	need precise location information
Range based	accurately detect boundary nodes, discover most boundary cycles	need accurate distance information, may falsely detect boundary cycles
Graph based	only need connectivity information	may miss coverage holes

Generally speaking, the performance of location based approaches and range based approaches is highly dependent on the accuracy of location or distance information. When accurate location information is available, location based approaches can not only detect all boundary nodes but also discover all boundary cycles. But

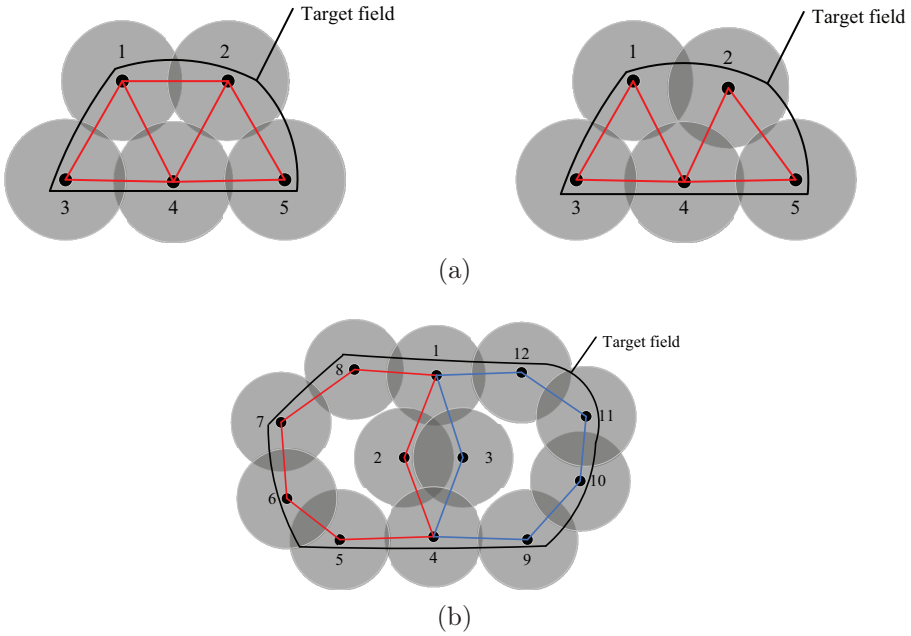


Figure 2.1: Examples illustrating that it is not sufficient to only detect boundary nodes in order to discover boundary cycles

in practice, such accurate information is often difficult to obtain, which restricts their applications. With precise distance information, range based approaches can accurately detect all coverage hole boundary nodes. But it may not be sufficient to find boundary cycles of coverage holes. For example, in Figure 2.1(a), it can be seen that all the nodes in the two WSNs are boundary nodes, but there are three coverage holes in the left WSN and only two coverage holes in the right WSN. It is also possible to associate a node to a wrong hole. Consider another example in Figure 2.1(b), the correct boundary cycles for the two coverage holes should be cycles formed by nodes 1, 2, 4, 5, 6, 7, 8 and by nodes 1, 3, 4, 9, 10, 11, 12. But from the local point of view of nodes 2 and 3, they can not determine which coverage hole they are bordering. So they may wrongly find the boundary cycles formed by nodes 1, 3, 4, 5, 6, 7, 8 and by nodes 1, 2, 4, 9, 10, 11, 12.

Some graph based approaches may not be directly used to detect coverage holes, but they can really provide some helpful insights for distributed algorithm design.

2.2 Homology based approaches

Homology based approaches attempt to detect coverage holes based on homology theory. The principle of homology theory consists in analysing topological properties

of some domain by algebraic computations. The main objects are known as simplicial complexes, which are generalization of graphs. Homology group is a topological invariant that can distinguish between topological spaces by measuring the number of holes in a topological space. These concepts may be relatively less known, so it is necessary to give a basic introduction first.

2.2.1 Mathematical background

In this section, we give a brief introduction to homology theory, see [39, 40, 41] for a thorough introduction to this subject. We first give the definitions of simplex and simplicial complex. After that, homology group is defined following definitions of chain group, cycle group and boundary group. Finally, the two most useful simplicial complexes of networks are introduced and their relations are presented.

2.2.1.1 Simplicial complex

Given a set of points V, a k-simplex σ is an unordered set $\{v_0, v_1, ..., v_k\} \subseteq V$ where $v_i \neq v_j$ for all $i \neq j$, k is the dimension of the simplex. In geometric realisation, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle with its interior included and a 3-simplex is a tetrahedron including its interior, see Figure 2.2. Any subset of $\{v_0, v_1, ..., v_k\}$ is called a face of σ. A simplicial complex X is a collection of simplices which satisfies two conditions: (1) any face of a simplex from X is also in X; (2) the intersection of any two simplices σ_1 and σ_2 is a face of both σ_1 and σ_2. An abstract simplicial complex is a purely combinatorial description of the geometric notion of a simplicial complex and therefore does not need the second condition.

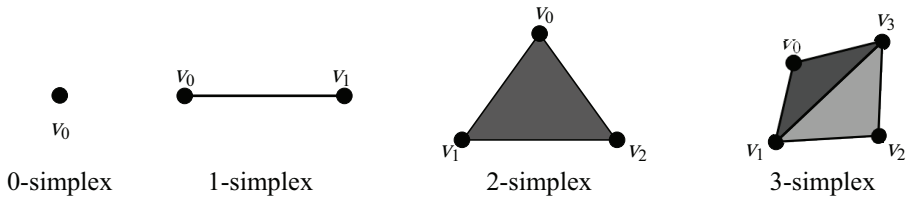


Figure 2.2: An example of simplices

The dimension of a simplicial complex X is the largest dimension of any simplex in X. A subcomplex of X is a simplicial complex $X^{(k)} \subset X$, where k indicates the dimension of $X^{(k)}$.

For example, in the simplicial complex shown in Figure 2.3, it contains six 0-simplices $\{1\}, \{2\}, \ldots, \{6\}$, eight 1-simplices $\{1, 2\}, \{1, 6\}, \{2, 3\}, \{2, 6\}, \{3, 4\}, \{3, 5\}$,

$\{4,5\},\{5,6\}$ and one 2-simplex $\{1,2,6\}$.

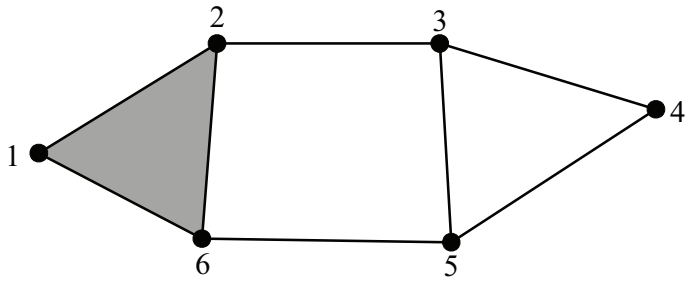


Figure 2.3: An example of simplicial complex

2.2.1.2 Homology group

Let X denote an abstract simplicial complex. One can define an orientation for every k-simplex in X. Denote the k-simplex $\{v_0, v_1, \cdots, v_k\}$ with an ordering by $[v_0, v_1, \cdots, v_k]$, and a change in the orientation corresponds to a change in the sign of the coefficient like

$$[v_0, \cdots, v_i, \cdots, v_j, \cdots, v_k] = -[v_0, \cdots, v_j, \cdots, v_i, \cdots, v_k]$$

Then we can define chain group, boundary map, cycle group, boundary group and finally homology group.

Definition 2.1. *Given an abstract simplicial complex X, for each $k > 0$, the k-chain group $C_k(X)$ is the vector space spanned by the set of oriented k-simplices of X. If k is larger than the dimension of X, $C_k(X)$ is defined to be 0.*

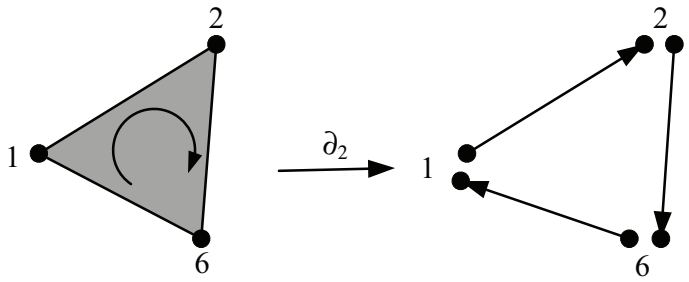
Definition 2.2. *The boundary map ∂_k is defined to be the linear transformation $\partial_k : C_k(X) \to C_{(k-1)}(X)$ which acts on the basis elements of $C_k(X)$ via*

$$\partial_k[v_0, \cdots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, \cdots, v_{i-1}, v_{i+1}, \cdots, v_k]$$

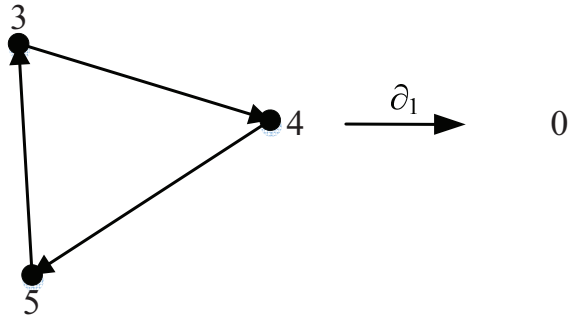
where the sum is algebraic summation.

Still consider the example shown in Figure 2.3, taking into account the orientation, we assume that it contains six 0-simplices $[1], [2], \ldots, [6]$, eight 1-simplices $[1,2], [1,6], [2,3], [2,6], [3,4], [3,5], [4,5], [5,6]$ and one 2-simplex $[1,2,6]$. According to Definition 2.2, we can see that the boundary of the 2-simplex $[1,2,6]$ is the sum of three 1-simplices $[1,2], [2,6]$ and $[6,1]$, as illustrated in Figure 2.4(a). While the

boundary of the sum of three 1-simplices $[3,4],[4,5]$ and $[5,3]$ is 0, shown in Figure 2.4(b).



(a)



(b)

Figure 2.4: Illustrations of boundary

Definition 2.3. *The k-cycle group of X is* $Z_k(X) = \ker \partial_k$.

Definition 2.4. *The k-boundary group of X is* $B_k(X) = \mathrm{im}\partial_{k+1}$.

From Definition 2.3 and 2.4, we can see that $Z_k(X)$ contains all the k-chains with no boundary and $B_k(X)$ contains all the k-chains which are a boundary of one $(k+1)$-chain. A simple calculation demonstrates that $\partial_k \circ \partial_{k+1} = 0$, which means that a boundary has no boundary. It follows that $B_k(X) \subset Z_k(X)$. The relations of $C_k(X), Z_k(X), B_k(X)$ are shown in Figure 2.5.

Then we can well define the homology group.

Definition 2.5. *The k-th homology group of X is the quotient vector space*

$$H_k(X) = \frac{Z_k(X)}{B_k(X)}$$

Definition 2.6. *The k-th Betti number of X is the dimension of $H_k(X)$:*

$$\beta_k = \dim H_k(X) = \dim Z_k(X) - \dim B_k(X).$$

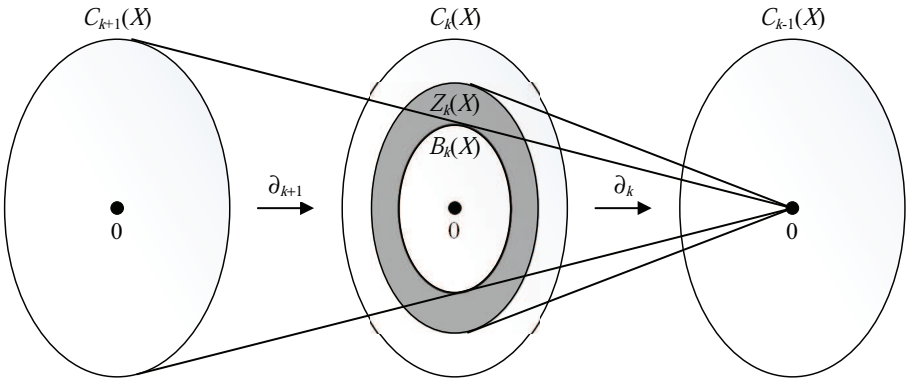


Figure 2.5: An example showing the relations of $C_k(X), Z_k(X), B_k(X)$

From Definition 2.5, we can see that the k-th homology group $H_k(X)$ is a set of equivalence classes of k-cycles. For any two k-cycles z and z', they are in the same equivalence class if and only if $z - z' \in B_k(X)$, that is their difference is a boundary of one $(k+1)$-chain. The Betti numbers are used to count the number of different dimensional holes in X. For example, β_0 indicates the number of 1-dimensional holes, that is the number of connected components. And β_1 counts the number of holes on the plane.

Consider the example in Figure 2.3, we use matrices to describe boundary maps, then we can get that ∂_0 is the null function on the set of 1-simplices, ∂_1 and ∂_2 are given as follows.

$$\partial_1 = \begin{array}{c} \\ [1] \\ [2] \\ [3] \\ [4] \\ [5] \\ [6] \end{array} \begin{array}{c} \begin{array}{cccccccc} [1,2] & [1,6] & [2,3] & [2,6] & [3,4] & [3,5] & [4,5] & [5,6] \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

$$\partial_2 = \begin{array}{c} \\ [1,2] \\ [1,6] \\ [2,3] \\ [2,6] \\ [3,4] \\ [3,5] \\ [4,5] \\ [5,6] \end{array} \begin{array}{c} [1,2,6] \\ \begin{pmatrix} 1 \\ -1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{array}$$

Then we can obtain

$$\beta_0 = \dim \ker \partial_0 - \dim \mathrm{im} \partial_1 = 6 - 5 = 1$$
$$\beta_1 = \dim \ker \partial_1 - \dim \mathrm{im} \partial_2 = 3 - 1 = 2$$

β_0 denotes the number of connect components and β_1 denotes the number of holes. It can be seen from Figure 2.3 that there are two holes surrounded by nodes 2, 3, 5, 6 and 3, 4, 5 respectively.

In above discussions, we only consider computation of the number of holes, without considering the localization of holes. Actually, it is possible to localize holes by computing non-trivial generators of the homology groups. But the standard computation usually exhibits high complexity, which is of quintic order in the number of simplices [38]. In [80], the authors proposed to use the Laplacian operators on chain complexes to detect and localize holes. The Laplacian operator $\mathcal{L}_k$ is defined from the boundary operators and their transposes:

$$\mathcal{L}_k = \partial_{k+1} \partial_{k+1}^* + \partial_k^* \partial_k \tag{2.1}$$

Then it is shown that the k-th Betti number is the dimension of the null space of $\mathcal{L}_k$ and the eigenvectors corresponding to the zero eigenvalues represent homology classes of the k-th homology group. For any k-simplex, it corresponds to a norm of its corresponding eigenvector component, by sequentially select the k-simplices with the highest norm, we may find the boundary cycles. But the found boundary cycles may not be true when two holes close to each other. We still use the example in Figure 2.3 to explain it.

According to (2.1), we can obtain

$$\mathcal{L}_0 = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & -1 \\ -1 & 3 & -1 & 0 & 0 & -1 \\ 0 & -1 & 3 & -1 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & -1 & 3 & -1 \\ -1 & -1 & 0 & 0 & -1 & 3 \end{pmatrix}$$

$$\mathcal{L}_1 = \begin{pmatrix} 3 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 2 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 3 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 2 & 1 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 2 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 2 & -1 \\ 0 & 1 & 0 & 1 & 0 & -1 & -1 & 2 \end{pmatrix}$$

$\mathcal{L}_2 = 3$

It is easy to compute that $\mathcal{L}_0$ has one zero eigenvalue and the corresponding eigenvector is

$$V_0 = \begin{pmatrix} 0.4082 \\ 0.4082 \\ 0.4082 \\ 0.4082 \\ 0.4082 \\ 0.4082 \end{pmatrix}$$

It can be seen that all vertices have the same norms in their corresponding eigenvector components, which means any vertex can be a generator of the homology group.

Similarly, we can find that $\mathcal{L}_1$ has two zero eigenvalues and the corresponding eigenvectors are

$$V_1 = \begin{pmatrix} 0.0306 \\ -0.0306 \\ 0.0918 \\ -0.0612 \\ -0.5386 \\ 0.6304 \\ -0.5386 \\ 0.0918 \end{pmatrix} \text{ and } V_2 = \begin{pmatrix} -0.1800 \\ 0.1800 \\ -0.5400 \\ 0.3600 \\ -0.2768 \\ -0.2632 \\ -0.2768 \\ -0.5400 \end{pmatrix}$$

We choose the three edges which have the highest norms in V_1. Then we can

find that they represent the hole bounded by three edges [3, 4], [4, 5] and [5, 3] in Figure 2.3. But from V_2, if we sequentially choose the edge with the highest norm, we may find the cycle formed by edges [2, 3], [3, 4], [4, 5], [5, 6] and [6, 2] in Figure 2.3. The cycle bounds two holes in Figure 2.3. It means that the cycle found is not accurate. In addition, we can see that $\mathcal{L}_2$ has no zero eigenvalue, which means that there is no 3-dimensional hole in Figure 2.3.

2.2.1.3 Abstract simplicial complexes for networks

For the coverage problem in WSNs, the two most useful abstract simplicial complexes are Čech complex and Rips complex. The Čech complex is defined as follows [38].

Definition 2.7 (Čech complex)**.** *Given a collection of sets $\mathcal{U}$, Čech complex of $\mathcal{U}$, $\check{C}(\mathcal{U})$, is the abstract simplicial complex whose k-simplices correspond to nonempty intersections of $k+1$ distinct elements of $\mathcal{U}$.*

The Čech complex captures the topology of the collection of sets as illustrated by the following theorem.

Theorem 2.1 (Čech theorem)**.** *The Čech complex of a collection of convex sets has the homotopy type of the union of the sets.*

Unfortunately, it is very difficult to compute Čech complex even if the precise information about the relative locations of sensors is provided because of its high complexity. So another more easily computable complex named Rips complex is introduced. It is defined as follows.

Definition 2.8 (Rips complex)**.** *Given a metric space (M, d), a finite set of points $\mathcal{V}$ on M and a fixed radius ϵ, the Rips complex of $\mathcal{V}$, $\mathcal{R}_\epsilon(\mathcal{V})$, is the abstract simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of points in $\mathcal{V}$ which are pairwise within distance ϵ of each other.*

The Rips complex can be constructed with the sole knowledge of the connectivity graph of the network and gives an approximate coverage by simple algebraic calculations. But Rips complex may miss some coverage hole. In fact, there exist following relations between Čech complex and Rips complex [42].

Theorem 2.2. *Let $\mathcal{V}$ be a set of points in $\mathbb{R}^d$ and $\check{C}_\epsilon(\mathcal{V})$ be the Čech complex of the cover of $\mathcal{V}$ by balls of radius ϵ. Then there is*

$$\mathcal{R}_{\epsilon'}(\mathcal{V}) \subset \check{C}_\epsilon(\mathcal{V}) \subset \mathcal{R}_{2\epsilon}(\mathcal{V}) \qquad \textit{whenever} \qquad \frac{\epsilon}{\epsilon'} \geq \sqrt{\frac{d}{2(d+1)}}$$

According to Theorem 2.2, the relation between Čech complex and Rips complex in terms of coverage holes can be derived, which will be presented in Chapter 3.

2.2.2 Homology based approaches

As a pioneer work, in [38], Ghrist and his collaborators introduced homology to detect coverage holes. They first introduced a combinatorial object, Čech complex, which fully characterizes coverage properties of a WSN (existence and location of holes). Unfortunately, this object is very difficult to construct even if the precise information about the relative locations of sensors is available. Thus, they introduced another more easily computable complex, Rips complex. This complex can be constructed with the sole knowledge of the connectivity graph of the network and gives an approximate coverage by simple algebraic calculations. Then their work is followed by [42, 81, 82, 83], where a relative homological criterion for coverage is presented. As regards implementation in real WSNs, these homology based approaches are necessarily centralized, which makes them impractical in large scale sensor networks.

The first steps of implementing the above ideas in a distributed way were taken in [80]. It is shown that combinatorial Laplacians are the right tools for distributed computation of homology groups, and thus can be used for decentralized coverage verification. The combinatorial Laplacians can be used to detect absence of holes or a single hole. But when there are multiple holes close to each other in WSNs, it is not clear how to distinguish them, as shown in Section 2.2.1.2. To address such limitations, a gossip like decentralized algorithm was proposed in [84] to compute homology groups, but its convergence is slow.

In [85, 86], the authors first presented a decentralized scheme based on combinatorial Laplacians to verify whether there is a coverage hole or not in a WSN. For the case when there are coverage holes, they further formulated the problem of localizing coverage holes as an optimization problem for computing a sparse generator of the first homology group of the Rips complex corresponding to the sensor network. But it is possible that some cycle found by their algorithm contains multiple holes next to each other. For the purpose of coverage verification, a distributed algorithm for homology computation was proposed in [87] based on reduction and co-reduction of simplicial complex. But they did not consider the problem of localising coverage holes.

Following the ideas of homology based approaches, two divide and conquer based

algorithms were proposed in [88] and [89] respectively. In [88], the Rips complex of a sensor network is first planarized by eliminating some crossing edges using the scheme proposed in [90, 91]. Then the planar simplicial complex is divided into subcomplexes. In each subcomplex, it is recursively divided until the holes are found. But the process of planarization needs location information of nodes. In [89], the network is iteratively divided into small partitions. And in each partition, it verifies whether there are coverage holes by checking the first homology group of the Rips complex corresponding to this partition. If it is non-trivial, this partition is divided again. Otherwise, it means there is no coverage hole in this partition. As they used the algorithm in [84] to check the first homology group of Rips complex, the scheme also suffers from the same problem as that in [84].

Table 2.3 gives some typical homology based approaches and their characteristics. All these homology based approaches assume that the communication radius of each sensor R_c is no larger than $\sqrt{3}$ times the sensing radius of the sensor R_s. For this point, there are two aspects to be illustrated: (1) if this assumption is satisfied, Rips complex can be used for coverage verification. According to Theorem 2.2, let $d = 2, \epsilon = R_s$ and $\epsilon' = R_c$, we can see if there is no coverage hole in Rips complex $\mathcal{R}_{\epsilon'}(\mathcal{V})$, there must be no coverage hole in the corresponding WSN. But as pointed out in [83], it is also possible that some coverage holes are detected in Rips complex while there is in fact no holes in corresponding WSN, as shown in Figure 2.6. (2) If this assumption is not satisfied, Rips complex may miss some coverage holes, see Figure 2.7. Therefore, although homology theory has the great potential to solve coverage hole problem in WSNs, much work is needed for further research.

Table 2.3: Characteristics of homology based approaches for coverage hole detection

References	Category	Centralized /Distributed	Complexity	Discover boundary cycles
Ghrist et al [38]	Homology based	Centralized	NA	Yes
Muhammad et al [84]	Homology based	Distributed	NA	Yes
Tahbaz-Salehi et al[86]	Homology based	Distributed	NA	Yes
Kanno et al [88]	Homology based	Distributed	NA	Yes
Chintakunta et al[89]	Homology based	Distributed	NA	Yes

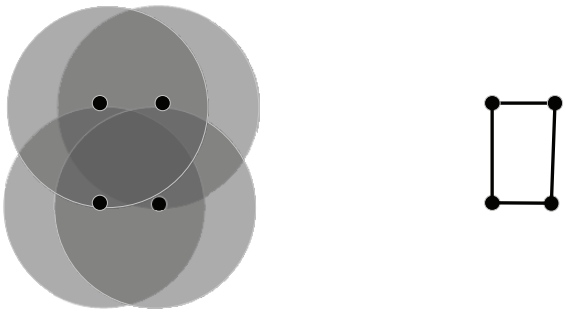


Figure 2.6: An example of Rips complex falsely detects a coverage hole, assume $R_c = R_s$

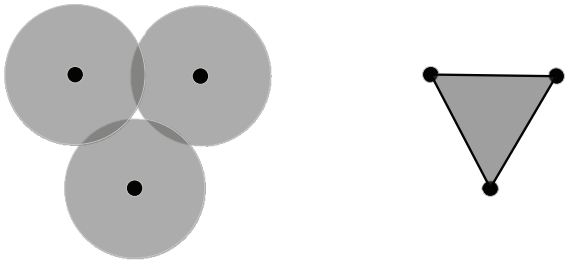


Figure 2.7: An example of Rips complex missing a coverage hole, assume $R_c = 2R_s$

Chapter 3

Accuracy of Homology based Coverage Hole Detection on Plane

3.1 Introduction

As discussed in Chapter 2, Čech complex and Rips complex are two useful tools for coverage hole detection. But Čech complex is rather difficult to construct while Rips complex is easy to construct, so homology based approaches usually use Rips complex to detect coverage holes. In addition, these homology based approaches always assume that the communication radius R_c of a sensor is no larger than $\sqrt{3}$ times the sensing radius R_s of the sensor. But if this assumption is not satisfied, Rips complex may miss some special coverage holes (these holes are defined as triangular holes). It is thus of paramount importance to determine the proportion of the area of such triangular holes to evaluate the accuracy of homology based coverage hole detection. There is not much work on the proportion of the area of triangular holes. Some recent work [92, 93, 94, 95] provided some results on coverage probability but with a different point of view. In [92, 93], the fraction of the area covered by sensors was analysed. In [94], the authors studied how the probability of coverage changes with the sensing radius or the number of sensors. In [95], a point on a plane is defined to be tri-covered if it lies inside a triangle formed by three nodes, and the probability of tri-coverage is analysed. None of them considered triangular holes.

In this chapter, we focus on homology based coverage hole detection for WSNs on plane. Firstly, we identify the relationship between Čech complex and Rips complex in terms of coverage holes. We find that their relationship depends on the ratio between communication and sensing radii and the holes missed by Rips complex

must be bounded by a triangle, we thus define them to be triangular holes and other holes to be non-triangular.

Secondly, we use the proportion of the area of triangular holes as a metric to evaluate the accuracy of homology based coverage hole detection. Such proportion is also related to the ratio between communication and sensing radii of each sensor. So for different ratios, we derive the closed form expressions for lower and upper bounds of the proportion.

Finally, extensive simulations have been run and it is shown that simulation results are well consistent with our analytical lower and upper bounds, with maximum differences of 0.5% and 3%.

3.2 Models and definitions

Consider a collection of stationary sensors (also called nodes) deployed on a planar target field according to a homogeneous Poisson point process with intensity λ. The Poisson point process is defined as follows.

Definition 3.1. *A Poisson point process Φ with intensity λ on a Borel set $\mathcal{B}(X)$ is characterized by two fundamental properties:*
1) For any subset A of $\mathcal{B}(X)$, the number of nodes in A, $n(A)$, is a random variable following Poisson distribution with parameter $\lambda|A|$: $\mathrm{P}(n(A) = k) = e^{-\lambda|A|}\frac{(\lambda|A|)^k}{k!}$, where $|A|$ is the area of A.
2) For any disjoint sets A_1, A_2 of $\mathcal{B}(X)$, the random variables $n(A_1)$ and $n(A_2)$ are independent.

As usual, isotropic radio propagation is assumed. Each sensor monitors a region within a circle of radius R_s and may communicate with other sensors within a circle of radius R_c. Let $\mathcal{V}$ denote the set of sensor locations in a WSN and $\mathcal{S} = \{s_v, v \in \mathcal{V}\}$ be the collection of sensing disks of these sensors: for a location v, $s_v = \{x \in \mathbb{R}^2 : \|x - v\| \leq R_s\}$. Then, according to Definition 2.7, the Čech complex of the WSN, denoted by $\check{C}_{R_s}(\mathcal{V})$, can be constructed as follows: a k-simplex $[v_0, v_1, \cdots, v_k]$ belongs to $\check{C}_{R_s}(\mathcal{V})$ whenever $\cap_{l=0}^{k} s_{v_l} \neq \emptyset$. Similarly, according to Definition 2.8, we consider here the metric space $(\mathbb{R}^2, d)$, then the Rips complex, denoted by $\mathcal{R}_{R_c}(\mathcal{V})$, can be constructed as follows: a k-simplex $[v_0, v_1, \cdots, v_k]$ belongs to $\mathcal{R}_{R_c}(\mathcal{V})$ whenever $\|v_l - v_m\| \leq R_c$ for all $0 \leq l < m \leq k$.

Figure 3.1 shows a WSN, its Čech complex and two Rips complexes for two different values of R_c. Depending on the ratio R_c over R_s, the Rips complex and

the Čech complex may be close or rather different. In this example, for $R_c = 2R_s$, the Rips complex sees the hole surrounded by nodes $2, 3, 5, 6$ as in the Čech complex whereas it is missed in the Rips complex for $R_c = 2.5R_s$. At the same time, the true coverage hole surrounded by nodes $1, 2, 6$ is missed in both Rips complexes.

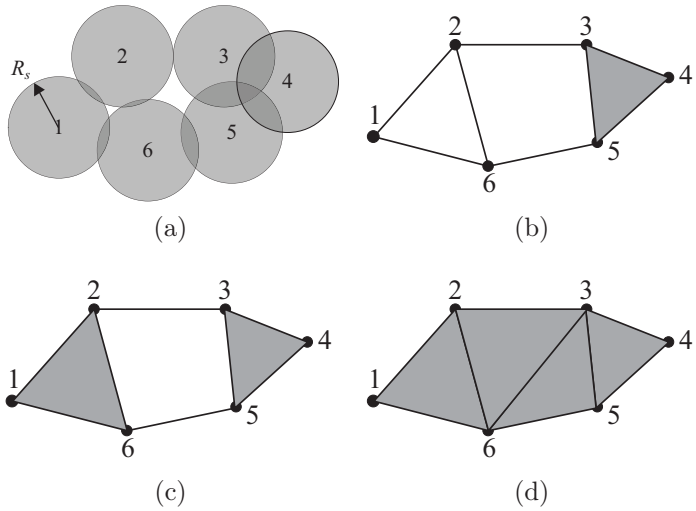


Figure 3.1: (a) A WSN, (b) Čech complex, (c) Rips Complex under $R_c = 2R_s$, (d) Rips Complex under $R_c = 2.5R_s$

In fact, the Čech theorem (Theorem 2.1) indicates that any coverage hole can be found in Čech complex. Furthermore, according to Theorem 2.2, let $d = 2$, $\epsilon = R_s$ and $\epsilon' = R_c$, there are following relations between Čech complex and Rips complex:

$$\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V}) \subset \mathcal{R}_{2R_s}(\mathcal{V}), \text{ whenever } R_c \leq \sqrt{3}R_s. \tag{3.1}$$

According to (3.1), some relationships between Čech complex and Rips complex in terms of coverage holes can be derived as illustrated in the following corollaries. For convenience, define $\gamma = R_c/R_s$.

Corollary 3.1. *When $\gamma \leq \sqrt{3}$, if there is no hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, there must be no hole in Čech complex $\check{C}_{R_s}(\mathcal{V})$.*

Proof. If there is no hole in $\mathcal{R}_{R_c}(\mathcal{V})$, it means that $\mathcal{R}_{R_c}(\mathcal{V})$ can be triangulated. Since $\gamma \leq \sqrt{3}$ means $R_c \leq \sqrt{3}R_s$, according to the first inclusion in (3.1), we have $\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V})$. Consequently, Čech complex $\check{C}_{R_s}(\mathcal{V})$ can also be triangulated. And when $R_c \leq \sqrt{3}R_s$, each triangle must be covered by the sensing ranges of its vertex nodes [42]. So there is no hole in $\check{C}_{R_s}(\mathcal{V})$. □

Corollary 3.2. *When $\gamma \geq 2$, if there is a hole in Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$, there must be a hole in Čech complex $\check{C}_{R_s}(\mathcal{V})$.*

Proof. If there is a hole in $\mathcal{R}_{R_c}(\mathcal{V})$, there must be a cycle with more than three edges in $\mathcal{R}_{R_c}(\mathcal{V})$ that can not be triangulated, as the cycle formed by nodes 2, 3, 5, 6 in Figure 3.1(c). Since $\gamma \geq 2$ means $R_c \geq 2R_s$, according to the second inclusion in (3.1), we can see that $\check{C}_{R_s}(\mathcal{V}) \subset \mathcal{R}_{2R_s}(\mathcal{V}) \subset \mathcal{R}_{R_c}(\mathcal{V})$. Consequently, there must also be a cycle in $\check{C}_{R_s}(\mathcal{V})$ which can not be triangulated. And there is a coverage hole in the cycle. □

Corollary 3.3. *When $\sqrt{3} < \gamma < 2$, there is no guarantee relation between Rips complex $\mathcal{R}_{R_c}(\mathcal{V})$ and Čech complex $\check{C}_{R_s}(\mathcal{V})$ in terms of holes.*

From Corollary 3.1, a sufficient condition for coverage verification can be derived, which has been figured out in [86]. But it is not a necessary condition. It is possible that there is no hole in $\check{C}_{R_s}(\mathcal{V})$, while there is a hole in $\mathcal{R}_{R_c}(\mathcal{V})$. From Corollary 3.2, we can find a necessary condition for the existence of a hole in $\check{C}_{R_s}(\mathcal{V})$. Corollary 3.3 indicates that when there is no hole in $\mathcal{R}_{R_c}(\mathcal{V})$, it is possible that there is a hole in $\check{C}_{R_s}(\mathcal{V})$. When there is a hole in $\mathcal{R}_{R_c}(\mathcal{V})$, it is also possible that $\check{\mathrm{C}}_{R_s}(\mathcal{V})$ contains no hole.

From the discussions above, we can find that a hole in a Čech complex not seen in a Rips complex must be bounded by a triangle. Based on this observation, a formal definition of 'triangular hole' and 'non-triangular hole' is given as follows.

Definition 3.2 (Triangular and non-triangular hole)**.** *For a pair of complexes $\check{C}_{R_s}(\mathcal{V})$ and $\mathcal{R}_{R_c}(\mathcal{V})$ of a WSN, a triangular hole is an uncovered region bounded by a triangle formed by three nodes v_0, v_1, v_2, where v_0, v_1, v_2 can form a 2-simplex which appears in $\mathcal{R}_{R_c}(\mathcal{V})$ but not in $\check{C}_{R_s}(\mathcal{V})$. Any other holes are non-triangular.*

From Definition 3.2, we can see from Figure 3.1 that when $R_c = 2R_s$, there is only one triangular hole that bounded by the triangle formed by nodes 1, 2 and 6. When $R_c = 2.5R_s$, there are two additional triangular holes, bounded by triangles formed by nodes 2, 3, 6 and 3, 5, 6 respectively.

3.3 Bounds on proportion of triangular holes

In this section, the conditions under which any point on the target field is inside a triangular hole are first given. From the discussion in Section 3.2, it is found that the

proportion of the area of triangular holes is related to the ratio γ. Three different cases are considered for the proportion computation. For each case, the upper and lower bounds of the proportion are derived.

3.3.1 Preliminary

Lemma 3.4. *For any point on the target field, it is inside a triangular hole if and only if the following two conditions are satisfied:*

1. *the distance between the point and its nearest node is larger than R_s.*
2. *the point is inside a triangle formed by three nodes with pairwise distance less than or equal to R_c.*

Lemma 3.5. *If there exists a point O which is inside a triangular hole, then $R_s < R_c/\sqrt{3}$.*

Proof. According to the definition of triangular holes, if there exists a triangular hole, then there must be a 2-simplex which is in $\mathcal{R}_{R_c}(\mathcal{V})$ but not in $\check{C}_{R_s}(\mathcal{V})$. If $R_s \geq R_c/\sqrt{3}$, then according to the first inclusion in (1), we have $\mathcal{R}_{R_c}(\mathcal{V}) \subset \check{C}_{R_s}(\mathcal{V})$, it means that there are no 2-simplices which are in $\mathcal{R}_{R_c}(\mathcal{V})$ but not in $\check{C}_{R_s}(\mathcal{V})$, there is a contradiction, so $R_s < R_c/\sqrt{3}$. □

Lemma 3.6. *If O is inside a triangular hole and l denotes the distance between O and its closest neighbour, we have $R_s < l \leq R_c/\sqrt{3}$.*

Proof. $R_s < l$ is a direct corollary from Lemma 3.4. We only need to prove $l \leq R_c/\sqrt{3}$. If point O is inside a triangular hole, it must be surrounded by a triangle formed by sensors with pairwise distance less than or equal to R_c. Assume it is surrounded by a triangle $\mathrm{N_0N_1N_2}$, as in Figure 3.2. The closest neighbour of O is not necessarily in the set $\{\mathrm{N_0}, \mathrm{N_1}, \mathrm{N_2}\}$. If $l > R_c/\sqrt{3}$, then $d_0 \geq l > R_c/\sqrt{3}$, $d_1 \geq l > R_c/\sqrt{3}$ and $d_2 \geq l > R_c/\sqrt{3}$. In addition, since $\alpha + \beta + \varphi = 2\pi$, there must be one angle no smaller than $2\pi/3$. Without loss of generality, assume $\alpha \geq 2\pi/3$, then according to the law of cosines, $d_{02}^2 = d_0^2 + d_2^2 - 2d_0d_2\cos\alpha > R_c^2/3 + R_c^2/3 - 2/3R_cR_c\cos(2\pi/3) = R_c^2$. So $d_{02} > R_c$. Since $\mathrm{N_0}$ and $\mathrm{N_2}$ are neighbours, $d_{02} \leq R_c$. There is a contradiction. Therefore $l \leq R_c/\sqrt{3}$. □

A homogeneous Poisson point process is stationary, thus without considering border effect [43], any point has the same probability to be inside a triangular hole as the origin O. This probability in a homogeneous setting is also equal to the

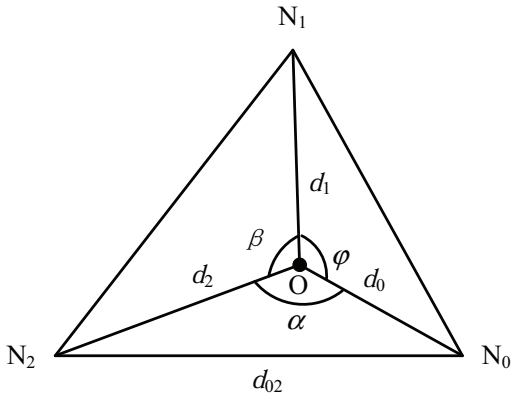


Figure 3.2: Illustration of O being inside a triangular hole

proportion of the area of triangular holes. We borrow part of the line of proof from [95] where a similar problem was analysed.

We consider the probability that the origin O is inside a triangular hole. Since the length of each edge in the Rips complex must be at most R_c, only the nodes within R_c from the origin can contribute to the triangle which bounds a triangular hole containing the origin. Therefore, we only need to consider the Poisson point process constrained in the closed ball $B(O, R_c)$, which is also a homogeneous Poisson point process with intensity λ. We denote this process as Φ. In addition, $T(x, y, z)$ denotes the property that the origin O is inside the triangular hole bounded by the triangle with points x, y, z as vertices. When n_0, n_1, n_2 are points of the process Φ, $T(n_0, n_1, n_2)$ is also used to denote the event that the triangle formed by the nodes n_0, n_1, n_2 bounds a triangular hole containing the origin. In addition, we use $T'(n_0, n_1, n_2)$ to denote the event that the nodes n_0, n_1, n_2 can not form a triangle which bounds a triangular hole containing the origin.

Let $\tau_0 = \tau_0(\Phi)$ be the node in the process Φ which is closest to the origin. There are two cases for the origin to be inside a triangular hole. The first case is that the node τ_0 can contribute to a triangle which bounds a triangular hole containing the origin. The second case is that the node τ_0 can not contribute to any triangle which bounds a triangular hole containing the origin but other three nodes can form a triangle which bounds a triangular hole containing the origin. So the probability

that the origin is inside a triangular hole can be defined as

$$\begin{aligned} p_{2d}(\lambda) &= \mathrm{P}\{O \text{ is inside a triangular hole}\} \\ &= \mathrm{P}\{\bigcup_{\{n_0,n_1,n_2\}\subseteq\Phi} T(n_0,n_1,n_2)\} \\ &= \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(\tau_0,n_1,n_2)\} + p_{2d}^{\mathrm{sec}}(\lambda) \end{aligned} \tag{3.2}$$

where

$$p_{2d}^{\mathrm{sec}}(\lambda) = \mathrm{P}\{\bigcup_{\{n_{i1},\cdots,n_{i5}\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(n_{i1},n_{i2},n_{i3}) \mid T'(\tau_0,n_{i4},n_{i5})\}$$

denotes the probability that τ_0 can not contribute to any triangle which bounds a triangular hole containing the origin but other three nodes can form a triangle which bounds a triangular hole containing the origin.

In the following parts, we will analyse this probability in three different cases.

3.3.2 Case $0 < \gamma \leq \sqrt{3}$

Theorem 3.7. *When $0 < \gamma \leq \sqrt{3}$, $p_{2d}(\lambda) = 0$.*

Proof. It is a direct corollary from Lemma 3.5. □

3.3.3 Case $\sqrt{3} < \gamma \leq 2$

Theorem 3.8. *When $\sqrt{3} < \gamma \leq 2$, we have $p_{2dl}(\lambda) < p_{2d}(\lambda) < p_{2du}(\lambda)$, where*

$$\begin{aligned} p_{2dl}(\lambda) =& 2\pi\lambda^2 \int_{R_s}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1 \end{aligned} \tag{3.3}$$

and

$$\begin{aligned} p_{2du}(\lambda) =& 2\pi\lambda^2 \int_{R_s}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1 + p_{2d}^{\mathrm{sec}}(\lambda) \end{aligned} \tag{3.4}$$

and

$$\varphi_l(r_0) = 2\arccos(R_c/(2r_0)) \tag{3.5}$$

$$\varphi_u(r_0) = 2\arcsin(R_c/(2r_0)) - 2\arccos(R_c/(2r_0)) \tag{3.6}$$

$$
\begin{aligned}
R_1(r_0,\varphi_1) &= \min(\sqrt{R_c^2 - r_0^2\sin^2\varphi_1} - r_0\cos\varphi_1, \\
&\qquad \sqrt{R_c^2 - r_0^2\sin^2(\varphi_1+\varphi_l(r_0))} + r_0\cos(\varphi_1+\varphi_l(r_0))) \qquad (3.7)\\
|S^+(r_0,\varphi_1)| &= \int_{\varphi_l(r_0)}^{\varphi_1}\int_{r_0}^{R_1(r_0,\varphi)} r dr d\varphi \\
|S^-(r_0,r_1,\theta_1)| &= \int_{\theta_{2l}}^{-\varphi_l(r_0)}\int_{r_0}^{R_2(r_0,r_1,\theta_1,\theta_2)} r dr d\theta_2 \\
\theta_{2l} &= \theta_1 - \arccos\frac{\cos(R_c/R) - \cos\theta_1\cos\theta_0}{\sin\theta_1\sin\theta_0} \\
R_2(r_0,r_1,\theta_1,\theta_2) &= \min(\sqrt{R_c^2 - r_0^2\sin^2\theta_2} - r_0\cos\theta_2, \\
&\qquad \sqrt{R_c^2 - r_1^2\sin^2(\theta_2-\theta_1)} + r_1\cos(\theta_2-\theta_1))
\end{aligned}
$$

$p_{2d}^{\mathrm{sec}}(\lambda)$ *is obtained by simulations.*

Proof. We first prove the lower bound. It can be obtained from (3.2) that

$$
p_{2d}(\lambda) > \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(\tau_0,n_1,n_2)\}
$$

So for the lower bound, we only consider the case that the closest node τ_0 must contribute to a triangle which bounds a triangular hole containing the origin.

Using polar coordinates, we assume the closest node τ_0 lies on (d_0,π). It is well known that the distance d_0 is a random variable with distribution

$$
F_{d_0}(r_0) = \mathrm{P}\{d_0 \le r_0\} = 1 - e^{-\lambda\pi r_0^2}
$$

Then the probability density function of d_0 can be found as

$$
f_{d_0}(r_0) = 2\pi\lambda r_0 e^{-\lambda\pi r_0^2} \qquad (3.8)
$$

From Lemma 3.6, we can obtain $R_c < d_0 \le R_c/\sqrt{3}$, then the above probability can be written as

$$
\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T(\tau_0,n_1,n_2)\} \\
&= \int_{R_s}^{R_c/\sqrt{3}} \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{r_0}} T((r_0,\pi),n_1,n_2)\} f_{d_0}(r_0) dr_0
\end{aligned} \qquad (3.9)
$$

where Φ'_{r_0} is the restriction of Φ in $B(O,R_c)\setminus B(O,r_0)$.

Once the node τ_0 is determined, the other two nodes must lie in the different half spaces: one in $H^+ = \mathbb{R}^+\times(0,\pi)$ and the other in $H^- = \mathbb{R}^+\times(-\pi,0)$. Assume n_1 lies

in H^+ and n_2 lies in H^-. Since the distance to τ_0 is at most R_c, n_1 and n_2 must also lie in the ball $B(\tau_0, R_c)$. Furthermore, the distance to the origin is at most R_c and larger than d_0, they should also lie in the region $A = B(O, R_c) \backslash B(O, d_0)$. Therefore, n_1 must lie in $G^+ = H^+ \bigcap B(\tau_0, R_c) \bigcap A$ and n_2 must lie in $G^- = H^- \bigcap B(\tau_0, R_c) \bigcap A$. In addition, considering the distance between n_1 and n_2 should be at most R_c and the origin should be inside the triangle formed by τ_0, n_1 and n_2, n_1 must lie in the shadow region $A^+ = G^+ \bigcap B(M_2, R_c)$, shown in Figure 3.3. M_2 is one intersection point between the circle $C(O, d_0)$ and the circle $C(\tau_0, R_c)$, such intersection point must exist in this case since $R_c = \gamma R_s \leq 2R_s < 2d_0$.

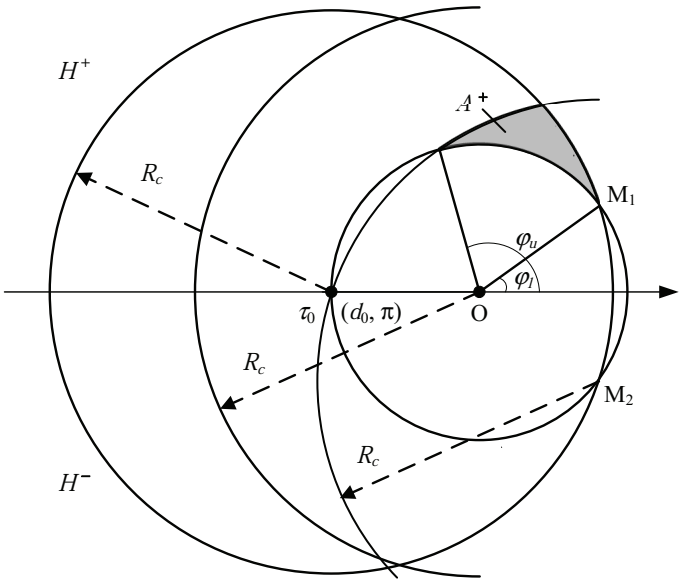


Figure 3.3: Illustration of region A^+ in the case $\sqrt{3} < \gamma \leq 2$

Ordering the nodes in A^+ by increasing polar angle so that $\tau_1 = (d_1, \varphi_1)$ has the smallest angle φ_1. And assume the nodes τ_0, τ_1 and another node $\tau_2 \in G^-$ can form a triangle which bounds a triangular hole containing the origin, then τ_2 must lie to the right of the line passing through τ_1 and O, denoted by $H^+(\varphi_1)$ which contains all points with polar angle $\varphi \in (\varphi_1 - \pi, \varphi_1)$. In addition, the distance to τ_1 is at most R_c, so the node τ_2 must lie in the region S^-, as illustrated in Figure 3.4.

$$S^-(\tau_0, \tau_1) = S^-(d_0, d_1, \varphi_1) = G^- \bigcap H^+(\varphi_1) \bigcap B(\tau_1, R_c)$$

Here we need to obtain the density of node τ_1. Considering the way τ_1 was defined, there should be no nodes with a polar angle less than φ_1, that is to say no

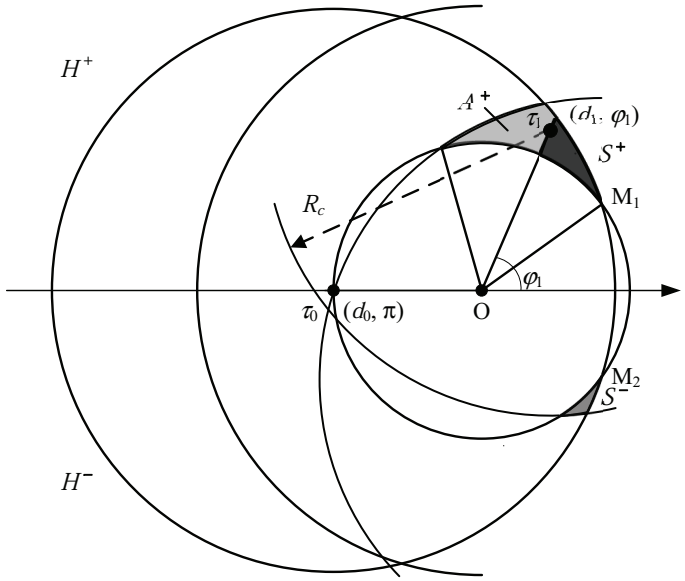


Figure 3.4: Illustration of regions S^+ and S^- in the case $\sqrt{3} < \gamma \leq 2$

nodes are in the region

$$S^+(\tau_0, \tau_1) = S^+(d_0, \varphi_1) = A^+ \bigcap H^+(\varphi_1)$$

Since the intensity measure of the Poisson point process in polar coordinates is $\lambda r dr d\varphi$, the density F_{τ_1} of τ_1 can be expressed as

$$F_{\tau_1}(dr_1, d\varphi_1) = \lambda r_1 e^{-\lambda |S^+(d_0, \varphi_1)|} dr_1 d\varphi_1 \tag{3.10}$$

The integration domain $D(d_0)$ with respect to parameters (d_1, φ_1) can be easily obtained. From the construction of the region A^+, we can get $\varphi_l(r_0) = 2\arccos(R_c/(2d_0))$ and $\varphi_u(r_0) = 2\arcsin(R_c/(2d_0)) - 2\arccos(R_c/(2d_0))$. So $\varphi_l(r_0) \leq \varphi_1 \leq \varphi_u(r_0)$ and $d_0 < d_1 \leq R_1(d_0, \varphi_1)$, where

$$\begin{aligned} R_1(d_0, \varphi_1) = &\min(\sqrt{R_c^2 - d_0^2 \sin^2 \varphi_1} - d_0 \cos \varphi_1, \\ &\sqrt{R_c^2 - d_0^2 \sin^2(\varphi_1 + \varphi_l(r_0))} + d_0 \cos(\varphi_1 + \varphi_l(r_0))) \end{aligned}$$

Assume only τ_0, τ_1 and nodes in $S^-(\tau_0, \tau_1)$ can contribute to the triangle which bounds a triangular hole containing the origin, we can get a lower bound of the probability that the origin is inside a triangular hole. It is a lower bound because it is possible that τ_1 can not contribute to any triangle which bounds a triangular hole containing the origin, but some other nodes with higher polar angles in the

region A^+ can contribute to such a triangle. For example, in Figure 3.5, if there is no node in S^- but there are some nodes in S'^-, then τ_1 can not contribute to any triangle which bounds a triangular hole containing the origin, but τ_1' can form such a triangle with τ_0 and another node in S'^-. Based on the assumption, we have

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{r_0}} T((r_0,\pi),n_1,n_2)\} > \mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{r_0}\bigcap S^-(\tau_0,\tau_1)} T((r_0,\pi),\tau_1,n_2)\} \\
&= \iint_{D(r_0)} \mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{r_0}\bigcap S^-(r_0,r_1,\varphi_1)} T((r_0,\pi),(r_1,\varphi_1),n_2)\} F_{\tau_1}(dr_1,d\varphi_1) \qquad (3.11) \\
&= \iint_{D(r_0)} \mathrm{P}\{\Phi'_{r_0}(S^-(r_0,r_1,\varphi_1)) > 0\} F_{\tau_1}(dr_1,d\varphi_1)
\end{aligned}$$

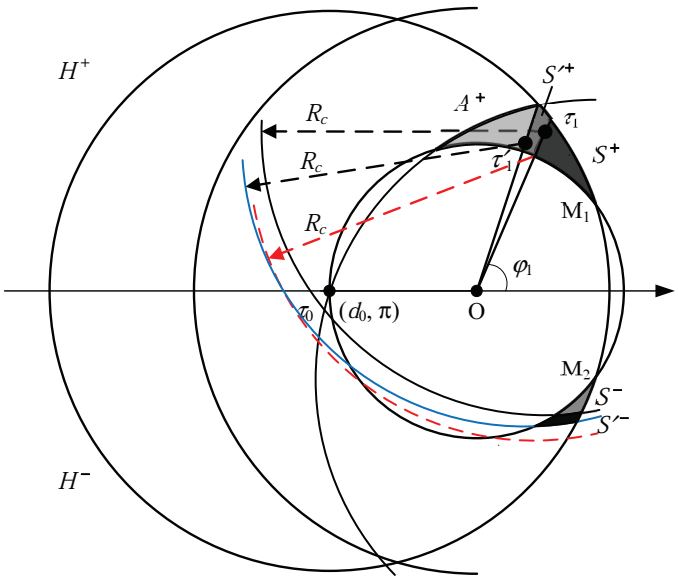


Figure 3.5: Illustration of regions S'^+ and S'^- in the case $\sqrt{3} < \gamma \leq 2$

Therefore, from (3.8), (3.9), (3.10) and (3.11), the lower bound shown in (3.3) can be derived.

Next we will prove the upper bound. As discussed in Section 3.3.1, there are two cases for the origin being inside a triangular hole. As for the second case that the closest node τ_0 can not but some other nodes can contribute to a triangle which bounds a triangular hole containing the origin, it is not easy to obtain a closed form expression for such probability, so we can get it by simulations. Simulation results show that this probability is less than 0.0025 % at any intensity λ when $\sqrt{3} < \gamma \leq 2$. So we still focus on the probability of the first case.

For the lower bound, we only considered the case that τ_1 contributes to a triangle which bounds a triangular hole containing the origin. For the upper bound, we

need to further consider the case that τ_1 can not but some other nodes in A^+ can contribute to such a triangle, shown in Figure 3.5. Assume the node $\tau_1' = (d_1', \varphi_1')$ with the second smallest polar angle in A^+ can contribute to such a triangle, it means that there is no node in $S^-(d_0, d_1, \varphi_1)$ but there is at lease one node in the region $S'^-(d_0, d_1, \varphi_1, d_1', \varphi_1') = S^-(d_0, d_1', \varphi_1')\backslash S^-(d_0, d_1, \varphi_1)$.

Then the density of the pair (τ_1, τ_1') is given as

$$F_{\tau_1,\tau_1'}(dr_1, d\varphi_1, dr_1', d\varphi_1') = \lambda^2 r_1 r_1' e^{-\lambda|S^+(d_0,\varphi_1')|} dr_1 d\varphi_1 dr_1' d\varphi_1' \tag{3.12}$$

The probability that τ_1 can not but τ_1' can form a triangle which bounds a triangular hole containing the origin with τ_0 and another node in $S'^-(d_0, d_1, \varphi_1, d_1', \varphi_1')$ can be expressed as

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_2,n_3\}\subseteq\Phi'_{r_0}\bigcap S^-(\tau_0,\tau_1)} T((r_0,\pi),\tau_1',n_2) \mid T'((r_0,\pi),\tau_1,n_3)\} \\
&= \iiiint \mathrm{P}\{\Phi'_{r_0}(S^-(r_0,r_1,\varphi_1)) = 0\} \\
&\qquad \times \mathrm{P}\{\Phi'_{r_0}(S'^-(r_0,r_1,\varphi_1,r_1',\varphi_1')) > 0\} F_{\tau_1,\tau_1'}(dr_1, d\varphi_1, dr_1', d\varphi_1') \\
&= \iiiint e^{-\lambda|S^-(r_0,r_1,\varphi_1)|} \times (1 - e^{-\lambda|S'^-(r_0,r_1,\varphi_1,r_1',\varphi_1')|}) F_{\tau_1,\tau_1'}(dr_1, d\varphi_1, dr_1', d\varphi_1')
\end{aligned} \tag{3.13}$$

As we can see from Figure 3.5, as long as τ_1' has a higher polar angle than τ_1 has, the sum of $|S^-(r_0, r_1', \varphi_1')|$ and $|S'^-(r_0, r_1, \varphi_1, r_1', \varphi_1')|$ will be always smaller than $|S^-(r_0, r_0, \varphi_1)|$.

Therefore we can get from (3.13)

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_2,n_3\}\subseteq\Phi'_{r_0}\bigcap S^-(\tau_0,\tau_1)} T((r_0,\pi),\tau_1',n_2) \mid T'((r_0,\pi),\tau_1,n_3)\} \\
&< \iiiint (e^{-\lambda|S^-(r_0,r_1,\varphi_1)|} - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) F_{\tau_1,\tau_1'}(dr_1, d\varphi_1, dr_1', d\varphi_1')
\end{aligned} \tag{3.14}$$

Furthermore, let $S'^+(d_0, \varphi_1, \varphi_1') = S^+(d_0, \varphi_1')\backslash S^+(d_0, \varphi_1)$, then

$$\iint \lambda r_1' e^{-\lambda|S'^+(d_0,\varphi_1,\varphi_1')|} dr_1' d\varphi_1' = 1 - e^{-\lambda|A^+\backslash S^+(d_0,\varphi_1)|} < 1 \tag{3.15}$$

It is the complement of the probability that no node is in the region $A^+\backslash S^+(d_0, \varphi_1)$.

From (3.11), (3.12), (3.14) and (3.15), we can obtain

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{r_0}\bigcap S^-(\tau_0,\tau_1)} T((r_0,\pi),\tau_1,n_2)\} \\
&+\mathrm{P}\{\bigcup_{\{n_2,n_3\}\subseteq\Phi'_{r_0}\bigcap S^-(\tau_0,\tau_1)} T((r_0,\pi),\tau'_1,n_2)\mid T'((r_0,\pi),\tau_1,n_3)\} \\
&<\iint_{D(r_0)}(1-e^{-\lambda|S^-(r_0,r_0,\varphi_1)|})F_{\tau_1}(dr_1,d\varphi_1)
\end{aligned} \tag{3.16}$$

Similarly, we can further consider the case that neither of τ_1 and τ'_1 can contribute to a triangle which bounds a triangular hole containing the origin, but other nodes with even higher polar angle can contribute to such a triangle. In this way, we can get the same result as (3.16).

Therefore, it can be derived that

$$\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{r_0}} T((r_0,\pi),n_1,n_2)\}<\iint_{D(r_0)}(1-e^{-\lambda|S^-(r_0,r_0,\varphi_1)|})F_{\tau_1}(dr_1,d\varphi_1) \tag{3.17}$$

From (3.2), (3.8), (3.9), (3.10) and (3.17), we can get the upper bound shown in (3.4).

Here we need to compute the areas of $S^+(r_0,\varphi_1)$, $S^-(r_0,r_1,\varphi_1)$ and $S^-(r_0,r_0,\varphi_1)$. In fact, the areas $|S^+(r_0,\varphi_1)|$, $|S^-(r_0,r_1,\varphi_1)|$ and $|S^-(r_0,r_0,\varphi_1)|$ have very similar expressions. For example, the area $|S^+(r_0,\varphi_1)|$ can be expressed as

$$|S^+(r_0,\varphi_1)|=\int_{\varphi_l(r_0)}^{\varphi_1}\int_{r_0}^{R_1(r_0,\varphi)} r dr d\varphi=\frac{1}{2}\int_{\varphi_l(r_0)}^{\varphi_1}[R_1^2(r_0,\varphi)-r_0^2]d\varphi \tag{3.18}$$

When $\varphi_1\le\pi/2-\arccos(R_c/(2r_0))$,

$$|S^+(r_0,\varphi_1)|=I(r_0,\varphi_1)-I(r_0,\varphi_l(r_0))$$

where

$$I(r_0,\varphi)=\frac{r_0^2\sin 2\varphi}{4}+\frac{R_c^2\varphi}{2}-\frac{R_c^2}{2}\arcsin\frac{r_0\sin\varphi}{R_c}-\frac{r_0\sin\varphi}{2}\sqrt{R_c^2-r_0^2\sin^2\varphi}-\frac{r_0^2\varphi}{2}$$

When $\pi/2-\arccos(R_c/(2r_0))<\varphi_1\le\varphi_u(r_0)$,

$$|S^+(r_0,\varphi_1)|=2|S^+(r_0,\pi/2-\arccos(\frac{R_c}{2r_0})|-|S^+(r_0,\pi-2\arccos(\frac{R_c}{2r_0})-\varphi_1|$$

Similarly, $|S^-(r_0,r_1,\varphi_1)|$ and $|S^-(r_0,r_0,\varphi_1)|$ can be obtained. The detailed com-

putation is presented in Section A.1 of Appendix A. □

3.3.4 Case $\gamma > 2$

Theorem 3.9. *When $\gamma > 2$, we have $p_{2dl}(\lambda) < p_{2d}(\lambda) < p_{2du}(\lambda)$, where*

$$\begin{aligned} p_{2dl}(\lambda) =& 2\pi\lambda^2 \Big\{ \int_{R_s}^{R_c/2} r_0 dr_0 \int_0^{\pi} d\varphi_1 \int_{r_0}^{R_1'(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1 \\ & + \int_{R_c/2}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_1,\varphi_1)|}) r_1 dr_1 \Big\} \end{aligned} \tag{3.19}$$

and

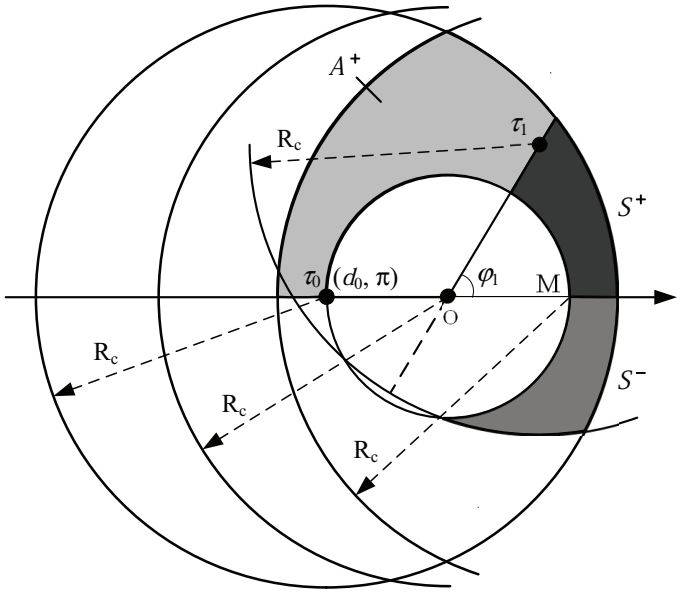
$$\begin{aligned} p_{2du}(\lambda) =& 2\pi\lambda^2 \Big\{ \int_{R_s}^{R_c/2} r_0 dr_0 \int_0^{\pi} d\varphi_1 \int_{r_0}^{R_1'(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1 \\ & + \int_{R_c/2}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ & \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1 - e^{-\lambda|S^-(r_0,r_0,\varphi_1)|}) r_1 dr_1 \Big\} + p_{2d}^{\text{sec}}(\lambda) \end{aligned} \tag{3.20}$$

and

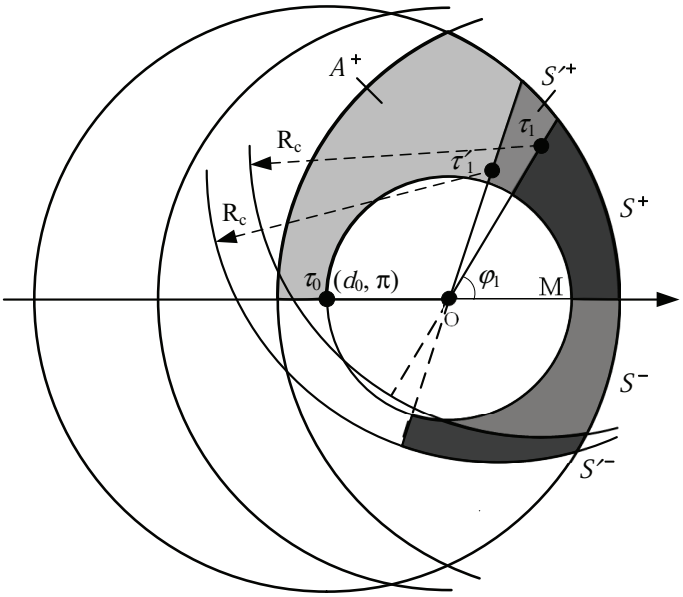
$$R_1'(r_0,\varphi_1) = \min(\sqrt{R_c^2 - r_0^2\sin^2\varphi_1} - r_0\cos\varphi_1, \sqrt{R_c^2 - r_0^2\sin^2\varphi_1} + r_0\cos\varphi_1)$$

$p_{2d}^{\text{sec}}(\lambda)$ is obtained by simulations.

In this case, we can use the same method as in Section 3.3.3 to get the lower and upper bounds, shown in (3.19) and (3.20) respectively. But we need to consider two situations $R_s < d_0 \leq R_c/2$ and $R_c/2 < d_0 \leq R_c/\sqrt{3}$. In the first situation, $d_0 \leq R_c/2$ means that the ball $B(O, d_0)$ is included in the ball $B(\tau_0, R_c)$. The illustrations for the regions A^+, S^+, S^-, S'^+ and S'^- in this situation are shown in Figure 3.6(a) and 3.6(b) respectively. In addition, the lower limit of integration for φ_1 is 0 and the upper limit is π. The computation of the area $|S^+(r_0,\varphi_1)|$ is the same as that in Section 3.3.3, but the computation of the area $|S^-(r_0,r_1,\varphi_1)|$ is a little different, which is shown in Section A.2 of Appendix A. The second situation is the same as that in Section 3.3.3. Furthermore, simulation results show that $p_{2d}^{\text{sec}}(\lambda)$ is less than 0.16% at any intensity when $2 < \gamma \leq 3$.



(a)



(b)

Figure 3.6: Illustrations of regions in case $\gamma > 2$. (a) the regions A^+, S^+ and S^-, (b) the regions S'^+ and S'^-

3.4 Performance evaluation

This section first introduces simulation settings. Simulation results are then presented and compared with analytical lower and upper bounds.

3.4.1 Simulation settings

A disk centered at the origin with radius R_c is considered in the simulations. The probability that the origin is inside a triangular hole is computed. Sensors are randomly distributed in the disk according to a homogeneous Poisson process with intensity λ. The sensing radius R_s of each node is set to be 10 meters and γ is chosen from 2 to 3 with interval of 0.2. So the communication radius R_c ranges from 20 to 30 meters with interval of 2 meters. λ is selected from 0.001 to 0.020 with interval of 0.001. For each γ, 10^7 simulations are run under each λ to check whether the origin is inside a triangular hole.

3.4.2 Performance evaluation

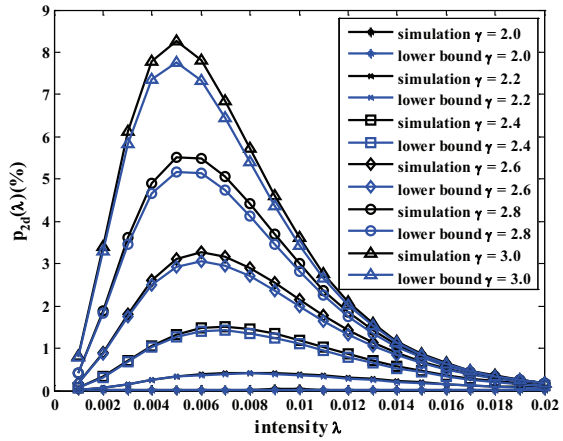
The probability $p_{2d}(\lambda)$ obtained by simulations is presented with the lower bound and upper bound in Figure 3.7(a) and 3.7(b) respectively. Simulation results for $p_{2d}^{\text{sec}}(\lambda)$ are shown in Figure 3.7(c). The detailed values are given in Table 3.1, 3.2 3.3 and 3.4 respectively.

It can be seen that for any value of γ, $p_{2d}(\lambda)$ has a maximum at a threshold value λ_c of the intensity. As a matter of fact, for $\lambda \leq \lambda_c$, the number of nodes is small. Consequently the probability of the origin being inside a triangular hole is relatively small too. With the increase of λ, the connectivity between nodes becomes stronger. As a result, the probability of the origin being inside a triangular hole increases. However, when the intensity reaches the threshold value, the origin is covered with maximum probability. $p_{2d}(\lambda)$ decreases for $\lambda \geq \lambda_c$. The simulations also show that λ_c decreases with the increase of γ.

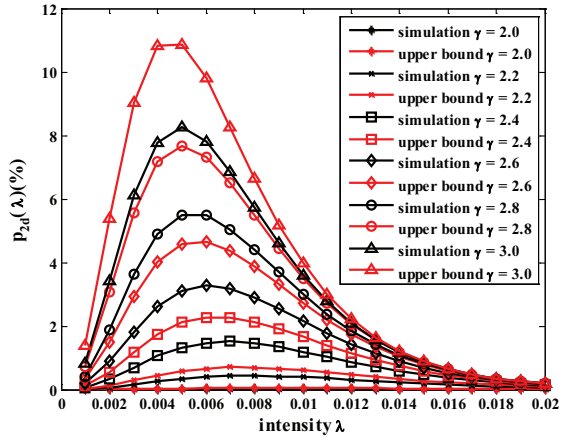
On the other hand, it can be seen from Figure 3.7(a) and 3.7(b) that for a fixed intensity λ, $p_{2d}(\lambda)$ increases with the increases of γ. That is because R_s is fixed. Then the larger R_c is, the higher is the probability of each triangle containing a coverage hole.

Furthermore, the maximum probability increases quickly with γ ranging from 2.0 to 3.0. These results can be used for planning of WSNs, which will be discussed in Section 3.4.3.

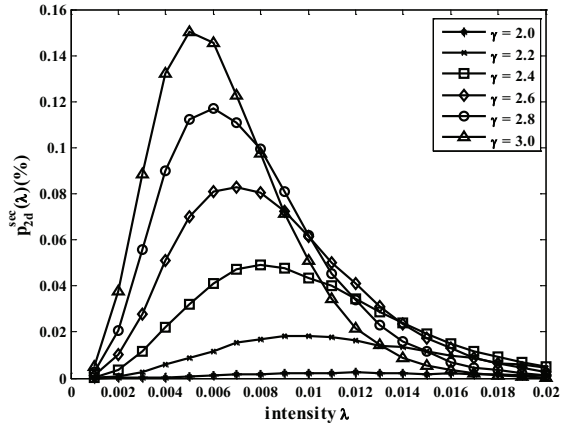
Finally, it can be found in Figure 3.7(a) that the probability obtained by simulations is very well consistent with the lower bound. The maximum difference between them is about 0.5% according to Table 3.3. Figure 3.7(b) shows that probability obtained by simulations is also consistent with the upper bound. The maximum difference between them is about 3% according to Table 3.3.



(a)



(b)



(c)

Figure 3.7: Proportion of the area of triangular holes (a) simulation results and lower bounds, (b) simulation results and upper bounds, (c) simulation results for $p_{2d}^{\mathrm{sec}}(\lambda)$

Table 3.1: $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.0, 2.2$ (%)

λ	$\gamma = 2.0$			$\gamma = 2.2$		
	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$
0.001	0.0005	0.0006	0.0011	0.0126	0.0126	0.0239
0.002	0.0031	0.0035	0.0061	0.0677	0.0694	0.1264
0.003	0.0074	0.0075	0.0144	0.1534	0.1553	0.2828
0.004	0.0124	0.0127	0.0242	0.2443	0.2545	0.4456
0.005	0.0171	0.0184	0.0335	0.3210	0.3335	0.5779
0.006	0.0210	0.0228	0.0411	0.3734	0.3921	0.6636
0.007	0.0236	0.0245	0.0462	0.3996	0.4222	0.7028
0.008	0.0250	0.0262	0.0487	0.4023	0.4267	0.6987
0.009	0.0252	0.0280	0.0493	0.3866	0.4139	0.6637
0.010	0.0245	0.0269	0.0479	0.3583	0.3886	0.6082
0.011	0.0231	0.0265	0.0451	0.3225	0.3512	0.5412
0.012	0.0213	0.0239	0.0418	0.2834	0.3080	0.4697
0.013	0.0192	0.0219	0.0372	0.2440	0.2673	0.3989
0.014	0.0170	0.0200	0.0330	0.2066	0.2275	0.3348
0.015	0.0148	0.0166	0.0285	0.1724	0.1902	0.2764
0.016	0.0127	0.0145	0.0250	0.1421	0.1591	0.2249
0.017	0.0108	0.0124	0.0211	0.1158	0.1284	0.1817
0.018	0.0091	0.0109	0.0179	0.0935	0.1042	0.1445
0.019	0.0076	0.0087	0.0146	0.0749	0.0836	0.1147
0.020	0.0063	0.0076	0.0122	0.0595	0.0654	0.0898

Table 3.2: $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.4, 2.6$ (%)

λ	$\gamma = 2.4$			$\gamma = 2.6$		
	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$
0.001	0.0622	0.0629	0.1130	0.1820	0.1848	0.3241
0.002	0.3147	0.3211	0.5597	0.8661	0.8917	1.4929
0.003	0.6739	0.6994	1.1729	1.7466	1.8157	2.9153
0.004	1.0157	1.0610	1.7301	2.4854	2.6147	4.0217
0.005	1.2643	1.3270	2.1078	2.9277	3.0995	4.5933
0.006	1.3957	1.4734	2.2785	3.0654	3.2670	4.6651
0.007	1.4192	1.5096	2.2701	2.9631	3.1675	4.3771
0.008	1.3596	1.4508	2.1301	2.7047	2.9082	3.8832
0.009	1.2451	1.3417	1.9113	2.3655	2.5394	3.3036
0.010	1.1011	1.1887	1.6561	2.0021	2.1450	2.7223
0.011	0.9469	1.0256	1.3974	1.6515	1.7763	2.1887
0.012	0.7960	0.8616	1.1523	1.3345	1.4309	1.7273
0.013	0.6567	0.7089	0.9325	1.0607	1.1363	1.3408
0.014	0.5334	0.5776	0.7440	0.8316	0.8874	1.0286
0.015	0.4275	0.4660	0.5859	0.6448	0.6862	0.7813
0.016	0.3389	0.3674	0.4559	0.4953	0.5253	0.5894
0.017	0.2660	0.2880	0.3522	0.3775	0.3979	0.4411
0.018	0.2070	0.2228	0.2697	0.2859	0.2985	0.3288
0.019	0.1600	0.1733	0.2054	0.2153	0.2246	0.2440
0.020	0.1229	0.1329	0.1551	0.1614	0.1680	0.1800

Table 3.3: $p_{2dl}(\lambda)$, $p_{2du}(\lambda)$ and simulation results for $p_{2d}(\lambda)$ under $\gamma = 2.8, 3.0$ (%)

λ	$\gamma = 2.8$			$\gamma = 3.0$		
	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$	$p_{2dl}(\lambda)$	$p_{2d}(\lambda)$	$p_{2du}(\lambda)$
0.001	0.4110	0.4194	0.7212	0.7912	0.8105	1.3722
0.002	1.8301	1.8947	3.0740	3.2854	3.4145	5.3836
0.003	3.4664	3.6261	5.5778	5.8312	6.1158	9.0434
0.004	4.6489	4.9111	7.1727	7.3642	7.7864	10.8372
0.005	5.1793	5.5055	7.6722	7.7633	8.2559	10.8698
0.006	5.1464	5.4880	7.3298	7.3343	7.8099	9.8046
0.007	4.7371	5.0501	6.5004	6.4488	6.8510	8.2590
0.008	4.1314	4.4196	5.4764	5.3967	5.7198	6.6504
0.009	3.4638	3.6929	4.4427	4.3604	4.5967	5.1905
0.010	2.8193	2.9963	3.5068	3.4344	3.6102	3.9673
0.011	2.2434	2.3763	2.7126	2.6547	2.7753	2.9872
0.012	1.7540	1.8577	2.0687	2.0235	2.1000	2.2259
0.013	1.3527	1.4175	1.5586	1.5263	1.5781	1.6483
0.014	1.0319	1.0866	1.1647	1.1422	1.1734	1.2141
0.015	0.7804	0.8102	0.8662	0.8497	0.8744	0.8920
0.016	0.5862	0.6093	0.6395	0.6293	0.6417	0.6538
0.017	0.4378	0.4527	0.4713	0.4645	0.4706	0.4785
0.018	0.3256	0.3343	0.3467	0.3419	0.3479	0.3498
0.019	0.2413	0.2450	0.2541	0.2512	0.2541	0.2557
0.020	0.1783	0.1836	0.1858	0.1843	0.1862	0.1868

Table 3.4: Simulation results for $p_{2d}^{\text{sec}}(\lambda)$

λ	Simulation results for $p_{2d}^{\text{sec}}(\lambda)$ under different γ (%)					
	2.0	2.2	2.4	2.6	2.8	3.0
0.001	0	0.0001	0.0004	0.0012	0.0024	0.0051
0.002	0.0001	0.0008	0.0037	0.0100	0.0208	0.0377
0.003	0.0002	0.0025	0.0116	0.0277	0.0556	0.0884
0.004	0.0004	0.0059	0.0219	0.0512	0.0899	0.1323
0.005	0.0008	0.0090	0.0321	0.0702	0.1121	0.1501
0.006	0.0012	0.0117	0.0409	0.0808	0.1172	0.1454
0.007	0.0015	0.0156	0.0474	0.0830	0.1109	0.1229
0.008	0.0017	0.0170	0.0490	0.0803	0.0995	0.0977
0.009	0.0021	0.0181	0.0475	0.0723	0.0809	0.0714
0.010	0.0023	0.0185	0.0436	0.0615	0.0621	0.0512
0.011	0.0022	0.0180	0.0399	0.0502	0.0455	0.0346
0.012	0.0025	0.0165	0.0344	0.0410	0.0339	0.0218
0.013	0.0020	0.0140	0.0285	0.0309	0.0232	0.0145
0.014	0.0020	0.0134	0.0238	0.0233	0.0158	0.0086
0.015	0.0016	0.0118	0.0192	0.0171	0.0116	0.0056
0.016	0.0019	0.0098	0.0148	0.0130	0.0069	0.0034
0.017	0.0016	0.0087	0.0118	0.0090	0.0047	0.0020
0.018	0.0015	0.0066	0.0091	0.0064	0.0034	0.0010
0.019	0.0010	0.0057	0.0070	0.0046	0.0020	0.0006
0.020	0.0010	0.0043	0.0050	0.0028	0.0010	0.0003

3.4.3 Discussions on applications

In this chapter, we only consider triangular holes. For non-triangular holes, we assume they can be detected and covered by additional nodes. Under this assumption, our analytical results can be used for planning of WSNs. For example, a WSN is used to monitor a planar target field and the ratio $\gamma = 2$, according to the analytical upper bounds, we can see that the maximum proportion of the area of triangular holes under $\gamma = 2$ is about 0.05%, which can be neglected. It means that as long as the planar target field can be triangulated by nodes, we can say the target field is covered. But if $\gamma = 3$ and at least 95% of the target field should be covered, then it means that the proportion of the area of triangular holes can be at most 5%. From the analytical upper bounds of $\gamma = 3$, it can be seen that when the intensity $\lambda = 0.009$, the upper bound is about 5%, so in order to cover at least 95% of the target field, the intensity of nodes should be larger than 0.009. Although in our simulations, we only consider γ between 2.0 and 3.0, similar results can be obtained for any other values of γ.

3.5 Chapter summary

In this chapter, we have considered the accuracy of homology based coverage hole detection on plane. We first derive the relationship between Čech complex and Rips complex in terms of coverage holes, which shows that their relationship is related to the ratio between communication and sensing radii of each sensor. Based on the relationship, the situations when Rips complex may miss coverage holes are identified. We find that the holes missed by Rips complex are always bounded by a triangle, so define them to be triangular holes. Furthermore, we use the proportion of the area of triangular holes to evaluate the accuracy of homology based coverage hole detection on plane. Sensors are assumed to be randomly deployed on a planar target field according to a homogeneous Poisson point process. Under such homogeneous configuration, without considering border effect, the proportion of the area of triangular holes is equivalent to the probability of any point in the target field being inside a triangular hole. Then we consider the probability in three different cases of the ratio between communication and sensing radii. For each case, the closed form expressions for the lower and upper bounds of the probability are derived. Simulation results are well consistent with the analytical lower and upper bounds, with maximum differences of 0.5% and 3%.

Chapter 4

Accuracy of Homology based Coverage Hole Detection on Sphere

4.1 Introduction

In Chapter 3, the accuracy of homology based coverage hole detection on plane has been analysed. But in some real applications, such as volcano monitoring [14] and forest monitoring [15], the target fields are complex surfaces. So it is also important to consider the coverage hole detection problem of WSNs on surfaces. On the other hand, from theoretical point of view, the coverage on 3D surfaces is quite a different problem from its counterpart in 2D plane. As sphere is the simplest case of 3D surfaces, we choose it as the first step for the accuracy analysis of homology based coverage hole detection in this chapter. The relationship between Čech complex and Rips complex on sphere is different from that on plane. Similar as on plane, the holes missed by Rips complex on sphere are always bounded by a spherical triangle and are thus defined to be spherical triangular holes. We also use the proportion of the area of spherical triangular holes to evaluate the accuracy of homology based coverage hole detection on sphere. There is not much work on the proportion of the area of spherical triangular holes. In [96], the authors proposed the surface coverage model and derived the expected coverage ratio under stochastic deployment on 3D surface. In [97], the expected coverage ratio under stochastic deployment on 3D rolling terrains was derived. Nevertheless, none of these research considered spherical triangular holes.

In this chapter, we first analyse the relationship between Čech complex and Rips complex in terms of coverage holes on sphere. we find that a hole in a Čech complex

missed by a Rips complex must be bounded by a spherical triangle. Based on that, a formal definition of spherical triangular hole is given.

Second, we choose the proportion of the area of spherical triangular holes as a metric to evaluate the accuracy of homology based coverage hole detection on sphere. Such proportion is analysed under a homogeneous setting and it is related to the communication and sensing radii of each sensor. Three cases are considered for the computation of such proportion. For each case, closed form expressions for lower and upper bounds of the proportion are derived.

Third, extensive simulations are performed to evaluate impacts of communication and sensing radii, radius of sphere on proportion of the area of spherical triangular holes. It is shown that simulation results are well consistent with the analytical lower bound, with a maximum difference of 0.5%, and consistent with the analytical upper bound, with a maximum difference of 3%. Furthermore, simulation results show that the radius of sphere has little impact on the proportion when it is much larger than communication and sensing radii.

4.2 Models and definitions

Consider a collection of stationary sensors (also called nodes) deployed randomly on a sphere $\mathbb{S}^2$ with radius R according to a homogeneous Poisson point process with intensity λ. For any two points p_1 and p_2 on $\mathbb{S}^2$, the distance between them $d(p_1, p_2)$ is defined to be the great circle distance, which is the shortest distance between any two points on the surface of a sphere measured along a path on the surface of the sphere. As usual, isotropic radio propagation is assumed. All sensors have the same sensing radius R_s and communication radius R_c on $\mathbb{S}^2$. It means for any sensor located at v on $\mathbb{S}^2$, any point p on $\mathbb{S}^2$ with $d(v, p) \leq R_s$ is within the sensing range of the sensor; and for any two sensors located at v_i, v_j on $\mathbb{S}^2$, they can communicate with each other if $d(v_i, v_j) \leq R_c$. In addition, we assume $R_s \ll R$, $R_c \ll R$.

On sphere, we also use Čech complex and Rips complex to capture coverage holes. Before constructing them, we need to point out that the realisation of k-simplex on sphere is different from that in Euclidean space $\mathbb{R}^d$. The realisation of 0-, 1-, 2- and 3-simplex in $\mathbb{R}^d$ has been shown in Figure 2.2, here we give the realisation of 0-, 1- and 2-simplex on a sphere $\mathbb{S}^2$ in Figure 4.1. We can see that on a sphere $\mathbb{S}^2$, a 0-simplex $[v_0]$ is a vertex , a 1-simplex $[v_0, v_1]$ is the shorter arc of the great circle passing through v_0 and v_1, a 2-simplex $[v_0, v_1, v_2]$ is a spherical triangle $v_0v_1v_2$ with its interior included.

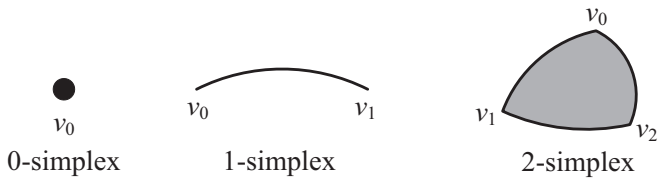


Figure 4.1: 0-, 1- and 2-simplex on sphere

Then we can construct Čech complex and Rips complex. Let $\mathcal{V}$ denote the set of sensor locations in a WSN on $\mathbb{S}^2$ with radius R and $\mathcal{S} = \{s_v,\ v \in \mathcal{V}\}$ denote the collection of sensing ranges of these sensors: for a location v, $s_v = \{x \in \mathbb{S}^2 : d(x, v) \leq R_s\}$. According to Definition 2.7, the Čech complex of the WSN on sphere, denoted by $\check{C}_{R_s}(\mathcal{V})$, can be constructed as follows: a k-simplex $[v_0, v_1, \cdots, v_k]$ belongs to $\check{C}_{R_s}(\mathcal{V})$ whenever $\cap_{l=0}^{k} s_{v_l} \neq \emptyset$. Similarly, according to Definition 2.8, we consider here the metric space $(\mathbb{S}^2, d)$, then the Rips complex, denoted by $\mathcal{R}_{R_c}(\mathcal{V})$, can be constructed as follows: a k-simplex $[v_0, v_1, \cdots, v_k]$ belongs to $\mathcal{R}_{R_c}(\mathcal{V})$ whenever $d(v_l, v_m) \leq R_c$ for all $0 \leq l < m \leq k$. In addition, since we consider only coverage holes on the sphere $\mathbb{S}^2$, it is sufficient to construct 2-dimensional Čech complex and 2-dimensional Rips complex of the WSN, denoted as $\check{C}^{(2)}_{R_s}(\mathcal{V})$ and $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ respectively.

Figure 4.2 shows a WSN, its Čech complex and two Rips complexes for two different values of R_c. Depending on the relation of R_c and R_s, the Rips complex and the Čech complex may be close or rather different. In this example, for $R_c = 2R_s$, the Rips complex sees the hole surrounded by $2, 3, 5, 6$ as in the Čech complex whereas it is missed in the Rips complex for $R_c = 2.5R_s$. At the same time, the true coverage hole surrounded by $1, 2, 6$ is missed in both Rips complexes.

Theorem 2.2 in Chapter 2 presents the relationship between Čech complex and Rips complex in Euclidean space, but that is not true for sphere. So we derive the relationship between $\check{C}^{(2)}_{R_s}(\mathcal{V})$ and $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ on sphere as follows.

Lemma 4.1. *Let $\mathcal{V}$ denote the set of node locations in a WSN on $\mathbb{S}^2$ with radius R, all nodes have the same sensing radius R_s and communication radius R_c, $R_s \ll R, R_c \ll R$, then*

$$\mathcal{R}^{(2)}_{R_c}(\mathcal{V}) \subset \check{C}^{(2)}_{R_s}(\mathcal{V}) \subset \mathcal{R}^{(2)}_{2R_s}(\mathcal{V}), \text{whenever } R_c \leq R \arccos([3\cos^2(R_s/R) - 1]/2) \quad (4.1)$$

Proof. The second inclusion is trivial because for any k-simplex $[v_0, v_1, \cdots, v_k] \in \check{C}^{(2)}_{R_s}(\mathcal{V})$, it means the sensing ranges of these nodes have a common intersection, so the pairwise distance $d(v_i, v_j) \leq 2R_s$ for all $0 \leq i < j \leq k$, which means

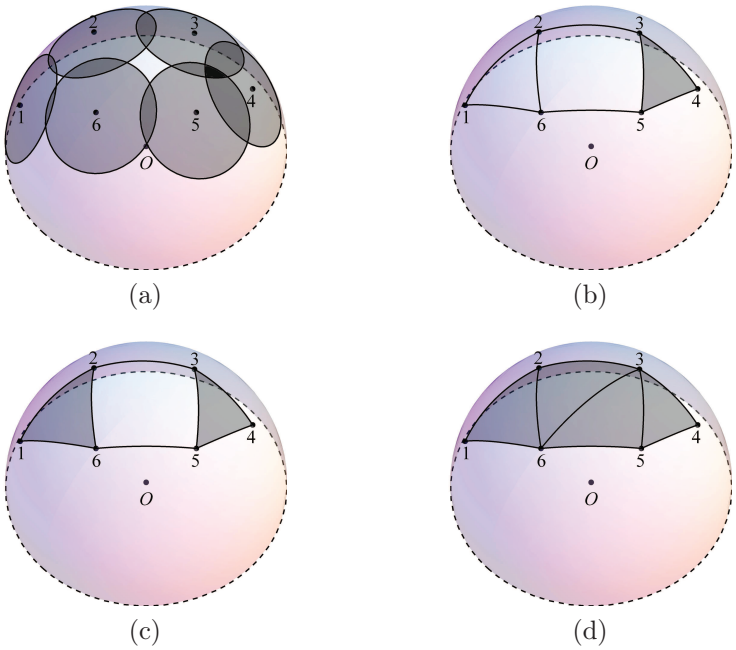


Figure 4.2: (a) A WSN, (b) Čech complex, (c) Rips Complex under $R_c = 2R_s$, (d) Rips Complex under $R_c = 2.5R_s$

$[v_0, v_1, \cdots, v_k] \in \mathcal{R}^{(2)}_{2R_s}(\mathcal{V})$.

As for the first inclusion, it is clear that $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ and $\check{C}^{(2)}_{R_s}(\mathcal{V})$ contain the same 0-simplices. It is also easy to see that all 1-simplices in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ must also be in $\check{C}^{(2)}_{R_s}(\mathcal{V})$ since for any 1-simplex $[v_i, v_j]$ with distance $d(v_i, v_j) \leq R_c \leq R\arccos([3\cos^2(R_s/R) - 1]/2) < R\arccos(2\cos^2(R_s/R) - 1) = 2R_s$, it means that the sensing ranges of the two nodes have a common intersection. So we only need to prove that all 2-simplices in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ must be in $\check{C}^{(2)}_{R_s}(\mathcal{V})$. It is equivalent to say that for any three nodes with pairwise great circle distance no larger than R_c, their sensing ranges must have a common intersection.

Assume a 2-simplex $[v_0, v_1, v_2] \in \mathcal{R}^{(2)}_{R_c}(\mathcal{V})$, then the three nodes v_0, v_1 and v_2 must determine a plane α. We consider the spherical cap on $\mathbb{S}^2$ cut off by the plane α. Since $R_c < R$, the spherical cap must be on a hemisphere. It is easy to see that the intersection of the plane α and sphere $\mathbb{S}^2$ is a circle c. Let O_1 be the center of circle c, O be the center of $\mathbb{S}^2$, P be the intersection of line OO_1 and $\mathbb{S}^2$.

Using spherical coordinates, we assume the point P has a spherical coordinate $(R, 0, 0)$. P may be inside[1] or outside the spherical triangle $v_0v_1v_2$, which is shown in Figure 4.3(a) and 4.3(b) respectively.

It can be seen that P has the same great circle distance to v_0, v_1 and v_2, denoted

[1]It also includes the case that P is on one arc of the spherical triangle $v_0v_1v_2$.

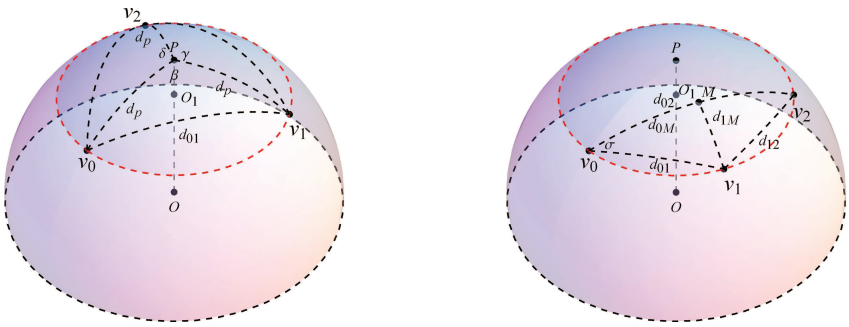


Figure 4.3: Illustrations of P and spherical triangle $v_0v_1v_2$: (a) P is inside the spherical triangle $v_0v_1v_2$, (b) P is outside the spherical triangle $v_0v_1v_2$

by d_p. If P is inside the spherical triangle $v_0v_1v_2$, as shown in Figure 4.3(a), then we can prove $d_p \leq R_s$. Since P lying inside the spherical triangle $v_0v_1v_2$ means $\beta+\gamma+\delta = 2\pi$, there must be one angle no smaller than $2\pi/3$. Without loss of generality, assume $\beta \geq 2\pi/3$. According to the spherical law of consines, we have $\cos(\beta) = \frac{\cos(d_{01}/R)-\cos^2(d_p/R)}{\sin^2(d_p/R)} \leq -1/2 \Rightarrow \cos(d_{01}/R) \leq [3\cos^2(d_p/R)-1]/2$. In addition, $d_{01} \leq R_c \leq R\arccos([3\cos^2(R_s/R)-1]/2) \Rightarrow \cos(d_{01}/R) \geq [3\cos^2(R_s/R)-1]/2$, and $0 < d_{01}/R, d_p/R < \pi/2$, so we have $[3\cos^2(R_s/R)-1]/2 \leq [3\cos^2(d_p/R)-1]/2 \Rightarrow d_p \leq R_s$, which means the point P is a common intersection of sensing ranges of v_0, v_1 and v_2, so $[v_0, v_1, v_2] \in \check{C}^{(2)}_{R_s}(\mathcal{V})$.

If P is outside the spherical triangle $v_0v_1v_2$, as shown in Figure 4.3(a), it indicates that the spherical triangle $v_0v_1v_2$ must be contained in half of the spherical cap. Assume v_0, v_1 and v_2 have spherical coordinates $(R, \theta, \varphi_0), (R, \theta, \varphi_1)$ and (R, θ, φ_2), where $\theta \in (0, \pi/2), \varphi_0 < \varphi_1 < \varphi_2$, then we have $\varphi_1 - \varphi_0, \varphi_2 - \varphi_1, \varphi_2 - \varphi_0 \in (0, \pi)$. Using d_{01}, d_{12}, d_{02} to denote the pairwise great circle distances between v_0, v_1, v_2, then according to the spherical law of consines, we have

$$\cos(d_{01}/R) = \cos^2\theta + \sin^2\theta\cos(\varphi_1 - \varphi_0) \tag{4.2}$$

$$\cos(d_{12}/R) = \cos^2\theta + \sin^2\theta\cos(\varphi_2 - \varphi_1) \tag{4.3}$$

$$\cos(d_{02}/R) = \cos^2\theta + \sin^2\theta\cos(\varphi_2 - \varphi_0) \tag{4.4}$$

In addition, we use σ to denote the angle between two arcs $\widehat{v_0v_1}$ and $\widehat{v_0v_2}$, M to denote the middle point of the arc $\widehat{v_0v_2}$ and d_{0M}, d_{1M} to denote great circle distances

between v_0, v_1 and M. It can be seen $d_{0M} = d_{02}/2$. Similarly, we have

$$\cos\sigma = \frac{\cos(d_{12}/R) - \cos(d_{01}/R)\cos(d_{02}/R)}{\sin(d_{01}/R)\sin(d_{02}/R)} \tag{4.5}$$

$$\cos\frac{d_{1M}}{R} = \cos\frac{d_{01}}{R}\cos\frac{d_{0M}}{2R} + \sin\frac{d_{01}}{R}\cos\frac{d_{0M}}{2R}\cos\sigma \tag{4.6}$$

From (4.5) and (4.6), we can obtain

$$\cos\frac{d_{1M}}{R} = \frac{\cos(d_{01}/R) + \cos(d_{12}/R)}{2\cos(d_{02}/(2R))} \tag{4.7}$$

Consequently

$$\cos\frac{d_{1M}}{R} - \cos\frac{d_{0M}}{R} = \frac{\cos\frac{d_{01}}{R} + \cos\frac{d_{12}}{R} - \cos\frac{d_{02}}{R} - 1}{2\cos(d_{02}/(2R))} \tag{4.8}$$

From (4.2), (4.3), (4.4) and (4.8), we get

$$\cos\frac{d_{1M}}{R} - \cos\frac{d_{0M}}{R} = \frac{\sin^2\theta\cos\frac{\varphi_2-\varphi_0}{2}\sin\frac{\varphi_1-\varphi_0}{2}\sin\frac{\varphi_2-\varphi_1}{2}}{\cos\frac{d_{02}}{2R}} \tag{4.9}$$

Since $0 < \varphi_1 - \varphi_0, \varphi_2 - \varphi_1, \varphi_2 - \varphi_0 < \pi$ and $0 < d_{1M}/R, d_{0M}/R, d_{02}/R < \pi/2$, it can be obtained from (4.9) $d_{1M} < d_{0M} \leq R_c/2 < R_s$, which means the point M is a common intersection of the sensing ranges of v_0, v_1 and v_2, so $[v_0, v_1, v_2] \in \check{C}^{(2)}_{R_s}(\mathcal{V})$. It means all 2-simplices in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ must be in $\check{C}^{(2)}_{R_s}(\mathcal{V})$. Consequently the first inclusion is proved. □

According to (4.1), some relationships between Čech complex and Rips complex in terms of coverage holes can be derived as illustrated in the following corollaries.

Corollary 4.2. *When $R_c \leq R\arccos([3\cos^2(R_s/R) - 1]/2)$, if there is no hole in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$, there must be no hole in $\check{C}^{(2)}_{R_s}(\mathcal{V})$.*

Corollary 4.3. *When $R_c \geq 2R_s$, if there is a hole in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$, there must be a hole in $\check{C}^{(2)}_{R_s}(\mathcal{V})$.*

Corollary 4.4. *When $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c < 2R_s$, there is no guarantee relation between $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ and $\check{C}^{(2)}_{R_s}(\mathcal{V})$ in terms of holes.*

From the discussions above, a hole in a $\check{C}^{(2)}_{R_s}(\mathcal{V})$ not seen in a $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ must be bounded by a spherical triangle. Based on this observation, a formal definition of spherical triangular hole is given as follows.

Definition 4.1 (Spherical triangular hole)**.** *For a pair of complexes $\check{C}^{(2)}_{R_s}(\mathcal{V})$ and $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ of a WSN, a spherical triangular hole is an uncovered region bounded by a spherical triangle formed by three nodes v_0, v_1, v_2, where v_0, v_1, v_2 can form a 2-simplex which appears in $\mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ but not in $\check{C}^{(2)}_{R_s}(\mathcal{V})$.*

From Definition 4.1, we can see from Figure 4.2 that when $R_c = 2R_s$, there is one spherical triangular hole bounded by the spherical triangle formed by nodes 1, 2 and 6. And when $R_c = 2.5R_s$, there are two additional spherical triangular holes, bounded by spherical triangles formed by nodes 2,3,6 and 3, 5, 6 respectively.

4.3 Bounds on proportion of spherical triangular holes

In this section, the conditions under which any point on $\mathbb{S}^2$ with radius R is inside a spherical triangular hole are first given. From the discussions in Section 4.2, it is found that the proportion of the area of spherical triangular holes is related to the relation of R_c and R_s. Three different cases are considered for the proportion computation. For each case, closed form expressions for lower and upper bounds of the proportion are derived.

4.3.1 Preliminary

Lemma 4.5. *For any point on $\mathbb{S}^2$, it is inside a spherical triangular hole if and only if the following two conditions are satisfied:*

1. *the great circle distance between the point and its closest node is larger than R_s.*
2. *the point is inside a spherical triangle formed by three nodes with pairwise great circle distance less than or equal to R_c.*

Lemma 4.6. *If there exists a point O which is inside a spherical triangular hole, then $R_s < R \arccos \sqrt{[1 + 2\cos(R_c/R)]/3}$.*

Proof. According to Definition 4.1, if there is a point O inside a spherical triangular hole, then there exists a 2-simplex $\sigma \in \mathcal{R}^{(2)}_{R_c}(\mathcal{V})$ while $\sigma \notin \check{C}^{(2)}_{R_s}(\mathcal{V})$, so $\mathcal{R}^{(2)}_{R_c}(\mathcal{V}) \not\subset \check{C}^{(2)}_{R_s}(\mathcal{V})$. According to (4.1), we have $R_c > R \arccos([3\cos^2(R_s/R) - 1]/2) \Rightarrow R_s < R \arccos \sqrt{[1 + 2\cos(R_c/R)]/3}$. □

Lemma 4.7. *Let O be a point inside a spherical triangular hole and l denote the great circle distance between O and its closest neighbour, then*
$R_s < l \leq R \arccos \sqrt{[1 + 2\cos(R_c/R)]/3}$.

The proof is similar as that of Lemma 4.1.

Since we assume nodes are distributed on $\mathbb{S}^2$ according to a homogeneous Poisson point process with intensity λ, any point has the same probability to be inside a spherical triangular hole without considering border effect [43]. This probability in a homogeneous setting is also equal to the proportion of the area of spherical triangular holes.

We use spherical coordinates (R, θ, φ) to denote points on $\mathbb{S}^2$ with radius R, where θ is polar angle and φ is azimuth angle. We consider the probability of the point N with spherical coordinates $(R, 0, 0)$ being inside a spherical triangular hole. Since the communication radius of each sensor is at most R_c, only the nodes within R_c from the point N can contribute to the spherical triangle which bounds a spherical triangular hole containing N. Therefore, we only need to consider the Poisson point process constrained on the spherical cap $C(N, R_c)$ which is also a homogeneous Poisson point process with intensity λ, where $C(N, R_c)$ denotes the spherical cap centered at point N and the maximum great circle distance between N and points on the spherical cap is R_c. We denote this process as Φ. In addition, $T_s(x, y, z)$ denotes the property that the point N is inside the spherical triangular hole bounded by the spherical triangle with points x, y, z as vertices. When n_0, n_1, n_2 are points of the process Φ, $T_s(n_0, n_1, n_2)$ is also used to denote the event that the spherical triangle formed by the nodes n_0, n_1, n_2 bounds a spherical triangular hole containing the point N. In addition, we use $T'_s(n_0, n_1, n_2)$ to denote the event that the nodes n_0, n_1, n_2 can not form a spherical triangle which bounds a spherical triangular hole containing the point N.

Let $\tau_0 = \tau_0(\Phi)$ be the node in the process Φ which is closest to the point N. There are two cases for the point N to be inside a spherical triangular hole. The first case is that the node τ_0 can contribute to a spherical triangle which bounds a spherical triangular hole containing the point N. The second case is that the node τ_0 can not contribute to any spherical triangle which bounds a spherical triangular hole containing the point N but other three nodes can form a spherical triangle which bounds a spherical triangular hole containing the point N. So the probability

that the point N is inside a spherical triangular hole can be defined as

$$\begin{aligned} p_s(\lambda) &= \mathrm{P}\{N \text{ is inside a spherical triangular hole}\} \\ &= \mathrm{P}\{\bigcup_{\{n_0,n_1,n_2\}\subseteq\Phi} T_s(n_0,n_1,n_2)\} \\ &= \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T_s(\tau_0,n_1,n_2)\} + p_s^{\mathrm{sec}}(\lambda) \end{aligned} \tag{4.10}$$

where

$$p_s^{\mathrm{sec}}(\lambda) = \mathrm{P}\{\bigcup_{\{n_{i1},\cdots,n_{i5}\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T_s(n_{i1},n_{i2},n_{i3}) \mid T_s'(\tau_0,n_{i4},n_{i5})\}$$

denotes the probability that the node τ_0 can not contribute to any spherical triangle which bounds a spherical triangular hole containing the point N but other three nodes can form a spherical triangle which bounds a spherical triangular hole containing the point N.

4.3.2 Case $0 < R_c \leq R\arccos([3\cos^2(R_s/R) - 1]/2)$

Theorem 4.8. *When $0 < R_c \leq R\arccos([3\cos^2(R_s/R) - 1]/2)$, $p_s(\lambda) = 0$.*

Proof. According to (4.1), when $0 < R_c \leq R\arccos([3\cos^2(R_s/R)-1]/2)$, $\mathcal{R}_{R_c}^{(2)}(\mathcal{V}) \subset \check{C}_{R_s}^{(2)}(\mathcal{V})$, it means that every 2-simplex in $\mathcal{R}_{R_c}^{(2)}(\mathcal{V})$ must be in $\check{C}_{R_s}^{(2)}(\mathcal{V})$, so there is no spherical triangular holes which means $p_s(\lambda) = 0$. □

4.3.3 Case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$

Theorem 4.9. *When $R\arccos([3\cos^2(R_s/R)-1]/2) < R_c \leq 2R_s$, we have $p_{sl}(\lambda) < p_s(\lambda) < p_{su}(\lambda)$, where*

$$\begin{aligned} p_{sl}(\lambda) = 2\pi\lambda^2 R^4 \int_{R_s/R}^{\theta_{0u}} \sin\theta_0 d\theta_0 \int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)} d\varphi_{s1} \int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})} \sin\theta_1 \\ \times e^{-\lambda|C(N,R\theta_0)|} e^{-\lambda|S^+(\theta_0,\varphi_{s1})|} (1 - e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|}) d\theta_1 \end{aligned} \tag{4.11}$$

and

$$\begin{aligned} p_{su}(\lambda) = 2\pi\lambda^2 R^4 \int_{R_s/R}^{\theta_{0u}} \sin\theta_0 d\theta_0 \int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)} d\varphi_{s1} \int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})} \sin\theta_1 \\ \times e^{-\lambda|C(N,R\theta_0)|} e^{-\lambda|S^+(\theta_0,\varphi_{s1})|} (1 - e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|}) d\theta_1 + p_s^{\mathrm{sec}}(\lambda) \end{aligned} \tag{4.12}$$

and

$$\theta_{0u} = \arccos\sqrt{[1+2\cos(R_c/R)]/3}$$
$$\varphi_m(\theta_0) = \arccos[(\cos(R_c/R) - \cos^2\theta_0)/\sin^2\theta_0]$$
$$\theta_{1u}(\theta_0,\varphi_{s1}) = \min\{\theta_{1u1}(\theta_0,\varphi_{s1}), \theta_{1u2}(\theta_0,\varphi_{s1})\}$$
$$\theta_{1u1}(\theta_0,\varphi_{s1}) = \arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2\varphi_{s1}}\right] + \arctan(\cos\varphi_{s1}\tan\theta_0)$$
$$\theta_{1u2}(\theta_0,\varphi_{s1}) = \arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2(\varphi_{s1}-\varphi_m(\theta_0))}\right] + \arctan(\cos(\varphi_{s1}-\varphi_m(\theta_0))\tan\theta_0)$$
$$|C(N,R\theta_0)| = 2\pi R^2(1-\cos\theta_0)$$
$$|S^+(\theta_0,\varphi_{s1})| = \int_{2\pi-\varphi_m(\theta_0)}^{\varphi_{s1}}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi)} R^2\sin\theta d\theta d\varphi$$
$$|S^-(\theta_0,\theta_1,\varphi_{s1})| = \int_{\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1})}^{\varphi_m(\theta_0)}\int_{\theta_0}^{\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)} R^2\sin\theta_2 d\theta_2 d\varphi_2$$
$$\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1}) = \varphi_{s1} - \arccos\frac{\cos(R_c/R)-\cos\theta_1\cos\theta_0}{\sin\theta_1\sin\theta_0}$$
$$\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2) = \min\{\theta_{1u1}(\theta_0,\varphi_2), \theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)\}$$
$$\theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2) = \arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2(\varphi_2-\varphi_{s1})}\right] + \arctan(\cos(\varphi_2-\varphi_{s1})\tan\theta_1)$$

$p_s^{\text{sec}}(\lambda)$ *is obtained by simulations.*

Proof. We first prove the lower bound. It can be obtained from (4.10) that

$$p_s(\lambda) > \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\setminus\{\tau_0(\Phi)\}} T_s(\tau_0,n_1,n_2)\}$$

So for the lower bound, we only consider the first case that the closest node τ_0 must contribute to a spherical triangle which bounds a spherical triangular hole containing the point N.

Using spherical coordinates, we assume the closest node τ_0 lies on $(R,\alpha_0,0)$ and use $|S|$ to denote the area of the set S, then we can get the distribution of α_0 as

$$F_{\alpha_0}(\theta_0) = P(\alpha_0 \le \theta_0) = 1 - e^{-\lambda|C(N,R\theta_0)|} \tag{4.13}$$

since the event $\alpha_0 > \theta_0$ means that the spherical cap $C(N,R\theta_0)$ does not contain any nodes from the process, which is given by the Poisson probability $e^{-\lambda|C(N,R\theta_0)|}$.

Furthermore, $|C(N, R\theta_0)|$ can be given as

$$|C(N, R\theta_0)| = \int_0^{\theta_0} \int_0^{2\pi} R^2 \sin\theta d\varphi d\theta = 2\pi R^2(1 - \cos\theta_0) \tag{4.14}$$

From (4.13) and (4.14), we can get the density of τ_0

$$F_{\alpha_0}(d\theta_0) = 2\pi\lambda R^2 \sin\theta_0 e^{-\lambda|C(N,R\theta_0)|} d\theta_0 \tag{4.15}$$

The integration range for θ_0 can be easily obtained. According to Lemma 4.7, we have $R_s < R\theta_0 \leq R \arccos\sqrt{[1 + 2\cos(R_c/R)]/3}$, so $R_s/R < \theta_0 \leq \theta_{0u} = \arccos\sqrt{[1 + 2\cos(R_c/R)]/3}$.

Therefore the probability of the first case can be given as

$$\begin{aligned} &\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi\backslash\{\tau_0(\Phi)\}} T_s(\tau_0, n_1, n_2)\} \\ &= \int_{R_s/R}^{\theta_{0u}} \mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{\theta_0}} T_s((R, \theta_0, 0), n_1, n_2)\} F_{\alpha_0}(d\theta_0) \end{aligned} \tag{4.16}$$

where Φ'_{θ_0} is the restriction of Φ in $C(N, R_c)\backslash C(N, R\theta_0)$.

Once the node τ_0 is determined, the other two nodes must lie in the different half spaces: one in $H^+ = \mathbb{R}^+ \times (0, \pi/2) \times (\pi, 2\pi)$ and the other in $H^- = \mathbb{R}^+ \times (0, \pi/2) \times (0, \pi)$. Assume n_1 lies in H^+ and n_2 lies in H^-. Since the great circle distance to τ_0 is at most R_c, n_1 and n_2 must also lie in the spherical cap $C(\tau_0, R_c)$. Furthermore, the great circle distance to the point N is at most R_c and larger than $R\alpha_0$, they should also lie in the region $A = C(N, R_c)\backslash C(N, R\alpha_0)$. Therefore, n_1 must lie in $G^+ = H^+ \bigcap C(\tau_0, R_c) \bigcap A$ and n_2 must lie in $G^- = H^- \bigcap C(\tau_0, R_c) \bigcap A$. In addition, considering the great circle distance between n_1 and n_2 should be at most R_c and the point N should be inside the spherical triangle formed by τ_0, n_1 and n_2, n_1 must lie in the shadow region $A^+ = G^+ \bigcap C(M_2, R_c)$, shown in Figure 4.4. M_1 and M_2 are two intersection points between bases of spherical caps $C(N, R\alpha_0)$ and $C(\tau_0, R_c)$, such intersection points must exist in this case since $R_c \leq 2R_s < 2R\alpha_0$.

Ordering the nodes in A^+ by increasing azimuth angle so that $\tau_1 = (R, \theta_1, \varphi_{s1})$ has the smallest azimuth angle φ_{s1}. And assume the nodes τ_0, τ_1 and another node $\tau_2 \in G^-$ can form a spherical triangle which bounds a spherical triangular hole containing the point N, then τ_2 must lie to the right of the great circle passing through τ_1 and N, denoted by $H^+(\varphi_{s1})$ which contains all points with azimuth angle $\varphi \in (\varphi_{s1} - \pi, \varphi_{s1})$. In addition, the great circle distance to τ_1 is no larger than R_c,

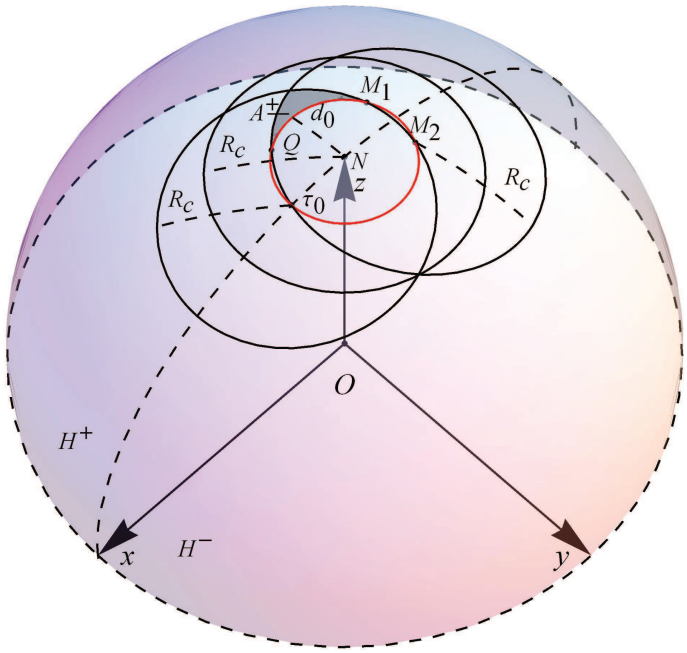


Figure 4.4: Illustration of region A^+ in the case $R\arccos([3\cos^2(R_s/R)-1]/2) < R_c \leq 2R_s$

so the node τ_2 must lie in the region S^-, as illustrated in Figure 4.5.

$$S^-(\tau_0,\tau_1) = S^-(\alpha_0,\theta_1,\varphi_{s1}) = G^- \bigcap H^+(\varphi_{s1}) \bigcap C(\tau_1,R_c)$$

Here we need to obtain the density of node τ_1. Considering the way τ_1 was defined, there should be no nodes with an azimuth angle less than φ_{s1} in A^+, that is to say no nodes are in the region

$$S^+(\tau_0,\tau_1) = S^+(\alpha_0,\varphi_{s1}) = A^+ \bigcap H^+(\varphi_{s1})$$

Since the intensity measure of the Poisson point process in spherical coordinates is $\lambda R^2 \sin\theta d\theta d\varphi$, the density F_{τ_1} of τ_1 can be given as

$$F_{\tau_1}(d\theta_1, d\varphi_{s1}) = \lambda R^2 \sin\theta_1 e^{-\lambda|S^+(\alpha_0,\varphi_{s1})|} d\theta_1 d\varphi_{s1} \tag{4.17}$$

Then we derive the integration domain $D(\alpha_0)$ with respect to parameters (θ_1,φ_{s1}). Assume the point M_2 has the spherical coordinate $(R,\alpha_0,\varphi_m), \varphi_m \in (0,\pi)$. Since the great circle distance between τ_0 and M_2 is R_c, then according to the spherical law of consines, we have $\cos\frac{R_c}{R} = \cos^2\alpha_0 + \sin^2\alpha_0\cos\varphi_m \Rightarrow \varphi_m(\alpha_0) = \arccos[(\cos\frac{R_c}{R} - \cos^2\alpha_0)/(\sin^2\alpha_0)]$. It can be seen that points M_1 and Q have the spherical coordinates $(R,\alpha_0,2\pi-\varphi_m(\alpha_0))$ and $(R,\alpha_0,2\varphi_m(\alpha_0))$ respectively, where Q is one intersec-

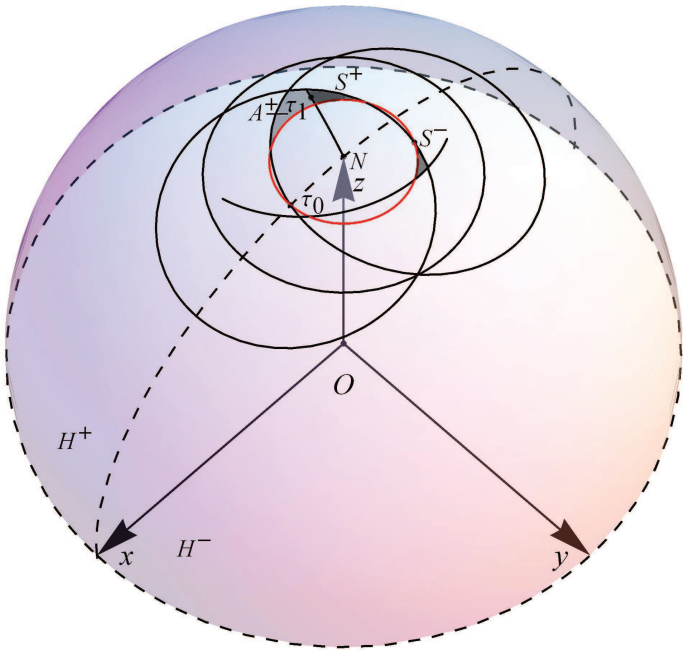


Figure 4.5: Illustration of regions S^+ and S^- in the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$

tion point between bases of spherical caps $C(N, R\alpha_0)$ and $C(M_2, R_c)$. Thus the integration range for φ_{s1} is $[2\pi - \varphi_m(\alpha_0), 2\varphi_m(\alpha_0)]$. In addition, assume any point with great circle distance R_c to τ_0 has the spherical coordinate (R, θ_t, φ_t), still using the spherical law of consines, we can obtain $\cos\frac{R_c}{R} = \cos\alpha_0\cos\theta_t + \sin\alpha_0\sin\theta_t\cos\varphi_t \Rightarrow \theta_t(\alpha_0, \varphi_t) = \arccos[\frac{\cos(R_c/R)}{\sqrt{1-\sin^2\alpha_0\sin^2\varphi_t}}] + \arctan(\cos\varphi_t\tan\alpha_0)$. Similarly, assume any point with great circle distance R_c to M_2 has the spherical coordinate $(R, \theta'_t, \varphi'_t)$, we have $\theta'_t(\alpha_0, \varphi'_t) = \arccos[\frac{\cos(R_c/R)}{\sqrt{1-\sin^2\alpha_0\sin^2(\varphi'_t-\varphi_m(\alpha_0))}}] + \arctan(\cos(\varphi'_t - \varphi_m(\alpha_0))\tan\alpha_0)$. Then the integration range for θ_1 is $[\theta_0, \theta_{1u}(\alpha_0, \varphi_{s1})]$, where $\theta_{1u}(\alpha_0, \varphi_{s1}) = \min\{\theta_{1u1}(\alpha_0, \varphi_{s1}), \theta_{1u2}(\alpha_0, \varphi_{s1})\}$, $\theta_{1u1}(\alpha_0, \varphi_{s1}) = \theta_t(\alpha_0, \varphi_{s1})$ and $\theta_{1u2}(\alpha_0, \varphi_{s1}) = \theta'_t(\alpha_0, \varphi_{s1})$. Furthermore, $|S^+(\alpha_0, \varphi_{s1})|$ can be expressed as

$$|S^+(\alpha_0, \varphi_{s1})| = \int_{2\pi-\varphi_m(\alpha_0)}^{\varphi_{s1}} \int_{\alpha_0}^{\theta_{1u}(\alpha_0,\varphi)} R^2\sin\theta d\theta d\varphi \tag{4.18}$$

Assume only τ_0, τ_1 and nodes in $S^-(\tau_0, \tau_1)$ can contribute to the spherical triangle which bounds a spherical triangular hole containing the point N, we can get a lower bound of the probability that the point N is inside a spherical triangular hole. It is a lower bound because it is possible that τ_1 can not contribute to a spherical triangle which bounds a spherical triangular hole containing point N, but some other nodes with higher azimuth angles in the region A^+ can contribute to such a spherical

triangle. For example, in Figure 4.6, if there is no node in S^- but there are some nodes in S'^-, then τ_1 can not contribute to any spherical triangle which bounds a spherical triangular hole containing point N, but τ_1' can form such a spherical triangle with τ_0 and another node in S'^-.

Based on the assumption, we have

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{\theta_0}} T_s((R,\theta_0,0),n_1,n_2)\}\\
>&\mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{\theta_0}\bigcap S^-(\tau_0,\tau_1)} T_s((R,\theta_0,0),\tau_1,n_2)\}\\
=&\iint_{D(\theta_0)}\mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{\theta_0}\bigcap S^-(\theta_0,\theta_1,\varphi_{s1})} T_s((R,\theta_0,0),(R,\theta_1,\varphi_{s1}),n_2)\}F_{\tau_1}(d\theta_1,d\varphi_{s1}) \qquad (4.19)\\
=&\iint_{D(\theta_0)}\mathrm{P}\{\Phi'_{\theta_0}(S^-(\theta_0,\theta_1,\varphi_{s1}))>0\}F_{\tau_1}(d\theta_1,d\varphi_{s1})\\
=&\iint_{D(\theta_0)}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})F_{\tau_1}(d\theta_1,d\varphi_{s1})
\end{aligned}$$

where $|S^-(\theta_0,\theta_1,\varphi_{s1})|$ can be expressed as

$$|S^-(\theta_0,\theta_1,\varphi_{s1})|=\int_{\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1})}^{\varphi_m(\theta_0)}\int_{\theta_0}^{\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)} R^2\sin\theta_2 d\theta_2 d\varphi_2 \qquad (4.20)$$

and

$$\begin{aligned}
\varphi_{2l}(\theta_0,\theta_1,\varphi_{s1})&=\varphi_{s1}-\arccos\frac{\cos(R_c/R)-\cos\theta_1\cos\theta_0}{\sin\theta_1\sin\theta_0}\\
\theta_{2u}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)&=\min\{\theta_{1u1}(\theta_0,\varphi_2),\theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)\}\\
\theta_{2u2}(\theta_0,\theta_1,\varphi_{s1},\varphi_2)&=\arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2(\varphi_2-\varphi_{s1})}\right]\\
&\quad+\arctan(\cos(\varphi_2-\varphi_{s1})\tan\theta_1)
\end{aligned}$$

Therefore, from (4.15), (4.16), (4.17) and (4.19), the lower bound shown in (4.11) can be derived.

Next we will prove the upper bound. As discussed in Section 4.3.1, there are two cases for the point N being inside a spherical triangular hole. As for the second case that the closest node τ_0 can not but some other nodes can contribute to a spherical triangle which bounds a spherical triangular hole containing the point N, we also get it by simulations as the case on plane. Simulation results show that this probability is less than 0.003% at any intensity λ when $R\arccos([3\cos^2(R_s/R)-1]/2)<R_c\leq 2R_s$. So we still focus on the probability of the first case.

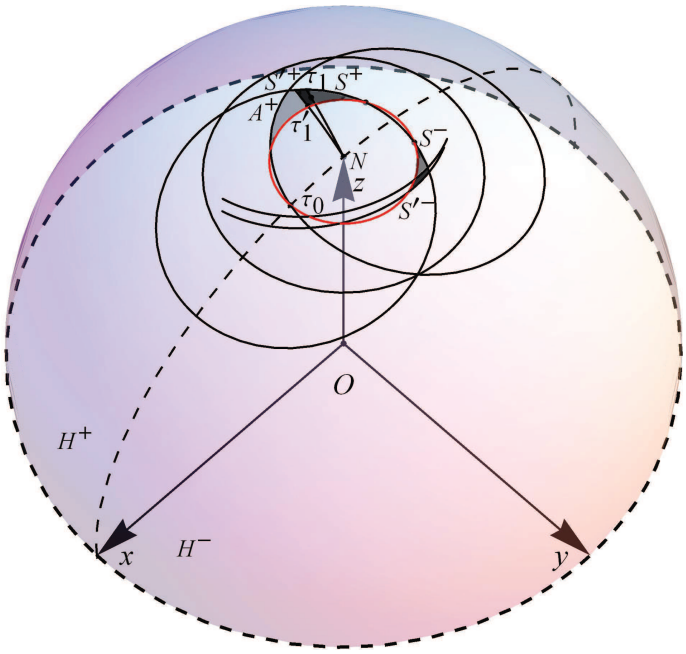


Figure 4.6: Illustration of regions S'^+ and S'^- in the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$

For the lower bound, we only considered the case that τ_1 contributes to a spherical triangle which bounds a spherical triangular hole containing point N. For the upper bound, we need to further consider the case that τ_1 can not but some other nodes in A^+ can contribute to such a spherical triangle, shown in Figure 4.6. Assume the node $\tau'_1 = (R, \theta'_1, \varphi'_{s1})$ with the second smallest azimuth angle in A^+ can contribute to such a spherical triangle, it means that there is no node in $S^-(\alpha_0, \theta_1, \varphi_{s1})$ but there is at least one node in the region $S'^-(\alpha_0, \theta_1, \varphi_{s1}, \theta'_1, \varphi'_{s1}) = S^-(\alpha_0, \theta'_1, \varphi'_{s1}) \backslash S^-(\alpha_0, \theta_1, \varphi_{s1})$.

Then the density of the pair (τ_1, τ'_1) is given as

$$F_{\tau_1,\tau'_1}(d\theta_1, d\varphi_{s1}, d\theta'_1, d\varphi'_{s1}) = \lambda^2 R^4 \sin\theta_1 \sin\theta'_1 e^{-\lambda|S^+(\alpha_0,\varphi'_{s1})|} d\theta_1 d\varphi_{s1} d\theta'_1 d\varphi'_{s1} \quad (4.21)$$

The probability that τ_1 can not but τ'_1 can form a spherical triangle which bounds a spherical triangular hole containing point N with τ_0 and another node

in $S'^{-}(\alpha_0, \theta_1, \varphi_{s1}, \theta'_1, \varphi'_{s1})$ can be given as

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_3,n_4\}\subseteq\Phi'_{\theta_0}\bigcap S^{-}(\tau_0,\tau'_1)} T_s((R,\theta_0,0),\tau'_1,n_4) \mid T'_s((R,\theta_0,0),\tau_1,n_3)\} \\
&= \iiiint \mathrm{P}\{\Phi'_{\theta_0}(S^{-}(\theta_0,\theta_1,\varphi_{s1}))=0\} \\
&\qquad \times \mathrm{P}\{\Phi'_{\theta_0}(S'^{-}(\theta_0,\theta_1,\varphi_{s1},\theta'_1,\varphi'_{s1}))>0\} F_{\tau_1,\tau'_1}(d\theta_1, d\varphi_{s1}, d\theta'_1, d\varphi'_{s1}) \\
&= \iiiint e^{-\lambda|S^{-}(\theta_0,\theta_1,\varphi_{s1})|} \times (1-e^{-\lambda|S'^{-}(\theta_0,\theta_1,\varphi_{s1},\theta'_1,\varphi'_{s1})|}) F_{\tau_1,\tau'_1}(d\theta_1, d\varphi_{s1}, d\theta'_1, d\varphi'_{s1})
\end{aligned} \tag{4.22}$$

As we can see from Figure 4.6, as long as τ'_1 has a higher polar angle than τ_1 has, the sum of $|S^{-}(\alpha_0,\theta_1,\varphi_{s1})|$ and $|S'^{-}(\alpha_0,\theta_1,\varphi_{s1},\theta'_1,\varphi'_{s1})|$ will be always smaller than $|S^{-}(\alpha_0,\alpha_0,\varphi_{s1})|$.

Therefore we can get from (4.22)

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{\{n_3,n_4\}\subseteq\Phi'_{\theta_0}\bigcap S^{-}(\tau_0,\tau'_1)} T_s((R,\theta_0,0),\tau'_1,n_4) \mid T'_s((R,\theta_0,0),\tau_1,n_3)\} \\
&< \iiiint (e^{-\lambda|S^{-}(\theta_0,\theta_1,\varphi_{s1})|} - e^{-\lambda|S^{-}(\theta_0,\theta_0,\varphi_{s1})|}) F_{\tau_1,\tau'_1}(d\theta_1, d\varphi_{s1}, d\theta'_1, d\varphi'_{s1})
\end{aligned} \tag{4.23}$$

Let $S'^{+}(\theta_0,\varphi_{s1},\varphi'_{s1}) = S^{+}(\theta_0,\varphi'_{s1}) \backslash S^{+}(\theta_0,\varphi_{s1})$, then

$$\iint \lambda R^2 \sin\theta'_1 e^{-\lambda|S'^{+}(\theta_0,\varphi_{s1},\varphi'_{s1})|} d\theta'_1 d\varphi'_{s1} = 1 - e^{-\lambda|A^{+}\backslash S^{+}(\theta_0,\varphi_{s1})|} < 1 \tag{4.24}$$

It is the complement of the probability that no node is in the region $A^{+}\backslash S^{+}(\theta_0,\varphi_{s1})$. From (4.19), (4.21), (4.23) and (4.24), we can obtain

$$\begin{aligned}
&\mathrm{P}\{\bigcup_{n_2\subseteq\Phi'_{\theta_0}\bigcap S^{-}(\tau_0,\tau_1)} T_s((R,\theta_0,0),\tau_1,n_2)\} \\
&+ \mathrm{P}\{\bigcup_{\substack{\{n_3,n_4\}\subseteq\Phi'_{\theta_0} \\ \bigcap S^{-}(\tau_0,\tau'_1)}} T_s((R,\theta_0,0),\tau'_1,n_4) \mid T'_s((R,\theta_0,0),\tau_1,n_3)\} \\
&< \iint_{D(\theta_0)} (1-e^{-\lambda|S^{-}(\theta_0,\theta_0,\varphi_{s1})|}) F_{\tau_1}(d\theta_1, d\varphi_{s1})
\end{aligned} \tag{4.25}$$

where $|S^{-}(\theta_0,\theta_0,\varphi_{s1})|$ has the similar expression as (4.20).

Similarly, we can further consider the case that neither of τ_1 and τ'_1 can contribute to a spherical triangle which bounds a spherical triangular hole containing point N, but other nodes with even higher azimuth angle can contribute to such a spherical triangle. In this way, we can get the same result as (4.25).

Therefore, it can be derived that

$$\begin{aligned}&\mathrm{P}\{\bigcup_{\{n_1,n_2\}\subseteq\Phi'_{\theta_0}} T_s((R,\theta_0,0),n_1,n_2)\}\\&<\iint_{D(\theta_0)}(1-e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|})F_{\tau_1}(d\theta_1,d\varphi_{s1})\end{aligned}\tag{4.26}$$

From (4.10), (4.15), (4.16), (4.21) and (4.26), the upper bound shown in (4.12) can be derived. □

4.3.4 Case $R_c > 2R_s$

Theorem 4.10. *When $R_c > 2R_s$, we have $p_{sl}(\lambda) < p_s(\lambda) < p_{su}(\lambda)$, where*

$$\begin{aligned}p_l(\lambda) = 2\pi\lambda^2R^4\Big\{&\int_{\frac{R_s}{R}}^{\frac{R_c}{2R}}\sin\theta_0 d\theta_0\int_{\pi}^{2\pi}d\varphi_{s1}\int_{\theta_0}^{\theta'_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1\\&\times e^{-\lambda|C(N,R\theta_0)|}e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})d\theta_1\\&+\int_{R_c/2R}^{\theta_{0u}}\sin\theta_0 d\theta_0\int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)}d\varphi_{s1}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 e^{-\lambda|C(N,R\theta_0)|}\\&\times e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})d\theta_1\Big\}\end{aligned}\tag{4.27}$$

and

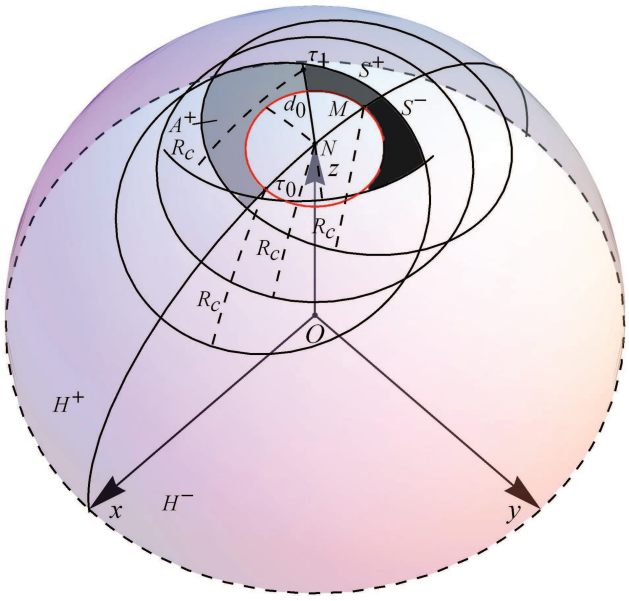
$$\begin{aligned}p_u(\lambda) = 2\pi\lambda^2R^4\Big\{&\int_{\frac{R_s}{R}}^{\frac{R_c}{2R}}\sin\theta_0 d\theta_0\int_{\pi}^{2\pi}d\varphi_{s1}\int_{\theta_0}^{\theta'_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1\\&\times e^{-\lambda|C(N,R\theta_0)|}e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|})d\theta_1\\&+\int_{R_c/2R}^{\theta_{0u}}\sin\theta_0 d\theta_0\int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)}d\varphi_{s1}\int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi_{s1})}\sin\theta_1 e^{-\lambda|C(N,R\theta_0)|}\\&\times e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_0,\varphi_{s1})|})d\theta_1\Big\}+p_s^{\mathrm{sec}}(\lambda)\end{aligned}\tag{4.28}$$

and

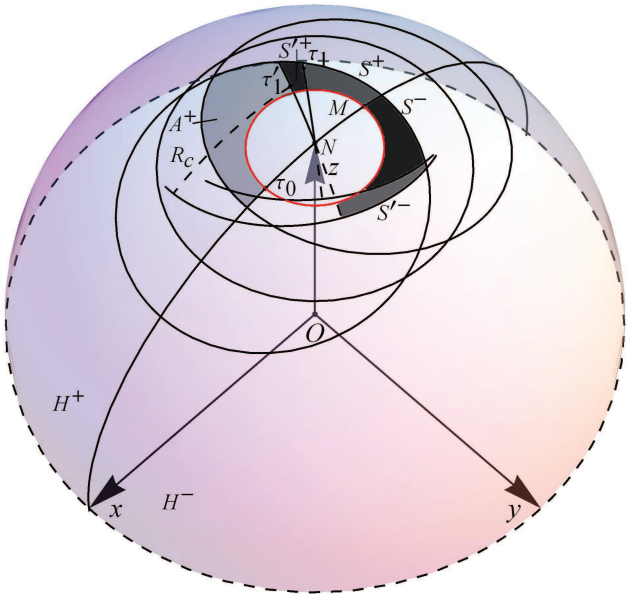
$$\begin{aligned}&\theta'_{1u}(\theta_0,\varphi_{s1}) = \min\{\theta_{1u1}(\theta_0,\varphi_{s1}),\theta'_{1u2}(\theta_0,\varphi_{s1})\}\\&\theta'_{1u2}(\theta_0,\varphi_{s1}) = \arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2\varphi_{s1}}\right]-\arctan(\cos\varphi_{s1}\tan\theta_0)\end{aligned}$$

$p_s^{\mathrm{sec}}(\lambda)$ is obtained by simulations.

We can use the same method as in Section 4.3.3 to get the lower and upper bounds, shown in (4.27) and (4.28) respectively. But we need to consider two situations $R_s/R < \theta_0 \le R_c/(2R)$ and $R_c/(2R) < \theta_0 \le \theta_{0u} = \arccos\sqrt{[1+2\cos(R_c/R)]/3}$.



(a)



(b)

Figure 4.7: Illustrations of regions in case $R_c > 2R_s$. (a) the regions A^+, S^+ and S^-, (b) the regions S'^+ and S'^-

In the first situation, $\theta_0 \leq R_c/(2R)$ means that the spherical cap $C(N, R\theta_0)$ is completely included in the spherical cap $C(\tau_0, R_c)$. The illustrations for the regions A^+, S^+, S^-, S'^+ and S'^- are shown in Figure 4.7(a) and 4.7(b) respectively. In addition, the integration range for φ_{s1} is $[\pi, 2\pi]$. The second situation is the same as that in Section 4.3.3. Furthermore, simulation results show that $p_s^{\text{sec}}(\lambda)$ is less than 0.16% at any intensity when $2R_s < R_c \leq 3$ and $R_s \ll R$, $R_c \ll R$.

4.3.5 Case $R \to \infty$

Intuitively, when $R \to \infty$, the cases on sphere should be the same as those on plane, which will be proved in this section. We choose the case $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ on sphere and the case $\sqrt{3} < \gamma = R_c/R_s \leq 2$ on plane for proof. For other cases, the proof is similar. For convenience, let $\theta_0 = r_0/R, \theta_1 = r_1/R, \varphi_{s1} = \pi + \varphi_1$.

Lemma 4.11.

$$\lim_{R\to\infty} R\arccos([3\cos^2(R_s/R) - 1]/2) = \sqrt{3}R_s$$

Proof. According to l'Hôpital's rule, we have

$$\begin{aligned}
&\lim_{R\to\infty} R\arccos([3\cos^2(R_s/R) - 1]/2) \\
=&R_s \lim_{x\to 0} \frac{\arccos([3\cos^2(x) - 1]/2)}{x} \qquad (\text{let } x = R_s/R) \\
=&R_s \lim_{x\to 0} \frac{3\cos x \sin x}{\sqrt{1 - ([3\cos^2(x) - 1]/2)^2}} \\
=&R_s \lim_{x\to 0} \frac{6\cos x \sin x}{\sqrt{(3 - 3\cos^2 x)(1 + 3\cos^2 x)}} = \sqrt{3}R_s
\end{aligned}$$

□

Lemma 4.11 means that when $R \to \infty$, the condition $R\arccos([3\cos^2(R_s/R) - 1]/2) < R_c \leq 2R_s$ is equivalent to the condition $\sqrt{3} < \gamma = R_c/R_s \leq 2$.

Lemma 4.12.

$$\lim_{R\to\infty} R\theta_{0u} = \lim_{R\to\infty} R\arccos\sqrt{[1 + 2\cos(R_c/R)]/3} = R_c/\sqrt{3}$$

Proof. According to l'Hôpital's rule, we have

$$\begin{aligned}
&\lim_{R\to\infty} R\arccos\sqrt{[1 + 2\cos(R_c/R)]/3} \\
=&R_c \lim_{x\to 0} \frac{\arccos\sqrt{[1 + 2\cos(x)]/3}}{x} \qquad (\text{let } x = R_c/R) \\
=&R_c \lim_{x\to 0} \frac{1}{\sqrt{3 - [1 + 2\cos(x)]}} \frac{\sin x}{\sqrt{1 + 2\cos(x)}} \\
=&R_c/\sqrt{3} \lim_{x\to 0} \frac{\sin x}{\sqrt{2 - 2\cos x}} = R_c/\sqrt{3}
\end{aligned}$$

□

Lemma 4.13. *Let $\theta_0 = r_0/R$, then we have*

$$\begin{aligned}\lim_{R\to\infty} \varphi_m(\theta_0) &= \lim_{R\to\infty} \arccos[(\cos(R_c/R) - \cos^2\theta_0)/\sin^2\theta_0] \\ &= \pi - 2\arccos(R_c/(2r_0)) = \pi - \varphi_l(r_0)\end{aligned}$$

where $\varphi_l(r_0)$ is shown in (3.5).

Proof. According to l'Hôpital's rule, we have

$$\begin{aligned}&\lim_{R\to\infty} \arccos[(\cos(R_c/R) - \cos^2\theta_0)/\sin^2\theta_0] \\ =& \arccos(\lim_{R\to\infty} (\cos(R_c/R) - \cos^2(r_0/R))/\sin^2(r_0/R)) \\ =& \arccos(\lim_{R\to\infty} \frac{R_c/R^2\sin(R_c/R) - 2r_0/R^2\sin(r_0/R)\cos(r_0/R)}{-2r_0/R^2\sin(r_0/R)\cos(r_0/R)}) \\ =& \arccos(\lim_{R\to\infty} 1 - \frac{R_c\sin(R_c/R}{2r_0\sin(r_0/R)}) \\ =& \arccos(1 - R_c^2/(2r_0^2))\end{aligned}$$

Meanwhile, we have $\cos(\pi - 2\arccos(R_c/(2r_0))) = 1 - R_c^2/(2r_0^2)$ and $0 \le \pi - 2\arccos(R_c/(2r_0)) \le \pi$, so $\pi - 2\arccos(R_c/(2r_0)) = \arccos(1 - R_c^2/(2r_0^2))$.

As from (3.5), we know $\varphi_l(r_0) = 2\arccos(R_c/(2r_0))$, Consequently

$$\begin{aligned}\lim_{R\to\infty} \arccos[(\cos(R_c/R) - \cos^2\theta_0)/\sin^2\theta_0] &= \pi - 2\arccos(R_c/(2r_0)) \\ &= \pi - \varphi_l(r_0)\end{aligned}$$

□

Lemma 4.14. *Let $\theta_0 = r_0/R, \varphi_{s1} = \pi + \varphi_1$, we have*

$$\begin{aligned}\lim_{R\to\infty} R\theta_{1u1}(\theta_0, \varphi_{s1}) &= \lim_{R\to\infty} R(\arccos\Big[\frac{\cos(R_c/R)}{\sqrt{1 - \sin^2\theta_0\sin^2\varphi_{s1}}}\Big] + \arctan(\cos\varphi_{s1}\tan\theta_0)) \\ &= \sqrt{R_c^2 - r_0^2\sin^2\varphi_1} - r_0\cos\varphi_1\end{aligned}$$

Proof. Still using l'Hôpital's rule, we have

$$
\begin{aligned}
&\lim_{R\to\infty} R(\arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2\theta_0\sin^2\varphi_{s1}}\right]+\arctan(\cos\varphi_{s1}\tan\theta_0))\\
=&\lim_{R\to\infty}\frac{\arccos\left[\cos(R_c/R)/\sqrt{1-\sin^2(r_0/R)\sin^2\varphi_{s1}}\right]}{1/R}+r_0\cos\varphi_{s1}\\
=&\lim_{R\to\infty}\frac{R^2}{\sqrt{1-\frac{\cos^2(R_c/R)}{1-\sin^2(r_0/R)\sin^2\varphi_{s1}}}}(\frac{R_c}{R^2}\sin\frac{R_c}{R}-\frac{r_0}{R^2}\cos\frac{R_c}{R}\sin\frac{r_0}{R}\cos\frac{r_0}{R}\sin^2\varphi_{s1})+r_0\cos\varphi_{s1}\\
=&\lim_{R\to\infty}\frac{R_c\sin(R_c/R)-r_0\sin(r_0/R)\sin^2\varphi_{s1}}{\sqrt{\sin^2(R_c/R)-\sin^2(r_0/R)\sin^2\varphi_{s1}}}+r_0\cos\varphi_{s1}\\
=&\lim_{R\to\infty}\frac{R_c^2-r_0^2\sin^2\varphi_{s1}}{R\sqrt{\sin^2(R_c/R)-\sin^2(r_0/R)\sin^2\varphi_{s1}}}+r_0\cos\varphi_{s1}\\
=&\sqrt{R_c^2-r_0^2\sin^2\varphi_{s1}}+r_0\cos\varphi_{s1}=\sqrt{R_c^2-r_0^2\sin^2\varphi_1}-r_0\cos\varphi_1
\end{aligned}
$$

□

According to Lemma 4.13 and 4.14, we can get the following lemma.

Lemma 4.15. *Let $\theta_0=r_0/R,\varphi_{s1}=\pi+\varphi_1$, we have*

$$
\begin{aligned}
&\lim_{R\to\infty} R\theta_{1u2}(\theta_0,\varphi_{s1})\\
=&\lim_{R\to\infty} R(\arccos\left[\frac{\cos(R_c/R)}{\sqrt{1-\sin^2\theta_0\sin^2(\varphi_{s1}-\varphi_m(\theta_0))}}\right]+\arctan(\cos(\varphi_{s1}-\varphi_m(\theta_0))\tan\theta_0))\\
=&\sqrt{R_c^2-r_0^2\sin^2(\varphi_1+\varphi_l(r_0))}+r_0\cos(\varphi_1+\varphi_l(r_0))
\end{aligned}
$$

According to Lemma 4.14 and 4.15, we obtain the lemma as follows.

Lemma 4.16. *Let $\theta_0=r_0/R,\varphi_{s1}=\pi+\varphi_1$, we have*

$$\lim_{R\to\infty} R\theta_{1u}(\theta_0,\varphi_{s1})=\lim_{R\to\infty} R\min\{\theta_{1u1}(\theta_0,\varphi_{s1}),\theta_{1u2}(\theta_0,\varphi_{s1})\}=R_1(r_0,\varphi_1)$$

where $R_1(r_0,\varphi_1)$ is shown in (3.7).

Lemma 4.17. *Let $\theta_0=r_0/R,\varphi_{s1}=\pi+\varphi_1$, we have*

$$\lim_{R\to\infty}|S^+(\theta_0,\varphi_{s1})|=|S^+(r_0,\varphi_1)|$$

where $|S^+(\theta_0,\varphi_{s1})|$ and $|S^+(r_0,\varphi_1)|$ are shown in (4.18) and (3.18).

Proof. From (4.18), we have

$$|S^+(\theta_0, \varphi_{s1})| = \int_{2\pi-\varphi_m(\theta_0)}^{\varphi_{s1}} \int_{\theta_0}^{\theta_{1u}(\theta_0,\varphi)} R^2 \sin\theta d\theta d\varphi$$

Let $\theta = r/R$ and $\varphi = \pi + \varphi'$, then use Lemma 4.13 and 4.16, we get

$$\begin{aligned}\lim_{R\to\infty} |S^+(\theta_0, \varphi_{s1})| &= \lim_{R\to\infty} \int_{\pi-\varphi_m(\theta_0)}^{\varphi_{s1}-\pi} \int_{R\theta_0}^{R\theta_{1u}(\theta_0,\pi+\varphi')} R\sin(r/R) dr d\varphi' \\ &= \int_{\varphi_l(r_0)}^{\varphi_1} \int_{r_0}^{R_1(r_0,\varphi')} r dr d\varphi' = |S^+(r_0, \varphi_1)|\end{aligned}$$

□

Similarly, we can get the following lemma.

Lemma 4.18. *Let $\theta_0 = r_0/R, \theta_1 = r_1/R, \varphi_{s1} = \pi + \varphi_1$, we have*

$$\lim_{R\to\infty} |S^-(\theta_0, \theta_1, \varphi_{s1})| = |S^-(r_0, r_1, \varphi_1)|$$

where $|S^-(\theta_0, \theta_1, \varphi_{s1})|$ and $|S^-(r_0, r_1, \varphi_1)|$ are shown in (4.20) and (A.1).

From the lemmas above, we can get the theorem as follows.

Theorem 4.19.

$$\lim_{R\to\infty} p_{sl}(\lambda) = p_{2dl}(\lambda)$$

where $p_{sl}(\lambda)$ and $p_{2dl}(\lambda)$ are shown in (3.3) and (4.11).

Proof. Let $\theta_0 = r_0/R, \theta_1 = r_1/R, \varphi_{s1} = \pi + \varphi_1$.

First, from (4.14), it is easy to check

$$\lim_{R\to\infty} |C(N, R\theta_0)| = \lim_{R\to\infty} 2\pi R^2(1 - \cos(r_0/R)) = \pi r_0^2$$

From (3.5) and (3.6), we can get $\varphi_u(r_0) = \pi - 2\varphi_l(r_0)$.

Then, using the lemmas above, we can obtain from (4.11)

$$\begin{aligned}\lim_{R\to\infty} p_{sl}(\lambda) &= \lim_{R\to\infty} 2\pi\lambda^2 \int_{R_s}^{R\theta_{0u}} R\sin\frac{r_0}{R} dr_0 \int_{2\pi-\varphi_m(\theta_0)}^{2\varphi_m(\theta_0)} d\varphi_{s1} \int_{R\theta_0}^{R\theta_{1u}(\theta_0,\varphi_{s1})} R\sin\frac{r_1}{R} \\ &\quad \times e^{-\lambda\pi r_0^2} e^{-\lambda|S^+(\theta_0,\varphi_{s1})|}(1-e^{-\lambda|S^-(\theta_0,\theta_1,\varphi_{s1})|})dr_1 \\ &= 2\pi\lambda^2 \int_{R_s}^{R_c/\sqrt{3}} r_0 dr_0 \int_{\varphi_l(r_0)}^{\varphi_u(r_0)} d\varphi_1 \int_{r_0}^{R_1(r_0,\varphi_1)} e^{-\lambda\pi r_0^2} \\ &\quad \times e^{-\lambda|S^+(r_0,\varphi_1)|}(1-e^{-\lambda|S^-(r_0,r_1,\varphi_1)|})r_1 dr_1 \\ &= p_{2dl}(\lambda)\end{aligned}$$

□

The upper bound can be proved similarly.

4.4 Performance evaluation

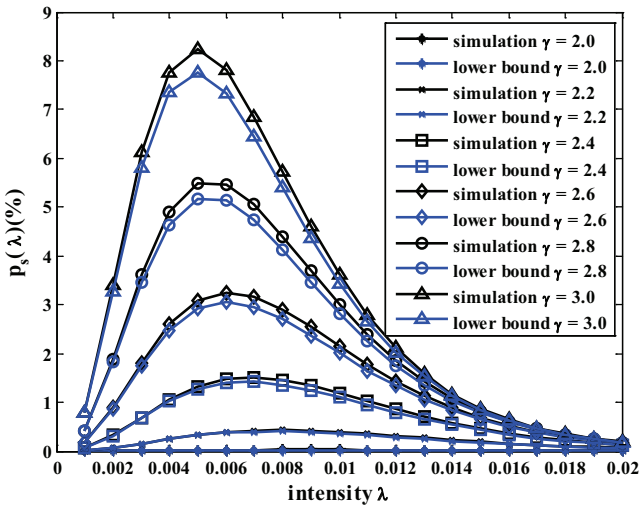
In this section, simulation settings are first given. Then simulation results are compared with analytical lower and upper bounds under different settings of R_s, R_c, R.

4.4.1 Simulation settings

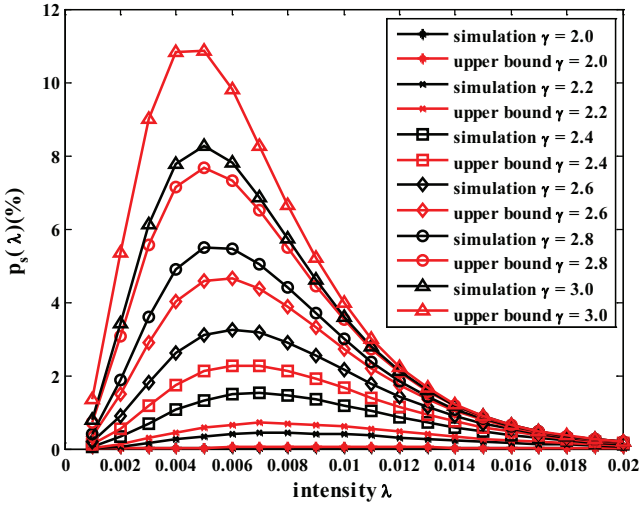
A sphere centered at the origin with radius R is considered in the simulations. The probability of the point with spherical coordinate $(R, 0, 0)$ being inside a spherical triangular hole is computed. Sensors are randomly distributed on the sphere according to a homogeneous Poisson point process with intensity λ. The sensing radius R_s of each node is set to be 10 meters and communication radius R_c is chosen from 20 to 30 meters with interval of 2 meters. Let $\gamma = R_c/R_s$, then γ ranges from 2 to 3 with interval of 0.2. In addition, λ is selected from 0.001 to 0.020 with interval of 0.001. For each pair of (λ, γ), 10^7 simulations are run to check whether the point with spherical coordinate $(R, 0, 0)$ is inside a spherical triangular hole.

4.4.2 Impact of R_s and R_c

As illustrated in Section 4.2, $R_s \ll R$ and $R_c \ll R$, here we choose $R = 10R_s$ to analyse the impact of R_s and R_c on the probability of any point being inside a spherical triangular hole. Under this configuration, the probability $p_s(\lambda)$ obtained by simulations is presented with the lower and upper bounds in Figure 4.8(a) and 4.8(b) respectively.



(a)



(b)

Figure 4.8: Proportion of the area of spherical triangular holes under $R = 10R_s$ (a) simulation results and lower bounds, (b) simulation results and upper bounds

It can be seen that for any value of γ, $p_s(\lambda)$ has a maximum at a threshold value λ_c of the intensity. As a matter of fact, for $\lambda \leq \lambda_c$, the number of nodes is small. Consequently the probability of any point being inside a spherical triangular hole is relatively small too. With the increase of λ, the connectivity between nodes becomes stronger. As a result, the probability of any point being inside a spherical triangular hole increases. However, when the intensity reaches the threshold value, the probability is up to its maximum. $p_s(\lambda)$ decreases for $\lambda \geq \lambda_c$. The simulations also show that λ_c decreases with the increase of γ.

On the other hand, it can be seen from Figure 4.8(a) and 4.8(b) that for a fixed

intensity λ, $p_s(\lambda)$ increases with the increases of γ. That is because when R_s is fixed, the larger R_c is, the higher is the probability of each spherical triangle containing a coverage hole.

Furthermore, the maximum probability increases quickly with γ ranging from 2.0 to 3.0. These results can also provide some insights for planning of WSNs, which will be discussed in Section 4.4.4.

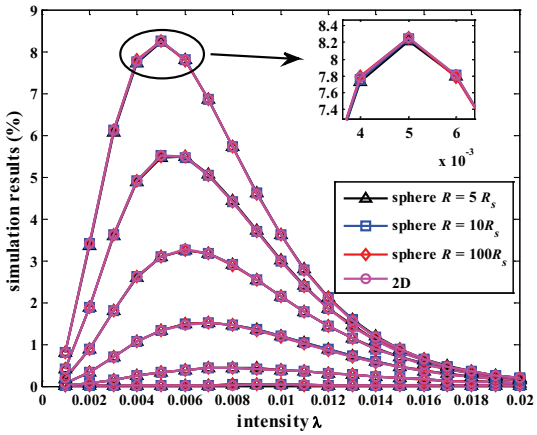
Finally, it can be found in Figure 4.8(a) that the probability obtained by simulations is very well consistent with the lower bound. The maximum difference between them is about 0.5%. Figure 4.8(b) shows that probability obtained by simulations is also consistent with the upper bound. The maximum difference between them is about 3%.

4.4.3 Impact of R

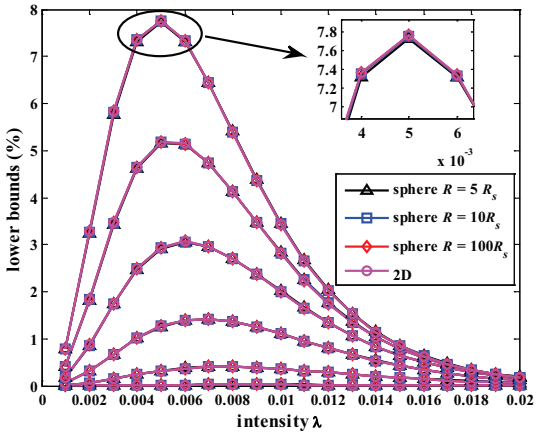
Although we assume $R_s \ll R$ and $R_c \ll R$, to better understand the impact of R on the probability of any point being inside a spherical triangular hole, we choose R to be $5R_s, 10R_s$ and $100R_s$. In addition, we also want to know the difference of the probability under spherical and 2D planar cases. Therefore, simulation results, lower and upper bounds of the probability under spheres with radii $5R_s, 10R_s, 100R_s$ and 2D plane are shown in Figure 4.9(a), 4.9(b) and 4.9(c) respectively. Simulation results for $p_s^{\text{sec}}(\lambda)$ under spheres with radii $5R_s, 10R_s, 100R_s$ and 2D plane are shown in Figure 4.10. The detailed values are presented in Appendix B.

It can be seen from Figure 4.9 that simulations results, lower and upper bounds under spheres with radii $5R_s, 10R_s, 100R_s$ and 2D plane are very close with each other. More precisely, the maximum difference of simulations results under spheres with radii $5R_s$ and $10R_s$ is about 0.045%, which is about 0.06% under spheres with radii $5R_s$ and $100R_s$ and is about 0.03% under spheres with radii $10R_s$ and $100R_s$. In addition, the maximum differences of simulation results between 2D planar case and spherical cases with radii $5R_s, 10R_s, 100R_s$ are 0.05%, 0.03% and 0.02% respectively. It means the larger the radius of sphere is, the more closer are the simulation results under sphere and 2D plane, it is because the larger the radius of sphere is, the more likely of the local of each node on the sphere to be planar.

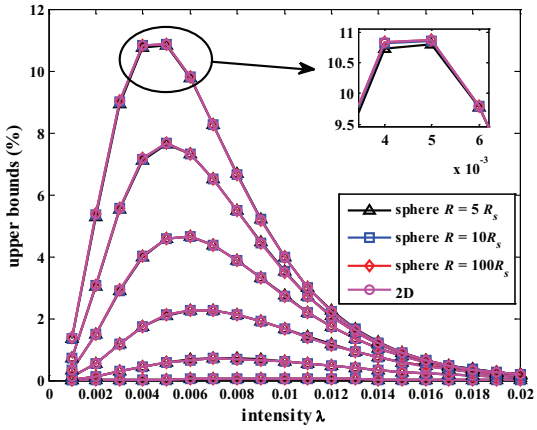
With respect to lower and upper bounds, it is found that under any two spheres with radii $5R_s, 10R_s, 100R_s$, the maximum difference of lower and upper bounds are 0.06% and 0.12% respectively. Furthermore, under spheres with radii $5R_s, 10R_s, 100R_s$ and 2D plane, the maximum difference of lower bounds is also 0.06%, and that of



(a)



(b)



(c)

Figure 4.9: Comparison of the proportion of the area of spherical triangular holes (a) comparison of simulation results, (b) comparison of lower bounds, (c) comparison of upper bounds

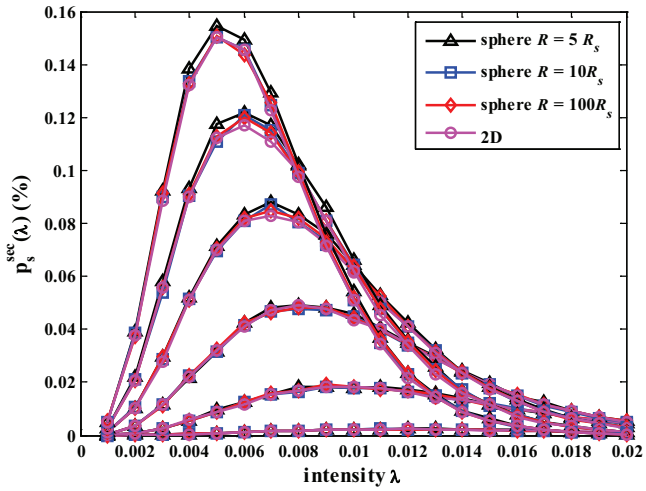


Figure 4.10: Simulation results for $p_s^{\text{sec}}(\lambda)$

upper bounds is also 0.12%. More importantly, under sphere with radius $100R_s$ and 2D plane, the maximum difference of lower bounds is 5×10^{-6} and that of upper bounds is 2.5×10^{-5}. It means the probabilities under cases of sphere with radius $100R_s$ and 2D plane are nearly the same, which is quite logical since when the radius of sphere is much more larger than the sensing radius of any node, the local of any node can be considered to be planar.

It can be further found that under above cases, the maximum differences of simulation results, lower and upper bounds are all so small that they can be neglected. Consequently, it also means that the radius of sphere has little impact on the probability of any point on the sphere to be inside a spherical triangular hole.

4.4.4 Discussions on applications

Similar to the case on plane, our analytical results can also be used for planning of WSNs. For example, a WSN is used to monitor a mountain and the ratio $\gamma = 2$, according to the analytical upper bounds, we can see that the maximum proportion of the area of spherical triangular holes under $\gamma = 2$ is about 0.06 %, which can be neglected. It means that as long as the surface of mountain can be spherically triangulated by nodes, we can say the mountain is covered. But if $\gamma = 3$ and at least 95% of the surface of the mountain should be covered, then it means that the proportion of the area of spherical triangular holes can be at most 5%. From the analytical upper bounds of $\gamma = 3$, it can be seen that when the intensity $\lambda = 0.009$, the upper bound is about 5%, so in order to cover at least 95% of the mountain, the

intensity of nodes should be larger than 0.009. Furthermore, our results can also be used in the scenarios when using satellite to cover the whole earth.

4.5 Chapter summary

In this chapter, we consider the accuracy of homology based coverage hole detection on sphere. First, the relationship between Čech complex and Rips complex on sphere is derived, which is different from that on plane. After that, the situations when Rips complex may miss coverage holes are identified and we find that the holes missed by Rips complex are always bounded by a spherical triangle. So we define them to be spherical triangular holes and use the proportion of the area of spherical triangular holes to evaluate the accuracy of homology based coverage hole detection on sphere. We consider a homogeneous case that nodes are randomly distributed on a sphere according to a homogeneous Poisson point process. Under such a setting and without considering border effect, the proportion of the area of spherical triangular holes is equivalent to the probability of any point on the sphere being inside a spherical triangular hole. Based on that, three different cases are considered and for each case, the closed form expressions for the lower and upper bounds of the probability are derived. Simulation results are well consistent with the analytical bounds. More importantly, it is shown that the radius of sphere has little impact on the probability as long as it is much larger than communication and sensing radii of each sensor.

Chapter 5

Graph based Distributed Coverage Hole Detection

5.1 Introduction

For triangular holes, we have computed the proportion of their area to evaluate the accuracy of homology based coverage hole detection approaches in Chapter 3. For non-triangular holes, we aim to design algorithms to detect them. As discussed in Section 2.1.3, graph based approaches usually only detect boundary nodes without discovering boundary cycles. In this chapter, we aim to design a graph based algorithm to find boundary cycles of non-triangular coverage holes in a WSN. In this algorithm, we design a method to detect boundary nodes by checking whether there exists a Hamiltonian cycle in their neighbour graphs. After that, some nodes are randomly selected to initiate the process to find boundary cycles. It is possible that some of the found cycles are not minimum or bound the same coverage holes, so it is necessary to make some choices among these cycles. Comparing our algorithm with a location based approach, we find that our algorithm can accurately discover the boundary cycles of more than 95% coverage holes when the intensity is low.

5.2 Models and assumptions

Consider a collection of stationary sensors deployed randomly on a planar target field. As usual, isotropic radio propagation is assumed. Each sensor monitors a region within a circle of radius R_s and may communicate with other sensors within a circle of radius R_c.

In addition, some other assumptions are as follows.

1. There are sensors located on the external boundary of the target field. They are known as fence sensors and other sensors are referred to as internal sensors. Each fence sensor has two fence neighbours. This is also the general assumption in many homology based algorithms [38, 42, 82, 86].

2. Although sensors are not aware of their locations, every sensor can know whether it is a fence or an internal node by using the mechanisms presented in [44] or other methods as in [45]. In fact, it is a conventional assumption adopted by many existing range based methods [44, 98] or connectivity based methods [45, 86].

3. Internal sensors are distributed on the planar target field according to a homogeneous Poisson point process with intensity λ.

4. Each sensor has an unique ID.

5. The network has only one connected component.

Base on these assumptions, we can use a graph $G(V, E)$ to represent the WSN, where V denotes all the nodes of WSN, for any two nodes, if they can communicate with each other, an edge connects them in the graph. For any node $v_i \in V$, its neighbour graph is the sub-graph of $G(V, E)$ induced by all its neighbours. In addition, we give the definition of equivalent transformation which will be used in the algorithm and the performance evaluation part.

Definition 5.1 (Equivalent transformation)**.** *An equivalent transformation is a sequential combination of vertex insertion, deletion or replacement as follows:*

- *vertex insertion. For a cycle* $< v_1, v_2, \ldots, v_i, v_{i+1}, \ldots, v_k >$*, a vertex* v' *can be inserted in the cycle if* v' *is a neighbour of* v_i *and* v_{i+1} *in the cycle.*

- *vertex deletion. For a cycle* $< v_1, v_2, \ldots, v_k >$*, a vertex* v_i *in the cycle can be deleted if its two neighbouring vertices in the cycle are neighbours.*

- *vertex replacement. For a cycle* $< v_1, v_2, \ldots, v_k >$*, a vertex* v_i *in the cycle can be replaced by* v' *if* v' *is a neighbour of* v_i *and its two neighbours or* v', v_i *and two neighbours of* v_i *have at least one common neighbour.*

Figure 5.1 presents an example showing vertex insertion, deletion and replacement.

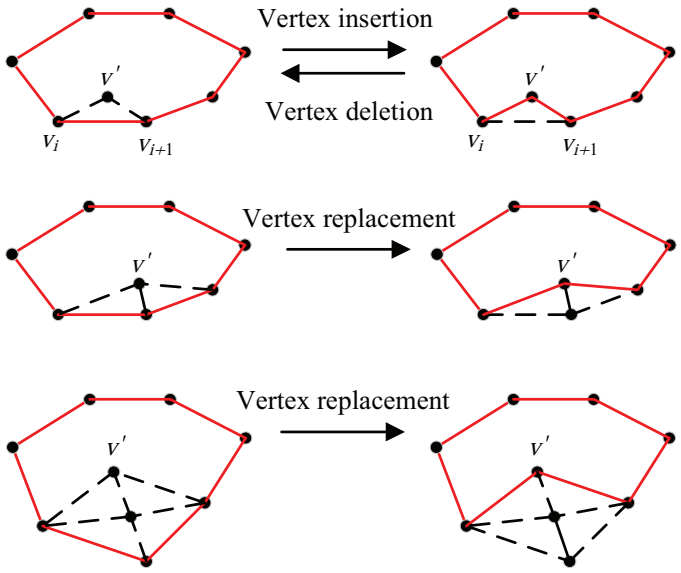


Figure 5.1: Examples of equivalent transformations

5.3 Graph based distributed algorithm

In this section, we will propose a graph based distributed algorithm to find boundary cycles of coverage holes in WSNs. The general steps of the algorithm are given first and the details of each step are presented in the following parts.

The entire process of our algorithm can be summarized as follows.

1. Each sensor gets its 1- and 2-hop neighbours information and constructs a neighbour graph. Then the Rips complex of the WSN can be constructed, as shown in Figure 5.2(a).

2. Based on the neighbour graph, each node can determine whether it can be a boundary node or not by checking whether there exists a Hamiltonian cycle in its neighbour graph. If there exists, the node is assumed to be a non-boundary node. Otherwise, it is assumed to be a boundary node. The results are shown in Figure 5.2(b∼c).

3. When each node has determined whether it is a boundary node or not, it can broadcast this information to its neighbours. Then each node knows the status of its neighbours. After that, some nodes can initiate boundary cycles discovery process to find cycles bounding holes, such as the nodes denoted by green diamond in Figure 5.2(d∼e).

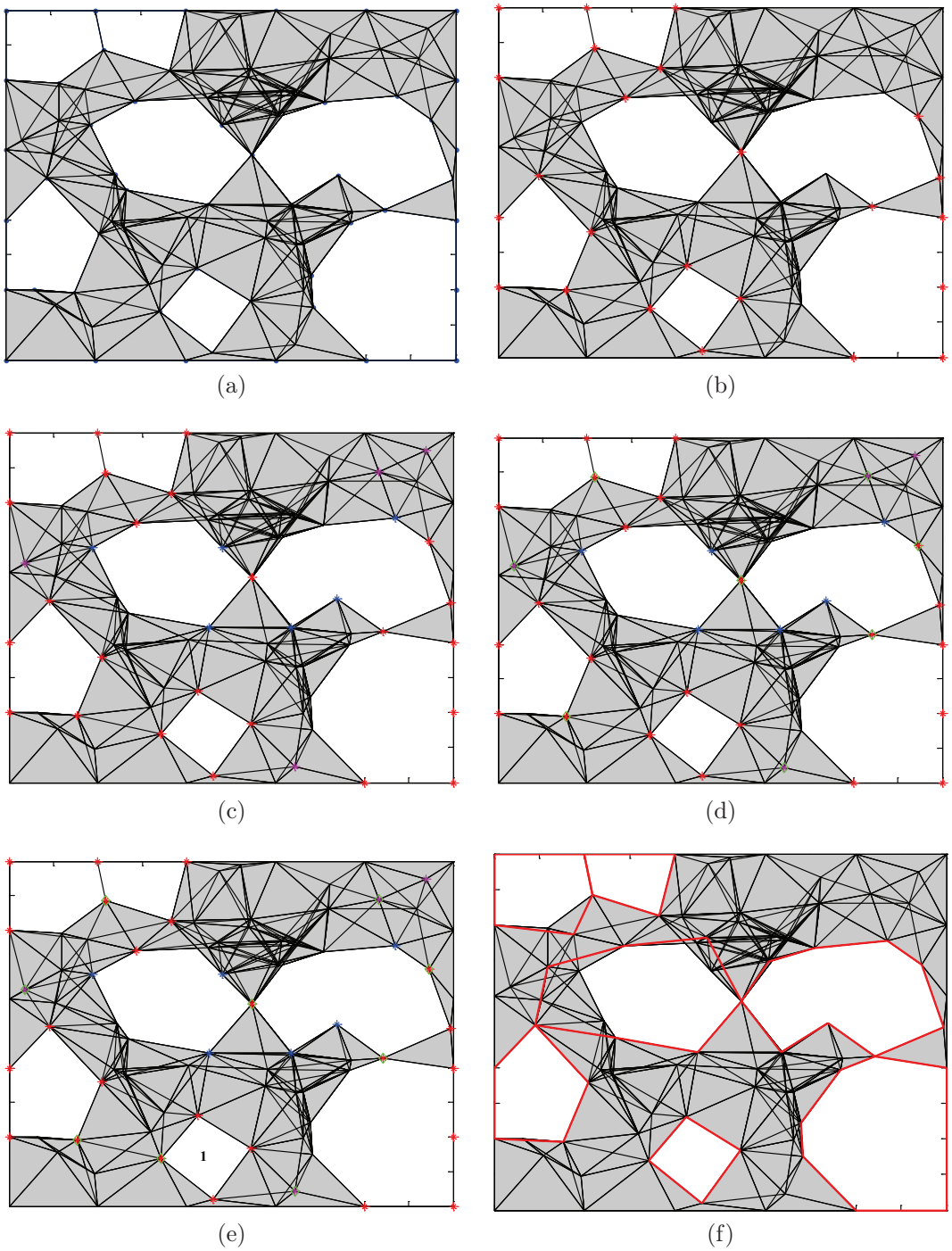


Figure 5.2: Procedures of graph based distributed algorithm. (a) Rips complex of a WSN, (b∼c) boundary nodes discovery, (d∼e) boundary cycles discovery, (f) cycles selection

4. It is possible that some cycles found in step 3 are not minimum or some cycles bound the same hole. So it is necessary to minimize and make choices among

all cycles found. The result is shown in Figure 5.2(f).

Each step is described in more detail in the following parts.

5.3.1 Neighbour discovery

In this step, each node needs to obtain all its 1- and 2-hop neighbours information. This can be easily achieved by two broadcasts of hello message. In the first one, each node broadcasts its id. When it gets all the ids of its 1-hop neighbours, each node continues to broadcast a hello message containing the ids of its 1-hop neighbours. After receiving the neighbour list of its neighbours, each node can construct its neighbour graph. Assume ID_i denotes the id of i-th node, $G_i(V_i, E_i)$ is the neighbour graph of node ID_i, where V_i denotes the set containing ids of its 1-hop neighbours and E_i is the set containing all the edges between its 1-hop neighbours. Notice that ID_i is not included in V_i. After that, the Rips complex corresponding to the WSN can be constructed. As the WSN is on a planar target field, it is sufficient to construct a 2-dimensional Rips complex of the WSN in order to find coverage holes. All 0- and 1-simplices can be easily obtained. As for 2-simplices, any three nodes which are neighbours of each other can form a 2-simplex. A Rips complex can be built according to this rule, shown in Figure 5.2(a). In addition, the 1- and 2-hop neighbours information will also be used in the step of cycles selection.

5.3.2 Boundary nodes discovery

After each node knows its 1- and 2-hop neighbours, it needs to determine whether it is a boundary node or not. we propose a method based on Hamiltonian cycle for the verification. For each internal node ID_i, if it has less than three neighbours or there exist no Hamiltonian cycles in its neighbour graph $G_i(V_i, E_i)$, this node is considered to be a boundary node. Otherwise, it is considered as a non-boundary node. For fence nodes, the node itself should be added to its neighbour graph and verify whether there exists a Hamiltonian cycle in the new graph. So the problem here is to find a method to search a Hamiltonian cycle in a graph. In general, it is a NP-complete problem. However, considering the graph in our problem is the neighbour graph of a node, it usually contains a small number of nodes, so it is possible to use an exhaustive search method to check the existence of a Hamiltonian cycle in a graph.

Figure 5.2(b) shows the result of boundary nodes detection, nodes denoted by red stars are boundary nodes. We call the method as Hamiltonian cycle based

boundary node detection (denoted by HC). We can see from the figure that some nodes lying on the boundary of a hole are not recognized. It is also possible that some boundary nodes found by the algorithm are not lying on the boundary of a hole (such case does not happen in this example). In order to find more boundary nodes, we aim to use the boundary nodes detected by the method HC. After some boundary nodes have been found, each boundary node broadcasts a message indicating its status (boundary) to its neighbours. So each node gets the status of its neighbours. Considering that any boundary node should have at least two boundary neighbours and the two neighbours should not be neighbours, we select the boundary nodes that have only one boundary neighbour or have more boundary neighbours but these neighbours are neighbours of each other. Each such node v chooses one of its non-fence neighbours which has the minimum degree and is not a neighbour of any boundary neighbour of v, and set the neighbour to be a boundary node. The process of adding boundary nodes is repeated for the new added nodes and until no nodes can be added. We call the method as improved Hamiltonian cycle based boundary node detection (denoted by IHC). According to this rule, more true boundary nodes can be found, such as the nodes denoted by blue stars in Figure 5.2(c). It is also possible that some nodes are wrongly detected as boundary nodes, such as the nodes denoted by magenta stars in Figure 5.2(c). This case is usually due to the fact that nodes near the external boundary of the target field have a relatively lower number of neighbours.

5.3.3 Boundary cycles discovery

After determining the status (boundary), each boundary node broadcasts a message indicating its status to its neighbours. Then, each node knows the status of its 1-hop neighbours. The next step is to find the cycles bounding holes. Since the algorithm is distributed, there is no central unit to select some nodes to initiate the process. Considering the fact that any internal node whose neighbour graph has more than one connected components must lie on the boundary of at least two coverage holes, we choose such nodes with higher priority to initiate the process. In addition, any other node which has no boundary neighbours or has the minimum id among all its boundary neighbours can also initiate the process. Figure 5.2(d) shows the result of selection. Nodes denoted by green diamonds are initiators.

Each selected node initiates the process by broadcasting a message. As seen from Figure 5.2(c), some nodes lying on the boundary are not recognized as boundary

nodes. Given that only boundary nodes continue broadcasting when they receive a broadcast message, it is possible that some holes can not be discovered. But if all nodes, no matter boundary nodes or not, broadcast the message, the message complexity will be high. So the structure of the message is designed as in Table 5.1.

Table 5.1: Message structure

node_seq	**type**
ids	0, 1 or 2

The element node_seq contains the ids of nodes that have received and forwarded the message. The element type indicates the type of message. We define three types of messages, denoted by 0, 1 and 2. 0 indicates that the last node in node_seq is a boundary node. 1 indicates that the last node is a non-boundary node and 2 indicates that the last two nodes are non-boundary nodes. Furthermore, we set a probability $p(0 < p < 1)$ for non-boundary nodes. When a non-boundary node receives a broadcast message with type 0, it will broadcast the message with probability p and set the message type as 1. When a non-boundary node receives a type 1 message, it will broadcast with probability p^2 and set the message type as 2. If a non-boundary node receives a type 2 message, it will neglect it. Any boundary node will set the message type as 0 when it broadcasts a message. In addition, for any fence node, if it is not a boundary node, it will not forward any message. When broadcasting a message, each node attaches its id to the node_seq of the message sequentially. In this way, the message contains all the ids of nodes from an initiating node to the current one. In addition, the node also keeps the message, showing that it has been visited by the path initiated by the initiator.

Whenever a node receives a message, it needs to first verify whether its id has been in the node_seq of the message. If so, the nodes deletes the message. Otherwise, it continues to check whether the message has the same initiator as some message kept by the node. If not, it means that the message is new and the node will broadcast the message as illustrated in the last paragraph. If yes, the node needs to verify whether the two messages can form a cycle. We use m_n and m_o to denote the new message and the message kept by the node. If the first ids in node_seqs of m_n and m_o are the same and node_seqs of m_n and m_o have no other common ids, the last id in the node_seq of m_n and the second last id in the node_seq of m_o[1] are not neighbours, then a cycle can be formed and the node keeps the cycle. It is possible that the cycle found is not minimum, this can be processed by the next

[1]the id of current node has been in m_0

step illustrated in Section 5.3.4. Furthermore, if the length of node_seq in m_n plus one is less than the length of node_seq in m_o, it means their exists a shorter path between the initiator of the message and the node, so the node replaces m_o by m_n and broadcasts m_n as illustrated in the last paragraph. In addition, in order to reduce the message complexity, when any initiating node receives a message initiated by another node, it will delete the message if the other initiator has higher id than itself.

After the above process, some boundary cycles can be found and it is also possible some cycles have not been discovered because there is no initiator selected in those cycles. In this case, some node can further be randomly selected to initiate the boundary cycles discovery process, such as the node denoted by green diamond in coverage hole 1 in Figure 5.2(e).

To better understand the process of the algorithm, we use an example in Figure 5.3 to explain it. The Rips complex of a WSN is shown in Figure 5.3(a). According to the rule defined in the step of boundary nodes discovery, we can see that nodes 1, 2, 4, 5, 13, 14 and 16 are boundary nodes based on Hamiltonian cycle verification, shown in Figure 5.3(b). Then node 16 has only one boundary neighbour, so it chooses one node from its non-fence neighbours which has the minimum degree and is not connected with node 1. In this way, node 7 is set to be a boundary node, shown in Figure 5.3(c). Then nodes 1 and 5 are selected as initiators, shown in Figure 5.3(d), since the id of node 1 is smaller than its boundary neighbours and node 5 has two connected components in its neighbour graph. They broadcast a message containing their ids respectively and set the message type as 0 because they are both boundary nodes. When their neighbours receive the messages, they can decide whether to transmit the message or not. For example, neighbours of node 5 (nodes 2, 4, 6, 7, 8, 14) receive the message. As nodes 6 and 8 are non-boundary internal nodes, they can choose to broadcast the message with probability p, assume node 8 chooses to transmit the message but node 6 not. Since the other neighbours of node 5 are boundary nodes, they will broadcast the message. This process is shown in Figure 5.3(e) (the message type is not shown in the figure). The node sequence in a red bracket indicates that a node receives a message but has not transmitted the message. Then in the next round, nodes 2, 4, 7, 8, 14 and 16 will broadcast (any non-boundary fence node does not broadcast any message). We assume they broadcast sequentially. The result of this round is shown in Figure 5.3(f), assuming nodes 6 and 8 choose to not transmit any message. The node sequence in a black bracket indicates the message kept by the node. Node 1 will not transmit any message since

it is an initiator and its id is smaller than any other initiators. After that, nodes 3, 5, 7 and 13 will continue broadcasting. And in this process, node 7 can find a cycle <1, 16, 7, 5, 2> and node 13 can find another cycle <5, 4, 13, 14>, shown in Figure 5.3(g). Similarly, node 5 will find the cycle <1, 2, 5, 7, 16> and node 14 will find the cycle <5, 14, 13, 4> in the next round, shown in Figure 5.3(h). Since the two sequences [1, 2, 5] and [1, 2, 4, 13] have two common nodes, node 14 will not form a cycle and just broadcast the message containing the sequence [1, 2, 5]. After that, no node will continue broadcasting messages. Until now, four cycles have been found. We can see that some of them actually bound the same coverage holes. So cycles selection should be performed, which will be illustrated in next section.

5.3.4 Cycles selection

After last step, many boundary cycles have been found. But it is possible that some cycles are not minimum or bound the same holes or even are not really bounding a hole. So it is necessary to analyse these cycles and make choices. This can be realized distributively by each node in the cycle. For any node in a cycle, it first checks whether there exists a shorter path between itself and other nodes in the cycle by using its 1- and 2-hop neighbours information. If there exists, the node can shorten the cycle. After that, it continues to check whether the cycle bounds the same hole as another cycle found before. For any two cycles, if one cycle can be converted to another one by the equivalent transformation presented in Definition 5.1, they are considered to bound the same hole. In addition, it is possible that some non-minimum cycle can not be minimized since each node has only its 1- and 2-hop neighbours information. It is also possible that some cycle bounds more than one holes. To this end, for any two cycles c_1, c_2 with length larger than 7, if $|c_1 \cap c_2| \geq 1/2|c_1|$ or $|c_1 \cap c_2| \geq 1/2|c_2|$, we delete the cycle with larger length since it is highly possible that the cycle with larger length bounds more than one holes in this case. It is also possible that a false cycle can not be deleted. We investigate such error probability in Section 5.4.3. Consider the example in Figure 5.3, four cycles have been found in the last step. In this step, the cycles found by nodes are transmitted to the nodes in the cycles and each node checks whether there exists a shorter path or whether two cycles are the same. After doing this, nodes can find the two cycles <5, 4, 13, 14> and <5, 14, 13, 4> are the same, the two cycles <1, 16, 7, 5, 2> and <1, 2, 5, 7, 16> are also the same. So only one of them is kept. The result is shown in Figure 5.3(i).

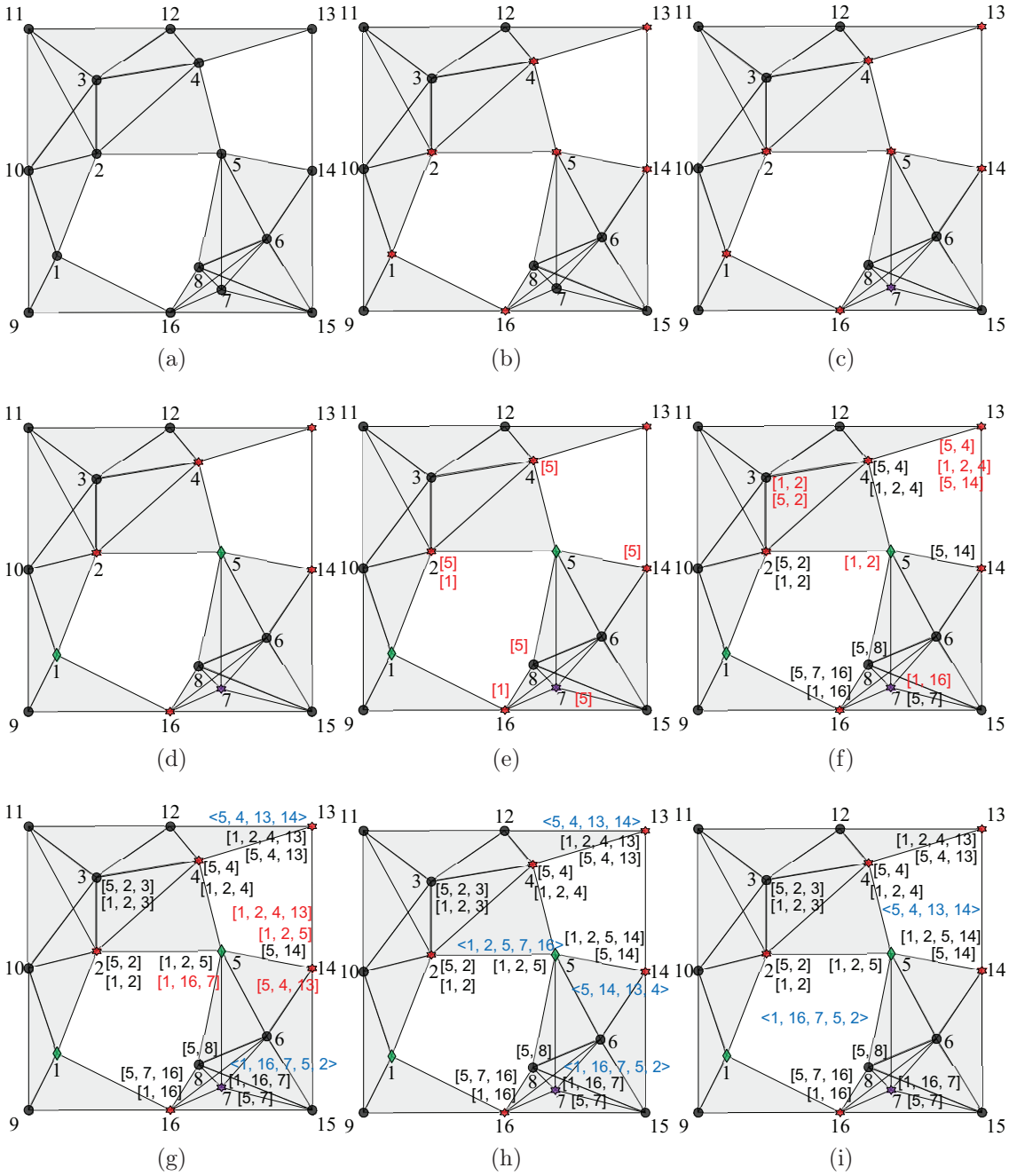


Figure 5.3: An example illustrating the process of this algorithm. (a) Rips complex of a WSN, (b∼c) boundary nodes discovery, (d∼h) boundary cycles discovery, (i) cycles selection

5.4 Simulations and performance evaluation

In this section, simulation settings are first given and probabilities of boundary nodes detection and boundary cycles detection are then presented.

5.4.1 Simulation settings

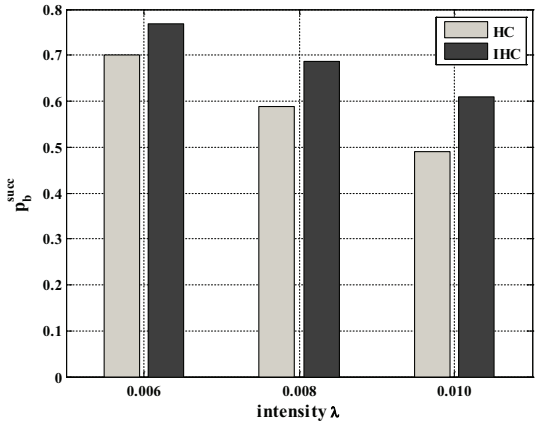
The target field is assumed to be a 100 $\times$ 100 m^2 square area. The sensing radius R_s of each node is 10 meters. The communication radius R_c is set to be 20 meters. There are fence sensors located along the edges of the square with 20 meters distance between neighbours. Other internal sensors are randomly distributed in the area according to a homogeneous Poisson point process with intensity λ. Under such settings, we approximately obtain that λ should be at least 0.006 for the WSN to be connected with high probability according to the results in [99] based on percolation theory. So we set λ to be 0.006, 0.008 and 0.01 in our simulations.

5.4.2 Probability of boundary nodes detection

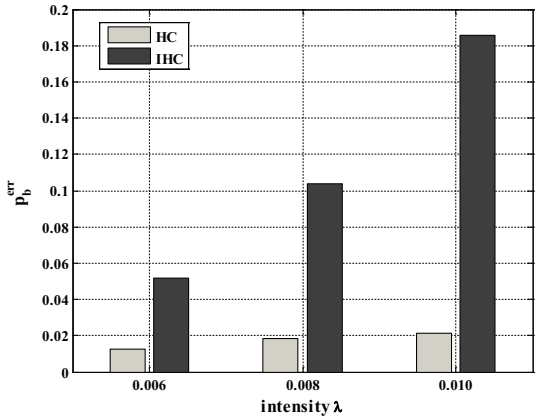
A key step of our algorithm is boundary nodes detection and we propose two methods (denoted by HC and IHC respectively) to detect them. Here we want to have an idea about how many boundary nodes detected by our methods are true or false. First, we use the location based approach in [47] to find all boundary nodes. Then we delete those nodes which only bound a triangular hole since our algorithm only detect non-triangular holes. We use success probability p_b^{succ} to denote the ratio between the number of true boundary nodes detected by our methods and the number of boundary nodes detected by the location based approach. Similarly, error probability p_b^{err} is used to denote the ratio between the number of false boundary nodes detected by our methods and the number of boundary nodes detected by the location based approach. Simulation results for success and error probabilities are shown in Figure 5.4(a) and (b) respectively, which are the average of 1000 simulations. It can be seen from Figure 5.4(a) that when λ is lower, p_b^{succ} of the two methods are relatively higher. That is because when λ is lower, there are fewer number of nodes in neighbour graphs and it is more accurate to detect boundary nodes by checking the existence of a Hamiltonian cycle in neighbour graphs. From Figure 5.4(b), we can see that when using improved Hamiltonian cycle based boundary nodes detection, p_b^{err} increases largely. The larger the density is, the larger is the increase.

5.4.3 Probability of boundary cycles detection

In order to evaluate the accuracy of our algorithm, we compare it with the location based algorithm (denoted as LBA) proposed in [47]. Since the location based algorithm can discover both triangular and non-triangular coverage holes, but our



(a)



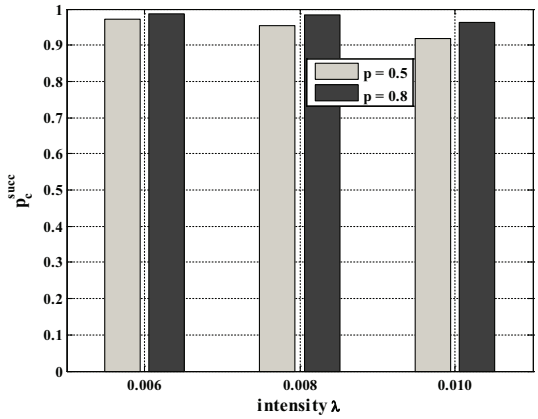
(b)

Figure 5.4: (a) Success probability of boundary nodes detection p_b^{succ}, (b) error probability of boundary nodes detection p_b^{err}

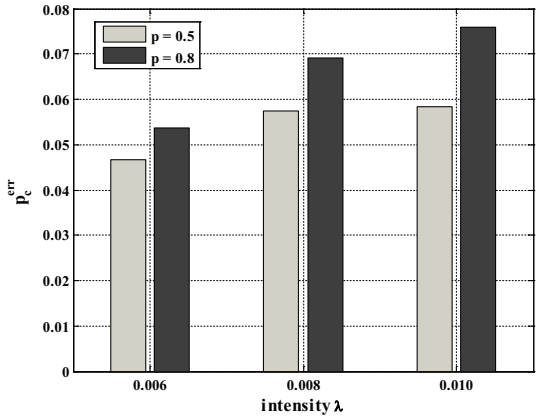
algorithm can only detect non-triangular coverage holes, we do not consider those triangular coverage holes in the comparison. As it is possible that there exist shorter paths in boundary cycles found by LBA, we first shrink them using 1-hop neighbour information of boundary nodes. After that, we compare those boundary cycles with what our algorithm finds. Two boundary cycles are considered to surround the same coverage hole if one cycle can be converted to another one by equivalent transformations presented in Definition 5.1. We can see from Definition 5.1 that equivalent transformations only need 1-hop neighbours information. We emphasize that only 1-hop neighbours information can be used in the comparison in order to evaluate the accuracy of boundary cycles found by our algorithm. For example, if one cycle c_1 found by our algorithm can not be converted to another cycle c_2 found by LBA using only 1-hop neighbours information but can be converted by using 2-hop neighbours information, we consider the cycle c_1 is not accurate and the corresponding coverage

hole is not found.

Based on the method presented above, we choose p to be 0.5 and 0.8. For each λ, we run 1000 simulations. We also use success and error probabilities, denoted by p_c^{succ} and p_c^{err}, to evaluate the accuracy of our algorithm for boundary cycles detection, which are shown in Figure 5.5(a) and (b) respectively. The detailed values are presented in Table 5.2.



(a)



(b)

Figure 5.5: (a) Success probability of boundary cycles detection p_c^{succ}, (b) error probability of boundary cycles detection p_c^{err}

From Figure 5.5(a), we can see that p_c^{succ} increases when p increases from 0.5 to 0.8. That is because when p is larger, there are more non-boundary nodes broadcasting messages, which can increase the number of boundary cycles detected. In addition, we can see that p has a larger impact on the probability when λ is higher. It is because when λ is higher, more nodes lying on the boundary are not recognized, these nodes may broadcast messages when p is higher.

Table 5.2: Number of boundary cycles

λ	p	n_c [a]	n_f [b]	n_m [c]	p_c^{succ}	p_c^{err}	n_L [d]
0.006	0.5	7213	347	216	0.9709	0.0467	7429
	0.8	7322	399	107	0.9856	0.0537	
0.008	0.5	6994	421	330	0.9549	0.0575	7324
	0.8	7202	507	122	0.9833	0.0692	
0.010	0.5	5690	362	514	0.9172	0.0583	6204
	0.8	5977	471	227	0.9634	0.0759	

[a] number of correct cycles found by our algorithm
[b] number of false cycles found by our algorithm
[c] number of cycles missed by our algorithm
[d] number of cycles found by LBA

5.5 Chapter summary

In this chapter, we have proposed a graph based algorithm to detect boundary cycles of coverage holes in a WSN. The algorithm includes four steps. In the first step, each node obtains its 1- and 2-hop neighbours information. After that, each node determines that it is a boundary node or not by checking whether there exists a Hamiltonian cycle in its neighbour graph. Then some boundary nodes are randomly selected to initiate the process to discover boundary cycles. The cycles found may not be minimum or bound the same coverage holes, so cycles selection is performed in the final step. We compare our algorithm with a location based approach, which shows that our algorithm can accurately find boundary cycles of more than 95% coverage holes when the intensity is low. But the algorithm has high complexity since each node needs to check the existence of a Hamiltonian cycle in its neighbour graph by exhaustive search, and the error probability of boundary cycle detection is high when the intensity is high, so we aim to design an efficient and accurate algorithm for coverage hole detection in WSNs, which will be presented in Chapter 6.

Chapter 6

Homology based Distributed Coverage Hole Detection

6.1 Introduction

In last chapter, we have proposed a graph based coverage hole detection algorithm, but that algorithm has high complexity. In this chapter, we aim to design an efficient distributed coverage hole detection algorithm. We design such an algorithm based on homology theory. The basic idea of this algorithm is that for the Rips complex of a WSN, we try to delete some vertices and edges without changing the homology while making the Rips complex more sparse and nearly planar. Then it is easy to find boundary edges, each of which is part of at most one triangle[1]. Finally such edges are connected in some order to form the boundary cycles.

The main contributions of this chapter are in three aspects. First, we define a rule for each vertex or edge to decide whether its deletion will change the homology of the Rips complex of the original WSN or not. Each vertex or edge can make such decision independently and only needs its 1-hop neighbour information.

Second, we propose a homology based algorithm to discover boundary cycles of coverage holes. In the algorithm, each node computes its weight independently. The weight of a node is a density indicator of its surrounding nodes. The larger the weight is, the higher is the probability that the node is not on the boundary of a coverage hole. Then some vertices and edges are deleted without changing the homology of Rips complex. After this process, the Rips complex becomes much sparser and nearly all boundary edges can be found by just checking whether the edge

[1]In this chapter, a triangle means a 2-simplex

is part of at most one triangle or not. Then coarse boundary cycles are discovered by connecting boundary edges in some order and exact boundary cycles are further found by minimizing coarse boundary cycles.

Third, extensive simulations are performed to evaluate the performance of our algorithm. Analysis shows that the worst case complexity of our algorithm is $O(n^3)$, where n is the maximum number of neighbours of each node. Comparisons with a boundary recognition algorithm show that our algorithm is more efficient and easy to implement in a distributed way. Comparisons with a location based algorithm further show that our algorithm can accurately detect the boundary cycles of about 99% coverage holes in about 99% cases.

6.2 Models and definitions

We use the same models for nodes as last chapter. In addition, we need to give some definitions that will be used in the process of this algorithm.

We say that a i-simplex $[v_{i0}, v_{i1}, ..., v_{ii}]$ is part of a j-simplex $[v_{j0}, v_{j1}, ..., v_{jj}]$ if $[v_{i0}, v_{i1}, ..., v_{ii}] \subset [v_{j0}, v_{j1}, ..., v_{jj}]$. So the vertex $[v_0]$ or $[v_1]$ is part of the edge $[v_0, v_1]$. The edge $[v_0, v_1]$ is part of the triangle $[v_0, v_1, v_2]$. In addition, we use $E(v)$ to denote all the edges that the node v is part of and $T(v)$ to denote all the triangles that the node v is part of.

Definition 6.1 (Index of a triangle)**.** *The index of a triangle $\triangle$ is the highest dimension of the simplex that the triangle is part of, denoted by $I_\triangle$.*

Definition 6.2 (Weight of a node)**.** *The weight of a fence node is defined to be 0. For any internal node v, if there exists one edge in $E(v)$ which is not part of any triangle, the weight w_v of node v is set to be 0; if not, the weight is the minimum index of all the triangles in $T(v)$, that is $w_v = \min_{\triangle \in T(v)} I_\triangle$.*

The weight of an internal node is an indicator of the density of its surrounding nodes. If the weight of an internal node is 0, the node must be on the boundary of a coverage hole. The larger the weight is, the higher is the probability that the node is not on the boundary of a coverage hole.

We also use the definition of simple-connectedness graph as in [46]. Let G be a simple graph with vertex set $V(G)$ and edge set $E(G)$. A cycle C is a sub-graph of G if it is connected and each vertex in C has degree two. The length of a cycle C is the number of its edges, denoted by $|E(C)|$. The cycle space $C(G)$ of a graph G contains all the cycles in G. The addition of two cycles C_1 and C_2 is defined as

$C_1 \oplus C_2 = (E(C_1) \cup E(C_2)) \setminus (E(C_1) \cap E(C_2))$. The triangle cycle subspace $C_T(G)$ of G is the set of all 3-length cycles in $C(G)$.

Definition 6.3 (Simple-Connectedness Graph)**.** *A connected graph G is of simple connectedness if its cycle space $C(G)$ is empty, or for any cycle C in $C(G)$, there exists a set of 3-length cycles $T_0 \subseteq C_T(G)$ such that $C = \sum_{T \in T_0} T$.*

Let X be a vertex (or edge) set in a graph G, we use $G[X]$ to denote the vertex-induced (or edge-induced) sub-graph by X. The neighbours of a vertex v in G is denoted by $N_G(v)$. The neighbour graph $\Gamma_G(v)$ of vertex v is denoted as $G[N_G(v)]$. The neighbour graph $\Gamma_G(e)$ of an edge $e = (u, v)$ is defined as $G[N_G(u) \cap N_G(v) \cup \{u, v\}] - e$. The neighbour set of k-simplex $[v_0, v_1, ..., v_k]$ is defined as $\bigcap_{i=0}^{k} N_G(v_i)$.

Definition 6.4 (Deletion of a k-simplex in Rips complex)**.** *A k-simplex $[v_0, v_1, \cdots, v_k]$ is deleted in a Rips complex $\mathcal{R}(\mathcal{V})$ means that the simplex and all simplices which the simplex is part of are deleted from $\mathcal{R}(\mathcal{V})$.*

Based on definitions above, we can give the definition of HP (Homology Preserving) transformation.

Definition 6.5 (HP Transformation)**.** *A HP transformation is a sequential combination of vertex (or edge) deletion as follows: a vertex (or edge) x of G is deletable if neighbour graph $\Gamma_G(x)$ (1) has two or more vertices; (2) is connected and (3) is a simple-connectedness graph.*

Theorem 6.1. *HP transformations do not change the number of coverage holes in Rips complex of a WSN.*

Proof. In order to prove HP transformations do not change the number of coverage holes in Rips complex of a WSN, we only need to prove that in the process of HP transformations, there is no new coverage holes created and no two coverage holes merged. If a new coverage hole is created when a vertex v (or edge e) is deleted, then the boundary cycle of the new coverage hole must be a cycle in $\Gamma_G(v)$ (or $\Gamma_G(e)$), which means $\Gamma_G(v)$ is not a simple-connectedness graph. It is contrary to the third condition in HP transformation, so there is no new coverage hole created. On the other hand, if two coverage holes are merged when a vertex v (or edge e) is deleted, then the neighbour graph $\Gamma_G(v)$ ($\Gamma_G(e)$) must not be connected, which is contrary to the second condition in HP transformation. So there are no two coverage holes merged. Consequently, the number of coverage holes will not be changed in the process of HP transformation. □

6.3 Homology based distributed algorithm

Our algorithm includes five components: weight computation, vertex and edge deletion, boundary edge detection, primary boundary cycles discovery and boundary cycles minimization, as shown in Figure 6.1.

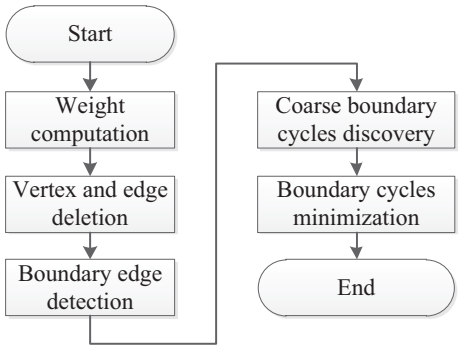


Figure 6.1: Flow chart of the algorithm

An example is used to illustrate the procedures of this algorithm in Figure 6.2. For a WSN with some coverage holes, the Rips complex of the WSN is first constructed, shown in Figure 6.2(a), then the algorithm aims to discover minimum boundary cycles of all holes. In weight computation component, each node computes its weight independently according to Definition 6.2. After obtaining the weight, each node continues to determine whether it can be deleted or not according to some rules defined hereafter. Figure 6.2(b) shows the result of vertex deletion. Furthermore, some special node will decide whether some edge can be deleted or not. Figure 6.2(b∼c) shows the process of such special edge deletion. After the second component, many boundary edges can be found, as the red line shown in Figure 6.2(d). But it is possible that some other boundary edges have not been found. Then in the third component, all or nearly all boundary edges will be found after deleting some edges, see Figure 6.2(e∼j). Then coarse boundary cycles can be easily discovered, as shown in Figure 6.2(k). It is possible that the found boundary cycles are not minimum. In this case, coarse boundary cycles will be minimized in the final component as shown in Figure 6.2(l).

6.3.1 Weight computation

In this component, each node computes its weight. For any fence node, its weight is 0. For any internal node, theoretically it needs to construct all the simplices it

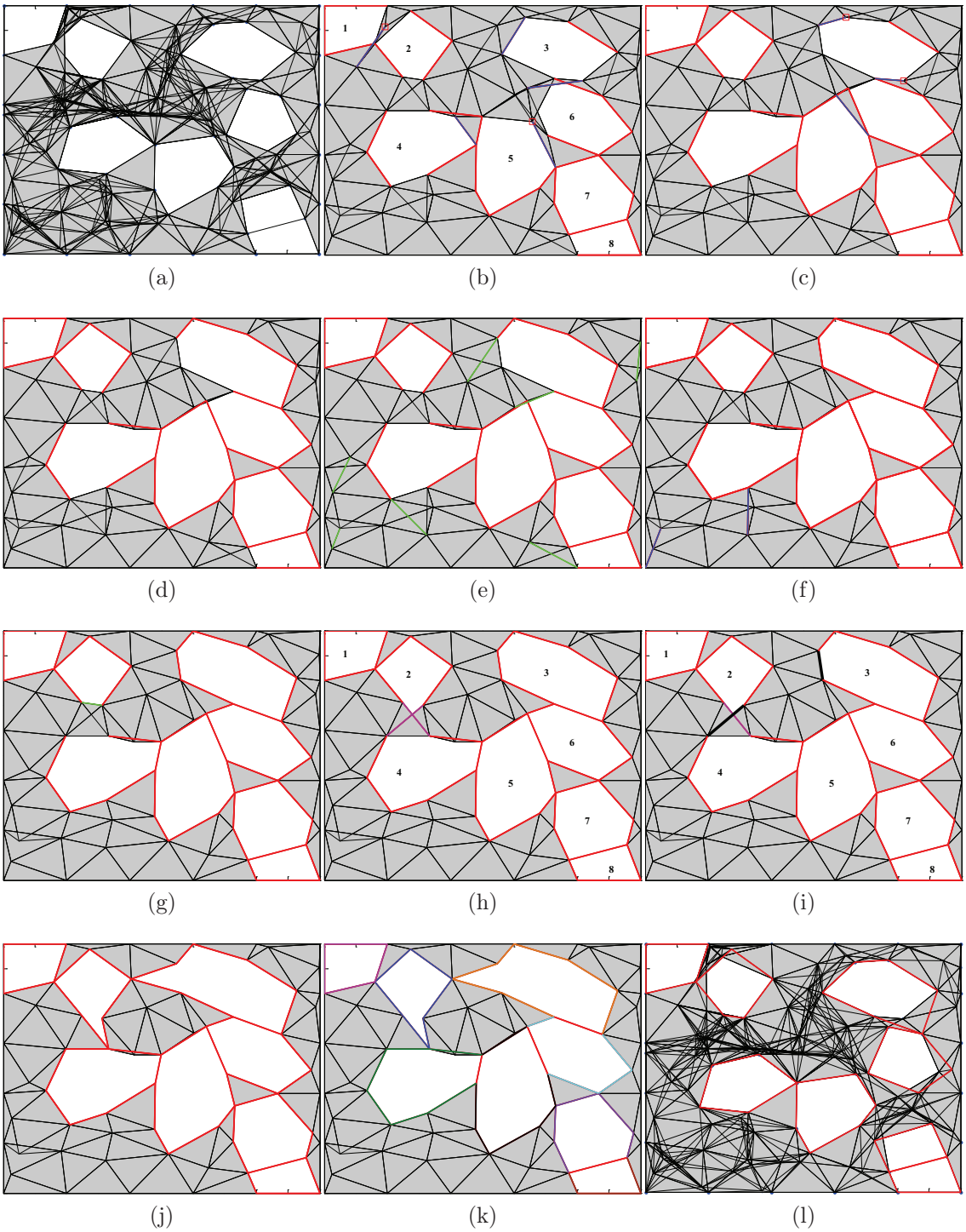


Figure 6.2: Procedures of the boundary detection algorithm. (a) Rips complex of a WSN, (b) after vertex deletion, (c∼d) edge deletion, (e∼j) boundary edge detection, (k) coarse boundary cycles discovery, (l) boundary cycles minimization

is part of. As we consider WSNs on a planar target field, each internal node only needs to construct all its 1-simplices and 2-simplices and their neighbours. This can also reduce the computation complexity. In order to do this, the node needs to obtain all its 1- and 2-hop neighbours information. This can be easily achieved by two broadcasts of hello message. In the first one, each node broadcasts its id. When it gets all the ids of its 1-hop neighbours, each node continues to broadcast a hello message containing the ids of its 1-hop neighbours. After receiving the neighbour list of its neighbours, the node can obtain its $E(v)$, the set of edges (1-simplices) and $T(v)$, the set of triangles. It can also get the neighbours of each simplex. For any $e \in E(v)$, let $n(e)$ denote the neighbour set of e. For any $t \in T(v)$, let $n(t)$ denote the neighbour set of t. Then the weight of node v can be computed as in Algorithm 3.

Algorithm 3 Weight computation (for internal node v)

Begin
  if $\exists e \in E(v), n(e)$ is empty **then**
    $w_v = 0$
  else if $\exists t \in T(v), n(t)$ is empty **then**
    $w_v = 2$
  else
    $w_v = 3$
  end if
END

6.3.2 Vertex and edge deletion

In this component, we conduct maximal vertex deletion without changing the number of coverage holes in the original WSN and also delete some special edges if such edges exist. For vertex deletion, we only consider internal nodes, fence nodes will never be deleted. In the following part, we use vertex and node interchangeably.

(1) vertex deletion

From the definition of weight, we can see that the higher the weight is, the higher is the probability that the sensing range of the node is fully covered by its neighbours, consequently the probability that the node does not lie on the boundary is higher. Meanwhile, if the deletion of a vertex may create a new coverage hole, it must not be deleted no matter how high the weight is. So we have such a rule for vertex deletion. If the weight of a vertex is smaller than 3, it should never be deleted. Otherwise, the vertex continues to check whether it is deletable or not

according to the HP transformation. After the verification, the vertex broadcasts a message indicating that it can be deleted or not. After receiving the status of all its neighbours, each deletable vertex continues to check whether it should be deleted. It can be found that the weight of any deletable vertex must be 3. We assume that the vertex with lower ID has the priority to be deleted first. So each deletable vertex just needs to check whether its ID is the lowest among all its deletable neighbours. If so, it should be deleted. Otherwise, it should not be deleted. Algorithm 4 gives the detailed process for vertex deletion. According to the rule, it can be seen that two neighbouring vertices will not be deleted simultaneously, so each vertex can make the decision independently. When a vertex is deleted, it broadcasts a message to its neighbours. All its neighbours will modify their simplices according to Definition 6.4 and compute their weights again. The procedure of vertex deletion terminates until no vertex can be deleted in the Rips complex. Figure 6.2(b) gives the initial result after vertex deletion.

Algorithm 4 Vertex deletion (for internal node v)

Begin
  if $w_v < 3$ **then**
    node v can not be deleted
  else if node v is not deletable according to HP transformation **then**
    node v can not be deleted
  else if the ID of node v is the smallest among all its deletable neighbours **then**
    node v is deleted
  end if
END

(2) edge deletion

After vertex deletion, we find one interesting thing. Edges having no neighbours must be on the boundary of holes, such as the common edge of coverage holes 7 and 8 in Figure 6.2(b). The edge having only one neighbour lies on the boundary of a coverage hole with high probability, such as the red edges shown in Figure 6.2(b). But it is possible that there exist some special such edges not lying on the boundary, such as the blue edge between coverage holes 1 and 2 in Figure 6.2(b). We try to delete such special edges. Since the edge has only one neighbour, deleting the edge will not create a new hole.

We call edges having at most one neighbour as boundary edges. Then we design a rule for deleting special boundary edges. If a vertex v has only one boundary edge vx and vx has only one neighbour y and deletion of vx will not make v have a new

boundary edge, then vx can be deleted. This can be easily achieved by checking whether vy has more than two neighbours, shown in Algorithm 5. After vx is deleted, the nodes v, x and y need to update their simplices according to Definition 6.4 and recompute their weights. According to this rule, nearly all boundary edges which does not lie on the boundary of any holes can be deleted. But it is also possible that some edges lying on the boundary are also deleted, such as the blue edges in coverage holes 3, 4, 5 and 6 in Figure 6.2(b). This is not a big issue, because deletion of such edges will not create new holes and just enlarge the current coverage holes. It can be solved in the boundary cycles minimization component. In addition, after edge deletion, it is possible that some vertices can be deleted again, such as the vertex denoted by red square in Figure 6.2(b∼c). If such a case happens, we can continue to do vertex deletion until no more vertex or edge can be deleted. Figure 6.2(d) shows the result after edge deletion.

Algorithm 5 Edge deletion (for internal node v)

```
Begin
  if v has only one boundary edge vx and vx has one neighbour y then
    N(vy) = neighbour set of vy
    if |N(vy)| > 2 then
      vx can be deleted
    end if
  end if
END
```

6.3.3 Boundary edge detection

After deleting some vertices and edges, we can find that nearly all boundary edges lie on the boundary of holes. It can also be found that some edges lying on the boundary have not been found. In this component, we try to find such edges as many as possible. In all cases, such edges have two or more neighbours. If we consider the nodes having one or more boundary edges as boundary nodes and other nodes as non-boundary nodes, then we try to delete some edges connecting non-boundary nodes and boundary nodes according to the HP transformation, such as the green edges shown in Figure 6.2(e). After that, it is possible that some new boundary edges are recognized and some special edges illustrated in Section 6.3.2 are also identified, as the blue edges shown in Figure 6.2(f). In this case, we can still use the rule to delete them. It is still possible that some edges lying on the boundary have not been discovered. Such case usually occurs when some boundary nodes are

neighbours and edges connecting them have more than one neighbours. In this case, we randomly delete some of such edges according to the HP transformation, such as the green edges in Figure 6.2(g). After that, some new boundary edges can be recognized, as shown in Figure 6.2(h).

But it is possible that the new found boundary edges can not construct a correct cycle with other boundary edges, as the magenta edges in coverage holes 2 and 4 in Figure 6.2(h). Such case is often due to the fact that two boundary edges cross with each other. It is found from numerous simulations that there are mainly three cases of crossing boundary edges, as shown in Figure 6.3, red lines denote boundary edges and black ones denote non-boundary edges. Similarly, we define some rules to delete such edges. We take node v as an example, in the top part of Figure 6.3(a), the two red lines connecting v and its neighbours are deleted, as shown in the bottom part of Figure 6.3(a). For the cases in Figure 6.3(b) and (c), the red line connecting v and its neighbour is deleted, as shown in the bottom part of Figure 6.3(b) and (c). According to such rules, some boundary edges can be deleted, such as the black bold edges in coverage hole 2 in Figure 6.2(i). It is also possible that certain boundary edges are deleted incorrectly. It is not a big issue as explained in last section.

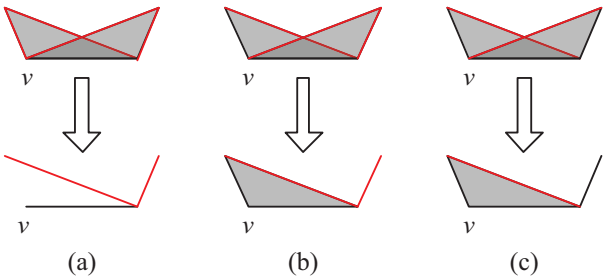


Figure 6.3: Illustration of crossing boundary edges

In addition, there exists also another case that a false boundary edge is found, as the blue edge in Figure 6.4. To this end, we can define a similar rule to delete them. For any internal node v, if it has two boundary neighbour nodes u and w, the edge vu and vw are not boundary edges and wu is a boundary edge, if the deletion of the edge wu can make at least one of the two edges vu and vw be a boundary edge, then wu can be deleted. According to such rule, the false boundary edge can be deleted and some correct boundary edge may also be deleted, as the black bold edge in coverage hole 3 in Figure 6.2(i). Similarly, it is not a big issue.

In general, after the process above, nearly all boundary edges can be found, as shown in Figure 6.2(j). But there exists one special case as in Figure 6.5. In this

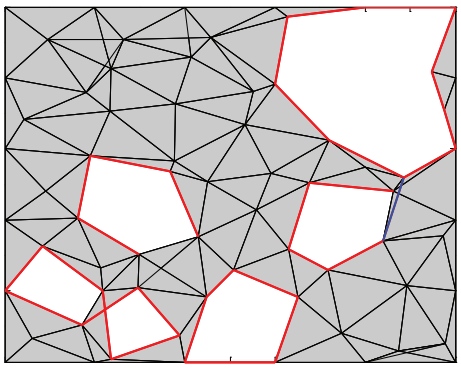


Figure 6.4: Illustration of false boundary edges

case, some edges lying on the boundary can not be detected. It can be solved in the next component.

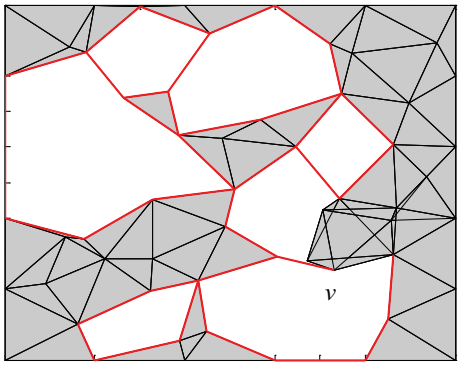


Figure 6.5: Special case when some boundary edges can not be detected

6.3.4 Coarse boundary cycles discovery

After boundary edges are detected in the former component, it is easy to discover the coarse boundary cycles. For two coarse boundary cycles, there are possible three types of relation: (1) having common boundary edges. It means two coarse boundary cycles have at least one common edges. (2) having common nodes. It indicates that two coarse boundary cycles does not have common edges but have at least one common node. (3) separated. Two coarse boundary cycles have neither common edges nor common nodes.

For two coarse boundary cycles with a common boundary edge uv, we can choose one node from u and v having three boundary edges as an initiator, for example, u is selected and u has three boundary edges uv, ux and uy. Then, u initiates the coarse boundary cycles discovery process by sending a message to v. The message

contains the ID of u. When v receives this message, it records the ID of the initiator and sets u as its father node. After that, v continues to send the message along its other boundary edges. Each node receiving the message makes the same process until node u receives the message along its two other boundary edges ux and uy. Then node u sends a message to x and y respectively noticing them to form coarse boundary cycles. Then node x sends a message containing its ID to its father node, its father node does the same thing until the message arrives at node u. Then a coarse boundary cycle is formed. Similarly, node y does the same thing and another cycle can be formed. After forming coarse boundary cycles, node u informs the sequence to the nodes in the cycles. For two coarse boundary cycles with common nodes or separated, we just need to randomly choose one node which has two boundary edges as an initiator. Then the node initiates the process to find the coarse boundary cycle by sending a message along one of its boundary edges. When it receives the message coming back along the other boundary edge, it discovers one coarse boundary cycle. In this way, all coarse boundary cycles can be found, as the cycles indicated by different colours in Figure 6.2(k).

As for the special case shown in Figure 6.5, when the node v receives a message from its boundary neighbour node, it broadcasts the message to all its neighbours except its father node. If its neighbour node is a boundary node, then the message can be sent along the boundary edges. If its neighbour node is not a boundary node but it has boundary neighbour nodes, then it can send the message to its boundary neighbour nodes. Else, it will not transmit the message again. In this way, the message goes along boundary edges most of the time and can return to the original node sending the message.

6.3.5 Boundary cycles minimization

It is possible that some coarse boundary cycles found are not minimum, so we need to minimize such cycles. This can be achieved by checking whether there exists a shorter path between any two nodes in the cycle. Since each node has its 1- and 2-hop neighbours information, it can locally check the existence of a shorter path in the cycle. If there exists, we shorten the cycle and continue to do the same verification until no such case exists. After that, it is still possible some cycle has not been minimized, such as the coverage hole 2 in Figure 6.2(k). So we use the following 2-hop shrinking process to make the cycle as shorter as possible. For any four adjacent nodes in the cycle, say a, b, c, d, if there exists one node x which is one

common neighbour of nodes a, b, c, d, then the cycle can be shortened by using x to replace nodes b and c.

In this way, we can nearly obtain most minimum cycles surrounding coverage holes. It is also possible that in some cases, we can not get the minimum cycles since each node only has its 1- and 2-hop neighbours information.

6.4 Performance evaluation

In this section, we first give simulation settings and then analyse the complexity of our algorithm. After that, we compare our algorithm with one boundary recognition algorithm and another location based coverage hole detection algorithm.

6.4.1 Simulation settings

For performance evaluation of the algorithm, we choose a 100 $\times$ 100 m^2 square area as the target field. The sensing radius R_s of each node is 10 meters. The communication radius R_c is set to be 20 meters and so $\gamma = 2$. There are fence sensors locating along the edges of the square with 20 meters distance between neighbours. Other internal sensors are randomly distributed in the area according to a homogeneous Poisson point process with intensity λ.

6.4.2 Complexity

The computation complexity of each step in the algorithm is shown in Table 6.1. In the weight computation component, each node only needs to check all its 2-simplices, so the computation complexity is $O(n^2)$, where n is the number of its 1-hop neighbours.

Table 6.1: Complexity of each step in the algorithm

Step	Complexity
Weight computation	$O(n^2)$
Vertex and edge deletion	$O(n^3)$
Boundary edge detection	$O(n^3)$
Coarse boundary cycles discovery	$O(1)$
Boundary cycles minimization	$O(1)$

In vertex deletion part, each node needs to check whether it is deletable or not according to HP transformation. This can be done by checking all its cycles in its

neighbour graph. It can build a spanning tree in its neighbour graph and check all fundamental cycles in the spanning tree. There are $E - n + 1$ fundamental cycles, where E is the number of edges in its neighbour graph, so the worst case computation complexity is $O(n^2)$. Since the node needs to recompute its weight and recheck whether it is deletable when any one of its neighbour is deleted, so the total worst case computation complexity is $O(n^3)$. As for the edge deletion, the node only needs to do simple verification as shown in Algorithm 5, the complexity is $O(1)$.

In the boundary edge detection component, the non-boundary nodes need to check whether the edge connecting itself with its boundary neighbours can be deleted or not according to HP transformation. So the worst case computation complexity is $O(n^3)$, the actual complexity is much less than that since for one edge, there are usually very few nodes in its neighbour graph. The boundary nodes also need to check whether the edges connecting itself and its boundary neighbours can be deleted or not, so the complexity is also $O(n^3)$. In addition, the boundary nodes need to check whether there exist special cases as illustrated in Section V.D. The worst case computation complexity for such process is $O(n^2)$. So the complexity of this step is $O(n^3)$.

As for the final two components, each node only needs to broadcast some messages and do some local computations, the complexity is $O(1)$. So the total worst case computation complexity for our algorithm is $O(n^3)$.

6.4.3 Comparison with boundary recognition algorithm

As discussed in Chapter 2, many boundary recognition algorithms have been proposed. Among them, the algorithm proposed in [46] is the most promising for coverage hole detection. So we want to compare it with our algorithm. We use BR (Boundary Recognition algorithm) and HBA (Homology Based Algorithm) to denote their algorithm and our algorithm respectively.

We first introduce the algorithm BR briefly. In BR, a FGP (Fundamental Group Preserving) transformation is defined. The algorithm includes four components: skeleton extraction, primary boundary cycles and refined inner boundary cycles and refined outer boundary cycle. In skeleton extraction, maximal vertex and edge deletion are performed according to FGP transformation and a skeleton graph is obtained. In this component, each node needs its k-hop neighbour information ($k \geq 2$). After that, the skeleton graph is separated into primary boundary cycles. Each primary boundary cycle contains one hole. Then primary boundary cycles are

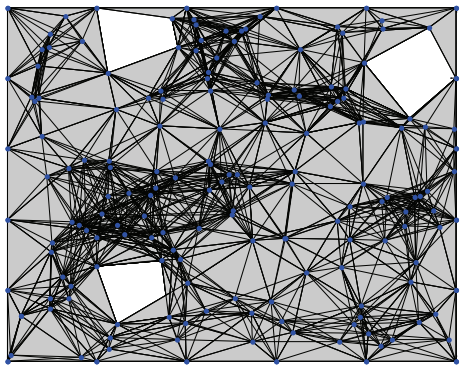
refined into tightest inner boundaries. For each boundary cycle C, it is extended in the graph of the WSN according to FGP transformation and a maximal graph G_C which is topologically equivalent to C is obtained. Then one vertex v in G_C is selected and v is extended in G_C to obtain a subgraph G_v of G_C. The edges in G_C but not in G_v are considered as gap edges and they found that any cycle in G_C surrounding the hole must contain at least one gap edge. So for each gap edge, a shortest cycle containing the gap edge is obtained. Among all these cycles, the shortest cycle is considered as the tightest inner boundary cycle. As we only consider the inner boundary, the component refined outer boundary cycle is not discussed here.

Comparing the two algorithms BR and HBA, we can find that they have the same worst case computation complexity. But our algorithm is more easy to implement in a distributed way for the following reasons. First, for the vertex deletion part, in our algorithm HBA, each node can independently decide whether it can be deleted or not. But in BR, it is possible that two nodes with k-hop distance can not be deleted simultaneously, so they need to interact with each other. Second, in order to find the minimum boundary cycles, the node in our algorithm HBA only needs to do some local verifications. But in BR, nearly all nodes in the whole WSN may be involved, which is neither efficient nor scalable for large scale WSNs.

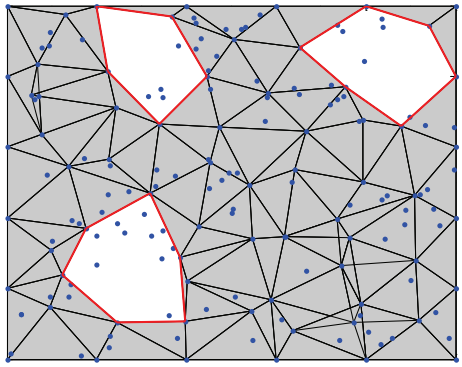
In addition, for the WSNs which have largely separated coverage holes, our algorithm is more efficient. This is because in BR, the skeleton graph is required to be connected in the process of skeleton extraction. If the coverage holes are largely separated, they have to be enlarged much in order to keep the connectivity of skeleton graph. There is no such requirement in our algorithm. Figure 6.6 gives an example to show this.

6.4.4 Comparison with location based algorithm

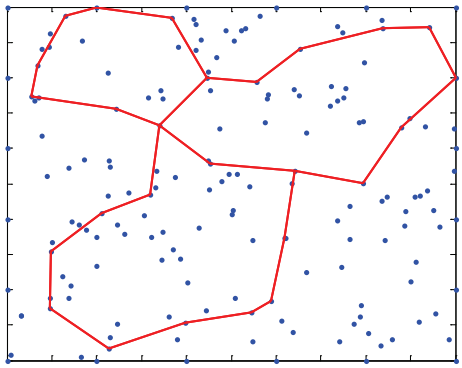
In order to evaluate the accuracy of our algorithm HBA, we compare it with the location based algorithm LBA proposed in [47]. We use the same way as discussed in Section 5.4.3 to do the evaluation. We set λ to be 0.008, 0.010 and 0.012 respectively. For each intensity, 1000 simulations are performed. Simulation results show that when λ is 0.008, there are nine times among the 1000 times when our algorithm can not find all non-triangular coverage holes. In each of the nine times, only one coverage hole is missed. There are 7363 non-triangular holes in total and 7354 ones found by our algorithm. When λ is 0.010 and 0.012, only one time among the 1000



(a)



(b)



(c)

Figure 6.6: (a) Rips complex of a WSN, (b) coarse boundary cycles found by HBA, (c) primary boundary cycles found by BR

times when our algorithm can not find all coverage holes. And in that time, only one coverage hole is missed. When λ is 0.010, there are 6114 non-triangular holes in total and 6113 ones found by our algorithm. When λ is 0.012, there are 4613 non-triangular holes in total, of which 4612 ones are found. The results are shown in Table 6.2. All these results show that our algorithm can find about 99% coverage holes in about 99% cases.

Table 6.2: Number of non-triangular holes

λ	number of non-triangular holes found by HBA	number of non-triangular holes missed by HBA	number of non-triangular holes found by LBA
0.008	7354	9	7363
0.010	6113	1	6114
0.012	4612	1	4613

6.5 Chapter summary

In this chapter, we have proposed a homology based distributed algorithm to detect boundary cycles of coverage holes in WSNs. The algorithm includes five components: weight computation, vertex and edge deletion, boundary edge detection, coarse boundary cycles discovery and boundary cycles minimization. The Rips complex of the WSN is first constructed. Then in weight computation, and each node computes its weight independently. The weight of a node is a density indicator of its surrounding nodes. The larger the weight is, the higher is the probability that the node is not on the boundary of a coverage hole. In vertex and edge deletion part, we define a rule for each vertex or edge to decide whether its deletion will change the homology or not and some vertices and edges are deleted without changing the homology of Rips complex. After this process, the Rips complex becomes much sparser. In boundary edge detection part, nearly all boundary edges can be found by just checking whether the edge is part of at most one triangle or not. Then coarse boundary cycles are discovered by connecting boundary edges in some order and exact boundary cycles are further found by minimizing coarse boundary cycles. Our algorithm is with the worst case computation complexity $O(n^3)$, where n is the maximum number of neighbours of each node. We also compare our algorithm with a boundary recognition algorithm and a location based algorithm, which shows that our algorithm is efficient and accurate to discover boundary cycles of coverage holes.

Chapter 7

Conclusions and Future Work

In this chapter, we summarize our major contributions and discuss future research directions.

7.1 Major contributions

This work aims at studying the applications of homology theory for coverage hole detection in WSNs. Two main aspects have been studied, namely accuracy of homology based coverage hole detection and distributed algorithms for coverage hole detection. Specifically, the main contributions can be summarized as follows.

- Accuracy of homology based coverage hole detection

  We first analysed the relationship between Čech complex and Rips complex in terms of coverage holes for WSNs on a planar target field. Their relation depends on the ratio γ between communication radius and sensing radius of each sensor. When $\gamma \leq \sqrt{3}$, Rips complex does not miss any coverage holes. While $\gamma > \sqrt{3}$, Rips complex may miss coverage holes and the holes missed by Rips complex are always bounded by triangles. Thus we defined them to be triangular holes and other holes to be non-triangular.

  Furthermore, we chose the proportion of the area of triangular holes as the metric to evaluate the accuracy of homology based coverage hole detection. Such proportion is related to γ. Then we analysed the proportion in three cases and for each case, closed form expressions for lower and upper bounds of the proportion are derived. Simulations results are well consistent with the analytical lower and upper bounds, with maximum differences of 0.5% and 3%. In addition, we extended the analysis to the sphere case. Simulation results

show that the radius of sphere has little impact on the proportion when it is much larger than communication and sensing radii of each sensor. More importantly, our analytical results can be used for planning of WSNs.

- Distributed algorithms for coverage hole detection

  For non-triangular holes, we proposed two algorithms to detect them. The first algorithm detects boundary nodes by checking whether there exists a Hamiltonian cycle in the neighbour graph of each node. Since it is a NP-complete problem to check the existence of a Hamiltonian cycle, the complexity of the algorithm is very high.

  The second algorithm we proposed is more efficient and can work in both sparse and dense WSNs. The basic idea is that for the Rips complex of a WSN, we try to delete some vertices and edges without changing the homology while making the Rips complex more sparse and nearly planar, then it is easy to find boundary cycles of coverage holes. Comparisons with a boundary recognition algorithm and a location based algorithm show that our algorithm is efficient and can accurately detect 99% coverage holes in 99% cases. Although in some special cases, our algorithm can not detect the accurate coverage hole boundary, it can still provide some useful information for repairing coverage holes.

In addition, there also exist some limitations in our work. First, for sensing and communication models of sensor nodes, we consider only the boolean model. Second, in the algorithms we proposed, we always assume nodes can receive messages correctly, without considering packet error or loss. These problems need further consideration in our future work.

7.2 Future research directions

This work mainly focuses on the applications of homology theory for coverage hole detection in WSNs on 2D plane and sphere. Homology theory can also be applied in 3D WSNs. In addition, it has also potential applications in cellular networks.

- Coverage hole detection in 3D WSNs

  For 2D WSNs, we only need to consider 2-dimensional Rips complex for coverage hole detection. For 3D WSNs, we will need to consider 3-dimensional

Rips complex for coverage hole detection. According to the relationship between Čech complex and Rips complex, we know that Rips complex may also miss coverage holes for 3D WSNs. The holes missed by Rips complex must be surrounded by tetrahedrons. So it is also essential to analyse the accuracy of Rips complex for coverage hole detection in 3D WSNs. Furthermore, we can also design distributed algorithms to detect boundaries of coverage holes. The difference lies in that the boundary of a coverage hole in 2D WSNs is a cycle formed by edges, while it is a polyhedron formed by triangles in 3D WSNs. It is more challenging since we need to define a new rule for any vertex, edge or triangle to determine whether its deletion will change the homology of the Rips complex.

- Coverage preserving node scheduling for WSNs

  For WSNs with very high density, the target field is often over covered. In order to extend the lifetime of WSNs, it is usually required to turn off some redundant nodes while preserving coverage. Although the homology based algorithm proposed in this thesis is mainly for coverage hole detection, it can also be used for node scheduling. It can be seen that when the WSN contains no coverage holes, some nodes can be switched off without changing the coverage according to the rule defined in the algorithm. This is because our algorithm does not change the homology in the process. This problem has been investigated in a centralized way by our group [100]. But the energy constraint of each node has not been considered in the investigation, which can be our future work.

- Auto-planning in heterogeneous networks

  Future heterogeneous networks will have more and more small cells overlaid by macro cells. The massive deployment of small cells faces a number of challenges, among which interference management is of utmost importance. The theory of algebraic topology can be both used to analyse the expected characteristics of network where spatial features are important but also to devise some new operating algorithms. One problem we have in mind of that is the automatic frequency planning in heterogeneous network. The problem is to allocate frequency bands dynamically so that the interferences are the least possible while guaranteeing a maximum coverage. We expect the algebraic topology can be useful for such a task. Indeed, one can construct two simplicial complexes: one representing coverage as before, the other representing

exclusion regions in the sense of cognitive radio, the exclusion region of a given BS means that any other BSs within this region can not use the same frequency sub-bands simultaneously. Using our previously defined algorithms [51, 100], we think that we can define frequency allocations satisfying as much as possible the two criteria.

Appendix A

Computation of Area $|S^-(r_0, r_1, \varphi_1)|$

A.1 Area $|S^-(r_0, r_1, \varphi_1)|$ in case $\sqrt{3} < \gamma \leq 2$

Here we give the detailed computation of the area $|S^-(r_0, r_1, \varphi_1)|$ in case $\sqrt{3} < \gamma \leq 2$. The region $S^-(r_0, r_1, \varphi_1)$ is shown in Figure A.1.

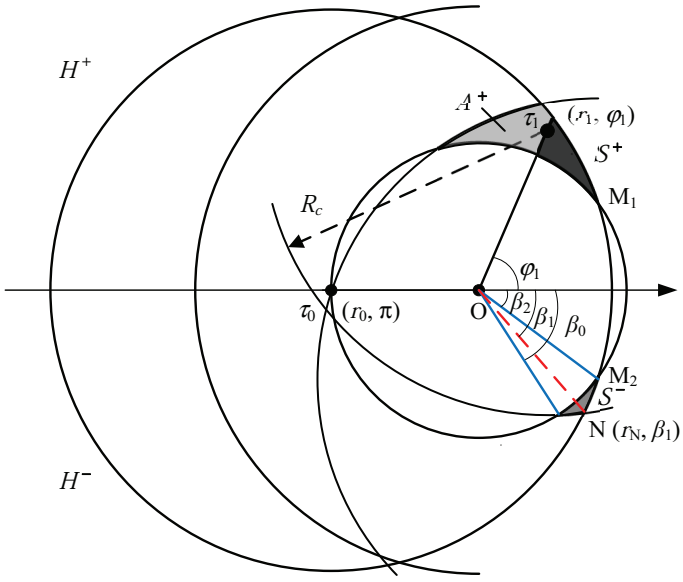


Figure A.1: Illustration of the region $S^-(r_0, r_1, \varphi_1)$ in the case $\sqrt{3} < \gamma \leq 2$

It can be seen from Figure A.1 that the area $|S^-(r_0, r_1, \varphi_1)|$ can be expressed as

$$|S^-(r_0, r_1, \varphi_1)| = \int_{\beta_0}^{\beta_2} d\varphi \int_{r_0}^{R_2(r_0, r_1, \varphi_1, \varphi)} r dr = \frac{1}{2} \int_{\beta_0}^{\beta_2} [R_2^2(r_0, r_1, \varphi_1, \varphi) - r_0^2] d\varphi \quad \text{(A.1)}$$

where $\beta_0 = \varphi_1 - \arccos \frac{r_0^2+r_1^2-R_c^2}{2r_0r_1}$, $\beta_2 = -\varphi_l = -2\arccos \frac{R_c}{2r_0}$ and

$$R_2(r_0, r_1, \varphi_1, \varphi) = \min(\sqrt{R_c^2 - r_0^2 \sin^2 \varphi} - r_0 \cos \varphi, \\ \sqrt{R_c^2 - r_1^2 \sin^2(\varphi - \varphi_1)} + r_1 \cos(\varphi - \varphi_1)) \tag{A.2}$$

In addition, we need to obtain β_1. For clear understanding, we give the triangle formed by τ_0, τ_1 and N in Figure A.2.

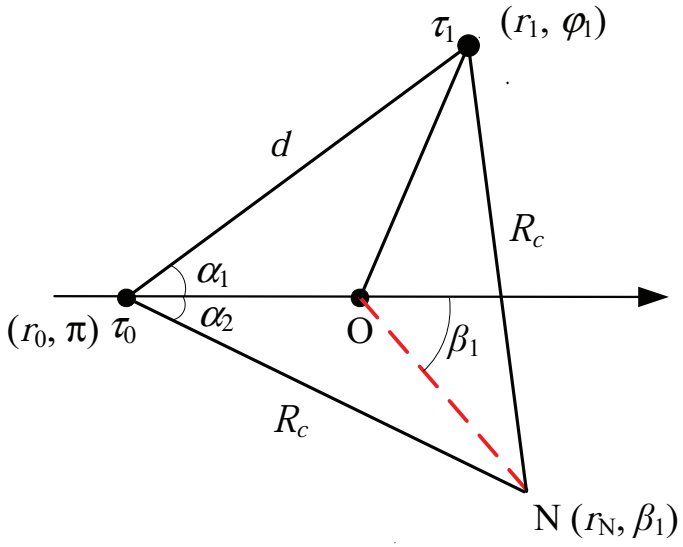


Figure A.2: For the computation of β_1

First, we can get

$$d = \sqrt{r_0^2 + r_1^2 + 2r_0r_1 \cos \varphi_1}$$

then

$$\alpha_1 = \arccos \frac{r_0^2 + d^2 - r_1^2}{2r_0d}$$

As

$$\alpha_1 + \alpha_2 = \arccos \frac{d}{2R_c}$$

So it can be obtained

$$\alpha_2 = \arccos \frac{d}{2R_c} - \arccos \frac{r_0^2 + d^2 - r_1^2}{2r_0d}$$

and

$$r_N = \sqrt{r_0^2 + R_c^2 - 2r_0R_c \cos \alpha_2}$$

Consequently, we can get β_1 as follows

$$\beta_1 = \arccos \frac{r_0^2 + r_N^2 - R_c^2}{2 r_0 r_N} - \pi$$

It can be seen from Figure A.1 that when $\beta_0 \leq \varphi \leq \beta_1$

$$R_2(r_0, r_1, \varphi_1, \varphi) = \sqrt{R_c^2 - r_1^2 \sin^2(\varphi - \varphi_1)} + r_1 \cos(\varphi - \varphi_1)$$

and when $\beta_1 < \varphi \leq \beta_2$

$$R_2(r_0, r_1, \varphi_1, \varphi) = \sqrt{R_c^2 - r_0^2 \sin^2 \varphi} - r_0 \cos \varphi$$

So (A.1) can be changed to

$$\begin{aligned} |S^-(r_0, r_1, \varphi_1)| =& \frac{1}{2} \int_{\beta_0}^{\beta_2} [R_2^2(r_0, r_1, \varphi_1, \varphi) - r_0^2] d\varphi \\ =& \frac{1}{2} \int_{\beta_0}^{\beta_1} \{[\sqrt{R_c^2 - r_1^2 \sin^2(\varphi - \varphi_1)} + r_1 \cos(\varphi - \varphi_1)]^2 - r_0^2\} d\varphi \\ &+ \frac{1}{2} \int_{\beta_1}^{\beta_2} \{[\sqrt{R_c^2 - r_0^2 \sin^2 \varphi} - r_0 \cos \varphi]^2 - r_0^2\} d\varphi \\ =& I'(r_1, \beta_1 - \varphi_1) - I'(r_1, \beta_0 - \varphi_1) + I(r_0, \beta_2) - I(r_0, \beta_1) \end{aligned} \tag{A.3}$$

where

$$I'(r_1, \varphi) = \frac{r_1^2 \sin 2\varphi}{4} + \frac{R_c^2 \varphi}{2} + \frac{R_c^2}{2} \arcsin \frac{r_1 \sin \varphi}{R_c} + \frac{r_1 \sin \varphi}{2} \sqrt{R_c^2 - r_1^2 \sin^2 \varphi} - \frac{r_0^2 \varphi}{2}$$

$$I(r_0, \varphi) = \frac{r_0^2 \sin 2\varphi}{4} + \frac{R_c^2 \varphi}{2} - \frac{R_c^2}{2} \arcsin \frac{r_0 \sin \varphi}{R_c} - \frac{r_0 \sin \varphi}{2} \sqrt{R_c^2 - r_0^2 \sin^2 \varphi} - \frac{r_0^2 \varphi}{2}$$

Replace r_1 by r_0 in (A.3), we can get the area of $S^-(r_0, r_0, \varphi_1)$.

A.2 Area $|S^-(r_0, r_1, \varphi_1)|$ in case $\gamma > 2$

As illustrated in Section 3.3.4, two situations should be considered in case $\gamma > 2$. They are $R_s < d_0 \leq R_c/2$ and $R_c/2 < d_0 \leq R_c/\sqrt{3}$. When $R_c/2 < d_0 \leq R_c/\sqrt{3}$, the computation of $|S^-(r_0, r_1, \varphi_1)|$ is the same as that in Appendix A.1. So we only

compute the area $|S^-(r_0, r_1, \varphi_1)|$ in the situation $R_s < d_0 \leq R_c/2$ in this part.

The area $|S^-(r_0, r_1, \varphi_1)|$ can also be expressed as

$$|S^-(r_0, r_1, \varphi_1)| = \int_{\beta_0}^{0} d\varphi \int_{r_0}^{R_2(r_0, r_1, \varphi_1, \varphi)} r dr = \frac{1}{2} \int_{\beta_0}^{0} [R_2^2(r_0, r_1, \varphi_1, \varphi) - r_0^2] d\varphi$$

where $R_2(r_0, r_1, \varphi_1, \varphi)$ is shown in (A.2) and

$$\beta_0 = \begin{cases} \varphi_1 - \pi & \text{if } r_1 \leq R_c - r_0 \\ \varphi_1 - \arccos \frac{r_0^2 + r_1^2 - R_c^2}{2 r_0 r_1} & \text{otherwise} \end{cases}$$

The two cases for β_0 are illustrated in Figure A.3.

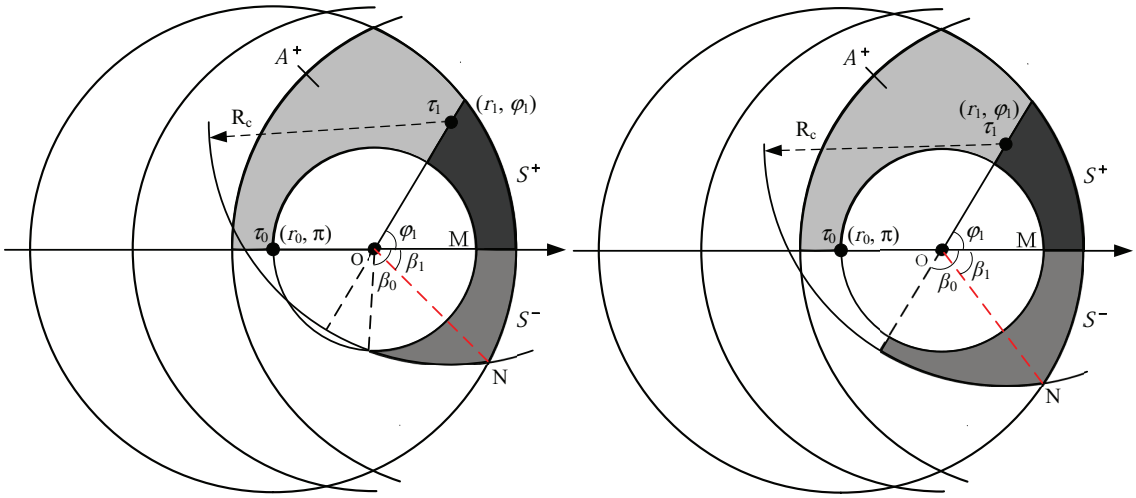


Figure A.3: Illustration of two cases for $\beta_0(r0, r1, \varphi_1)$

Furthermore, since the upper limit of integral for φ is 0, we need to check the value of $R_2(r_0, r_1, \varphi_1, 0)$.

If $R_2(r_0, r_1, \varphi_1, 0) = \sqrt{R_c^2 - r_1^2 \sin^2 \varphi_1} + r_1 \cos \varphi_1$, it means that when $\beta_0 \leq \varphi \leq 0$, $R_2(r_0, r_1, \varphi_1, \varphi) = \sqrt{R_c^2 - r_1^2 \sin^2(\varphi - \varphi_1)} + r_1 \cos(\varphi - \varphi_1)$, which is shown in Figure A.4.

Then we can get the area $|S^-(r_0, r_1, \varphi_1)|$ as

$$\begin{aligned} |S^-(r_0, r_1, \varphi_1)| &= \frac{1}{2} \int_{\beta_0}^{0} \{[\sqrt{R_c^2 - r_1^2 \sin^2(\varphi - \varphi_1)} + r_1 \cos(\varphi - \varphi_1)]^2 - r_0^2\} d\varphi \\ &= I'(r_1, -\varphi_1) - I'(r_1, \beta_0 - \varphi_1) \end{aligned}$$

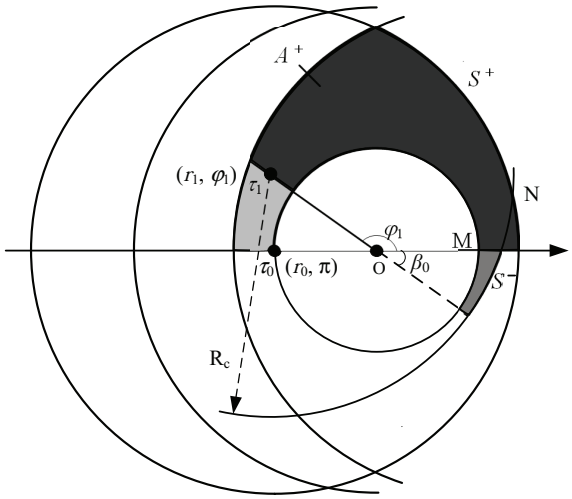


Figure A.4: Illustration of $R_2(r_0, r_1, \varphi_1, 0)$

where

$$I'(r_1, \varphi) = \frac{r_1^2 \sin 2\varphi}{4} + \frac{R_c^2 \varphi}{2} + \frac{R_c^2}{2} \arcsin \frac{r_1 \sin \varphi}{R_c} + \frac{r_1 \sin \varphi}{2} \sqrt{R_c^2 - r_1^2 \sin^2 \varphi} - \frac{r_0^2 \varphi}{2}$$

If $R_2(r_0, r_1, \varphi_1, 0) = R_c - r_0$, which is the case in Figure A.3, then we can obtain β_1 as in Section A.1 and the area $|S^-(r_0, r_1, \varphi_1)|$ as follows.

$$\begin{aligned} |S^-(r_0, r_1, \varphi_1)| &= I'(r_1, \beta_1 - \varphi_1) - I'(r_1, \beta_0 - \varphi_1) + I(r_0, 0) - I(r_0, \beta_1) \\ &= I'(r_1, \beta_1 - \varphi_1) - I'(r_1, \beta_0 - \varphi_1) - I(r_0, \beta_1) \end{aligned}$$

where

$$I(r_0, \varphi) = \frac{r_0^2 \sin 2\varphi}{4} + \frac{R_c^2 \varphi}{2} - \frac{R_c^2}{2} \arcsin \frac{r_0 \sin \varphi}{R_c} - \frac{r_0 \sin \varphi}{2} \sqrt{R_c^2 - r_0^2 \sin^2 \varphi} - \frac{r_0^2 \varphi}{2}$$

Appendix B

Detailed Values of Simulation Results and Bounds on Sphere

Table B.1: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.0, 2.2$ (%)

λ	$\gamma = 2.0$			$\gamma = 2.2$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.0006	0.0006	0.0011	0.0126	0.0134	0.0239
0.002	0.0032	0.0032	0.0063	0.0678	0.0694	0.1265
0.003	0.0076	0.0076	0.0148	0.1537	0.1558	0.2835
0.004	0.0128	0.0129	0.0249	0.2451	0.2531	0.4463
0.005	0.0177	0.0185	0.0345	0.3223	0.3395	0.5807
0.006	0.0216	0.0226	0.0421	0.3753	0.3934	0.6673
0.007	0.0244	0.0257	0.0476	0.4020	0.4285	0.7064
0.008	0.0258	0.0275	0.0503	0.4052	0.4342	0.7047
0.009	0.0260	0.0288	0.0508	0.3898	0.4216	0.6688
0.010	0.0253	0.0280	0.0494	0.3617	0.3902	0.6127
0.011	0.0239	0.0264	0.0466	0.3258	0.3550	0.5468
0.012	0.0220	0.0248	0.0432	0.2866	0.3131	0.4756
0.013	0.0199	0.0229	0.0389	0.2471	0.2684	0.4054
0.014	0.0176	0.0204	0.0344	0.2094	0.2248	0.3395
0.015	0.0154	0.0175	0.0302	0.1750	0.1915	0.2799
0.016	0.0133	0.0150	0.0259	0.1443	0.1591	0.2284
0.017	0.0113	0.0131	0.0219	0.1178	0.1314	0.1842
0.018	0.0095	0.0108	0.0184	0.0952	0.1088	0.1472
0.019	0.0079	0.0096	0.0155	0.0763	0.0853	0.1166
0.020	0.0066	0.0079	0.0127	0.0607	0.0686	0.0919

Table B.2: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.4, 2.6$ (%)

λ	$\gamma = 2.4$			$\gamma = 2.6$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.0617	0.0621	0.1120	0.1800	0.1810	0.3198
0.002	0.3130	0.3227	0.5556	0.8582	0.8779	1.4759
0.003	0.6710	0.6972	1.1663	1.7340	1.8110	2.8902
0.004	1.0128	1.0594	1.7231	2.4720	2.6030	3.9935
0.005	1.2626	1.3281	2.1023	2.9173	3.0835	4.5714
0.006	1.3957	1.4748	2.2780	3.0599	3.2596	4.6534
0.007	1.4211	1.5142	2.2720	2.9628	3.1745	4.3775
0.008	1.3632	1.4612	2.1344	2.7089	2.9099	3.8895
0.009	1.2502	1.3520	1.9185	2.3731	2.5526	3.3164
0.010	1.1070	1.1988	1.6665	2.0117	2.1574	2.7370
0.011	0.9532	1.0349	1.4068	1.6620	1.7819	2.2050
0.012	0.8024	0.8729	1.1615	1.3451	1.4413	1.7422
0.013	0.6629	0.7189	0.9421	1.0706	1.1447	1.3552
0.014	0.5391	0.5847	0.7523	0.8407	0.9024	1.0404
0.015	0.4327	0.4723	0.5932	0.6527	0.6998	0.7922
0.016	0.3434	0.3673	0.4616	0.5020	0.5313	0.5982
0.017	0.2698	0.2929	0.3578	0.3832	0.4079	0.4488
0.018	0.2103	0.2276	0.2740	0.2905	0.3057	0.3342
0.019	0.1627	0.1753	0.2088	0.2191	0.2300	0.2481
0.020	0.1251	0.1371	0.1583	0.1645	0.1726	0.1840

Table B.3: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 5R_s, \gamma = 2.8, 3.0$ (%)

λ	$\gamma = 2.8$			$\gamma = 3.0$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.4052	0.4128	0.7082	0.7780	0.7935	1.3416
0.002	1.8091	1.8711	3.0291	3.2417	3.3616	5.2894
0.003	3.4351	3.5920	5.5143	5.7720	6.0728	8.9249
0.004	4.6179	4.8769	7.1119	7.3111	7.7338	10.7383
0.005	5.1564	5.4688	7.6306	7.7284	8.2208	10.8076
0.006	5.1347	5.4757	7.3086	7.3200	7.7953	9.7798
0.007	4.7361	5.0693	6.4998	6.4513	6.8566	8.2651
0.008	4.1386	4.4233	5.4859	5.4106	5.7332	6.6721
0.009	3.4763	3.7200	4.4633	4.3806	4.6133	5.2202
0.010	2.8346	3.0127	3.5300	3.4568	3.6285	3.9983
0.011	2.2594	2.4024	2.7364	2.6767	2.8028	3.0159
0.012	1.7695	1.8684	2.0897	2.0436	2.1208	2.2512
0.013	1.3667	1.4476	1.5780	1.5438	1.5944	1.6691
0.014	1.0442	1.0911	1.1804	1.1570	1.1919	1.2311
0.015	0.7908	0.8212	0.8782	0.8619	0.8807	0.9062
0.016	0.5948	0.6169	0.6501	0.6392	0.6463	0.6644
0.017	0.4449	0.4592	0.4792	0.4723	0.4838	0.4868
0.018	0.3313	0.3397	0.3529	0.3481	0.3535	0.3565
0.019	0.2458	0.2544	0.2590	0.2561	0.2565	0.2608
0.020	0.1819	0.1832	0.1899	0.1881	0.1899	0.1908

Table B.4: Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 5R_s$

λ	Simulation results for $p_{\text{sec}}(\lambda)$ under different γ (%)					
	2.0	2.2	2.4	2.6	2.8	3.0
0.001	0	0.0001	0.0004	0.0014	0.0026	0.0051
0.002	0.0001	0.0008	0.0036	0.0103	0.0220	0.0391
0.003	0.0002	0.0028	0.0114	0.0297	0.0583	0.0924
0.004	0.0004	0.0055	0.0218	0.0518	0.0931	0.1382
0.005	0.0008	0.0098	0.0317	0.0715	0.1175	0.1544
0.006	0.0010	0.0124	0.0425	0.0832	0.1216	0.1491
0.007	0.0015	0.0153	0.0481	0.0880	0.1169	0.1292
0.008	0.0018	0.0184	0.0491	0.0833	0.1018	0.1022
0.009	0.0021	0.0181	0.0482	0.0762	0.0861	0.0756
0.010	0.0022	0.0177	0.0460	0.0641	0.0660	0.0544
0.011	0.0023	0.0184	0.0406	0.0530	0.0492	0.0367
0.012	0.0026	0.0174	0.0349	0.0424	0.0359	0.0236
0.013	0.0024	0.0158	0.0297	0.0327	0.0257	0.0153
0.014	0.0022	0.0138	0.0244	0.0238	0.0170	0.0092
0.015	0.0022	0.0115	0.0197	0.0183	0.0117	0.0065
0.016	0.0019	0.0099	0.0146	0.0136	0.0078	0.0034
0.017	0.0016	0.0083	0.0124	0.0099	0.0048	0.0020
0.018	0.0014	0.0069	0.0092	0.0065	0.0034	0.0013
0.019	0.0013	0.0056	0.0071	0.0044	0.0021	0.0007
0.020	0.0010	0.0047	0.0055	0.0033	0.0013	0.0004

Table B.5: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.0, 2.2$ (%)

λ	$\gamma = 2.0$			$\gamma = 2.2$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.0005	0.0006	0.0011	0.0126	0.0127	0.0239
0.002	0.0031	0.0031	0.0061	0.0677	0.0689	0.1265
0.003	0.0074	0.0081	0.0145	0.1535	0.1571	0.2829
0.004	0.0125	0.0134	0.0245	0.2445	0.2519	0.4454
0.005	0.0173	0.0183	0.0337	0.3213	0.3331	0.5782
0.006	0.0211	0.0224	0.0412	0.3739	0.3895	0.6652
0.007	0.0238	0.0251	0.0465	0.4001	0.4232	0.7034
0.008	0.0252	0.0279	0.0492	0.4030	0.4318	0.6990
0.009	0.0254	0.0278	0.0498	0.3874	0.4138	0.6649
0.010	0.0247	0.0273	0.0483	0.3591	0.3900	0.6089
0.011	0.0233	0.0255	0.0456	0.3233	0.3524	0.5424
0.012	0.0214	0.0241	0.0419	0.2841	0.3122	0.4708
0.013	0.0193	0.0221	0.0378	0.2448	0.2680	0.4008
0.014	0.0171	0.0196	0.0332	0.2073	0.2293	0.3355
0.015	0.0149	0.0174	0.0291	0.1730	0.1899	0.2775
0.016	0.0129	0.0148	0.0250	0.1426	0.1587	0.2259
0.017	0.0110	0.0127	0.0213	0.1163	0.1294	0.1820
0.018	0.0092	0.0109	0.0178	0.0939	0.1056	0.1453
0.019	0.0077	0.0091	0.0150	0.0752	0.0813	0.1151
0.020	0.0064	0.0069	0.0123	0.0598	0.0664	0.0902

Table B.6: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.4, 2.6$ (%)

λ	$\gamma = 2.4$			$\gamma = 2.6$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.0621	0.0633	0.1126	0.1815	0.1844	0.3229
0.002	0.3143	0.3233	0.5584	0.8640	0.8882	1.4891
0.003	0.6731	0.6924	1.1711	1.7433	1.8151	2.9093
0.004	1.0149	1.0584	1.7291	2.4819	2.6077	4.0144
0.005	1.2638	1.3277	2.1059	2.9249	3.0912	4.5867
0.006	1.3957	1.4768	2.2784	3.0639	3.2581	4.6614
0.007	1.4196	1.5107	2.2698	2.9629	3.1699	4.3798
0.008	1.3604	1.4585	2.1296	2.7056	2.9030	3.8841
0.009	1.2463	1.3429	1.9124	2.3673	2.5450	3.3067
0.010	1.1025	1.1897	1.6593	2.0044	2.1548	2.7270
0.011	0.9484	1.0268	1.3993	1.6540	1.7771	2.1923
0.012	0.7975	0.8690	1.1540	1.3371	1.4297	1.7307
0.013	0.6582	0.7169	0.9358	1.0631	1.1390	1.3452
0.014	0.5348	0.5824	0.7457	0.8338	0.8946	1.0318
0.015	0.4288	0.4649	0.5876	0.6467	0.6926	0.7843
0.016	0.3400	0.3706	0.4575	0.4969	0.5281	0.5911
0.017	0.2669	0.2874	0.3528	0.3789	0.3971	0.4428
0.018	0.2078	0.2266	0.2704	0.2870	0.3027	0.3301
0.019	0.1607	0.1715	0.2054	0.2162	0.2276	0.2451
0.020	0.1234	0.1331	0.1556	0.1621	0.1693	0.1812

Table B.7: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 10R_s, \gamma = 2.8, 3.0$ (%)

λ	$\gamma = 2.8$			$\gamma = 3.0$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.4095	0.4171	0.7178	0.7878	0.7988	1.3641
0.002	1.8247	1.8936	3.0625	3.2742	3.4057	5.3602
0.003	3.4583	3.6133	5.5592	5.8160	6.1184	9.0140
0.004	4.6408	4.8961	7.1570	7.3505	7.7578	10.8118
0.005	5.1732	5.5014	7.6589	7.7541	8.2506	10.8531
0.006	5.1432	5.4654	7.3267	7.3304	7.8045	9.7974
0.007	4.7367	5.0489	6.5027	6.4492	6.8548	8.2604
0.008	4.1330	4.4019	5.4781	5.4000	5.7205	6.6557
0.009	3.4668	3.6944	4.4466	4.3653	4.6061	5.1974
0.010	2.8230	2.9974	3.5140	3.4399	3.6109	3.9742
0.011	2.2473	2.3838	2.7199	2.6601	2.7777	2.9941
0.012	1.7578	1.8524	2.0745	2.0285	2.1101	2.2329
0.013	1.3561	1.4223	1.5644	1.5306	1.5815	1.6539
0.014	1.0349	1.0834	1.1698	1.1458	1.1735	1.2186
0.015	0.7830	0.8180	0.8687	0.8527	0.8779	0.8950
0.016	0.5883	0.6091	0.6427	0.6317	0.6456	0.6561
0.017	0.4396	0.4597	0.4734	0.4664	0.4736	0.4804
0.018	0.3270	0.3373	0.3480	0.3434	0.3487	0.3516
0.019	0.2424	0.2491	0.2553	0.2524	0.2560	0.2568
0.020	0.1792	0.1815	0.1870	0.1852	0.1870	0.1879

Table B.8: Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 10R_s$

λ	Simulation results for $p_{\text{sec}}(\lambda)$ under different γ (%)					
	2.0	2.2	2.4	2.6	2.8	3.0
0.001	0	0.0001	0.0003	0.0011	0.0024	0.0048
0.002	0.0001	0.0009	0.0034	0.0106	0.0210	0.0386
0.003	0.0002	0.0025	0.0115	0.0286	0.0538	0.0901
0.004	0.0005	0.0055	0.0227	0.0513	0.0905	0.1335
0.005	0.0007	0.0088	0.0316	0.0696	0.1109	0.1503
0.006	0.0010	0.0126	0.0414	0.0808	0.1207	0.1455
0.007	0.0015	0.0153	0.0469	0.0870	0.1150	0.1245
0.008	0.0018	0.0162	0.0475	0.0805	0.0995	0.0988
0.009	0.0022	0.0181	0.0471	0.0732	0.0811	0.0720
0.010	0.0023	0.0179	0.0448	0.0633	0.0645	0.0512
0.011	0.0024	0.0180	0.0397	0.0505	0.0478	0.0349
0.012	0.0023	0.0164	0.0340	0.0411	0.0350	0.0230
0.013	0.0023	0.0148	0.0297	0.0322	0.0248	0.0151
0.014	0.0019	0.0131	0.0236	0.0237	0.0173	0.0091
0.015	0.0019	0.0120	0.0193	0.0177	0.0113	0.0053
0.016	0.0017	0.0100	0.0149	0.0127	0.0077	0.0031
0.017	0.0016	0.0082	0.0112	0.0091	0.0049	0.0019
0.018	0.0013	0.0068	0.0087	0.0065	0.0032	0.0013
0.019	0.0013	0.0056	0.0063	0.0047	0.0020	0.0005
0.020	0.0010	0.0043	0.0049	0.0032	0.0012	0.0005

Table B.9: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.0, 2.2$ (%)

λ	$\gamma = 2.0$			$\gamma = 2.2$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.0006	0.0006	0.0011	0.0126	0.0130	0.0239
0.002	0.0031	0.0031	0.0061	0.0677	0.0698	0.1264
0.003	0.0078	0.0078	0.0144	0.1534	0.1578	0.2828
0.004	0.0129	0.0129	0.0243	0.2443	0.2547	0.4451
0.005	0.0187	0.0187	0.0334	0.3209	0.3385	0.5778
0.006	0.0226	0.0226	0.0408	0.3733	0.3920	0.6644
0.007	0.0247	0.0247	0.0461	0.3995	0.4243	0.7028
0.008	0.0273	0.0273	0.0487	0.4022	0.4281	0.6985
0.009	0.0278	0.0278	0.0492	0.3865	0.4168	0.6645
0.010	0.0271	0.0271	0.0477	0.3582	0.3864	0.6079
0.011	0.0257	0.0257	0.0451	0.3224	0.3477	0.5405
0.012	0.0238	0.0238	0.0414	0.2833	0.3099	0.4701
0.013	0.0218	0.0218	0.0373	0.2440	0.2659	0.4001
0.014	0.0193	0.0193	0.0330	0.2066	0.2272	0.3350
0.015	0.0172	0.0172	0.0289	0.1724	0.1900	0.2758
0.016	0.0149	0.0149	0.0249	0.1421	0.1572	0.2250
0.017	0.0127	0.0127	0.0211	0.1158	0.1297	0.1817
0.018	0.0108	0.0108	0.0177	0.0935	0.1054	0.1447
0.019	0.0086	0.0086	0.0148	0.0748	0.0838	0.1151
0.020	0.0074	0.0074	0.0122	0.0595	0.0672	0.0903

Table B.10: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.4, 2.6$ (%)

λ	$\gamma = 2.4$			$\gamma = 2.6$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.1847	0.0634	0.1129	0.1847	0.1847	0.3243
0.002	0.8858	0.3206	0.5596	0.8858	0.8858	1.4928
0.003	1.8142	0.6908	1.1724	1.8142	1.8142	2.9170
0.004	2.6096	1.0587	1.7303	2.6096	2.6096	4.0212
0.005	3.0924	1.3243	2.1080	3.0924	3.0924	4.5932
0.006	3.2566	1.4775	2.2798	3.2566	3.2566	4.6655
0.007	3.1608	1.5086	2.2689	3.1608	3.1608	4.3784
0.008	2.8923	1.4484	2.1290	2.8923	2.8923	3.8844
0.009	2.5444	1.3401	1.9116	2.5444	2.5444	3.3054
0.010	2.1512	1.1832	1.6566	2.1512	2.1512	2.7228
0.011	1.7772	1.0236	1.3970	1.7772	1.7772	2.1908
0.012	1.4286	0.8615	1.1524	1.4286	1.4286	1.7273
0.013	1.1334	0.7100	0.9333	1.1334	1.1334	1.3414
0.014	0.8922	0.5779	0.7435	0.8922	0.8922	1.0291
0.015	0.6854	0.4626	0.5853	0.6854	0.6854	0.7821
0.016	0.5240	0.3705	0.4564	0.5240	0.5240	0.5892
0.017	0.4017	0.2894	0.3520	0.4017	0.4017	0.4410
0.018	0.2998	0.2256	0.2698	0.2998	0.2998	0.3283
0.019	0.2279	0.1749	0.2054	0.2279	0.2279	0.2439
0.020	0.1689	0.1334	0.1553	0.1689	0.1689	0.1803

Table B.11: $p_{sl}(\lambda)$, $p_{su}(\lambda)$ and $p_s(\lambda)$ under $R = 100R_s, \gamma = 2.8, 3.0$ (%)

λ	$\gamma = 2.8$			$\gamma = 3.0$		
	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$	$p_{sl}(\lambda)$	$p_s(\lambda)$	$p_{su}(\lambda)$
0.001	0.4109	0.4188	0.7215	0.7911	0.8086	1.3723
0.002	1.8300	1.8796	3.0737	3.2851	3.4049	5.3829
0.003	3.4661	3.6178	5.5773	5.8308	6.1097	9.0459
0.004	4.6485	4.9135	7.1730	7.3637	7.7923	10.8368
0.005	5.1789	5.4868	7.6722	7.7628	8.2563	10.8698
0.006	5.1460	5.4754	7.3318	7.3339	7.7878	9.8022
0.007	4.7368	5.0488	6.5032	6.4484	6.8581	8.2612
0.008	4.1311	4.4128	5.4758	5.3964	5.7122	6.6519
0.009	3.4635	3.6919	4.4426	4.3602	4.5862	5.1919
0.010	2.8191	2.9994	3.5084	3.4342	3.6076	3.9679
0.011	2.2432	2.3784	2.7148	2.6545	2.7726	2.9887
0.012	1.7539	1.8533	2.0699	2.0234	2.1079	2.2273
0.013	1.3526	1.4219	1.5597	1.5262	1.5761	1.6480
0.014	1.0318	1.0794	1.1654	1.1421	1.1773	1.2141
0.015	0.7803	0.8173	0.8662	0.8496	0.8703	0.8916
0.016	0.5861	0.6069	0.6396	0.6292	0.6436	0.6538
0.017	0.4378	0.4551	0.4713	0.4644	0.4714	0.4784
0.018	0.3255	0.3356	0.3464	0.3418	0.3490	0.3498
0.019	0.2412	0.2498	0.2541	0.2512	0.2555	0.2558
0.020	0.1783	0.1832	0.1860	0.1843	0.1848	0.1868

Table B.12: Simulation results for $p_s^{\text{sec}}(\lambda)$ when $R = 100R_s$

λ	Simulation results for $p_{\text{sec}}(\lambda)$ under different γ (%)					
	2.0	2.2	2.4	2.6	2.8	3.0
0.001	0	0.0001	0.0003	0.0014	0.0028	0.0054
0.002	0.0001	0.0008	0.0037	0.0101	0.0208	0.0374
0.003	0.0002	0.0026	0.0112	0.0297	0.0556	0.0916
0.004	0.0005	0.0055	0.0224	0.0511	0.0908	0.1327
0.005	0.0007	0.0090	0.0325	0.0705	0.1127	0.1508
0.006	0.0010	0.0126	0.0424	0.0817	0.1197	0.1436
0.007	0.0015	0.0157	0.0465	0.0847	0.1142	0.1256
0.008	0.0018	0.0170	0.0481	0.0818	0.0992	0.0996
0.009	0.0021	0.0190	0.0481	0.0743	0.0811	0.0731
0.010	0.0021	0.0183	0.0443	0.0622	0.0639	0.0520
0.011	0.0023	0.0174	0.0396	0.0525	0.0478	0.0362
0.012	0.0022	0.0170	0.0346	0.0411	0.0352	0.0235
0.013	0.0021	0.0153	0.0294	0.0316	0.0244	0.0143
0.014	0.0020	0.0137	0.0234	0.0239	0.0165	0.0088
0.015	0.0020	0.0113	0.0187	0.0180	0.0118	0.0053
0.016	0.0018	0.0099	0.0153	0.0129	0.0072	0.0035
0.017	0.0016	0.0087	0.0117	0.0090	0.0048	0.0020
0.018	0.0013	0.0068	0.0092	0.0061	0.0032	0.0011
0.019	0.0012	0.0061	0.0071	0.0046	0.0021	0.0008
0.020	0.0010	0.0048	0.0053	0.0032	0.0012	0.0004

References

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002. 1, 33

[2] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, August 2008. 1, 33

[3] Dan Li, Kerry D. Wong, Yu Hen Hu, and Akbar M. Sayeed. Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine*, 19(2): 17–29, March 2002. 1, 33

[4] Richard R. Brooks, Parameswaran Ramanathan, and Akbar M. Sayeed. Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8):1163–1171, August 2003. 1, 33

[5] Anish Arora, Prabal Dutta, Sandip Bapat, Vinod Kulathumani, Hongwei Zhang, Vinayak Naik, Vineet Mittal, Hui Cao, Murat Demirbas, Mohamed G. Gouda, Young ri Choi, Ted Herman, Sandeep S. Kulkarni, Umamaheswaran Arumugam, Mikhail Nesterenko, Adnan Vora, and Mark Miyashita. A line in the sand: a wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605–634, December 2004. 1, 33

[6] Gyula Simon, Miklós Maróti, Ákos Lédeczi, György Balogh, Branislav Kusy, András Nádas, Gábor Pap, János Sallai, and Ken Frampton. Sensor network-based countersniper system. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 1–12, Baltimore, MD, USA, November 2004. 1, 33

[7] Songhwai Oh, Phoebus Chen, Michael Manzo, and Shankar Sastry. Instrumenting wireless sensor networks for real-time surveillance. In *IEEE Inter-*

national Conference on Robotics and Automation, pages 3128–3133, Orlando, Florida, USA, May 2006. 1, 33

[8] Mikkel Baun Kjaergaard, Jakob Langdal, Torben Godsk, and Thomas Toftkjaer. Entracked: energy-efficient robust position tracking for mobile devices. In *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys'09)*, pages 221–234, Kraków, Poland, June 2009. 1, 33

[9] Mikkel Baun Kjaergaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys'11)*, pages 307–320, Bethesda, Maryland, USA, June 2011. 1, 33

[10] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS X)*, pages 96–107, San Jose, CA, USA, October 2002. 1, 33

[11] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA'02)*, pages 88–97, Atlanta, Georgia, USA, September 2002. 1, 33

[12] Robert Szewczyk, Eric Osterweil, Joseph Polastre, Michael Hamilton, Alan Mainwaring, and Deborah Estrin. Habitat monitoring with sensor networks. *Communications of the ACM*, 47(6):34–40, June 2004. 1, 33

[13] Gilman Tolle, Joseph Polastre, Robert Szewczyk, David Culler, Neil Turner, Kevin Tu, Stephen Burgess, Todd Dawson, Phil Buonadonna, David Gay, and Wei Hong. A macroscope in the redwoods. In *Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys'05)*, pages 51–63, San Diego, California, USA, November 2005. 1, 33

[14] Geoffrey Werner-Allen, Konrad Lorincz, Matt Welsh, Omar Marcillo, Jeff Johnson, Mario Ruiz, and Jonathan Lees. Deploying a wireless sensor net-

work on an active volcano. *IEEE Internet Computing*, 10(2):18–25, March 2006. 1, 33, 75

[15] Lufeng Mo, Yuan He, Yunhao Liu, Jizhong Zhao, Shao-Jie Tang, Xiang-Yang Li, and Guojun Dai. Canopy closure estimates with greenorbs: sustainable sensing in the forest. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, pages 99–112, Berkeley, California, USA, November 2009. 1, 33, 75

[16] Wen-Zhan Song, Renjie Huang, Mingsen Xu, Behrooz A. Shirazi, and Richard LaHusen. Design and deployment of sensor network for real-time high-fidelity volcano monitoring. *IEEE Transaction on Parallel and Distributed Systems*, 21(11):1658–1674, November 2010. 1, 33

[17] Xufei Mao, Xin Miao, Yuan He, Xiang-Yang Li, and Yunhao Liu. Citysee: Urban CO_2 monitoring with sensors. In *IEEE INFOCOM*, pages 1611–1619, Orlando, FL, USA, March 2012. 1, 33

[18] Ning Xu, Sumit Rangwala, Krishna Kant Chintalapudi, Deepak Ganesan, Alan Broad, Ramesh Govindanand, and Deborah Estrin. A wireless sensor network for structural monitoring. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, pages 13–24, Baltimore, MD, USA, November 2004. 1, 33

[19] Krishna Chintalapudi, Tat Fu, Jeongyeup Paek, Nupur Kothari, Sumit Rangwala, John Caffrey, Ramesh Govindan, Erik Johnson, and Sami Masri. Monitoring civil structures with a wireless sensor network. *IEEE Internet Computing*, 10(2):26–34, March-April 2006. 1, 33

[20] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th International Symposium on Information Processing in Sensor Networks*, pages 254–263, Cambridge, Massachusetts, USA, April 2007. 1, 33

[21] Kameswari Chebrolu, Bhaskaran Raman, Nilesh Mishra, Phani Kumar Valiveti, and Raj Kumar. Brimon: a sensor network system for railway bridge monitoring. In *Proceedings of the 6th international conference on Mobile systems, applications, and services (MobiSys'08)*, pages 2–14, Breckenridge, CO, USA, June 2008. 1, 33

[22] Frank Stajano, Neil Hoult, Ian J. Wassell, Peter Bennett, Campbell Middleton, and Kenichi Soga. Smart bridges, smart tunnels: Transforming wireless sensor networks from research prototypes into robust engineering infrastructure. *Ad Hoc Networks*, 8(8):872–888, November 2010. 1, 33

[23] Christos Efstratiou, Nigel Davies, Gerd Kortuem, Joe Finney, Rob Hooper, and Mark Lowton. Experiences of designing and deploying intelligent sensor nodes to monitor hand-arm vibrations in the field. In *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys'07)*, pages 127–138, San Juan, Puerto Rico, USA, June 2007. 1, 33

[24] Chris R. Baker, Kenneth Armijo, Simon Belka, Merwan Benhabib, Vikas Bhargava, Nathan Burkhart, Artin Der Minassians, Gunes Dervisoglu, Lilia Gutnik, M. Brent Haick, Christine Ho, Mike Koplow, Jennifer Mangold, Stefanie Robinson, Matt Rosa, Miclas Schwartz, Christo Sims, Hanns Stoffregen, Andrew Waterbury, Eli S. Leland, Trevor Pering, and Paul K. Wright. Wireless sensor networks for home health care. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, volume 2, pages 832–837, Niagara Falls, Ontario, Canada, May 2007. 1, 33

[25] Konrad Lorincz, Bor rong Chen, Geoffrey Werner Challen, Atanu Roy Chowdhury, Shyamal Patel, Paolo Bonato, and Matt Welsh. Mercury: a wearable sensor network platform for high-fidelity motion analysis. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys'09)*, pages 183–196, Berkeley, CA, USA, November 2009. 1, 33

[26] Eugene I. Shih, Ali H. Shoeb, and John V. Guttag. Sensor selection for energy-efficient ambulatory medical monitoring. In *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys'09)*, pages 347–358, Kraków, Poland, June 2009. 1, 33

[27] Jungkeun Yoon, Brian Noble, and Mingyan Liu. Surface street traffic estimation. In *Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys'07)*, pages 220–232, San Juan, Puerto Rico, USA, June 2007. 1, 33

[28] Xu Li, Wei Shu, Minglu Li, Hong-Yu Huang, Pei-En Luo, and Min-You Wu. Performance evaluation of vehicle-based mobile sensor networks for traffic

monitoring. *IEEE Transactions on Vehicular Technology*, 58(4):1647–1653, May 2009. 1, 33

[29] Malik Tubaishat, Peng Zhuang, Qi Qi, and Yi Shang. Wireless sensor networks in intelligent transportation systems. *Wireless Communications & Mobile Computing*, 9(3):287–302, March 2009. 1, 33

[30] T. Semertzidis, K. Dimitropoulos, A. Koutsia, and N. Grammalidis. Video sensor network for real-time traffic monitoring and surveillance. *Intelligent Transport Systems, IET*, 4(2):103–112, June 2010. 1, 33

[31] A. Pascale, M. Nicoli, F. Deflorio, B. Dalla Chiara, and U. Spagnolini. Wireless sensor networks for traffic management and road safety. *Intelligent Transport Systems, IET*, 6(1):67–77, March 2012. 1, 33

[32] M. Bottero, B. Dalla Chiara, and F.P. Deflorio. Wireless sensor networks for traffic monitoring in a logistic centre. *Transportation Research Part C: Emerging Technologies*, 26:99–124, January 2013. 1, 33

[33] Seapahn Meguerdichian, Farinaz Koushanfar, Miodrag Potkonjak, and Mani B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *IEEE INFOCOM*, pages 1380–1387, Anchorage, Alaska, USA, April 2001. 1, 33

[34] Mihaela Cardei and Jie Wu. *Coverage in Wireless Sensor Networks in Handbook of Sensor Networks*, chapter 19. CRC Press, 2004. 1, 33

[35] Guiling Wang, Guohong Cao, and Thomas F. La Porta. Movement-assisted sensor deployment. In *IEEE INFOCOM*, pages 2469–2479, Hong Kong, China, March 2004. 2, 34, 40

[36] Guiling Wang, Guohong Cao, Tom La Porta, and Wensheng Zhang. Sensor relocation in mobile sensor networks. In *IEEE INFOCOM*, volume 4, pages 2302–2312, Miami, FL, USA, March 2005. 2, 34

[37] Yue Wang, Jie Gao, and Joseph S.B. Mitchell. Boundary recognition in sensor networks by topological methods. In *ACM MobiCom*, pages 122–133, Los Angeles, California, USA, September 2006. 2, 34, 42, 43

[38] Robert Ghrist and Abubakr Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proceedings of the 4th International Conference*

on Information Processing in Sensor Networks, pages 254–260, Los Angeles, California, USA, April 2005. 2, 8, 34, 49, 51, 52, 53, 104

[39] Mark Anthony Armstrong. *Basic Topology*. Springer, 1983. 4, 45

[40] James R. Munkres. *Elements of algebraic topology*, volume 2. Addison-Wesley, 1984. 4, 45

[41] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. 4, 45

[42] Vin de Silva, Robert Ghrist, and Abubakr Muhammad. Blind swarms for coverage in 2-d. In *Proceedings of Robotics: Science and Systems*, pages 335–342, Cambridge, MA, June 2005. 9, 51, 52, 57, 104

[43] Christian Bettstetter and Johannes Zangl. How to achieve a connected ad hoc network with homogeneous range assignment: an analytical study with consideration of border effects. In *Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network*, pages 125–129, Stockholm, Sweden, September 2002. 12, 59, 82

[44] Yigal Bejerano. Simple and efficient k-coverage verification without location information. In *IEEE INFOCOM*, pages 897–905, Phoenix, Arizona, USA, April 2008. 23, 41, 43, 104

[45] Dezun Dong, Yunhao Liu, Kebin Liu, and Xiangke Liao. Distributed coverage in wireless ad hoc and sensor networks by topological graph approaches. In *IEEE ICDCS*, pages 106–115, Genova, Italy, June 2010. 23, 42, 104

[46] Dezun Dong, Yunhao Liu, and Xiangke Liao. Fine-grained boundary recognition in wireless ad hoc and sensor networks by topological methods. In *ACM MobiHoc*, pages 135–144, New Orleans, Louisiana, USA, May 2009. 25, 42, 43, 118, 129

[47] Bin Tong and Wallapak Tavanapong. On discovering sensing coverage holes in large-scale sensor networks. Technical Report TR 06-03, Computer Science, Iowa State University, March 2006. 32, 35, 40, 43, 113, 130

[48] Feng Yan, Philippe Martins, and Laurent Decreusefond. Accuracy of homology based approaches for coverage hole detection in wireless sensor networks. In *IEEE ICC*, pages 497–502, Ottawa, Canada, June 2012. 36

[49] Feng Yan, Philippe Martins, and Laurent Decreusefond. Accuracy of homology based coverage hole detection for wireless sensor networks on sphere. Submitted to IEEE Transaction on Wireless Communications, 2013. URL http://arxiv.org/abs/1303.0001. 36

[50] Feng Yan, Philippe Martins, and Laurent Decreusefond. Connectivity-based distributed coverage hole detection in wireless sensor networks. In *IEEE GLOBECOM*, pages 1–6, Houston, TX, USA, December 2011. 36

[51] Feng Yan, Anaïs Vergne, Philippe Martins, and Laurent Decreusefond. Homology-based distributed coverage hole detection in wireless sensor networks. Submitted to IEEE/ACM Transactions on Networking, 2013. URL http://arxiv.org/abs/1302.7289. 37, 136

[52] Qing Fang, Jie Gao, and Leonidas. J. Guibas. Locating and bypassing routing holes in sensor networks. In *IEEE INFOCOM*, pages 2458–2468, Hong Kong, China, March 2004. 40, 43

[53] Amitabha Ghosh. Estimating coverage holes and enhancing coverage in mixed sensor networks. In *29th Annual IEEE International Conference on Local Computer Networks*, pages 68–76, Tampa, FL, USA, November 2004. 40

[54] Bogdan Cărbunar, Ananth Grama, Jan Vitek, and Octavian Cărbunar. Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor Networks*, 2(1):94–128, Feburary 2006. 40

[55] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, and Sung Nok Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. John Wiley, 2nd edition, 2000. 40

[56] Chi Zhang, Yanchao Zhang, and Yuguang Fang. Localized algorithms for coverage boundary detection in wireless sensor networks. *Wireless networks*, 15(1):3–20, January 2009. 40, 41

[57] Chi Zhang, Yanchao Zhang, and Yuguang Fang. A coverage inference protocol for wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(6): 850–864, June 2010. 40, 41, 43

[58] Chifu Huang and YuChee Tseng. The coverage problem in a wireless sensor network. In *Proceedings of the 2nd ACM International Workshop on Wireless*

Sensor Networks and Applications (WSNA'03), pages 115–121, San Diego, California, USA, September 2003. 40, 43

[59] Xiaorui Wang, Guoliang Xing, Yuanfang Zhang, Chenyang Lu, Robert Pless, and Christopher Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *ACM SenSys'03*, pages 28–39, Los Angeles, CA, USA, November 2003. 40

[60] Honghai Zhang and Jennifer C. Hou. Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc & Sensor Wireless Networks*, 1: 89–124, March 2005. 40

[61] Mohamed K. Watfa and Sesh Commuri. Energy-efficient approaches to coverage holes detection in wireless sensor networks. In *Proceedings of the 2006 IEEE International Symposium on Intelligent Control*, pages 131–136, Munich, Germany, October 2006. 40

[62] Chi Zhang, Yanchao Zhang, and Yuguang Fang. Detecting coverage boundary nodes in wireless sensor networks. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pages 868–873, Ft. Lauderdale, Florida, USA, April 2006. 40, 41, 43

[63] Chi Zhang, Yanchao Zhang, and Yuguang Fang. Localized coverage boundary detection for wireless sensor networks. In *Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks (QShine'06)*, Waterloo, ON, Canada, August 2006. 40, 41

[64] Yigal Bejerano. Coverage verification without location information. *IEEE Transactions on Mobile Computing*, 11(4):631–643, April 2012. 41

[65] Gaurav S. Kasbekar, Yigal Bejerano, and Saswati Sarkar. Generic coverage verification without location information using dimension reduction. *IEEE/ACM Transactions on Networking*, 20(6):1991–2004, December 2012. 41

[66] Xiaoyun Li, David K. Hunter, and Kun Yang. Distributed coordinate-free hole detection and recovery. In *IEEE GLOBECOM*, pages 1–5, San Francisco, CA, USA, November 2006. 41, 43

[67] Xiaoyun Li and David K.Hunter. Distributed coordinate-free algorithm for full sensing coverage. *International Journal of Sensor Networks*, 5(3):153–163, June 2009. 41

[68] Stefan Funke. Topological hole detection in wireless sensor networks and its applications. In *joint workshop on Foundations of mobile computing*, pages 44–53, Cologne, Germany, September 2005. 41, 42

[69] Stefan Funke and Christian Klein. Hole detection or: "how much geometry hides in connectivity?". In *Proceedings of the 22nd annual symposium on computational geometry*, pages 377–385, Sedona, Arizona, USA, June 2006. 41, 42

[70] Alexander Kröller, Sándor P. Fekete, Dennis Pfisterer, and Stefan Fischer. Deterministic boundary recognition and topology extraction for large sensor networks. In *Proceedings of the 17th annual ACM-SIAM symposium on Discrete algorithm (SODA'06)*, pages 1000–1009, Miami, Florida, USA, January 2006. 42

[71] Hongyu Zhou, Su Xia, Miao Jin, and Hongyi Wu. Localized algorithm for precise boundary detection in 3d wireless networks. In *IEEE ICDCS*, pages 744–753, Genova, Italy, June 2010. 42

[72] Thanh Le Dinh. Topological boundary detection in wireless sensor networks. *Journal of information processing systems*, 5(3):145–150, September 2009. 42

[73] Wei-Cheng Chu and Kuo-Feng Ssu. Decentralized boundary detection without location information in wireless sensor networks. In *IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1720–1724, Paris, France, April 2012. 42

[74] Olga Saukh, Robert Sauter, Matthias Gauger, and Pedro José Marrón. On boundary recognition without location information in wireless sensor networks. *ACM Transactions on Sensor Networks*, 6(3):20:1–20:35, June 2010. 42

[75] Dennis Schieferdecker, Markus Völker, and Dorothea Wagner. Efficient algorithms for distributed detection of holes and boundaries in wireless networks. In *Proceedings of the 10th international conference on Experimental algorithms*, pages 388–399, Crete, Greece, May 2011. 42

[76] Warren S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, December 1952. 42

[77] Sándor P. Fekete, Alexander Kröller, Dennis Pfisterer, and Stefan Fischer. Neighborhood-based topology recognition in sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks*, volume 3121 of *Lecture Notes in Computer Science*, pages 123–136. Springer, 2004. 42

[78] Sándor P. Fekete, Michael Kaufmann, Alexander Kröller, and Katherina Lehmann. A new approach for boundary recognition in geometric sensor networks. In *Proceedings of the 17th Canadian Conference on Computational Geometry*, pages 84–87, Windsor, ON, Canada, August 2005. 42

[79] Xu Li, Shibo He, Jiming Chen, Xiaohui Liang, Rongxing Lu, and Sherman Shen. Coordinate-free distributed algorithm for boundary detection in wireless sensor networks. In *IEEE GLOBECOM*, pages 1–5, Houston, TX, USA, December 2011. 42

[80] Abubakr Muhammad and Magnus Egerstedt. Control using higher order laplacians in network topologies. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 1024–1038, Kyoto, Japan, July 2006. 49, 52

[81] Vin De Silva and Robert Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *International Journal of Robotics Research*, 25(12):1205–1222, December 2006. 52

[82] Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7:339–358, April 2007. 52, 104

[83] Vin de Silva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54(1):10–17, January 2007. 52, 53

[84] Abubakr Muhammad and Ali Jadbabaie. Decentralized computation of homology groups in networks by gossip. In *Proceedings of American Control Conference*, pages 3438–3443, New York, NY, USA, July 2007. 52, 53

[85] Alireza Tahbaz-Salehi and Ali Jadbabaie. Distributed coverage verification in sensor networks without location information. In *Proceedings of the 47th IEEE Conference on Decision and Control*, pages 4170–4176, Cancun, Mexico, December 2008. 52

[86] Alireza Tahbaz-Salehi and Ali Jadbabaie. Distributed coverage verification in sensor networks without location information. *IEEE Transactions on Automatic Control*, 55(8):1837–1849, August 2010. 52, 53, 58, 104

[87] P. Dlotko, R. Ghrist, M. Juda, and M. Mrozek. Distributed computation of coverage in sensor networks by homological methods. *Applicable Algebra in Engineering, Communication and Computing*, 23(1-2):29–58, April 2012. 52

[88] Jinko Kanno, Jack G. Buchart, Rastko R. Selmic, and Vir Phoha. Detecting coverage holes in wireless sensor networks. In *Proceedings of the 17th Mediterranean Conference on Control and Automation*, pages 452–457, Thessaloniki, Greece, June 2009. 53

[89] Harish Chintakunta and Hamid Krim. Divide and conquer: Localizing coverage holes in sensor networks. In *IEEE SECON*, pages 359–366, Boston, Massachusetts, USA, June 2010. 53

[90] Abubakr Muhammad and Magnus Egerstedt. Connectivity graphs as models of local interactions. In *Proceedings of the 43rd IEEE Conference on Decision and Control*, volume 1, pages 124–129, Atlantis, Paradise Island, Bahamas, December 2004. 53

[91] Abubakr Muhammad and Magnus Egerstedt. Connectivity graphs as models of local interactions. *Journal of Applied Mathematics and Computation*, 168 (1):243–269, September 2005. 53

[92] Benyuan Liu and Don Towsley. A study of the coverage of large-scale sensor networks. In *IEEE MASS*, pages 475–483, Fort Lauderdale, Florida, USA, October 2004. 55

[93] Loukas Lazos and Radha Poovendran. Stochastic coverage in heterogeneous sensor networks. *ACM Transactions on Sensor Networks*, 2(3):325–358, August 2006. 55

[94] Peng-Jun Wan and Chih-Wei Yi. Coverage by randomly deployed wireless sensor networks. *IEEE Transactions on Information Theory*, 52(6):2658–2669, June 2006. 55

[95] Xiaoyun Li, David K. Hunter, and Sergei Zuyev. Coverage properties of the target area in wireless sensor networks. *IEEE Transactions on Information Theory*, 58(1):430–437, January 2012. 55, 60

[96] Ming-Chen Zhao, Jiayin Lei, Min-You Wu, Yunhuai Liu, and Wei Shu. Surface coverage in wireless sensor networks. In *IEEE INFOCOM*, pages 109–117, Rio de Janeiro, Brazil, April 2009. 75

[97] Liang Liu and Huadong Ma. On coverage of wireless sensor networks for rolling terrains. *IEEE Transaction on Parallel and Distributed Systems*, 23 (1):118–125, January 2012. 75

[98] Gaurav S. Kasbekar, Yigal Bejerano, and Saswati Sarkar. Lifetime and coverage guarantees through distributed coordinate-free sensor activation. In *ACM MobiCom*, pages 169–180, Beijing, China, September 2009. 104

[99] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *ACM MobiHoc*, pages 80–91, Lausanne, Switzerland, June 2002. 113

[100] Anaïs Vergne, Laurent Decreusefond, and Philippe Martins. Reduction algorithm for simplicial complexes. In *IEEE INFOCOM*, Turin, Italy, April 2013. URL `http://hal.archives-ouvertes.fr/hal-00688919`. 135, 136

Homologie Simpliciale et Couverture Radio dans un Réseau de Capteurs

Feng YAN

RESUME : La théorie de l'homologie fournit des solutions nouvelles et efficaces pour régler le problème de trou de couverture dans les réseaux de capteurs sans fil. Ils sont basés sur deux objets combinatoires nommés complexe de Čech et complexe de Rips. Le complexe de Čech peut détecter l'intégralité des trous de couverture, mais il est très difficile à construire. Le complexe de Rips est facile à construire, mais il est imprécis dans certaines situations.

Dans la première partie de cette thèse, nous choisissons la proportion de la surface de trous manqués par le complexe de Rips comme une mesure d'évaluer l'exactitude de la détection de trou de couverture basée sur l'homologie. Des expressions fermées pour les bornes inférieures et supérieures de la proportion sont dérivés. Les résultats de simulation sont bien compatibles avec les bornes inférieure et supérieure d'analyse, avec des différences maximales de 0.5% et 3%. En outre, nous étendons l'analyse au cas de la sphère.

Dans la deuxième partie, nous proposons d'abord un algorithme distribué basé sur les graphes pour détecter les trous non triangulaires. Cet algorithme présente une grande complexité. Nous proposons donc un autre algorithme distribué plus efficace basé sur l'homologie. Cet algorithme ne nécessite que des informations de 1- et 2-saut nœuds voisins et a la complexité $O(n^3)$ où n est le nombre maximum de nœuds voisins à 1 saut. Il peut détecter avec précision les cycles frontières d'environ 99% des trous de couverture dans environ 99% des cas.

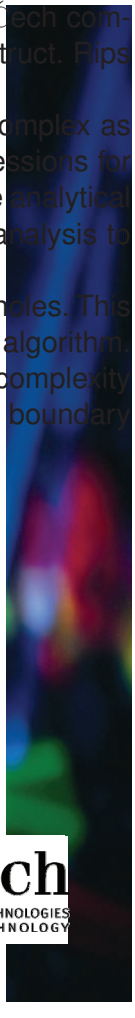
MOTS-CLEFS : Homologie simpliciale, trou de couverture, réseaux de capteurs sans fil

ABSTRACT : Homology theory provides new and powerful solutions to address the coverage hole problem in wireless sensor networks (WSNs). They are based on two combinatorial objects named Čech complex and Rips complex. Čech complex can detect all coverage holes, but it is very difficult to construct. Rips complex is easy to construct but it may be not accurate in some situations.

In the first part of this thesis, we choose the proportion of the area of holes missed by Rips complex as a metric to evaluate the accuracy of homology based coverage hole detection. Closed form expressions for lower and upper bounds of the proportion are derived. Simulation results are well consistent with the analytical lower and upper bounds, with maximum differences of 0.5% and 3%. In addition, we extend the analysis to the sphere case.

In the second part, we first propose a graph based distributed algorithm to detect non-triangular holes. This algorithm exhibits high complexity. We thus propose another efficient homology based distributed algorithm. This algorithm only requires 1- and 2-hop neighbour nodes information and has the worst case complexity $O(n^3)$ where n is the maximum number of 1-hop neighbour nodes. It can accurately detect the boundary cycles of about 99% coverage holes in about 99% cases.

KEY-WORDS : Homology theory, coverage hole, wireless sensor networks



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY