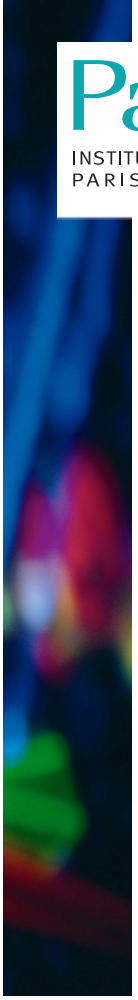
Simplicial homology : applied to wireless networks

Ngoc Khuyen Le



ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

TELECOM ParisTech

2016-ENST-0039

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

Ngoc-Khuyen LE

le 24 juin 2016

Homologie simpliciale

appliquée aux réseaux sans fil

Directeur de thèse : **Professeur Philippe MARTINS GONÇALVES**

Jury

M. Loutfi NUAYMI, Maître de Conférences HDR, Télécom Bretagne	Rapporteur
Mme. Catherine GLOAGUEN, Chargé de Recherche HDR, Orange Labs	Rapporteur
M. Tijani CHAHED, Professeur, Telecom SudParis	Examinateur
M. Xavier LAGRANGE, Professeur, Telecom Bretagne	Examinateur
M. Bartlomiej BLASZCZYSZYN, Professeur, INRIA	Examinateur
M. Jean-Marc KÉLIF, Docteur, Orange Labs	Examinateur
Mme. Lina MROUEH, Docteur, ISEP Paris	Examinateur

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,
my dear wife, sister,
and my lovely young daughter.

Acknowledgements

This thesis would not have been possible without the essential and gracious support of many individuals.

Foremost, I would like to express my profound gratitude to my thesis advisor, Professor Philippe Martins, for his support, guidance and encouragement throughout my doctoral research. I thank him for the great effort he has put into training me in the scientific field.

Beside my thesis advisor, I also want to express my sincere gratitude to Professor Laurent Decreusefond, for his valuable suggestion and advice. I greatly appreciate the time he spent to explain me a lot about mathematics.

My sincere acknowledgements go to my thesis committee members: Professor Nuaymi Loutfi, Professor Gloaguen Catherine, Professor Chahed Tijani, Professor Lagrange Xavier, Professor Blaszczyszyn Bartlomiej, Doctor Kélif Jean-Marc, Doctor Mroueh Lina, for generously offering their time to carefully read the manuscript, and for their insightful comments, hard questions which incented me to widen my research from various perspectives.

I am also thankful to my colleagues Doctor Anaïs Vergne and Doctor Yan Feng for the interesting discussions that I have with them, and for all their help.

Last, but not least, I would like to express my special thanks to my beloved wife, Pharmacist Thi-Anh-Hoa Nguyen, for being beside me during all of my life, my career, and the time I write this thesis. Although the doctoral study is challenging, her unbounded love and support are always my inspiration and motivation to move forward. Many thanks go to my wonderful daughter, Ngoc-Anh-Thu-Léna Le, for always making me smile. I would also thank my grandparents, my parents, my sister, my brother in-law, and two nephews, for their invaluable support despite of a long distance away from Vietnam.

Abstract

Simplicial homology is a useful tool to access important information about the topology of wireless networks such as: coverage and connectivity. In this thesis, we model the wireless network as a random deployment of cells. Firstly, we introduce an algorithm to construct the Čech complex, which describes exactly the topology of the network. Then, the Čech complex is used in further applications. The first application is to save transmission power for wireless networks. This application not only maximizes the coverage of the network but also minimizes its transmission power. At the same time, the coverage and the transmission power are optimized. The second application is to balance the traffic load in wireless networks. This application controls the transmission power of each cell in the network, always under the coverage constraint. With the controlled transmission power, the users are redirected to connect to the lower traffic loaded cells. Consequentially, the balanced traffic load is obtained for the network.

Résumé

Homologie simpliciale est un outil très efficace pour accéder à des informations importantes sur la topologie des réseaux sans fil, tels que: la couverture et la connectivité. Dans cette thèse, nous modélisons le réseau sans fil comme un déploiement aléatoire des cellules. Tout d'abord, nous introduisons un algorithme pour construire le complexe de Čech, qui décrit exactement la topologie du réseau. Ensuite, le complexe de Čech est utilisé dans des applications avancées. La première application est d'économiser l'énergie de transmission pour les réseaux sans fil. Cette application non seulement maximise la couverture de le réseau, mais réduit également la puissance de transmission. En même temps, la couverture et la puissance de transmission sont optimisées. La deuxième application est pour équilibrer la charge de trafic dans les réseaux sans fil. Cette application contrôle la puissance de transmission de chaque cellule dans le réseau, toujours sous contrainte de couverture. Avec la puissance d'émission contrôlée, les utilisateurs sont redirigés vers des cellules de charge plus faibles. Par conséquent, la charge du trafic est répartie entre les différentes cellules.

List of publications

- N. K. Le, P. Martins, L. Decreusefond and A. Vergne, "Construction of the Generalized Čech Complex," 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, 2015, pp. 1-5.
- N. K. Le, P. Martins, L. Decreusefond and A. Vergne, "Simplicial homology based energy saving algorithms for wireless networks," 2015 IEEE International Conference on Communication Workshop (ICCW), London, 2015, pp. 166-172.
- Chapter 5 subjects to a future publication.

Contents

List of Figures

List of Tables

Chapter 0

French detailed summary - Résumé long en français

0.1 Introduction

0.1.1 Motivations

Les communications sans fil ont réalisé des progrès permettant de proposer des applications de plus en plus puissantes et innovantes. Ces dernières demandent davantage de capacité et sont à l'origine d'une croissance notable de la charge du trafic mobile enregistrée ces dernières années. Selon une étude récente [4], la charge du trafic mobile continuera à augmenter de façon spectaculaire dans les années à venir. Cependant, la capacité des cellules ne peut pas être étendue à l'infini. La seule solution serait alors, de densifier les réseaux sans fil, notamment en divisant les macro-cellules existantes en un grand nombre de petites cellules. Dans ce contexte, la gestion des interférences devient un problème fondamental. Le chevauchement des zones de couverture entre les cellules dégrade fortement la qualité du service et provoque également une augmentation de la puissance consommée. Pour résoudre ce problème, la couverture du réseau doit donc être optimisée. Sachant que la prochaine génération de réseaux sans fil s'appuiera sur des réseaux ultra-denses et hétérogènes, la gestion topologique du réseau sera le facteur clé pour optimiser l'usage des ressources radio sous contraintes de consommation d'énergie et d'interférences.

Dans cette thèse, nous considérons le déploiement des réseaux sans fil comme étant aléatoire. Les stations de base peuvent être déployées partout et être de toute nature. La topologie du réseau peut être capturée par un objet mathématique appelé complexe de Čech [43]. Ce dernier rend compte exactement de la topologie d'un réseau [23]. Des informations sur la topolo-

gie, telles que la connectivité et la couverture du réseau, sont obtenues par des calculs matriciels simples.

Récemment, les complexes simpliciaux particuliers, appelés complexes de Rips, ont été utilisés avec succès pour détecter les trous de couverture dans les réseaux de capteurs sans fil [21, 56, 67]. Il existe également des applications développées basées sur l'homologie simpliciale pour les réseaux de communication sans fil. Dans [62], les auteurs ont introduit une méthode de réduction qui peut être utilisée pour désactiver les cellules afin d'économiser l'énergie. Dans [63], les auteurs ont développé une méthode basée sur l'homologie simpliciale pour planifier automatiquement la fréquence dans les réseaux sans fil. Et dans [64], un algorithme pour récupérer le réseau après un incident a été proposé. Les complexes simpliciaux sont donc utilisés de plus en plus dans les réseaux de communication sans fil.

Notre première motivation est d'utiliser les complexes de Čech pour optimiser la topologie du réseau. En particulier notre objectif est de réduire la puissance d'émission de chaque cellule tout en gardant la couverture maximale du réseau. Certaines cellules peuvent être désactivées, tandis que les cellules actives restantes possèdent la région de chevauchement minimale. En conséquence, une connection continue est fournie aux appareils mobiles. Dans le même temps, les interférences ainsi que les pertes d'énergie sont considérablement réduites.

La demande des utilisateurs n'est pas constante et est très variable. Par conséquent, la surcharge peut se produire n'importe où et n'importe quand. Un mécanisme de mise à jour de la taille cellule a été spécifié. Les cellules qui sont libres augmentent leur puissance de transmission afin d'élargir leur couverture. Cela donne la possibilité aux cellules surchargées de réduire leur couverture. Grâce à la procédure de transfert [1], certains utilisateurs seront redirigés des cellules surchargées vers les cellules libres. Par conséquent, la charge est automatiquement équilibrée.

0.1.2 Contributions

Comme nous l'avons déjà mentionné dans l'introduction, la topologie de réseau est le facteur clé pour concevoir et activer les fonctionnalités de la prochaine génération de réseaux sans fil. Le complexe de Čech est un outil utile pour capturer exactement la topologie du réseau. Cependant, le complexe de Čech pour les réseaux sans fil n'a pas été prospecté. C'est la raison pour laquelle nous commençons par la construction du complexe de Čech pour les réseaux sans fil. Tous les détails sur la construction du complexe de Čech sont présentés au Chapitre 3. La complexité de ces algorithmes est calculée. Nous avons montré que le complexe de Čech minimal qui capture

des informations sur la couverture et la connectivité du réseau est construit en temps polynomial. Nous proposons également une version parallèle et distribuée de cet algorithme.

Une fois le complexe de Čech construit, nous pouvons maintenant considérer ses applications. Le chapitre suivant, introduit la première application de ce complexe. Dans ce chapitre, nous considérons le déploiement aléatoire d'un réseau sans fil. Le rayon de chaque cellule peut être modifié en changeant sa puissance d'émission. Nous utilisons le complexe de Čech pour capturer la topologie du réseau. Ainsi, la couverture du réseau est traitable à partir de l'homologie du complexe de Čech. Nous optimisons la puissance d'émission de chaque cellule sous la contrainte de couverture, en conservant la couverture maximale du réseau. Le but de cette optimisation est de minimiser les interférences causées par une puissance d'émission non optimisée des cellules. Après l'optimisation, certaines cellules redondantes sont désactivées. La cellule restante, qui est active, possède la plus petite interférence avec ses cellules voisines. La perte de puissance, qui provoque principalement l'interférence, est réduite autant que possible. Le recuit simulé et la descente rapide sont des méthodes heuristiques. Alors que le recuit simulé trouve une approximation de l'optimum global, le déplacement en descente trouve seulement l'optimum local et converge plus rapidement. Nous proposons un algorithme de recuit simulé et un algorithme de descente rapide pour optimiser le réseau, puis nous comparons les résultats obtenus par ces algorithmes. Nous introduisons également la version parallèle et distribuée de l'algorithme de descente pour un calcul plus rapide. Les complexités de ces algorithmes sont également calculées et comparées.

Dans le chapitre suivant de la thèse, nous considérons l'équilibrage de charge dans les réseaux sans fil. Le trafic généré par chaque utilisateur n'est pas constant. En outre, les utilisateurs ne sont pas toujours sédentaires, ils peuvent se déplacer. Par conséquent, la charge de trafic de chaque cellule varie avec le temps. Parfois, certaines cellules peuvent être surchargées, tandis que d'autres sont encore libres. Il est nécessaire de rediriger certains utilisateurs des cellules surchargées vers les cellules libres. Nous proposons un algorithme pour équilibrer la charge de trafic pour les réseaux sans fil dans ce chapitre. Les cellules peuvent s'agrandir ou rétrécir toujours sous la contrainte de couverture : il n'y a pas de trous de couverture. Le appareil mobile choisit toujours la station de base dont le signal est le meilleur. Par conséquent, en modifiant la puissance d'émission des cellules, nous pouvons rediriger les utilisateurs des cellules surchargées suivant la procédure de transfert intercellulaire. Les utilisateurs sont toujours couverts avec suffisamment de signal reçu pour leur connexion. La puissance d'émission des cellules est également optimisée pour que l'interférence entre les cellules soit minimale.

Par la suite, la charge du trafic des cellules est répartie et les utilisateurs ont une meilleure condition à communiquer. La simulation pour évaluer la performance de cet algorithme d'équilibrage de charge est présentée dans la dernière partie de ce chapitre.

Le dernier chapitre du manuscrit récapitule les contributions majeures de la thèse. Des pistes de recherche futures sont également présentées.

0.2 Contexte mathématique

0.2.1 Complexe simplicial

Soit $S_k = \{v_0, v_1, \ldots, v_k\}$ un ensemble géométriquement indépendant de $k+1$ points dans $\mathbb{R}^n$, où $n > k$. L'enveloppe convexe de S_k est un k-simplexe, notée s_k. Le nombre k est sa dimension et $v_0, v_1, \ldots, v_k$ sont ses sommets : voir la Figure 1 pour les exemples.

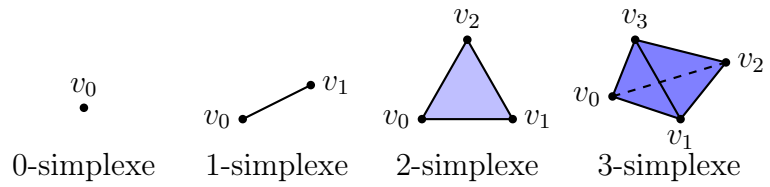


Figure 1: Un exemple de simplexes.

Tout sous-ensemble de S_k est également géométriquement indépendant, donc l'enveloppe convexe de chaque sous-ensemble de S_k est un simplexe d'une dimension inférieure. Soit l la dimension de ce simplexe, ce simplexe est une l-face de s_k.

Le complexe simplicial K dans $\mathbb{R}^n$ est une collection de simplexes dans $\mathbb{R}^n$ tel que : (1) chaque face d'un simplexe dans cette collection est aussi un simplexe inclus dans cette collection, et (2) l'intersection de deux simplexes quelconques dans cette collection est une face de chacun d'eux.

Un complexe simplicial abstrait est une collection K d'ensembles finis non vides, de sorte que si A est un élément dans K, il en est de même pour tout sous-ensemble non vide de A . Cette condition est semblable à la première condition du complexe simplicial. Le complexe simplicial abstrait est une description purement combinatoire du complexe simplicial géométrique, donc n'a pas besoin de la seconde condition sur la propriété d'intersection entre les simplexes.

Dans le reste de ce manuscrit, le mot « abstrait », l'adjectif dans la notion de « complexe simplicial abstrait » sera éliminé pour une meilleure lecture.

0.2.2 Groupe d'homologie

Soit K un complexe simplicial, nous présentons maintenant la façon dont son groupe d'homologie est obtenu. Premièrement, nous définissons l'orientation des simplexes. L'orientation d'un simplexe est déterminée par l'ordre de ses sommets. L'orientation d'un simplexe est inversée si l'ordre de ses sommets est transformé par une permutation impaire. Par exemple, si deux sommets d'un simplexe sont échangés, alors l'orientation de ce simplexe passe à l'opposé. Elle est désignée par le signe négatif comme :

$$[v_0, v_1, \ldots, v_i, \ldots, v_j, \ldots, v_k] = -[v_0, v_1, \ldots, v_j, \ldots, v_i, \ldots, v_k],$$

où $[v_o, v_1, \ldots, v_k]$ présente le simplexe $\{v_0, v_1, \ldots, v_k\}$ avec l'ordre de ses sommets.

Étant donné que tous les k-simplexes de K sont orientés, nous pouvons définir une k-chaîne simpliciale comme une combinaison linéaire finie de ces k-simplexes orientés. Pour chaque k, le groupe k-chaîne $C_k(K)$ est l'espace vectoriel formé par l'ensemble des k-simplexes orientés de K.

Nous définissons un opérateur bord ∂_k, qui agit sur les éléments de base $[v_0, v_1, \ldots, v_k]$ de C_k, par :

$$\partial_k[v_0, v_1, \ldots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k],$$

Nous vérifions notamment que $\partial_k \circ \partial_{k+1} = 0$ pour tous les k. Nous appelons une k-chaîne un k-cycle si son bord est nul. Le groupe de k-cycles, noté $Z_k(K)$, est le noyau de ∂_k. $B_k(K)$, le groupe des k-bords, est automatiquement l'image de ∂_{k+1}. Puisque $\partial_k \circ \partial_{k+1} = 0$ pour chaque k, il en découle que tout k-bord est aussi un k-cycle, il s'ensuit que $B_k(K) \subset Z_k(K)$ pour tout k. Deux k-cycles sont dits homologues si leur différence est dans le groupe des k-bords. On peut maintenant définir le k-ième groupe d'homologie de K comme étant l'espace vectoriel quotient du noyau de la différentielle par son image :

$$H_k(K) = Z_k(K)/B_k(K).$$

La dimension de k-ième groupe d'homologie est appelée le k-ième nombre de Betti :

$$\begin{aligned} \beta_k &= \dim H_k \\ &= \dim Z_k - \dim B_k \\ &= \dim \ker \partial_k - \dim \operatorname{im} \partial_{k+1}. \end{aligned} \tag{1}$$

Ce nombre est utilisé pour résoudre des problèmes de couverture. Étant donné un complexe simplicial, le k-ième nombre de Betti compte le nombre

de k-cycles indépendants, qui est trou k-ième dimensionnels dans ce complexe simplicial. Par exemple, le nombre de Betti à dimension 0 compte les composants connectés tandis que le nombre de Betti à dimension 1 compte les trous de couverture.

0.2.3 Complexe simplicial de réseaux sans fil

Définition 1 (Complexe de Čech) *Étant donnée une collection d'ensemble $\mathcal{U}$, le complexe de Čech, dénoté $\check{C}(\mathcal{U})$, est un complexe simplicial dont les k-simplexes correspondent à une intersection non vide de $k+1$ éléments distincts d'$\mathcal{U}$.*

Nous utilisons le complexe de Čech pour présenter les réseaux sans fil dans cette thèse, parce qu'il capture parfaitement la couverture d'un domaine [23].

Définition 2 (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 donne seulement une approximation de la couverture et peut manquer les trous de couverture [66]. Par conséquent, le complexe de Rips ne convient pas pour être une représentation de réseaux sans fil.

0.3 Construction du complexe de Čech des réseaux sans fil

Le complexe de Čech possède un avantage important car il représente exactement la topologie des domaines de recouvrement. Le complexe de Čech représente un groupe de cellules par un simplexe si toutes ces cellules ont une intersection non vide. Le complexe de Rips représente un groupe de cellules par un simplexe si tous les deux sont voisins. Le complexe de Rips détecte que la relation de voisinage entre les cellules, donc il représente parfois de façon inexacte la topologie du réseau, voir la Figure 2 pour un exemple. Dans cette figure, il y a trois cellules avec un trou de couverture à l'intérieur. Ce trou est représenté par un triangle vide dans la représentation par le complexe de Čech. Cependant, toutes les paires des cellules sont voisines, donc le complexe de Rips représente chaque paire des cellules par un triangle rempli. Cela signifie qu'il n'y a pas de trous de couverture dans la représentation de Rips. Le complexe de Čech localise avec succès le trou de couverture alors que le complexe de Rips ne le fait pas.

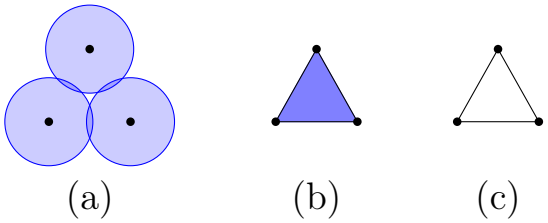


Figure 2: (a) cellules, (b) complexe de Rips, (c) complexe de Čech.

Récemment, un algorithme pour construire le complexe de Čech a été proposé [20] pour des applications dans la science graphique qui ne considèrent que les ensembles de couvertures de même taille. En conséquence, cet algorithme ne peut pas être appliqué pour calculer le complexe de Čech, qui présente la topologie du réseau sans fil, qui est composé de cellules de taille différente. La construction du complexe de Čech pour des réseaux sans fil est un problème important et intéressant mais il n'a pas été résolu.

Dans ce chapitre, nous proposons la construction du complexe de Čech d'un réseau sans fil. L'algorithme de construction est présenté en trois versions : centralisée, parallèle et distribuée. Nos algorithmes calculent le complexe de Čech minimal qui fournit des informations sur la connectivité et la couverture du réseau en temps polynomial.

0.3.1 Modèle du système

Nous considérons un réseau sans fil composé de N cellules distinctes. Nous supposons que chaque cellule utilise la propagation isotrope. La couverture de la i-ième cellule est modélisée comme :

$$c_i(v_i, r_i) = \{x \in \mathbb{R}^2 \ : \|x - v_i\| \leq r_i\},$$

où $\|.\|$ est la distance euclidienne, le sommet v_i présente l'emplacement de la station de base et r_i est le rayon de couverture de la i-ième cellule. Soit $\mathcal{U}$ la collection des cellules, alors $\mathcal{U} = \{c_i, i = 0, 1, \ldots, (N-1)\}$. Le complexe de Čech d'$\mathcal{U}$, $\check{\mathbf{C}}(\mathcal{U})$, est défini comme le complexe de Čech du réseau sans fil. Dans le complexe de Čech, chaque sommet v_i, (0-simplexe), correspond à la i-ième cellule c_i dans le réseau. Une arête, (1-simplexe), représente la connexion, ou l'intersection, entre deux cellules. Chaque k-simplexe, où $k \geq 2$, représente l'intersection commune de la couverture d'ensemble $(k+1)$ des cellules correspondantes de ce simplexe : voir à la Figure 3 pour un exemple.

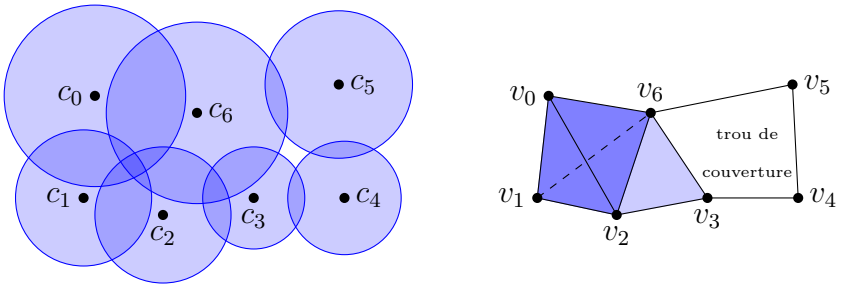


Figure 3: Cellules et sa représentation de Čech.

0.3.2 Construction du complexe de Čech

On désigne $\mathbf{S}_k$ la collection de tous les k-simplexes du complexe de Čech. L'algorithme dans la Figure 4 résume la construction du complexe de Čech. Cet algorithme cherche consécutivement des simplexes de plus en plus dimensionnels. Il commence par les simplexes de la plus basse dimension, et s'arrête aux simplexes de la plus haute dimension car il ne peut plus trouver un simplexe de plus grandes dimensions.

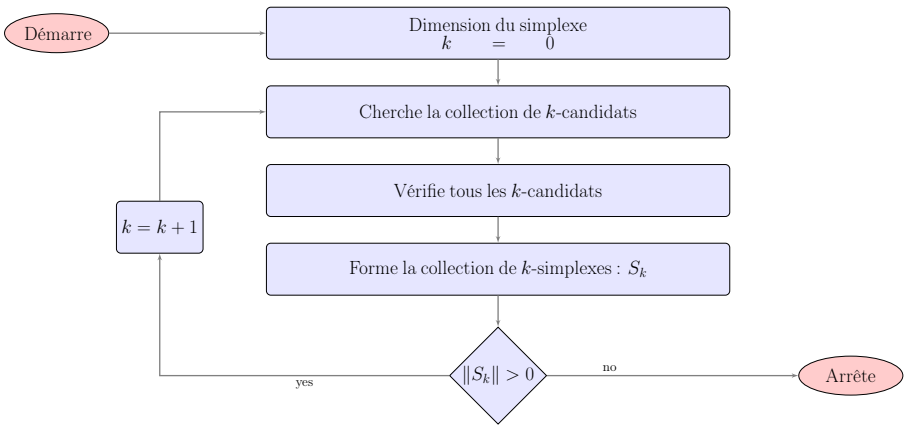


Figure 4: L'algorithme de la construction du complexe de Čech

La collection de 0-simplexes, $\mathbf{S}_0$, est alors évidemment la liste des sommets correspondants des cellules. Chaque 1-simplexe est une arête qui relie deux cellules se chevauchant. En d'autres termes, il s'agit d'une paire de deux cellules voisines.

Notons $\hat{s}_k = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k]$ une combinaison de $(k+1)$ sommet. Si $\hat{s}_k$ est un k-simplexe, alors chaque paire $[\hat{v}_i, \hat{v}_j]$ est un 1-simplexe, pour tous $0 \leq i < j \leq k$. Donc, s'il y a une paire $[\hat{v}_i, \hat{v}_j]$ qui n'est pas un 1-simplexe,

alors $\hat{s}_k$ n'est pas un k-simplexe. Si chaque paire $[\hat{v}_i, \hat{v}_j]$ est un 1-simplexe, pour tout $0 \leq i < j \leq k$, donc $\hat{s}_k = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k]$ possède une probabilité d'être un k-simplexe. Cette combinaison $\hat{s}_k$ est appelée un candidat à être un k-simplexe. Nous trouvons d'abord tous les candidats, et puis nous les vérifions.

Si $\hat{s}_k$ est un k-simplexe, il présente l'intersection commune de toutes les cellules correspondantes $\hat{c}_i(\hat{v}_i, \hat{r}_i)$, où $i = 0, 1, \ldots, k$. Nous désignons $\mathbf{I}$ comme étant cette intersection.

$$\mathbf{I} = \cap \hat{c}_i, \text{pour tous } i = 0, 1, \ldots, k;$$

Soit p un point qui appartient à $\mathbf{I}$, alors p doit appartenir à toutes les cellules correspondantes $\hat{c}_i$, où $i = 0, 1, \ldots, k$. On désigne la cercle $\hat{b}_i$ le frontière de la cellule $\hat{c}_i$, et $\mathbf{X}$ la collection des points d'intersection de chaque paire de ces cercles.

$$\mathbf{X} = \{\hat{b}_m \cap \hat{b}_n \mid 0 \leq m < n \leq k\}.$$

Il n'y a que deux cas possibles :

- Le premier cas : $\mathbf{X} \cap \mathbf{I} = \varnothing$. Il n'y a pas de point d'intersection qui appartienne à $\mathbf{I}$. Dans ce cas, le plus petit cercle, $\hat{b}_{\min} = \min\{\hat{b}_i \mid i = 0, 1, \ldots, k\}$, doit être à l'intérieur de tous les autres cercles $\hat{b}_i$ tel que $\hat{b}_i \neq \hat{b}_{\min}$ (Figure 5).

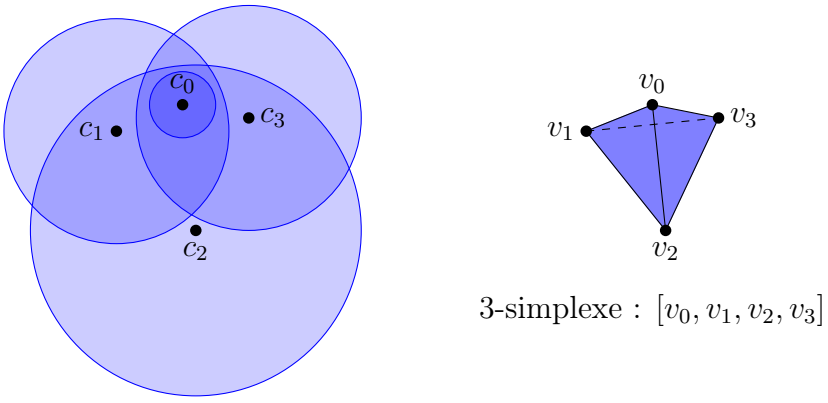


Figure 5: La plus petite cellule est à l'intérieur des autres cellules.

- Le second cas : $\mathbf{X} \cap \mathbf{I} \neq \varnothing$. II doit exister deux cercles $\hat{b}_m$ et $\hat{b}_n$ (la frontière de $\hat{c}_m$ et $\hat{c}_n$), dont au moins un point d'intersection appartient à toutes les autres cellules correspondantes $\hat{c}_i$, où $i \neq m$ and $i \neq n$ (Figure 6).

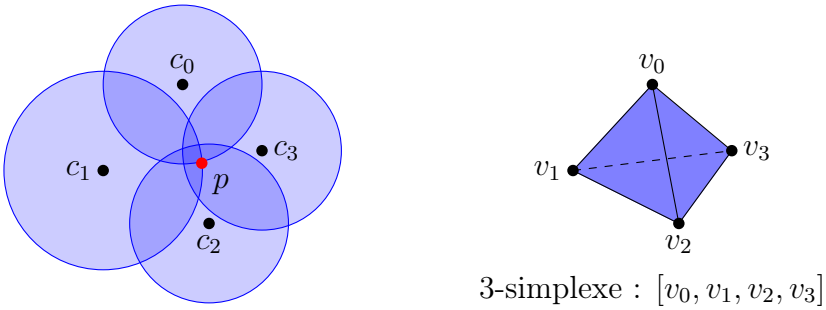


Figure 6: Il y a une paire de cellules dont un point d'intersection (le point rouge) est à l'intérieur tous les autres cellules.

Si ce candidat ne satisfait pas à ces deux cas ci-dessus, ce n'est pas un k-simplexe (Figure 7).

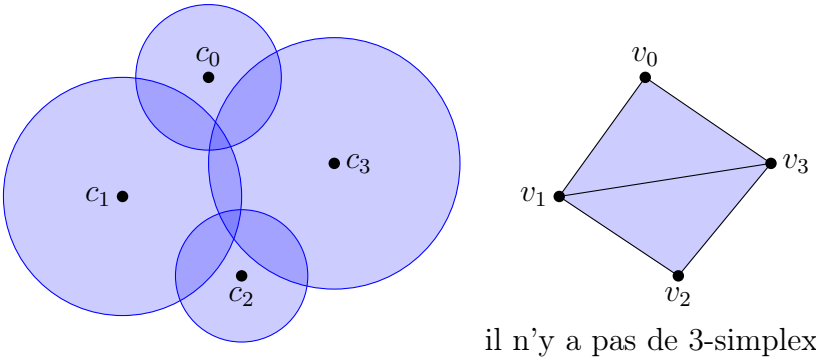


Figure 7: Il n'existe pas une paire de cellules dont un point d'intersection (le point rouge) est à l'intérieur tous les autres cellules.

0.3.3 Construction parallèle du complexe de Čech

Nous divisons le domaine du réseau en sous-domaines par des lignes de frontière verticales. Toutefois, la division par des lignes de frontière casse les connections entre les cellules qui sont sur des côtés différents de chaque ligne de frontière. Les connections entre des cellules voisins sont perdues si ces cellules se trouvent d'un et d'autre de cette ligne de frontière. Pour restaurer toutes les connections brisées qui sont perdues par la division de domaine, pour chaque cellule coupée par une ligne de frontière, nous ajoutons tous ses voisins au domaine correspondant qui contient cette cellule. Ensuite, nous calculons le complexe de Čech de chaque sous-domaine sur un ordinateur

séparé. Ces sous complexes sont intégrés pour réaliser le complexe intégral de l'ensemble du réseau.

0.3.4 Construction distribuée du complexe de Čech

Nous supposons que chaque cellule c_i peut communiquer avec d'autres cellules par la radio à une distance $d_i = 2r_i$. Nous supposons qu'il existe suffisamment de créneaux fréquentiels pour que les cellules communiquent par radio sans collision. Chaque cellule est également connectée par un réseau de backhaul. À l'état initial, chaque cellule diffuse un message ping avec sa position et son rayon sur le canal radio. Si une cellule reçoit un message ping, elle vérifie si la cellule qui a envoyé ce message ping est une voisine. Si elles sont voisines, alors la cellule qui a reçu le message ping envoie une confirmation de relation avec sa position et son rayon à la cellule qui a envoyé le message ping en utilisant le réseau de backhaul. Après avoir reçu la confirmation, la cellule qui a envoyé le message ping ajoute la cellule qui a envoyé la confirmation dans sa collection de voisines. Nous supposons que toutes les cellules peuvent répondre à la confirmation dans une période de t_{ack}. Étant donné qu'il n'y a pas de collision dans la transmission radio, cette période t_{ack} est la période d'une trame radio. Après cette période t_{ack}, chaque cellule détecte sa collection de voisines. Nous désignons la collection des voisines de la cellule c_i comme $\mathbf{N}_i$.

Chaque cellule calcule alors ses simplexes locaux en vérifiant l'intersection entre elle et ses voisines. Cependant, le voisinage est une relation à double sens. Par conséquent, la vérification de l'intersection pourrait être dupliquée par des cellules différentes qui sont des voisines. Pour éviter la duplication redondante, chaque cellule vérifie l'intersection en suivant une règle de main droite. Cette règle est que chaque cellule vérifie seulement l'intersection avec les voisines qui sont sur son côté droit. S'il y a une voisine qui possède la même coordonnée horizontale, la cellule vérifie seulement l'intersection avec cette voisine si la voisine possède une coordonnée verticale plus élevée. Après avoir calculé son complexe local, la cellule supprime tous les simplexes qui sont des faces de tout simplexe dans son complexe local. Pour chaque simplexe qui n'est pas une face d'un simplexe dans son complexe local, elle transmet ce simplexe à chaque cellule qui appartient à ce simplexe. En conséquence, chaque cellule détecte tous ses simplexes.

Par exemple, dans la Figure 3, la cellule c_2 vérifie l'intersection avec seulement la cellule c_3 et c_6. Il détecte le simplexe $[v_2; v_3; v_6]$. Il reçoit ses autres simplexes du voisin c_0 et c_1.

0.3.5 Complexité

La collection $\mathbf{S}_0$ de 0-simplexes est évidemment la collection de cellules, alors la complexité pour trouver $\mathbf{S}_0$ est $\mathcal{O}(N)$. La computation 1-simplexes est de trouver les cellules voisines, dont la complexité est C_N^2. Pour obtenir des informations sur la connectivité du réseau, nous devons calculer le numéro de Betti β_0. Donc, le complexe de Čech doit être construit jusqu'à la dimension 1. La complexité pour construire le complexe de Čech dans ce cas est $\mathcal{O}(N^2)$. Pour obtenir des informations sur la couverture du réseau, nous devons calculer le numéro Betti β_1. Donc, le complexe de Čech doit être construit jusqu'au moins la dimension 2. Soit n le nombre moyen de voisines de chaque cellule, qui est aussi le nombre moyen de ses 1-simplexes. Le nombre de candidats qui contiennent cette cellule est au maximum C_n^k en moyenne. Donc, il y a environ NC_n^k candidats qui peuvent être k-simplexes. Par conséquent, la complexité pour trouver tous les k-simplexes est typiquement NC_n^k. La complexité pour construire le complexe de Čech construit à dimension d est :

$$T(\check{\mathbf{C}}, d) = N + C_N^2 + N \sum_{k=2}^{d} C_n^k.$$

Avec $d = 2$, le complexe de Čech est le minimum qui donne des informations sur la couverture. Sa construction a une complexité $\mathcal{O}(N^2 + Nn^2)$. Cela signifie que le complexe de Čech minimal peut être calculé en temps polynomial. Si le complexe de Čech est calculé jusqu'à une dimension supérieure, en supposons que $d = \infty$, la somme $\sum_{k=2}^{d} C_n^k$ peut être bornée par 2^n. La complexité pour construire le complexe de Čech dans ce cas est alors maximum $\mathcal{O}(N^2 + N2^n)$. Bien que ce maximum contienne un terme exponentiel qui est 2^n, il ne devrait pas être un nombre significatif car le nombre moyen de voisines de chaque cellule n est normalement beaucoup plus petit que le nombre de cellules dans le réseau N.

Si le complexe de Čech est calculé parallèlement sur plusieurs ordinateurs, chaque sous-domaine possède N/m cellules en moyenne, où m est le nombre de CPUs. Si le complexe de Čech est calculé jusqu'à la dimension 2, la complexité du calcul sur chaque ordinateur est $\mathcal{O}(N^2/m^2 + Nn^2/m)$. Si le complexe de Čech est calculé jusqu'à sa dimension la plus élevée, la complexité du calcul sur chaque ordinateur est autant $\mathcal{O}(N^2/m^2 + N2^n/m)$.

Le calcul distribué du complexe de Čech se fait sur chaque cellule. Chaque cellule calcule le complexe de Čech pour sa région locale. Le nombre moyen de cellules dans cette région est $(n+1)$. Donc, la complexité du calcul distribué du complexe de Čech jusqu'à la dimension 2 et sa dimension la plus élevée sur chaque cellule sont $\mathcal{O}(n^3)$ et $\mathcal{O}(n2^n)$, respectivement.

0.3.6 Simulation et résultats

La Figure 8 illustre un complexe de Čech d'un déploiement de cellules selon un processus de Poisson de densité $\lambda = 5$ sur un espace [5x5]. Les 0-simplexes sont marqués comme des points noirs, les 1-simplexes sont dessinés comme des arêtes. Les 2-simplexes, 3-simplexes et 4-simplexes sont représentés par la couleur jaune, verte, et rouge, respectivement.

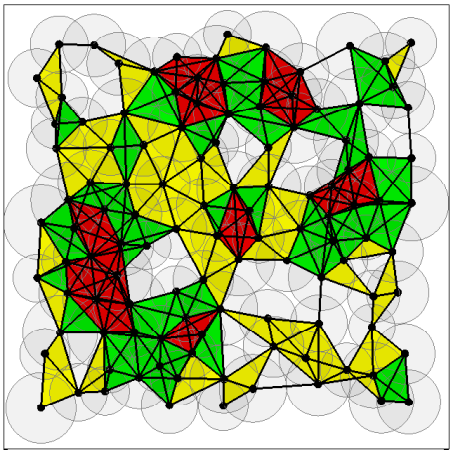


Figure 8: Un complexe de Čech du réseau sans fil.

Nous déployons aléatoirement des cellules selon un processus de Poisson sur un carré de taille [30, 5]. La densité de la cellule varie de 1 (moyen) à 2 (élevé). Le rayon de chaque cellule est une variable aléatoire de 0,5 à 1.

Dans chaque déploiement, le complexe de Čech du réseau est calculé par des calculs centralisés, parallèles et distribués. Tous ces calculs donnent la même valeur du complexe de Čech. Le calcul parallèle est effectué sur trois ordinateurs distincts. Le calcul distribué se fait sur chaque cellule et il comporte trois étapes : la détection des cellules voisines, le calcul des simplexes locaux et la transmission des simplexes. La période de détection des voisins dépend du protocole que nous utilisons pour transmettre le message par les canaux radio et par le réseau backhaul. Par exemple, LTE est un système radio dont une trame radio est de 10 ms [5]. Si nous utilisons le système LTE pour transmettre le message, la durée pour envoyer le message ping est une trame radio, qui est de seulement 10 ms. Le temps d'envoyer un message par réseau backhaul, qui est typiquement un réseau optique, est beaucoup plus petit et peut-être ignoré. Pour obtenir le complexe de Čech intégré, nous devons collecter tous les simplexes locaux qui sont calculés pour chaque cellule. Nous considérons le temps de construction du calcul distribué comme

la valeur maximale du temps de construction du calcul local à chaque cellule. Le temps de construction du complexe de Čech est noté dans le Tableau 1 :

Table 1: Temps de construction (seconde) du complexe de Čech

Densité de cellules	Calcul centralisé		Calcul parallèle		Calcul distribué	
	$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$
1,0	0,144	0,526	0,033	0,235	0,005	0,053
1,5	0,946	11,825	0,206	5,674	0,006	1,433
2,0	4,088	218,290	0,885	124,452	0,007	26,479

Le temps de construction augmente rapidement avec la densité des cellules. Le nombre de cellules N et le nombre moyen de voisins de chaque cellule n augmentent avec la densité des cellules. En outre, le temps de construction augmente de façon exponentielle avec n. Le temps de calcul de la construction parallèle et de la construction distribué est beaucoup plus petit que le calcul centralisé, en raison de sa plus faible complexité de calcul sur chaque ordinateur. Le calcul distribué a le plus petit temps de construction parce qu'il est distribué dans le calcul local à chaque cellule.

Le nombre de transmissions qui sont envoyées par chaque cellule, ainsi que la taille de chaque transmission sont importants pour évaluer les performances de la construction distribuée du complexe de Čech. Le message ping n'est envoyé qu'une seule fois par chaque cellule. Si une cellule reçoit un message ping et elle détecte que l'expéditeur est un voisin, il envoie un message d'accusé de réception. Ce message a une taille constante, il ne contient que son identifiant, sa position et son rayon de couverture. La taille de ce message est petite, donc nous considérons uniquement le nombre moyen de messages d'accusé de réception envoyés par une cellule. Après la construction du complexe local, chaque cellule envoie sa liste de simplexes suivant les règles de droite. Cette liste peut être longue et sa taille n'est pas constante. Nous enregistrons cette liste dans un fichier de texte, où l'identifiant de chaque cellule dans un simplexe est séparé par une espace et chaque simplexe dans la liste est séparé par une virgule. Chaque caractère dans ce texte a une taille qui est un octet : voir le Tableau 2 pour l'évaluation du nombre de messages de chaque cellule et la taille de chaque message (en octets).

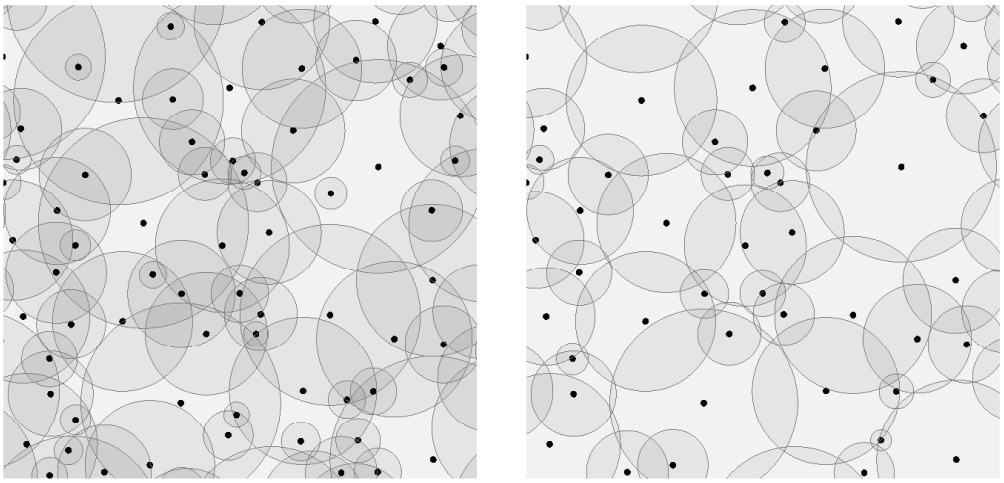
Table 2: Performance de la construction distribuée du complexe de Čech

Densité des cellules	Nombre de ACKs	Nombre de transmissions		Taille de tranmission (Octets)	
		$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$
1,0	5,52	3,05	3,03	23,59	20,14
1,5	8,32	4,59	4,55	38,82	48,61
2,0	11,01	6,07	6,10	45,43	170,01

Pour construire le complexe de Čech minimal, qui donne des informations sur la couverture et la connectivité du réseau, chaque cellule n'a besoin que de 6 transmissions à la densité la plus élevée. Si le complexe de Čech est construit jusqu'à sa plus haute dimension, le nombre de transmission est augmenté mais ce n'est pas un grand nombre.

0.4 Algorithmes basés sur l'homologie simpliciale pour économiser l'énergie dans les réseaux sans fil

Traditionnellement, les cellules sont déployées selon la configuration hexagonale. Comme cette configuration est optimisée pour des cellules de même taille, elle n'est pas toujours pratique. La prochaine génération de réseaux sans fil sera hétérogène, et comprendra plusieurs types de cellules telles que les macrocellules, les microcellules et les picocellules. Donc, dans ce chapitre, nous considérons le déploiement d'un réseau sans fil comme un déploiement aléatoire de cellules. Ensuite, nous maximisons la couverture, et en même temps minimisons les régions de chevauchement du réseau. Nous proposons deux algorithmes heuristiques : un recuit simulé et un mouvement descendant. L'algorithme de recuit simulé fournit l'approximation de la taille optimisée globale des cellules. L'algorithme de mouvement descendant donne toujours la taille optimisée locale des cellules mais il exécute plus vite. Cependant, nos simulations montrent que l'optimum local s'approche de l'optimum global. Nous proposons également des améliorations à ces algorithmes afin de réduire le temps d'exécution avec également la possibilité et de pouvoir effectuer les calculs sur des ordinateurs parallèles ou distribués. Un exemple de cellules avant et après l'optimisation est illustré à la Figure 9. L'analyse de la performance de ces algorithmes, ainsi que la densité de probabilité (pdf) des rayons optimisés des cellules sont également analysés et discutés.



(a) Avant l'optimisation. (b) Après l'optimisation.

Figure 9: Cellules avant et après l'optimisation.

0.4.1 Modèle du système

Nous considérons un réseau sans fil dont les cellules sont déployées aléatoirement sur le plan. Nous supposons que chaque cellule utilise la propagation isotrope, et que chaque cellule prend en charge les utilisateurs dans une limite d'un rayon de couverture à partir de sa station de base. Soient c_i la i-ième cellule du réseau, v_i sa position par rapport à station de base et r_i son rayon de couverture, et $p_{t,i}$ la puissance d'émission de la i-ième cellule, la relation entre sa puissance d'émission $p_{t,i}$ et son rayon de couverture r_i est simplifiée comme : $p_{t,i} = r_i^{\gamma}$, où γ est l'exposant d'affaiblissement. Soient $r_{i,\min}$ et $r_{i,\max}$ le rayon maximal et minimale d'émissionde la i-ième cellule c_i, respectivement, la puissance de transmission totale du système est : $P = \sum_{i=0}^{N-1} r_i^{\gamma}$; où N est le nombre de cellules dans le réseau, et $r_{i,\min} \leq r_i \leq r_{i,\max}$.

Nous utilisons le complexe de Čech pour représenter la topologie du réseau sans fil. Nous rappelons que le complexe de Čech représente exactement la topologie du réseau et que son k-ième nombre β_k de Betti compte les trous k-ièmes dimensionnels. Si toutes les cellules du réseau sont connectées, β_0 sera égal à 1. S'il n'y a pas de trou, β_1 sera nul.

Définition 3 (Indice d'un sommet) *Soit X un complexe simplicial, l'indice d'un sommet v dans X est le plus grand entier k tel que pour tout $i \leq k$ chaque $(i-1)$-simplexe dans X qui contient v est une face d'au moins i-simplexe dans X qui contient v.*

L'indice d'un sommet indique combien de fois la cellule correspondante chevauche ses voisines. L'indice 0 indique que la cellule correspondante est isolée, elle est connectée à aucune cellule. L'indice 1 indique que la cellule correspondante est connectée aux cellules voisines par des 1-simplexes. L'indice 2 indique que la cellule correspondante est reliée avec les voisines par des simplexes de dimension 2 ou supérieure. Voir la Figure 10 pour un exemple d'indices des vertices.

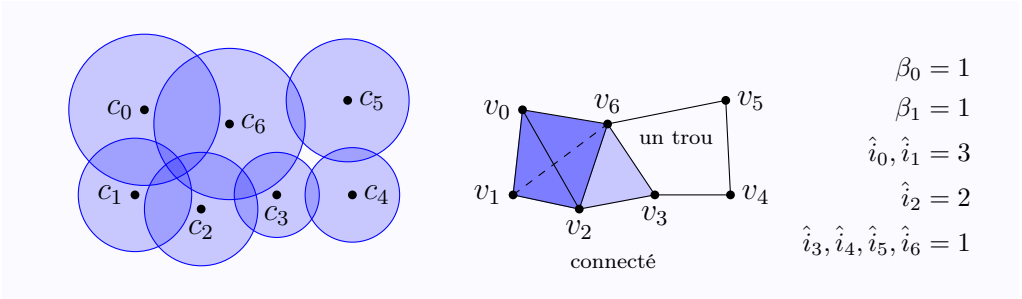


Figure 10: Exemple des indices des vertices dans un complexe de Čech.

0.4.2 Algorithmes pour économiser l'énergie

Étant donné un réseau sans fil, nous maximisons sa couverture, et en même temps minimisons sa puissance de transmission totale. Tout d'abord, nous assurons une couverture maximale pour le réseau. Chaque cellule est allumée et est réglée pour fonctionner avec la puissance d'émission la plus élevée. À cet état initial, le réseau possède la plus grande couverture. Cependant, un grand nombre de cellules se chevauchent. La zone de chevauchement entre les cellules provoque la perte de puissance d'émission due à une interférence. Nous pouvons alors optimiser la puissance d'émission en minimisant la zone de chevauchement. Cependant, la couverture globale du réseau devrait être conservée. En d'autres termes, les deux nombres de Betti β_0 et β_1 du complexe de Čech du réseau ne doivent pas être modifiés. Le problème d'optimisation peut alor s'écrire comme :

$$\begin{aligned} \min_{r} \quad & \sum_{i=0}^{N-1} r_i^{\gamma} \\ \text{s.t.} \quad & \beta_0 = \beta_0^* \\ & \beta_1 = \beta_1^* \\ & r = (r_0, r_1, \ldots, r_{N-1}), \end{aligned}$$

où β_0^* et β_1^* sont les nombres de Betti du complexe de Čech du réseau à l'état initial.

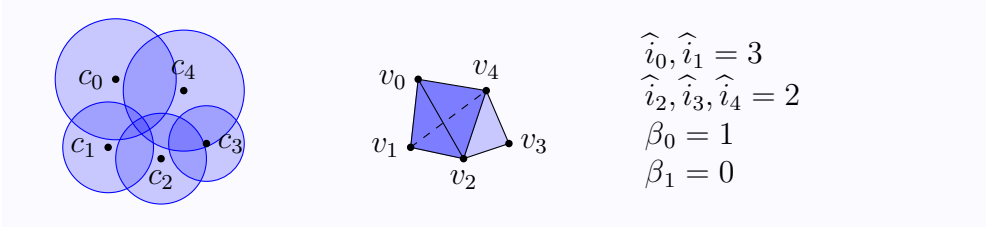
Dans le processus d'optimisation, une cellule peut être agrandie ou rétrécie. Nous supposons que tous les trous de couverture et les cellules à bord du réseau sont déjà connus. Nous ne modifions que le rayon de couverture des cellules internes qui ne sont pas à la frontière du réseau.

L'élargissement d'une cellule ne crée pas de trou de couverture. Cependant, une décroissance de cellule peut en créer un nouveau. Nous devons vérifier s'il y a un nouveau trou de couverture après une décroissance de la cellule. Pour ce faire, nous calculons à nouveau le complexe de Čech et ses deux nombres de Betti β_0 et β_1. Si ces nombres ne sont pas modifiés (identiques aux nombres β_0^* et β_1^*), il n'y a pas de nouveau trou de couverture, et la connectivité est aussi conservée, la décroissance est alors acceptée. Au contraire, si l'un des nombres Betti β_0 et β_1 est modifié, la décroissance est refusée.

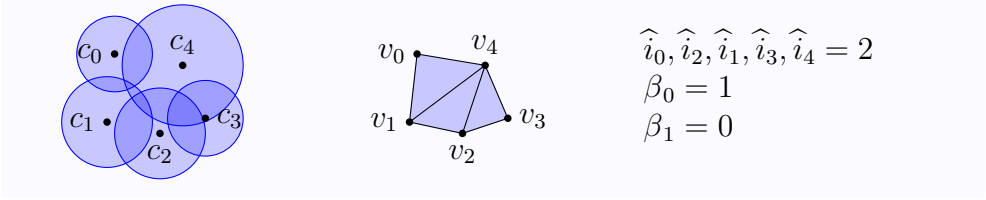
Un exemple de vérification de la couverture globale est illustré dans la Figure 11. Dans la Figure 11(a), toutes les cellules sont à leur stade initial. Chaque cellule possède sa couverture maximale. Ces cellules se chevauchent. Elles sont représentées par un tétraèdre et un triangle. Les nombres de Betti sont $\beta_0 = 1$ et $\beta_1 = 0$, qui indiquent que toutes les cellules sont jointes ensemble dans une composante et il n'y a aucun trou. Les cellule c_0 et c_1 ont l'indice 3. Dans la Figure 11(b), après la réduction du rayon de la cellule c_0, le complexe de Čech est maintenant composé de trois triangles remplis. Les nombres de Betti β_0 et β_1 ne sont pas modifiés mais les indices de la cellule c_0 et de la cellule c_1 sont maintenant réduits à 2. Cela signifie que le niveau de chevauchement est réduit et qu'aucun trou ne s'affiche. L'algorithme accepte la réduction de la cellule c_0. Mais, dans la Figure 11(c), la cellule c_4 essaie de réduire son rayon et crée un nouveau trou de couverture. Ce trou est représenté par un rectangle vide et le nombre de Betti β_1 a maintenant la valeur 1. Les indices des cellules c_0, c_1, c_2, c_4 sont tous réduit à 1 car ces cellules sont maintenant sur la bordure d'un trou de couverture, la réduction de la cellule c_4 est refusée. Son rayon est ramené à sa valeur précédente.

Algorithme de recuit simulé

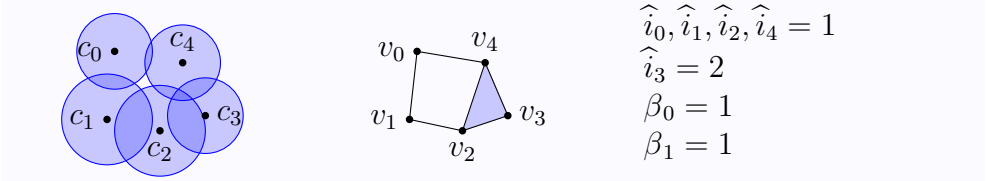
Le recuit simulé est une méthode heuristique pour trouver une approximation de l'optimum global. Etant donnée une solution initiale, où toutes les cellules possèdent leur rayon maximal, notre algorithme de recuit simulé trouve une approximation de la solution optimisée globale en suivant le programme de refroidissement : $T_k = T_0\alpha^k$. Dans ce programme, T_0 est la température initiale, et le facteur de refroidissement α est un nombre réel



(a) État initial, chaque cellule possède son rayon maximal de couverture.



(b) Première réduction, la cellule c_0 est acceptée pour réduire son rayon.



(c) Deuxième réduction, la cellule c_4 est refusée pour réduire son rayon.

Figure 11: Réduction du rayon de la cellule et la représentation de Čech.

positif tel que $0 < \alpha < 1$. À chaque température T_k, il choisit aléatoirement une cellule c et essaie d'augmenter ou de diminuer son rayon d'une quantité de Δr_c. Alors, la différence de consommation de l'énergie est calculée par $\Delta P = (r_c \pm \Delta r_c)^\gamma - r_c^\gamma$, où γ est l'exposant d'affaiblissement. Si le rayon est diminué, la transition est appelée un mouvement descendant. Ce mouvement descendant n'est accepté que si aucun trou est produit. Dans l'algorithme de recuit simulé, l'indice de cellule n'est pas considéré. L'algorithme ne vérifie que la couverture du réseau après une réduction du rayon de couverture d'une cellule. Donc, le complexe de Čech construit jusqu'à la dimension 2 est suffisant. Si le rayon est augmenté, la transition est appelée un mouvement montant. Dans ce cas, $\Delta P > 0$, ce mouvement montant est accepté avec probabilité $\exp(-\Delta P/T_k)$. Grâce à des mouvements montants, le processus peut sortir d'un minimal local pour s'approcher du minimum global. Ce processus est répété L fois à cette température T_k pour obtenir l'état d'équilibre thermodynamique. Le nombre L est appelé la longueur de la température

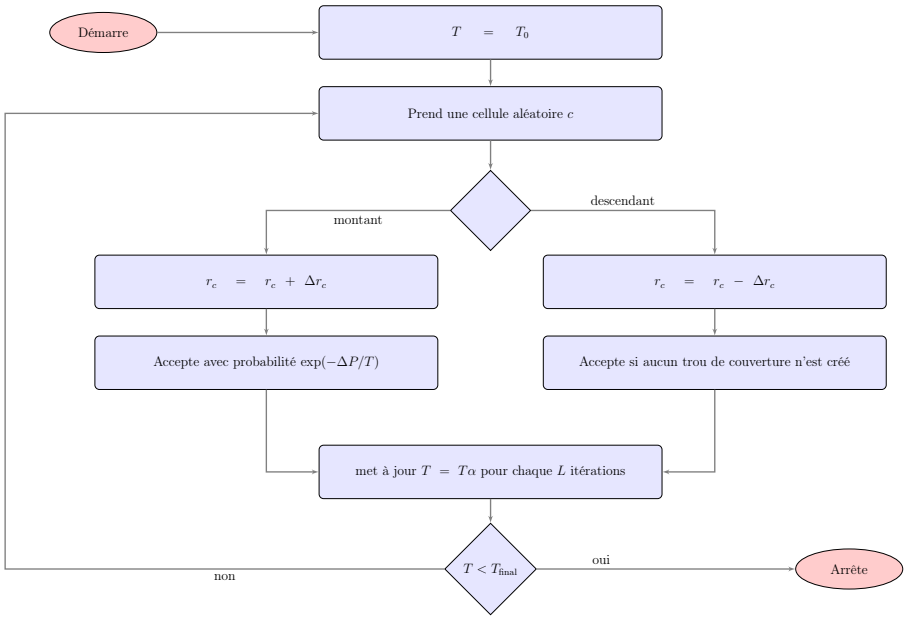


Figure 12: L'algorithme de recuit simulé

T_k. Après L répétitions, la température est progressivement diminuée par le programme de refroidissement $T_k = T_0\alpha^k$, pour $k = 0$ à K. La température initiale T_0 est choisie suffisamment grande pour que la probabilité d'un mouvement montant à l'état initial soit proche de 1. Le nombre d'étapes du programme de refroidissement K est choisi suffisamment grand pour que la probabilité d'accepter un mouvement montant soit proche de 0 à la température finale T_k. La configuration finale du rayon est alors l'approximation de la solution minimale globale. Voir la Figure 12 pour l'algorithme de recuit simulé.

Algorithme de mouvement descendant

Dans l'algorithme de descente (Figure 13), seules les décroissances des cellules sont acceptées, donc cet algorithme donne seulement la solution optimale locale. Cet algorithme concerne le niveau de chevauchement des cellules. Le niveau de chevauchement de chaque cellule correspond à l'indice du sommet qui représente cette cellule dans le complexe de Čech. Le processus de réduction commence par la cellule dont l'indice est le plus élevé. S'il y a plus d'une cellule dont les indices sont maximaux, la cellule plus grande est choisie. Après chaque réduction de rayon, la couverture du système est vérifiée. Si aucun nouveau trou n'est apparu, la réduction est acceptée, dans le cas contraire, la réduction est refusée. Cette cellule est marquée comme non

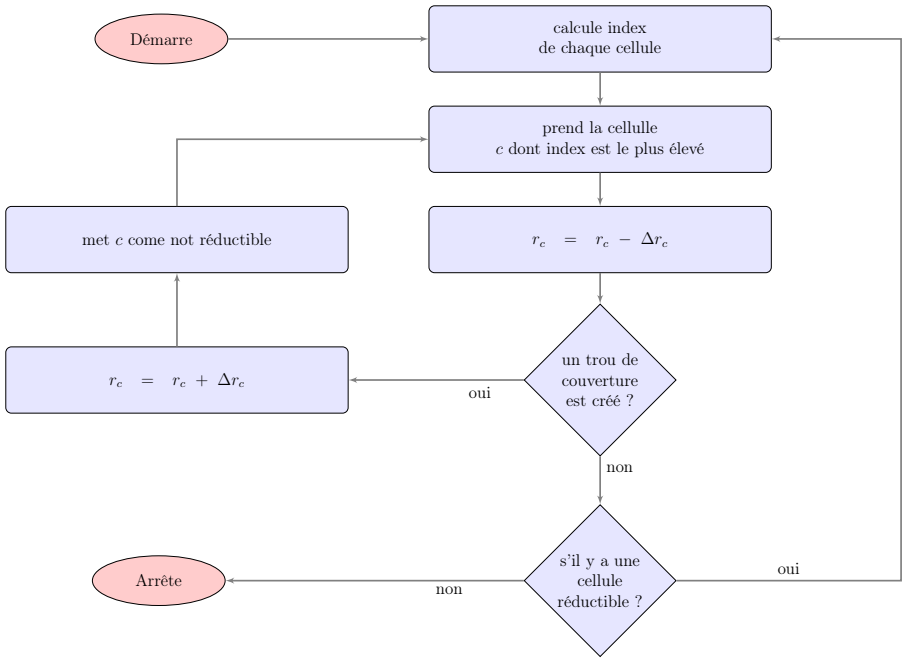


Figure 13: L'algorithme de mouvement descendant

réductible et son index est mis à -1 pour éviter une réduction répétée. Son rayon est également ramené à sa valeur précédente. Le processus de réduction se poursuit avec une autre cellule et se termine lorsque chaque cellule est irréductible.

0.4.3 Algorithmes améliorés pour économiser l'énergie

Au cours du processus d'optimisation de ces algorithmes, la topologie du réseau doit être mise à jour après chaque modification du rayon d'une cellule. Cependant, cette modification ne change que la topologie d'une région locale qui est composée de la cellule dont le rayon a changé et de cellules voisines, ce qui permet de ne mettre à jour que la topologie de la région locale de la cellule. Tout d'abord, nous introduisons l'algorithme « reconstruction rapide du complexe de Čech », qui recalcule le complexe de Čech complet du réseau après chaque modification de rayon d'une cellule dans le réseau. Ensuite, la vérification de la couverture du réseau peut être réduite à la vérification de la couverture de cette région locale. De plus, l'indice d'un sommet peut être calculé en utilisant la topologie de sa région locale. Par conséquent, nous présentons également l'algorithme « vérification rapide de la couverture du réseau » et « calcul rapide d'indice du sommet ».

De plus, deux cellules qui ne sont pas voisines n'ont pas de relation topologique. Nous pouvons optimiser les cellules qui ne sont pas voisines en même temps. L'optimisation peut être effectuée de manière parallèle et répartie pour obtenir des performances supérieures. Enfin, nous introduisons la version parallèle et la version distribuée des algorithmes d'économie d'énergie.

Algorithme parallèle pour économiser l'énergie

Dans l'algorithme de recuit simulé, chaque modification du rayon se fait en une seule itération. Cependant, chaque itération dans cet algorithme n'est pas indépendante de l'autre, car elle doit suivre la séquence des températures dans le processus de refroidissement. Comme chaque état de l'algorithme de recuit simulé contient la modification de l'état précédent, il est difficile de paralléliser le recuit simulé sans changer son processus séquentiel, ce qui peut provoquer à la solution non optimisée [33].

Nous développons l'algorithme de recuit simulé avec pour objectif de trouver la solution optimisée globale de la taille des cellules. Nous ne développons pas la version parallèle ou distribuée de l'algorithme de recuit simulé. L'algorithme de mouvement descendant possède des itérations indépendantes. Dans chaque itération, cet algorithme choisit une cellule dont l'indice est le plus élevé pour tenter de réduire son rayon. S'il y a plusieurs cellules qui ne sont pas voisines et possèdent l'indice le plus élevé, nous pouvons essayer de réduire parallèlement le rayon des cellules. Après un décroissement du rayon d'une cellule, on forme sa région locale, et puis on vérifie la couverture de cette région locale. S'il n'y a pas de trou de couverture, le décroisement est accepté. Dans le cas contraire, il est refusé.

Algorithme distribué pour économiser l'énergie

Nous supposons que chaque cellule peut se connecter à l'autre par un réseau backhaul. À la première étape, chaque cellule transmet sa position et son rayon de couverture aux autres. Par conséquent, chaque cellule connaît la position et le rayon de couverture des autres. Elle peut maintenance former sa région locale.

Un indice plus élevé d'une cellule indique que la région locale de la cellule est plus dense. Nous essayons de réduire le rayon d'une cellule c dont l'indice est plus élevé avec une probabilité plus élevée, de sorte que nous fixons la minuterie de la cellule c à une valeur aléatoire uniforme qui varie de 1 à $t_{\max}/\hat{i}_c$, où $\hat{i}_c$ est l'indice de la cellule c. Lorsque la minuterie d'une cellule est expirée, cette cellule essaie de réduire son rayon de couverture. Si deux cellules qui sont voisines essaient de réduire leur rayon en même temps, le trou

de couverture peut ne pas être détecté car l'information sur le rayon de l'autre n'est pas mise à jour. Par conséquent, avant d'essayer de réduire le rayon, chaque cellule envoie un message « pause » à ses voisines. Ensuite, la cellule réduit son rayon de couverture et vérifie la couverture de sa région locale. Si aucun trou n'est trouvé, la cellule confirme la réduction et envoie la nouvelle valeur de son rayon de couverture à ses voisines. Elle envoie également le message « continue » à ses voisines pour leur dire qu'ils peuvent continuer. Si une cellule a reçu un message « pause », elle interrompt son processus et attend que le message « continue » soit reçu. Ensuite, le processus peut se poursuivre normalement.

0.4.4 Complexité

Nos algorithmes nécessitent le calcul du complexe de Čech, ses nombres de Betti β_0 et β_1 et l'indice de chaque sommet. Soient N le nombre de cellules et n le nombre moyen de voisins de chaque cellule, la complexité pour calculer le complexe de Čech jusqu'à la dimension 2 et jusqu'à sa dimension la plus élevée est $\mathcal{O}(N^2 + Nn^2)$ et $\mathcal{O}(N^2 + N2^n)$, respectivement. Le nombre de k-simplexes dans ce complexe de Čech a une limite supérieure qui est NC_n^k. La complexité pour calculer β_0 et β_1 est alors $\mathcal{O}(N^3n^6)$. La complexité pour calculer l'indice de toute cellule est $\mathcal{O}(N^3n^3)$ si le complexe de Čech est construit pour dimension 2, et $\mathcal{O}(N^3 2^{2n})$ si le complexe de Čech est construit jusqu'à sa dimension la plus élevée. Dans l'algorithme de recuit simulé, le nombre d'itérations est fixé par le programme de température. Dans l'algorithme de mouvement descendant, le processus de réduction est répété jusqu'à ce qu'il n'y ait pas d'espace redondant. Après chaque réduction de rayon, l'espace réduit moyen d'une cellule est $\mathbb{E}\left[\pi r^2 - \pi(r - \Delta r)^2\right]$. Nous supposons que $\Delta r \ll r_{\max}$, donc l'espace réduit est approximé par $\pi r_{\max} \Delta r$. Le nombre de réductions dans l'algorithme de descente est $S_{\text{redondant}}/\Delta S = (N\pi r_{\max}^2 - S)/(\pi r_{\max}\Delta r)$ qui est l'ordre $\mathcal{O}(N)$. Les calculs avec des techniques accélérées sont basés sur la région locale. Le nombre moyen de cellules dans une région locale est n, ce qui est beaucoup moins que N, donc les calculs accélérés ont une complexité plus petite.

Le Tableau 3 note la complexité de nos algorithmes centralisés soit à la dimension 2, soit à sa dimension la plus élevée. Le Tableau 4 compare la complexité de l'algorithme centralisé, parallèle et distribué.

Table 3: La complexité des algorithmes centralisés.

Complexité	$d_{\max} = 2$	$d_{\max} = \infty$
algorithmes sans technique accélérée :		
complexe de Čech complet	$O(N^2 + Nn^2)$	$O(N^2 + N2^n)$
β_0, β_1 (verification de couverture)	$O(N^3n^6)$	$O(N^3n^6)$
indices de tout vertice	$O(N^3n^3)$	$O(N^32^{2n})$
algorithme de recuit simulé	$O(KLN^3n^6)$	-
algorithme de mouvement descendant	$O(N^4n^6)$	$O(N^42^{2n})$
algorithmes avec techniques accélérées :		
recalcul rapide de complexe de Čech	$\mathcal{O}(n^3)$	$\mathcal{O}(n2^n)$
verification rapide de couverture	$\mathcal{O}(n^9)$	$\mathcal{O}(n^9)$
calcul rapide d'indice d'un vertex	$\mathcal{O}(n^6)$	$\mathcal{O}(n^32^{2n})$
algorithme de recuit simulé	$\mathcal{O}(KLn^9)$	-
algorithme de mouvement descendant	$\mathcal{O}(Nn^9)$	$\mathcal{O}(Nn^32^{2n})$

Table 4: La complexité de l'algorithme centralisé, parallèle et distribué

Complexité	$d_{\max} = 2$	$d_{\max} = \infty$
algorithme de mouvement descendant		
sans technique accélérée	$O(N^4n^6)$	$O(N^42^{2n})$
avec techniques accélérées	$\mathcal{O}(Nn^9)$	$\mathcal{O}(Nn^32^{2n})$
algorithme parallèle	$\mathcal{O}(Nn^9/m)$	$\mathcal{O}(Nn^32^{2n}/m)$
algorithme distribué	$\mathcal{O}(n^9)$	$\mathcal{O}(n^32^{2n})$

0.4.5 Simulation et résultats

Nos algorithmes pour économiser l'énergie ont été évalués sur un espace [10 x 10] où les cellules ont été déployées aléatoirement selon un processus de Poisson. Chaque cellule a son rayon de couverture qui peut varier de $r_{\min} = 0,1$ à $r_{\max} = 1$. La densité des cellules est réglée à différentes valeurs de 0,2 à 1. L'exposant d'affaiblissement est $\gamma = 3$.

Les simulations d'algorithmes de recuit simulé ont été exécutées avec la température initiale $T_0 = 1,95$. Cette température initiale fournit une probabilité d'acceptation initiale d'un déplacement montant qui est $0,95$. A chaque température, le processus est répété $L = 1000$ fois. Ensuite, la température est réduite par un facteur de refroidissement réglé à $\alpha = 0,95$. Le nombre

d'étapes dans le programme de température est $K = 100$. Avec ce programe de température, la probabilité d'acceptation finale d'un déplacement montant est $0,05$. L'algorithme de mouvement descendant a été evalué avec le complexe de Čech construit jusqu'aux dimensions maximales 2 et 10. Le nombre d'itérations dans cette simulation est 1000.

L'algorithme parallèle n'est qu'une version accélérée de l'algorithme centralisé. Par conséquent, nous n'effectuons pas la simulation de cet algorithme. La simulation de l'algorithme distribué est réalisée avec le complexe de Čech construit jusqu'à la dimension 2.

Puissance consommée moyenne par cellule avec rayon optimisé

L'algorithme de recuit simulé, qui donne une approximation de la solution optimale globale, économise plus d'énergie, voir la Figure 14. À la densité la plus élevée de cellule, chaque cellule fonctionne avec 35% de sa puissance maximale en moyenne, économisant ainsi 65% d'énergie. L'algorithme de mouvement descendant économise 62% d'énergie avec le complexe de Čech construit à la dimension 2 et il économise 60% d'énergie avec le complexe de Čech construit à la dimension 10.

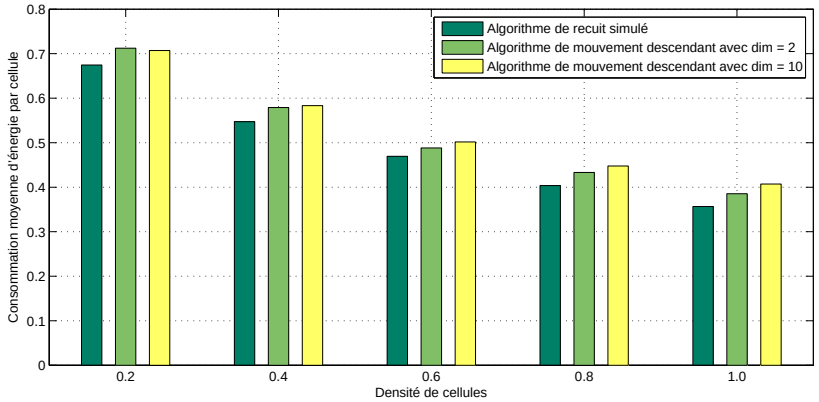


Figure 14: Consommation de l'énergie de la cellule optimisée par différents algorithmes centralisés.

L'algorithme de mouvement descendant avec le complexe de Čech construit jusqu'à la dimension 2 a une meilleure performance que celle avec le complexe de Čech construit à sa plus haute dimension. De plus, l'algorithme de mouvement descendant avec le complexe de Čech de dimension supérieure a la plus grande complexité. Par conséquent, l'algorithme de mouvement descendant avec le complexe de Čech construit jusqu'à la dimension 2 est recommandé pour un algorithme centralisé d'économie d'énergie. Parce que

l'algorithme de mouvement descendant avec le complexe de Čech construit pour dimension 2 a de meilleures performances que celui avec le complexe de Čech construit à la dimension supérieure, nous effectuons la simulation pour l'algorithme distribué avec le complexe de Čech construit seulement pour dimension 2. Il y a très peu de différence ($\leq 3\%$), voir la Figure 15.

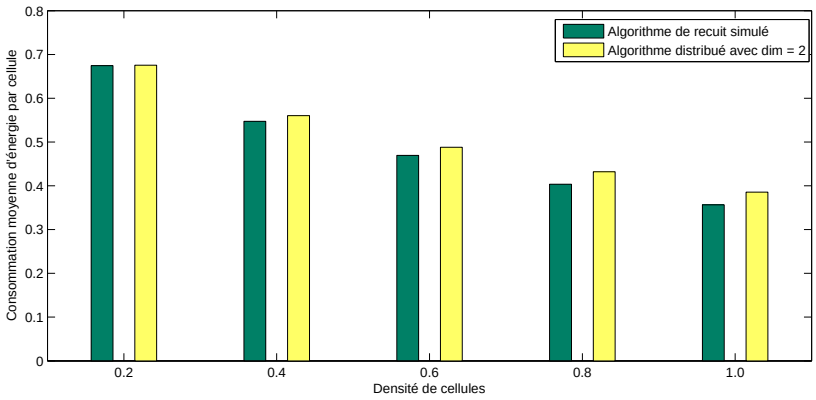


Figure 15: Consommation de l'énergie de la cellule optimisée par l'algorithme de recuit simulé et par l'algorithme distribué.

Densité de probabilité du rayon optimisé

Bien que l'énergie moyenne consommée par des cellules optimisées par les algorithmes différents soit pratiquement égale, le densité de probabilité du rayon optimisé obtenu par ces algorithmes est assez différent comme illustré dans les Figures 16, 17 et 18 pour trois densités de cellules : 1 (haute), 0,6 (moyenne) et 0,2 (faible). L'algorithme de recuit simulé donne une approximation de la solution optimale globale. Le densité de probabilité du rayon optimisé par cet algorithme montre que le déploiement optimal doit contenir des grandes cellules et des petites cellules. Les grandes cellules doivent assurer une grande couverture de fond pour le réseau, tandis que les petites cellules doivent assurer des environnements intérieurs. Cependant, l'algorithme de recuit simulé n'éteint pas beaucoup de cellules. Il conserve de nombreuses cellules de travail. Le nombre de cellules pouvant être désactivées en utilisant l'algorithme de recuit simulé est toujours inférieur à 10% pour toutes les valeurs de densité.

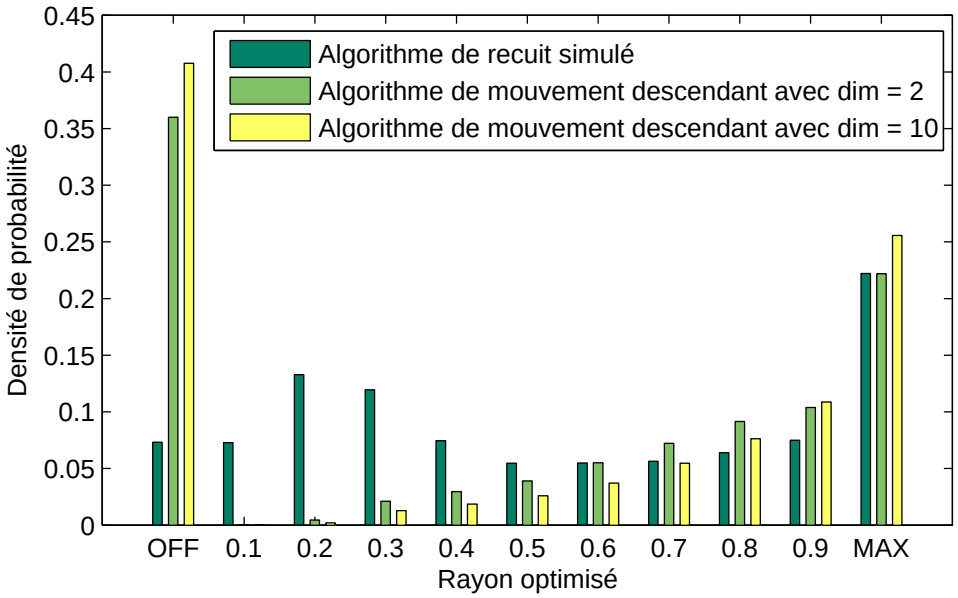


Figure 16: Densité de probabilité de rayon optimisé à la densité de cellule = 1,0 par algorithmes centralisés

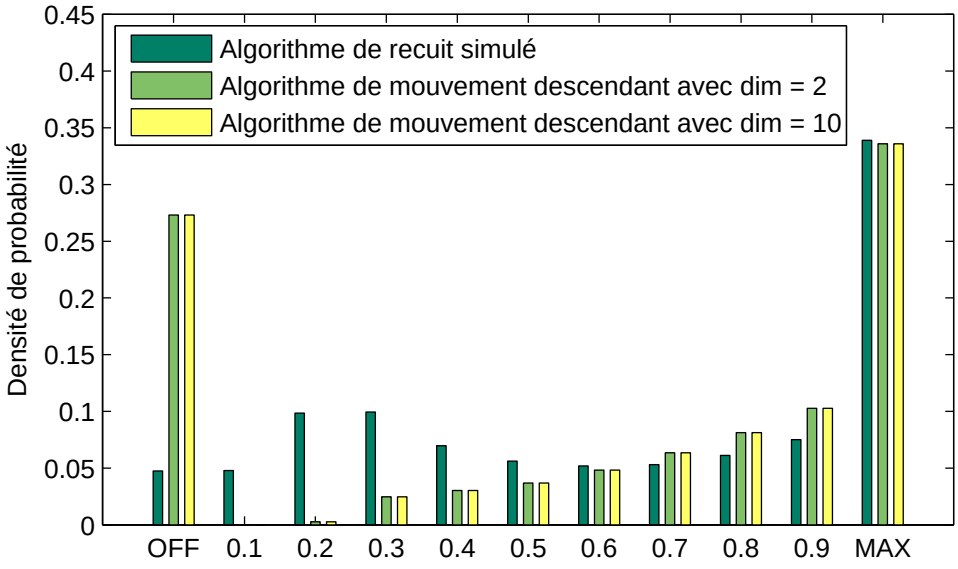


Figure 17: Densité de probabilité de rayon optimisé à la densité de cellule = 0,6 par algorithmes centralisés

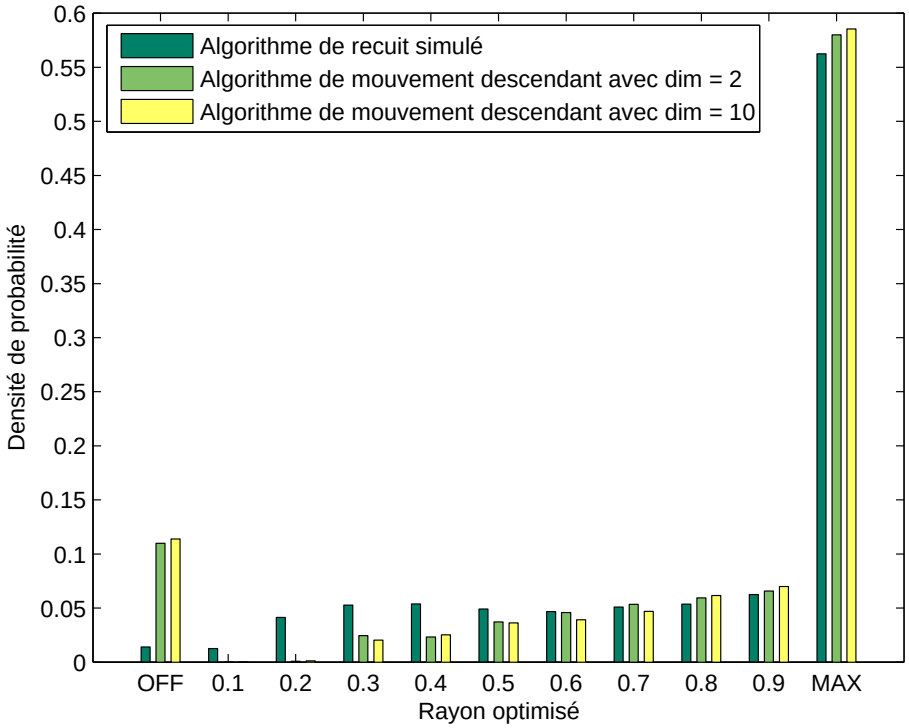


Figure 18: Densité de probabilité de rayon optimisé à la densité de cellules = 0,2 par algorithmes centralisés

Au contraire, l'algorithme de mouvement descendant désactive de nombreuses cellules. Le nombre de cellules désactivées obtenu en utilisant l'algorithme de mouvement descendant avec la dimension 2 est supérieur à 35% et avec la dimension 10 est supérieur à 40%. La différence d'énergie consommée entre les cellules optimisées par l'algorithme de recuit simulé et celle optimisée par l'algorithme de mouvement descendant est faible. L'algorithme de mouvement descendant peut avoir un meilleur compromis entre le coût d'énergie de transmission et le coût de déploiement matériel.

La Figure 19 illustre la densité de probabilité du rayon optimisé par l'algorithme distribué avec le complexe de Čech construit pour dimension 2 pour une densité différente de cellules. Il est similaire à celui optimisé par l'algorithme de mouvement descendant avec le complexe de Čech construit jusqu'à la dimension 2. Ce résultat montre que l'algorithme distribué et l'algorithme de mouvement descendant ont la même performance.

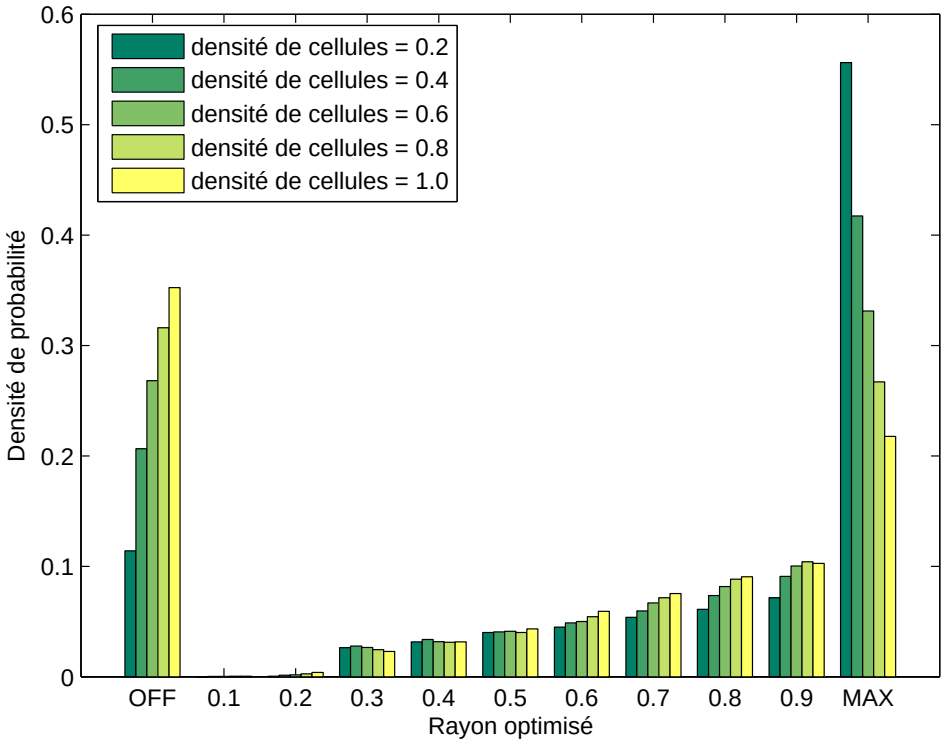


Figure 19: Densité de probabilité de rayon optimisé par algorithme distribué à densité différente de cellules

0.5 Algorithme distribué d'équilibrage de charge

Nous introduisons un algorithme distribué d'équilibrage de charge pour les réseaux sans fil, qui non seulement résout le problème de surcharge pour les stations de base, mais assure également une haute qualité de service pour chaque utilisateur. Notre algorithme modifie la puissance d'émission de chaque station de base afin de rediriger certains utilisateurs dans des cellules surchargées vers des cellules à faible charge. Ensuite, les charges de trafic des cellules sont équilibrées. Nous utilisons le complexe de Čech pour représenter la topologie du réseau. La topologie du réseau est toujours mise à jour. Il nous permet de toujours assurer la couverture du réseau ainsi que d'éviter l'interférence due à la puissance d'émission redondante. De plus, notre algorithme ne nécessite aucune information sur la position des utilisateurs.

0.5.1 Modèle du système

Nous considérons un réseau sans fil qui est très chargé. Nous supposons que la couverture du réseau est satisfaite avant d'effectuer l'algorithme d'équilibrage de charge. Il signifie qu'avant d'appliquer notre algorithme d'équilibrage de charge, il n'y a pas de trou de couverture dans le réseau. Nous considérons un réseau sans fil qui contient une collection de cellules :

$$\mathcal{C} = \{c_i(v_i, r_i) \mid i = 1, 2, \ldots, N\};$$

où N est le nombre de cellules, et $c_i(v_i, r_i)$ est la i-ième cellule dont la position de sa station de base est v_i et le rayon de couverture est r_i. Soient $p_{t,i}$ la puissance d'émission de la cellule c_i, u un utilisateur dans cette cellule, et la distance de l'utilisateur u à la station de base c_i $d_{u,i}$, l'affaiblissement de propagation de la station de base c_i à la position de l'utilisateur u est $L_i(d_{u,i})$. Pour estimer l'affaiblissement de propagation $L_i(d_{u,i})$, nous utilisons le modèle COST-231 [8]. La puissance reçue d'un utilisateur u à partir de la station de base de la i-ième cellule c_i est : $p_{r,i,u} = p_{t,i} - L_i(d_{u,i})$. L'utilisateur u est couvert par la cellule c_i si sa puissance reçue $p_{r,i,u}$ n'est pas inférieure à sa sensibilité p_s : $p_{r,i,u} \geq p_s$. Par conséquent, le rayon de couverture de la i-ième cellule c_i est la distance maximale $d_{u,i}$ que satisfait $p_{t,i} - L_i(d_{u,i}) \geq p_s$. Nous supposons que la sensibilité de chaque utilisateur dans le réseau est le même p_s . Le rayon de la cellule c_i est estimé par la puissance d'émission $p_{t,i}$.

0.5.2 Algorithme distribué d'équilibrage de charge

Chaque station de base c_i estime périodiquement la charge de trafic demandée, notée ρ_i, qui est générée par ses utilisateurs enregistrés comme suit :

$$\rho_i = \sum_k \rho_{i,k}/\mu_i;$$

où $\rho_{i,k}$ est le nombre de resource blocks demandés du k-ième utilisateur et μ_i est le capacité de la station de base c_i. La capacité d'une station de base est le nombre total de resource blocks disponibles dans cette station de base.

Nous supposons que toutes les cellules sont connectées par un réseau backhaul. Chaque cellule peut transmettre sa position et son rayon de couverture à d'autres cellules via ce réseau de backhaul. Nous disons que deux cellules sont voisines si la couverture de chacune des cellules croise la couverture de l'autre. Chaque cellule peut alors calculer sa table de voisines.

La station de base c_i envoie sa charge de trafic demandée ρ_i, sa position v_i et son rayon de couverture actuel r_i à ses voisines. Par conséquent, chaque station de base connaît des informations sur la charge de trafic demandée

à lui-même et à tous ses cellules voisins. Il connaît aussi la position et la couverture de tous ses voisines. Cela aide chaque cellule à calculer la couverture autour d'elle.

Une station de base est surchargée si sa charge demandée est supérieure à 1. Elle est chaude si elle n'est pas surchargée mais si sa charge demandée est supérieure à un seuil de charge haut ρ_w tel que $0 < \rho_w < 1$. Autrement, la station de base est froide.

Si une station de base c_i est surchargée, elle doit réduire sa charge de trafic demandée ρ_i. Une solution est de rediriger certains de ses utilisateurs enregistrés vers des cellules voisines, et devrait choisir une cellule voisine qui a beaucoup de ressources libres, qui est froide, pour demander une aide. La cellule voisine qui est choisie est appelée un assistant. La cellule voisine qui est chaude ne sera pas choisie pour être un assistant. Cela évite un assistant de devenir surchargé après avoir reçu les utilisateurs redirigés de la station de base surchargé c_i. La redirection des utilisateurs doit s'assurer que les utilisateurs redirigés reçoivent un signal fort pour se connecter à la station de base cible, et que en même temps, la station de base cible doit posséder suffisamment de ressources libres pour attribuer aux utilisateurs redirigés. Cependant, les utilisateurs dans la station de base surchargée peuvent être loin de la station de base de l'assistant qui est une cellule voisine. L'assistant doit augmenter sa puissance d'émission d'un montant Δp.

Soit $\mathcal{N}_i$ la collection des voisines de la station de base c_i. La station de base c_i. qui est surchargée doit choisir une cellule voisine froide, qui sera son assistant, c_h qui a la plus faible charge de trafic :

$$c_h = \arg \min_{c_j \in \mathcal{N}_i} \rho_j \text{ tels que } \rho_j \leq \rho_w.$$

S'il n'y a pas d'assistant disponible, il n'y a donc pas de cellule voisine qui peut aider c_i. La cellule c_i doit réessayer le processus dans la trame radio suivante. Lorsque c_h, l'assistant est choisi, la station de base c_i envoie maintenant une demande de l'aide à c_h et attend. Une fois la demande de l'aide reçue, l'assistant c_h augmente sa puissance d'émission d'un montant Δp_h pour attirer les utilisateurs à s'y connecter. Certains utilisateurs se reconnecteront à l'assistant après son augmentation de la puissance d'émission, cela se fait après le processus de transfert dans [1]. En même temps, l'assistant c_h transmet son nouveau rayon de couverture à d'autres cellules. Il demande également à ses voisines de réduire si possible leur puissance d'émission. Cependant, une réduction rapide de la puissance d'émission peut créer un trou de couverture. Chaque voisine de l'assistant c_h, et aussi la cellule surchargée c_i, essaie de réduire sa puissance d'émission d'un montant $\Delta p_k/2$, où c_k est le k-ième voisine de l'assistant c_h. Si cette réduction ne crée pas de trou

de couverture, la réduction est confirmée et cette cellule envoie son nouveau rayon de couverture à d'autres cellules. La voisine qui est l'assistant d'une autre cellule ignorera cette demande de réduire la puissance de transmission.

La réduction de la puissance d'émission doit satisfaire la condition de couverture, c'est-à-dire qu'aucun trou de couverture ne se présente. Pour vérifier si un trou de couverture apparaît, la cellule qui a réduit sa puissance d'émission construit un complexe de Čech pour sa région locale. Cette région locale contient cette cellule et ses cellules voisines. Puis, elle calcule les nombres de Betti de ce complexe de Čech. Si aucun trou de couverture n'est créé après la réduction, les nombres de Betti doivent être $\beta_0 = 1$ et $\beta_1 = 0$. Sinon, un trou de couverture apparaît. Si un trou de couverture apparaît, cette cellule doit retourner sa puissance d'émission à sa valeur précédente.

Cependant, il existe un cas particulier. Dans un réseau avec des petites cellules déployées, telles que : picocellules et microcellules, si une petite cellule est surchargée et qu'elle choisit une macrocellule pour être un assistant, elle n'envoie pas de demande de l'aide à la macrocellule. La petite cellule essaie automatiquement de réduire sa puissance d'émission, puis vérifie la couverture et diffuse les nouvelles informations à ses voisines. La puissance d'émission des macrocellules est beaucoup plus élevée que celle des petites cellules. Ceci est pour éviter la modification inutile de la puissance d'émission de macrocellule, qui affecte beaucoup d'autres petites cellules à l'intérieur de la macrocellule.

L'algorithme dans la Figure 20 décrit la procédure pour équilibrer de charge pour chaque macrocellule, et l'algorithme dans la Figure 21 décrit la procédure pour équilibrer de charge pour chaque petite cellule. Le processus d'équilibrage de charge est répété chaque trame radio, voir [5] pour les détails de trame radio et resource blocks des réseaux.

Il convient de noter qu'un trou de couverture peut apparaître en raison d'une information obsolète du rayon de la couverture. Par conséquent, chaque fois qu'une cellule essaie de réduire son rayon de couverture, elle envoie d'abord un signal de « pause » à ses voisins, puis elle met à jour les informations des voisins et traite la réduction. Après la réduction, la cellule envoie un signal « continue » à ses voisins pour leur indiquer qu'ils peuvent continuer. Si une cellule reçoit un signal « pause », elle interrompt le processus d'équilibrage de charge et attend qu'un signal « continue » soit reçu.

0.5.3 Complexité

L'algorithme d'équilibrage de charge comporte deux étapes principales. La première étape estime la charge de trafic μ de chaque cellule. Soit m le

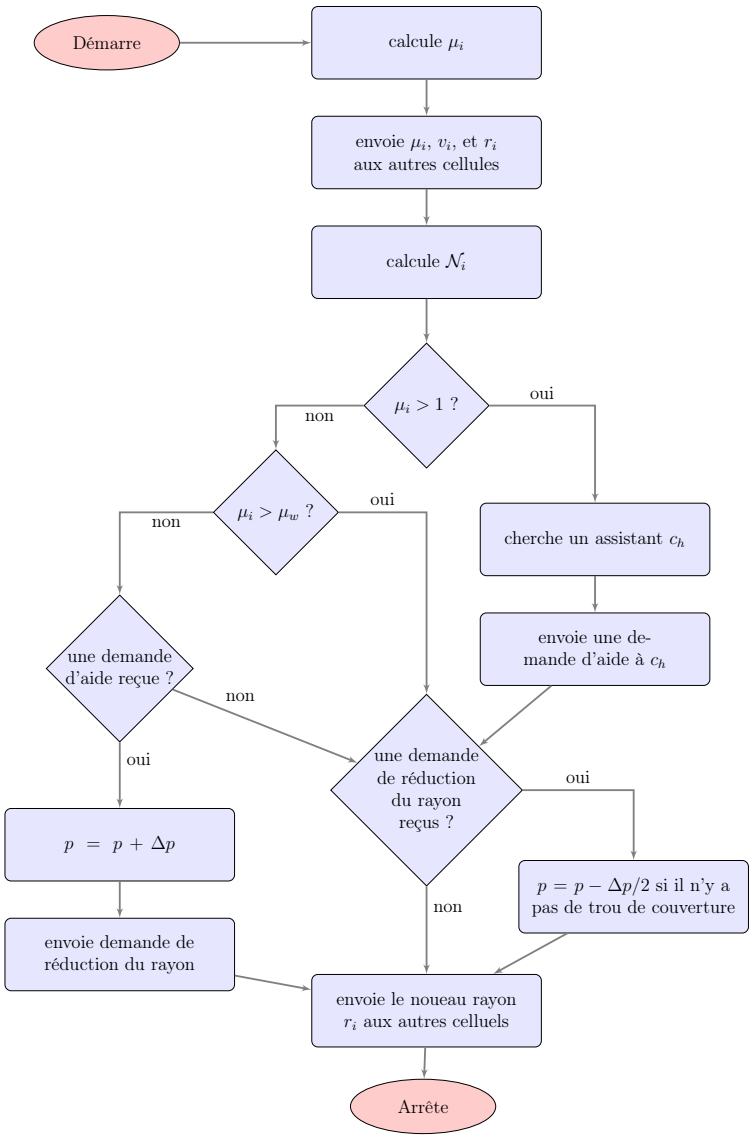


Figure 20: L'algorithme d'équilibrage de charge d'une macrocellule c_i

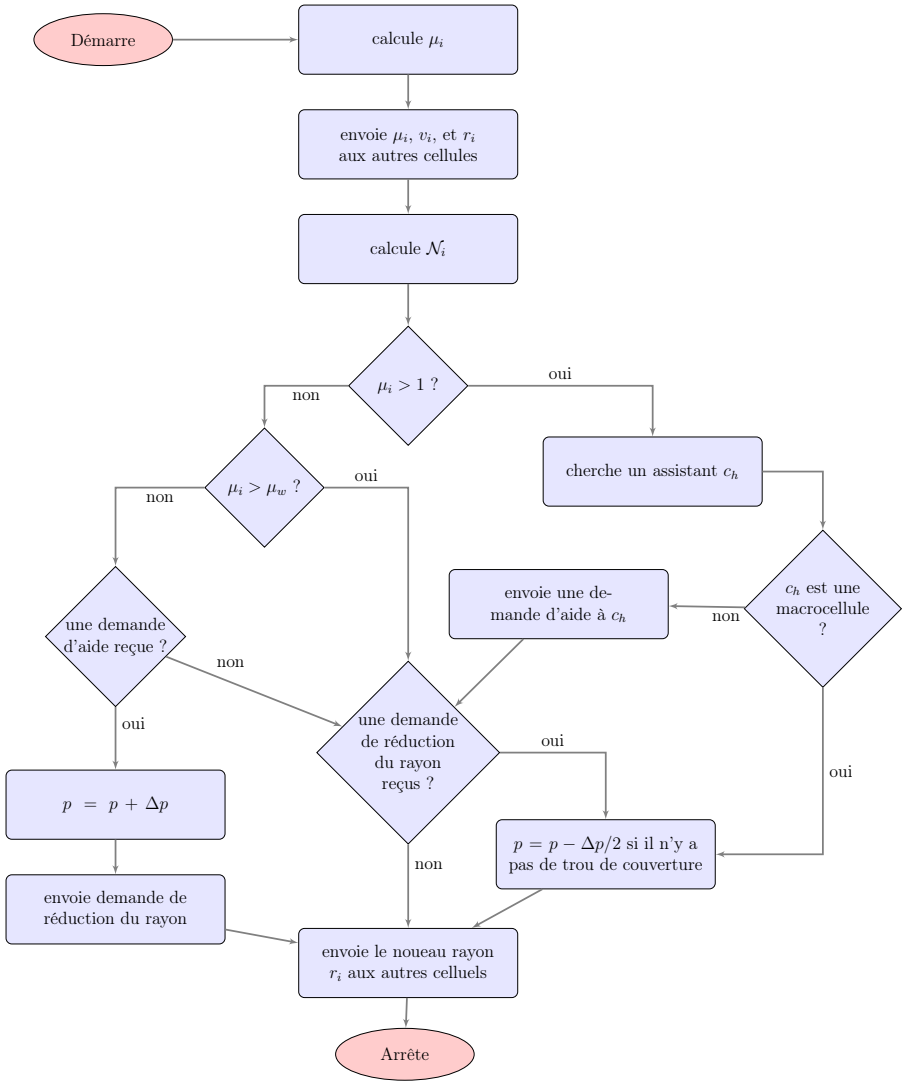


Figure 21: L'algorithme d'équilibrage de charge d'une petite cellule c_i

nombre moyen d'utilisateurs dans chaque cellule, la complexité pour estimer la charge de trafic dans chaque cellule est $\mathcal{O}(m)$. La deuxième étape vérifie la couverture de la région locale. La région locale est composée de la cellule qui modifie sa puissance d'émission et ses cellules voisines. Cette étape nécessite le calcul du complexe de Čech de la région locale et de ses nombres de Betti. La complexité de cette étape est $\mathcal{O}(n^9)$, où n est le nombre moyen de voisins de chaque cellule. La complexité de l'algorithme d'équilibrage de charge appliqué pour chaque cellule est $\mathcal{O}(m + n^9)$.

0.5.4 Simulation et résultats

Déploiement du réseau

Nous déployons les macrocellules selon le modèle hexagonal. Dans chaque macrocellule, nous déployons deux micro cellules, qui sont placées uniformément dans la couverture des macrocellules sous une condition que la distance de la station de base d'une microcellule à celle d'une macrocellule est supérieure aux deux tiers du rayon de couverture de la macrocellule. Les utilisateurs mobiles sont uniformément distribués dans la couverture du réseau. Les paramètres concernant les macrocellules et les microcellules sont résumés dans le Tableau 5 et le Tableau 6.

Table 5: Parametères des macrocellules

Paramètres	Suppositions
Fréquence de porteur	1800MHz
Capacité	50 resource blocks
Distribution des stations de base	hexagonal
Utilisateurs	distribué uniformément
Puissance maximale d'émission	46 dBm
Puissance initiale d'émission	42 dBm
Gain de l'antenne	18 dB
Affaiblissement de cable	2 dB
Rayon initial	900m
Nombre de macrocellules	91 cellules

Nous supposons que le nombre de resource blocks demandés par chaque utilisateur suit une loi de Poisson. En moyenne, chaque utilisateur demande deux resource blocks. Nous générons les resource blocks demandés de

Table 6: Paramètres des microcellules

Paramètres	Suppositions
Fréquence de porteur	1800MHz
Capacité	15 resource blocks
Distribution des stations de base	distribué uniformément
Utilisateurs	distribué uniformément
Puissance maximale d'émission	30 dBm
Puissance initiale d'émission	24 dBm
Gain de l'antenne	18 dB
Affaiblissement de cable	2 dB
Nombre de microcellules	182 cellules

l'utilisateur de telle sorte que le nombre total de resource blocks demandés dans les cellules micro et les cellules macro soit proportionnel à leur capacité.

Nous utilisons le modèle COST-231 [7] pour estimer l'affaiblissement de la propagation radio avec la correction $c = 3$ pour les régions urbaines.

$$L = 46,3 + 33,9 \log f - 13,82 \log h_B - a(h_R) + (44,9 - 6,55 \log h_B) \log d + c,$$

où $a(h_R) = (1,1 \log f - 0,7)h_R - (1,56 \log f - 0,8)$. La hauteur de l'antenne de la station de base h_B est 30m et la hauteur de l'antenne mobile h_m est $1,5$m. La sensibilité de l'utilisateur mobile est de -104 dBm. L'affaiblissement au niveau de corps de l'utilisateur mobile est 3 dB.

Dans notre simulation, chaque trame radio est de 10 ms. Dans chaque trame radio, l'algorithme d'équilibrage de charge est effectué pour chaque cellule.

Performances avec charge de trafic variable

Nous considérons la performance de l'algorithme d'équilibrage de charge lorsque les utilisateurs changent souvent leur nombre de resource blocks demandés. Nous supposons que chaque utilisateur change sa demande de resource blocks toutes les 10 trames radio. Dans chaque chaque trame radio, chaque cellule applique l'algorithme d'équilibrage de charge une fois.

Nous supposons que la charge de trafic moyenne dans le réseau est de 0,9, cela signifie que le nombre total de resource blocks demandés est de 90% de la capacité totale du réseau en moyenne. Bien que la charge de trafic moyenne soit inférieure à 1, le nombre de resource blocks demandés par chaque utilisateur est aléatoire. Certaines cellules du réseau peuvent

être en surcharge tandis que d'autres sont libres. Nous définissons la perte de chaque cellule comme le nombre total de resource blocks demandés dans cette cellule qui ne sont pas servis. La perte du réseau est la perte accumulée de chaque cellule.

Dans cette simulation, le seuil de charge haut ρ_w est réglé à 0,9 pour chaque cellule. Cela signifie qu'une cellule dont la charge de trafic est supérieure à 0,9 va ignorer la demande d'aide des cellules surchargées. Le pas de puissance Δp est 1dB pour chaque cellule. Nous effectuons la simulation dans 250 trames radio et enregistrons la perte du réseau dans deux cas : sans équilibrage de charge appliqué et avec équilibrage de charge appliqué.

Nous présentons un exemple de simulation dans la Figure 22. La perte du système dans le cas où l'équilibrage de charge n'est pas appliqué est dessinée comme la ligne rouge. La perte du système dans le cas où l'équilibrage de charge est appliqué est dessinée comme la ligne bleue. La ligne bleue est sous la ligne rouge, ce qui montre qu'en utilisant l'équilibrage de charge, la perte du système est réduite. La réduction de la perte est le gain de capacité du système. Dans la Figure 22, la différence entre la ligne rouge et la ligne bleue décrit le gain de capacité du système grâce à l'algorithme d'équilibrage de charge.

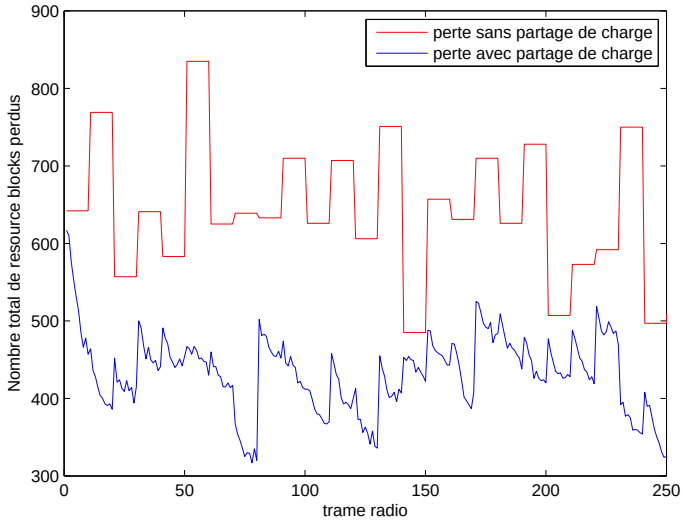


Figure 22: Perte du réseau dont trafic est variable avec et sans partage de charge.

Nous répétons cette simulation 1000 fois, et calculons la réduction moyenne de la perte du réseau lorsque l'équilibrage de charge est appliqué. Cet algo-

rithme d'équilibrage de charge réduit de 25,7% la perte, et gagne 2,3% de la capacité de l'ensemble du système.

Performances avec charge de trafic constante

Nous considérons la performance de l'algorithme d'équilibrage de charge lorsque la charge de trafic est maintenue constante. En effet, la demande de resource blocks de chaque utilisateur est inchangée pendant la simulation. Certaines propriétés de l'algorithme d'équilibrage de charge telles que : le gain de capacité, la vitesse de convergence et la stabilité sont considérées à une charge de trafic totale du réseau T, qui est supposée séquentiellement de 0,5 (bas) à 0,9 (élevé).

Nous effectuons l'algorithme d'équilibrage de charge avec différentes valeurs du seuil de charge haut ρ_w et de le pas de puissance Δp. La simulation est réalisée en 200 trames radio. Chaque fois que nous changeons la valeur d'un de ces paramètres, nous répétons la simulation 100 fois. Le seuil de charge haut ρ_w et le pas de puissance Δp sont supposés être les mêmes pour chaque cellule.

La Figure 23 montre les performances de l'algorithme d'équilibrage de charge lorsque la charge de trafic du réseau est $T = 0,9$, une valeur élevée. Le seuil de charge haut ρ_w est 0,9 et le pas de puissance Δp est 1 dB pour toutes les cellules. Les utilisateurs ne changent pas leur demande de resource blocks dans chaque simulation, alors la perte du réseau est la même dans chaque trame radio. La perte moyenne du réseau lorsque l'équilibrage de charge n'est pas appliqué est constante (ligne rouge). La ligne bleue trace la perte moyenne du réseau lorsque l'équilibrage de charge est appliqué. La ligne pointillée montre la borne supérieure et la borne inférieure de l'intervalle de confiance de 95% de la perte moyenne du réseau équilibré. La perte du réseau équilibré est rapidement réduite, elle est réduite de 34% dans 28 trames radio. Lorsque $T = 0,5$ (Figure 24), la perte du réseau est réduite de 80% pendant 16 trames radio.

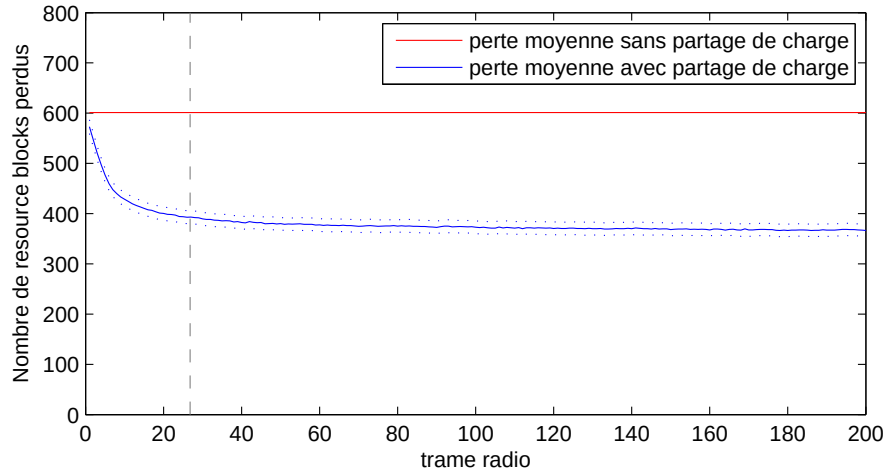


Figure 23: Perte moyenne du réseau avec charge du trafic $T = 0,9$, pas de puissance $\Delta p = 1$ dB et seuil charge haut $\rho_w = 0,9$.

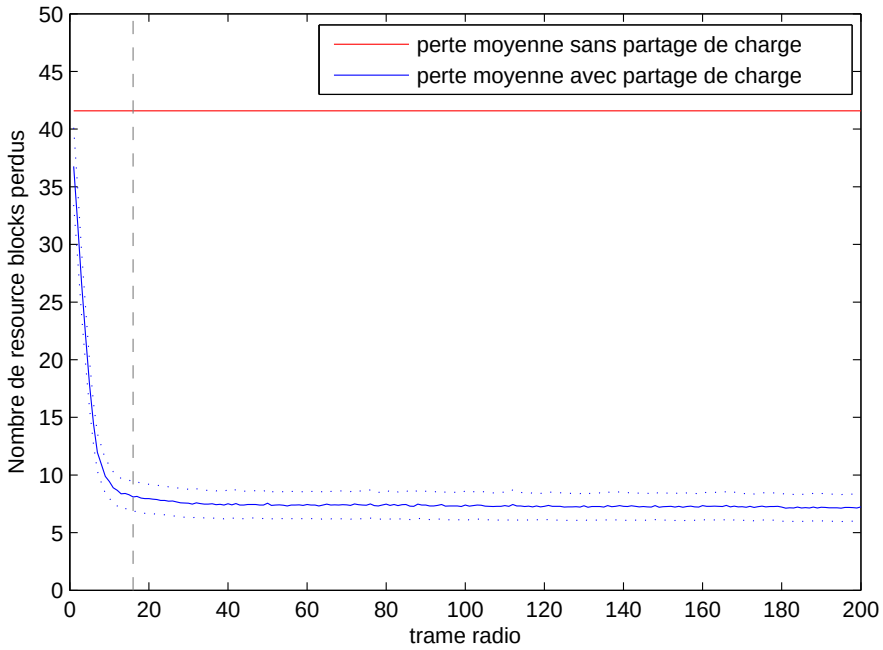
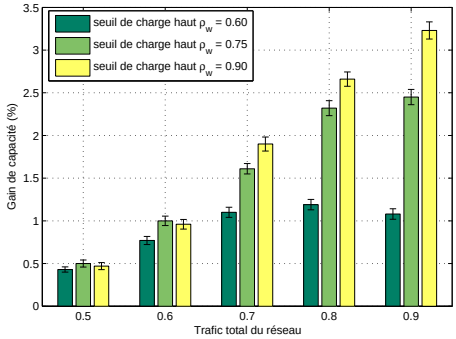


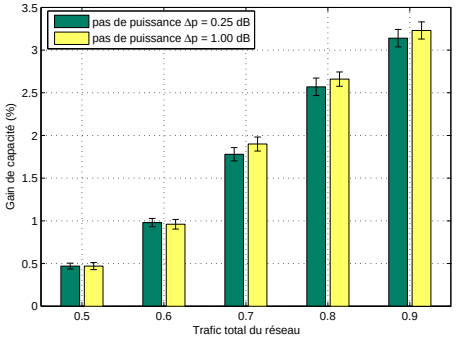
Figure 24: Perte moyenne du réseau avec charge du trafic $T = 0,5$, pas de puissance $\Delta p = 1$ dB et seuil charge haut $\rho_w = 0,9$.

La réduction dans la perte est le gain de capacité du réseau. Le gain

de capacité est enregistré au dernier moment de la simulation, la 200-ième trame radio. La Figure 25(a) compare le gain de capacité du réseau à différentes charges de trafic demandées T lorsque l'équilibrage de charge est appliqué avec différents seuils de charge haut ρ_w, le pas de puissance Δp est 1 dB. Lorsque la charge de trafic demandée du réseau est élevée, le seuil de charge chaud plus élevé ρ_w est donne le gain de capacité plus élevé. Sur la Figure 25(b), le gain de capacité est comparé lorsque l'équilibrage de charge est appliqué avec des pas de puissance différents $\Delta p = 1$ dB et $\Delta p = 0,25$ dB. Le seuil de charge haut ρ_w est 0,9. A chaque charge de trafic demandée T, les gains de capacité à long terme avec des pas de puissance Δp différents sont pratiquement égaux



(a) valeurs différentes ρ_w et $\Delta p = 1$ dB



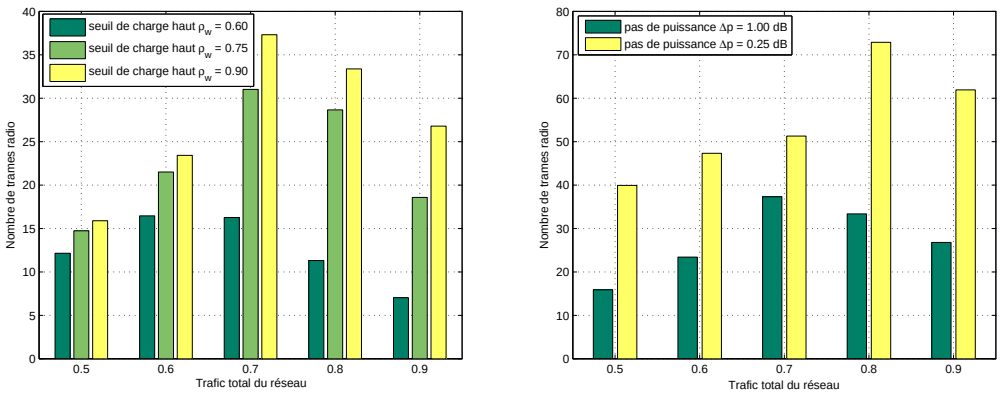
(b) valeurs différentes de Δp et $\rho_w = 0,9$

Figure 25: Gain de capacité du système

Les différentes valeurs du seuil de charge haut ρ_w et du pas de puissance Δp peuvent également affecter d'autres propriétés de performance telles que : la vitesse de convergence et la stabilité. Nous considérons que la perte du réseau équilibré approche la valeur de la stabilité dans les 100 premiers trames radio. Ensuite, nous calculons la valeur moyenne de la perte du réseau équilibré dans les 100 dernières périodes de temps. Nous désignons cette moyenne par M. Lorsque la perte du réseau équilibré est inférieure à la valeur $1,05 \times M$, nous disons que le réseau converge à l'instant courant t. La durée entre le point de départ et le temps courant t est définie comme le temps de convergence du réseau.

Le temps de convergence est représenté sur la Figure 26(a) avec différents seuils de charge haut ρ_w, et sur la Figure 26(b) avec différents pas de puissance Δp. Avec un seuil de charge haut plus élevé ρ_w, il y a plus d'assistants disponibles, donc plus de demandes d'aide peuvent être satisfaites. Par conséquent, le processus des aides est plus long, et le temps de convergence est

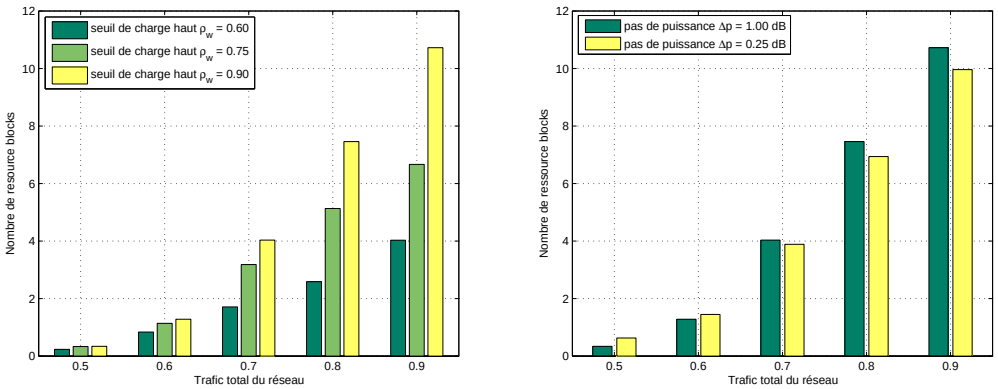
plus long. Le temps de convergence augmente si le pas de puissance Δp est réduit. Une augmentation significative du temps de convergence est observée lorsque le pas de puissance Δp est réduit, voir la Figure 26(b). Comme le montre la Figure 25(b), les gains de capacité avec des pas de puissance Δp différents sont pratiquement les mêmes. Ceci montre qu'un pas de puissance Δp plus petit réduit la vitesse de convergence mais ne change pas le gain de capacité à long terme. Dans le réseau de charge de trafic variable, nous ne devrions pas choisir un pas de puissance trop petit Δp en raison de la convergence lente.



(a) valeurs différentes de ρ_w et $\Delta p = 1$ dB (b) valeurs différentes de Δp et $\rho_w = 0,9$

Figure 26: Le temps de convergence

Nous utilisons l'écart-type de la perte du réseau équilibré à partir de la valeur moyenne M, notée σ en tant que mesure de la stabilité du réseau. La Figure 27(a) et la Figure 27(b) montrent des valeurs σ avec ρ_w et Δp différents. Plus le seuil de charge haut est élevé, meilleure est la stabilité du système. En outre, la réduction du pas de puissance peut augmenter un peu la stabilité du système.



(a) valeurs différentes de ρ_w et $\Delta p = 1$ dB (b) valeurs différentes de Δp et $\rho_w = 0,9$

Figure 27: L'écart type de la perte du réseau équilibré

0.6 Conclusions

Nous avons étudié des applications de l'homologie simpliciale dans les réseaux sans fil. Les résultats majeurs de notre travail portent sur 3 points :

- **La construction du complexe de Čech des réseaux sans fil.**

  Le complexe de Čech représente exactement la topologie des réseaux sans fil. Donc, nous avons développé la construction du complexe de Čech destinée aux applications dans les réseaux sans fil. Cette construction est présentée dans des versions centralisées, parallèles et distribuées. Il est à noter que la complexité de cette construction est de temps polynomial.

- **Les algorithmes basés sur l'homologie simpliciale pour économiser l'énergie dans les réseaux sans fil.**

  Dans le déploiement aléatoire de cellules d'un réseau sans fil, les algorithmes d'économie d'énergie permettent tout d'abord de maximiser la couverture du réseau. Toutes les cellules sont configurées pour fonctionner avec la puissance d'émission maximale. Le complexe de Čech est utilisé comme un outil pour capturer la topologie du réseau. Ensuite, ces algorithmes réduisent la puissance d'émission autant que possible pour chaque cellule sans modifier la topologie du réseau. Enfin, la puissance d'émission est minimisée et la couverture est maintenue à la valeur maximale. Ces algorithmes sont disponibles en versions centralisées, parallèles et distribuées. Sur la base de notre simulation, nos

algorithmes peuvent économiser jusqu'à 65% de la puissance d'émission maximale du système en temps polynomial.

- **L'algorithme distribué d'équilibrage de charge**

  Nous avons introduit un algorithme distribué pour équilibrer la charge de trafic dans les réseaux sans fil. Cet algorithme augmente la puissance d'émission des cellules à faible charge et réduit l'une des cellules surchargées sous la contrainte de couverture, ce qui entraine la redistribution de certains utilisateurs des cellules surchargées vers les cellules libres. Le réseau garantit la couverture et la meilleure condition de connection pour les utilisateurs. Cet algorithme améliore la capacité de 2,3% de l'ensemble du réseau lorsque les demandes de charge de trafic d'utilisateurs varient très rapidement.

  Cependant, les valeurs du seuil de charge haut ρ_w et du pas de puissance Δp doivent être soigneusement sélectionnées pour chaque cellule afin de conserver les meilleures performances pour le réseau. La valeur supérieure du seuil de charge haute ρ_w donne un gain de capacité plus élevé, mais il rend le réseau moins stable. Un pas de puissance faible Δp réduit la vitesse de convergence de l'algorithme. Dans le cas où les utilisateurs modifient leur demande de resource blocks très rapidement, un petit pas de puissance Δp réduira le gain de capacité de l'algorithme en raison de la vitesse lente de convergence.

Chapter 1

Introduction

1.1 Motivations

Advancements in wireless communication technologies enable more and more innovative and effective applications. Indeed, they are widely used not only to facilitate people life but also to improve business jobs. However, with the popularity of wireless applications, which always need frequently exchanged data, a notable growth of mobile traffic load is recorded in recent years. According to a recent projection [4], the mobile traffic load will continue to increase dramatically in coming years. By 2021, the mobile traffic load will be ten times higher than the one of 2015. This exponential growth of mobile traffic load is really a hard challenge for the next generation of wireless networks.

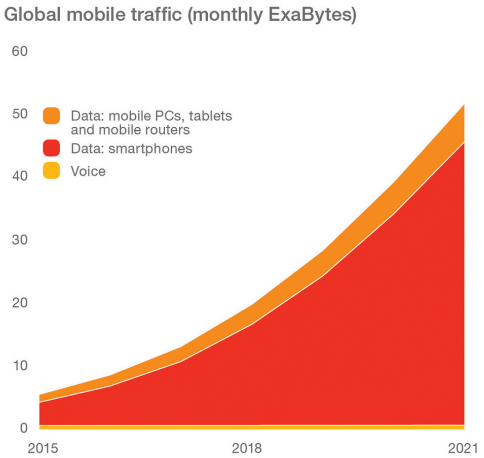


Figure 1.1: Global mobile traffic [4]

To support the exponential growth of mobile traffic load, only improving the existing macro cells is not adequate. In fact, the capacity of macro cells can not be extended to unlimited value. The promising solution is to densify the wireless cells. The wireless cells can be classified as macro cells and small cells such as micro cells, pico cells, femto cells, etc. The macro cells can be split, this increases the spectral efficiency due to a higher frequency reuse factor. In addition, smaller cells use less transmission power because of the low propagation loss in short distance [45]. Following [53], it is estimated that 80% of traffic load comes from indoor locations, such a huge traffic in small local regions. However, the indoor mobile users often suffer from poor coverage, but there should be the same experience for indoor mobile users and the outdoor ones. The small cell has an advantage that it is easy to implement anywhere we need. A lot of small cells can be installed inside buildings, stadiums, parks, etc. in order to immediately boost the capacity there. These small cells also improve the coverage for indoor locations. In the LTE-Release 13 version, the carrier for mobile users can be integrated in Wi-Fi access points, which are almost connected to super high speed optical network [9, 10]. Therefore, each Wi-Fi access point can work as a small cell access point in mobile networks. At the same time, they can receive the offloaded traffic from macro cells when needed.

The next generation of wireless networks will be an ultra dense one, where massive macro cells and small cells are seamlessly deployed. However, an unoptimized ultra dense network may suffer from the interference problem, which strongly degrades the quality of the network. The coverage of network should be optimized. The topology of the network will be the key factor to optimize and enable features for the ultra dense network. In addition, the traffic load is huge but it is not constant: it fluctuates over time. The network should have the ability to adapt to the fluctuation of traffic load in time, and does not need any human task. The new paradigm to design and manage the next generation of wireless networks will be based on topology.

In this thesis, we consider the wireless network as a random deployment. The base station can be put everywhere and can be any kind. Then, the network will be optimized to avoid the interference. Some redundant cells can be turned off to save power. The active cells work with the optimized transmission power to minimize the interference with neighbors. The redundant transmission power is also neglected. With the optimized topology, the network is always covered and provides seamless connection to mobile users. If the overload happens somewhere, the network will detect that and will automatically update the size of each cell by modifying the transmission power of each one. With this adaptation, the mobile users in overloaded cells will be redirected to the free cells. The network is self-optimized.

The topology of the network can be captured by a tool from algebraic topology named Čech complex. Actually, the Čech complex is an abstract simplicial complex. A strong advantage of Čech complex is that it can capture exactly the topology of the network [23]. Simplicial complex gives information about coverage of the network by only matrix computations via its homology. Recently, it has been applied to detect the coverage holes in wireless sensor networks [21, 56, 67]. There are also some developed applications that are based on simplicial homology for wireless communication networks. In [62], the authors introduced a reduction method that can be used to turn off cells for power saving. In [63], the authors developed a homological method to automatically plan the frequency in wireless networks. And in [64], an algorithm to recover the network after a disaster has been proposed. The simplicial complex can be applied more in wireless communication networks. In this thesis, we use simplicial complex as a tool to represent the topology of network and to develop our applications.

1.2 Thesis contributions and outline

The thesis is organized as follows. First, we provide the state of the art in the first part of Chapter 2. In the later part of Chapter 2, we introduce all the mathematical backgrounds that are used for our works. Indeed, we introduce the simplicial complex, its homology group, and how it can help us to get the information about the topology of the network such as: connectivity and coverage. Some illustrated examples are also given. Then, our contributions will be presented in the consecutive chapters.

As we discussed in the introduction, the network topology is the key factor to design and enable the features of the next generation of wireless networks. The Čech complex is a useful tool to capture exactly the topology of the network. However, the Čech complex for wireless networks has not been developed. That is the reason why we begin with the construction of the Čech complex for wireless networks. All the details about the construction of the Čech complex are presented in Chapter 3. We have shown that the minimal Čech complex that captures information about coverage and connectivity of the network is constructed in only polynomial time. This tells us that the algorithm to construct the Čech complex is really applicable. We also propose the parallelized and distributed version of this algorithm, which divides the whole computation into smaller parts and computes them on several computers at the same time. Therefore, these versions enable a faster computation. The complexity of these algorithms are computed for different cases: the minimal Čech complex and the Čech complex with the highest

dimension. Some simulations have been done. The purpose of these simulations is to verify the performance of these proposed algorithms in real time. The process of the Čech complex computation is shown by corresponding illustrated pictures. This work has been published in [43].

Once the Čech complex has been already constructed, we now can consider some of its applications. The next chapter, Chapter 4, introduces the first one. In this chapter, we consider the random deployment of a wireless network. It is not required to install the base stations according some fixed patterns such as: hexagon, grid lines. It is free to put them everywhere we need. The coverage radius of each cell can be modified by changing its transmission power. The coverage is the most important factor which decides the quality of services of every network. We use the Čech complex to capture the topology of the network, and the coverage of the network is tractable from the homology of this Čech complex. We optimize the transmission power of each cell under the coverage constraint: keeping the maximal coverage of the network. The purpose of this optimization is to minimize the interference caused by unoptimized transmission power of cells. After the optimization, some redundant cells are turned off. The remaining cells, who are active, are optimized. Each one has the smallest interference with its neighbors. The wastage power, which mainly causes the interference, is avoided. Therefore, the total transmission power of the network is minimized. Simulated annealing is a heuristic method to find an approximation of the global optimum. Downhill move is also a heuristic method, but it finds local optimum and runs faster. We propose a simulated annealing algorithm and a downhill one to optimize the network, then we compare the results obtained by these algorithms. We also introduce the parallel and distributed version of the downhill algorithm for a faster computation. The complexities of these algorithms are also computed and compared in different cases where the minimal Čech complex or the one with its highest dimension is used. Some simulations are also done in order to analyze the performance of these algorithms. We have published this work in [44].

The next chapter of the thesis considers the traffic problem in the wireless networks. The traffic generated by each user is not constant. In addition, the users are not always standing, they can move. Therefore, the traffic load of each cell varies with time. Sometimes, some cells can be overloaded, while others are still free. It is better if we can redirect some users from the overloaded cells into the free cells. As we know, the mobile user always chooses the base station whose signal is best. Therefore, by modifying the transmission power of cells, we can redirect users from the overloaded cells to free ones following the handover procedure [1]. We propose an algorithm to balance the traffic load for wireless networks in this chapter. The cells can

enlarge or shrink always under the coverage constraint. The users are always covered with sufficient received signal for their connection. We also have a strategy to reduce the interference due to the modification of cells' size. As the result, the users have a better condition to communicate and the load of each cell is balanced. The simulation to evaluate the performance of the proposed algorithm is presented in the later part of this chapter. The results of this chapter will be submitted to a future conference.

Finally, the Chapter 6 draws conclusion. Also in this chapter, notable contributions are reminded, and some possible directions for future works are discussed.

Chapter 2

Related works and mathematical background

2.1 Related works

2.1.1 Representations of wireless network

For many years, the hexagonal model is used to represent and design the wireless networks. In this model, each cell is represented by a hexagon, and is placed next to each other without overlap. This model is practical to represent the macro cells system [32]. More fixed pattern models are available in [14]. However, the wireless networks have changed so much. The current wireless networks have not only macro cells but also a lot of small cells such as: femto cells, pico cells, micro cells. The next generation of wireless networks is ultra dense cells one. These cells have different sizes. In addition, the small cells are almost deployed inside buildings, stations, parks, which do not follow any fixed pattern. Therefore, the hexagonal model is not convenient to represent the current and future wireless networks.

Another approach is to consider the wireless network as a random deployment. The Voronoi diagram can be used to represent the network. Given a set of vertices, each cell of a vertex in the Voronoi diagram is a region in the plane such that every point in this region has the closer distance to this vertex than to other vertices. However, the Voronoi diagram does not give any information about the coverage of the network. Voronoi diagrams consider only the position of base stations and ignore their transmission power. They are often used in applications such as topology control, and routing in Wireless Adhoc Networks [55,59]. Some probability models have been introduced to examine the coverage of a random sensor network [42,65]. However, these works give only asymptotic results and the same radius cells are assumed.

Recently, simplicial complex is used to represent the random deployment. The homology of the simplicial complex gives us the information about the topology of the deployment, such as: connectivity and coverage. Ghrist introduced an article about homological sensor networks in [56]. The deployment of the sensor networks is often dense and random. It looks like a small scale or miniature picture of wireless communication networks. Therefore, the simplicial complex is also suitable to represent the wireless communication networks as well. There are some developed applications that are based on simplicial homology. A literature review about these applications will be denoted in the next sub-section.

2.1.2 Simplicial homology

The simplicial homology is a tool from algebraic topology. Recently, notable applications based on simplicial homology have been developed. Most of them are applied in Wireless Sensor Networks. The homological sensor networks is introduced by Ghrist *et al* in [56]. The first article that introduces the simplicial homology to solve the coverage problem is published in [23]. Then, a method based on simplicial homology to detect coverage holes is presented in [31]. In this article, the authors introduce the Čech complex as a tool to capture exactly the topology of the network, and the Rips complex as an approximation of the Čech complex. An algorithm to construct the Čech complex is proposed in [20], but this algorithm can only build the Čech complex for a collection of same sized cells. Therefore, this algorithm is not appropriate to construct the Čech complex for wireless networks, in which cells have different sizes. The Rips complex does not require the location information. A location free method to detect coverage holes in Wireless Sensor Networks has been introduced in [21]. However, the Rips complex is still an approximation of the Čech complex. It can capture the topology of network with failure. The accuracy of homological method is discussed in [66]. Some relations of the Čech complex and the Rips complex are noted in [22]. Some distributed versions of the algorithms to detect coverage holes are also presented in [24, 67]. In [62], the authors introduce a reduction algorithm for simplicial complex. This algorithm can be used to turn off redundant cells without making coverage holes.

There are also some developed applications that are based on simplicial homology for wireless communication networks. In [63], the authors introduce the simplicial homology for future wireless networks. In [64], an algorithm to recover the network after a disaster is proposed.

2.1.3 Energy saving in wireless networks

Energy saving is a hot topic for many researchers in recent years. A survey about technologies for energy saving has been done in [18]. Following this survey, base station is the part that consumes most energy in wireless networks. Therefore, the most effective way to reduce the consumed power in wireless networks is to improve the base station. A new architecture of base station is discussed in [29]. However, replacing hardware parts is high cost.

Some technologies for green wireless networks has been proposed in [28, 34]. Energy for radio transmission can be saved during low-traffic time. Base stations often have more than one antennas, some of these antennas can be turned off [8, 11, 57]. Furthermore, some base stations are possible to be turned off [39, 49] during low-traffic time and waked up when needed. In addition, the discontinuous radio transmission is also concerned in [30] to save power. The transmission power of small cells can be adapted with the traffic load for a better energy saving [47].

Energy can be saved by using higher spectral efficiency system. In [46], transmission power is minimized with given throughput required. In [60], the authors optimize the resource allocation scheme for energy efficiency. The spectral efficiency can be increased by using smaller cells due to higher frequency reuse [45]. In [13], the authors show that deployment of pico cells along with traditional macro cells can improve the energy efficiency of the networks.

However, using smaller cells makes networks become dense, and it increases interference. Interference causes the wastage of transmission power, so the interference management is needed. In dense networks, interference is an important problem. Many methods are proposed to cancel the interference, see [12] for an overview.

We have reviewed some recent works aimed at saving energy. These works consider the reduction of unnecessary transmission power during the low-traffic time, the spectral efficiency improvements and the interference reduction in wireless networks. However, coverage, one of the most important key factors of the wireless networks, has not been optimized. The optimized coverage, which reduces the interference and avoids the redundant transmission power, is needed.

2.1.4 Load balancing in wireless networks

There exists some load balancing methods. The first one is to borrow the bandwidth from the low traffic loaded cells to give to the overloaded cells. Some borrowing algorithms can be found in [27, 37, 38, 68]. However, the

bandwidth borrowing may increase interference in networks. When the cell size is small, the co-channel interference is hard and it degrades the quality of services. Nowadays, the wireless system is single carrier one in the uplink [61], so this method is not useful.

Another method is to modify the hysteresis margin in handover scheme between overloaded cell and low traffic loaded cell. This method forces the users to shift from the overloaded cell into the low traffic loaded cell. For more details about the handover scheme and the hysteresis margin, see [1]. Some applications based on this method are discussed in [17, 48]. However, the low traffic loaded cell, which is the target cell, is one of the overloaded cell's neighbors. Therefore, with the long distance from these shifted users to the target cell, these shifted users may have bad connection. In addition, almost all of the resource allocation schemes do not give access to the users whose connection is bad [58]. Finally, these shifted users are still out of connection. The target base station should increase its transmission power to cover these shifted users.

Recently in [25], the authors change the power of antenna beams in selected direction. The overloaded base station will reduce the power of the antenna beams in the direction to the place in which some users are in outage, while the target one, the low traffic loaded base station, increases its power of the antenna beams. Due to the handover scheme, the users that are in outage will reconnect to the target base station, who has stronger signal. However, this method requires the full access to the position of the users, which is not always available. In addition, modifying the power of antenna beams changes the shape of cells, but no method to verify the coverage after this modification is presented.

The load balancing should guarantee not only the free resource for users, but also the coverage as well as the conditions of connection for them.

2.1.5 Centralized, Parallel and Distributed computing

In this thesis, we propose algorithms in some different versions: centralized one, parallel one and distributed one. Therefore, we need to denote here some basic notions about varied types of computing.

The definition about the centralized computing is quite clear: the centralized computing is the computing done in only one central location. If the computing is done in different locations, it is called decentralized computing. There are two main types of decentralized computing: parallel on and distributed one. However, the terms "parallel computing" and "distributed computing" sometimes are not clearly distinct.

Parallel computing is the simultaneous execution of the same task on multiple processors in order to obtain the results faster. The parallel computing is usually used with applications that can be split into smaller tasks that can run simultaneously on different processors. In parallel computing, all processors may have access to a shared memory in order to exchange information between processors.

A distributed system is a collection of autonomous computers that communicate with each other to achieve the results. The computers in distributed system do not have any shared memory or processor. They exchange the information by messages. A distributed system can be designed as a client-server system, or as a peer-to-peer system.

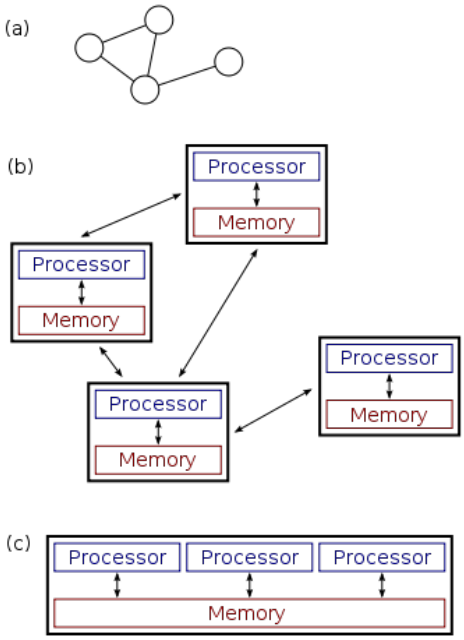


Figure 2.1: Parallel system and distributed system [3]

In the Figure 2.1, the system (a) and (b) are distributed, while the system (c) is parallel. For more details about the parallel computing and distributed computing, see [2, 40].

2.2 Mathematical background

In this section, we introduce some notions of algebraic topology, they are: simplicial complex, its homology group, and the Čech complex, which are the tools that we used in our applications in the later parts. We also explain how

these tools can be used to capture and analyze the topology of the network. For more details about algebraic topology, see [35, 50].

2.2.1 Simplicial complex

A complex in general means a collection of basic elements that are structured under certain conditions. In simplicial complex, the basic elements are simplices. Let $S_k = \{v_0, v_1, \ldots, v_k\}$ be a geometrically independent set of $k+1$ points in $\mathbb{R}^n$, where $n > k$. The convex hull of S_k is called a k-simplex, denoted s_k. The number k is its dimension and $v_0, v_1, \ldots, v_k$ are its vertices. A single point is always geometrically independent, so are two distinct points, three non-collinear points, four non-coplanar points, and so on. Thus, a single point is a 0-simplex, an edge is an 1-simplex, a triangle is a 2-simplex, a tetrahedron is a 3-simplex, and on and on. See Figure 2.2 for instances.

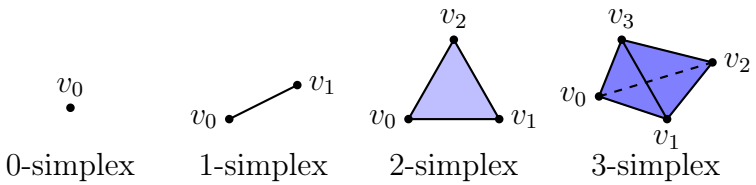


Figure 2.2: An example of simplices.

Any subset of S_k is also geometrically independent, therefore the convex hull of this subset is a simplex of a lower dimension. Let l be the dimension of this simplex, this simplex is called a l-face of s_k. So, any k-simplex always have $k+1$ $(k-1)$-faces. For example, a tetrahedron, which is a 3-simplex, has four 2-faces which are triangles.

The simplicial complex K in $\mathbb{R}^n$ is a collection of simplices in $\mathbb{R}^n$ such that: (1) every face of a simplex in this collection is also a simplex included in this collection, and (2) the intersection of any two simplices in this collection is a face of each of them. For example, any simplex can be considered as a simplicial complex that contains itself and its faces of lower dimension.

To describe a simplex, it is enough to know the collection of its vertices. Therefore, we can denote a simplex by its vertices collection. For example, the set $\{v_0, v_1, \ldots, v_k\}$ denotes the k-simplex spanned by the vertices $v_0, v_1, \ldots, v_k$. By doing so, the simplicial complex becomes a system of subset of vertices, which is a purely combinatorial structure. This system is called an abstract simplicial complex. Formally, an abstract simplicial complex is a collection K of non-empty finite sets, such that if A is an element in K, so is every nonempty subset of A. This condition is similar to the first

condition of the simplicial complex. The abstract simplicial complex is a purely combinatorial description of the geometric simplicial complex, therefore does not need the second condition about the property of intersection between simplices.

For the remainder of this manuscript, the word "abstract", the adjective in the notion "abstract simplicial complex" might be dropped for a better reading. In fact, all the "simplicial complex" in the later part indicate "abstract simplicial complex".

2.2.2 Homology group

Let us denote K an abstract simplicial complex, we now introduce the way how its homology group is obtained. Firstly, we define the orientation for simplices. The orientation of a simplex is determined by an ordering of its vertices. The orientation of a simplex change into the inverse one if the ordering of its vertices is transformed by an odd permutation. For instance, if two vertices of a simplex are swapped, then the orientation of this simplex changes to the opposite one, denoted by the negative sign as:

$$[v_0, v_1, \ldots, v_i, \ldots, v_j, \ldots, v_k] = -[v_0, v_1, \ldots, v_j, \ldots, v_i, \ldots, v_k],$$

where $[v_o, v_1, \ldots, v_k]$ presents the simplex $\{v_0, v_1, \ldots, v_k\}$ together with an ordering of its vertices. Thus, a 0-simplex has only one orientation. A simplex with dimension not zero always has exactly two orientations, which are opposite.

Once all the k-simplices of K are oriented, we can define a simplicial k-chain as a finite linear combination of these oriented k-simplices. For each k, the k-chain group $C_k(K)$ is the vector space spanned by the set of oriented k-simplices of K.

Let $[v_0, v_1, \ldots, v_k]$ be a k-simplex, which is viewed as a basic element of C_k. The boundary map ∂_k is defined to be the linear transformation $\partial_k : C_k(K) \to C_{k-1}(K)$ which acts on basis elements via:

$$\partial_k[v_0, v_1, \ldots, v_k] = \sum_{i=0}^{k} (-1)^i [v_0, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k],$$

where $[v_0, v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_k]$ is the i-th face obtained by deleting the i-th vertex. This formula tells us that the boundary of a k-simplex is its collection of $k-1$ faces, as illustrated in Figure 2.3.

The repetitions of the boundary operator give a chain complex: a sequence of vector spaces and linear transformations

$$\ldots \xrightarrow{\partial_{k+2}} C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \xrightarrow{\partial_{k-1}} \ldots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0.$$

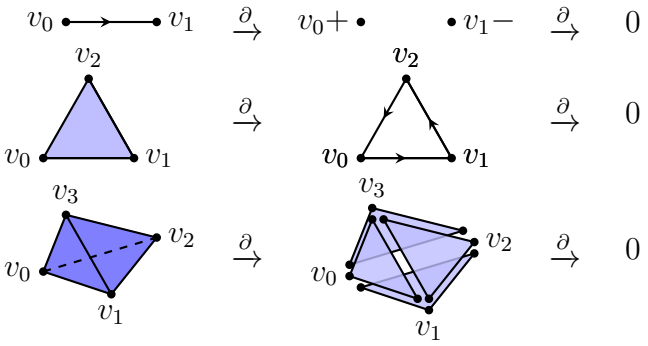


Figure 2.3: Boundary operator.

A simple computation shows that $\partial_k \circ \partial_{k+1} = 0$, this means that the boundary of a chain has no boundary. See Figure 2.4 for an example. The boundary of a 2-simplex $[v_0, v_1, v_2]$ is a 1-chain that is the sum of $[v_0, v_1]$, $[v_1, v_2]$ and $[v_2, v_0]$. The boundary of each 1-simplex in this chain is its two endpoints with opposite orientations. Therefore, the boundary of this 1-chain is the term of summations canceled in pairs.

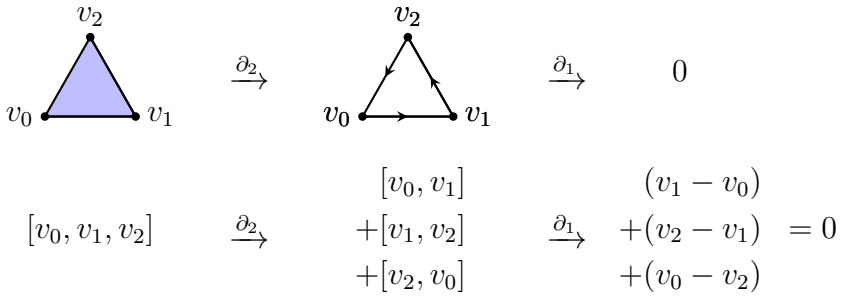


Figure 2.4: Boundary has no boundary.

We call a k-chain a k-cycle if its boundary is zero. So, the group of k-cycles, denoted as $Z_k(K)$, is the kernel of $\partial_k : C_k(K) \to C_{k-1}(K)$. Let $B_k(K)$ be the group of k-boundaries, it is automatically the image of $\partial_{k+1} : C_{k+1}(K) \to C_k(K)$. Since $\partial_k \circ \partial_{k+1} = 0$ for each k, it implies that every k-boundary is also a k-cycle. It follows that $B_k(K) \subset Z_k(K)$ for all k. Two k-cycles are said to be homologous if their difference is in the group of k-boundaries. We can now define the k-th homology group of K be the quotient vector space:

$$H_k(K) = Z_k(K)/B_k(K).$$

The dimension of k-th homology group is called the k-th Betti number.

$$\begin{aligned}\beta_k &= \dim H_k \\ &= \dim Z_k - \dim B_k \\ &= \dim \ker \partial_k - \dim \operatorname{im} \partial_{k+1}\end{aligned} \tag{2.1}$$

This number has an important meaning in solving coverage problems. It counts the number of independent k-cycles, which is k-th dimensional holes. For example, the 0-dimensional Betti number counts the connected components while the 1-dimensional Betti number counts the coverage holes.

We explain the relation between the homology group and its dimension, the Betti number, by studying an example of simplicial complex in Figure 2.5. This relation shows how the k-dimensional Betti number can count the k-dimensional holes. This figure shows a simplicial complex whose geometric representation is composed of a filled triangle and an empty triangle. This simplicial complex contains four vertices: v_0, v_1, v_2, and v_3; five edges: $[v_0, v_1]$, $[v_0, v_2]$, $[v_1, v_2]$, $[v_1, v_3]$ and $[v_2, v_3]$; one triangle $[v_0, v_1, v_2]$. This simplicial complex is connected, because every vertex is connected by edges. The 1-cycle group of this simplicial complex contains three elements:

$$\begin{aligned}c_1 &= [v_0, v_1] + [v_1, v_2] + [v_2, v_0]; \\ c_2 &= [v_1, v_3] + [v_3, v_2] + [v_2, v_1]; \\ c_3 &= [v_0, v_1] + [v_1, v_3] + [v_3, v_2] + [v_2, v_0].\end{aligned}$$

The 1-boundary group of this simplicial complex contains only one element, that is the chain c_1. We can see that the chain c_1 is an 1-cycle and also an 1-boundary. In addition, the difference between the cycle c_2 and c_3 is c_1, an 1-boundary. We say that the cycles c_2 and c_3 are homologous by module c_1. So, the number of independent 1-cycles, the 1-dimensional Betti number β_1, is one. This concludes that although there are three different 1-cycles, all these 1-cycles cover the same 1-dimension hole.

We can also compute the Betti numbers of this simplicial complex to verify its geometrical properties. The dimension of this simplicial complex is two, so we need to compute the boundary maps ∂_1 and ∂_2. They are presented in matrix form:

$$\partial_1 = \begin{array}{c} \\ [v_0] \\ [v_1] \\ [v_2] \\ [v_3] \end{array} \begin{array}{c} \begin{array}{ccccc} [v_0, v_1] & [v_0, v_2] & [v_1, v_2] & [v_1, v_3] & [v_2, v_3] \end{array} \\ \begin{pmatrix} -1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & -1 & 0 \\ 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

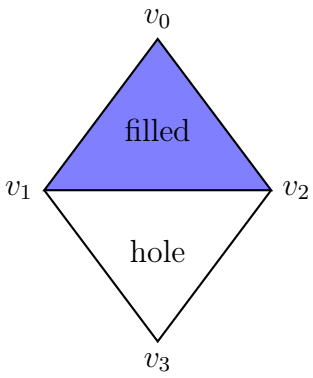


Figure 2.5: The Betti number counts the holes.

$$\partial_2 = \begin{array}{c} \\ [v_0,v_1] \\ [v_0,v_2] \\ [v_1,v_2] \\ [v_1,v_3] \\ [v_2,v_3] \end{array} \begin{array}{c} [v_0,v_1,v_2] \\ \left(\begin{array}{c} 1 \\ -1 \\ 1 \\ 0 \\ 0 \end{array} \right. \end{array}$$

The boundary map ∂_0 is the function that gives result 0 for all vertices. Then the dimension of its kernel is the number of vertices. The Betti numbers are:

$$\begin{aligned} \beta_0 &= \dim \ker \partial_0 - \dim \operatorname{im} \partial_1 \\ &= 4 - 3 \\ &= 1 \\ \beta_1 &= \dim \ker \partial_1 - \dim \operatorname{im} \partial_2 \\ &= 2 - 1 \\ &= 1 \end{aligned}$$

The number $\beta_0 = 1$ shows that the simplicial complex is connected, and $\beta_1 = 1$ shows that there is one empty hole.

2.2.3 Simplicial complex of wireless networks

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

As presented in [23], if the cover is good, that is if the cover sets and all nonempty finite intersections of cover sets are contractible, then the Čech

complex captures the topology of the cover. We should notice that, the union of convex sets provides the good cover.

Theorem 1 (Čech theorem) *The Čech complex of a good cover has the homotopy type of the union of the cover sets.*

Because the Čech complex captures perfectly the coverage of a domain, we use it to present the wireless networks in this thesis.

Definition 2 (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 an abstract simplicial complex whose k-simplices correspond to unordered $(k+1)$-tuples of point in $\mathcal{V}$ which are pairwise within distance ϵ of each other.*

The Rips gives only an approximation of coverage and may miss coverage holes [66]. Therefore, the Rips complex is not appropriate to be a representation of wireless networks.

Chapter 3

Construction of the Čech complex of wireless networks

In this chapter, we introduce our algorithms to construct the Čech complex of a wireless network, in centralized, parallelized, and distributed versions. As the Čech complex provides the exact topology of the covering domains, these algorithms help in applications that manage and optimize the coverage of the network. The minimal dimensional Čech complex that satisfies requirements of these applications can be constructed in only polynomial time. This work has been published in [43].

3.1 Introduction

A wireless network may not only contain a great number of cells, but it may also comprise a wide variety of cells, such as pico cell, micro cell and macro cell. For this reason, network management could be untractable without access to the network topology.

As described in Chapter 2, the Čech complex has a strong advantage that it represents exactly the topology of the covering domains. The Čech complex represents a group of cells by a simplex if all of them have a non-empty intersection. The Rips complex represents a group of cells by a simplex if every two of them are neighbors. The Rips complex still describes the neighborhood relation between cells, therefore it sometimes represents inaccurately the topology of the network. See Figure 3.1 for an example. In this figure, there are three cells with a coverage hole inside them. This hole is represented by an empty triangle in the Čech representation. However, any two of these cells are neighbors so the Rips complex represents these cells by a filled triangle. It means that there is no coverage hole in the Rips rep-

resentation. The Čech complex detects successfully the coverage hole while the Rips complex does not.

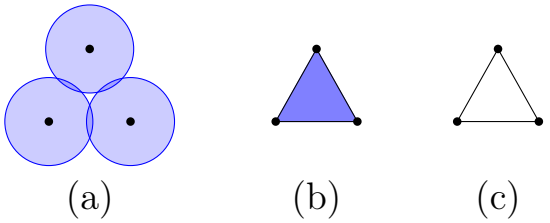


Figure 3.1: (a) Cells, (b) Rips complex, (c) Čech complex.

Recently, an algorithm to construct the Čech complex has been introduced in [20]. However, this algorithm aims for applications in graphic science which only consider same sized cover sets. As a result, this algorithm can not be applied to compute the topology of a wireless network, who is composed of different sized cells. The construction of the Čech complex of a wireless network is an interesting problem but it has not been solved.

In this chapter, we propose a centralized algorithm, as well as its parallelized and distributed versions, to construct the Čech complex of a wireless network. Our algorithms compute the minimal Čech complex that gives information about connectivity and coverage of the network in only polynomial time.

The rest of this chapter is organized as follows. Section 3.2 describes the network model. In Section 3.3, we present entirely the details about the construction of the Čech complex in centralized way. The parallelized and distributed versions of this construction are presented in the Section 3.4 and in the Section 3.5. The complexity of the centralized, parallelized and distributed constructions is discussed in Section 3.6. The simulation results are presented in Section 3.7. Finally, we discuss about the applications of our algorithms in the last section.

3.2 System model

We consider a wireless network composed of N distinct cells. We assume that each cell uses isotropic propagation. The coverage of the i-th cell is modeled as:

$$c_i(v_i, r_i) = \{x \in \mathbb{R}^2 : \|x - v_i\| \leq r_i\},$$

where $\|.\|$ is Euclidean distance, the vertex v_i presents the base station location and r_i is coverage radius of the i-th cell. Let $\mathcal{U}$ be the collection of

cells, then $\mathcal{U} = \{c_i, i = 0, 1, \ldots, (N-1)\}$. The Čech complex of $\mathcal{U}$, $\check{\mathbf{C}}(\mathcal{U})$, is defined as the Čech complex of the wireless network. As the coverage of each cell is a convex set, then the Čech complex of the wireless network $\check{\mathbf{C}}(\mathcal{U})$ describes exactly its topology (the Čech theorem, Chapter 2).

In the Čech complex, each vertex v_i, (a 0-simplex), corresponds to the i-th cell c_i in the network. An edge, (a 1-simplex), represents the connection, or the intersection, between two cells. Each k-simplex, where $k \geq 2$, represents the common intersection of the coverage of together $(k+1)$ corresponding cells of this simplex. For example, in the Figure 3.2, the 2-simplex $[v_2, v_3, v_6]$ means the overlap of the coverage inside cell c_2, cell c_3 and cell c_6. There is no coverage hole inside these cells. The higher dimensional simplex is, the higher overlap times is. The 3-simplex $[v_0, v_1, v_2, v_6]$ means the four corresponding cells: c_0, c_1, c_2 and c_6, together, have a common intersection. In contrast, a chain of 1-simplices indicates a coverage hole inside the corresponding cells of the chain. For example, the chain $[v_3, v_4] + [v_4, v_5] + [v_5, v_6] + [v_6, v_3]$ shows a coverage hole inside four cells c_3, c_4, c_5 and c_6. To analyze the network topology, we use characteristics of the homology of the Čech complex. For more details about homology, please see Chapter 2.

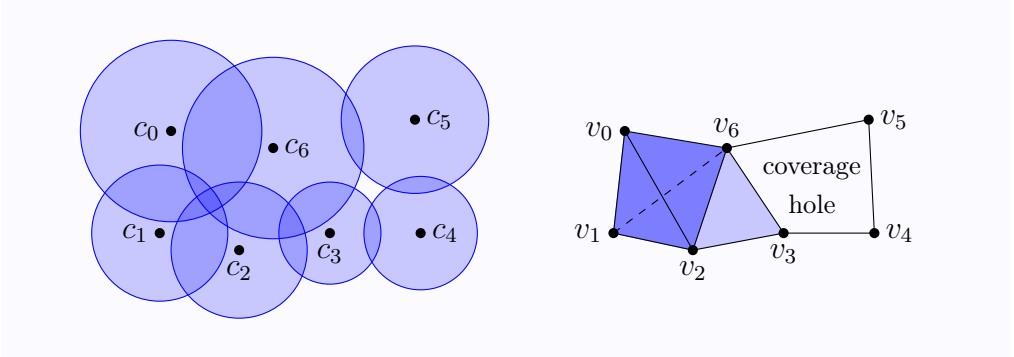


Figure 3.2: Cells and their Čech representation.

3.3 Construction of the Čech complex

As the definition of the Čech complex, each k-simplex represents the common intersection of together its $(k+1)$ corresponding cells. We denote $\mathbf{S}_k$ the collection of all k-simplices of the complex. As each vertex, that is a 0-simplex, corresponds to a cell of the network. The collection of 0-simplices, $\mathbf{S}_0$, is then obviously the list of corresponding vertices of cells.

$$\mathbf{S_0} = \{v_i \mid i = 0, 1, \ldots, (N-1)\}.$$

Each 1-simplex is an edge that connects two overlapping cells. In other words, it is a pair of two neighbor cells. The collection $\mathbf{S}_1$ of all 1-simplices is found as in the Algorithm 1:

Algorithm 1 Construction of 1-simplices

Input: $\mathbf{S_0}$
Output: $\mathbf{S_1}$
  for $i = 1$ **to** $N - 1$ **do**
    for $j = i + 1$ **to** N **do**
      if $c_i(v_i, r_i) \cap c_j(v_j, r_j) \neq \varnothing$ **then**
        add $[v_i, v_j]$ into $\mathbf{S_1}$;
      end if
    end for
  end for
  return $\mathbf{S_1}$;

When k is bigger than one, the procedure to find out the collection $\mathbf{S}_k$ of all k-simplices is more complex. For each combination of $(k + 1)$ cells in total N cells of the network, we need to verify if all these $(k + 1)$ cells have a common intersection. Let us denote $\hat{s}_k = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k]$ a combination of $(k + 1)$ cells in N cells of the network. If $\hat{s}_k$ is a k-simplex, then each pair $[\hat{v}_i, \hat{v}_j]$ is a 1-simplex, for all $0 \leq i < j \leq k$. So if there is a pair $[\hat{v}_i, \hat{v}_j]$ that is not a 1-simplex, then $\hat{s}_k$ is definitively not a k-simplex. So, only the combination $\hat{s}_k$ that each pair $[\hat{v}_i, \hat{v}_j]$, where $0 \leq i < j \leq k$, is a 1-simplex has a probability to be a k-simplex. This combination is called a candidate to be a k-simplex. We should firstly find out the collection $\hat{\mathbf{S}}_\mathbf{k}$ of all candidates to be a k-simplex, as in the Algorithm 2.

Algorithm 2 Searching for candidates to be a k-simplex

Input: $\mathbf{S_0}, \mathbf{S_1}$, and k;
Output: $\hat{\mathbf{S}}_\mathbf{k}$ the collection of candidates to be a k-simplex;

```
Ŝ_k = ∅;
for i = 0 to |S_0| − 1 do
  v̂_0 = v_i the i-th vertex in S_0;
  V = {v_j ∈ S_0 and j ≠ i | [v_i, v_j] ∈ S_1} the collection of neighbors of v_i;
  if |V| ≥ k then
    V̂ = collection of all combinations of k vertices in V;
    for t = 0 to |V̂| do
      [v̂_1, v̂_2, ..., v̂_k] = the t-th combination in V̂;
      if [v̂_m, v̂_n] ∈ S_1 for all m, n that 1 ≤ m < n ≤ k then
        add ŝ_k = [v̂_0, v̂_1, ..., v̂_k] into Ŝ_k;
      end if
    end for
  end if
end for
return S_k;
```

Let $\hat{s}_k = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k]$ be a candidate to be a k-simplex. The corresponding cells of this candidate are $\hat{c}_i(\hat{v}_i, \hat{r}_i)$, where $i = 0, 1, \ldots, k$. We now need to verify if all these corresponding cells have a common intersection.

Let us assume that $\hat{s}_k$ is a k-simplex. It means that there is a common intersection of all these corresponding cells. We denote $\mathbf{I}$ to be this intersection, we have:

$$\mathbf{I} = \cap \hat{c}_i, \text{for } i = 0, 1, \ldots, k;$$

Let p is a point that belongs to $\mathbf{I}$, then p must belong to all corresponding cells $\hat{c}_i$, where $i = 0, 1, \ldots, k$. We denote circle $\hat{b}_i$ the boundary, or the cover, of the cell $\hat{c}_i$, and $\mathbf{X}$ the collection of intersection points of every pair of these circles.

$$\mathbf{X} = \{\hat{b}_m \cap \hat{b}_n \mid 0 \leq m < n \leq k\}.$$

There are only two possible cases:

- The first case: $\mathbf{X} \cap \mathbf{I} = \varnothing$. There is no intersection point that belongs to $\mathbf{I}$. In this case, the smallest circle, $\hat{b}_{\min} = \min\{\hat{b}_i \mid i = 0, 1, \ldots, k\}$, must be inside all other circles $\hat{b}_i$ that $\hat{b}_i \neq \hat{b}_{\min}$. See an example in Figure 3.3.

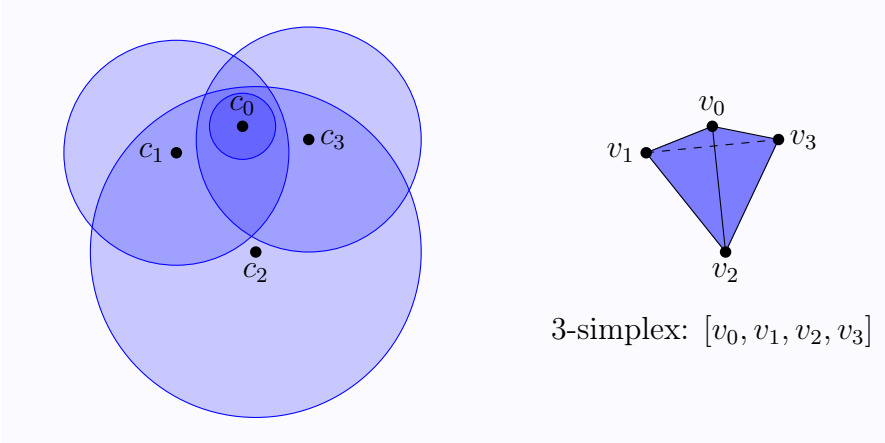


Figure 3.3: The smallest cell is inside the others.

- The second case: $\mathbf{X} \cap \mathbf{I} \neq \varnothing$. There must exist two circles $\hat{b}_m$ and $\hat{b}_n$ (the boundary of $\hat{c}_m$ and $\hat{c}_n$), that at least one of their intersection points belongs to all other corresponding cells $\hat{c}_i$, where $i \neq m$ and $i \neq n$. See an example in Figure 3.4.

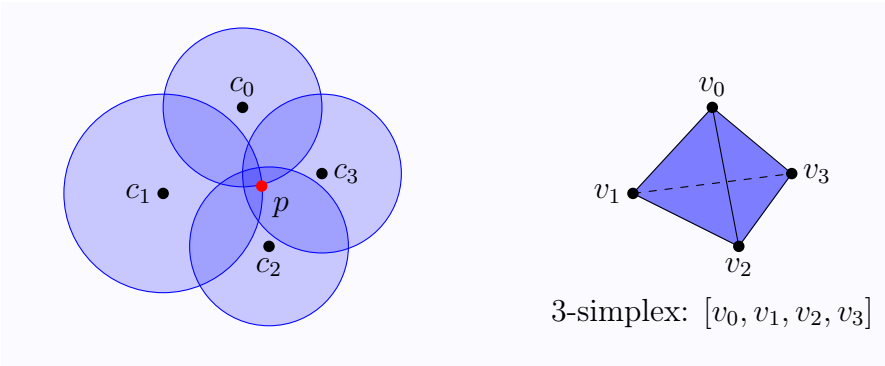


Figure 3.4: There is a pair of cells whose one intersection point (the red point) is inside the others.

So, to verify a candidate to be a k-simplex, we need to verify if it satisfies one of these two cases above. If this candidate does not satisfy both of these two cases above, then it is not a k-simplex. See an example in Figure 3.5.

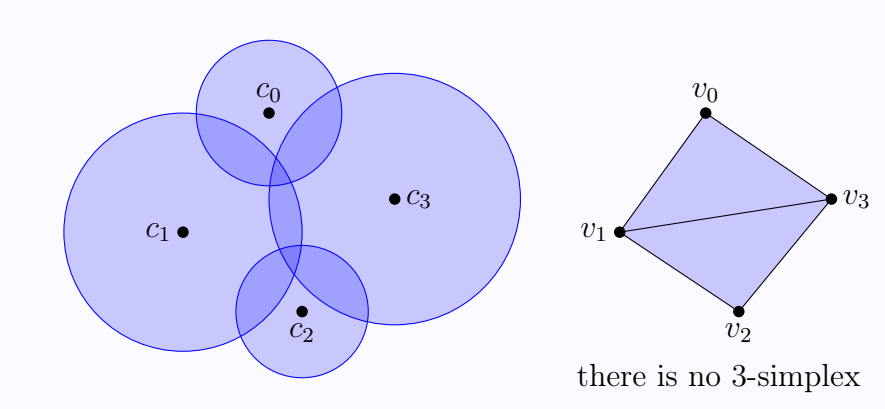


Figure 3.5: No intersection point of any pair of two cells is inside the others.

The procedure to verify a candidate, if it is confirmed to be a k-simplex, is summarized as in the Algorithm 3.

Algorithm 3 Verification of a candidate to be a k-simplex

Input: $\hat{s}_k = [\hat{v}_0, \hat{v}_1, \ldots, \hat{v}_k]$ a candidate to be a k-simplex;
Output: **true** if only if $\hat{s}_k$ is a k-simplex;
  $\hat{\mathcal{C}} = \{\hat{c}_i \mid \hat{c}_i(\hat{v}_i, \hat{r}_i) =$ corresponding cell of $\hat{v}_i\}$;
  $\hat{\mathcal{B}} = \{\hat{b}_i \mid \hat{b}_i =$ boundary of cell $\hat{c}_i\}$;
  verification = **false**;
  $i_{\min} = \arg\min(\hat{r}_i \mid i = 0, 1, \ldots, k)$;
  if $\hat{b}_{i_{\min}}$ is inside $\hat{b}_i$, $\forall i \neq i_{\min}$, $i = 0, 1, \ldots, k$ **then**
    verification = **true**;
  else
    for each pair $[\hat{b}_m, \hat{b}_n] \mid 0 \leq m < n \leq k$ **do**
      if $\exists p \in \{\hat{b}_m \cap \hat{b}_n\} \mid p \in \hat{c}_i, \forall i \neq m, n$ and $i = 0, 1, \ldots, k\}$ **then**
        verification = **true**;
        break;
      end if
    end for
  end if
  return verification;

The Algorithm 4 describes the full details about the construction of the Čech complex. This algorithm consecutively finds the collection of increasingly dimensional simplices. It begins with the slowest dimensional simplices, the $\mathbf{S}_0$, and stops at the highest dimensional simplices as it can not find any higher dimensional simplex more.

Algorithm 4 Construction of the Cech complex

Input: $\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
Output: $\check{\mathbf{C}}(\mathcal{U})$ the Čech complex of $\mathcal{U}$;

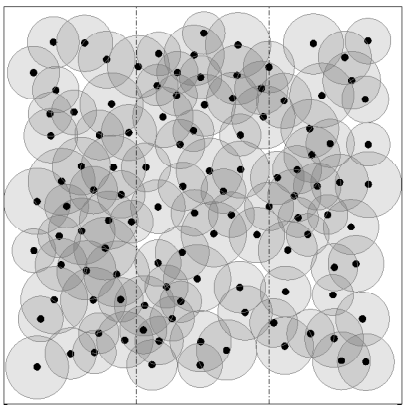
```
S_0 = {v_i | i = 0, 1, ..., (N − 1)} the collection of vertices;
S_1 = collection of 1-simplices (call to Algorithm 1);
while (1) do
    S_k = ∅ the collection of k-simplices;
    Ŝ_k = collection of candidates to be k-simplices (call to Algorithm 2);
    for ŝ_k ∈ S_k do
        if ŝ_k is confirmed to be a k-simplex (call to Algorithm 3) then
            add ŝ_k into S_k;
        end if
    end for
    if |S_k| ≠ 0 then
        k = k + 1;
    else
        break;
    end if
end while
Č(U) = {S_0, S_1, ..., S_k};
return Č(U);
```

3.4 Parallel construction of Čech complex

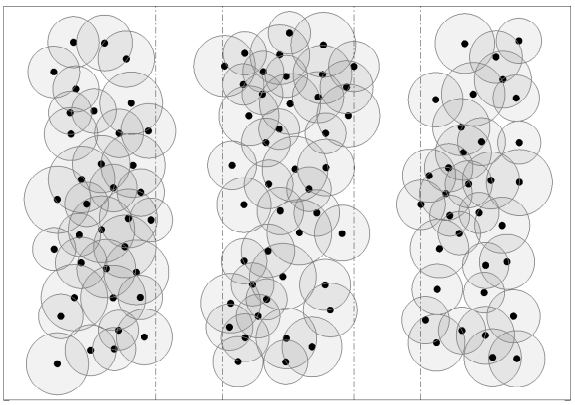
In this section, we introduce a parallel computing method to accelerate the speed of the Čech complex construction. To do so, we split the domain of the network into sub-domains. Then, the Čech complex of each sub-domain, the sub-Čech complex, is parallel computed on one of separate computers. Finally, all the sub-Čech complexes are integrated into one to receive the integral Čech complex of the whole network.

We demonstrate the network split in Figure 3.6. The original domain of the network is split by the vertical border lines, as in Figure 3.6(a). After the split, the cells of the network belong to separate sub-domains , as in Figure 3.6(b). However, a cell near the border line between its sub-domain and the adjacent sub-domain may be the neighbor of some cells that are on the adjacent sub-domain. Obviously, one cell among this cell and its neighbors must intersect with the border line. We call the cell that intersects with the border line a border cell. And, we call the connection between a border cell and its neighbors that are on the adjacent side of the border

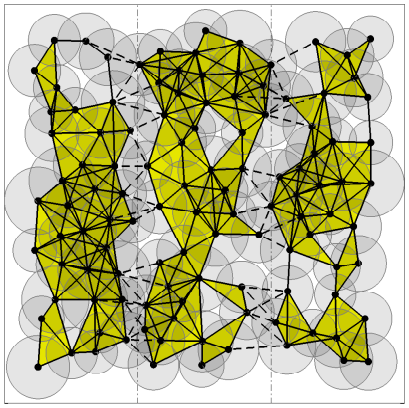
line a cross-border connection. The split of the network will break all the cross-border connections, as shown in Figure 3.6(c). After the split, these cross-border connections are missed, as shown in Figure 3.6(d).



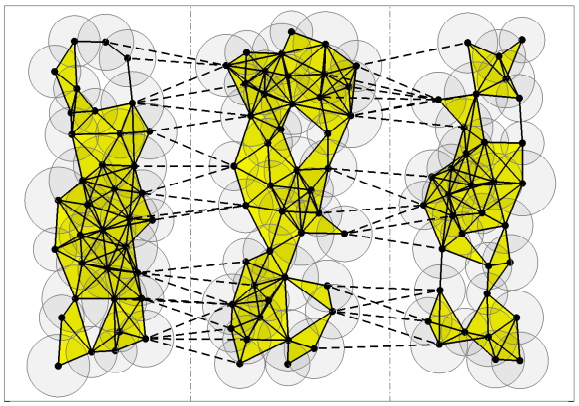
(a) The split of the network domain into sub-domains



(b) Cells in separated sub-domains of the network



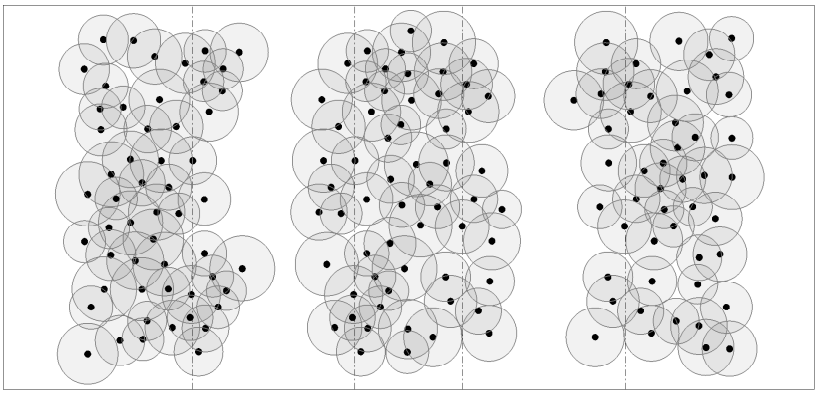
(c) Broken cross-border connections of cells around the border line



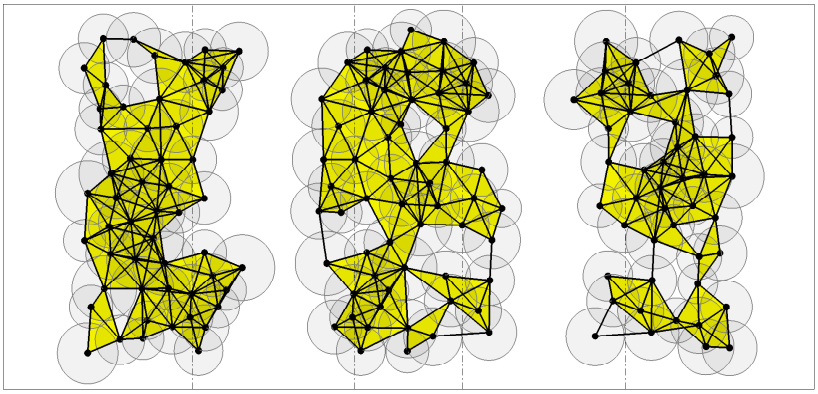
(d) Connection among cells in separated sub-domains after the split of network

Figure 3.6: Network split into sub-domains.

To conserve these cross-border connections, for each sub-domain, we need to find all the border cells and add their neighbors to the sub-domain, as in Figure 3.7(a). Then, the cross-border connections of these border cells are restored, as in Figure 3.7(b).



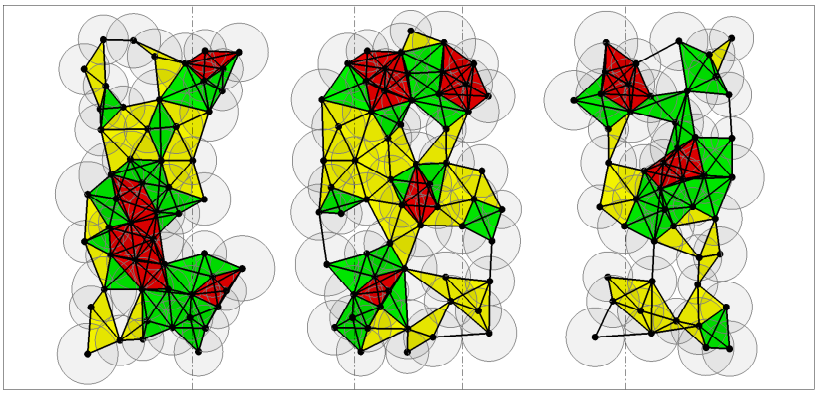
(a) Cells of sub-domains with added neighbors of the border cells



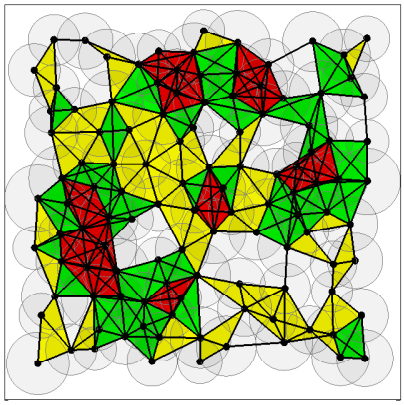
(b) Restored cross-border connections of the border cells

Figure 3.7: Connection of border cells restoration.

We now can compute parallel the sub-Čech complexes of the sub-domains (Figure 3.8(a)) and then integrate them to get the integral Čech complex of the whole domain (Figure 3.8(b)).



(a) The sub-Čech complexes of sub-domains



(b) The integrated Čech complex

Figure 3.8: The integration of sub-Čech complexes into one.

The parallel construction of the Čech complex is described in the Algorithm 5. In this algorithm, we denote $\mathcal{U}$ the domain of the network, which is the collection of cells, and m the number of available computers. The domain $\mathcal{U}$ is divided into m equal separated sub-domains $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_m$ by the vertical border lines. The sub-Čech complexes of these sub-domains are $\check{\mathbf{C}}_1, \check{\mathbf{C}}_2, \ldots, \check{\mathbf{C}}_m$ and the integral Čech complex is $\check{\mathbf{C}}$, the output of the algorithm. In this algorithm, the "parfor" means the loop that its iterations can be done parallel.

Algorithm 5 Parallel construction of the Čech complex

Input: $\mathcal{U}$, m;
Output: $\check{\mathbf{C}}(\mathcal{U})$ the Čech complex of $\mathcal{U}$;
split $\mathcal{U}$ into sub-domains $\mathcal{U}_1, \mathcal{U}_2, \ldots, \mathcal{U}_m$;
parfor $i = 1$ **to** m **do**
$\mathcal{U}_i^* = \mathcal{U}_i$;
for each border line $\triangle$ of $\mathcal{U}_i$ **do**
for each cell c in $\mathcal{U}_i$ **do**
if the cell c and the border line $\triangle$ are intersected **then**
add all neighbors of the cell c into $\mathcal{U}_i^*$
end if
end for
end for
$\check{\mathbf{C}}_i(\mathcal{U}_i)$ = the Čech complex of $\mathcal{U}_i^*$ (call to the Algorithm 4);
end parfor
$\check{\mathbf{C}}(\mathcal{U}) = \cup\check{\mathbf{C}}_i(\mathcal{U}_i)$ for $i = 1, 2, \ldots, m$;
return $\check{\mathbf{C}}(\mathcal{U})$;

3.5 Distributed construction of the Čech complex

We assume that each cell c_i can communicate with other cells over radio within a distance $d_i = 2r_i$. We assume that there are enough frequency slots for cells to communicate over radio without collision. Every cell is also connected by a backhaul network. At the initial state, each cell broadcasts a ping message with its position and its radius over radio channel. If a cell receives a ping message, it verifies if the cell that sent this ping message is a neighbor. If they are neighbors, then the cell that received the ping message sends a relationship confirmation together with its position and its radius to the cell that sent the ping message by using the backhaul network. After receiving the confirmation, the cell that sent the ping message adds the cell that sent the confirmation into its collection of neighbors. We assume that all the cells can reply the confirmation within a period t_{ack}. After this period t_{ack}, every cell detects its collection of neighbors. We denote the collection of neighbors of the cell c_i as $\mathbf{N}_i$.

When the collection of neighbors is available, each cell computes its simplices by verifying the intersection of its local region, which is comprised of this cell and its neighbors. As each pair of neighbors forms a 1-simplex, the collection of 1-simplices of each cell is easily found. To find all k-simplices,

where $k \geq 2$, each cell verifies if a group of it and its k neighbors has a common intersection. If these cells have a common intersection, then this group forms a k-simplex. However, neighborhood is a two-way relationship. Therefore, the verification of intersection could be duplicated by different cells that are neighbors. To avoid the redundant duplication, each cell verifies the intersection by following a right hand rule. This rule is that each cell verifies only with neighbors that are on its right hand side. If there is a neighbor which has the same horizontal coordinate, it only verifies this neighbor if it has higher vertical coordinate. After computing its local complex, it removes all the simplices that are faces of any simplex in its local complex. For each simplex that is not a face of any simplex in its local complex, it transmits this simplex to every cell that belongs to this simplex. As a result, every cell detects all its simplices. For example, in Figure 3.2, the cell c_2 verifies the intersection with only the cell c_3 and c_6. It detects the simplex $[v_2, v_3, v_6]$. It receives its other simplices from the neighbor c_0 and c_1. The Algorithm 6 reports the distributed computation of Čech complex for each cell. We denote the cell that is computing the simplices as c_i. The highest dimension of simplices that are considered is $\dim_{\max}$. The output of this algorithm, the collection of simplices of the cell c_i is denoted as $\check{\mathbf{C}}_i$.

The global Čech complex that represents the topology of the whole network is sometimes needed. There should be a master cell that controls the topology of the network. This global Čech complex can be easily built by integrating the simplices computed from every cell. Each cell sends its computed simplices that contain only the vertices satisfied the right hand rule. One more time, this rule is useful, it avoids sending the duplicated simplices.

Algorithm 6 Distributed computation of Čech complex

Input: c_i the cell that is computing;
Output: $\check{\mathbf{C}}_i$ the collection of simplices of the cell c_i;

```
broadcast {ping, v_i, r_i} over radio;
S_0 = v_i;
while count time < t_ack do
  if {ping, v_j, r_j} is received from c_j then
    if d(v_i, v_j) < r_i + r_j then
      add v_j to N_i;
      if x_i < x_j then
        add v_j to S_0;
        add [v_i, v_j] to S_1;
      end if
      if x_i == x_j and y_i < y_j then
        add v_j to S_0;
        add [v_i, v_j] to S_1;
      end if
    end if
  end if
end while
for k = 2 to dim_max do
  for l = 1 to C^k_{|S_0|} do
    ŝ = l-th combination of k vertices in S_0/{v_i};
    ŝ = ŝ ∪ {v_i};
    ĉ = corresponding cells of ŝ;
    if intersection of all cells in ĉ is not empty then
      add ŝ to S_k;
    end if
  end for
end for
Čech_i = {S_0, S_1, ..., S_k};
send Čech_i to every corresponding cell of vertices in S_0;
n_i = |N_i|;
while n_i > 0 do
  if Čech_j is received from c_j in N_i/S_0 then
    Čech_i = Čech_i ∪ Čech_j;
    n_i = n_i − 1;
  end if
end while
return Čech_i;
```

3.6 Complexity

We denote $T(k)$ the complexity to construct all k-simplices. The complexity to construct the Čech complex is $T(\check{\mathbf{C}}, d)$ that:

$$T(\check{\mathbf{C}}, d) = \sum T(k), \text{ for } k = 0, 1, \ldots, d,$$

where d is the highest dimension of the Čech complex that we consider.

If we only want to know the connectivity of the network, then only the 0-th Betti number β_0 is needed. So, we only need to compute the Čech complex up to dimension one, $(d = 1)$. The collection of 0-simplices is obviously the collection of N cells, then its complexity is N. Computing 1-simplices is to find neighbors of each cell. Its complexity is C_N^2, where N is the number of cells. The complexity to construct the Čech complex that gives information about the connectivity is $T(\check{\mathbf{C}}, 1)$ as:

$$\begin{aligned} T(\check{\mathbf{C}}, 1) &= T(0) + T(1) \\ &= N + C_N^2 \\ &= \mathcal{O}(N^2) \end{aligned}$$

To get information about the coverage of the network, we need to compute the 1-st Betti number β_1. So, the Čech complex need to be built up to dimension two, or higher, $(d \geq 2)$.

Each candidate is a combination of $(k + 1)$ cells that each cell in this combination is neighbor of the others. Let n be the average number of neighbors of each cell. For each cell, the number of candidates that contain this cell is at most C_n^k in average. So, there are about NC_n^k candidates to be k-simplices. The complexity to find all k-simplices typically is NC_n^k.

The complexity to construct the Čech complex built up to dimension d is:

$$T(\check{\mathbf{C}}, d) = N + C_N^2 + N \sum_{k=2}^{d} C_n^k.$$

With $d = 2$, the Čech complex is the minimal one that gives information about coverage. Its construction has complexity of $\mathcal{O}(N^2 + Nn^2)$. It means that the minimal Čech complex can be computed in polynomial time.

If we want more information about network topology, we compute the Čech complex up to a higher dimension. Let we assume that $d = \infty$, the sum $\sum_{k=2}^{d} C_n^k$ can be upper bounded by 2^n. The complexity to construct the Čech complex in this case is then as much $\mathcal{O}(N^2 + N2^n)$. Although it contains an exponential term that is 2^n, it should not be a significant number. It is

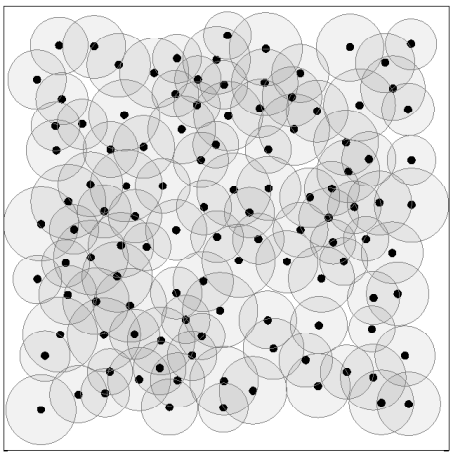
because the average number of neighbors of each cell n is normally much smaller than the number of cells in the network N.

If the Čech complex is parallel computed on several computers, the complexity of each sub-Čech complex computation is much smaller than the centralized one. Let m be the number of available computers. The network should be split into m sub-domains, and each sub-domain has N/m cells in average. The Čech complex of each sub-domain is computed on one of separate computers. If the Čech complex is computed up to dimension two, the complexity of the computation on each computer is $\mathcal{O}(N^2/m^2 + Nn^2/m)$. It is typically m^2 times smaller than the complexity of the centralized computation, because the number of cells N is normally much bigger than the average number of neighbors of each cell n. If the Čech complex is computed up to its highest dimension, the complexity of the computation on each computer is as much $\mathcal{O}(N^2/m^2 + N2^n/m)$. If the cells are deployed with a high density, the average number of neighbors of each cell, n, may be also high. So, the dominate term is now 2^n. In this case, the complexity of the computation of each computer is only m times smaller than the complexity of the centralized computation.

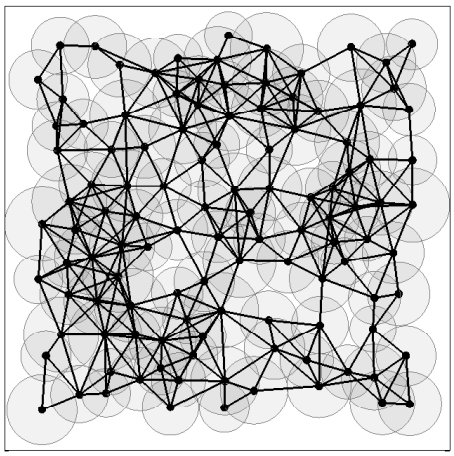
The distributed construction of the Čech complex is done on each cell. Each cell computes the Čech complex for only its local region. The number of cells in this region is $(n+1)$ in average. So, the complexity of the distributed construction of the Čech complex up to dimension 2 and its highest dimension on each cell are $\mathcal{O}(n^3)$ and $\mathcal{O}(n2^n)$, respectively.

3.7 Simulation results

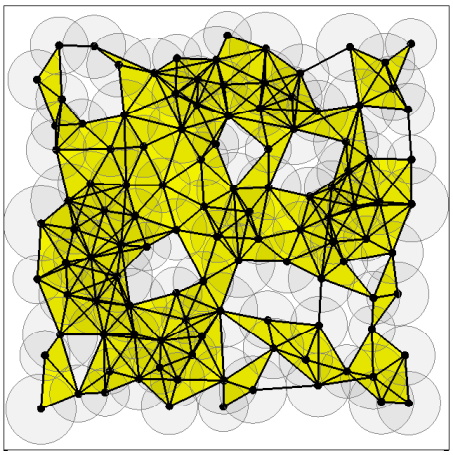
First of all, we want to show an example of the construction of the Čech complex as in Figure 3.9. In Figure 3.9(a), we draw a wireless network that contains many different sized cells. The cells are randomly distributed according to the Poisson point process with the density of cells $\lambda = 5$ on the space $[5 \times 5]$. The radius of each cell can vary from 0.1 to 0.5. The covering domain of each cell is drawn in gray, while its cover is drawn as a circle in black. The white region shows a coverage hole. The darker gray region is, the more overlapping region is. The center of each cell is marked as a black point. In Figure 3.9(b), we show the collection of 1-simplices that is the connectivity map. Each pair of two neighbor cells is represented by an edge in black. All cells are connected. In Figure 3.9(c), we add 2-simplices into the map. Each 2-simplex is represented by a yellow triangle. The empty region is still not colored.



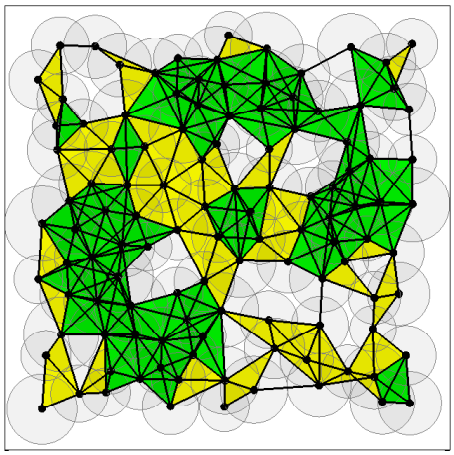
(a) Wireless Network



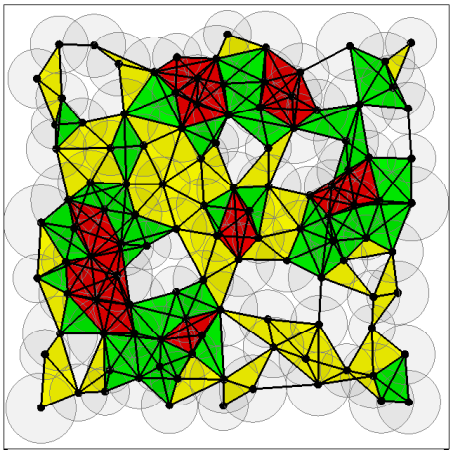
(b) 1-simplices (connectivity map)



(c) 2-simplices (coverage map in yellow color)



(d) 3-simplices (high overlapping region in green color)



(e) 4-simplices (extremely high overlapping region in red color)

Figure 3.9: The construction of the Čech complex of a wireless network.

There are totally 12 empty holes, as shown not only on the figure, but also on the computer, as our computation of the 1-st Betti number shows that $\beta_1 = 12$. In Figure 3.9(d), we show 3-simplices in green. The green region represents the common intersection of 4 cells. The highest dimensional simplices are drawn in Figure 3.9(e). Each 4-simplex is shown in red, it presents the most overlapped region.

To evaluate the performance of the algorithms to construct the Čech complex, we did some simulations of the construction in all centralized, parallelized and distributed ways. We randomly deploy cells according to the Poisson point process on a square of size $[30 \times 5]$. The density of cell is varied from 1 (medium) to 2 (high). The radius of each cell is a random variable from 0.5 to 1. We use our algorithms to construct the Čech complex up to dimension 2 and up to its highest dimension $d_{\max}$. The parallel computation is done on three separate computers. The simulation is repeated 1000 times. The average construction time in each case is noted in the Table 3.1. We note that, the Čech complex built up to dimension 2 is the minimal one that give information about coverage.

Table 3.1: Construction time (second) of the Čech complex

Density of cells	Centralized computation		Parallel computation		Distributed computation	
	$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$
1.0	0.144	0.526	0.033	0.235	0.005	0.053
1.5	0.946	11.825	0.206	5.674	0.006	1.433
2.0	4.088	218.290	0.885	124.452	0.007	26.479

As we can observe, the construction time increases quickly when we increase the density of cells. Because, both the number of cells N and the average number of neighbors of each cell n increase with the density of cells. In addition, the construction time increases exponentially with n as shown in Section 3.6. The parallel and distributed algorithms have the much smaller computation time than the centralized one, due to their lower complexity of computation.

The number of transmissions that are sent by each cell, as well as the size of each transmission are important to evaluate the performance of the distributed construction of the Cech complex. The message ping is sent only one time by each cell. If a cell receives a ping message, and it detects that the sender is a neighbor, then it sends an acknowledgment message. This message has a constant size, it contains only its id, its position and its radius of coverage. The size of this message is small, so we consider only the number of acknowledge messages (ACKs) that are sent by a cell. After the

construction of the local complex, each cell sends its list of simplices following the right hand rules. This list may be long and its size is not constant. We write this list to a text file, where each cell's vertex in a simplex is separated by a space and each simplex in the list is separated by a comma. The size of each character in this text is one byte. See Table 3.2 for the evaluation of the number of transmissions of each cell and the size of each transmission (in bytes).

Table 3.2: Performance of the distrbuted construction of the Čech complex

Density of cells	Number of ACKs	Number of transmissions		Size of a transmission (bytes)	
		$d = 2$	$d = d_{\max}$	$d = 2$	$d = d_{\max}$
1.0	5.52	3.05	3.03	23.59	20.14
1.5	8.32	4.59	4.55	38.82	48.61
2.0	11.01	6.07	6.10	45.43	170.01

To construct the minimal Čech complex, which gives information about coverage and connectivity of the network, each cell needs only about 6 transmissions at the highest density of cells. If the Čech complex is built up to its highest dimension, the number of transmission is increased but it is not a big number.

3.8 Conclusion and Discussion

We have proposed a centralized algorithm as well as its parallized and distributed versions to construct the Čech complex of a wireless network. The minimal Čech complex that gives information about connectivity and coverage of the network can be computed in only polynomial time. These algorithms are well computable tools to analyze the topology of the wireless network. Although these algorithms only consider the coverage in 2D, it is possible to develop their extended versions to compute the Čech complex of cover sets in 3D, as Wireless Sensor Network is one example.

Chapter 4

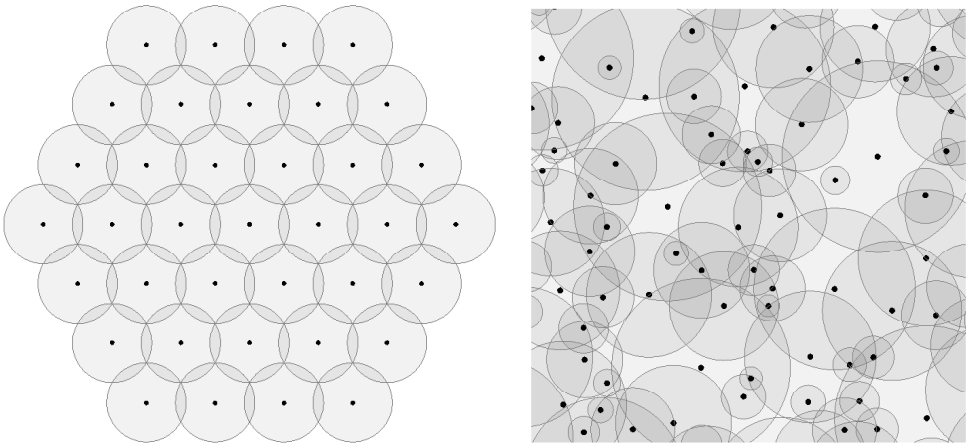
Simplicial homology based algorithms for energy saving in wireless networks

Energy saving is one of the most investigated problems in wireless networks. In this chapter, we propose algorithms to save energy in wireless networks by minimizing overlap regions between cells which mainly cause the wastage of transmission power. Our algorithms are based on simplicial homology which gives information about coverage of the network by only using matrix computations. Our simulations show that, our algorithms can save at most 65% of system's maximal transmission power in polynomial time. The probability density function (pdf) of optimized cell's radii is analyzed and discussed. This work has been published in [44].

4.1 Introduction

Traditionally, cells are deployed according to the hexagonal configuration, as in Figure 4.1(a). As this configuration is optimized for same sized cells, it is not always convenient. The next generation of wireless networks will be heterogeneous one, which comprises several types of cells such as macro cells, micro cells, and pico cells. While macro cells provide services in a large range, the micro cells and pico cells support users in only local areas, for example, inside buildings, campus, and parks. The heterogeneous network should consider cells as a random deployment, as in Figure 4.1(b).

In this chapter, we maximize the coverage and minimize the overlap regions of the network at the same time. We propose two heuristic algorithms: simulated annealing one and downhill one. The simulated annealing algo-



(a) Hexagonal deployment (b) Random deployment

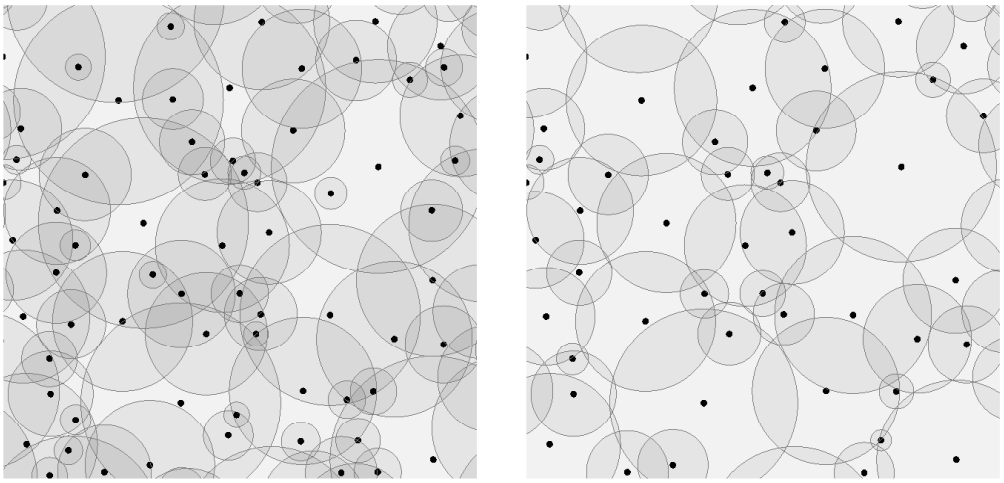
Figure 4.1: Wireless network deployments

rithm finds the approximation of the global optimized size of cells. The downhill algorithm still gives the local optimized size of cells. However, our simulations show that the local optimized solution approaches closely the global optimized one. We also propose some improvements to these algorithms to reduce its execution time, as well as to be executed on parallel or distributed computers. One example of simulated cells before and after the optimization is shown in Figure 4.2. The probability density function (pdf) of optimized cell's radii is also analyzed and discussed.

The rest of this chapter is organized as follows. Section 4.2 introduces our system model. Section 4.3 is devoted to the description of our algorithms. In the next section, we propose some improvements to these algorithms as well as the parallel and distributed versions. In Section 4.5, we discuss the complexity of these algorithms. The simulations and results are described in Section 4.6. In the last section, we conclude the chapter and propose some future work directions.

4.2 System model

We consider a wireless network whose cells are randomly deployed on the plane. We assume that each cell uses isotropic propagation. It supports users within a coverage radius from its base station. Let c_i denote the i-th cell of the network, v_i denote its base station position and r_i denote its coverage radius. Let $p_{t,i}$ the transmission power of the i-th cell and $p_{r,i,u}$ the received power of a mobile user u in this cell. According to the simplified



(a) Before optimization. (b) After optimization.

Figure 4.2: Cells before and after optimization.

path loss model [6], we have:

$$p_{r,i,u} = p_{t,i} K_0 \left[d_0/d\right]^{\gamma}, \tag{4.1}$$

where K_0 is a constant factor, d is the distance from the base station to the position of the mobile user u, d_0 is a reference distance, and γ is the path loss exponent factor of the radio environment. The received power of the user in this cell must be bigger than its sensitivity p_s, so we have:

$$p_{t,i} K_0 \left[d_0/d\right]^{\gamma} \geq p_s. \tag{4.2}$$

The distance from the base station to the position of a mobile user is upper bounded as:

$$d \leq \sqrt[\gamma]{p_{t,i} K_0 d_0^{\gamma}/p_s} \tag{4.3}$$

The upper bound of d is defined the coverage radius of the i-th cell r_i. We have:

$$r_i = \sqrt[\gamma]{p_{t,i} K_0 d_0^{\gamma}/p_s} \tag{4.4}$$

The relation between its transmission power p_i and its coverage radius r_i is:

$$p_{t,i} = K_0' r_i^{\gamma}, \tag{4.5}$$

where $K_0' = p_s/\left(K_0 d_0^{\gamma}\right)$. In this chapter, we assume that $K_0' = 1$ for simplification. The relation between the transmission power and the coverage radius of a cell is rewritten as:

$$p_{t,i} = r_i^{\gamma}, \tag{4.6}$$

The total transmission power of the system is:

$$P = \sum_{i=0}^{N-1} r_i^{\gamma}, \tag{4.7}$$

where N is the number of cells in the network.

Let $p_{i,\max}$ and $p_{i,\min}$ be the maximum and minimum transmission power, respectively, of the i-th cell. The maximum coverage radius $r_{i,\max}$ and the minimum coverage radius $r_{i,\min}$ of the i-th cell are estimated by Equation 4.6. Depending on the type of installed base station such as macro, micro and pico, the range of transmission power $[p_{i,\min}, p_{i,\max}]$ and the range of coverage radius $[r_{i,\min}, r_{i,\max}]$ of each cell may be different from another. Each cell can adjust its transmission power within the range to reduce or increase its coverage radius an amount Δr_i.

We use the Čech complex to represent the topology of the wireless network. In this Čech complex, each cell is represented by a vertex at its base station position. Two overlapping cells, which are neighbor cells, are represented by an edge, a 1-simplex, connecting two corresponding vertices. Three overlapping cells are represented by a 2-simplex, which is a filled triangle. Four overlapping cells are represented by a 3-simplex, which is a filled tetrahedron. And so on, $(k+1)$ overlapping cells are represented by a k-simplex. The coverage hole is the empty space inside a chain of 1-simplices.

Once the Čech complex of the wireless network is computed, we can get information about its connectivity and its coverage. The zero-dimensional Betti number β_0 of the Čech complex counts the connected component of the network. If all cells of the network are connected, the β_0 will be one. The one-dimensional Betti number β_1 of the Čech complex counts the coverage holes in the network. If there is no hole, the β_1 will be zero.

Definition 3 (Index of a vertex) *Given X an abstract simplicial complex, the index of a vertex v in X is the biggest integer k such that for every $i \leq k$ each $(i-1)$-simplex in X that contains v is a face of at least one i-simplex in X that contains v.*

The index of a vertex tells how many times its corresponding cell overlaps with its neighbors. The index zero tells that the corresponding cell is isolated. It is not connected to any cell. The index one tells that the corresponding cell is connected with neighbors by only edge. The index two tells that the corresponding cell is connected with neighbors by at least a two dimensional simplex. We denote $\hat{i}_c$ the index of the cell c.

The Figure 4.3 shows a wireless network, its Čech complex and its coverage properties. The common intersection of four cells: cell c_0, cell c_1, cell

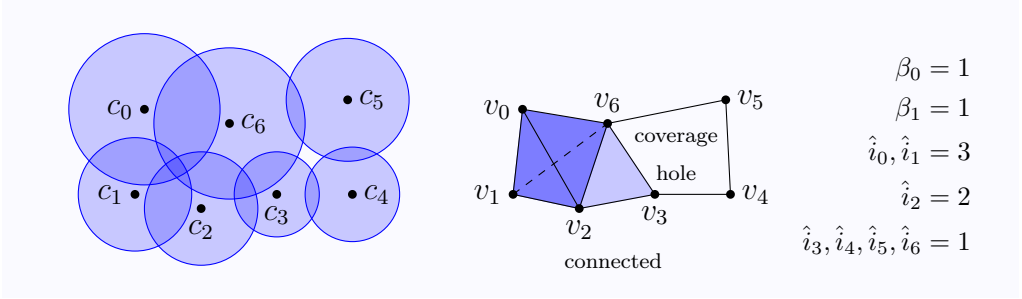


Figure 4.3: Analysis of the coverage of the network by using Čech representation.

c_2 and cell c_6 is represented by a 3-simplex, which is a filled tetrahedron, $[v_0, v_1, v_2, v_6]$. The common intersection of three cells: cell c_2, cell c_3 and cell c_6 is represented by a 2-simplex, which is a filled triangle. The coverage hole inside four cells: cell c_3, cell c_4, cell c_5 and cell c_6 is represented by the empty space inside the chain of 1-simplices: $[v_3, v_4] + [v_4, v_5] + [v_5, v_6] + [v_6, v_1]$. The zero-dimensional Betti number β_0 is one, this indicates that the network is connected. The one-dimensional Betti number β_1, which is one, indicates that there is one coverage hole in the network. The index of cell c_0 and cell c_1 are three. This means that cell c_0 and cell c_1 always have common intersection with three neighbors. The index of cell c_2 is two, it means that cell c_2 always has common intersection with at least two neighbors. The index of cell c_3, c_4, c_5 and c_6 are one. This means that each one in these cells doesn't have common intersection with every two of its neighbors.

4.3 Energy saving algorithms

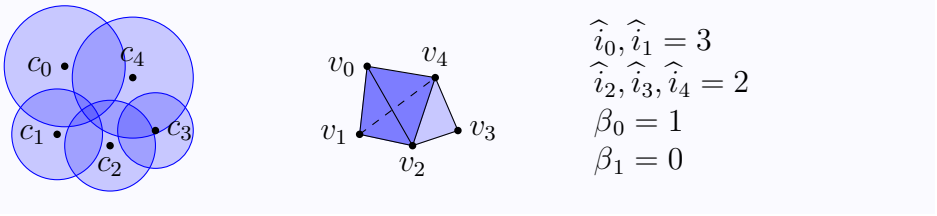
Consider a wireless network, we maximize its coverage and minimize its total transmission power at the same time. Firstly, we ensure a maximal coverage for the network. Each cell is turned on and is set to work with the highest transmission power. At this initial state, the network has the largest coverage. However, many cells are hardly overlapped. The overlap region between cells causes the wastage of transmission power due to interference. We can optimize the transmission power by minimizing the overlap region. However, the global coverage of the network should be conserved. In other words, the two Betti numbers β_0 and β_1 of the Čech complex of the network should be unmodified. The optimization problem can be written as:

$$\begin{aligned} \min_{r} \quad & \sum_{i=0}^{N-1} r_i^{\gamma} \\ \text{s.t.} \quad & \beta_0 = \beta_0^* \\ & \beta_1 = \beta_1^* \\ & r = (r_0, r_1, \ldots, r_{N-1}), \end{aligned} \tag{4.8}$$

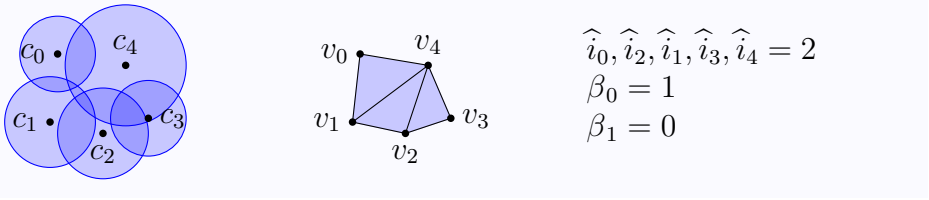
where β_0^* and β_1^* are the Betti numbers of the Čech complex of the network at the initial state where every cell is working with the maximal transmission power.

We propose two algorithms to optimize the transmission power of the network: the simulated annealing algorithm and the downhill algorithm. In the optimization process, a cell can be enlarged or decreased. We assume that all the existed coverage holes, the boundary cells and the fenced cells of the network are already known. We only modify the coverage radius of the internal cells that are not boundary or fenced. The enlargement of a cell does not create any coverage hole. However, a decrease of cell may create a new one. We need to verify if there is a new coverage hole after a decrease of cell. To do that, we recompute the Čech complex and its two Betti numbers β_0 and β_1. If two these numbers are the same as the numbers β_0^* and β_1^* at the initial state, there is no new coverage hole and the connectivity is kept. Then, the decrease is accepted. Conversely, if one of the Betti numbers β_0 and β_1 is changed, the decrease is refused.

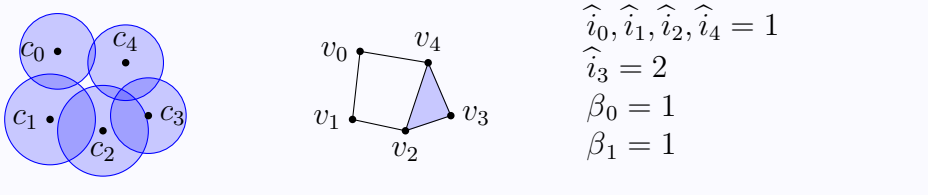
We can see an example of the global coverage verification in Figure 4.4. This figure represents the internal cells of a network. The external area of these cells are covered by fenced cells. To emphasis the verification if a new coverage hole appears, we do not draw the fenced cells here. In Figure 4.4(a), all cells are at their initial stage. Each cell reaches its maximal range. These cells are hardly overlapped. They are represented by one tetrahedron and one triangle. The Betti numbers are $\beta_0 = 1$ and $\beta_1 = 0$. They indicate that all cells are joined together in one component and there is no hole. Cells c_0 and c_1 have index three. In Figure 4.4(b), after the reduction of radius of cell c_0, the Čech complex now includes three filled triangles. The Betti numbers β_0 and β_1 are not changed but the indices of cell c_0 and cell c_1 are now reduced to two. This means that the overlapping level is reduced and no hole appears. The algorithm accepts the reduction of cell c_0. But, in Figure 4.4(c), cell c_4 tries to reduce its radius and creates a new coverage hole. This hole is represented by an empty rectangle and the first Betti number β_1 is now changed to one. The indices of cell c_0, cell c_1, cell c_2, cell c_4 are all reduced to one because these cells are now on the border of an empty hole.



(a) Initial state, each cell reach its own maximal coverage radius.



(b) First reduction, the cell c_0 is accepted to reduce its radius.



(c) Second reduction, the cell c_4 is refused to reduce its radius.

Figure 4.4: Reduction of cell radius and Čech representation.

Then, the reduction of cell c_4 is refused. Its radius is returned to its previous value.

4.3.1 Simulated annealing algorithm

Simulated annealing (SA) is a heuristic method to find an approximation of the global optimum of a cost function that may have several local optima. It simulates the annealing process in metal work. A solid is melted at a high temperature and then is slowly cooled until the frozen temperature. The structure at the frozen temperature has the minimum energy configuration. For more details about the SA algorithm, see papers [15, 41].

Starting with an initial solution, where all cells are set to their maximal radius, our SA algorithm finds the approximation of global optimized solution by following the cooling schedule: $T_k = T_0\alpha^k$. In this function, T_0 is the initial temperature, and α, the cooling factor, is a real positive number such that

$0 < \alpha < 1$. The cooling factor guarantees a smooth cooling schedule. At each temperature T_k, it randomly chooses a cell c and attempts to increase or decrease its radius by an amount of Δr_c. Then, the difference of power consumption is calculated by $\Delta P = (r_c \pm \Delta r_c)^\gamma - r_c^\gamma$, where γ is the path loss exponent. If the radius is decreased, the transition is called a downhill move. This downhill move is only accepted if no hole appears. In the SA algorithm, the index of cell is not considered. The algorithm only verifies the connectivity and coverage of the network after a reduction of coverage radius of a cell. So, the Čech complex with dimension two is enough to satisfy the requirements. If the radius is increased, the transition is called an uphill move. In this case, $\Delta P > 0$, this uphill move is accepted with probability $\exp(-\Delta P/T_k)$. Thanks to uphill moves, the process can jump out from a local minimum to search for the global minimum. This process is repeated L times at this temperature T_k to get the thermodynamic equilibrium state. The number L is called the length of the temperature T_k. After each temperature length, the temperature is gradually decreased by the cooling schedule $T_k = T_0\alpha^k$, for $k = 0$ to K. The initial temperature T_0 is chosen large enough to make the probability of an uphill move at initial state to be close to one. The number of schedule steps K is chosen large enough to make the probability of accepting an uphill move to be near zero at the final temperature T_k. The final configuration of radius is then close to the global minimum solution. Our SA algorithm is summarized in the Algorithm 7.

Algorithm 7 Simulated annealing algorithm

Input: $\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
Output: $\mathcal{U}_{\text{opt}}$ the collection of optimized cells;

```
turn on and maximize the radius for each cell in U;
compute Č(U) the Čech complex of U;
compute β0* and β1* of Č(U);
for k = 1 → K do
  T = α^k T0;
  for l = 1 → L do
    c = a random cell ∈ U;
    sign = a random value ∈ {−1; 1};
    if sign = −1 then
      r_c = r_c − Δr_c;
      compute Č(U) the Čech complex of U;
      compute β0 and β1 of Č(U);
      if β0 ≠ β0* or β1 ≠ β1* then
        r_c = r_c + Δr_c;
      end if
    else
      compute ΔP = (r_c + Δr_c)^γ − r_c^γ;
      r_c = r_c + Δr_c with probability exp(−ΔP/T);
    end if
  end for
end for
return U;
```

4.3.2 Downhill algorithm

In the downhill algorithm, only the reductions of cells' radius are accepted so it only gives the local optimal solution. This algorithm concerns the overlapping level of cells. The overlapping level of each cell can be known by computing the index of the corresponding vertex in the Čech complex.

We firstly consider the computation of the index of a vertex in a simplicial complex. Let X be an abstract simplicial complex and v be a vertex in X. From the definition of the index of a vertex, we verify if each k-simplex of v is a face of at least one $(k+1)$-simplex of v. Starting at $k = 0$, it increases k by one if all k-simplices are verified. The algorithm stops if there is a k-simplex of v which is not a face of any $(k+1)$-simplex. The highest value of k is the index of v. The index computation for a vertex v is described in Algorithm 8.

Let the index of a vertex be also the index of the corresponding cell.

Algorithm 8 Index computation for a vertex v

Input: $\check{\mathbf{C}}$ the Čech complex;
v the vertex to compute the index;
Output: index of every vertex in $\check{\mathbf{C}}$;

```
k = 1;
while exist k-simplex of v do
  S_k = collection of k-simplices of v in Č;
  S_{k+1} = collection of (k+1)-simplices of v;
  for each k-simplex s in S_k do
    if s is not a face of any (k+1)-simplex ∈ S_{k+1} then
      return k;
    end if
  end for
  k = k + 1;
end while
return k;
```

The reduction process begins at the cell whose index is the highest. If there are more than one cell whose indices are maximal, the larger cell is chosen. After each reduction of radius, the coverage of the system is verified. If no new hole appeared, the reduction is accepted. Otherwise, the reduction is refused. This cell is flagged as not reducible and its index is set to -1 to avoid a repeating reduction. Its radius is also reversed to its previous value. The reduction process continues with another cell. This process terminates when every cell is irreducible. The details of the reduction process is introduced in the Algorithm 9.

Algorithm 9 Downhill algorithm

Input: $\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
Output: $\mathcal{U}_{\text{opt}}$ the collection of optimized cells;

```
turn on and maximize the radius for each cell in U;
compute Č(U) the Čech complex of U;
compute β*_0 and β*_1 of Č(U);
compute the index for each cell in U;
flag boundary cells to be not reducible;
flag the cells whose index ≤ 1 to be not reducible;
while exist a reducible cell do
  I = collection of the cells whose index is highest;
  c = the largest cell in I;
  Č_old = Č(U);
  r_c = r_c − Δr_c;
  if r_c < r_{c,min} then
    turn off the cell c;
  end if
  compute Č(U) the Čech complex of U;
  compute β_0 and β_1 of Č(U);
  if β_0 = β*_0 and β_1 = β*_1 then
    compute the index for each reducible cell in U;
  else
    r_c = r_c + Δr_c;
    Č(U) = Č_old;
    set c as not reducible cell;
    set index of c to -1;
  end if
end while
return U;
```

4.4 Improvements to energy saving algorithms

In this section, we introduce some techniques to enhance the performance of the energy saving algorithms. During the optimization process of these algorithms, the radius of each cell may be modified many times. After each modification of the radius of a cell, the topology of the network needs to be updated. However, this modification can only make change to topology of the region that is comprised of the cell whose radius changed and its neighbors. This region is called the local region of the cell that changed its radius. We should only update the topology of the local region of this cell.

The network coverage verification can be reduced to the coverage verification of this local region. In addition, two cells that are not neighbors do not have any topological relation. The index of a vertex can be computed by using the topology of its local region. Furthermore, if we have multiple computers, we can do these computations in parallel and distributed ways to achieve higher performance.

4.4.1 Quick Čech complex rebuild

Our energy saving algorithms requires many recomputations of the Čech complex due to the modifications of radius of cells in the optimization process. In fact, the change of radius of a cell can only make a topology change in the local region that is comprised of this cell and its neighbors. So, the recomputation of the full Čech complex after a radius change of a cell is reduced to the recomputation of the Čech complex for only this local region, as introduced in the Algorithm 10.

Algorithm 10 Quick Čech complex re-build algorithm

Input: $\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
c^* the cell that changed its radius;
$\check{\mathbf{C}}(\mathcal{U})$ the Čech complex of $\mathcal{U}$ before the radius change of c^*;

Output: the updated $\check{\mathbf{C}}(\mathcal{U})$;

```
v* = the corresponding vertex of the cell c*;
for each simplex s ∈ Č(U) do
  if v* ∈ s then
    remove s from Č(U);
  end if
end for
N = neighbors collection of the cell c*;
U* = N ∪ {c*};
Č*(U*) = the Čech complex of U*;
for each simplex s ∈ Č*(U*) do
  if v* ∈ s then
    add s to Č(U);
  end if
end for
return Č(U);
```

4.4.2 Quick verification of the network coverage

Only the cells that are not fenced or boundary cells can try to reduce the coverage radius. The radius reduction of one cell only makes topology change in the local region that is comprised of this cell and its neighbors. If there is a new coverage hole, it must be inside this local region. This means that if there is no new coverage hole after the radius reduction, the Betti numbers β_0 and β_1 of the Čech complex of this local region are unchanged. The verification of the network coverage can be reduced to the coverage verification in only this local region as in the Algorithm 11.

Algorithm 11 Quick coverage verification after a radius reduction of one cell

Input: c^* the cell that changed its radius;
$\mathcal{N}$ = neighbors collection of the cell c^*;
$\check{\mathbf{C}}^*$ = the Čech complex of $\{c^*\} \cup \mathcal{N}$ before the radius reduction;

Output: **true** if and only if there is no new coverage hole.

compute Betti numbers β_0^* and β_1^* of $\check{\mathbf{C}}^*$;
$\check{\mathbf{C}}$ = the Čech complex of $\{c^*\} \cup \mathcal{N}$ after the radius reduction;
compute Betti numbers β_0 and β_1 of $\check{\mathbf{C}}$;
if $\beta_0 = \beta_0^*$ and $\beta_1 = \beta_1^*$ **then**
  verification = **true**;
else
  verification = **false**;
end if
return verification;

4.4.3 Quick index computation of vertex

The computation of the index of a vertex only needs the collection of the simplices that contain this vertex. These simplices can be all found out by computing the Čech complex of the local region of the corresponding cell of this vertex. The local region of a cell is comprised of this cell and its neighbors. As the local Čech complex is smaller than the global one, the computation of the vertex can be accelerated by doing it from the local Čech complex. The quick computation of the index of a vertex is introduced in the Algorithm 12.

Algorithm 12 Quick index computation of vertex algorithm

Input: $\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
v the vertex to compute the index;

Output: $\hat{i}_v$ the index of v;

c = the corresponding cell of the vertex v;
$\mathcal{N} = \{c^* \in \mathcal{U} \mid c^*$ is neighbor of $c\}$;
$\mathcal{U}^* = \{c\} \cup \mathcal{N}$;
$\check{\mathbf{C}}(\mathcal{U}^*)$ = the Čech complex of $\mathcal{U}^*$;
$\hat{i}_v$ = the index of v in $\check{\mathbf{C}}(\mathcal{U}^*)$;
return $\hat{i}_v$;

4.4.4 Parallel computations for energy saving algorithms

We can improve more the speed of the optimization process by using the parallel computations on several computers. The key to enable the parallel computing is that the computation can be split to smaller independent tasks. Then, these tasks can be separately done on different computers.

We should note that only the cells that are not fenced or boundary cells can try to reduce the coverage radius. The fenced and boundary cells are assumed to be known. As we know, if a cell in the network modifies its radius, this modification may only affect the topology between this cell and its neighbors. It is topologically independent from a cell that is not its neighbor. In addition, this cell can also independently verify if it creates a new coverage hole due to its reduction of radius. To do so, it constructs the Čech of a collection of cells that includes itself and its neighbors. Then, it verifies the two Betti numbers, β_0 and β_1, of this Čech complex. If the Betti numbers β_0 and β_1 are not changed, it does not create any new coverage hole. Furthermore, its index can be updated from this Čech complex as well. This suggests that we can change the radius for some cells that are not neighbors at the same time.

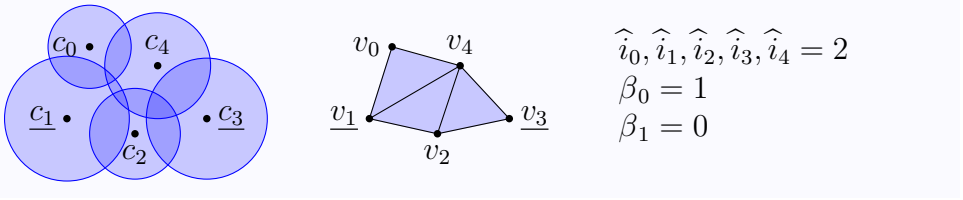
In the simulated annealing algorithm, each modification of the radius is done in one iteration. However, each iteration in this algorithm is not independent from another, because it must follow the sequence of temperatures in the cooling process. Since each state in the simulated annealing algorithm contains the modification of the previous state, the simulated annealing algorithm is considered an inherently sequential process [33]. It is difficult to parallelize the simulated annealing without changing its nature sequential process, which may cause to the non-optimized solution. However, some parallel simulated annealing techniques, which eliminate the sequential dependencies, have been proposed, for example [16, 19, 52, 54].

In this chapter, we develop the simulated annealing algorithm aimed at finding the global optimized solution of the cells' size. We do not develop the parallel or distributed simulated annealing version.

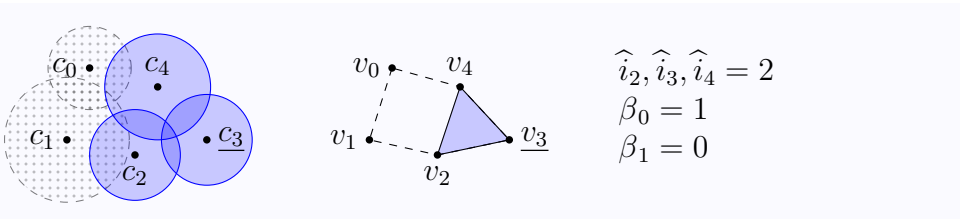
The downhill algorithm has independent iterations. In each iteration, it choose a cell whose index is highest to try to reduce its radius. There are maybe more than one cells that have the highest index. We can try to reduce the radius for all the topologically independent cells among these cells.

We have an example of parallel reductions in Figure 4.5. The Figure 4.5(a) shows the current cells of the network. There are five cells and all of these cells have index two. The two biggest cells are the cell c_1 and the cell c_3. These two cells are not neighbors, so they are topologically independent. They are chosen to reduce the radius at the same time. In this figure, they are underlined. The Figure 4.5(b) shows the point of view of the cell c_3. This cell has two neighbors, they are the cell c_2 and the cell c_4. The other cells are not its neighbors (drawn in dotted objects), and are not considered by the cell c_3, who is trying to reduce its radius. The cell c_3 constructs the Čech complex for itself and its two neighbors c_2 and c_4 to know if the reduction of its radius creates a new coverage hole. After its radius reduction, it still connects with its two neighbors by a filled 2-simplex. Its Betti numbers are: $\beta_0 = 1$, and $\beta_1 = 0$. It means that, there is no coverage hole. So, the cell c_3 updates its reduced radius. The second cell that is trying to reduce its radius is the cell c_1. Currently, it has three neighbors, they are c_0, c_2, c_4. It follows the same process as the cell c_3 did. However, after its reduction of radius, it connects with its neighbors by a chain of 1-simplices: $[v_1, v_2] + [v_2, v_4] + [v_4, v_0] + [v_0, v_1]$. The Betti numbers of this chain are: $\beta_0 = 1$ and $\beta_1 = 1$. These Betti numbers show that there is a coverage hole inside this chain. It will not update its radius. The Figure 4.5(d) shows the cells after two parallel attempts to reduce radius. The radius of the cell c_3 is reduced and the radius of the cell c_1 is unchanged.

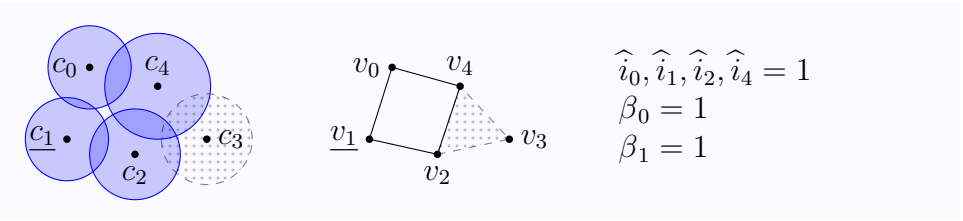
The parallelized version of the downhill algorithm is described as in the Algorithm 13. In this algorithm, the "parfor" means the loop that the iterations can be done parallel.



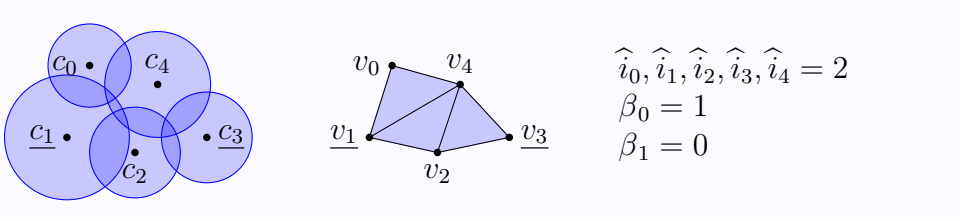
(a) Actual cells of the network, the cell c_1 and the cell c_3 are chosen to parallel reduce radius.



(b) From the point of view of the cell c_3, its reduction of radius does not make a new coverage hole.



(c) From the point of view of the cell c_1, its reduction of radius makes a new coverage hole.



(d) The updated network, only the cell c_3 reduces its radius.

Figure 4.5: Parallel reductions of the cell radius and the Čech representation.

Algorithm 13 Parallel downhill energy saving algorithm

Input: m = the number of available computers;
$\mathcal{U} = \{c_i(v_i, r_i) \mid i = 0, 1, \ldots, (N-1)\}$ the collection of cells;
Output: $\mathcal{U}_{\text{opt}}$ the collection of optimized cells;

```
turn on and maximize the radius for each cell in U;
compute the index for each cell in U;
flag boundary cells to be not reducible;
flag the cells whose index ≤ 1 to be not reducible;
while exist a reducible cell do
  I = collection of the cells whose index is highest;
  for each cell c in |I| do
    N = the collection of neighbors of the cell c;
    if N_c ∩ I ≠ ∅ then
      remove c from I;
    end if
  end for
  if |I| > m then
    M = m largest cells in I;
  else
    M = I;
  end if
  parfor each cell c in |M| do
    N = the collection of neighbors of the cell c;
    U* = {c} ∪ N;
    compute Č(U*) the Čech complex of U* and its Betti numbers β*_0, β*_1;
    r_old = r_c;
    r_c = r_c − Δr_c;
    if r_c < r_c,min then
      turn off the cell c;
    end if
    compute Č(U*) the Čech complex of U* and its Betti numbers β_0, β_1;
    if β_0 = β*_0 and β_1 = β*_1 then
      parfor each cell c* in U* do
        update index of cell c*;
      end parfor
    else
      r_c = r_old;
      flag the cell c to be not reducible and update î_c = −1;
    end if
  end parfor
end while
return U;
```

4.4.5 Distributed energy saving algorithm

In this section, we introduce a distributed version of the downhill energy saving algorithm. This algorithm is applied for each cell in the network. We should note that only the cells that are not fenced or boundary cells can try to reduce the coverage radius. As we discussed in the Section 4.4.4, if a cell tries to reduce its coverage radius, it only needs to verify the coverage of a group of cells that are composed of itself and its neighbors. The reduction can be done locally and therefore can be distributed.

We assume that every cell is connected to each other by a backhaul network. At the first step, each cell needs to search for its neighbors. Each cell transmits its position and its coverage radius to the others. Therefore, each cell knows the position and the coverage radius of the others. Each cell can compute its neighbors set by following the Algorithm 1.

Once the neighbors set is established, each cell now starts its reduction process. On each cell, there is a timer which counts down to zero. The timer is set to a uniform random value from 1 to $t_{\max}$, where $t_{\max}$ is the maximal value of the timer. If we consider the index of cells, we can know how dense the local region of each cell is. A higher index of cell is, the denser local region of each cell is. If we want to try to reduce the radius for a cell c whose index is high with a higher probability, we can set the timer to a uniform random value which varies from 1 to $t_{\max}/\hat{i}_c$, where $\hat{i}_c$ is the index of the cell c. The index $\hat{i}_c$ of the cell c can be computed as in the Algorithm 12.

When the timer of a cell is expired, this cell tries to do a reduction. If two cells that are neighbors try to reduce their radius at the same time, the coverage hole may not be detected due to the outdated information about the radius of each other. Therefore, before trying to reduce the radius, each cell sends a "pause" message to its neighbors. Then, this cell reduces its coverage radius and verifies the coverage around it by using the Algorithm 11. If no coverage hole appears, the cell confirms the reduction and sends the new value of coverage radius to its neighbors. It also sends the "continue" message to its neighbors to tell them that they can continue. If a cell received a "pause" message, it pauses its process and waits until the message "continue" is received. Then, it continues its process normally.

There is a special case where two neighbor cells whose timers expire at the same time send the "pause" message to each other simultaneously. One of these two cells receives a "pause" from another before it sends "continue" message to other cells. This cell cancels the current reduction step and sets a new value for its timer and waits to retry.

If a cell tries to reduce its coverage radius and makes a coverage hole, it reverses its coverage radius to the previous value and stops the reduction

process. This cell is set to irreducible.

The distributed energy saving algorithm applied for each cell is described in the Algorithm 14.

Algorithm 14 Distributed downhill energy saving algorithm for each cell

Input: c a cell in the network;
Output: the optimal radius for c;

```
set timer = uniform(0, t_max);
transmit the position and the coverage radius of c to other cells;
collect the information about the position and coverage of other cells;
while (1) do
  N_c = the collection of neighbors of c;
  if a "pause" received then
    wait until "continue" received;
  end if
  r_old = r_c;
  r_c = r_c − Δr_c;
  if r_c < r_c,min then
    r_c = 0;
    verify the coverage;
    if no coverage hole appears then
      send "continue" to neighbors in N_c;
      transmit the "turning off" to other cells;
      break;
    else
      r_c = r_old;
      send "continue" to neighbors in N_c;
    end if
  else
    verify the coverage;
    if no coverage hole appears then
      send r_c to neighbors in N_c;
      send "continue" to neighbors in N_c;
    else
      r_c = r_old;
      send "continue" to neighbors in N_c;
      break;
    end if
  end if
  compute index î_c of the cell c;
  set timer = uniform(0, t_max/î_c);
end while
return;
```

4.5 Complexity

Our energy saving algorithms: simulated annealing one and downhill one, require two main computations: the Čech complex of the network and its two Betti numbers β_0 and β_1. The downhill algorithm requires an additional computation: the indices of the vertices.

The details of the Čech computation has been described in the previous chapter. If the Čech complex is built up the dimension two, $d = 2$, it has the complexity $T(\check{C}, 2) = \mathcal{O}(N^2 + Nn^2)$, where N is the number of cells in the network and n is the average number of neighbors of each cell. In the case that the Čech complex is built up to its highest dimension, $d = d_{\max}$, the complexity of its construction is $T(\check{C}, d_{\max}) = \mathcal{O}(N^2 + N2^n)$.

The computation of the two Betti numbers β_0 and β_1 of the Čech complex are based on the rank computation of its homology space. The complexity of these computations are discussed in [26]. Let we denote m_k the average number of k-simplices of the Čech complex. The computation of β_0 has the complexity of $\mathcal{O}(m_1)$. The complexity of the computation of β_1 is equal to the complexity of the rank computation of a square matrix m_2 rows and m_2 columns, which is $\mathcal{O}(m_2^3)$. The complexity to compute both these two Betti numbers β_0 and β_1 is $\mathcal{O}(m_1 + m_2^3)$. The number of k-simplices in the Čech complex can be upper bounded by NC_n^k, where N is the number of cells in this Čech complex and n is the average number of neighbors of each cell. So, the complexity to compute both these two Betti numbers β_0 and β_1 is then as much $\mathcal{O}(N^3 n^6)$.

In the simulated annealing algorithm, the optimization process is repeated during the length of each temperature in the cooling schedule. The length of each temperature is L loops. The temperature is updated by the cooling schedule, whose size is K steps. So, there are totally KL loops in this algorithm. In each loop, the Čech complex and its Betti numbers β_0 and β_1 are computed. In this algorithm, the Čech complex is still built up to dimension two. So, the complexity of the simulated annealing algorithm is:

$$\begin{aligned} T_{\text{simulated annealing}} &= KL\left(\mathcal{O}(N^2 + Nn^2) + \mathcal{O}(N^3 n^6)\right) \\ &= \mathcal{O}(KLN^3 n^6) \end{aligned}$$

In the downhill algorithm, the index of each vertex is needed. To compute the index of a vertex, we need firstly find the collection S_k of all k-simplices that contain this vertex. This step has the complexity of $\sum_{k=0}^{d_{\max}} m_k$, where m_k is the number of k-simplices in the Čech complex, and $d_{\max}$ is its highest dimension. For each k-simplex in the collection S_k, we need to check if this k-simplex is a part of every $(k+1)$-simplices in the collection S_{k+1}. This step has complexity of $\sum_{k=0}^{d_{\max}-1} m_k m_{k+1}$. The complexity to compute the

index of each vertex is $\sum_{k=0}^{d_{\max}} m_k + \sum_{k=0}^{d_{\max}-1} m_k m_{k+1}$. Because the number of k-simplices is upper bounded by NC_n^k, the complexity to compute the index of each vertex is upper bounded by $N\sum_{k=0}^{d_{\max}} C_n^k + N^2 \sum_{k=0}^{d_{\max}-1} C_n^k C_n^{k+1}$. So, the complexity to compute indices for all N vertices in the Čech complex is upper bounded by $N^2 \sum_{k=0}^{d_{\max}} C_n^k + N^3 \sum_{k=0}^{d_{\max}-1} C_n^k C_n^{k+1}$. If the Čech complex is still built up to dimension two, the complexity to compute indices for all vertices is $\mathcal{O}(N^3 n^3)$. If the Čech complex is computed up to its highest dimension, the sum $\sum_{k=0}^{d_{\max}} C_n^k$ can be upper bounded by 2^n. Using the identity of Vandermonde, we can approximate the sum $\sum_{k=0}^{d_{\max}-1} C_n^k C_n^{k+1}$ by C_{2n}^{n-1} which is less than 2^{2n}. So, the complexity to compute the indices for all the vertices in this case is then as much $\mathcal{O}(N^3 2^{2n})$.

In the downhill algorithm, the reduction process is repeated until there is no redundant space. Let S be the space of the target deployment. Let us assume that all cells have the same maximum coverage radius $r_{\max}$. The redundant space, which is reducible, is $S_{\text{redundant}} = N\pi r_{\max}^2 - S$. The space reduced after a radius reduction of a cell is $\Delta S = \pi r^2 - \pi(r - \Delta r)^2$, where r is the current coverage radius and Δr is the reduced amount of the coverage radius. The average reduced space after each reduction is $\mathbb{E}\left[\pi r^2 - \pi(r - \Delta r)^2\right]$, which is $\pi r_{\max}\Delta r - \Delta r^2$. We assume that $\Delta r \ll r_{\max}$, then the average reduced space is approximated by $\pi r_{\max}\Delta r$. So the number of reductions in the downhill algorithm W is $S_{\text{redundant}}/\Delta r = (N\pi r_{\max}^2 - S)/(\pi r_{\max}\Delta r)$, which has order of $\mathcal{O}(N)$. The complexity of the downhill algorithm in the case that the Čech complex is built up to dimension two is:

$$\begin{aligned} T_{\text{downhill},2} &= W\left[\mathcal{O}(N^2 + Nn^2) + \mathcal{O}(N^3 n^6) + \mathcal{O}(N^3 n^3)\right] \\ &= \mathcal{O}(N^4 n^6). \end{aligned}$$

In the case that the Čech complex is built up to its highest dimension, the complexity of the downhill algorithm is:

$$\begin{aligned} T_{\text{downhill},d_{\max}} &= W\left[\mathcal{O}(N^2 + N2^n) + \mathcal{O}(N^3 n^6) + \mathcal{O}(N^3 2^{2n})\right] \\ &= \mathcal{O}(N^4 2^{2n}). \end{aligned}$$

The complexity of our algorithms can be reduced by using the accelerated techniques that are described in the Section 4.4. After each radius reduction, we need to verify only the coverage for the local region composed of the cell whose radius is reduced and its neighbors. The Čech complex is rebuilt following the Algorithm 10. The average number of cells in this local region is still $n + 1$, so this computation has the complexity: $T(\check{C}_{\text{rebuilt}}, 2) = \mathcal{O}\left((n+1)^2 + (n+1)n^2\right)$, which is $\mathcal{O}(n^3)$. In the case that the Čech complex is built up to its highest dimension, $d = d_{\max}$, the complexity of its construction is $T(\check{C}_{\text{rebuilt}}, d_{\max}) = \mathcal{O}\left((n+1)^2 + (n+1)2^n\right)$, which is

$\mathcal{O}(n2^n)$. Because the local region has $n+1$ cells in average, the number of the k-simplices in the Čech complex of this local region is upper bounded by $(n+1)C_n^k$. The complexity to compute the Betti numbers of the Čech complex of this group is reduced to $\mathcal{O}(n^9)$. To compute the index of one vertex following the quick Algorithm 12, we still need to compute its index in the Čech complex of its local region. So, its complexity is only $\mathcal{O}(n^6)$ if the Čech complex is constructed up to dimension two, and $\mathcal{O}(n^3 2^{2n})$ if the Čech complex is constructed up to its highest dimension.

The Table 4.1 lists the complexity of our algorithms in two probabilities: the Čech complex is constructed up to dimension 2 and up to its highest dimension.

Table 4.1: The complexity of centralized algorithms.

Complexity	$d_{\max} = 2$	$d_{\max} = \infty$
algorithms without accelerated techniques:		
full Čech complex	$O(N^2 + Nn^2)$	$O(N^2 + N2^n)$
β_0, β_1 (coverage verification)	$O(N^3 n^6)$	$O(N^3 n^6)$
indices of all vertices	$O(N^3 n^3)$	$O(N^3 2^{2n})$
simulated annealing algorithm	$O(KLN^3 n^6)$	-
downhill algorithm	$O(N^4 n^6)$	$O(N^4 2^{2n})$
algorithms with accelerated techniques:		
quick rebuilt Čech complex	$\mathcal{O}(n^3)$	$\mathcal{O}(n2^n)$
quick coverage verification	$\mathcal{O}(n^9)$	$\mathcal{O}(n^9)$
quick index of one vertex	$\mathcal{O}(n^6)$	$\mathcal{O}(n^3 2^{2n})$
simulated annealing algorithm	$\mathcal{O}(KLn^9)$	-
downhill algorithm	$\mathcal{O}(Nn^9)$	$\mathcal{O}(Nn^3 2^{2n})$

We should note that, in the simulated annealing algorithm, the Čech complex is only built up to dimension two. The simulated annealing algorithm and the downhill algorithm with the Čech complex built up to dimension two have polynomial time complexity. However, the total number of loops in the simulated annealing algorithm, KL, are much greater than the number of loops in the downhill algorithm, which is $\mathcal{O}(N)$. So, the simulated annealing algorithm has higher complexity than the downhill one with dimension two. With the Čech complex built up to its highest dimension, the downhill algorithm has the highest complexity. It gives the solution in polynomial of exponential time.

The parallel downhill algorithm requires the computation of the Čech complex, its Betti numbers β_0, β_1 and the index of each vertex in only the

local region of each cell. These computation can be done by using our improved algorithms in the Section 4.4. Each one of these computations is independent from another, and it is computed on different computers. Let m be the number of available computers. So, on each computer, the computation of the parallel algorithm has the complexity that is m times less than the complexity of the accelerated downhill algorithm (see Table 4.1).

The distributed algorithm is done locally at each cell. Each cell computes the Čech complex, its Betti numbers β_0, β_1 and its index in only the local region that includes this cell and its neighbors. These computations can be done by using the accelerated algorithms. The complexity of the distributed algorithm for each cell is $\mathcal{O}(n^9)$ when the Čech complex is built up to dimension two, and is $\mathcal{O}(n^3 2^{2n})$ when the Čech complex is built up to its highest dimension. The Table 4.2 compares the complexity of the centralized, parallel and distributed algorithm.

Table 4.2: The complexity of centralized, parallel and distributed algorithm

Complexity	$d_{\max} = 2$	$d_{\max} = \infty$
downhill algorithm		
without accelerated techniques	$O(N^4 n^6)$	$O(N^4 2^{2n})$
with accelerated techniques	$\mathcal{O}(Nn^9)$	$\mathcal{O}(Nn^3 2^{2n})$
parallel algorithm	$\mathcal{O}(Nn^9/m)$	$\mathcal{O}(Nn^3 2^{2n}/m)$
distributed algorithm	$\mathcal{O}(n^9)$	$\mathcal{O}(n^3 2^{2n})$

4.6 Simulation and results

Our energy saving algorithms were evaluated on a space 10×10 where cells were deployed randomly according to the Poisson point process. Each cell has the coverage radius that can vary from $r_{\min} = 0.1$ to $r_{\max} = 1$. The density of cells λ was set to different values from 0.2 to 1. The path loss exponent γ is 3. Our simulations were repeated 1000 times.

The simulated annealing algorithm simulations were executed with the initial temperature $T_0 = 1.95$. This initial temperature provides a high initial accepting probability of an uphill move, which is 0.95. At each temperature, the process is repeated $L = 1000$ times during the temperature length. Then, the temperature is reduced by a cooling factor which was set to $\alpha = 0.95$. The number of steps in temperature schedule is $K = 100$. With this temperature schedule, the final accepting probability of an uphill move is 0.05. The Čech

complex in the simulated annealing algorithm is built up to the dimension two, the smallest dimension that allows to verify the coverage of the network.

The downhill algorithm was tested with the Čech complex built for two different maximal dimensions two and ten. The number of loops in this simulation was set to 1000.

The parallel algorithm is only an accelerated version of the centralized one. Therefore, we do not perform the simulation for this algorithm. The simulation for the distributed algorithm is performed with the Čech complex built up to dimension two. The purpose of this simulation is to compare its performance with the simulated annealing one and the centralized downhill one.

4.6.1 Average consumption power per cell with optimized radius.

Before the optimization, each cell is set to work with its maximum coverage radius $r_{\max} = 1$. At this state, each cell transmits the maximal power, which is 1, following the Equation 4.6. After performed our energy saving algorithms, the average consumed power per cell with the optimized radius is shown in Figure 4.6. The higher density of cells is, the more power is saved. The simulated annealing algorithm, which gives an approximation of the global optimal solution, saves most power. At the highest density of cell, each cell operates with 35% of its maximal power in average, thus saving 65% power. The downhill algorithm saves 62% power with the Čech complex built up to dimension two and it saves 60% power with the Čech complex built up to dimension 10. These results show that the solution of the downhill algorithm also approaches the global optimal one. In the downhill algorithm, if the Čech complex is built up to dimension two, all the reducible cells have the same index two. So, there is no priority among cells. The downhill algorithm in this case tries to reduce energy in the same way for every one. If the Čech complex is built up to its highest dimension, the cells in the dense region, which often have the higher index, have priority to reduce their coverage radius first. In this case, the downhill algorithm quickly reduces the overlap in only dense regions. The downhill with the Čech complex built up to dimension two has the better performance than the one with the Čech complex built up to its highest dimension. In addition, the downhill algorithm with the higher dimensional Čech complex has the higher complexity. Therefore, the downhill algorithm with the Čech complex built up to dimension two is recommended for a centralized energy saving algorithm.

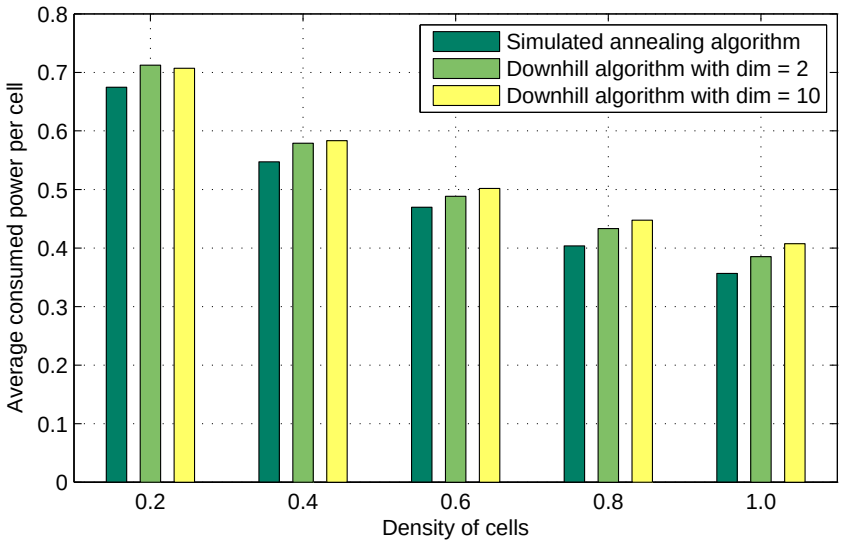


Figure 4.6: Consumption power per cell minimized by different centralized algorithms.

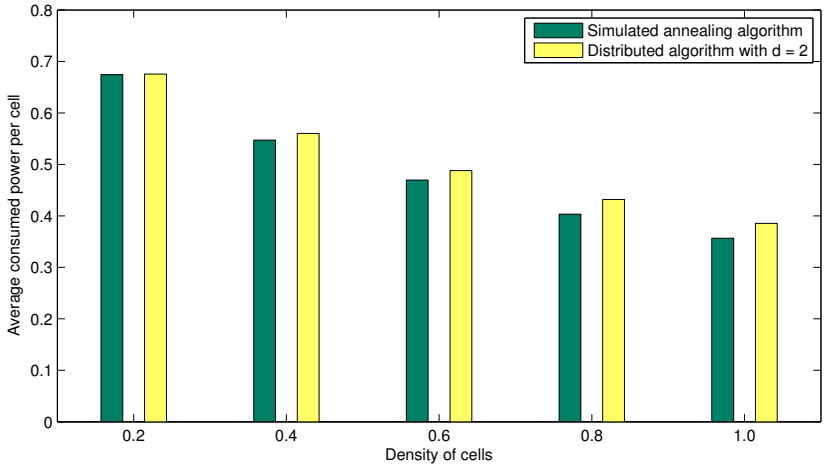


Figure 4.7: Consumption power per cell minimized by simulated annealing algorithm and distributed algorithm.

Because the downhill algorithm with the Čech complex built up to dimension two has better performance than the one with the Čech complex built up to the higher dimension, we perform the simulation for the distributed algorithm only with the Čech complex built up to dimension two. The performance of the distributed algorithm is compared with the perfor-

mance of the simulated annealing algorithm in Figure 4.7. At the density one, the distributed algorithm saves 62% power, which is 3% smaller than the result of the simulated annealing one. The distributed algorithm with the Čech complex built up to dimension two has the same performance with the centralized downhill one. The distributed algorithm is applied for each cell and has the smallest complexity. Therefore, the distributed algorithm with the minimal Čech complex is also recommended for the optimization of the networks.

4.6.2 Probability density function of optimized radius.

Although the average power consumed per cell by different algorithms are almost equal, the pdf of the optimized radius obtained by these algorithms are quite different as shown in Figure 4.8, 4.9 and 4.10 for three densities of cells: 1 (high), 0.6 (medium) and 0.2 (low).

The simulated annealing algorithm gives an approximation of the global optimal solution. The pdf of optimized radius by this algorithm shows that the optimal deployment should contain both big cells and small cells. The big cells should ensure a large background coverage for the network, while the small cells should ensure small locals in indoor environments. However, simulated annealing algorithm does not turn off many cells. It retains many working cells. The number of cells that can be turned off by using the simulated annealing algorithm is always less than 10% for all values of density. The turned off cells are redundant cells. They are not needed in real deployment. In contrast, the working cells are needed. So, the optimal deployment should contains a large number of base stations, this may cause a high device cost.

Conversely, the downhill algorithm turns off many cells. The number of turned off cells obtained by using downhill algorithm with dimension two is higher than 35%, and with dimension ten is higher than 40% for the high density. It retains a small number of working cells. As we discussed in the Section 4.6.1, the difference of consumed power between the cells optimized by the simulated annealing algorithm and the one optimized by the downhill algorithm is small. The downhill algorithm may have a better trade off between the transmission power cost and the hardware deployment cost.

The pdf of optimized radius by distributed algorithm with the Čech complex built up to dimension two for different density of cells is shown in Figure 4.11. It is similar to the one optimized by the downhill algorithm with the Čech complex built up to dimension two. This result shows that, the distributed algorithm and the downhill algorithm at dimension two have the same performance.

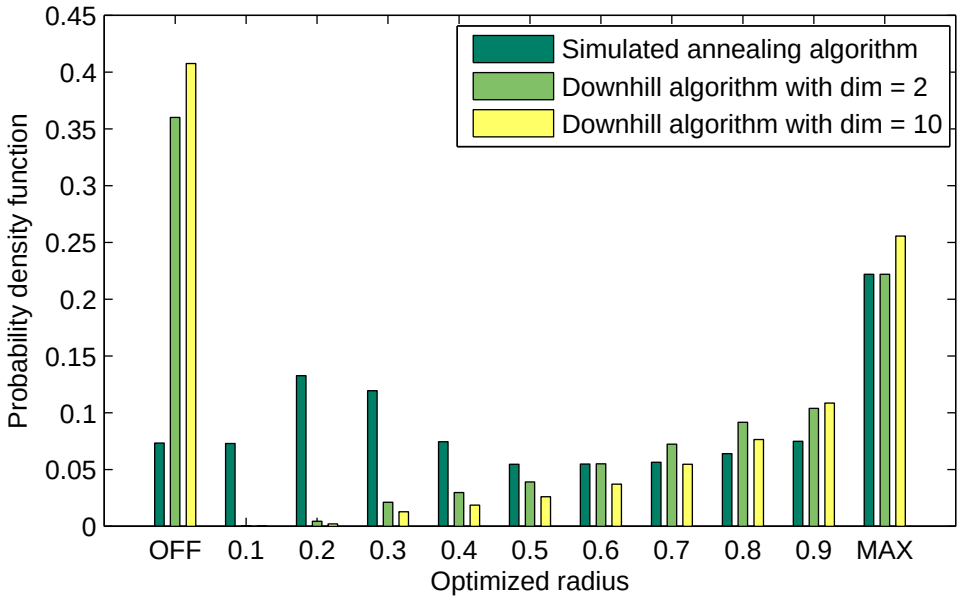


Figure 4.8: Pdf of optimized radius at cell density = 1.0 by centralized algorithms

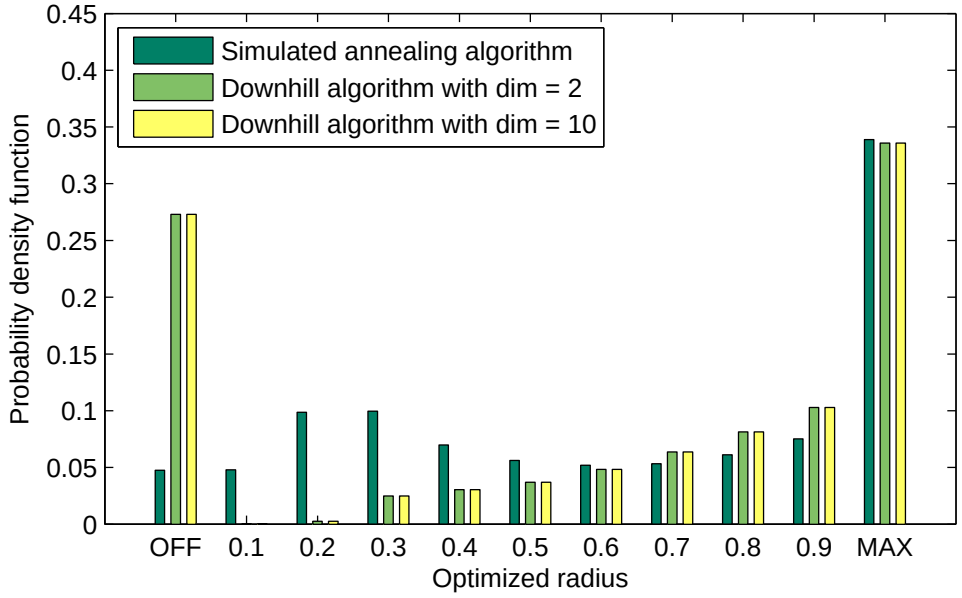


Figure 4.9: Pdf of optimized radius at cell density = 0.6 by centralized algorithms

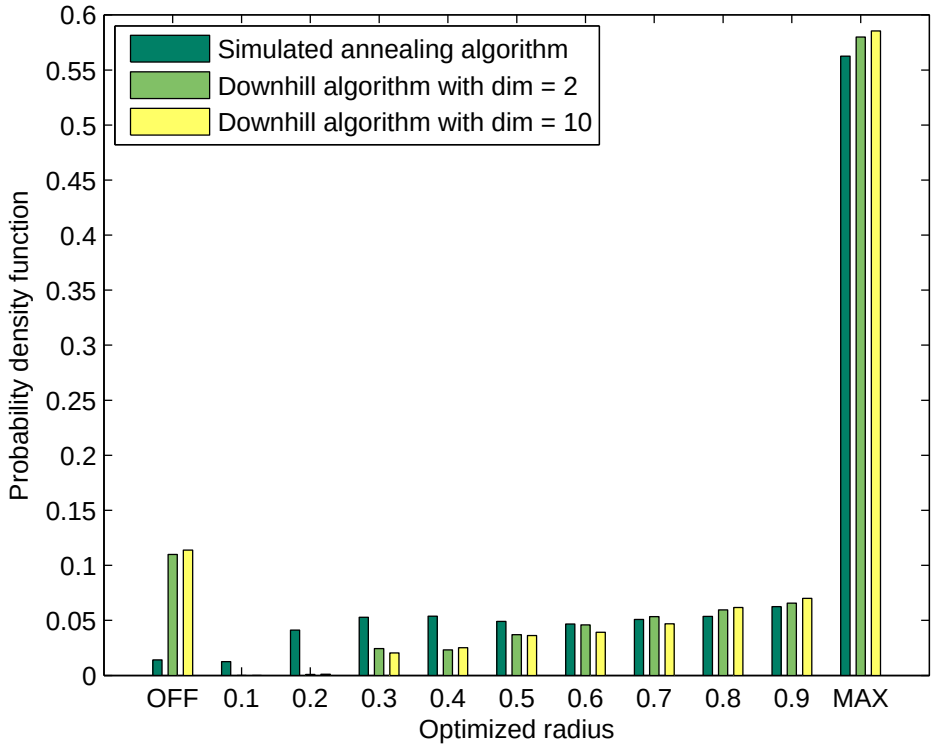


Figure 4.10: Pdf of optimized radius at cell density = 0.2 by centralized algorithms

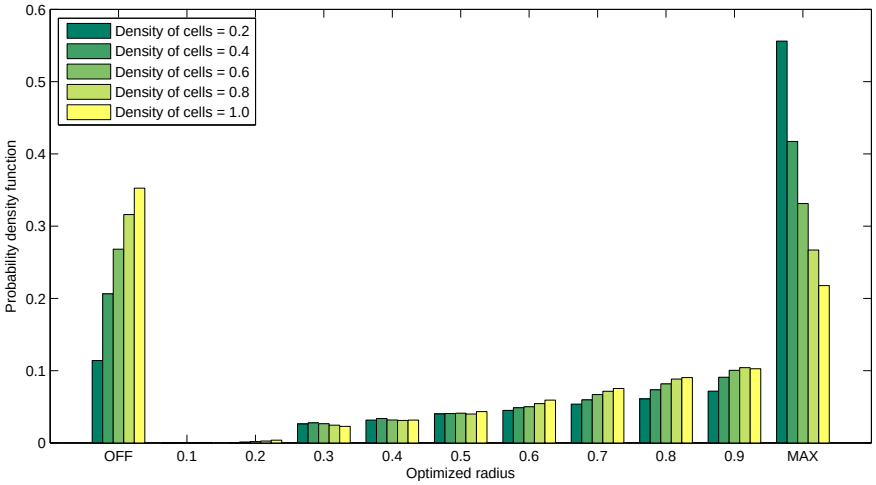


Figure 4.11: Pdf of optimized radius by distributed algorithm at different density of cells

4.7 Conclusion and Discussion

This chapter introduced two homology based algorithms for saving energy in wireless networks: simulated annealing one and downhill one. Some improvements to these algorithms are also presented to reduce their execution time as well as to perform them on parallel or distributed computers. The simulated annealing algorithm, which neglects priority between cells, saves the most power. But, it can only turn off a small number of cells. On the other hand, by considering priority between cells, the downhill algorithm firstly reduces power for cells in dense region and can turn off a larger number of cells. The higher the intensity of cells is, the higher the number of cells that can be turned off is. In addition, the difference between the total consumed power minimized by the two variants of algorithm is less than 5%. The downhill algorithm with dimension two has the lowest complexity among centralized algorithms. It achieves the solution in polynomial time. The distributed version can be computed locally and has lower complexity. It has the same performance as the downhill one. The downhill algorithm and the distributed algorithm with the Čech complex built up to dimension two are recommended to optimize the wireless systems transmission powers.

Chapter 5

Distributed load balancing for wireless networks

In this chapter, we introduce a distributed load balancing algorithm for wireless networks. Traffic load in wireless networks is sometimes unbalanced. Some cells are overloaded, while others remain free. Our algorithm controls the transmission power of each cell in the network, not only to satisfy the coverage constraint, but also to redirect users from overloaded cells to the lower traffic loaded ones. As a result, the traffic load of the wireless network is balanced. The simulation shows that this algorithm improves 2.3% capacity of the whole network when the demands of traffic load of users are fast varying.

5.1 Introduction

The traffic load in the mobile networks is growing very quickly. Most of users now have a smart-phone and use applications on it. However, most of these applications require connection to frequently update and exchange data. In addition, the users always want the applications to run everywhere, every time, with the fastest speed. Therefore, the bandwidth of networks needs to be used more effectively.

Due to the mobility of users, some cells can be overloaded at some moments while the others are still underloaded at the same time. These underloaded cells have free resources but they are not used. In the others words, these free resources are wasted. The traffic load in the network should be balanced for cells. However, the networks configuration is traditionally manual and costly. There should be a self-organized method to balance the traffic load between cells in real time.

In this chapter, we introduce a load balancing algorithm for wireless networks, which not only prevents the overload problem for the base stations but also ensures a strong quality of services for every user. Our algorithm changes the transmission power for each base station in order to redirect some users in overloaded cells to low loaded cells. Then, the traffic loads of cells are balanced. The topology of the network is always updated. We use the simplicial Čech complex to represent the topology of the network. Thanks to simplicial homology theory, the coverage of the network is always manageable. This allows us to always ensure the coverage of the network as well as to avoid the interference due to the redundant transmission power. In addition, our algorithm redirects the users to the selected cells that has enough free resources to allocate to these users. Furthermore, our algorithm does not need any information about the position of users. Therefore, it is practical in real applications.

The remainder of this chapter is organized as follows. The Section 5.2 introduces the system model. The Section 5.3 is devoted to describe all the details about our distributed load balancing algorithm. The complexity of this algorithm is discussed in Section 5.4. The simulation results will be presented in the Section 5.5. Finally, we conclude and discuss about our works in the Section 5.6.

5.2 System model

We are considering a wireless network that is high traffic loaded. We assume that the coverage of the network is satisfied before performing the load balancing algorithm. This means that before applying our load balancing algorithm, there is no coverage hole in the network.

We consider a wireless network that contains a collection of cells:

$$\mathcal{C} = \{c_i(v_i, r_i) \mid i = 1, 2, \ldots, N\};$$

where N is the number of cells, the index i indicates the i-th cell in the network, and $c_i(v_i, r_i)$ is the cell centered at the position of its base station v_i and covering the space within a radius r_i.

Let $p_{t,i}$ be the transmission power of the i-th cell c_i and u be a user in this cell. The distance from the user u to the base station c_i is $d_{u,i}$. The path loss from the base station c_i to the position of the user u is $L_{i,u}(d_{u,i})$. To estimate the path loss $L_{i,u}(d_{u,i})$, we use the COST-231 model [7]. The received power of a user u from the base station of the i-th cell c_i is:

$$p_{r,i,u} = p_{t,i} - L_{i,u}(d_{u,i}).$$

Let p_s be the sensitivity of the mobile device, which is the the smallest received power that allows the mobile user to decode exactly the information. The user u is covered by the cell c_i if its received power is not less than its sensitivity: $p_{r,i,u} \geq p_s$. Therefore, the coverage radius of the i-th cell c_i is the maximum distance $d_{u,i}$ that satisfies $p_{t,i} - L_i(d_{u,i}) \geq p_s$. We assume that the sensitivity of every user in the network is the same p_s. The radius of the cell c_i is controlled by the transmission power $p_{t,i}$. For more information about cell's size mapping, see [36].

5.3 Distributed load balancing algorithm

In this section, we describe entirely the details about our load balancing algorithm. At each base station c_i, it periodically estimates the requested traffic load, denoted ρ_i, which is generated by its registered users as:

$$\rho_i = \sum_k \rho_{i,k}/\mu_i;$$

where $\rho_{i,k}$ is the number of requested resource blocks of the k-th registered user and μ_i is the capacity of the base station c_i. The capacity of a base station is the total number of resource blocks that are available in this base station.

We assume that all the cells are connected by the backhaul network. Each cell can transmit information about its position and its coverage radius to other cells via this backhaul network. We say that two cells are neighbors if the coverage of each one intersects with the coverage of the other. Each cell can compute its table of neighbors by following the Algorithm 1.

The base station c_i sends the information about its requested traffic load ρ_i, its position v_i and its current coverage radius r_i to its neighbors. Therefore, each base station knows information about requested traffic load of itself and all its neighbors. It also knows the position and the coverage of all its neighbors too. This helps each cell in computing the coverage around it.

A base station is called hot, or overloaded if its requested load is greater than one. It is called warm if it is not hot and its requested load is greater than a warm load threshold ρ_w such that $0 < \rho_w < 1$. Otherwise, the base station is called cool.

If a base station is hot, then it needs to reduce its requested traffic load. One solution is to redirect some of its registered users to its neighbors. It should choose a neighbor that has a lot of free resources, that is cool, to request a help. The neighbor that is chosen is called a helper. A neighbor that is warm will not be chosen to be a helper. This will prevent a helper to

become hot after received the redirected users from the hot base station c_i. The redirection of users needs to make sure that the redirected users have enough strength of received signal to connect to the target base station, and at the same time, the target base station must have enough free resources to allocate to these redirected users. However, the users in the hot base station can be far from the helper base station who is a neighbor. The helper needs to increase its transmission power. Denote Δp_j the power step that a cell c_j can increase to help its neighbor.

Let $\mathcal{N}_i$ be the collection of neighbors of the base station c_i. The base station c_i who is overloaded needs to choose a cool neighbor, who will be a helper, c_h that has the lowest traffic load:

$$c_h = \arg\min_{c_j \in \mathcal{N}_i} \rho_j \text{ such that } \rho_j \leq \rho_w.$$

If there is no available helper, so there is no neighbor who can help c_i. The cell c_i needs to retry the process in the next time frame. When the helper base station c_h is chosen, the base station c_i now sends a help request to c_h and waits. Once received the help request, the helper c_h increases its transmission power by an amount Δp_h to attract the users to connect to it. Some users will reconnect to the helper after its increase of transmission power, this is done following the handover process in [1]. At the same time, the helper transmits its new coverage radius to other cells. It also requests its neighbors to reduce their transmission power if possible. However, a fast reduction of transmission power may create a coverage hole. Each neighbor of the helper c_h, that included the hot cell c_i, tries to reduce its transmission power an amount $\Delta p_k/2$, where c_k is the k-th neighbor of the helper c_h. If this reduction does not create any coverage hole, the reduction is confirmed and this cell sends its new coverage radius to other cells. The neighbor that is the helper of another cell will ignore this reduce transmission power request. The reduction of transmission power must satisfy the coverage condition that no coverage hole appears. To verify if a coverage hole appears, the reduced transmission power cell constructs a Čech complex for itself and its neighbors. Then the information about the coverage can be tractable through the computation of the Betti numbers of this Čech complex as in the Algorithm 15. We have assumed that, before applying the load balancing algorithm, there is no coverage hole in the network. Therefore, if no coverage hole appears after the reduction, the Betti numbers must be $\beta_0 = 1$ and $\beta_1 = 0$. Otherwise, a coverage hole appears. If a coverage hole appears, this cell must return its transmission power to the previous value. For the details about the construction of the Čech complex, we refer to Chapter 3.

Algorithm 15 Verification if a coverage hole appears

Input: c_i the cell that will reduce its transmission power;
$\mathcal{N}_i$ the neighbors set of c_i;
Output: **true** $\iff$ a coverage hole appears;
compute $\check{\mathbf{C}}$ = the Čech complex of $c_i \cup \mathcal{N}_i$;
compute β_0 and β_1 of $\check{\mathbf{C}}$;
verification = false;
if $\beta_0 \neq 1$ or $\beta_1 \neq 0$ **then**
verification = **true**;
end if
return verification;

However, there is a special case. Considering a network with small cells deployed, such as: pico cells and micro cells, if a small cell is hot, and it chooses a macro cell to be a helper, then it does not send a help request to the macro cell. The small cell automatically tries to reduce its transmission power, verifies the coverage as in the Algorithm 15, and broadcasts the new information to its neighbors. The transmission power of the macro cells is much higher than the one of small cells. This is to prevent the unnecessary modification of macro cell transmission power, which affects many other small cells inside the macro cell. The procedure to balance the load for each base station is described in the Algorithm 16. The process of load balancing is repeated every radio time frame. For more details about resource blocks and radio time frames, see [5].

We should note that, a coverage hole may appear due to an outdated information about the coverage radius. Therefore, whenever a cell tries to reduce its coverage radius, it firstly sends a "pause" signal to its neighbors. Then, it updates the information from neighbors and processes the reduction. When it finishes the reduction, it sends a "continue" message to its neighbors to indicate them that they can continue. If a cell receives a "pause" signal, it pauses the balancing process and waits until a "continue" signal is received. To avoid a special case that two neighbors send "pause" at the same time, and both of them receive the "pause", if one cell sends the "pause" and receives another "pause" before it sends a "continue", it waits in a period then tries to send "pause" again.

Algorithm 16 Load balancing algorithm applied to each cell

Input: ρ_w the warm load threshold;
c_i the i-th cell in the network;
Output: balanced traffic load for the cell c_i;

```
send position and coverage radius to other cells;
while (1) do
  estimate the traffic load ρ_i;
  collect position and coverage radius of other cells;
  get the neighbors set N_i;
  send traffic load to neighbors of c_i;
  collect traffic load of neighbors;
  if ρ_i > 1 then
    helper cell c_h = arg min_{c_j ∈ N_i} ρ_j such that ρ_j < ρ_w;
    if c_i is pico cell or micro cell and c_h is macro cell then
      update coverage radius of neighbors;
      p_{t,i} = p_{t,i} − Δp_{t,i}/2;
      if a coverage hole appears then
        p_{t,i} = p_{t,i} + Δp_{t,i}/2;
      end if
    else
      send request to increase power to c_h;
    end if
  end if
  if c_i received a help request and c_i is the helper then
    p_{t,i} = p_{t,i} + Δp_{t,i};
    send request to reduce the transmission power to neighbors of c_i;
  else
    if a reduce power request message received and c_i is not a helper then
      update information from neighbors;
      p_{t,i} = p_{t,i} − Δp_{t,i}/2;
      if a coverage hole appears then
        p_{t,i} = p_{t,i} + Δp_{t,i}/2;
      end if
    end if
  end if
  send position and coverage radius to other cells;
  wait until next radio time frame;
end while
return ;
```

5.4 Complexity

The load balancing algorithm has two main steps. The first step is to estimate the traffic load of each cell. Let m be the average number of users in each cell. The base station of each cell needs to read the request of each user to estimate the traffic load in this cell. The complexity to estimate the traffic load in each cell is $\mathcal{O}(m)$.

The second step is to verify the coverage of the local region. The local region is composed of the cell that modifies its transmission power and its neighbors. This step requires the computation of the Čech complex of the local region and its Betti numbers. The complexities of these computations have been discussed in Section 4.5. The complexity of this step is $\mathcal{O}(n^9)$, where n is the average number of neighbors of each cell.

The complexity of the load balancing algorithm that is applied for each cell is $\mathcal{O}(m + n^9)$.

5.5 Simulation results

5.5.1 Network deployment

We deploy the macro cells according to the hexagonal model. The number of macro cells is 91. Each macro cell has a capacity of 50 resource blocks. The maximal transmission power of each macro cell's base station is 46 dBm. At the starting state, the transmission power of each base station is set to 42 dBm. The initial radius of each macro cell is 900m.

In each macro cell, we deploy two micro cells, which are placed uniformly in the coverage of macro cells. However, the micro cells are often deployed in the rear side of the macro cell to increase the coverage at cell edge. In addition, if a micro cell is deployed too close to the macro cell's base station, due to the much stronger signal from the base station of the macro cell, the users will connect to the macro cell even they are very close to the micro cell's base station. In our simulation, we assume that the distance from the base station of a micro cell to the one of a macro cell is longer than two third of the coverage radius of the macro cell. Each micro cell has a capacity of 15 resource blocks. The maximal transmission power of a micro cell is 30 dBm, and the starting one is 24 dBm. The Figure 5.1 shows an example of network deployment.

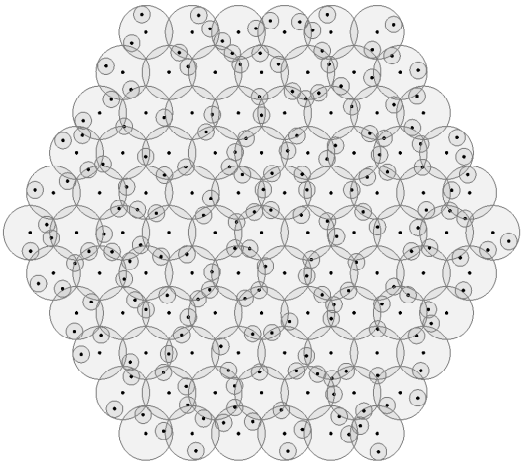


Figure 5.1: A hexagonal network.

The parameters about the macro cells and micro cells are summarized in Table 5.1 and Table 5.2.

Table 5.1: Macro cell parameters

Parameters	Assumptions
Carrier Frequency	1800MHz
Bandwidth	50 resource blocks
Distribution of base station	hexagonal
Users	uniformly distributed
Maximum transmission power	46 dBm
Initial transmission power	42 dBm
Antenna gain	18 dB
Cable loss	2 dB
Initial radius	900m
Number of macro cells	91 cells

We assume that the number of requested resource blocks of each user is following a Poisson law. In average, each user requests two resource blocks. We denote R_{total} the total number of requested resource blocks, which is generated by all users in the network. And, we denote C_{total} the total capacity of all cells in the network, where the capacity of each cell is the total number

Table 5.2: Micro cell parameters

Parameters	Assumptions
Carrier Frequency	1800MHz
Bandwidth	15 resource blocks
Distribution of base station	uniformly distributed
Users	uniformly distributed
Maximum transmission power	30 dBm
Initial transmission power	24 dBm
Antenna gain	18 dB
Cable loss	2 dB
Number of micro cells	182 cells

of available resource blocks in this cell. The traffic load T of the network is: $T = R_{\text{total}}/C_{\text{total}}$. In our simulation, we analyze the performance of the algorithm with a given traffic load T, therefore we generate the total number of requested resource blocks R_{total} according to the Poisson point process which has intensity $T \times C_{\text{total}}$.

We assume that the total number of requested resource blocks in the micro cells and macro cells are proportional to its capacity. The number of requested resource blocks in micro cells is $R_{\text{micro}} = R_{\text{total}} \times \frac{C_{\text{micro}}}{C_{\text{mico}}+C_{\text{macro}}}$. We then generate the users in micro cells first. The users are uniformly distributed in coverage of all micro cells. The number of requested resource blocks of each user is generated following a Poisson law with intensity that is two resource blocks per user. When the accumulated number of requested resource blocks in all micro cells reaches the number R_{micro}, we stop the process of generating the users in micro cells. Similarly, we generate the users in macro cells until the total number of requested resource blocks in the network reaches R_{total}.

We assume the COST-231 model for the radio propagation [7] with the correction $c = 3$ for urban areas.

$$L = 46.3 + 33.9 \log f - 13.82 \log h_B - a(h_R) + (44.9 - 6.55 \log h_B) \log d + c,$$

where $a(h_R) = (1.1 \log f - 0.7)h_R - (1.56 \log f - 0.8)$. The base station antenna height h_B is 30m and the mobile antenna height h_m is 1.5m The sensitivity of mobile user is -104 dBm. The body loss at mobile user is 3 dB.

In our simulation, each time frame is 10ms. In each time frame, the load balancing algorithm is performed for each cell.

5.5.2 Performance with varying traffic load

In this section, we consider the performance of the load balancing algorithm when the users change their number of requested resource blocks frequently. We assume that each user changes its demand of resource blocks every 10 time frames. In each time frame, each cell applies the load balancing algorithm one time.

We assume that the average traffic load in the network is 0.9, it means the total number of requested resource blocks is 90% the total capacity of the network in average. Although the average traffic load is less than one, the number of resource blocks demanded by each user is random. Some cells in network are overload while others are free. We define the outage of each cell as the number of requested resource blocks in this cell that are not serviced. The outage of the network is the accumulated outage of every cell.

In this simulation, the warm load threshold ρ_w is set to 0.9 for every cell. It means that a cell whose traffic load is higher than 0.9 will decline the help request from the hot cells. The power step Δp is 1dB for every cell. We perform the simulation in 250 time frames and record the outage of network in two cases: without load balancing applied and with load balancing applied.

We show one example of simulation in Figure 5.2. The outage of system in the case that the load balancing is not applied is drawn as the red line. The outage of system in the case that the load balancing is applied is drawn as the blue line. Because the users change their demand of resource blocks every 10 time frames, the outage of system is also changed every 10 time frames. The blue line is under the red line, this shows that by using the load balancing, the outage of system is reduced. The number of reduced resource blocks in outage is the capacity gain of system. In the Figure 5.2, the difference between the red line and the blue line describes the capacity gain of the load balancing algorithm.

We repeat this simulation 1000 times, and compute the average reduction of the outage of the network when the load balancing is applied. This load balancing algorithm reduces by 25.7% the number of resource blocks in outage, and gains 2.3% capacity of the whole system.

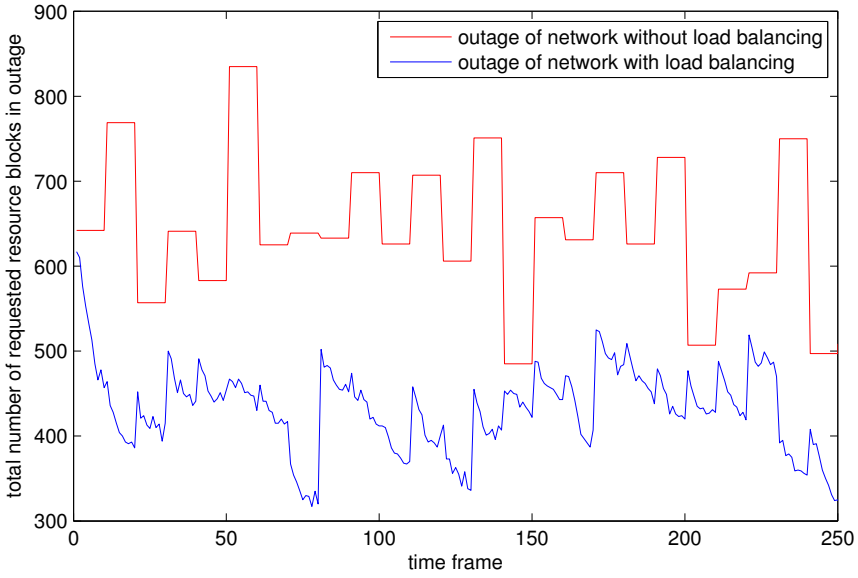


Figure 5.2: Outage of a varying traffic network with and without load balancing.

5.5.3 Performance with constant traffic load

In this section, we consider the performance of the load balancing algorithm when the traffic load is kept constant. Indeed, the demand of resource blocks of each user is unchanged during the simulation. Some properties of the load balancing algorithm such as: capacity gain, convergence speed and stability are considered at different traffic load T of the network from the low one to the high one. When the traffic load of the network is low, it is also necessary to analyze the performance of the load balancing algorithm. Due to the mobility of users, some cells may be overloaded while others are not. For instance, a train runs on the ground from one cell to another. The average traffic load of the network is assumed to be low, but the train may contain many people inside it, and can overwhelm the cell in which the train is traveling. The outage of each cell is the number of requested resource blocks of this cell that are not serviced. The outage of network is the accumulated outage of every cell.

We deploy the network as described in Section 5.5.1. The total traffic load of network T is sequentially assumed from 0.5 (low) to 0.9 (high). We perform the load balancing algorithm with different values of the warm load threshold ρ_w and the power step Δp. The simulation is performed in 200

time frames. Each time we change the value of one of these parameters, we repeat the simulation 100 times. The warm load threshold ρ_w and the power step Δp are assumed to be the same for every cell.

First, we consider the performance of the load balancing algorithm when the network is high loaded in Figure 5.3 and when the network is low loaded in Figure 5.4.

The Figure 5.3 shows the performance of the load balancing algorithm when the traffic load of the network is $T = 0.9$, a high value. The warm load threshold ρ_w is set to 0.9 and the power step Δp is 1 dB for all cell. The users do not change their demand of resource blocks in each simulation, then the outage of network is the same in every time frame. The average outage of network when the load balancing is not applied is drawn by a red line. The blue line draws the average outage of network when the load balancing is applied. The dotted line shows the upper bound and lower bound of the 95% confidence interval of the average outage of the balanced network. The outage of the balanced network is quickly reduced. The outage of the network is reduced by 34% within 28 time frames.

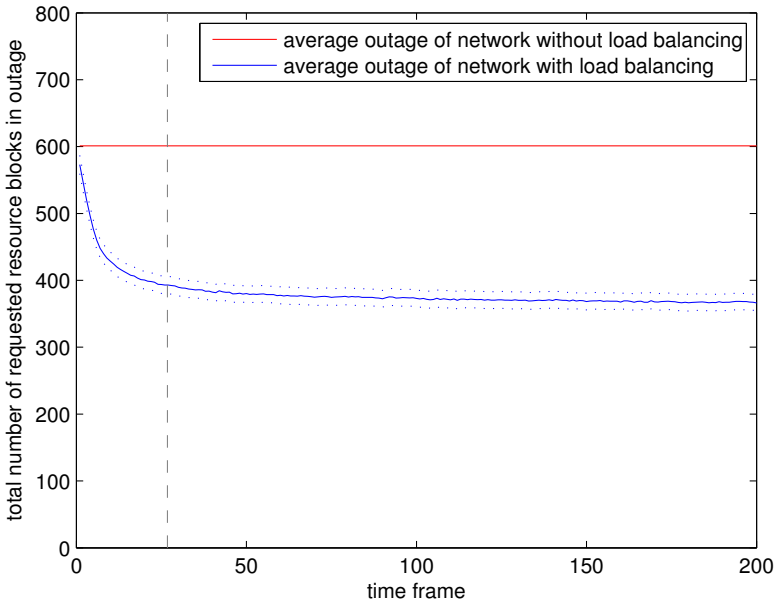


Figure 5.3: Average outage of the network when the traffic load of network is $T = 0.9$, the power step is $\Delta p = 1$ dB, the warm load threshold is $\rho_w = 0.9$.

The Figure 5.4 shows the performance of the load balancing when the traffic load is $T = 0.5$, a low value. The warm load threshold ρ_w is also set

to 0.9 and the power step Δp is 1 dB for all cell. The outage of the balanced system is reduced very fast. During 16 time frames, the outage of the system is reduced by 80%.

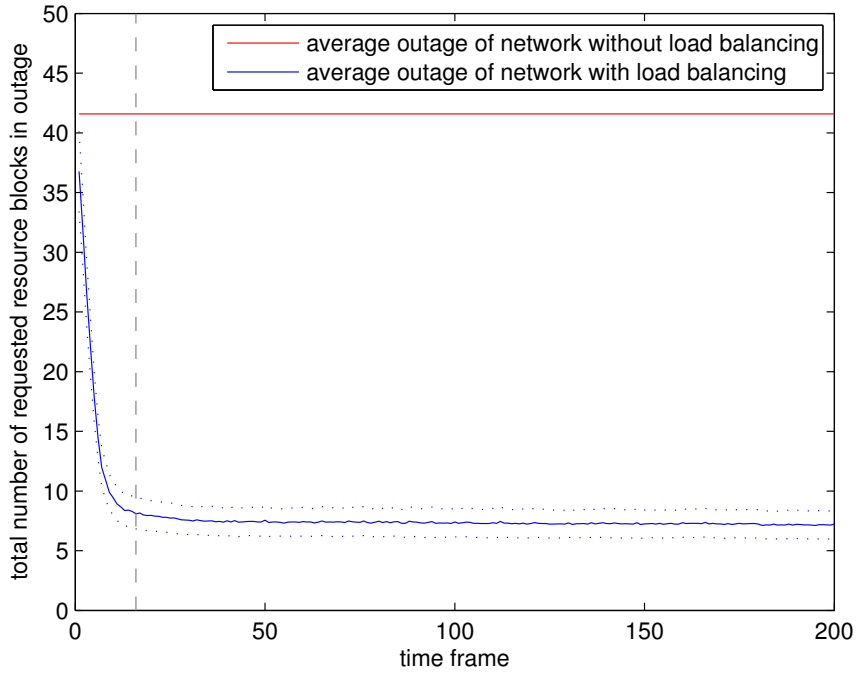


Figure 5.4: Average outage of the network when the traffic load of network is $T = 0.5$, the power step is $\Delta p = 1$ dB, the warm load threshold is $\rho_w = 0.9$.

The outage of the system fluctuates little in later time frames. When the traffic load of network is higher, the fluctuation is also higher. We explain this fluctuation phenomenon by the fact that: some cells which are very hot redirect a large number of users to other cells. In fact, when the average traffic load of the whole network is very high, most cells have quite high traffic load and some cells become overloaded and require another to help. Therefore, the outage of network may fluctuate a little.

However, the most important thing is that the outage of the balanced network is reduced quickly in first time frames. When users change their demand of resource blocks frequently, the little fluctuation in later time frames is not important. The number of reduced requested resource blocks in outage is the capacity gain of the network. The Figure 5.5 compares the capacity gain of network at different traffic loads T when the load balancing is applied with different warm load thresholds ρ_w. The power step Δp is set to 1 dB.

The capacity gain is recorded at the last time frame of simulation, the 200-th time frame.

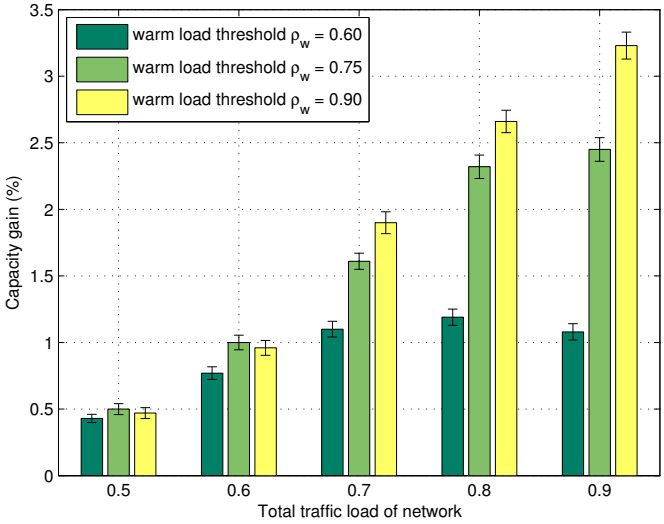


Figure 5.5: Capacity gain of system with different warm load thresholds ρ_w, and $\Delta p = 1$ dB

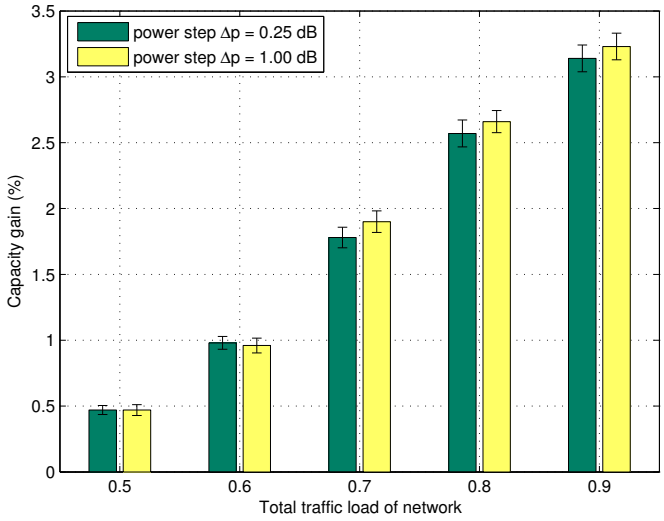


Figure 5.6: Capacity gain of system with different power steps Δp, and $\rho_w = 0.9$

When the traffic load of the network is high, the higher warm load threshold ρ_w is, the higher capacity gain is. In fact, when the warm load threshold is set to high value, it allows many cells become helpers. Therefore, the more overloaded cells have a chance to be helped. This lets more users redirect to the free cells. So, the higher capacity gain is achieved.

In Figure 5.6, the capacity gain is compared when the load balancing is applied with different power steps. We should choose only the power steps that are much smaller than the transmission power of base station. We choose the $\Delta p = 1$ dB and $\Delta p = 0.25$ dB for the comparison. The warm load threshold ρ_w is set to 0.9. At each traffic load T, the capacity gains in long term with different power steps Δp are almost equal.

The different values of the warm load threshold ρ_w and the power step Δp may also affect other properties of performance such as: convergence speed and stability. We considered that the outage of balanced network will approach the stability value within the first 100 time frames. Then, we compute the mean value of the outage of balanced network in the last 100 time frames. We denote this mean by M. When the outage of the balanced network goes below the value $1.05 \times M$, we say that the network converged at the current time frame t. The duration from starting point to this current time frame t is defined as the convergence time of the network.

The convergence time is shown in Figure 5.7 with different warm load thresholds, and in Figure 5.8 with different power steps. With higher warm load threshold, there are more available helpers, so more requests to help can be satisfied. Therefore, the process of requests and helps are longer. As a result, the convergence time is higher. The convergence time increases if the power step Δp is reduced. A significant increase of convergence time is shown when the power step is reduced in Figure 5.8. As shown in Figure 5.6, the capacity gains with different power steps Δp are almost the same. This shows that the smaller Δp reduces the speed of convergence but does not change the capacity gain in long term. In varying traffic load network, we should not choose a so small power step Δp because of the slow convergence speed.

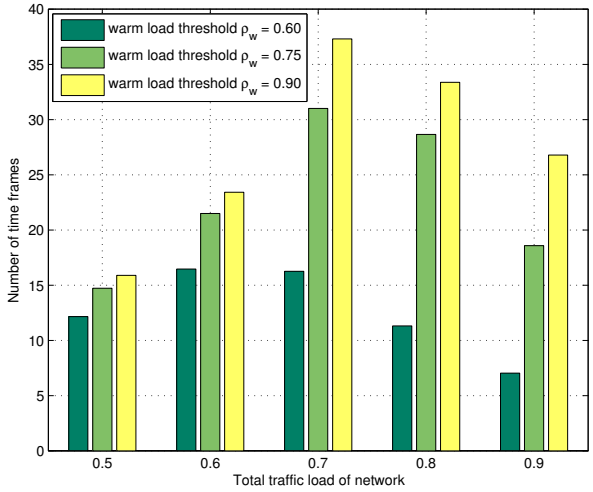


Figure 5.7: Convergence time with different warm load thresholds ρ_w, and $\Delta p = 1$ dB

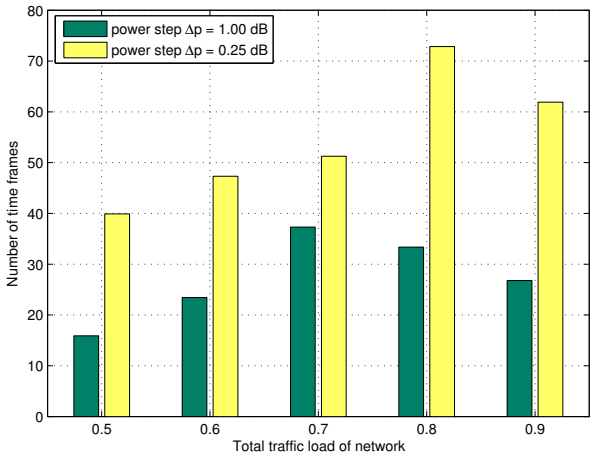


Figure 5.8: Convergence time with different power steps Δp, and $\rho_w = 0.9$

When we reached the convergence time, we want to know about the stability of the network after the converged point. We use the standard deviation of the outage of the network from the mean value M, denoted as σ, as the measurement for the stability of the network.

The Figure 5.9 shows the standard deviation of the outage of balanced network by different warm load thresholds ρ_w. This figure shows that the higher warm load threshold is, the lower stability of the system is. There

is a trade-off here. The higher warm load threshold allows higher capacity gain, as shown in Figure 5.5. However, it also makes the network less stable. With higher ρ_w, the more cells are allowed to be helpers. However, some helpers that have high traffic load will be overloaded after receiving many users from the overloaded cells. These cells will request another cell to help. This makes the network less stable.

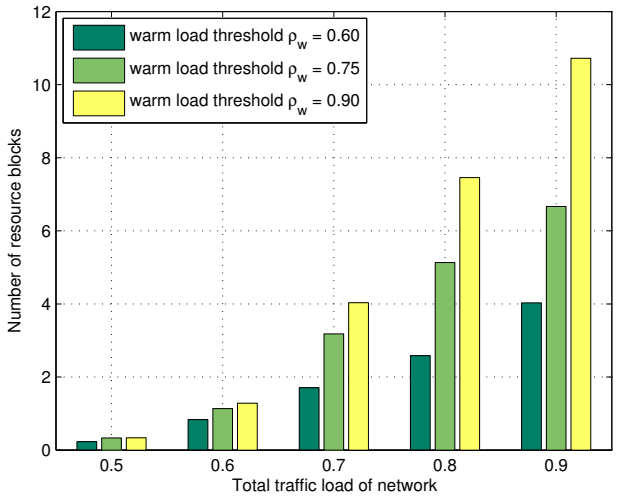


Figure 5.9: Standard deviation of the outage of balanced network with different warm load thresholds ρ_w, and $\Delta p = 1$ dB

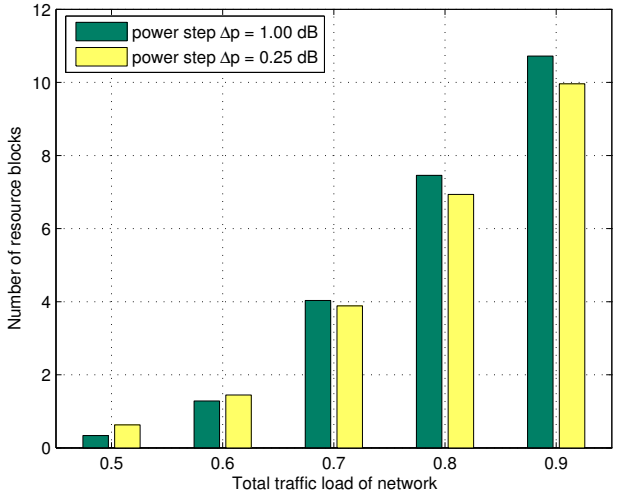


Figure 5.10: Standard deviation of the outage of balanced network with different power steps Δp, and $\rho_w = 0.9$

The Figure 5.10 shows that reducing the power step can increase a little the stability of the system. To increase the stability of the system, we should use the smaller warm load threshold ρ_w or the smaller power step Δp.

The purpose of the load balancing algorithm is to reduce as much as possible the outage of the network, and also should keep the network stable. Let U be the mean value of the number of resource blocks in outage that are reduced in the time frames after the convergence point. We introduce a parameter W, which is the ratio of standard deviation of the outage of the balanced network σ to U. We have $W = \sigma/U$.

A small value of W indicates the fluctuation of the outage of the balanced network is much smaller than the number of resource blocks in outage that are reduced. Therefore, the fluctuation can be neglected. In this case, the network has a good performance. A bigger value of W shows the fluctuation is notable when compared with the number of resource blocks in outage that are reduced. In this case, the network has a bad performance. The value of W is represented in the Table 5.3 with different values of the warm load threshold and the power step at different traffic loads.

Table 5.3: values of W

traffic load T	0.5	0.6	0.7	0.8	0.9
$\rho_w = 0.90$, $\Delta p = 1.00$ dB	0.0107	0.0193	0.0301	0.0404	0.0480
$\rho_w = 0.90$, $\Delta p = 0.25$ dB	0.0184	0.0218	0.0330	0.0387	0.0454
$\rho_w = 0.75$, $\Delta p = 1.00$ dB	0.0107	0.0175	0.0282	0.0314	0.0391
$\rho_w = 0.60$, $\Delta p = 1.00$ dB	0.0083	0.0161	0.0225	0.0320	0.0535

The very high or very low values of the warm load threshold ρ_w give the bad performance of the network. The very high ρ_w makes the network less stable while the very low ρ_w reduces the capacity gain of the network.

The power step Δp does not affect much to the performance of the system in long term when the traffic load requested by users are fixed. However, the smaller Δp reduces the convergence speed and reduces the performance of the network when the traffic load requested by users are fast varying.

The value of the warm load threshold ρ_w and the power step Δp should be controlled for each cell to keep the best performance and stability for the network.

5.6 Conclusion and Discussion

In this paper, we introduce a distributed load balancing algorithm for self organized wireless networks. This algorithm does not need information about the position of users, as well as does not need to modify the hysteresis parameter in the handover process. Our algorithm always provides sufficient power and resource blocks to allocate to users, and also guarantees the coverage of the network. This algorithm improves 2.3% capacity of the whole network when the demands of traffic load of users are varying very fast.

The simulation shows that the value of the warm load threshold ρ_w and the power step Δp should be carefully selected for each cell to keep the best performance and stability for the network. The higher value of the warm load threshold ρ_w gives the higher capacity gain. However, it makes the network less stable. In contrast, the lower value of ρ_w increases the convergence speed and stability of the network, but it gives the lower capacity gain. The power step Δp should be much smaller than the transmission power of the base station. However, a too small value of Δp reduces the convergence speed of the network and therefore reduces the capacity gain of the load balancing algorithm when the traffic load requested by users are fast varying.

Chapter 6

Conclusion

6.1 Main contributions

This work aimed at studying some applications of simplicial homology in wireless networks. The main contributions can be summarized as follows.

- **The construction of the Čech complex of wireless networks**

  The Čech complex captures exactly the topology of the wireless networks. We develop an algorithm to construct the Čech complex aimed for applications in wireless networks. The construction is based on the verification of the intersection of a group of given discs on the plane. The algorithm constructs the Čech complex from the lowest dimensional simplices until the highest ones. In the best of our knowledge, this is the first algorithm that can construct the Čech complex for a collection of discs that have different sizes. This algorithm is presented in centralized, parallel and distributed versions.

- **Simplicial homology based energy saving algorithms for wireless networks**

  Given a random deployment of wireless cells on the plane, the energy saving algorithms firstly maximize the coverage of these cells. All the cells are set to work with the maximal transmission power. The Čech complex is used as a tool to capture the topology of the network. Then, these algorithms reduce the transmission power as much as possible for each cell without modifying the topology of the network. Finally, the transmission power is minimized while the coverage is kept as maximal one. These algorithms are available in centralized, parallel and distributed versions. Based on our simulation, our algorithms can save at most 65% of system's maximal transmission power in polynomial time.

- **Distributed load balancing for wireless networks**

  We introduced a distributed algorithm to balance the traffic load for wireless networks. This algorithm increases the transmission power of the low traffic loaded cells and reduces the one of the overloaded cells under the coverage constraint. This is to redirect some users from overloaded cells to lower loaded cells by the handover protocol. The network guarantees both coverage and free resource blocks for users for their best condition of connection. This algorithm improves 2.3% capacity of the whole network when the demands of traffic load of users are varying very fast. However, the value of the warm load threshold ρ_w and the power step Δp should be carefully selected for each cell to keep the best performance for the network. The higher value of the warm load threshold ρ_w gives the higher capacity gain, but it makes the network less stable. The small power step Δp reduces the convergence speed of the algorithm. In the case that the users change their request of resource blocks very fast, the small power step Δp will reduce the capacity gain of the algorithm due to the slow convergence speed.

6.2 Future research directions

In the future, this work can be continued with applications for Self-Organized Networks. We consider a wireless network that is able to automatically optimize its deployment layout to adapt for the traffic load change. Most users use the wireless services in daylight time but they sleep at night. Therefore, the traffic load in daylight time is much higher than one in night time [51]. The network is aware of the traffic load change. It changes its layout to dense and small scale one to support higher traffic load in daylight time. However, it switches its layout to large scale one when the traffic load is reduced at night. Many base stations can be turned off to save energy at night and can be automatically turned on to work in daylight time when it is needed. The cells' size is automatically optimized for the traffic load change.

In this work, we use the Poisson point process to model the deployment of base stations. Although the Poisson point process is widely used to model the wireless networks, it locates each wireless cell independently with each other. It means that the correlation of positions of a cell and another is ignored. Similarly, the correlation of positions of users is not available. Therefore, it is not suitable to model the wireless networks with repulsion and attraction. The Ginibre point process is a determinantal point process that takes into account the repulsion and attraction. This work can also be investigated with the Ginibre point process for better simulation results.

Bibliography

[1] 3GPP TS 36.133 V13.2.0 (2016-01). http://www.3gpp.org/dynareport/36133.htm. (accessed 10/04/2016).

[2] Distributed and Parallel computing. http://wla.berkeley.edu/~cs61a/fa11/lectures/communication.html. (accessed 20/04/2016).

[3] Distributed computing. https://en.wikipedia.org/wiki/Distributed_computing. (accessed 20/04/2016).

[4] Ericsson mobility report. http://www.ericsson.com/res/docs/2015/mobility-report/ericsson-mobility-report-nov-2015.pdf. (accessed 10-04-2016).

[5] LTE Resource Guide. http://web.cecs.pdx.edu/~fli/class/LTE_Reource_Guide.pdf. (accessed 10/04/2016).

[6] Signal propagation and path loss models. http://web.stanford.edu/class/ee359/pdfs/lecture2.pdf. (Accessed on 03/03/2016).

[7] COST Action 231. Digital mobile radio towards future generation systems COST-231 final report. http://www.lx.it.pt/cost231/final_report.htm. (accessed 10/04/2016).

[8] Z. Al-Husseiny and P. Frenger. Enhancing lte energy performance with antenna muting and dynamic psi-omni configuration. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2015.

[9] 4G Americas. Integration of cellular and wi-fi networks. http://www.4gamericas.org/files/3114/0622/2546/Integration_of_Cellular_and_WiFi_Networks_White_Paper-_9.25.13.pdf. (Accessed on 10/04/2016).

[10] 4G Americas. Mobile broadband evolution towards 5g: Rel-12 & rel-13 and beyond. http://www.4gamericas.org/files/8914/3576/4557/

4G_Americas_Mobile_Broadband_Evolution_Toward_5G-Rel-12_Rel-13_June_2015x.pdf. (Accessed on 10/04/2016).

[11] M. Amirijoo, Z. Chai, P. Frenger, B. Olin, and J. Moe. Self-optimizing antenna muting - energy consumption and user thoughput analysis. In *2012 International Symposium on Wireless Communication Systems (ISWCS)*, pages 46–50, Aug 2012.

[12] J. G. Andrews. Interference cancellation for cellular systems: a contemporary overview. *IEEE Wireless Communications*, 12(2):19–29, April 2005.

[13] M. W. Arshad, A. Vastberg, and T. Edler. Energy efficiency improvement through pico base stations for a green field operator. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 2197–2202, April 2012.

[14] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun, and Ten H. Lai. Deploying wireless sensors to achieve both coverage and connectivity. In *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '06, pages 131–142, New York, NY, USA, 2006. ACM.

[15] Dimitris Bertsimas and John Tsitsiklis. Simulated annealing. *Statist. Sci.*, 8(1):10–15, 02 1993.

[16] Nicolas Boissin and Jean-Luc Lutton. A parallel simulated annealing algorithm. *Parallel Comput.*, 19(8):859–872, August 1993.

[17] Heejung Byun and Junglok Yu. Automatic handover control for distributed load balancing in mobile communication networks. *Wirel. Netw.*, 18(1):1–7, January 2012.

[18] T. Chen, Y. Yang, H. Zhang, H. Kim, and K. Horneman. Network energy saving technologies for green wireless access networks. *IEEE Wireless Communications*, 18(5):30–38, October 2011.

[19] King-Wai Chu, Yuefan Deng, and John Reinitz. Parallel simulated annealing by mixing of states. *J. Comput. Phys.*, 148(2):646–662, January 1999.

[20] Stefan Dantchev and Ioannis Ivrissimtzis. Efficient construction of the Čech complex. *Computers & Graphics*, 36(6):708 – 713, 2012. Joint Symposium on Computational Aesthetics (CAe), Non-Photorealistic Anima-

tion and Rendering (NPAR), and Sketch-Based Interfaces and Modeling (SBIM).

[21] V. De Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *Int. J. Rob. Res.*, 25(12):1205–1222, December 2006.

[22] Vin de Silva and Robert Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7:339–358, 2007.

[23] Vin de Silva, Robert Ghrist, and Abubakr Muhammad. Blind swarms for coverage in 2-D. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[24] P. Dotko, R. Ghrist, M. Juda, and M. Mrozek. Distributed computation of coverage in sensor networks by homological methods. *Appl. Algebra Eng., Commun. Comput.*, 23(1):29–58, April 2012.

[25] Lin Du, J. Bigham, and L. Cuthbert. Towards intelligent geographic load balancing for mobile cellular networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 33(4):480–491, Nov 2003.

[26] H. Edelsbrunner and S. Parsa. On the computational complexity of betti numbers: Reductions from matrix rank. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 152–160. SIAM, 2014.

[27] J. S. Engel and M. M. Peritsky. Statistically-optimum dynamic server assignment in systems with interfering servers. In *Decision and Control, 1972 and 11th Symposium on Adaptive Processes. Proceedings of the 1972 IEEE Conference on*, pages 177–177, Dec 1972.

[28] G. P. Fettweis, K. C. Chen, and R. Tafazoli. Green radio: Energy efficiency in wireless networks. *Journal of Communications and Networks*, 12(2):99–102, April 2010.

[29] Georg Fischer. Next-generation base station radio frequency architecture. *Bell Lab. Tech. J.*, 12(2):3–18, August 2007.

[30] P. Frenger, P. Moberg, J. Malmodin, Y. Jading, and I. Godor. Reducing energy consumption in lte with cell dtx. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5, May 2011.

[31] Robert Ghrist and Abubakr Muhammad. Coverage and hole-detection in sensor networks via homology. In *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.

[32] Andrea. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005.

[33] Daniel R. Greening. Parallel simulated annealing techniques. *Phys. D*, 42(1-3):293–306, June 1990.

[34] C. Han, T. Harrold, S. Armour, I. Krikidis, S. Videv, P. M. Grant, H. Haas, J. S. Thompson, I. Ku, C. X. Wang, T. A. Le, M. R. Nakhai, J. Zhang, and L. Hanzo. Green radio: radio techniques to enable energy-efficient wireless networks. *IEEE Communications Magazine*, 49(6):46–54, June 2011.

[35] Allen Hatcher. *Algebraic topology*. Cambridge University Press, 2001.

[36] H. Holma and A. Toskala. *LTE for UMTS - OFDMA and SC-FDMA Based Radio Access*. Wiley Publishing, 2009.

[37] H. Jiang and S. S. Rappaport. Cbwl: a new channel assignment and sharing method for cellular communication systems. *IEEE Transactions on Vehicular Technology*, 43(2):313–322, May 1994.

[38] Hua Jiang and S. S. Rappaport. Prioritized channel borrowing without locking: a channel sharing strategy for cellular communications. In *Global Telecommunications Conference, 1993, including a Communications Theory Mini-Conference. Technical Program Conference Record, IEEE in Houston. GLOBECOM '93., IEEE*, pages 276–280 vol.1, Nov 1993.

[39] Z. Jin, Z. Pan, N. Liu, W. Li, J. Wu, and T. Deng. Dynamic pico switch on/off algorithm for energy saving in heterogeneous networks. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2015.

[40] Dermot Kelly. Parallel and Distributed Systems. http://www.cs.nuim.ie/~dkelly/CS402-06/Introduction.htm. (accessed 20/04/2016).

[41] Scott Kirkpatrick. Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, 34(5):975–986, 1984.

[42] Henri Koskinen. On the coverage of a random sensor network in a bounded domain. In *in Proceedings of 16th ITC Specialist Seminar*, pages 11–18, 2004.

[43] Ngoc-Khuyen Le, P. Martins, L. Decreusefond, and A. Vergne. Construction of the generalized Čech complex. In *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, pages 1–5, May 2015.

[44] Ngoc-Khuyen Le, P. Martins, L. Decreusefond, and A. Vergne. Simplicial homology based energy saving algorithms for wireless networks. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 166–172, June 2015.

[45] T. A. Le and M. R. Nakhai. Possible power-saving gains by dividing a cell into tiers of smaller cells. *Electronics Letters*, 46(16):1163–1165, August 2010.

[46] C. Li, S. H. Song, J. Zhang, and K. B. Letaief. Maximizing energy efficiency in wireless networks with a minimum average throughput requirement. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1130–1134, April 2012.

[47] K. Y. Lin, J. Y. Chen, F. C. Ren, and C. J. Chang. Taps: Traffic-aware power saving scheme for clustered small cell base stations in lte-a. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2015.

[48] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan. Load balancing in downlink lte self-optimizing networks. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5, May 2010.

[49] J. Manssour, P. Frenger, L. Falconetti, S. Moon, and M. Na. Smart small cell wake-up field trial: Enhancing end-user throughput and network energy performance. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2015.

[50] James Munkres. *Elements of algebraic topology*. Perseus Books, 1984.

[51] Oliver Blume and Harald Eckhardt and Siegfried Klein and Edgar Kuehn and Wieslawa M. Wajda. Energy savings in mobile networks based on adaptation to traffic statistics.

[52] Esin Onbaşoğlu and Linet Özdamar. Parallel simulated annealing algorithms in global optimization. *Journal of Global Optimization*, 19(1):27–50.

[53] Monica Paolini. Beyond data caps - an analysis of the uneven growth in data traffic, 2011.

[54] D.Janaki Ram, T.H. Sreenivas, and K.Ganapathy Subramaniam. Parallel simulated annealing algorithms. *J. Parallel Distrib. Comput.*, 37(2):207–212, September 1996.

[55] E. M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, Apr 1999.

[56] Vin De Silva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54:2007, 2007.

[57] P. Skillermark and P. Frenger. Enhancing energy efficiency in LTE with antenna muting. In *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pages 1–5, May 2012.

[58] J. So, H. C. Jeon, and D. Ahn. Joint proportional fair scheduling for uplink and downlink in wireless networks. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–4, May 2011.

[59] I. Stojmenovic, A. P. Ruhil, and D. K. Lobiyal. Voronoi diagram and convex hull based geocasting and routing in wireless networks. In *Computers and Communication, 2003. (ISCC 2003). Proceedings. Eighth IEEE International Symposium on*, pages 51–56 vol.1, June 2003.

[60] J. Tang, D. K. C. So, E. Alsusa, K. A. Hamdi, and A. Shojaeifard. Resource allocation for energy efficiency optimization in heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 33(10):2104–2117, Oct 2015.

[61] Agilent Technologies. 3GPP LTE:Introducing Single-Carrier FDMA. http://cp.literature.agilent.com/litweb/pdf/5989-7898EN.pdf. (Accessed on 03/03/2016).

[62] A. Vergne, L. Decreusefond, and P. Martins. Reduction algorithm for simplicial complexes. In *INFOCOM, 2013 Proceedings IEEE*, pages 475–479, April 2013.

[63] A. Vergne, L. Decreusefond, and P. Martins. Simplicial homology for future cellular networks. *IEEE Transactions on Mobile Computing*, 14(8):1712–1725, Aug 2015.

[64] A. Vergne, I. Flint, L. Decreusefond, and P. Martins. Homology based algorithm for disaster recovery in wireless networks. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*, pages 685–692, May 2014.

[65] 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.

[66] F. Yan, P. Martins, and L. Decreusefond. Accuracy of homology based approaches for coverage hole detection in wireless sensor networks. In *2012 IEEE International Conference on Communications (ICC)*, pages 497–502, June 2012.

[67] Feng Yan, P. Martins, and L. Decreusefond. Connectivity-based distributed coverage hole detection in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–6, Dec 2011.

[68] Ming Zhang and T. S. P. Yum. Comparisons of channel-assignment strategies in cellular mobile telephone systems. *IEEE Transactions on Vehicular Technology*, 38(4):211–215, Nov 1989.

HOMOLOGIE SIMPLICIALE APPLIQUÉE AUX RÉSEAUX SANS FIL

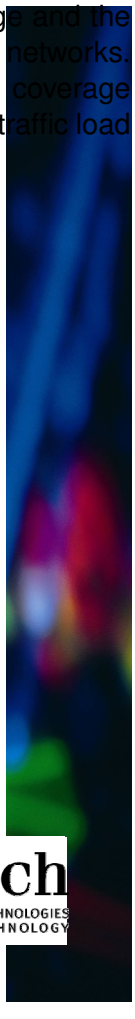
Ngoc-Khuyen LE

RESUME : Homologie simpliciale est un outil très efficace pour accéder à des informations importantes sur la topologie des réseaux sans fil, tels que : la couverture et la connectivité. Dans cette thèse, nous modélisons le réseau sans fil comme un déploiement aléatoire des cellules. Tout d'abord, nous introduisons un algorithme pour construire le complexe de Čech, qui décrit exactement la topologie du réseau. Ensuite, le complexe de Čech est utilisé dans des applications avancées. La première application est d'économiser l'énergie de transmission pour les réseaux sans fil. Cette application non seulement maximise la couverture de le réseau, mais réduit également la puissance de transmission. En même temps, la couverture et la puissance de transmission sont optimisées. La deuxième application est pour équilibrer la charge de trafic dans les réseaux sans fil. Cette application contrôle la puissance de transmission de chaque cellule dans le réseau, toujours sous contrainte de couverture. Avec la puissance d'émission contrôlée, les utilisateurs sont redirigés vers des cellules de charge plus faibles. Par conséquent, la charge du trafic est répartie entre les différentes cellules.

MOTS-CLEFS : homologie simpliciale ; réseaux mobiles ; économie d'énergie dans les réseaux mobiles.

ABSTRACT : Simplicial homology is a useful tool to access important information about the topology of wireless networks such as : coverage and connectivity. In this thesis, we model the wireless network as a random deployment of cells. Firstly, we introduce an algorithm to construct the Čech complex, which describes exactly the topology of the network. Then, the Čech complex is used in further applications. The first application is to save transmission power for wireless networks. This application not only maximizes the coverage of the network but also minimizes its transmission power. At the same time, the coverage and the transmission power are optimized. The second application is to balance the traffic load in wireless networks. This application controls the transmission power of each cell in the network, always under the coverage constraint. With the controlled transmission power, the users are redirected to connect to the lower traffic load cells. Consequentially, the balanced traffic load is obtained for the network.

KEY-WORDS : simplicial homology ; mobile networks ; energy saving in mobile networks.



TELECOM ParisTech

EDITE

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY