



HAL
open science

Contribution à des problèmes algorithmiques dans les graphes : applications aux réseaux de capteurs, et recherche de motifs

Valentin Pollet

► **To cite this version:**

Valentin Pollet. Contribution à des problèmes algorithmiques dans les graphes : applications aux réseaux de capteurs, et recherche de motifs. Autre [cs.OH]. Université Montpellier, 2019. Français. NNT : 2019MONT054 . tel-02485814

HAL Id: tel-02485814

<https://theses.hal.science/tel-02485814>

Submitted on 20 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**THÈSE POUR OBTENIR LE GRADE DE DOCTEUR
DE L'UNIVERSITE DE MONTPELLIER**

En Informatique

École doctorale : Information, Structures, Systèmes

Unité de recherche LIRMM UMR 5506

**Contribution à des problèmes algorithmiques dans les
graphes : applications aux réseaux de capteurs, et
recherche de motifs**

Présentée par Valentin POLLET

Le 3 Octobre 2019

Sous la direction de Jean-Claude KÖNIG et Vincent BOUDET

Devant le jury composé de

David Coudert, Directeur de Recherche, INRIA, COATI-I3S, Nice

Christian Laforest, Professeur à l'Université Clermont-Auvergne, LIMOS, Clermont-Ferrand

Cristina Bazgan, Professeur à l'Université Paris-Dauphine, LAMSADE, Paris

Marianne Huchard, Professeur à l'Université de Montpellier, LIRMM, Montpellier

Vincent Boudet, Maître de Conférences à Université de Montpellier, LIRMM, Montpellier

Jean-Claude König, Professeur à l'Université de Montpellier, LIRMM, Montpellier

Rapporteur

Rapporteur

Examinatrice

Examinatrice

Co-encadrant

Directeur



**UNIVERSITÉ
DE MONTPELLIER**

Contribution à des problèmes algorithmiques dans les graphes

Application aux réseaux de capteurs, et recherche de motifs

Valentin Pollet

Résumé

On propose, dans cette thèse, des résultats concernant trois problèmes algorithmiques dans les graphes.

Dans un premier temps, on étudie le problème de diffusion à distance 2 dans les réseaux ad-hoc sans fil en communication synchrone : le temps est découpé en étapes et les communications ont lieu simultanément à chaque étape. Le problème consiste, étant donné un réseau et une source, à planifier les communications des noeuds pour diffuser une information détenue par la source à distance 2 de celle-ci. La motivation est théorique : on cherche de « bonnes » bornes en fonction de la topologie du réseau. Ces bornes peuvent porter sur, par exemple, le nombre d'étapes pour compléter une diffusion à distance 2 ou encore le nombre de noeuds qu'il est possible d'informer en une étape. Les classes de réseaux considérées sont les graphes d'intervalles, d'intervalles circulaires, et de disques unitaires.

On présente ensuite des résultats sur une variante du problème de recherche de motifs colorés : le problème du plus grand sous-graphe connexe équilibré. Étant donné un graphe dont les sommets sont colorés avec deux couleurs, on en cherche le plus grand sous-graphe connexe contenant autant de sommets de chaque couleur. L'étude est ici purement théorique, et on se concentre sur la classification, au sens de la complexité, du problème dans des classes de graphes très restreintes : les graphes bipartis cubiques, les graphes bipartis de diamètre quatre et les graphes de diamètre trois. On montre que le problème reste difficile dans toutes ces classes de graphes, et sa non-approximabilité à facteur constant dans les graphes bipartis de degré maximum quatre.

Enfin, on présente une nouvelle approche pour un problème motivé par l'application : la maximisation de durée de vie dans les réseaux de capteurs. Ces réseaux consistent en un grand nombre de capteurs autonomes qui peuvent communiquer entre eux par des technologies sans-fil. Ces capteurs peuvent

être tributaires d'une batterie limitée, ce qui a fait naître de nombreux problèmes d'optimisation. On s'intéresse au problème de planifier les allumages et mises en veille de capteurs déployés sur une vaste zone contenant des points d'intérêts à surveiller. On désire qu'à tout instant l'ensemble des cibles soient surveillées (un capteur allumé proche d'elles) et l'ensemble de capteurs allumés pour ce faire doivent communiquer entre eux pour faire remonter l'information à une station de base. On propose une nouvelle approche de la tolérance aux pannes dans ce contexte, ainsi qu'une validation de notre solution par simulation.

Abstract

In this thesis, we give results on three algorithmical problems in graphs.

First, we study the Distance-2 Broadcast Scheduling problem in wireless ad-hoc networks. We assume that time is discretized into slots. At each slot, all the communication events take place simultaneously. Given a network and a source node detaining a piece of information, distance-2 broadcast scheduling consists in scheduling the communication events to spread the information to every node within distance 2 of the source. The study is rather theoretical : we aim at establishing "good" bounds depending on the structure of the network. For instance, we give upper bounds on the number of slots sufficient to complete broadcast, or lower bounds on the number of nodes that can be informed in a limited number of slots. We refine those bounds in various classes of networks characterized by the structure of the underlying graph.

We then tackle the Balanced Connected Subgraph problem : given a graph which vertices are colored either red or blue, find the largest connected subgraph containing as many red vertices as blue vertices. Our study focuses on the classification of the problem when restricted to "small" classes of graphs, namely bipartite cubic graphs, graphs of diameter three and bipartite graphs of diameter four. We show that the problem remains NP-complete in each of those classes. In addition, we prove it non-approximable within constant factor in the case of bipartite graphs of maximum degree four.

Last, we give a novel approach to fault tolerance for extending lifetime of wireless sensor networks. These networks are used for a variety of applications including surveillance and monitoring. They consist of large amounts of autonomous devices with sensing capabilities deployed to monitor vast areas.

Usually, these devices run on limited batteries and communicate wirelessly. Thus, numerous optimization problems have appeared. We study the problem of scheduling sensor activity in order to extend network lifetime. The goal is to ensure that at all time each target is monitored by at least one sensor, and the set of active sensors does relay the information to a specified communication base. We describe a new approach which addresses this problem with a fault-tolerant solution. We then validate our approach by simulation.

TABLE DES MATIÈRES

1	Introduction	1
1.1	Notions de complexité	2
1.1.1	Problèmes de décision	5
1.1.2	Problèmes d'optimisation	6
1.2	Comment approcher un problème?	7
1.3	Organisation des contributions	8
2	Définitions, outils, méthodes	11
2.1	Théorie des graphes	11
2.2	Algorithmes d'approximation	16
2.3	Méthodes exactes	16
2.3.1	Programmation linéaire en nombres entiers	16
2.3.2	Algorithme polynomial à paramètre fixé	22
3	Diffusion dans les réseaux ad-hoc sans fil	25
3.1	Introduction	25
3.1.1	Problématiques abordées	27
3.1.2	Travaux connexes	28
3.1.3	Organisation du chapitre	31
3.2	Cas des intervalles	31
3.3	Cas des intervalles circulaires	38
3.4	Cas des disques unitaires	44
3.4.1	Certaines instances nécessitent $O(\log n)$ étapes	46
3.4.2	$O(\log n)$ étapes suffisent	48
3.4.3	Maximiser la proportion d'informés en une étape	52
3.5	Conclusion	58
4	Recherche de sous-graphe connexe équilibré	61
4.1	Introduction	61
4.1.1	Notations	65

4.1.2	Organisation du chapitre	65
4.2	Complexité classique	66
4.2.1	Graphes bipartis de diamètre quatre	66
4.2.2	Graphes de diamètre trois	68
4.2.3	Graphes bipartis cubiques	72
4.3	Non-approximabilité à facteur constant	79
4.4	Algorithme polynomial à paramètre fixé	83
4.5	Conclusion	85
5	Maximisation de durée de vie de réseaux de capteurs sans fil	87
5.1	Introduction	87
5.1.1	Notations	90
5.1.2	Organisation du chapitre	91
5.2	Génération de colonnes	91
5.2.1	Formulation	91
5.2.2	Sous-problème de séparation	93
5.3	Tolérance aux pannes	96
5.3.1	Utilisation de 2-couvertures	96
5.3.2	Rôle de suppléant	98
5.3.3	Stratégie de récupération	100
5.4	Résultats expérimentaux	102
5.5	Conclusion, perspectives	106
6	Conclusion	111
6.1	De la diffusion à distance 2	111
6.1.1	Rappel du problème	111
6.1.2	Contributions	112
6.1.3	Perspectives	113
6.2	De la recherche de sous-graphe connexe équilibré	113
6.2.1	Rappel du problème	113
6.2.2	Contributions	114
6.2.3	Perspectives	114
6.3	De la maximisation de durée de vie	115
6.3.1	Rappel du problème	115
6.3.2	Contributions	115
6.3.3	Perspectives	116

CHAPITRE 1

INTRODUCTION

La théorie de la complexité permet de classifier les différents problèmes algorithmiques en fonction de leur difficulté de résolution. Les problèmes peuvent être dits de **décision** lorsqu'il s'agit de répondre **oui** ou **non** à une question, ou d'**optimisation** lorsque la résolution implique l'optimisation d'un critère et de donner une solution. Ce distinguo est primordial puisque les problèmes sont classifiés différemment selon leur nature.

Outre permettre de classifier et hiérarchiser les problèmes, la théorie de la complexité introduit les notions de **réductibilité** et de **complétude**. Assez intuitivement, un problème A est **réductible** à un problème B si dans tous les cas la résolution de A peut être ramenée à la résolution de B . La **complétude** d'un problème A appartenant à une classe \mathcal{C} implique la réductibilité de tout autre problème de la classe à A ; on dit alors que A est **\mathcal{C} -complet**. Un problème complet d'une classe est donc au moins aussi difficile à résoudre que tous les autres problèmes de la classe.

Les classes de problèmes de décision **P** et **NP** forment la pierre angulaire de la théorie de la complexité, dans la mesure où de nombreux résultats reposent sur des hypothèses les concernant. On dit qu'un problème de décision appartient à la classe **P** s'il existe un algorithme résolvant toute instance du problème en temps raisonnable. On parle de temps raisonnable lorsqu'il est polynomial en la taille de la donnée. Un problème appartient à **NP** lorsque l'on peut certifier qu'une solution à une instance satisfait bien aux contraintes du problème en temps polynomial en la taille de l'instance.

Les problèmes **NP-complets**, les plus difficiles de la classe **NP**, sont parmi les plus étudiés puisqu'ils sont suspectés de ne pas être résolubles efficacement sur les machines déterministes au nombre d'états dénombrable que sont les ordinateurs actuels. Les méthodes de résolution ne sont, dans ce cas, pas

exécutables en temps polynomial et on dit d'un tel problème qu'il n'est pas **tractable**. À l'opposé, les problèmes de la classe **P**, admettent tous des algorithmes de résolution polynomiaux, aussi ces problèmes sont-ils considérés comme faciles à résoudre.

Pour classer un problème d'optimisation, on s'intéresse généralement à la variante décisionnelle qui lui est associée. Par exemple, si A est un problème de minimisation d'un coût, la variante décisionnelle consiste à décider s'il existe une solution de coût inférieur ou égal à un k donné. Naturellement, un problème d'optimisation dont la variante décisionnelle est **NP-complète** ne saurait être tractable, et on peut alors s'intéresser aux algorithmes d'approximation avec garantie de performance pour le résoudre. Ces algorithmes sont généralement polynomiaux et fournissent une solution qui n'est pas optimale mais dont la qualité est garantie par un ratio entre le coût de la solution approchée et le coût d'une solution optimale. On parle alors de ratio d'approximation.

Le ratio d'approximation permet de hiérarchiser les problèmes d'optimisation. Si un problème d'optimisation est approximable à ratio constant, le problème appartient à la classe **APX**. Si en revanche tous les algorithmes approchés exhibables pour un problème ont un ratio qui dépend de la taille de l'entrée, le problème n'est plus **APX** mais peut, par exemple, être **Log-APX** lorsque la dépendance est logarithmique. Il peut également exister un schéma d'approximation polynomial pour un problème, dans ce cas il appartient à la classe **PTAS**. Un schéma d'approximation est un algorithme approché dont le ratio peut être arbitrairement proche de 1, et donc la solution qu'il retourne arbitrairement proche de l'optimal, mais dont le temps d'exécution dépend de cette proximité. Naturellement, un problème **PTAS** est plus facilement approchable qu'un problème **APX**, qui est lui-même plus facilement approchable qu'un problème **Log-APX**. En ce sens apparait la hiérarchie des approximations, et l'on peut par exemple montrer, sous l'hypothèse que $P \neq NP$, qu'un problème ne peut pas appartenir à **APX** ou à **PTAS**, et ainsi établir une hiérarchie des problèmes d'optimisation.

1.1 Notions de complexité

La résolution de problèmes informatiques se fait au moyen d'algorithmes. On évalue généralement la performance d'un algorithme en mesurant sa

complexité temporelle et spatiale dans le pire des cas. L'étude du cas moyen peut sembler être une mesure plus naturelle, mais s'avère souvent être complexe, notamment parce que la distribution des données n'est pas connue en avance. Dans le cadre de cette thèse, on étudiera essentiellement la complexité temporelle des algorithmes. La complexité temporelle d'un algorithme est généralement évaluée en définissant un type d'opération (addition, comparaison, etc.) comme étant élémentaire et en comptant le nombre de ces opérations élémentaires qui sont effectuées lors d'une exécution de l'algorithme.

Définition 1.1 : Complexité temporelle

Soit \mathcal{A} un algorithme pour un problème \mathcal{P} . On appelle **complexité temporelle** de \mathcal{A} la fonction donnant le nombre maximal d'opérations élémentaires que \mathcal{A} devra effectuer pour résoudre une instance de \mathcal{P} .

De plus, on s'intéressera plutôt à l'ordre de grandeur de la complexité, ainsi la notation O sera employée.

Définition 1.2 : Notation O

Soient f et g deux fonctions de \mathbb{N} dans \mathbb{R} . On dit que f est un « grand O » de g , noté $f = O(g)$ si : $\exists c, \exists n_0$ tels que $\forall n > n_0, f(n) \leq c \times g(n)$. Autrement dit, g borne asymptotiquement f à un facteur constant près.

Cette notation permet d'éliminer les constantes multiplicatives et additives dans l'expression des fonctions de complexités. On parle alors de complexité, dans le pire des cas, en $O(f(n))$ où n est une variable représentant la taille d'une instance et f une fonction calculable. f détermine l'efficacité de l'algorithme. Si f est un polynôme l'algorithme est efficace, et son efficacité va dépendre du degré du polynôme. Si f est une exponentielle, il l'est bien moins.

Si ce n'est pas précisé, un temps « exponentiel » ou « polynomial » dépendra de la taille de l'instance. Ainsi lorsqu'on parlera de résoudre un problème en temps exponentiel, il sera sous entendu qu'il existe un algorithme permettant de le résoudre avec une complexité temporelle exponentielle en la taille de l'instance.

En théorie de la complexité, les problèmes sont caractérisés par les classes auxquelles ils appartiennent. Ces classes sont des ensembles de problèmes

dont le temps ou l'espace nécessaire à leur résolution sont similaires. Deux notions sont essentielles à la classification par les temps de résolution. La première est la **réduction en temps polynomial**.

Définition 1.3 : Réduction

Une **réduction** d'un problème \mathcal{A} vers un problème \mathcal{B} est une fonction calculable permettant de transformer toute instance de \mathcal{A} en une instance de \mathcal{B} telle que \mathcal{A} est positive si et seulement si \mathcal{B} est positive. On dit alors que \mathcal{A} se réduit à \mathcal{B} . Si la transformation peut être calculée en temps polynomial, la réduction est dite **polynomiale**, et la taille de l'instance résultante est polynomialement bornée par la taille de l'instance d'origine.

Si une réduction polynomiale existe d'un problème \mathcal{A} à un problème \mathcal{B} , alors \mathcal{B} est au moins aussi difficile à résoudre que \mathcal{A} puisque toute instance de \mathcal{A} peut être transformée en temps polynomial à une instance de \mathcal{B} . On peut vérifier que si \mathcal{B} est un problème résoluble en temps polynomial, alors \mathcal{A} l'est aussi. La contraposée est donc valide : si \mathcal{A} n'est pas résoluble en temps polynomial, alors \mathcal{B} ne l'est pas non plus. En utilisant les réductions polynomiales, on peut définir la **\mathcal{C} -difficulté** d'un problème.

Définition 1.4 : \mathcal{C} -difficulté

Soient \mathcal{P} un problème et \mathcal{C} une classe de complexité. \mathcal{P} est **\mathcal{C} -difficile** si et seulement si tout problème de \mathcal{C} se réduit à \mathcal{P} en temps polynomial.

Au vu de la définition de réduction polynomiale, les problèmes \mathcal{C} -difficiles sont des problèmes au moins aussi difficiles à résoudre que les problèmes \mathcal{C} . La \mathcal{C} -difficulté dénote de la difficulté de résolution d'un problème par rapport à la classe \mathcal{C} . Si de plus le problème appartient à \mathcal{C} , on arrive à la notion de **complétude**.

Définition 1.5 : \mathcal{C} -complétude

Soient \mathcal{P} un problème et \mathcal{C} une classe de complexité. \mathcal{P} est **\mathcal{C} -complet** si et seulement si \mathcal{P} est \mathcal{C} -difficile et \mathcal{P} appartient à \mathcal{C} .

1.1.1 Problèmes de décision

Les classes de problèmes de décision permettent d'établir une hiérarchie en terme de difficulté de résolution. Les classes habituellement considérées sont **P** et **NP**, qui peuvent être définies comme il suit.

Définition 1.6 : Classe P

Un problème de décision appartient à la classe **P** si et seulement si il existe un algorithme polynomial pour le résoudre.

Définition 1.7 : Classe NP

Un problème de décision \mathcal{P} appartient à la classe **NP** si et seulement s'il possède un **vérifieur** polynomial. On appelle **vérifieur** pour \mathcal{P} un algorithme V à réponse booléenne, tel que pour toute instance x positive de \mathcal{P} , il existe un c tel que $V(x, c)$. c est appelé le certificat de x . De manière équivalente, il est possible de vérifier en temps polynomial qu'une instance est positive lorsqu'une solution est fournie.

L'inclusion de la classe **P** dans la classe **NP** est triviale puisque s'il est possible de résoudre une instance en temps polynomial, vérifier qu'une solution donnée est bonne ne pourra pas être plus long que la résolution elle-même, ainsi un problème dans **P** est aussi dans **NP**. En revanche, on ne sait toujours pas si l'inclusion est stricte.

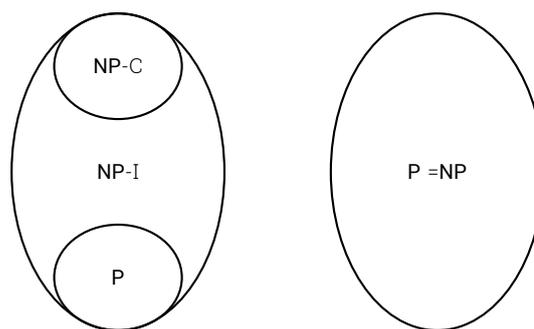


Figure 1.1 – $P \neq NP$ ou $P = NP$?

Bien que le problème soit toujours ouvert, puisque aucune preuve de l'inclusion stricte ou de l'égalité n'a encore été vérifiée, la communauté scientifique

s'accorde à dire que l'inclusion est stricte. Dans ce cas, les problèmes **NP**-complets ne pourraient être résolus en temps polynomial. Cette hypothèse est cruciale puisqu'il existe des problèmes **NP**-complets, qui sont donc au moins aussi difficiles à résoudre que tous les autres problèmes de **NP**. Le premier problème à avoir été montré **NP**-complet est celui de la satisfiabilité de formule par Cook [27]. Par la suite, la complétude d'une multitude de problèmes a pu être montrée, en utilisant les **réductions en temps polynomial**. Les premiers catalogues de problèmes **NP**-complets ont été proposés par Karp [50] puis Garey et Johnson [40]. De plus, si $\mathbf{P} \neq \mathbf{NP}$ alors il existe des problèmes de **NP** qui ne sont ni dans **P**, ni **NP**-complets [54]; ils forment la classe **NP**-intermédiaire (voir figure 1.1).

1.1.2 Problèmes d'optimisation

Les problèmes d'optimisation consistent à déterminer le maximum ou le minimum d'une fonction soumise à des contraintes. Pour déterminer la difficulté d'un tel problème, on considère sa variante décisionnelle.

Définition 1.8 : Variante décisionnelle

Soit \mathcal{P} un problème d'optimisation d'une fonction f , qui à toute solution x d'une instance I de \mathcal{P} associe un réel $f(x)$. Si \mathcal{P} est un problème de maximisation, sa variante décisionnelle est la suivante.

\mathcal{P} -décisionnel

Donnée : I instance de \mathcal{P} , $k \in \mathbb{R}$.

Question : Existe-t-il une solution x de I telle que $f(x) \geq k$?

Si \mathcal{P} est un problème de minimisation, on a $f(x) \leq k$ dans la question.

La variante décisionnelle étant un problème de décision, il peut appartenir aux classes **P** et **NP**. Si sa variante décisionnelle est un problème **NP**-complet, alors le problème d'optimisation est considéré comme difficile à résoudre. À l'inverse, si sa variante décisionnelle est un problème polynomial, alors le problème d'optimisation sera généralement facile à résoudre en résolvant la variante décisionnelle pour un petit nombre de valeurs possibles de la fonction à optimiser.

1.2 Comment approcher un problème ?

Lors de l'étude d'un problème, différents angles d'approches peuvent être adoptés en fonction de ce qui nous amène à l'étudier. On peut vouloir résoudre le problème pour répondre à une problématique « réelle ». On peut, à l'opposé, simplement désirer étudier le problème pour son intérêt fondamental. Dans tous les cas, la classification du problème d'un point de vue de la complexité permet de mieux nous situer et de décider comment aborder le problème.

On peut être confronté à un problème de la classe **P**. S'il est nécessaire de résoudre un tel problème, alors on disposera d'algorithmes souvent efficaces pour le résoudre. Si l'intérêt du problème est plus d'ordre théorique, on pourra chercher à améliorer la complexité des algorithmes ou encore à prouver l'inexistence d'algorithmes d'une complexité donnée (par exemple le problème du tri de listes ne peut être résolu, en général, avec une complexité linéaire).

Si le problème est **NP**-difficile, les choses se corsent. Résoudre un problème **NP**-difficile est, par définition, difficile, mais ledit problème peut être la seule modélisation d'une problématique, disons industrielle, dont on dispose. Dans ce cas, plusieurs approches sont possibles pour un problème d'optimisation :

- Les méthodes exactes (Programmation linéaire, algorithmes **FPT** ...).
- Les heuristiques, qui consistent à modifier un algorithme exact en **orientant** la recherche. La complexité est améliorée, mais la solution que l'algorithme heuristique renvoie peut être arbitrairement mauvaise.
- Les méta-heuristiques, comme les algorithmes de recherche locale, qui sont des schémas de résolution génériques pouvant être adaptés à différents problèmes. Là encore, les temps d'exécution peuvent être meilleurs que des résolutions exactes, mais les solutions trouvées peuvent, là encore, être arbitrairement mauvaises.
- Les algorithmes d'approximation en temps polynomial. Dans ce cas, les algorithmes sont spécifiques à chaque problème et retournent une solution dont le rapport à l'optimal est garanti par un ratio.

Le choix de la méthode va bien sûr dépendre des attentes. Si on souhaite trouver des solutions à des instances de grande taille rapidement, on se tournera plutôt vers les méta-heuristiques ou les algorithmes d'approximations qui ont généralement des complexités polynomiales. Si en revanche les

instances sont assez petites ou le temps à disposition long, on pourra se tourner vers les méthodes exactes qui fournissent des solutions optimales.

Dans d'autres cas, le désir de résoudre un problème n'est pas ce qui pousse à l'étudier. On peut simplement vouloir l'étudier pour enrichir la classification des problèmes existante. Par exemple, le problème de satisfiabilité de formule (SAT) s'est vu être enrichi de toutes sortes de contraintes : restriction de la taille des clauses (k -SAT), restriction du nombre d'occurrence des variables (SAT- k), graphe sous-jacent planaire (PLANAR SAT), une variable vraie par clause (1-in- k -SAT), etc. Bien entendu, on peut considérer des paires ou triplets de variantes en même temps pour obtenir encore de nouvelles variantes : planaire, clauses de taille k et occurrence p donne PLANAR k -SAT- p . Des dizaines de variantes de SAT ont ainsi été imaginées et classifiées. Il ne fait aucun doute que nombre d'entre elles n'ont d'intérêt que la classification, et l'engouement pour le problème n'en a pourtant pas pâti.

On peut expliquer cela par le fait que les études théoriques de classification créent une certaine inertie en recherche. Par exemple, pour montrer qu'un problème est **NP**-difficile on utilise souvent des réductions qui transforment un problème déjà existant en notre problème dont on souhaite montrer la difficulté. Bien entendu, par définition, on peut réduire SAT à tout problème **NP**-difficile, mais la construction peut être loin d'être évidente. Plus le nombre de problèmes classifié est grand, plus on dispose de points de départ pour une réduction. Une fois la réduction trouvée, notre problème s'ajoute à ceux déjà classifié et augmente le nombre de points de départ. À cela s'ajoute le fait qu'une réduction polynomiale est en fait un algorithme, à chaque problème classifié on facilite donc les prochaines classifications et potentiellement la recherche d'algorithmes.

1.3 Organisation des contributions

Dans cette thèse, on expose l'étude de trois problèmes. L'angle d'approche pour chaque problème diffère, et dépend des raisons qui ont motivé l'étude.

D'abord, le problème de la diffusion à distance 2 est examiné. Sous des hypothèses de communication fortes, on cherche à planifier les communications dans un réseau pour distribuer une information à distance 2 d'une source. La motivation ici est d'ordre théorique. Le problème de diffusion est largement étudié, et de nombreux algorithmes existent pour le résoudre rapidement

et efficacement. Dans notre cas, on cherche à restreindre les instances du problème de diffusion que l'on considère. Pour chaque restriction, on espère établir des bornes supérieures et inférieures pour différentes problématiques. En un sens, l'intérêt que l'on porte au problème est d'ordre topologique : établir des bornes supérieures et inférieures en fonction de la topologie du réseau considéré.

Le second problème étudié est un cas particulier du problème de recherche de motifs colorés : la recherche de sous-graphes connexes équilibrés en couleurs dans des graphes bicolores. Étant donné un graphe bicolore, on cherche le plus grand sous-graphe connexe contenant autant de sommets de chaque couleur. On est confronté dans ce cas à un problème relativement difficile dans la mesure où il est **NP**-complet et non-approximable dans des classes de graphes très restreintes. L'intérêt que l'on porte à ce problème est ici purement théorique, et les résultats proposés s'inscrivent dans un élan de classification : quand le problème devient-il « facile » à résoudre ?

Enfin, le troisième problème, davantage motivé par l'application, est celui de la maximisation de durée de vie dans les réseaux de capteurs. Dans ce cas, les applications concrètes du problème sont nombreuses et le besoin de le résoudre est réel. De nombreux algorithmes efficaces existent pour la résolution, mais sont souvent impuissants face aux imprévus qui peuvent apparaître dans une application réelle. On introduit donc une nouvelle approche de l'ordre, presque, du protocole pour améliorer la tolérance aux pannes de ces réseaux, ainsi qu'une méthode exacte pour résoudre le problème proposé.

Le chapitre 2 rappelle quelques définitions et méthodes de théorie des graphes, algorithmique, programmation linéaire et tractabilité à paramètre fixé. Le chapitre 3 est dédié à l'étude du problème de diffusion à distance 2. Le chapitre 4 se consacre aux résultats de complexité sur le problème du sous-graphe connexe équilibré. Enfin, notre approche « nouvelle » de la tolérance aux pannes pour la maximisation de durée de vie dans les réseaux de capteurs est présentée dans le chapitre 5.

CHAPITRE 2

DÉFINITIONS, OUTILS, MÉTHODES

Contenu

2.1. Théorie des graphes	11
2.2. Algorithmes d'approximation	16
2.3. Méthodes exactes	16
2.3.1 Programmation linéaire en nombres entiers	16
2.3.2 Algorithme polynomial à paramètre fixé	22

2.1 Théorie des graphes

Définition 2.1 : Graphe

Soit V un ensemble. Soit $\mathcal{R} : V \times V \rightarrow \{0, 1\}$ un prédicat binaire entre les éléments de V . $\mathcal{R}(x, y) = 1$ si x est en relation avec y , 0 sinon. Soit $A = \{(x, y) : \mathcal{R}(x, y) = 1\}$ l'ensemble des couples de V en relation. Le couple (V, A) est appelé un **graphe**, V étant son ensemble de sommets et A son ensemble d'arcs. Un graphe **non-orienté** encode un prédicat symétrique et les arcs sont appelés **arêtes**, sinon il est **orienté**. Si le prédicat est vrai pour tout couple, le graphe est **complet**.

Une première définition importante est celle de sous-graphe induit. On appelle sous-graphe la restriction d'un graphe à un ensemble de ses sommets

ou d'arcs.

Définition 2.2 : Sous-graphe induit

Soit $G = (V, A)$ un graphe. Soit $X \subset V$ un ensemble de sommets. Le sous-graphe induit par X est le graphe $H = (X, E(X))$ tel que $E(X) = \{a = (x, y) \in A \mid x \in X \text{ et } y \in X\}$. Autrement dit H est la restriction de G aux sommets de X , on le note $\mathbf{G[X]}$. (Figure 2.1)

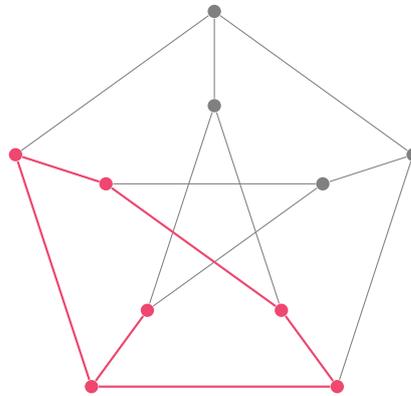


Figure 2.1 – Exemple de sous-graphe induit. Les sommets roses induisent les arêtes roses dans le graphes en gris.

Définition 2.3 : Chemin

Soient $G = (V, A)$ un graphe orienté, et x et y deux sommets distincts de G . Un chemin de longueur k de x à y dans G est un ensemble d'arcs (a_1, a_2, \dots, a_k) tel que, en notant $a_i = (u_i, v_i)$:

- $u_1 = x$ et $v_k = y$;
- pour tout $i \in \{1, \dots, k - 1\}$, $v_i = u_{i+1}$

S'il existe un chemin de x à y , on dit que y est **accessible** depuis x . Dans le cas non-orienté si y est accessible depuis x alors y est accessible depuis x . (Figure 2.2)

Définition 2.4 : Connexité

Soient $G = (V, A)$ un graphe non-orienté. G est connexe si pour toute paire de sommets distincts x et y , x est accessible depuis y . (Figure 2.2)

Définition 2.5 : Distance

Soient $G = (V, A)$ un graphe orienté, et x et y deux sommets distincts de G tels que y est accessible depuis x . La **distance** de x à y correspond au nombre minimum d'arcs dans un chemin de x à y , on note $d(x, y)$ cette distance. Un chemin de longueur $d(x, y)$ est appelé un **plus court chemin** de x à y . (Figure 2.2)

Définition 2.6 : Diamètre

Soit $G = (V, A)$ un graphe non-orienté connexe. La diamètre de G , $D(G)$, correspond à la distance la plus grande séparant deux de ses sommets, ou encore la longueur du plus long des plus courts chemins dans G . Formellement : $D(G) = \max_{(u,v) \in V \times V} d(u, v)$.

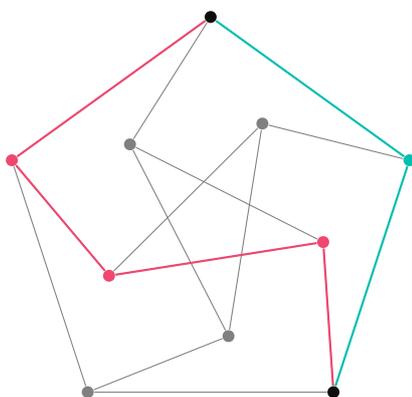


Figure 2.2 – Exemple de chemins et distance. Les arêtes roses forment un chemin de longueur 4 entre les sommets noirs. Les arêtes en bleu s'arcelle forment un chemin de longueur 2. Comme ils ne sont pas directement reliés, la distance entre les sommets noirs est 2. On peut également noter que ce graphe est de diamètre 2 : toute paire de sommets peut être reliée par un chemin de longueur 2.

Définition 2.7 : Voisinage

Soit $G = (V, A)$ un graphe non-orienté et x un de ses sommets. Le voisinage de x est l'ensemble des sommets reliés à x par une arête, on le note $N(x)$, formellement : $N(x) = \{y \in V : \{x, y\} \in A\}$.

Dans le cas orienté, on distingue le voisinage sortant, $N^+(x)$ comme l'ensemble des sommets accessibles depuis x par un arc et $N^-(x)$ l'ensemble des sommets. Formellement :

$$N^+(x) = \{y \in V : (x, y) \in A\} \text{ et } N^-(x) = \{y \in V : (y, x) \in A\}$$

On peut également définir les voisinages fermés, notés $N[x]$, $N^+[x]$ et $N^-[x]$ qui correspondent aux voisinages auxquels on a ajouté le sommet x .

Définition 2.8 : Degré

Soit $G = (V, A)$ un graphe non-orienté et x un de ses sommets. Le degré de x correspond à la taille de son voisinage : $d(x) = |N(x)|$.

Dans le cas orienté, on distingue le degré sortant, $d^+(x) = |N^+(x)|$ et le degré entrant $d^-(x) = |N^-(x)|$.

On peut également définir les degrés fermés, notés $d[x]$, $d^+[x]$ et $d^-[x]$ qui correspondent aux tailles des voisinages fermés.

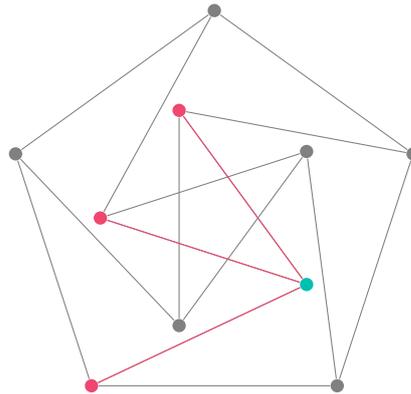


Figure 2.3 – Exemple de voisinages et degrés. Le sommet bleu sarcelle a pour voisins les sommets roses. Il est donc de degré 3.

Définition 2.9 : Coloration

Soient $G = (V, A)$ un graphe non-orienté et k un entier. Une k -coloration de G est une fonction c qui à tout sommet de G associe un entier (une couleur) compris entre 1 et k . Si pour toute arête de G les deux extrémités reçoivent des couleurs différentes, alors la coloration est **propre**, et on dit que le graphe est **k-coloriable**. (Figure 2.4)

Définition 2.10 : Graphe biparti

Soit $G = (V, A)$ un graphe orienté. G est biparti si on peut partitionner son ensemble de sommets en X et Y de sorte que pour tout arc ait une extrémité dans X et l'autre dans Y , formellement : $\forall (x, y) \in A, x \in X$ et $y \in Y$, ou $x \in Y$ et $y \in X$. Dans le cas non-orienté, la propriété d'être biparti est équivalente à la propriété d'être 2-coloriable. (Figure 2.4)

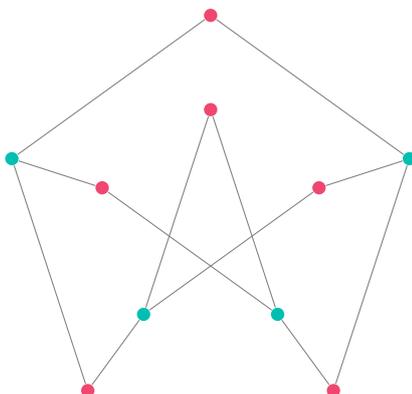


Figure 2.4 – Exemple de 2-coloration. Chaque sommet reçoit une couleur. Cette 2-coloration est propre puisque les arêtes ont toutes une extrémité de chaque couleur. Le graphe est donc 2-coloriable, et donc biparti.

2.2 Algorithmes d'approximation

Définition 2.11 : ρ -approximation

Soit \mathcal{P} un problème de maximisation (**resp.** minimisation). Pour toute instance I de \mathcal{P} , soit OPT_I la valeur d'une solution optimale. Soit $\rho \in]0, 1]$ (**resp.** $[1, +\infty[$). Alors un algorithme \mathcal{A} est une ρ -approximation pour le problème \mathcal{P} si pour toute instance I de \mathcal{P} :

- \mathcal{A} s'exécute en temps polynomial.
 - $\mathcal{A}(I)$ retournée par l'algorithme est bien une solution.
 - A_I , la valeur de $\mathcal{A}(I)$, vérifie $A_I \geq \rho \times OPT_I$ (**resp.** $A_I \leq \rho \times OPT_I$)
- ρ est appelé le **ratio d'approximation** de l'algorithme.

Ces algorithmes ont un intérêt double puisqu'ils sont d'une part polynomiaux et d'autre part ils fournissent une solution qui ne pourra être arbitrairement mauvaise, contrairement aux heuristiques. En pratique, les **méta**-heuristiques fournissent souvent de meilleures solutions, mais ne garantissent pas que ces solutions soient proches de l'optimal. Utiliser un algorithme d'approximation permet d'y remédier puisque si l'on considère une instance d'un problème de maximisation pour laquelle un algorithme $\frac{1}{3}$ -approché renvoie une solution de valeur 100, on est certain que l'optimal aura une valeur d'au plus 300. Si maintenant une **méta**-heuristique renvoie une solution de valeur 296 sur cette même instance, on aura la garantie que cette solution est bonne. À l'inverse, si l'heuristique renvoie une solution de valeur 30 on saura qu'il s'agit d'une mauvaise solution.

2.3 Méthodes exactes

2.3.1 Programmation linéaire en nombres entiers

Définition 2.12 : Programme linéaire

Soient $x \in \mathbb{R}^n$ un vecteur de variables réelles, $c \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$ des vecteurs de réels, et $A \in \mathbb{M}_{m,n}(\mathbb{R})$ une matrice $m \times n$ réelle. On restreint généralement les variables à être positives pour obtenir un programme

sous la forme suivante :

$$\begin{array}{ll} \max_x & c^T x \\ \text{sous} & Ax \leq b \\ & x \geq 0 \end{array}$$

Pour donner un exemple plus concret, considérons un boulanger pouvant préparer des pains au chocolat et des croissants. Pour concevoir un pain au chocolat (ou chocolatine, paraîtrait-il), le boulanger a besoin de 50g de farine, 15g de beurre et 5g de sucre. Pour élaborer un croissant, il a besoin de 40g de farine, 25g de beurre et 10g de sucre. Récemment finaliste de l'émission **Le meilleur boulanger**, notre boulanger, jouissant de sa notoriété nouvelle, a ouvert un établissement en plein coeur de Paris et se permet donc de pratiquer des tarifs quelque peu exorbitants : 10€ le pain au chocolat et 9€ le croissant. Pour sa défense, ils sont sensiblement plus gros que la moyenne. Nous sommes mercredi, avant-veille du chassé croisé estival et jour des enfants, aussi notre boulanger a-t-il décidé de fermer boutique à compter du lendemain - jeudi - pour profiter des routes dégagées afin de rejoindre sa récente acquisition immobilière sur la Côte d'Azur. Il lui reste 2kg de farine, 1kg de beurre, 500g de sucre et on l'a prévenu de l'arrivée d'une horde de touristes prêts à payer le prix fort pour déguster ses viennoiseries sensiblement plus grosses que la moyenne. Pour maximiser son profit, il fait appel au fleuron de l'optimisation mathématique : la programmation linéaire. Pour cela, il note x_1 le nombre de pains au chocolat qu'il va préparer, et x_2 le nombre de croissants. Il fixe ensuite aisément son objectif : maximiser $10x_1 + 9x_2$; qui correspond, à peu de choses près, à sa marge brute en vendant x_1 pains au chocolat et x_2 croissants. Il établit ensuite des contraintes traduisant les limites de ses stocks de farine, de beurre et de sucre :

$$50x_1 + 40x_2 \leq 2000 \quad (2.1)$$

$$15x_1 + 25x_2 \leq 1000 \quad (2.2)$$

$$5x_1 + 10x_2 \leq 500 \quad (2.3)$$

Son souci du détail et de la finition l'ayant porté jusqu'en finale du concours boulanger, il décide de mettre en forme son programme linéaire comme il

suit.

$$\begin{aligned}
 &\text{minimiser} && 10x_1 + 9x_2 \\
 &\text{sous} && 50x_1 + 40x_2 \leq 2000 \\
 &&& 15x_1 + 25x_2 \leq 1000 \\
 &&& 5x_1 + 10x_2 \leq 500 \\
 &&& x_1, x_2 \geq 0
 \end{aligned}$$

Il peut à présent résoudre son programme linéaire à l'aide de son solveur favori qui lui donne la solution $x_1 = \frac{200}{13}$, $x_2 = \frac{400}{13}$ pour un bénéfice total de $\frac{5600}{13}$ €. Après un petit calcul mental pour arrondir ces fractions, notre boulanger détermine qu'il va pouvoir préparer 15 pains au chocolat et 30 croissants pour gagner 420€, il décide donc de s'y mettre.

Une notion primordiale dans la résolution de programmes linéaires est la **dualité**. Étant donné un programme linéaire **primal**, on peut définir son programme **dual** qui donne une borne (supérieure ou inférieure selon que le primal soit une maximisation ou minimisation) sur la valeur du primal.

Définition 2.13 : Programme linéaire dual

$$\begin{array}{ll}
 \max_x & c^T x \\
 \text{sous} & Ax \leq b \\
 & x \geq 0
 \end{array}
 \qquad
 \begin{array}{ll}
 \min_y & b^T y \\
 \text{sous} & A^T y \geq c \\
 & y \geq 0
 \end{array}$$

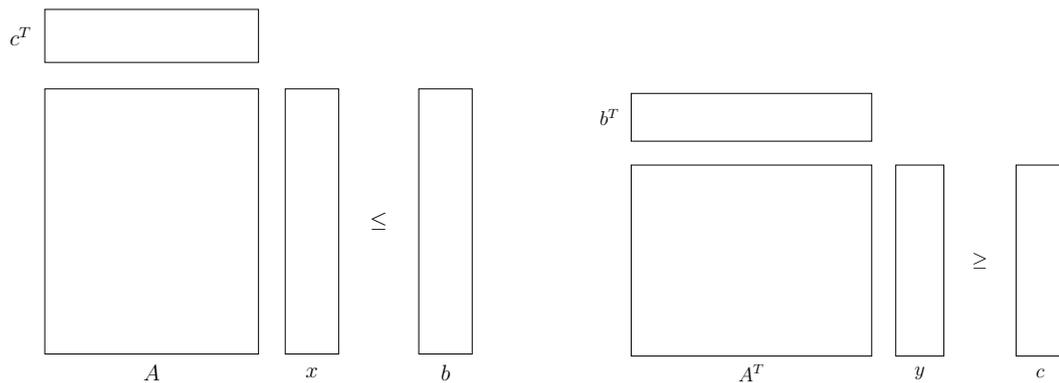


Figure 2.5 – Représentation d'un programme linéaire et de son dual sous la forme de matrices/vecteurs.

Beaucoup de raisonnement et techniques de programmation linéaire uti-

lisent le fait qu'à l'optimal, un primal et son dual bornés ont la même valeur objectif.

Bien que la résolution de programmes linéaires date de la fin des années 40, la complexité était au départ inconnue. Les algorithmes alors utilisés étaient en pratique efficaces. L'algorithme du simplexe, élaboré par George Dantzig en 1947 (pour la petite histoire du simplexe, se référer à [28]) a longtemps été, et est toujours, le plus utilisé pour résoudre des programmes linéaires, bien que Klee et Minty aient montré en 1972 que certaines instances prenaient un temps exponentiel à résoudre. En fait, le problème de résoudre un programme linéaire a depuis été montré comme étant polynomial par Grötschel, Lovász et Schrijver [44] en 1981, en utilisant la méthode des ellipsoïdes.

En revanche, si les variables d'un programme linéaires sont des nombres entiers, le problème est **NP-complet**. La variante portant sur des variables binaires (valant 0 ou 1) est d'ailleurs un problème **NP-complet** de Karp [50].

Reprenons l'exemple du boulanger. Il avait choisi de modéliser son problème par un programme linéaire dont les variables étaient réelles. La solution optimale était $x = \frac{200}{13}$ pains au chocolat et $y = \frac{400}{13}$ croissants. Ne pouvant, à son grand dam, vendre des portions de pains au chocolat, il avait du arrondir ces valeurs à l'inférieur pour garantir que son stock était respecté. On obtenait ainsi 15 pains au chocolat et 30 croissants. Cette solution entière n'est pas optimale, puisque fabriquer 16 pains au chocolat et 30 croissants respecte les contraintes et est strictement meilleure. On voit donc que la stratégie consistant à ignorer l'intégrité des variables, résoudre le programme linéaire et arrondir grossièrement n'est pas optimale (et heureusement puisque la résolution de programmes linéaires en nombres entiers est un problème **NP-complet**). Pour résoudre un programme linéaire en nombres entiers, on utilise souvent des algorithmes de « séparation et évaluation » (**Branch and bound** [55]) qui sont en fait des arbres de branchement : à chaque noeud de l'arbre on branche sur les valeurs que peut prendre une variable.

Coupe et branchement

Dans certains cas, il peut être intéressant de formuler un problème par un ensemble de contraintes en très grand nombre, par exemple exponentiel en le nombre de variables. Prenons comme exemple le problème de l'arbre de Steiner orienté. Étant donné un graphe $D = (V, A)$, un sommet particulier $r \in V$ appelé **racine**, et un ensemble de sommets $T \subset V$ appelés **terminaux**,

on cherche le plus petit ensemble d'arcs S tel qu'il existe un chemin de r à t n'empruntant que des arcs de S , pour tout terminal t . On va modéliser ce problème par un programme linéaire en nombres entiers. Pour cela, on va utiliser la notion de (s, t) -coupe dans les graphes orientés.

Définition 2.14 : (s, t) -coupe

Soit $D = (V, A)$ un graphe orienté. Soient s et t deux sommets distincts de D . Un ensemble d'arcs C est une (s, t) -coupe dans D s'il n'existe pas de chemin de s à t dans $D[A \setminus C]$.

On remarque qu'étant donné un graphe $D = (V, A)$, une racine r , et des terminaux T , si $S \subset A$ est un arbre de Steiner orienté pour r et T , alors pour tout terminal t , et pour toute (r, t) -coupe C dans D , S contient au moins un arc de D : $S \cap C \neq \emptyset$. En effet, s'il existait une (r, t) -coupe C dans D telle que S ne contienne pas d'arc de C , alors il n'existerait pas de chemin de r à t dans $D[A \setminus C]$. On peut maintenant formuler le problème de l'arbre de Steiner dirigé en programme linéaire, étant donné une instance $(D = (V, A), r, T)$ du problème, on définit un vecteur de variable $y \in \{0, 1\}^{|A|}$ tel que pour tout $a \in A$, $y_a = 1$ si l'arc appartient à la structure et 0 sinon. On cherche à minimiser la taille de la structure donc la somme des y_a et on obtient le programme linéaire suivant :

$$\begin{aligned} \min \quad & \sum_{a \in A} y_a \\ \text{s.t.} \quad & \sum_{a \in C} y_a \geq 1 \quad \forall C \subset A : \exists t \in T, C \text{ réalise une } (r, t)\text{-coupe} \\ & y_a \in \{0, 1\} \quad \forall a \in A \end{aligned}$$

Ce programme linéaire est loin d'être compact puisqu'il a un nombre de contraintes égal au nombre de coupes possibles, qui peut être exponentiel en le nombre d'arcs. Néanmoins, certaines contraintes sont redondantes et on utilise cette redondance pour résoudre cette formulation par un algorithme de génération de contraintes.

On démarre avec le programme linéaire relâché (sans contrainte d'intégrité) avec très peu de contraintes comme par exemple les coupes correspondant aux voisinages entrants des terminaux. On résout une première fois et on

obtient un vecteur solution y^* satisfaisant les contraintes initiales.

On va maintenant chercher des contraintes que le vecteur y^* ne satisfait pas. Si on en trouve, on les ajoute et sinon on sait que y^* vérifie l'ensemble de toutes les contraintes, y compris celles qui n'ont pas été ajoutée au programme linéaire. On dit que l'on cherche à séparer y^* , c'est le sous-problème de séparation. Dans le cas du problème de Steiner, étant donné un vecteur réel y^* , on cherche un ensemble d'arcs C réalisant une (r, t) -coupe pour un terminal t tel que $\sum_{a \in C} y_a^* < 1$.

Ce problème de séparation revient en fait à affecter à chaque arc $a \in A$ une capacité égale à y_a et à ensuite chercher une coupe de capacité strictement inférieure à 1. Il s'avère que ce problème est polynomial, et on peut le résoudre efficacement par un algorithme de flot grâce au théorème flot maximum-coupe minimum de Ford et Fulkerson [37].

Lorsqu'il n'est plus possible d'ajouter de contraintes, on peut lancer la résolution du programme linéaire avec les contraintes d'intégrité. À chaque solution entière trouvée, on résout à nouveau le problème de séparation pour déterminer si la solution entière est bien réalisable, si elle ne l'est pas on ajoute au modèle une contrainte bannissant cette solution (et ses symétries si possible) et on continue la résolution.

L'intérêt de cette approche pour le problème de Steiner, par rapport à une formulation plus compacte, est l'écart souvent très faible entre la solution relâchée et la solution entière, et relativement peu de sous-problèmes de séparation résolus, comme le montrent les expérimentations de Chopra et collab. [25]. En pratique, la solution relâchée est souvent entière et la procédure de séparation-évaluation n'est pas utilisée.

Génération de colonnes

La génération de colonnes est en quelque sorte la méthode duale de la génération de coupe. Prenons le programme linéaire (en nombres réels) suivant :

$$\begin{array}{ll} \max_x & c^T x \\ \text{sous} & Ax \leq b \\ & x \geq 0 \end{array}$$

Dans le cas d'une formulation propice à la génération de coupes, le programme a un grand nombre de contraintes ce qui se traduit par une matrice

A qui a un grand nombre de lignes. Si on considère le dual de ce problème :

$$\begin{array}{ll} \min_y & b^T y \\ \text{sous} & A^T y \geq c \\ & y \geq 0 \end{array}$$

On a alors, non plus un grand nombre de contraintes, mais un grand nombre de variables. La génération d'une coupe dans le primal correspond à la génération d'une colonne (comprendre ajouter une variable) dans le dual.

En utilisant un principe similaire aux algorithmes basés sur la génération de coupes, on peut baser des algorithmes sur la génération de colonnes. Le fonctionnement est le même, on résout une première fois le problème relâché avec un nombre de variables faible. On résout un sous-problème qui détermine s'il existe une variable qui peut améliorer la valeur de la fonction objectif, si oui on l'ajoute. Lorsqu'il n'est plus possible d'ajouter des variables améliorantes au programme, on lance la résolution avec contraintes d'intégrité. À la différence des méthodes basées sur la génération de coupes, ici le processus d'ajout de variables améliorantes doit être effectué à chaque noeud de l'arbre de branchement.

Dans le cas où les variables d'une formulation sont réelles, alors utiliser un algorithme de génération de colonnes est strictement équivalent à utiliser un algorithme de génération de coupes dans le dual.

2.3.2 Algorithme polynomial à paramètre fixé

Une solution élégante pour résoudre de manière exacte est l'utilisation d'algorithmes polynomiaux à paramètre fixé, qu'on appelle aussi des algorithmes **FPT**, **Fixed Parameter Tractable**. En général, la complexité d'un algorithme est exprimée comme la fonction d'une variable, cette variable étant la taille des données en entrée de l'algorithme. La théorie de la tractabilité à paramètre fixé enrichit l'expression de la complexité en ajoutant de nouvelles variables aux fonctions, permettant ainsi une expression plus fine des complexités. Prenons pour exemple le problème de la couverture des arêtes par les sommets d'un graphe.

Couverture par Sommets (CS)

Donnée : $G = (V, E)$ un graphe, $k \in \mathbb{N}$ un entier.

Question : Existe-t-il $S \subset V$ tel que $|S| \leq k$ et toute arête est incidente à au moins un sommet de S ?

Bien entendu ce problème est **NP**-complet dans le cas général [50], ce qui laisse présager qu'il sera difficile à résoudre. On peut néanmoins le résoudre efficacement dans le cas où la taille de la solution, k , est petite. Pour cela, considérons l'algorithme 1 qui repose sur le fait que pour toute arête d'un graphe donné, au moins une des extrémités appartient à une couverture de taille au plus k si elle existe.

Algorithme 1 : CS

Entrées : $G = (V, E)$ un graphe, $k \in \mathbb{N}$.

Résultat : 1 si G contient une couverture de taille au plus k , 0 sinon.

début

si $E = \emptyset$ **alors retourner** 1;

sinon si $k = 0$ **alors retourner** 0;

sinon

 Prendre une arête $xy \in E$;

retourner $\text{CS}(G - \{x\}, k - 1)$ **ou** $\text{CS}(G - \{y\}, k - 1)$

fin

fin

Une preuve de complétude et correction de l'algorithme est proposée par Niedermeier [58] dans son Invitation à la Complexité Paramétrée. Pour la complexité, dans le pire des cas, chaque appel entraîne deux appels récursifs. À chaque appel, la valeur de k décroît et $k = 0$ correspond à un appel terminal. On a donc au pire 2^k appels récursifs. À chaque appel, retirer un sommet du graphe en entrée peut être implémenté en $O(|E|)$. On obtient donc une complexité de l'algorithme en $O(|E| \times 2^k)$. En complexité « classique », on considérerait que k pourrait valoir $O(|V|)$, on aurait donc une complexité $O(|E| \times 2^{|V|})$.

On peut donner des définitions plus précises de ce qu'est un paramètre ou encore une paramétrisation.

Définition 2.15 : Problème paramétré

Soit \mathcal{P} un problème de décision. Une **paramétrisation** κ de \mathcal{P} est une fonction qui à toute instance x de \mathcal{P} associe un entier $\kappa(x)$ (la valeur du paramètre sur l'instance x) calculable en temps polynomial en $|x|$. Le couple (\mathcal{P}, κ) est appelé **problème paramétré**.

En reprenant l'exemple de la couverture par sommets, une instance est la donnée d'un graphe et d'un entier $k : x = (G, k)$. La fonction $\kappa : (G, k) \rightarrow k$ est une fonction calculable en temps polynomial, on l'appelle la paramétrisation naturelle.

On définit enfin deux classes, **FPT** et **XP**, qui, à l'instar de **P** et **NP** pour les problèmes de décision, permettent de classifier les problèmes paramétrés.

Définition 2.16 : Classe FPT

Soit (\mathcal{P}, κ) un problème paramétré. \mathcal{P} appartient à la classe **FPT** s'il existe un algorithme décidant toute instance x de \mathcal{P} en temps $f(\kappa(x)) \times g(|x|)$ où f est une fonction calculable quelconque et g un polynôme.

Définition 2.17 : Classe XP

Soit (\mathcal{P}, κ) un problème paramétré. \mathcal{P} appartient à la classe **XP** s'il existe un algorithme décidant toute instance x de \mathcal{P} en temps $f(\kappa(x)) + |x|^{g(\kappa(x))}$ où f et g sont des fonctions calculables quelconques.

L'algorithme donné en exemple pour la couverture par sommets a une complexité en $O(|E| \times 2^k)$, **Couverture par Sommets** paramétré par la taille de la solution (le paramètre naturel) appartient donc à **FPT**. Une complexité de la forme $O(|E|^k)$ aurait placé le problème dans la classe **XP**.

CHAPITRE 3

DIFFUSION DANS LES RÉSEAUX AD-HOC SANS FIL

Contenu

3.1. Introduction	25
3.1.1 Problématiques abordées	27
3.1.2 Travaux connexes	28
3.1.3 Organisation du chapitre	31
3.2. Cas des intervalles	31
3.3. Cas des intervalles circulaires	38
3.4. Cas des disques unitaires	44
3.4.1 Certaines instances nécessitent $O(\log n)$ étapes	46
3.4.2 $O(\log n)$ étapes suffisent	48
3.4.3 Maximiser la proportion d'informés en une étape	52
3.5. Conclusion	58

3.1 Introduction

On considère les réseaux sans fil dans lesquels les sommets ne peuvent communiquer que par **diffusion** : lorsqu'un sommet souhaite partager une information, tous les sommets à sa portée reçoivent cette information. La principale contrainte à prendre en compte lors de l'étude de la communication

dans de tels réseaux est le phénomène d'interférences. À un instant donné, si un sommet reçoit simultanément deux informations de deux sommets voisins distincts, alors on considère ces informations perdues.

Pour mieux appréhender le phénomène d'interférences, on considère que le temps est discrétisé, découpé en étapes (ou **slots**). On se place également sous l'hypothèse de communication synchrone : les étapes sont des fragments temporels élémentaires, et les communications qui auraient lieu durant le même fragment sont considérées comme simultanées. Le réseau est modélisé par un graphe non-orienté, et on suppose une connaissance parfaite du réseau à tout instant pour résoudre les problèmes de communication rencontrés.

On s'intéresse dans ce chapitre à des problématiques liées à la diffusion dans ces réseaux sans-fil. Étant donné un sommet source possédant une information, planifier les émissions des sommets du réseau pour que tous les sommets reçoivent l'information. Dans certain cas, on a une vue d'ensemble du réseau pour résoudre ce problème ; on parle alors d'algorithme de diffusion centralisé. Dans d'autres cas, les noeuds n'ont pas connaissance les uns des autres et on doit concevoir des algorithmes distribués : chaque noeud a un ensemble d'actions qu'il peut entreprendre en fonction des messages qu'il reçoit.

Le problème de diffusion à distance d est une variante du problème de diffusion dans laquelle une partie seulement du réseau doit être informée. Étant donné un sommet -- une source -- et une distance d (en nombre d'arcs), planifier les émissions pour que tous les sommets à distance d de la source reçoivent l'information. On dit qu'un sommet est **informé** lorsqu'à une étape il reçoit l'information d'**exactement un sommet** lui-même déjà informé. On s'intéresse dans la suite au problème de diffusion à distance 2 initialement étudié par Alon et collab. [2], Bar-Yehuda et collab. [3] puis, plus tard, Cogan et collab. [26]. Tous les sommets à distance 1 de la source peuvent être informés en une étape : la source émet seule.

Le problème consiste donc à planifier les émissions des sommets à distance 1 de la source pour informer tous les sommets à distance 2 strictement. On appellera **émetteurs** les sommets à distance 1 de la source et **récepteurs** les sommets à distance 2 strictement. On supposera que les récepteurs ne peuvent pas être utilisés pour informer d'autres récepteurs. Dans ces conditions, le problème de diffusion à distance 2 est simplifié : on peut supposer que le graphe induit par les sommets émetteurs et récepteurs est biparti. On appelle

ce graphe biparti (dans lequel sont ignorées les arêtes entre émetteurs et entre récepteurs) le **graphe biparti de diffusion**.

Diffusion à Distance 2

Donnée : $G = (X, Y, E)$ un graphe biparti de diffusion, $k \in \mathbb{N}$

Question : Existe-t-il k ensembles d'émetteurs $X_1, \dots, X_k \subset X$ tels que :

$$\forall y \in Y, \exists X_i : |N(y) \cap X_i| = 1 ?$$

Une solution à ce problème est un schéma de diffusion en k (au plus) étapes, qui correspond à k ensembles d'émetteurs. Les résultats annoncés dans ce qui suit ne tiennent pas compte de la première étape triviale lors de laquelle la source communique l'information à ses voisins. Ainsi, lorsque l'on parlera de compléter une diffusion à distance 2 en, par exemple, trois étapes, il s'agira en fait de trois étapes pour informer les récepteurs après réception de l'information par les émetteurs : quatre étapes au total.

On s'intéressera également au problème de la diffusion maximum en une étape qui consiste à maximiser le nombre de récepteurs informés en une étape d'émission.

Diffusion Maximum

Donnée : $G = (X, Y, E)$ un graphe biparti de diffusion, $p \in \mathbb{N}$

Question : Existe-t-il un ensemble d'émetteurs S tel que :

$$|\{y \in Y : |N(y) \cap S| = 1\}| \geq p ?$$

3.1.1 Problématiques abordées

On s'intéresse dans ce chapitre à deux aspects des problèmes de diffusion à distance 2 et de diffusion maximum : leur complexité et les bornes sur le nombre d'étapes ou de sommets informés en une étape. On s'est penché sur ces aspects des deux problèmes dans trois cas particuliers :

- Lorsqu'il est possible d'ordonner les récepteurs pour que les voisinages des émetteurs forment des intervalles.
- Lorsqu'il est possible d'ordonner les récepteurs pour que les voisinages des émetteurs forment des intervalles circulaires.

- Lorsque les voisinages des émetteurs sont donnés par un graphe de disques unitaires.

Notons que le cas des intervalles circulaires généralise le cas des intervalles et que les résultats obtenus sur les bornes seront les mêmes. On décrit tout de même les résultats qui ont été d'abord obtenus sur les intervalles puis ceux dans les intervalles circulaires. L'intérêt est double : les preuves dans le cas des intervalles circulaires utilisent des mécanismes similaires aux preuves dans le cas des intervalles ; et les complexités des algorithmes ou réductions polynomiales utilisées sont moindres dans le cas des intervalles.

3.1.2 Travaux connexes

Le modèle de communication retenu est classique : graphe sous-jacent non-orienté, temps discrétisé, communication synchrone et connaissance parfaite du réseau à tout instant. De nombreuses études sous ces hypothèses ont vu le jour, comme par exemple [3, 22, 23, 24, 30, 52].

Les travaux présentés dans ce chapitre sont principalement inspirés par Alon et collab. [2], Bar-Yehuda et collab. [3], Cogan et collab. [26], qui traitent du problème de diffusion à distance 2 dans des graphes sans structure particulière. Ils ont pu concevoir, indépendamment, des algorithmes polynomiaux produisant des schémas de diffusion en $O(\log^2 n)$ étapes, où n est le nombre de récepteurs. Ils montrent également que la borne supérieure de $O(\log^2 n)$ étapes est serrée en exhibant une famille de graphes infinie requérant $\log^2 n$ étapes pour compléter la diffusion à distance 2.

À notre connaissance, le problème de diffusion à distance 2 n'a pas été étudié dans le cas où le graphe sous-jacent n'est pas quelconque. Dessmark et Pelc [30] traitent du problème de diffusion (général) lorsque le graphe sous-jacent est un graphe de disques. Ils proposent un algorithme centralisé de diffusion en $O(D)$ étapes où D est le diamètre du graphe. On voit ici que l'hypothèse de ne pas utiliser les noeuds à distance 2 d'une source pour en informer d'autre est forte : sans cette hypothèse, la diffusion à distance 2 peut être complétée en $O(1)$ étapes, or on montre que $\log n$ étapes peuvent être nécessaires dans notre cas. Notons également que la notation « grand O » dans le schéma de diffusion en $O(D)$ étapes calculé par l'algorithme de Dessmark et Pelc [30] cache en fait une constante multiplicative d'environ 400. En pratique, pour toute instance que l'on peut être amené à manipuler,

notre solution en $4 + 4 \log n$ étapes sera meilleure.

D'un point de vue théorique, résoudre le problème de diffusion à distance 2 sous les hypothèses précédemment exposées est en fort lien avec le problème de la couverture exacte.

Couverture Exacte

Donnée : X un ensemble fini, $C \subset 2^X$ des sous-ensembles de X .

Question : Existe-t-il $S \subset C$ tel que $\forall x \in X, \exists ! c \in S : x \in c$?

En effet, un graphe biparti de diffusion (X, Y) peut être vu comme une instance de **Couverture Exacte** et une diffusion en une étape correspond à une couverture exacte de Y par X . Des variantes géométriques de **Couverture Exacte** ont été étudiées, par exemple la couverture par disques unitaires montrée NP-difficile dans une colonne de Johnson [48] cherche à déterminer l'existence d'une couverture exacte de points du plan par des disques unitaires. Étant donné un biparti de diffusion (X, Y) induit par des disques unitaires, savoir s'il existe un schéma de diffusion en une étape est équivalent à l'existence d'une couverture exacte des récepteurs (points) par les émetteurs (disques unitaires). Un résultat intéressant est celui établi par Chan et Grant [18], qui donne un algorithme polynomial pour résoudre le problème de couverture de points du plan par des pseudo-disques pondérés. Leur résultat peut aisément être adapté pour résoudre en temps polynomial le problème de décider l'existence d'un schéma de diffusion en une étape dans le cas des disques unitaires.

Coloration sans conflit

Le problème de diffusion à distance 2 sous les hypothèses de communication mentionnées précédemment est en fait très proche du problème de la coloration sans conflit : étant donné un ensemble fini U et un ensemble C de sous-ensembles de U , affecter une couleur à chaque sous-ensemble de sorte que pour tout sommet u de U , il existe une couleur qui apparaît exactement une fois dans l'ensemble des sous-ensembles contenant u . On dit qu'une telle coloration est une coloration de C sans conflit avec U .

Coloration Sans Conflit

Donnée : U un ensemble fini, $C \subset 2^U$ des sous-ensembles de U , $k \in \mathbb{N}$

Question : Existe-t-il une k -coloration, χ , de C telle que :

$$\forall u \in U, \exists i \in \{1, \dots, k\} : |\{c \in C : u \in c \text{ et } \chi(c) = i\}| = 1?$$

Ce problème a été introduit par Even et collab. [34] pour résoudre des problèmes d'affectation de fréquences. Dans notre cas, un graphe biparti de diffusion (X, Y) peut être vu comme une instance de **Coloration Sans Conflit**, avec $U = Y$ et $C = \{N(x) : x \in X\}$. Dans ces conditions, une k -coloration de C sans conflit avec U correspond à un schéma de diffusion en k étapes dans lequel les émetteurs émettent tous exactement une fois.

Even et collab. [34] montrent que $\log |C|$ couleurs suffisent à colorer C sans conflit avec n'importe quel ensemble X de points du plan inclut dans l'union de C lorsque C est :

- un ensemble de disques,
- un ensemble de rectangles parallèles aux axes dont le ratio entre le plus grand et le plus petit rectangle est borné par une constante,
- un ensemble de régions convexes obtenu par translation/homothétie d'une région convexe symétrique centralement.

Har-Peled et Smorodinsky [47] améliorent encore ces résultats en proposant $\log |C|$ couleurs lorsque C est un ensemble de pseudo-disques (régions délimitées par des courbes fermées telles que toute paire de courbes s'intersecte en au plus deux points), et $\log^2 |C|$ lorsque C est un ensemble quelconque de rectangles parallèles aux axes.

Dans une autre optique, Katz et collab. [51] s'intéressent en quelque sorte au dual de **Coloration Sans Conflit**, en cherchant à colorer des points sans conflit avec des intervalles. Dans ce cas, X est un ensemble d'intervalles, et C est induit par un ensemble de points (à chaque point on associe le sous-ensemble d'intervalles le contenant). Cheilaris et Smorodinsky [19] montrent ensuite qu'il est quasi-polynomial de décider si un ensemble de point peut être k -coloré sans conflit avec des intervalles. Enfin, Dhannya et Narayanaswamy [31] proposent un algorithme polynomial pour résoudre ce problème et closent les conjectures émises auparavant [19, 51].

Pour d'autres variantes et résultats autour du problème de coloration sans conflit, Smorodinsky, parmi les plus actifs contributeurs à ce problème, a publié en 2013 [64] une excellente vue d'ensemble.

Bien que très proche, le problème de diffusion à distance 2 semble tout de même moins contraint que le problème de coloration sans conflit dans la mesure où les émetteurs pourraient être sollicités plusieurs fois (donc recevoir plusieurs couleurs) ou ne pas émettre du tout. En ce sens, une variante intéressante de la coloration sans conflit est celle introduite par Bärtschi et Grandoni [4] qui cherche une multi-coloration de l'ensemble de sommets d'un hypergraphe sans conflit avec ses hyperarêtes (chaque hyperarête doit contenir un sommet qui est le seul à recevoir une couleur). Ils prouvent un algorithme polynomial produisant des $\log^2 m$ -multi-colorations sans conflit d'hypergraphes, où m est le nombre d'hyperarêtes, et chaque sommet reçoit au plus $\log m$ couleurs. Là encore, un graphe biparti de diffusion peut être vu comme un hypergraphe, et les résultats de Alon et collab. [2] et Cogis et collab. [26] qui proposent des schémas de diffusions en $\log^2 m$ étapes sont très proches des $\log^2 m$ -multi-colorations trouvées par l'algorithme de Bärtschi et Grandoni [4].

3.1.3 Organisation du chapitre

On montre en section 3.2 une série de résultats répondant aux problématiques de borne supérieure sur le nombre d'étapes, minimisation du nombre d'étapes, borne sur la proportion d'informés et maximisation du nombre d'informés, dans le cas des intervalles.

Les résultats de la section 3.3 sont ensuite adaptés au cas des intervalles circulaires. Dans ce cas, on montre également que limiter le nombre d'émissions des sommets dégrade les résultats qu'il est possible d'obtenir.

Enfin, le cas des disques unitaires est traité en section 3.4. On montre une borne logarithmique serrée sur le nombre d'étapes et un résultat négatif sur la proportion qu'il est possible d'informer en une étape.

3.2 Cas des intervalles

Considérons que les voisinages des émetteurs sont des intervalles de l'ensemble des récepteurs : sur la figure 3.1 les y_i représentent les sommets à

couvrir, et les émetteurs couvrent des intervalles de ces y_i .

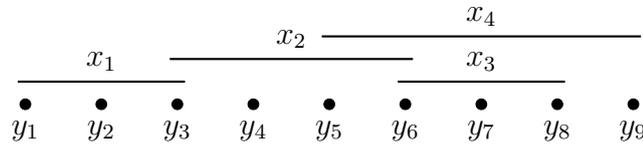


Figure 3.1 – Biparti de diffusion dans le cas des intervalles. y_1, \dots, y_9 sont des récepteurs. Les intervalles x_1, x_2, x_3, x_4 représentent les voisinages de quatre émetteurs.

On commence par décrire l'algorithme 2, polynomial, qui calcule deux ensembles disjoints d'émetteurs.

Algorithme 2 : Diffusion Intervalles

Données : $G = (X, Y, E)$ un biparti de diffusion tel que $Y = \{y_1, \dots, y_m\}$ et les voisinages de X forment des intervalles de Y .

Résultat : $X_1, X_2 \subset X$ une diffusion en deux étapes telle que $X_1 \cap X_2 = \emptyset$.

début

Trier X par ordre de fin d'intervalle croissant;

$X_1, X_2 \leftarrow \emptyset, \emptyset$;

$i \leftarrow 0$;

tant que Y n'est pas entièrement marqué **faire**

Noter y le premier sommet de Y à ne pas être marqué;

Noter x le plus grand sommet de X tel que $y \in N(x)$;

Ajouter x à $X_{(i \bmod 2)+1}$;

Marquer tous les sommets de $N(x)$;

$i \leftarrow i + 1$;

fin

retourner X_1, X_2

fin

Le théorème 3.1 montre que l'algorithme 2 produit en fait des schémas de diffusion en deux étapes. Il répond donc à la première problématique qui est de déterminer une borne supérieure sur le nombre d'étapes (ici deux).

Théorème 3.1

Deux étapes suffisent à compléter la diffusion dans le cas des intervalles.

Preuve

Montrons que l'algorithme 2 retourne un schéma de diffusion en deux étapes.

Notons $S = \{s_1, \dots, s_k\}$ les intervalles correspondant aux sommets de X sélectionnés durant la procédure. Pour tout i compris entre 1 et $k-3$, on a $s_i \cap s_{i+2} = \emptyset$. En effet, supposons que ce ne soit pas le cas pour un i donné : $s_i \cap s_{i+2} \neq \emptyset$. s_{i+1} termine strictement après s_i puisqu'il a été sélectionné après s_i pour couvrir un sommet non-marqué. s_i commence strictement avant s_{i+1} , sinon s_{i+1} aurait été sélectionné à sa place. Par des arguments similaires, on peut établir que s_{i+2} termine strictement après s_{i+1} et s_{i+1} commence strictement avant s_{i+2} . Dans ces conditions, on a $s_{i+1} \subset s_i \cup s_{i+2}$ ce qui est absurde puisque s_{i+1} a été sélectionné pour couvrir un sommet. On a donc $s_i \cap s_{i+2} = \emptyset$.

Il suffit maintenant d'observer que tous les sommets de X_1 (resp. X_2) ont même parité, et leurs voisinages ne s'intersectent donc pas deux à deux (puisque $s_i \cap s_{i+2} = \emptyset$). Faire émettre tous les sommets de X_1 à une étape ne crée pas d'interférence, de même pour X_2 . Si l'algorithme termine, la totalité de Y est informée. X_1, X_2 est donc bien un schéma de diffusion en deux étapes.

L'algorithme 2 fournit une solution qui a de bonnes propriétés : les deux ensembles d'émetteurs sont disjoints. On peut exploiter cette propriété pour répondre à une deuxième de nos problématiques, à savoir la proportion de récepteurs qu'il est possible d'informer en une étape.

Corollaire 3.1.1

Il est toujours possible d'informer $\frac{2}{3}$ des récepteurs en une étape dans le cas des intervalles.

Preuve

Soit X_1, X_2 le schéma de diffusion en deux étapes retourné par notre procédure. Notons t_1 , (resp. t_2) le nombre de récepteurs informés par X_1 uniquement (resp. X_2 uniquement). On note également t_{12} le nombre de récepteurs touchés à la fois par X_1 et par X_2 . Notons h_1 (resp. h_2)

le nombre de récepteurs informés lorsque X_1 (resp. X_2) émet, et h_{12} le nombre de récepteurs informés lorsque X_1 et X_2 émettent simultanément. En notant N le nombre total de récepteurs, les équations suivantes sont vérifiées :

$$h_1 = t_1 + t_{12} \quad (3.1)$$

$$h_2 = t_2 + t_{12} \quad (3.2)$$

$$h_{12} = t_1 + t_2 \quad (3.3)$$

$$N = t_1 + t_2 + t_{12} \quad (3.4)$$

En combinant les équations on obtient :

$$\begin{aligned} h_1 + h_2 + h_{12} &= 2t_1 + 2t_2 + 2t_{12} \\ &= 2N \end{aligned}$$

On a enfin $\max(h_1, h_2, h_{12}) \geq \frac{h_1+h_2+h_{12}}{3} = \frac{2}{3}N$, ainsi on atteint $\frac{2}{3}$ des récepteurs en faisant émettre X_1 ou X_2 ou les deux à la fois, d'où le corollaire.

La proportion de récepteurs informés en une étape donnée par le corollaire 3.1.1 est la meilleure possible. Il suffit de considérer l'instance illustrée en figure 3.2. Cette instance est également un exemple « serré » pour la borne prouvée en théorème 3.1.

L'algorithme 2 retournant en temps polynomial des schémas de diffusion en deux étapes, on sait qu'il est une deux approximation du problème de minimisation du nombre d'étapes dans le cas des intervalles, puisqu'au mieux un schéma comporte 1 étape. En fait, on peut décider en temps polynomial si une étape suffit. Pour cela, considérons que les intervalles sont ordonnés

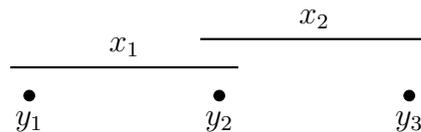


Figure 3.2 – Instance atteignant le ratio de $\frac{2}{3}$ récepteurs informés en une étape.

totallement par ordre de début croissant, puis ordre de fin croissant en cas d'égalité. Pour deux intervalles I et J tels que $I < J$, si J démarre immédiatement après I (comprendre I termine à y_i et J démarre à y_{i+1}) on dit qu'ils sont consécutifs et on note $I \rightarrow J$. On peut alors réduire la décision d'une diffusion à distance 2 en une étape au calcul d'un chemin dans un graphe via la construction suivante.

Construction 3.1

Soit $G = (X, Y, E)$ un graphe biparti de diffusion dans le cas des intervalles. $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$ et les voisinages des sommets de X forment des intervalles de Y . Pour tout $i \in \{1, \dots, n\}$, on note $I_i = N(x_i)$. On construit un graphe $D = (V, A)$ de la manière suivante (voir figure 3.3).

- À chaque intervalle correspond un sommet, et on ajoute une source s et un puits t : $V = \{v_1, \dots, v_n, s, t\}$.
- On relie s aux sommets correspondant à des intervalles contenant y_1 , t à ceux contenant y_m et toute paire d'intervalles se suivant directement : $A = \{v_i v_j : I_i \rightarrow I_j\} \cup \{s v_i \mid y_1 \in I_i\} \cup \{t v_i \mid y_m \in I_i\}$

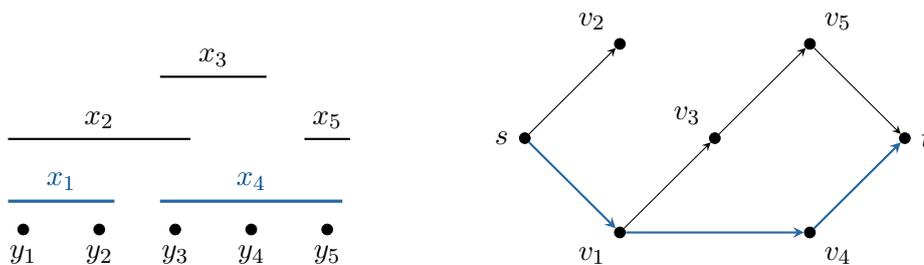


Figure 3.3 – Exemple d'application de la construction 3.1. À gauche une instance du problème de diffusion dans le cas des intervalles. À droite le graphe résultant. Les deux intervalles bleus forment une couverture exacte de Y et correspondent à un chemin de s à t de longueur 3 dans le graphe.

On montre alors le théorème suivant en réduisant le problème de diffusion à distance 2 en une étape au calcul d'un plus court chemin.

Théorème 3.2

Décider s'il est possible de compléter une diffusion à distance 2 en une étape est polynomial dans le cas des intervalles, et on peut minimiser le nombre d'émetteurs pour réaliser la diffusion en une étape si cela est possible.

Preuve

Soit $G = (X, Y, E)$ un graphe biparti de diffusion dans le cas des intervalles. Soit $D = (V, A)$ le graphe obtenu à partir de G par la construction 3.1 (voir figure 3.3). On remarque qu'il existe un chemin de s à t de longueur $k + 1$ dans D si et seulement s'il existe une couverture exacte de Y par k sommets de X . Trouver un chemin de s à t de longueur minimale est polynomial, d'où le théorème.

Le théorème 3.2 montre également que le problème de minimisation du nombre d'étape est polynomial dans le cas des intervalles. Il suffit pour cela de décider en temps polynomial si une étape suffit. Le cas échéant, le minimum est 1, sinon le minimum est 2 et l'algorithme 2 produit un schéma de diffusion en deux étapes.

Enfin, on peut améliorer le résultat du corollaire 3.1.1 et montrer que le problème de maximiser le nombre de récepteurs informés en une étape est polynomial. Pour cela, on utilise la construction suivante qui transforme une instance dans le cas des intervalles en graphe orienté acyclique.

Construction 3.2

Soit $G = (X, Y, E)$ un graphe biparti de diffusion dans le cas des intervalles. $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_m\}$ et les voisinages des sommets de X forment des intervalles de Y . Pour tout $i \in \{1, \dots, n\}$, on note $I_i = N(x_i) = \{y_{i_1}, \dots, y_{i_k}\}$. On construit un graphe $D = (V, A)$ de la manière suivante, pour un $\varepsilon < 0$ (voir figure 3.4).

- Pour tout $x_i \in X$, ajouter $\{v_{i_1}^i, \dots, v_{i_k}^i, v_f^i\}$, $k + 1$ sommets à D ; placer un arc $v_{i_j}^i v_{i_{j+1}}^i$ de poids 1 pour tout $j \in \{1, \dots, k\}$.
- Pour tout $I_i, I_j \in \mathcal{I}$ tels que $I_i < I_j$,
 - si $I_i \cap I_j = \emptyset$, placer un arc de poids ε entre v_f^i et $v_{j_1}^j$;

- sinon, $I_i \cap I_j = \{y_{l_1}, \dots, y_{l_p}\}$, placer un arc de poids ε entre $v_{l_1}^i$ et $v_{l_{p+1}}^j$

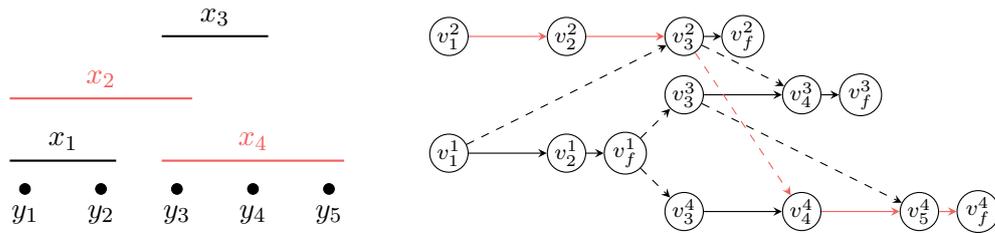


Figure 3.4 – Exemple d'application de la construction 3.2. À gauche une instance du problème de diffusion dans le cas des intervalles. À droite le graphe résultant; les arcs pleins sont de poids 1 et les arcs pointillés de poids ε (négatif). Si x_2 et x_4 émettent ensemble, 4 sommets sont couverts (interférences créées en y_3 .); cela se traduit par le chemin rouge de longueur $4 + \varepsilon$ dans le graphe à droite.

On utilise cette construction pour montrer le théorème suivant, en exploitant la polynomialité du problème du plus long chemin dans les graphes orientés acycliques.

Théorème 3.3

Maximiser le nombre de récepteurs informés en une étape est polynomial dans le cas des intervalles, et on peut minimiser le nombre d'émetteurs utilisés.

Preuve

Soit $G = (X, Y, E)$ un graphe biparti de diffusion dans le cas des intervalles. Soit $D = (V, A)$ le graphe obtenu à partir de G par la construction 3.2 (voir figure 3.4). Il existe un chemin de poids $k + p\varepsilon$ dans D si et seulement s'il existe une solution couvrant k sommets avec $p + 1$ intervalles dans l'instance initiale. Trouver un plus long chemin dans un graphe orienté acyclique étant polynomial, on a donc le théorème.

3.3 Cas des intervalles circulaires

Dans cette section, on suppose que les voisinages des émetteurs forment des intervalles circulaires de l'ensemble des récepteurs (figure 3.5).

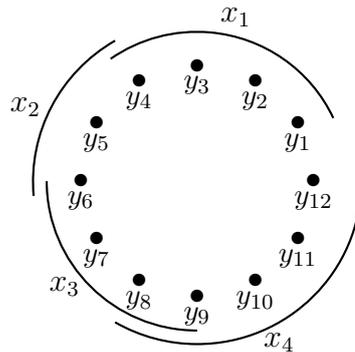


Figure 3.5 – Biparti de diffusion dans le cas des intervalles circulaires. Comme dans la figure 3.1, les y_i sont des sommets à informer et les arcs représentent les voisinages des sommets x_j .

Remarquons d'abord que deux étapes ne suffisent plus si l'on restreint les sommets à n'émettre qu'une fois. Dans la figure 3.6 tous les x_i doivent émettre (ils ont chacun dans leur portée un sommet qu'aucun autre n'atteint) et si deux d'entre eux émettent à une même étape alors un y_j est interféré.

En revanche, trois étapes suffisent et le théorème suivant repose sur une légère modification de l'algorithme 2.

Théorème 3.4

Trois étapes suffisent à compléter une diffusion à distance 2 dans le cas des intervalles circulaires.

Preuve

On utilise l'algorithme 2. On peut réutiliser la preuve du théorème 3.1 sauf que dans le cas d'intervalles circulaires s_k et s_1 peuvent s'intersecter. On peut donc modifier l'algorithme pour que le dernier sommet sélectionné soit placé dans un nouvel ensemble X_3 . Ce faisant, on garantit que les intervalles de X_1 (et de X_2) sont deux à deux disjoints et X_1, X_2, X_3 est bien un schéma de diffusion en trois étapes.

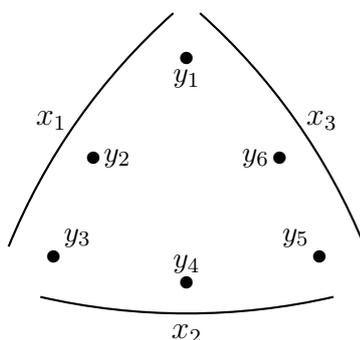


Figure 3.6 – Limiter le nombre d'émissions change l'optimal. Si chaque sommet x_i est restreint à n'émettre qu'une fois, alors trois étapes sont nécessaires pour compléter la diffusion. Si cette restriction est levée, deux étapes suffisent et $(x_1, x_2), (x_1, x_3)$ est un schéma de diffusion en deux étapes.

Le théorème 3.4 montre que trois étapes suffisent pour compléter la diffusion dans le cas des intervalles circulaires. Dans le cas où les sommets ne sont autorisés à n'émettre qu'une fois, cette borne est serrée. Si cette contrainte sur le nombre d'émissions est relâchée deux étapes suffisent comme le montre le corollaire suivant.

Corollaire 3.4.1

Deux étapes suffisent à compléter une diffusion à distance 2 dans le cas des intervalles circulaires si les sommets peuvent émettre deux fois.

Preuve

Soit X_1, X_2, X_3 le schéma de diffusion en trois étapes retourné par l'algorithme 2 après la modification apportée en preuve du théorème 3.4. On remarque que X_3 est constitué de l'unique intervalle s_k . Dans le pire des cas s_k intersecte s_1 et s_{k-1} . En effet si s_k intersecte aussi s_2 alors s_1 est contenu dans l'union de s_k et s_2 , on peut alors retirer s_1 de la solution et décaler le reste des indices pour se ramener au cas où s_k n'intersecte que s_1 et s_{k-1} . Dans ces conditions, si $s_{k-1} \in X_1$ alors s_k n'intersecte aucun intervalle de X_2 et on peut l'y ajouter pour obtenir un schéma de diffusion en deux étapes. Sinon, $s_{k-1} \in X_2$, on peut alors poser $X'_1 = X_1 \cup \{s_k\}$, $X'_2 = X_2 \cup \{s_k\}$ et X'_1, X'_2 est un schéma de

diffusion en deux étapes.

La procédure adaptée en preuve du théorème 3.4 permet à nouveau de montrer qu'une bonne proportion de récepteurs peut être informée en une étape en utilisant un raisonnement similaire à celui du corollaire 3.1.1.

Corollaire 3.4.2

Il est toujours possible de couvrir $\frac{2}{3}$ des récepteurs en une étape dans le cas des intervalles circulaires.

Preuve

Soit X_1, X_2, X_3 le schéma de diffusion en trois étapes retourné par l'algorithme 2 après la modification apportée en preuve du théorème 3.4. Notons t_1 , (resp. t_2 , et t_3) le nombre de sommets de Y appartenant au voisinage de X_1 uniquement (resp. X_2 uniquement, et X_3 uniquement). On note également t_{ij} le nombre de sommets appartenant à la fois au voisinage de X_i et au voisinage de X_j . Puisque aucun sommet n'appartient à trois intervalles de la solution, aucun n'est couvert à la fois par X_1, X_2 et X_3 . Notons h_{ij} le nombre de récepteurs informés lorsque X_i et X_j émettent simultanément. En notant N le nombre total de récepteurs, les équations suivantes sont vérifiées :

$$h_{12} = t_1 + t_2 + t_{13} + t_{23} \quad (3.5)$$

$$h_{13} = t_1 + t_3 + t_{12} + t_{23} \quad (3.6)$$

$$h_{23} = t_2 + t_3 + t_{12} + t_{13} \quad (3.7)$$

$$N = t_1 + t_2 + t_3 + t_{12} + t_{13} + t_{23} \quad (3.8)$$

En combinant les équations on obtient :

$$\begin{aligned} h_{12} + h_{13} + h_{23} &= 2t_1 + 2t_2 + 2t_3 + 2t_{12} + 2t_{13} + 2t_{23} \\ &= 2N \end{aligned}$$

On a enfin $\max(h_{12}, h_{13}, h_{23}) \geq \frac{h_{12}+h_{13}+h_{23}}{3} = \frac{2}{3}N$, d'où le corollaire.

La proportion de couverture en une étape donnée en corollaire 3.4.2 est forcément la meilleure possible puisque l'ensemble des instances du cas non-circulaire est contenu dans l'ensemble des instances du cas circulaire.

Le théorème suivant exploite le théorème 3.2 pour décider en temps polynomial l'existence d'un schéma de diffusion en une étape dans le cas des intervalles circulaires.

Théorème 3.5

Décider si une étape suffit à compléter la diffusion est polynomial dans le cas des intervalles circulaires.

Preuve

Il suffit de se ramener au cas non-circulaire n fois en décidant, pour chaque intervalle qu'il sera émetteur et à considérer le graphe d'intervalles non-circulaires obtenu en retirant tous les intervalles intersectant l'intervalle choisi.

Dans le cas où les sommets sont restreints à n'émettre qu'une fois, trois étapes suffisent à compléter la diffusion. On a vu que ces trois étapes étaient parfois nécessaires, et qu'il était polynomial de décider la faisabilité de la diffusion en une étape, aussi se pose-t-on la question de la complexité du problème de décider la faisabilité de la diffusion en deux étapes dans le cas des intervalles circulaires (avec émission unique). La construction suivante réduit le problème de la diffusion en deux étapes à la recherche de cycles dans un graphe orienté. Par la suite, on suppose que les récepteurs sont ordonnés cycliquement et que pour deux intervalles circulaires I_i et I_j , I_i est plus petit que I_j ($I_i < I_j$) si I_i débute avant I_j dans l'ordre cyclique.

Construction 3.3

Soit $G = (X, Y, E)$ un graphe biparti de diffusion. On note $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ et $\mathcal{I} = (I_1, \dots, I_n)$ les intervalles corres-

pondant aux voisinages de X . On construit un graphe $D = (V, A)$ tel que :

- $V = \{v_1, \dots, v_n\}$
- $A = \{v_i v_j : I_i < I_j, I_i \cap I_j \neq \emptyset\} \cup \{v_i v_j : I_i \rightarrow I_j\}$

On colore ensuite les arcs correspondants à des chevauchements en rouge, et les arcs correspondants à des intervalles consécutifs en bleu.

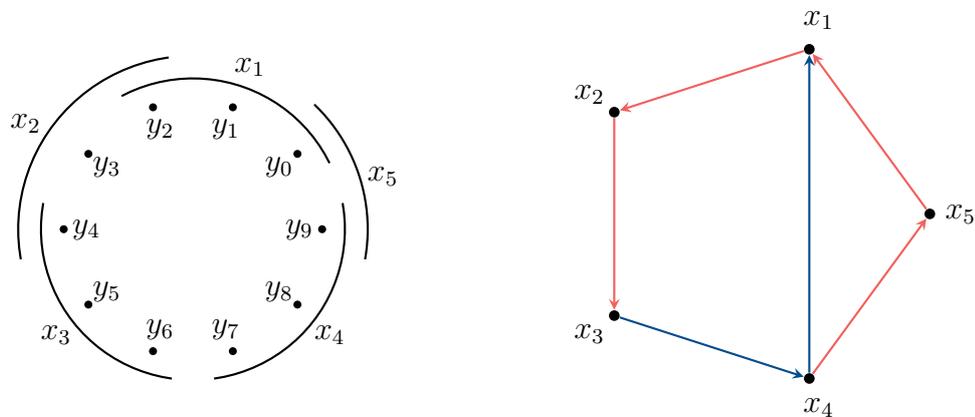


Figure 3.7 – Exemple d'application de la construction 3.3. À gauche une instance du problème de diffusion dans le cas des intervalles circulaires. À droite le graphe construit. Les arcs bleus correspondent à des intervalles qui se suivent, et les arcs rouges à des intervalles se chevauchant.

Théorème 3.6

Décider si deux étapes suffisent à compléter la diffusion est polynomial dans le cas des intervalles circulaires.

Preuve

Soit $G = (X, Y, E)$ un biparti de diffusion. Soit $D = (V, A)$ le graphe obtenu après construction 3.3. On remarque qu'un cycle orienté dans le graphe obtenu correspond à un ensemble d'émetteurs dont l'union des voisinages contient tous les récepteurs. On peut à présent montrer qu'il existe un schéma de diffusion en deux étapes dans G si et seulement

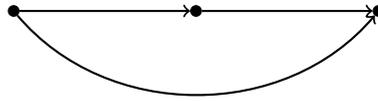
si :

- Il existe un cycle induit pair d'arcs rouges dans D
- ou il existe un cycle induit contenant au moins un arc bleu.

\Leftarrow S'il existe un cycle induit pair d'arcs rouges dans D , le schéma consistant à faire émettre l'ensemble des sommets en position paire (pour un point de départ quelconque) dans le cycle à une étape et l'ensemble des sommets en position impaire à une autre étape est un schéma de diffusion en deux étapes. S'il existe un cycle induit contenant un arc bleu, il suffit de prendre la destination de l'arc bleu comme point de départ pour ordonner les sommets du cycle et le schéma proposé précédemment fonctionne toujours.

\Rightarrow S'il existe un schéma de diffusion en deux étapes dans G , alors l'ensemble des émetteurs utilisés dans le schéma correspond à un cycle dans D . Supposons que ce cycle ne soit constitué que d'arcs rouges, en nombre impair. Si le cycle est induit, alors, les arcs rouges correspondant à des chevauchements, les émetteurs correspondant à deux sommets consécutifs dans le cycle ne peuvent émettre à la même étape; le cycle étant impair il faudrait au moins trois étapes pour compléter la diffusion en utilisant les sommets du cycle. Le cycle n'est donc pas induit, il existe un arc interceptant deux sommets du cycle. Si un tel arc est bleu, alors il existe un cycle contenant un arc bleu dans D . Sinon, tous les arcs interceptant deux sommets du cycle sont rouges. Par construction, l'arc est de la forme $v_i v_j$ avec $i \leq j - 2$, et comme x_i intersecte x_j , il intersecte également tous les intervalles x_k avec $i < k \leq j$. Il existe donc un arc de la forme $x_i x_{i+2}$ qui permet "rendre pair" le cycle rouge impair initial.

Pour tester ces conditions en temps polynomial, on peut d'abord tester l'existence d'un cycle dans D . Si un cycle, C , existe, alors si C contient au moins un arc bleu, la réponse est positive. Sinon, C est un cycle rouge et s'il est pair la réponse est positive. Sinon, C est un cycle rouge impair, il suffit de rechercher la structure suivante dans le graphe :



Si une telle structure existe, alors il existe un cycle pair. Sinon, tous les cycles ont même parité et sont donc impairs la réponse est alors négative.

3.4 Cas des disques unitaires

Supposons maintenant qu'à chaque émetteur corresponde un disque du plan représentant sa portée de communication.

Dans les cas où tous les disques ont le même rayon (on dit qu'ils sont unitaires), le problème de diffusion à distance 2 peut être simplifié. On peut supposer que le sommet « racine » détenant l'information à diffuser est l'origine d'un repère, son disque de communication est alors le disque unité. L'ensemble des sommets voisins de la racine sont des points à l'intérieur du disque unité, notons $\mathcal{D} = \{D_1, \dots, D_n\}$ l'ensemble de disques correspondants. Tout disque de \mathcal{D} contient l'origine. On utilise par la suite les notions de zone de réception et de signature définies comme il suit et illustrées en figure 3.9.

Définition 3.1 : Zone de réception

Soit \mathcal{D} un ensemble ordonné de disques. On appelle **zone de réception**, induite par \mathcal{D} , toute partie du plan, hors du disque unité, contenue dans au moins un disque de \mathcal{D} .

Définition 3.2 : Signature

Soit \mathcal{D} un ensemble ordonné de disques. Soit Z une zone de réception induite par \mathcal{D} . La **signature** de Z correspond aux indices des disques qui la contiennent. On la note $\sigma(Z)$.

On va montrer qu'il existe une famille d'instances nécessitant $1 + \log n$

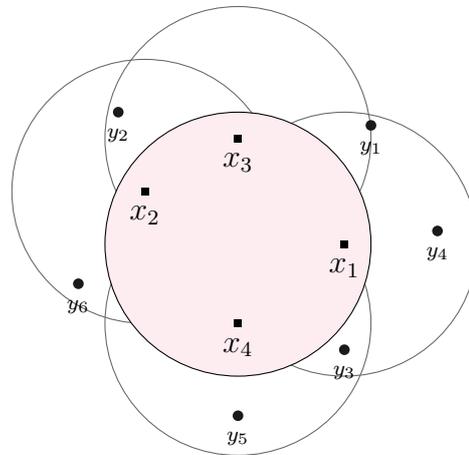


Figure 3.8 – Instance dans le cas des disques unitaires. En rose, le disque unité représente la portée de la source. Les carrés sont les sommets voisins de la source (les émetteurs). Les cercles délimitent les disques de communication des émetteurs. Les points à l'extérieur du disque unité sont les récepteurs.

étapes pour compléter la diffusion. On montrera ensuite que toute instance du problème de diffusion à distance 2 dans le cas des disques unitaires peut être réduite à des instances de la famille sus-citées.

Notons que l'existence d'une famille d'instances nécessitant $O(\log n)$ étapes pour compléter la diffusion à distance 2 est assez décourageante : on ne peut pas espérer une meilleure borne que celle démontrée par Even et collab. [34] pour la coloration de disques (pas forcément unitaires) sans conflit avec des points. On rappelle qu'ils montrent que $O(\log n)$ couleurs suffisent pour colorer tout ensemble de disques de sorte que tout point du plan n'appartient soit à aucun disque, soit à au moins un disque qui est le seul à recevoir sa couleur. Dans notre cas, les disques sont unitaires et la coloration correspond à une affectation des étapes d'émission. On montre donc, dans la section suivante, que cette borne logarithmique n'est pas améliorée lorsque les disques sont unitaires et chaque sommet peut recevoir plusieurs couleurs (comprendre « émettre à plusieurs étapes »).

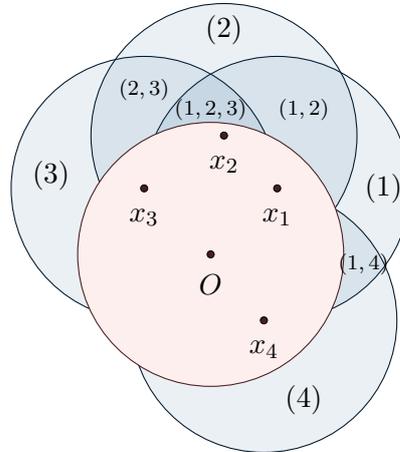


Figure 3.9 – Exemple de zones de réceptions et signatures pour quatre disques de centres x_1, x_2, x_3 et x_4 . Les zones bleues partitionnent l'union des disques hors du disque unité (rouge); ce sont les zones de réception induites par les quatre disques. Les signatures des zones de réception sont inscrites en leur sein.

3.4.1 Certaines instances nécessitent $O(\log n)$ étapes

Considérons un ensemble de n disques distincts, $\mathcal{D} = (D_1, \dots, D_n)$, centrés sur un rayon du disque unité, tels que pour tout i, j si $i < j$ alors le centre de D_i est plus proche de l'origine que le centre de D_j (voir figure 3.10). Toute zone de réception induite par \mathcal{D} a une signature de la forme $(i, i + 1, \dots, i + j)$.

On peut exploiter la forme des signatures des zones de réception en les représentant sous la forme de la matrice suivante, dite **de signature**, qui donne l'ensemble des zones de réception distinctes qui existent dans la configuration considérée.

(1)				
(1, 2)	(2)			
(1, 2, 3)	(2, 3)	(3)		
\vdots	\vdots	\vdots	\ddots	
(1, 2, ..., n)	(2, 3, ..., n)	(3, 4, ..., n)	...	(n)

Supposons maintenant que chaque zone de réception contienne au moins un sommet à couvrir. Notons t_k le nombre d'étapes nécessaires pour compléter la diffusion dans une instance de taille k .

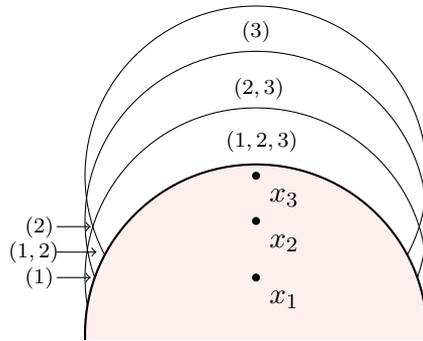


Figure 3.10 – Instance critique. x_1 , x_2 et x_3 sont dans le disque unité rouge. Les signatures des zones de réception sont de la forme $(i, i + 1, \dots, i + j)$ pour des i et j donnés. $1 + \log n = 2$ étapes sont nécessaires.

Les propriétés suivantes sont vérifiées.

1. Il existe une zone de réception S de signature $(1, 2, \dots, n)$, il existe donc, dans toute solution un ensemble émetteur X_i tel que $|X_i| = 1$.
2. Étant donnée une solution X_1, \dots, X_k , l'ordre sur les X_i n'importe pas : Pour tout i, j , échanger X_i et X_j dans la solution ne peut pas la rendre invalide ou plus courte.

On peut donc supposer sans perte de généralité que $|X_1| = 1$ (à la première étape il n'y a qu'un seul émetteur). Notons i l'indice de l'émetteur choisi pour la première étape ($X_1 = \{x_i\}$). Les zones couvertes à cette première étape sont en vert dans la matrice suivante.

(1)					
(1, 2)	(2)				
⋮	⋮	⋮			
(1, ..., i)	(2, ..., i)	...	(i)		
⋮	⋮	⋮	⋮	⋮	
(1, ..., n - 1)	(2, ..., n - 1)	...	(i, ..., n - 1)	...	(n - 1)
(1, ..., n)	(2, ..., n)	...	(i, ..., n)	...	(n - 1, n) (n)

Les zones devant encore être informées sont disjointes et on obtient pour ces zones deux matrices de signature ayant la même forme que la première.

(1)	(i + 1)
(1, 2) (2)	(i + 1, i + 2) (i + 2)
⋮ ⋮ ⋮	⋮ ⋮ ⋮
(1, ..., i - 1) (2, ..., i - 1) ... (i - 1)	(i + 1, ..., n) (i + 2, ..., n) ... (n)

On remarque qu'au moins une matrice restante est de dimension au moins $\lceil \frac{n-1}{2} \rceil$. Cette matrice définit un problème diffusion de taille au moins la moitié de la taille du problème initial. L'équation de récurrence suivante pour t_n est donc vérifiée.

$$t_n \geq t_{\lceil \frac{n-1}{2} \rceil} + 1 \quad (3.9)$$

Cette récurrence se résout en :

$$t_n \geq 1 + \lceil \log_2(n) \rceil \quad (3.10)$$

On remarque que cette borne est atteinte, puisqu'il suffit à chaque instance récursivement résolue de choisir le sommet d'indice $\frac{n}{2}$ qui couvrira au moins la moitié des zones pour serrer les équations de récurrence.

3.4.2 $O(\log n)$ étapes suffisent

On va à présent montrer que $O(\log n)$ suffisent. Pour cela, étant donnée une instance, on va la transformer en une instance similaire à celles définies à la section précédente. La transformation consiste à retirer les disques qui sont contenus dans l'union d'autres jusqu'à ce que tous les disques induisent des zones de réception propres.

Définition 3.3 : Zone de réception propre

Soit \mathcal{D} un ensemble de disques. Soit $D \in \mathcal{D}$. D induit une **zone propre** s'il existe $p \in D$ tel que pour tout $D' \neq D \in \mathcal{D}$, $d \notin D'$.

Notons que filtrer une instance pour ne conserver que les disques induisant des zones propres peut être réalisé en temps polynomial. L'idée est, une fois l'instance filtrée, de diviser le plan en quatre secteurs, puis de s'intéresser séquentiellement à la restriction de l'instance à chaque secteur. Au final, chaque restriction d'instance à quart de plan ne contiendra que des zones de

réception à la forme semblable à celles de la section précédente, on pourra donc compléter la diffusion dans chaque secteur en $O(\log n)$ étapes.

Dans ce qui suit D_u désigne le disque unité centré en l'origine O . Les divisions du plan considérées sont incidentes à O . Lorsqu'un disque sera noté D_i , C_i sera le cercle correspondant (le bord de D_i).

On a le résultat suivant qui décrit comment deux cercles peuvent s'intersecter dans un quart de plan.

Lemme 3.7

Soit Z un quart de plan. Soient C et C' deux cercles du plan contenant l'origine. C et C' s'intersectent au plus une fois dans $Z \setminus D_u$.

Preuve

Notons A et B les centres de C_1 et C_2 . Soient P et Q les points d'intersection entre C_1 et C_2 , les bords de D_1 et D_2 . Soit C le point d'intersection entre (OA) et C_1 . La figure 3.11 illustre les notations. On peut supposer sans perte de généralité que A et B sont contenus dans un demi-disque centré en O . On peut également supposer que A est plus proche de O que B ne l'est.

Notons α, β et γ les angles (\vec{OP}, \vec{OQ}) , (\vec{AQ}, \vec{AP}) et (\vec{CP}, \vec{CQ}) . Puisque β et γ interceptent le même arc de C_1 , le théorème de l'angle au centre nous permet d'affirmer $\gamma = \frac{\beta}{2}$. Par construction, comme le triangle OPQ est inclus dans le triangle CPQ , on a $\alpha \geq \gamma$. On a aussi $\beta > \pi$ puisque sinon PQ est un diamètre d'à la fois C_1 et C_2 et dans ce cas $C_1 = C_2$. On a enfin $\alpha > \frac{\pi}{2}$, et donc P et Q ne peuvent être tous les deux dans $Z \setminus D_u$.

Soit \mathcal{D} un ensemble de disques contenant l'origine. Soit Z un secteur d'angle $\frac{\pi}{2}$. Notons $\mathcal{D}|_Z$ l'ensemble des disques de \mathcal{D} qui contiennent au moins un point de Z . Notons \mathcal{D}' l'ensemble des disques de $\mathcal{D}|_Z$ induisant une zone propre. On peut ordonner \mathcal{D}' comme sur la figure 3.12 par ordre d'apparition des zones propres dans le sens trigonométrique, de sorte que si S_i et S_j , sont les zones propres induites par D_i et D_j respectivement, alors $i < j \implies \forall P_i \in S_i, \forall P_j \in S_j, (\widehat{\vec{OP}_i, \vec{OP}_j}) \in]0, 2\pi[$. On note alors $\mathcal{D}' = \{D_1, \dots, D_m\}$.

On considère par la suite que toute restriction d'un ensemble de disques à

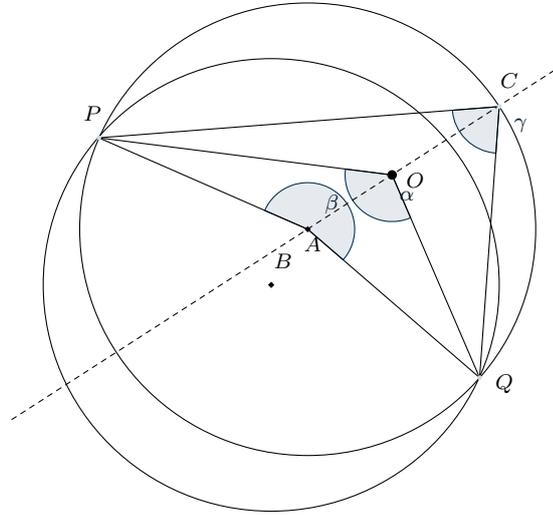


Figure 3.11 – Aide à la preuve du lemme 3.7. Les cercles de centres A et B s'intersectent en P et Q . C est l'intersection entre la droite (OA) et C_1 . L'angle α vaut au moins $\frac{\pi}{2}$.

un quart de plan est filtrée : $\mathcal{D}|_Z$ ne contient que les disques de \mathcal{D} induisant une zone propre dans Z .

Le lemme suivant établit qu'un ensemble de disques restreint à un quart de plan induit des zones de réception dont les signatures sont des intervalles.

Lemme 3.8

Soit Z un quart de plan. Soit \mathcal{D} un ensemble ordonné de disques. Soit S une zone de réception induite par $\mathcal{D}|_Z$. Alors la signature de S est de la forme $\sigma(S) = (i, i + 1, \dots, i + p)$ pour $i, p \in \mathbb{N}$.

Preuve

Considérons trois disques $\{D_1, D_2, D_3\}$ tels qu'il existe une zone de signature $(1, 3)$. Notons P l'unique intersection de C_1 et C_2 dans Z . Notons Z^- (resp. Z^+) la partie de Z située avant (resp. après) la droite (OP) (voir figure 3.13). Notons $D_i^+ := D_i \cap Z^+$ et $D_i^- := D_i \cap Z^-$ pour $i = 1, 2, 3$.

C_1 et C_2 ne s'intersectant qu'une fois, on sait que $D_2^- \subset D_1^-$ et

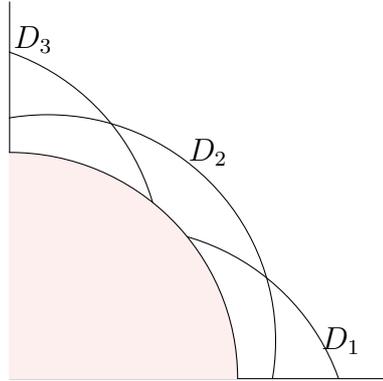


Figure 3.12 – Établissement d'un ordre sur les disques d'un quart de plan en utilisant l'angle à leur zone propre. En rouge le quart de disque unitaire appartenant au quadrant. La zone propre de D_1 apparaît la première (angle le plus petit) puis D_2 et enfin D_3 .

$D_1^+ \subset D_2^+$. La zone de signature $(1, 3)$ ne peut donc pas apparaître dans Z^+ . Elle apparaît donc dans Z^- .

1. Supposons que C_1 et C_3 ne s'intersectent pas dans Z^- , alors $D_3^- \subsetneq D_1^-$ puisque sinon la zone propre induite par D_3 apparaît avant la zone propre induite par D_1 . On remarque que C_2 et C_3 s'intersectent dans Z^+ puisque sinon
 - a) soit $D_2^+ \subsetneq D_3^+$ et alors D_2 n'induit pas de zone propre puisque $D_2^- \subsetneq D_1^-$, ce qui est absurde;
 - b) soit $D_3^+ \subsetneq D_2^+$ et alors la zone propre induite par D_3 apparaît dans Z^- , donc avant celle induite dans D_2 qui apparaît dans Z^+ , ce qui est absurde.

C_2 et C_3 s'intersectent donc dans Z^+ et ne peuvent s'intersecter dans Z^- par lemme 3.7. Alors $D_3^- \subsetneq D_2^-$ puisque sinon $D_2^- \subsetneq D_3^-$ et $P \in D_3^-$, donc $D_3^- \cap D_1^- \neq \emptyset$ or $D_3^- \subsetneq D_1^-$, ce qui est absurde. On a donc $D_3^- \subsetneq D_2^-$, dans ce cas la zone de signature $(1, 3)$ ne peut exister, ce qui est absurde.

2. Supposons que C_1 et C_3 s'intersectent dans Z^- . Alors C_3 intersecte C_1 avant P , sa zone propre apparaît donc avant celle de C_2 , ce qui est absurde.

Dans les deux cas on aboutit à une absurdité, la zone de signature $(1, 3)$ ne peut donc pas apparaître. La signature de S est de la forme $(i, i + 1, \dots, i + p)$, on a montré le lemme.

La conséquence directe de ce lemme est que $O(\log n)$ étapes suffisent pour compléter la diffusion à distance 2 dans le cas des disques unitaires.

Théorème 3.9

$O(\log n)$ étapes suffisent pour compléter la diffusion à distance 2 dans le cas des disques unitaires. De plus un tel schéma est calculable en temps polynomial.

Preuve

Il suffit de partitionner le plan en quatre secteurs d'angle $\frac{\pi}{2}$, Z_1 , Z_2 , Z_3 , et Z_4 . Considérons Z_1 , alors en utilisant le lemme 3.8 toute zone apparaissant dans Z_1 a une signature de la forme $(i, i + 1, \dots, i + j)$. On peut donc construire une matrice de signature ayant une forme similaire à celle considérée en section 3.4.1, et couvrir toutes les zones de réception apparaissant dans Z_1 en $1 + \log n$ étapes. On peut ensuite faire de même pour Z_2 , Z_3 et Z_4 , séquentiellement, et couvrir toutes les zones de réception en $4 + 4 \log n$ étapes. Déterminer les restrictions de l'instance aux quarts de plan, filtrer les disques n'induisant pas de zone propre à chaque quadrant et enfin trouver le schéma de diffusion en $1 + \log n$ étapes, peut être réalisé en temps polynomial, on a donc le théorème.

3.4.3 Maximiser la proportion d'informés en une étape

Enfin, on montre que contrairement au cas des intervalles et intervalles circulaires, il n'est pas possible de garantir la couverture d'une proportion constante des récepteurs quelle que soit l'instance dans le cas des disques unitaires. Pour cela, on construit une famille infinie d'instances dont la proportion de sommets qu'il est possible d'informer en une étape est une fonction décroissante de taille de l'instance.

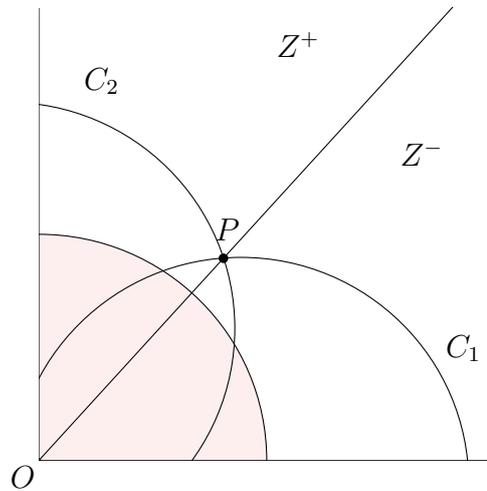


Figure 3.13 – Aide à la preuve du lemme 3.8. Partitionnement du quart de plan Z par la droite OP . P est le point d'intersection entre C_1 et C_2 .

Considérons une instance, I , de la famille construite en section 3.4.1. I est composée de m disques émetteurs de centres alignés sur un rayon du cercle unité. Les zones de réceptions de I ont des signatures qui sont des intervalles et peuvent être représentées sous forme matricielle, comme en section 3.4.1. Considérons maintenant que toutes les zones de réception induites par les disques émetteurs ne contiennent pas de récepteurs. On peut modifier légèrement la représentation matricielle pour traduire cette présence ou non, il suffit pour cela de placer un 1 dans la (i, j) , $1 \leq i < j \leq m$, si la zone $(i, i + 1, \dots, j)$ comporte au moins un récepteur et un 0 sinon (voir figure 3.14).

Considérons une instance I_0 de la famille construite en section 3.4.1 contenant V_0 récepteurs. Notons m le nombre d'émetteurs de I_0 . Supposons qu'optimalement U_0 récepteurs peuvent être informés en une étape. On construit une instance I_1 avec U_0 copies de I_0 . La matrice signature de I_1 est obtenue comme il suit (voir figure 3.15).

- On place U_0 copies de la matrice signature de I_0 bout à bout en diagonale.
- Pour i allant de 1 à U_0 :
 - Pour j allant de $i + 1$ à U_0 :
 - Placer un 1 dans la zone de signature $(1 + (i - 1) \times m, \dots, j \times m)$

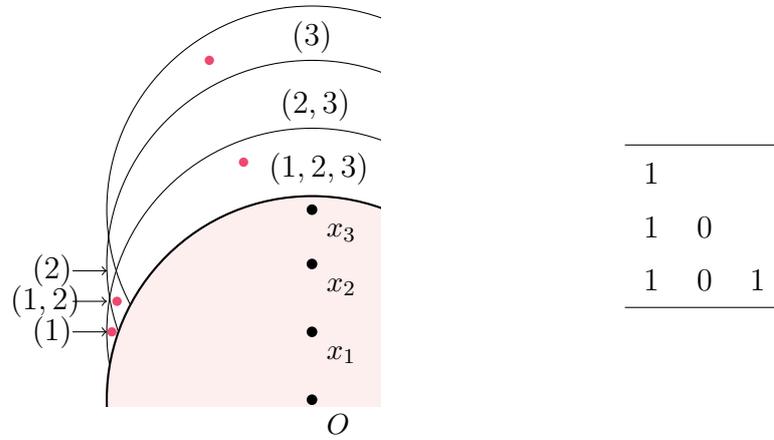


Figure 3.14 – À gauche, x_1 , x_2 et x_3 induisent des zones de réception dont les signatures sont de la forme $(i, i + 1, \dots, i + j)$. Les points roses sont des récepteurs. À droite la matrice de signature correspondante, les cellules $(2, 2)$ et $(2, 3)$ contiennent des 0 car les zones de réceptions correspondantes ne contiennent pas de récepteurs.

Lemme 3.10

Couvrir U_0^2 récepteurs de I_1 en une étape est optimal.

Preuve

On peut couvrir U_0^2 récepteurs de I_1 en une étape en utilisant dans chacune des copies de I_0 le schéma permettant de couvrir U_0 récepteurs.

Soit $k \leq U_0$. Notons I_1^{k-1} l'instance restreinte aux $k - 1$ premières copies de I_0 (on a $I_1 = I_1^{U_0}$). Supposons que le schéma optimal pour I_1^{k-1} consiste à couvrir U_0 récepteurs dans chacune des copies pour couvrir un total de $(k - 1)U_0$ récepteurs. Supposons à présent que l'optimal de I_1^k soit strictement supérieur à kU_0 . Dans ces conditions, un 1 de la dernière ligne doit être couvert par ce schéma optimal, sinon l'optimal de I_1^{k-1} est plus grand que $(k - 1)U_0$ ce qui est absurde.

Considérons le 1 ainsi couvert le plus à gauche sur la dernière ligne. Ce 1 est après la p -ième copie de I_0 ($p \leq k - 2$). Pour couvrir ce récepteur, il ne peut y avoir qu'un seul émetteur entre la p -ième copie et la k -ième; on doit donc « perdre » au moins $(k - p - 1)U_0$ récepteurs. Le gain en

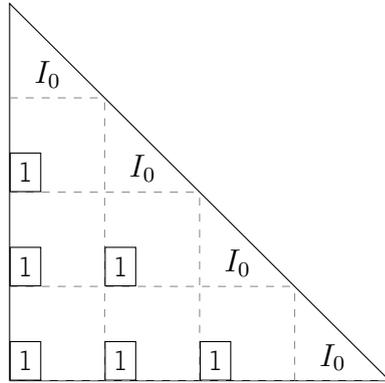


Figure 3.15 – Construction de la matrice de signature de I_1 à partir de celle de I_0 . Ici on place $U_0 = 4$ copies et on complète avec $U_0(U_0-1)/2$ récepteurs.

récepteur couverte est d'au plus $\frac{(k-p)^2}{4}$: le sommet "central" émet seul et couvre la zone verte sur la figure 3.16, la zone rouge est forcément interférée par un sommet de la p -ième copie sinon le 1 couvert ne serait pas le plus à gauche.

Notons $G(p)$ le gain total de 1 couverts en fonction de p .

$$G(p) = \frac{(k-p)^2}{4} - (k-p-1)U_0$$

$$G'(p) = \frac{1}{2}p - \frac{1}{2}k + U_0$$

$$G'(p) = 0 \iff p = k - 2U_0$$

Or, $k \leq U_0$ donc G' s'annule pour une valeur de p négative. G est donc monotone croissante, elle atteint donc son maximum pour $p = k - 2$. $G(k-2) = 1 - U_0 < 0$, le gain en 1 couverts est donc toujours négatif. On a donc l'optimal de I_1^k qui ne peut être supérieur à kU_0 lorsque $k \leq U_0$, il s'en suit que l'optimal de $I_1^{U_0} = I_1$ ne peut être supérieur à U_0^2 , on a donc le lemme.

Notons à présent I_n l'instance obtenue à partir de I_0 en répétant n fois le processus décrit pour passer de I_0 à I_1 . Notons V_n le nombre total de récepteurs dans I_n et U_n le nombre optimal de récepteurs pouvant être informés en une

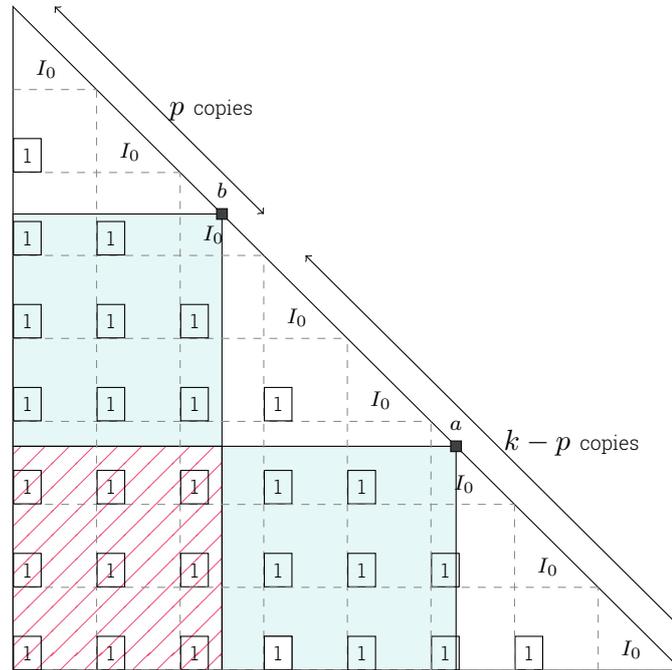


Figure 3.16 – Aide à la preuve du lemme 3.10 : instance I_1^k . Le 1 en blanc sur la dernière ligne est supposé exister, et être le plus à gauche couvert. Ce 1 est à l'aplomb de la $(p+1)$ -ième copie. a est l'unique émetteur dans les $k-p$ dernières copies. b est un émetteur dans la p -ième copie. En vert les zones informées, hachurées de rouge les zones interférées, le reste n'est jamais atteint.

étape. Il découle du lemme 3.10 qu'informer U_{n-1}^2 sommets de I_n en une étape est optimal. Les équations de récurrence sur U_n et V_n suivantes sont alors vérifiées.

$$U_{n+1} = U_n^2$$

$$V_{n+1} = V_n U_n + \frac{U_n(U_n - 1)}{2}$$

La proportion de récepteurs de l'instance I_n pouvant être couverte en une étape est donnée par le rapport $\frac{U_n}{V_n}$. Le lemme suivant montre que ce rapport a une limite nulle.

Lemme 3.11

$$\lim_{n \rightarrow +\infty} \frac{U_n}{V_n} = 0$$

Preuve

$$\begin{aligned} \frac{V_{n+1}}{U_{n+1}} &= \frac{V_n}{U_n} + \frac{U_n - 1}{2U_n} \\ &= \frac{V_n}{U_n} + \frac{1}{2} - \frac{1}{2U_n} \end{aligned}$$

On peut supposer, sans perte de généralité, U_n supérieur à 2, on obtient alors :

$$\begin{aligned} \frac{V_{n+1}}{U_{n+1}} &\geq \frac{V_n}{U_n} + \frac{1}{2} - \frac{1}{4} \\ &\geq \frac{V_n}{U_n} + \frac{1}{4} \end{aligned}$$

Ainsi, $\lim_{n \rightarrow +\infty} \frac{V_n}{U_n} = +\infty$, d'où le lemme.

La construction présentée dans cette section donne une famille infinie d'instances dont la proportion pouvant être couverte en une étape tend vers 0. On a donc le théorème suivant établissant l'impossibilité de garantir une « bonne » proportion de sommets informés en une étape dans le cas des disques unitaires.

Théorème 3.12

Il n'existe pas de constante $c \in]0, 1]$ telle que pour toute instance du problème de diffusion dans les graphes de disques unitaires, la proportion de sommets pouvant être couverts en une étape est au moins c .

3.5 Conclusion

On a étudié trois problèmes liés à la diffusion à distance 2 dans trois cas particulier. Dans le cas où le graphe biparti de diffusion est un graphe d'intervalles (resp. intervalles circulaires), on a montré que :

- Deux (resp. trois) étapes suffisent.
- $\frac{2}{3}$ des sommets peuvent être prévenus en une étape.
- Minimiser le nombre d'étapes est polynomial.

De plus dans le cas des intervalles circulaires, on a montré que limiter le nombre de d'émissions possible pour chaque sommet change la valeur de la meilleure borne supérieure sur le nombre d'étapes en passant de trois à deux étapes en autorisant les sommets à émettre deux fois. Dans les deux cas, intervalles et intervalles circulaires, l'étude s'est montrée exhaustive puisque toutes les bornes proposées sont accompagnées d'exemples sur lesquelles elles sont atteintes.

Dans le cas où le graphe biparti de diffusion est induit par des disques unitaires, on a montré que $O(\log n)$ étapes, où n est le nombre d'émetteurs, suffisent à compléter la diffusion ainsi qu'une famille d'instances nécessitant $1 + \log n$ étapes. Malgré cette borne logarithmique sur le nombre d'étapes, on a pu établir qu'il n'était pas toujours possible de couvrir une proportion constante des sommets en une étape. Résultat quelque peu surprenant puisque si on pouvait couvrir une proportion constante des sommets en une étape, alors on pourrait retrouver un schéma de diffusion en un nombre logarithmique d'étapes.

La complexité du problème de minimiser le nombre d'étapes dans le cas des disques reste encore inconnue. On sait néanmoins que si l'on considère les instances, comme en sous-section 3.4.1, dont les zones ont des signatures en intervalles, alors le problème est polynomial en se ramenant à la coloration de points sans conflit avec des intervalles, en utilisant l'algorithme de Dhannya et Narayanaswamy [31]. Ce résultat est encourageant, bien que la réduction utilisée en sous-section 3.4.2 pour se ramener au cas des signatures en intervalles peut largement dégrader la valeur de la solution optimale. On sait également que maximiser le nombre de récepteurs informés en une étape (et de ce fait savoir si une étape suffit pour compléter la diffusion à distance 2) est polynomial en utilisant les résultats de Chan et Grant [18] sur le problème de la couverture d'ensemble pondérée dans un contexte géométrique.

Dans le cadre de travaux futurs, des résultats similaires sur la minimisation du nombre d'étapes, de la proportion de sommets qu'il est possible d'informer en une étape, ou encore des bornes supérieures sur le nombre d'étapes pourraient être établis dans d'autres classes de graphe. L'idée derrière une telle étude serait par exemple de trouver la plus grande classe de graphes pour laquelle $\log n$ étapes suffisent. Un panorama détaillé des complexités et bornes dans un grand nombre de cas permettrait d'adapter la conception de réseaux au besoin : s'il faut pouvoir informer un grand nombre de sommets en une étape, utiliser telle classe ; s'il faut pouvoir diffuser rapidement, utiliser celle-ci, **etc.**

CHAPITRE 4

RECHERCHE DE SOUS-GRAPHE CONNEXE ÉQUILIBRÉ

Contenu

4.1. Introduction	61
4.1.1 Notations	65
4.1.2 Organisation du chapitre	65
4.2. Complexité classique	66
4.2.1 Graphes bipartis de diamètre quatre	66
4.2.2 Graphes de diamètre trois	68
4.2.3 Graphes bipartis cubiques	72
4.3. Non-approximabilité à facteur constant	79
4.4. Algorithme polynomial à paramètre fixé	83
4.5. Conclusion	85

4.1 Introduction

De nombreux problèmes d'optimisation combinatoire consistent à trouver des sous-graphes induits vérifiant une propriété donnée. Par exemple, les propriétés d'être une clique ou d'être un ensemble indépendant sont recherchées dans les problèmes de Clique Maximum ou de Stable Maximum dont la NP-complétude a été montrée par Karp [50]. Dans une approche de généralisation,

Garey et Johnson [41] décrivent les problèmes (GT21 - GT22) de « Sous-Graphe induit vérifiant une propriété Π » et « Sous-Graphe induit connexe vérifiant une propriété Π ». Ils montrent que si la propriété se « comporte bien » alors des résultats peuvent être montrés aisément. Par exemple si Π est une propriété héréditaire et non-triviale, alors le problème de trouver un sous-graphe connexe vérifiant Π est **NP**-complet et quelques résultats d'approximation sont vrais.

On propose de nouveaux résultats de complexité pour un cas particulier du problème de recherche de motif coloré introduit par Lacroix et collab. [53]. Dans sa forme la plus générale, ce problème de recherche de motif coloré consiste à, étant donné un graphe sommet-coloré et un multi-ensemble de couleurs, le motif, trouver un sous-graphe connexe dont les couleurs forment exactement le motif. Par exemple si un graphe était coloré avec, entre autres, les trois couleurs rose, bleu et vert, un motif pourrait être le multi-ensemble $\{\text{rose}, \text{rose}, \text{vert}, \text{vert}, \text{bleu}\}$, une solution serait alors n'importe quel sous-graphe connexe à cinq sommets dont deux sommets sont roses, deux verts et un bleu (voir figure 4.1).

Motif Coloré (MC)

Donnée : $G = (V, E)$ un graphe coloré, \mathcal{M} un multi-ensemble de couleurs.

Question : Existe-t-il un sous-graphe connexe de G dont le multi-ensemble des couleurs est exactement \mathcal{M} ?

Ce problème s'avère difficile à résoudre : Lacroix et collab. [53] montrent qu'il est **NP**-complet, même lorsque le graphe donné est un arbre. Il est revanche, dans ce cas, FPT en prenant comme paramètre la taille du motif. Fellows et collab. [35, 36] améliorent ces résultats à deux reprises en montrant que **Motif Coloré** est **NP**-complet dans les arbres de degré maximum 3, même si le motif est un ensemble (plutôt qu'un multi-ensemble) de couleurs. Ils montrent de plus que le problème est FPT en prenant comme paramètre la taille du motif, et XP en prenant comme paramètre combiné la largeur arborescente et le nombre de couleurs. En revanche, ils soulignent que le problème n'est pas FPT paramétré par le nombre de couleurs. Enfin, ils introduisent également le cas particulier des motifs à deux couleurs et établissent la **NP**-complétude dans ce cas, même si le graphe en entrée est biparti de degré maximum 4. Du point de vue de l'approximation, Dondi et collab. [32] introduisent la

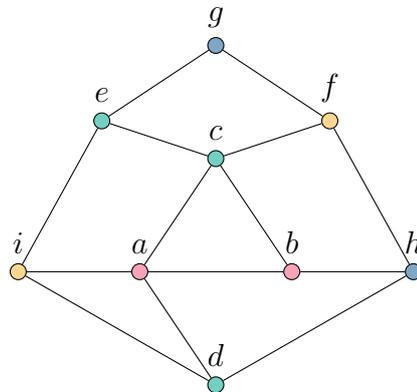


Figure 4.1 – Exemple de recherche de motif dans un graphe 4-coloré. Le motif $\{\text{rose}, \text{rose}, \text{vert}, \text{vert}, \text{bleu}\}$ est retrouvé dans les ensembles de sommets $\{a, b, c, e, g\}$, $\{a, b, c, e, h\}$ et $\{a, b, c, d, h\}$. En revanche l'ensemble de sommets $\{a, b, c, d, g\}$ ne convient pas : le sous-graphe induit n'est pas connexe.

variante d'optimisation du problème qui cherche le plus grand sous-graphe connexe dont le multi-ensemble de couleurs est inclus dans le motif donné. Ils montrent que cette variante est APX-difficile, même si le graphe donné est un arbre de degré maximum 3.

On s'intéresse dans ce chapitre à une variante du problème de recherche de motifs à deux couleurs, comme décrit par Fellows et collab. [35]. Remarquons de prime abord qu'étant donné un graphe G coloré et un motif \mathcal{M} , on peut retirer de G tous les sommets dont la couleur n'apparaît pas dans \mathcal{M} pour obtenir une instance équivalente. Dans ces conditions, on peut supposer sans perte de généralité que le graphe donné est 2-coloré dans le cas où le motif ne contient que deux couleurs. On peut alors définir une variante d'optimisation du problème de recherche de motif à deux couleurs comme l'ont défini Bhore et collab. [6] : étant donné un graphe 2-coloré, trouver un sous-graphe connexe, le plus grand possible, contenant autant de sommets de chaque couleur. On dira qu'un tel sous-graphe est **équilibré**. Plus formellement :

Sous-Graphe Connexe Équilibré (SGCE)

Donnée : $G = (V, E)$ un graphe 2-coloré, $k \in \mathbb{N}$ un entier.

Question : Existe-t-il un sous-graphe connexe de G de taille au moins $2k$ contenant autant de sommets de chaque couleur ?

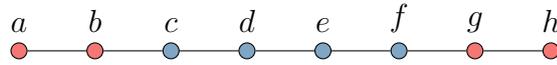


Figure 4.2 – Exemple de sous-graphe connexe équilibré dans un graphe 2-coloré. Ici, $\{b, c\}$, $\{a, b, c, d\}$ et le graphe entier sont connexes et équilibrés mais il n'existe pas de sous-graphe connexe équilibré de taille 6 (pour relier b et g quatre sommets bleus sont nécessaires).

Sous-Graphe Connexe Équilibré diffère du problème de recherche de motif dans la mesure où il existe une discontinuité dans la valeur des solutions : un graphe peut contenir un SGCE de taille $2k$ mais pas forcément de taille $2k - 2$, $2k - 4$ etc. (voir figure 4.2) Si on considère la variante d'optimisation introduite par Dondi et collab. [32], tout sous-graphe d'une solution est lui-même solution. On peut, par contre, résoudre **Sous-Graphe Connexe Équilibré** en résolvant $O(|V|)$ fois **Motif Coloré** (il suffit d'essayer successivement $\{\text{rouge}, \text{bleu}\}$, $\{\text{rouge}, \text{rouge}, \text{bleu}, \text{bleu}\}$, etc.). L'inverse n'est pas aussi direct puisque étant donné un graphe et un motif sur deux couleurs, décider s'il existe une solution à **Motif Coloré** ne peut être directement décidé en résolvant **Sous-Graphe Connexe Équilibré**, eu égard à la discontinuité dans la valeur des solutions.

Bhore et collab. [6] montrent que **Sous-Graphe Connexe Équilibré** est NP-complet dans les graphes bipartis, les graphes de cordes et les graphes planaires. Il est en revanche polynomial dans les arbres, splits, les graphes de diamètre 2 ainsi que les graphes bipartis lorsque la coloration est propre. Comme Bhore et collab. [6] le soulignent, **Sous-Graphe Connexe Équilibré** est en lien avec le problème du **Sous-Graphe Connexe de Poids Maximum** (SGCPM) introduit par Johnson [49] lorsque les poids ont pour valeurs possibles 1 et -1 . Étant donnée une instance de **Sous-Graphe Connexe Équilibré**, on peut affecter un poids 1 aux sommets d'une couleur et un poids -1 aux autres. On cherche alors dans ces conditions le plus grand sous-graphe de poids exactement 0, et non pas de poids maximum.

Sous-Graphe Connexe Équilibré est également proche du problème de l'arbre de Steiner. En effet, savoir si il existe un sous-graphe équilibré connexe contenant **tous** les sommets de la couleur la moins présente dans le graphe, par exemple, peut être décidé en résolvant une instance du problème de l'arbre de Steiner. Il suffit pour cela de considérer que les sommets de ladite couleur sont des terminaux et de chercher un arbre de Steiner de taille au plus deux fois le nombre de terminaux. Dans ce cas, les algorithmes efficaces

pour résoudre le problème de Steiner comme celui de Fuchs et collab. [38], ou encore les différents algorithmes FPT (nombre de terminaux par Dreyfus et Wagner [33] ou plus récemment Björklund et collab. [7], largeur arborescente par Chimani et collab. [21], **etc.**) peuvent être utilisés.

Bhore et collab. [6] décrivent des applications du problème SGCPM et sous-entendent qu'utiliser **Sous-Graphe Connexe Équilibré** puisse être plus adapté dans certains cas. On ne motive pas l'étude du problème par davantage d'applications, mais plutôt par son intérêt théorique. En effet, le problème est, du point de vue de la complexité, assez difficile et ses généralisations sont pléthores. On peut par exemple relâcher la contrainte d'équilibre et la remplacer par un ratio entre le nombre de sommets d'une couleur et le nombre de sommets de l'autre couleur à respecter. On peut également augmenter le nombre de couleurs de la coloration du graphe, imposer des propriétés structurelles supplémentaires à la solution recherchée (être un arbre, un chemin, être 2-connexe ...), ou encore considérer une coloration des arêtes plutôt que des sommets.

4.1.1 Notations

Dans ce chapitre, les graphes considérés sont non-orientés et connexe. Pour un graphe G , on note $\Delta(G)$ (ou plus simplement Δ en l'absence d'ambiguïté) le degré maximum de G . De la même manière, on note $D(G)$ (ou D) son diamètre. Étant donné un sous-ensemble X de sommets ou d'arêtes de G , on note $G[X]$ le sous-graphe induit par X . Étant donné un sommet x d'un graphe $G = (V, E)$, on note $N(x)$ le voisinage ouvert de x : $\{y \in V : xy \in E\}$; et $N[x]$ le voisinage fermé de x : $N[x] = N(x) \cup \{x\}$. Pour un ensemble de sommets D , on note $N(D)$ et $N[D]$ les voisinages ouvert et fermé de D : $N(D) = \cup_{x \in D} N(x)$ et $N[D] = \cup_{x \in D} N[x]$

On suppose sans perte de généralité que lorsqu'un graphe est 2-coloré, les couleurs utilisées sont le **bleu** et le **rouge**, et que **la couleur rouge est la moins fréquente**.

4.1.2 Organisation du chapitre

On montre dans la section 4.2 une série de résultats de **NP**-complétude dans des restrictions à des classes de graphes : les graphes bipartis de diamètre

quatre, les graphes de diamètre 3 et les graphes bipartis 3-réguliers. On propose ensuite dans la section 4.3 une **gap-réduction** pour établir l'inapproximabilité dans le cas des graphes bipartis de degré maximum 4. Enfin, la section 4.4 est dédiée à la preuve d'un algorithme FPT pour le paramètre du nombre de sommets de la couleur la moins représentée (le rouge).

4.2 Complexité classique

4.2.1 Graphes bipartis de diamètre quatre

On montre ici que **Sous-Graphe Connexe Équilibré** reste NP-complet dans les graphes bipartis de diamètre quatre. La réduction encode le problème **Ensemble Dominant** qui consiste étant donné un graphe à trouver un ensemble de sommets (le dominant) tel que tout sommet du graphe appartient au dominant ou en est voisin. **Ensemble Dominant** faisait partie des vingt-et-un problèmes NP-complets de Karp [50].

Ensemble Dominant (DS)

Donnée : $G = (V, E)$ un graphe, $k \in \mathbb{N}$ un entier.

Question : Existe-t-il $D \subset V$ tel que $|D| \leq k$ et $N[D] = V$?

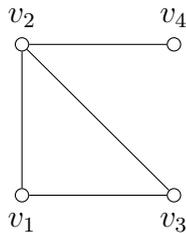
Ce problème reste en fait difficile dans les graphes de diamètre 2 comme le soulignent Lokshtanov et collab. [56]. La construction suivante permet de transformer un graphe de diamètre 2 en un graphe biparti 2-coloré de diamètre quatre.

Construction 4.1

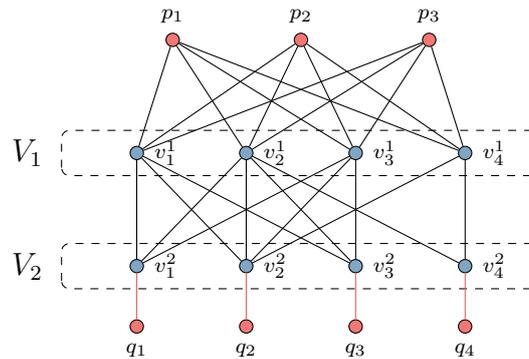
Soit $G = (V, E)$ un graphe à n sommets et $k \in \mathbb{N}$ un entier. On construit G' , graphe 2-coloré comme il suit (voir figure 4.3).

- Créer $2n$ sommets bleus $V_1 = \{v_1^1, \dots, v_n^1\}$ et $V_2 = \{v_1^2, \dots, v_n^2\}$.
- Créer $n + k$ sommets rouges $Q = \{q_1, \dots, q_n\}$ et $P = \{p_1, \dots, p_k\}$.
- Pour tout $i \in \{1, \dots, n\}$
 - Ajouter l'arête $v_i^2 q_i$.
 - Ajouter l'arête $v_i^1 v_i^2$.

- Pour tout $j \in \{1, \dots, k\}$ ajouter l'arête $v_i^1 p_j$.
- Pour toute arête $v_i v_j \in E$, ajouter les arêtes $v_i^1 v_j^2$ et $v_j^1 v_i^2$.



(a) Graphe de diamètre 2.



(b) Graphe obtenu après construction 4.1.

Figure 4.3 – Exemple d'application de la construction 4.1. Dans ce cas $n = 4$ et $k = 3$. À gauche, un graphe de diamètre 2. À droite le graphe obtenu après la construction, de diamètre 4.

La construction 4.1 est polynomiale. Le graphe obtenu est biparti. Si le graphe d'origine G est de diamètre 2 alors G' est de diamètre 4. En effet, toute paire $(x, y) \in V_1 \times V_1$ ou $V_2 \times V_2$ peut être connectée par un chemin de longueur 2 correspondant dans G . Il s'en suit que toute paire $(x, y) \in V_1 \times V_2$ peut être connectée par un chemin de longueur 3. Puisque tous les sommets rouges ont un voisin dans V_1 ou dans V_2 , et $G'[P \cup V_1]$ est un graphe biparti complet, toute paire de sommets rouges peut être connectée par un chemin de longueur au plus 4.

Théorème 4.1

Sous-Graphe Connexe Équilibré est NP-complet dans les graphes bipartis de diamètre 4.

Preuve

Soit G un graphe de diamètre 2, et $k \in \mathbb{N}$. Soit G' le graphe obtenu à partir de G par la construction 4.1. Montrons que G contient un ensemble dominant de taille k si et seulement si G' a un sous-graphe

connexe équilibré de taille $2(n + k)$.

\Rightarrow Si G contient un ensemble dominant D de taille k , alors notons D_1 l'ensemble des sommets de V_1 (dans G') correspondant à D . L'ensemble de sommets $B := \{v \in V' : v \text{ est rouge}\} \cup V_2 \cup D_1$ contient $n + k$ sommets rouges et $n + k$ sommets bleus. Puisque D est un ensemble dominant dans G , tout sommet de V_2 est connecté à au moins un sommet de D_1 . $G'[B]$ est donc connexe et équilibré.

\Leftarrow Si G' contient un SGCE S de taille $2(n + k)$, alors S contient tous les sommets rouges. Comme tous les sommets rouges doivent être connectés entre eux, S contient forcément V_2 puisque tous les sommets de V_2 possèdent tous un sommet pendant rouge. V_2 étant de taille n , S doit contenir exactement k autres sommets bleus pour être équilibré; ces sommets appartiennent à V_1 . Comme V_2 induit un stable, chaque sommet de V_2 doit être connecté à au moins un sommet de V_1 dans S , les sommets de G correspondant à $V_1 \cap S$ forment donc un ensemble dominant de taille k dans G .

Ensemble Dominant étant NP-complet dans les graphes de diamètre 2 [56], et **Sous-Graphe Connexe Équilibré** appartenant clairement à NP, la réduction ci-dessus montre le théorème.

4.2.2 Graphes de diamètre trois

Pour montrer que **Sous-Graphe Connexe Équilibré** reste NP-difficile dans les graphes de diamètre trois, on propose une réduction à partir du problème de **Sous-Graphe Connexe Bariolé** qui étant donné un graphe coloré consiste à trouver un sous-graphe connexe dans lequel les couleurs apparaissent au plus une fois.

Sous-Graphe Connexe Bariolé (SGCB)

Donnée : $G = (V, E)$ un graphe p -coloré, $k \in \mathbb{N}$ un entier

Question : Existe-t-il un sous-graphe connexe de G de taille au moins k dans lequel chaque couleur apparaît au plus une fois?

Ce problème est NP-difficile, à défaut d'une référence pour sa preuve, on

propose d'adapter un résultat de Fellows et collab. [35] sur la recherche de motif. Pour rappel, l'énoncé du problème est le suivant : étant donné $G = (V, E)$ un graphe coloré, et \mathcal{M} un multi-ensemble de couleurs, existe-t-il un sous-graphe connexe de G dont le multi-ensemble des couleurs est exactement \mathcal{M} ?

Fellows et collab. [35] prouvent dans leur théorème 1 que le problème de recherche de motif est **NP-difficile** même lorsque le motif est bariolé - chaque couleur apparaît au plus une fois. En examinant plus attentivement la preuve, il s'avère que le problème reste même **NP-complet** lorsque le motif est l'ensemble de toutes les couleurs. On peut donc utiliser leur résultat pour montrer que **Sous-Graphe Connexe Bariolé** est **NP-difficile**. Pour cela, il suffit de remarquer qu'étant donné un graphe p -coloré, déterminer s'il en existe un sous-graphe connexe bariolé de taille p est **NP-difficile**. Ce problème correspond en effet à chercher un motif égal à l'ensemble de toutes les couleurs que Fellows et collab. [35] montrent comme étant difficile. Le problème de trouver le plus grand sous-graphe connexe bariolé d'un graphe p -coloré donné est donc aussi **NP-difficile**, puisqu'un tel sous-graphe de taille p est le plus grand possible et est déjà difficile à trouver. On a donc le lemme suivant.

Lemme 4.2

Sous-Graphe Connexe Bariolé est **NP-complet**.

On peut à présent réduire **Sous-Graphe Connexe Bariolé** dans les graphes généraux à **Sous-Graphe Connexe Équilibré** dans les graphes de diamètre trois. L'idée de la construction est de faire correspondre une clique formée d'un sommet rouge et de sommets bleus à chaque classe de couleur. On connecte ensuite ces cliques en s'assurant que toute paire de sommets rouges est connectée par un chemin de longueur exactement 3.

Construction 4.2

Soit $G = (V, E)$ un graphe et $c : V \rightarrow \{1, \dots, p\}$ une p -coloration de ses sommets. On construit $G' = (V', E')$ un graphe 2-coloré de la manière suivante (voir figure 4.4).

- $V' = C_1 \cup C_2 \cdots \cup C_p$ avec $C_i = \{x \in V : c(x) = i\} \cup \{r_i\}$.
- Pour tout $i \in \{1, \dots, p\}$, $G'[C_i]$ est une clique dont tous les

sommets sauf r_i sont bleus.

- Pour tout $uv \in E$, ajouter l'arête correspondant dans G' .
- Pour tous C_i, C_j tels qu'il n'existe pas d'arête entre C_i et C_j , ajouter un sommet bleu x_i^j à C_i (connecté à tout C_i) et un sommet bleu x_j^i à C_j (connecté à tout C_j), ainsi que l'arête $x_i^j x_j^i$.

La construction 4.2 est bien polynomiale. Le graphe obtenu est un graphe de diamètre 3 puisqu'il est formé de cliques deux à deux reliées par au moins une arête. On utilise ensuite cette construction dans la preuve du théorème suivant.

Théorème 4.3

Sous-Graphe Connexe Équilibré est NP-complet dans les graphes de diamètre 3.

Preuve

Soit $G = (V, E)$ un graphe et $c : V \rightarrow \{1, \dots, p\}$ une p -coloration de ses sommets. Soit $G' = (V', E')$ le graphe 2-coloré obtenu à partir de G par la construction 4.2. Soit $k \in \mathbb{N}$, $k \geq 3$. On va montrer que G a un sous-graphe connexe bariolé de taille au moins k si et seulement si G' a un sous-graphe connexe équilibré de taille au moins $2k$.

\Rightarrow Si G a un SGCB, S , de taille au moins k , alors les sommets correspondants à S dans G' forment un sous-graphe connexe bleu empruntant au plus un sommet par clique. Il suffit d'ajouter à S les sommets rouges des cliques empruntées pour obtenir un sous-graphe connexe équilibré de taille au moins $2k$ dans G' .

\Leftarrow Si G' a un SGCE, S , de taille au moins $2k$ alors il contient k sommets rouges appartenant à k cliques différentes. Comme les voisinages des sommets rouges sont deux à deux disjoints, chaque sommet rouge a exactement un voisin bleu de sa clique qui appartient à S . Supposons qu'un sommet x_i^j appartienne à S , alors il est l'unique voisin de r_i . Pour pouvoir connecter r_i à un autre sommet rouge, x_j^i doit appartenir à la solution. À présent comme x_j^i appartient à la solution, r_j ne peut avoir d'autre voisin bleu dans la solution. Dans ces conditions, S ne

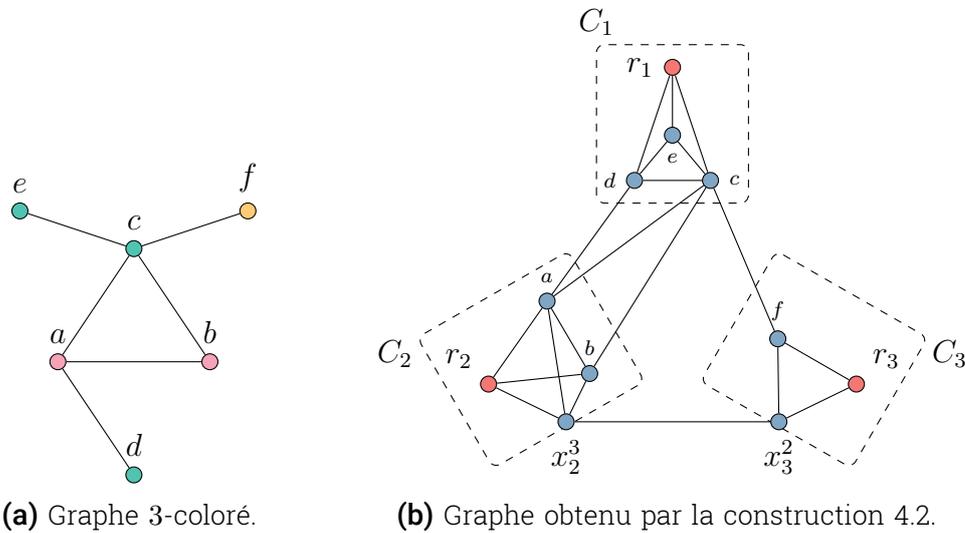


Figure 4.4 – Construction 4.2 appliquée à un graphe 3-coloré. La clique C_1 correspond aux sommets verts c, d et e , la clique C_2 aux sommets roses a et b , et la clique C_3 au sommet orange f . Les sommets x_2^3 et x_3^2 viennent de l'absence d'arête jaune-rose et permettent de relier r_2 et r_3 par un chemin de longueur 3.

peut être de taille supérieure à 4 ce qui est absurde puisque $k \geq 3$ et $|S| \geq 2k$. On a donc chaque sommet rouge qui a exactement un voisin bleu de sa clique dans S et ces sommets correspondent à des sommets de G . Comme les sommets rouges dans S sont de degré 1, l'ensemble des sommets bleus de S est connexe et correspond à un sous-graphe connexe de G . De plus, comme S contient au plus un sommet bleu par clique, l'ensemble de sommets correspondant dans G est bien un SGCB de taille au moins k .

La discussion ci-dessus conclut la réduction polynomiale de **Sous-Graphe Connexe Bariolé** à **Sous-Graphe Connexe Équilibré**. De plus, le problème étant clairement dans NP et le graphe obtenu par la construction 4.2 étant de diamètre 3, on a le théorème.

4.2.3 Graphes bipartis cubiques

On établit ensuite que **Sous-Graphe Connexe Équilibré** reste NP-complet lorsque l'on restreint le graphe en entrée à être biparti cubique. On utilise une réduction depuis **Couverture Exacte par Triplets avec Occurrence 3 (X3C3)** qui consiste, étant donné un ensemble fini X et des triplets de cet ensemble tels que chaque élément de X appartient à exactement trois triplets, à déterminer s'il existe une couverture exacte de X par un sous-ensemble de triplets. Ce problème a été montré NP-complet par Gonzalez [43].

Couverture Exacte par Triplets avec Occurrence 3 (X3C3)

Donnée : X un ensemble fini et C un ensemble de triplets de X tel que chaque élément de X apparaît dans exactement trois triplets.

Question : Existe-t-il $C' \subset C$ tel que $\bigcup_{c \in C'} c = X$ et $c \cap c' = \emptyset$ pour tout $c, c' \in C$?

L'idée générale de la construction est d'encoder chaque ensemble d'une instance de X3C3 par un graphe bleu (un gadget-ensemble) et les éléments de l'ensemble par des sommets rouges. Après ajout de sommets et arêtes idoines, le graphe résultant contient un sous-graphe connexe équilibré si et seulement si l'instance de X3C3 initiale était positive.

Construction 4.3

Soit (X, C) une instance de X3C3, on note $C = \{c_1, \dots, c_{3q}\}$ et $X = \{x_1, \dots, x_{3q}\}$. On suppose sans perte de généralité que q est pair. On construit une instance $G = (V, E)$ de **Sous-Graphe Connexe Équilibré** à partir de (X, C) de la manière suivante (se référer à la figure 4.5).

- Pour tout $x_i \in X$, créer un sommet x_i (un sommet-élément).
- Pour tout $c_i \in C$, noter $c_i = \{x_j, x_k, x_l\}$:
 - Créer un sous-graphe H_i de 14 sommets. On note t_i^1, t_i^2 et t_i^3 (resp. s_i^1, s_i^2 et s_i^3) les sommets de degré 2 en haut (resp. en bas) de H_i .
 - Ajouter les arêtes $s_i^1 x_j, s_i^2 x_k$ et $s_i^3 x_l$.
 - Créer 6 sommets rouges partitionnés en $A_i = \{a_i^1, a_i^2, a_i^3\}$ et $B_i = \{b_i^1, b_i^2, b_i^3\}$ et ajouter les arêtes $a_i^1 t_i^1, a_i^2 t_i^2$ et $a_i^3 t_i^3$. Ajouter les arêtes $a_i^1 b_i^2, a_i^3 b_i^2, a_i^2 b_i^1$ et $a_i^2 b_i^3$;

- Créer 2 (si $i \leq \frac{q}{2}$) ou 3 (sinon) graphes bipartis 2-réguliers à 6 sommets connectés "comme un chemin" par des couplages. Connecter le premier de ces graphes à B_i par un couplage. Créer un sommet rouge (sommet-terminal) p_i connecté aux 3 sommets de degré 2 restants au bout du chemin de graphes bipartis 2-réguliers.
- Pour tout $i \in \{1, \dots, 3q - 1\}$, ajouter les arêtes $a_i^3 b_{i+1}^1$ et $b_i^3 a_{i+1}^1$;
- Ajouter les arêtes $a_1^1 b_1^1$ et $a_{3q}^3 b_{3q}^3$.

La construction 4.3 est bien polynomiale. Le résultat de la construction est bien un graphe biparti cubique. Il contient $24q$ sommets rouges et $93q$ sommets bleus. Le lemme suivant montre que pour connecter tous les sommets-terminaux rouges (p_i) entre eux "beaucoup" de sommets bleus sont nécessaires puisqu'il faut pour chaque sommet-terminal atteindre l'accordéon rouge en passant par au moins deux sommets par graphe biparti sur le chemin; l'idée étant qu'une fois ces sommets nécessaires dans une solution, il ne reste que "peu" de sommets bleus pour connecter les sommets-éléments.

Lemme 4.4

Soit (X, C) une instance de X3C3, $C = \{c_1, \dots, c_{3q}\}$ et $X = \{1, \dots, 3q\}$. Soit $G = (V, E)$ l'instance de **Sous-Graphe Connexe Équilibré** obtenue à partir de (X, C) par la construction 4.3. Un SGSE de taille $48q$ dans G contient au plus $7q$ sommets bleus appartenant à des gadgets-ensembles.

Preuve

Comme $|S| = 48q$, S contient tous les sommets rouges. En particulier S contient $\{p_1, \dots, p_{3q}\}$ et les ensembles B_i . Comme $G[S]$ est connexe, chaque sommet p_i doit être connecté à B_i . Notons λ_i le nombre de sommets (internes) d'un plus court chemin entre p_i et B_i dans $G[S]$. λ_i est au moins 4 lorsque $i \leq \frac{q}{2}$ et au moins 6 le reste du temps. On peut mettre en équation le nombre de sommets nécessaires pour connecter $\{p_1, \dots, p_{3q}\}$ au reste de S :

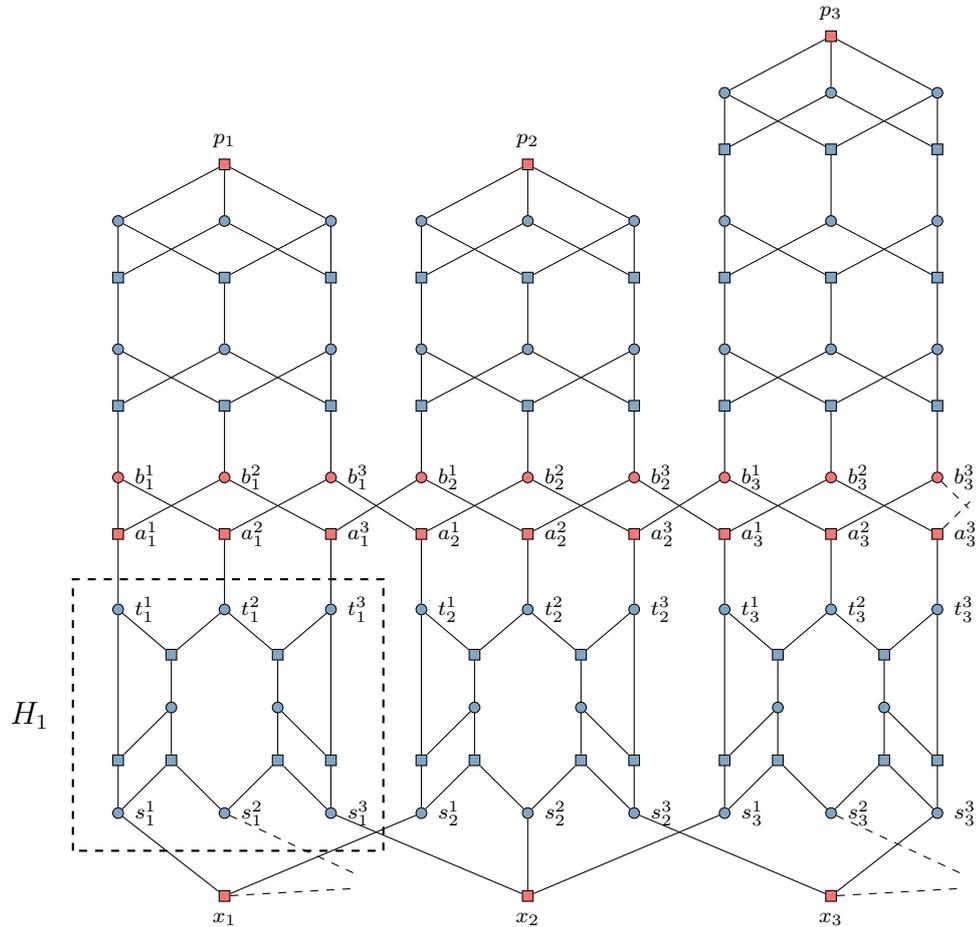


Figure 4.5 – Exemple d'application de la construction 4.3. Une instance de X3C3 est encodée en une instance de SGCE (biparti cubique). La figure est partielle. H_1 est le gadget-ensemble de l'ensemble c_1 . Les sommets s_1^1, s_1^2 et s_1^3 connectent l'ensemble aux éléments qu'il contient : x_1, x_2 et un autre élément qui n'est pas visible sur la figure (pointillé). Les sommets s_1^1, s_1^2 et s_1^3 connectent l'ensemble à l'accordéon rouge qui sépare les gadgets-ensembles des puits rouges p_i . Les puits rouges sont eux reliés à l'accordéon par des chemins de graphes bipartis 2-réguliers.

$$\sum_{i=1}^{3q} \lambda_i = \sum_{i=1}^{\frac{q}{2}} \lambda_i + \sum_{i=\frac{q}{2}+1}^{3q} \lambda_i \quad (4.1)$$

$$\geq 4 \times \frac{q}{2} + 6 \times \frac{5q}{2} \quad (4.2)$$

$$\geq 17q \quad (4.3)$$

Comme S contient $24q$ sommets bleus et au moins $17q$ d'entre eux appartiennent à des chemins connectant les sommets de $\{p_1, \dots, p_{3q}\}$ aux sommets de $\{B_1, \dots, B_{3q}\}$, le nombre de sommets bleus appartenant à des gadgets-ensembles dans S ne peut excéder $7q$ ce qui conclut la preuve.

On montre ensuite l'existence et unicité d'une solution à un programme linéaire en nombres entiers. Ce programme linéaire décrit le coût en sommets bleus de connecter les sommets-éléments aux autres sommets rouges.

Lemme 4.5

Soit $q \in \mathbb{N}$. $u_2 = u_3 = 0$ et $u_1 = q$ est l'unique solution optimale du programme linéaire en nombres entiers (L) suivant.

$$\begin{aligned} \min \quad & 7u_1 + 5u_2 + 3u_3 \\ \text{s.t.} \quad & 3u_1 + 2u_2 + u_3 = 3q \quad (1) \\ & u_1, u_2, u_3 \in \mathbb{N} \end{aligned}$$

Preuve

Soit (u_1, u_2, u_3) une solution optimale de (L) .

- Si $u_2 > 0$, alors $u_3 = 0$ car si $u_3 > 0$ alors $(u_1 + 1, u_2 - 1, u_3 - 1)$ a un meilleur coût que (u_1, u_2, u_3) qui contredit l'optimalité supposée de la solution. En supposant $u_3 = 0$, la contrainte (1) devient $3u_1 + 2u_2 = 3q$ qui implique $u_2 \geq 3$ puisque $u_2 > 0$. Dans ces conditions, $(u_1 + 2, u_2 - 3, u_3)$ a un meilleur coût, ce qui, étant absurde, implique $u_2 = 0$.

- Supposons à présent $u_3 > 0$. Comme $u_2 = 0$, la contrainte (1) est équivalente à $3u_1 + u_3 = 3q$ qui implique $u_3 \geq 3$. Là encore, une absurdité est à remarquer : $(u_1 + 1, u_2, u_3 - 3)$ est de coût strictement meilleur que (u_1, u_2, u_3) , on a donc $u_3 = 0$.

Avec $u_2 = u_3 = 0$, on a forcément $u_1 = q$ pour satisfaire (1) et on a donc montré le lemme.

On peut à présent utiliser les lemmes proposés pour montrer le théorème suivant établissant la NP-complétude de **Sous-Graphe Connexe Équilibré** dans les graphes bipartis.

Théorème 4.6

Sous-Graphe Connexe Équilibré reste NP-complet dans les graphes bipartis (sous)-cubiques.

Preuve

Soit (X, C) une instance de X3C3 et $G = (V, E)$ le graphe obtenu à partir de (X, C) par la construction 4.3. Montrons que (X, C) est une instance positive si et seulement si G contient un SGCE de taille $48q$.

⇐ Si G contient un SGCE S de taille $48q$, alors on montre que S contient exactement q chemins de longueur 7 appartenant à q gadgets-ensembles différents. Pour ce faire, on raisonne sur les différentes manières d'utiliser les gadgets-ensembles pour connecter les éléments. Notons $R \subset V$ l'ensemble des sommets rouges de G et $R^\uparrow \subset R$ l'ensemble des sommets rouges de G qui ne correspondent pas à des éléments (on parle des sommets rouges "du haut").

Comme $|S| = 48q$, S contient tous les sommets rouges. En particulier, S contient tous les sommets-éléments. Pour qu'un élément x soit connecté à R^\uparrow , S doit contenir un chemin de x à R^\uparrow et ce chemin passe par des gadgets-ensembles. Ici deux cas sont à envisager ; dans l'un il existe un chemin entre x et R^\uparrow qui utilise des sommets n'appartenant qu'à un seul gadget-ensemble ; dans l'autre n'importe quel chemin entre x et R^\uparrow passe par plusieurs gadgets jusqu'à atteindre un gadget qui les

connecte à R^\uparrow .

Considérons à présent la figure 4.6. On va affecter à chaque gadget utile un "type" qui correspond à la manière d'utiliser ce gadget pour connecter (au moins) un élément vers le haut dans la solution. On abuse ici la notation : on dit que S contient un gadget alors que l'on devrait dire que S contient au moins un sommet dudit gadget.

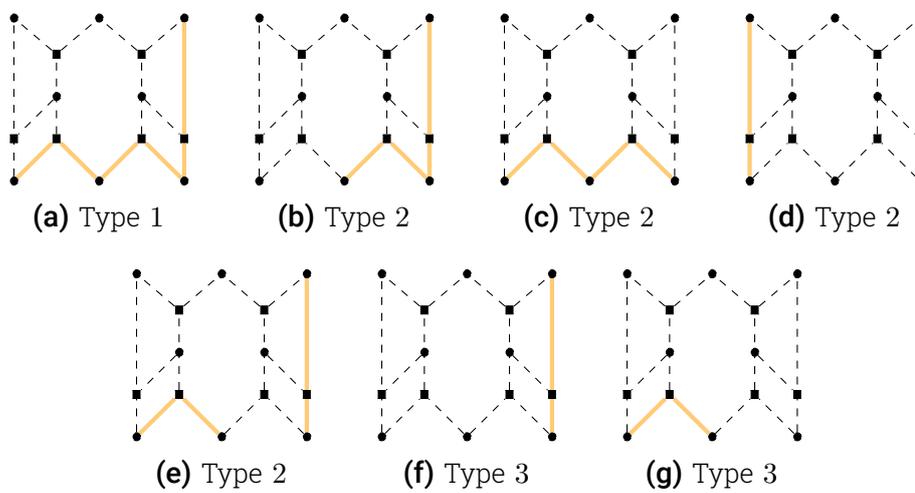


Figure 4.6 – Différents cas d'utilisation des gadgets dans un SGCE. Les arêtes en orange représentent les chemins entre les éléments et le haut de la construction. Le type 1 connecte 3 éléments directement vers le haut en utilisant au moins 7 sommets. Le type 2 en connecte 2 en utilisant au moins 5 sommets et le type 3 en connecte 1 en utilisant au moins 3 sommets.

- Les gadgets de type 1 connectent 3 éléments à R^\uparrow . Au moins 7 sommets de ces gadgets appartiennent à la solution.
- Les gadgets de type 2 connectent 2 éléments à R^\uparrow . Trois cas sont possibles :
 - Les deux éléments sont connectés vers R^\uparrow directement par ce gadget (voir sous-figure 4.6b et sous-figure 4.6d).

- Un des deux éléments est connecté directement, tandis que l'autre est connecté vers le haut par un autre gadget (sous-figure 4.6e).
- Les deux éléments sont connectés par un autre gadget (sous-figure 4.6c).

Dans tous les cas au moins 5 sommets de chaque gadget de type 2 appartiennent à la solution.

- Les gadgets de type 3 connectent 1 élément vers R^\dagger , soit directement (sous-figure 4.6f), soit par un autre gadget (sous-figure 4.6g). Dans les deux cas au moins 3 sommets de chaque gadget de type 3 appartiennent à la solution.

Notons u_1 le nombre de gadgets de type 1, u_2 de type 2 et u_3 de type 3. Par construction, on a $3u_1 + 2u_2 + u_3 = 3q$, et le nombre de sommets bleus de S appartenant à des gadgets-ensembles est au moins $7u_1 + 5u_2 + 3u_3$.

(u_1, u_2, u_3) est une solution au programme linéaire (L) défini dans le lemme 4.5 et $7u_1 + 5u_2 + 3u_3$ est la fonction objective de (L) . Comme (L) a une unique solution optimale de coût $7q$, (u_1, u_2, u_3) doit être cette solution puisque sinon $7u_1 + 5u_2 + 3u_3 > 7q$, ce qui est impossible par lemme 4.4. On a donc $u_1 = q$ et $u_2 = u_3 = 0$. S contient donc exactement q gadgets de type 1, et pour chacun de ces gadgets exactement 7 sommets bleus appartiennent à S . Puisque chaque gadget de type 1 connecte 3 sommets rouges, l'ensemble des gadgets de type 1 dans S correspond à une couverture exacte dans (X, C) .

\Rightarrow Si (X, C) est positive, alors il existe q ensemble de C couvrant exactement X . En prenant tous les sommets rouges dans G , et pour chaque ensemble dans la couverture exacte un chemin bleu de longueur 7 dans le gadget correspondant (sous-figure 4.6a), ainsi que $17q$ sommets bleus pour connecter l'accordéon rouge aux sommets terminaux, on obtient un SGCE de taille $48q$ dans G .

Sous-Graphe Connexe Équilibré appartenant à clairement NP, et X3C3 étant NP-complet, on a montré la NP-complétude de **Sous-Graphe Connexe Équilibré** dans les graphes bipartis cubiques. De plus, certains

arêtes peuvent être retirées du graphe sans invalider le reste de la preuve (deux des trois arêtes incidentes aux sommets terminaux p_i par exemple), le résultat reste donc vrai dans les graphes sous-cubiques.

4.3 Non-approximabilité à facteur constant

Dans cette section, on montre que **Sous-Graphe Connexe Équilibré** n'est pas approximable à facteur constant, même dans les graphes bipartis de degré maximum 4. Commençons par montrer que **Sous-Graphe Connexe Équilibré** n'est pas approximable dans les graphes bipartis. On réutilise la construction proposée par Bhore et collab. [6] qu'ils emploient pour montrer que trouver un SGCE contenant un sommet spécifié est **NP-complet**. On décrira ensuite une technique pour réduire le degré maximum lors de la construction. La réduction part du problème de couverture exacte par triplets, montrée **NP-difficile** par Garey et Johnson [41] :

Couverture Exacte par Triplets (X3C)

Donnée : X un ensemble fini et C un ensemble de triplets de X .

Question : Existe-t-il $C' \subset C$ tel que $\bigcup_{c \in C'} c = X$ et $c \cap c' = \emptyset$ pour tout $c, c' \in C'$?

Le principe est d'encoder une instance de X3C par la représentation en graphe biparti classique. On colore ensuite les sommets correspondant à des ensembles en bleu et ceux correspondant à des éléments en rouge. On place ensuite le sommet obligatoire « loin » des sommets rouges de sorte qu'il soit nécessaire d'inclure un grand nombre de sommets bleus dans tout SGCE contenant le sommet obligatoire.

Construction 4.4

Soit (X, C) une instance de X3C, on note $X = (x_1, \dots, x_{3q})$ et $C = (c_1, \dots, c_m)$. On construit une instance $G = (V, E)$ de **Sous-Graphe Connexe Équilibré** comme il suit (voir figure 4.7).

- Pour tout $x_i \in X$, créer un sommet rouge x_i (sommet-élément).

- Pour tout $c_i \in C$, créer un sommet bleu c_i (sommet-ensemble), et pour tout $x_k \in c_i$, ajouter l'arête $x_k c_i$.
- Créer un sommet bleu y et le connecter à tous les c_i .
- Créer un chemin bleu sur $2q$ sommets dont une extrémité est y et noter x l'autre extrémité.

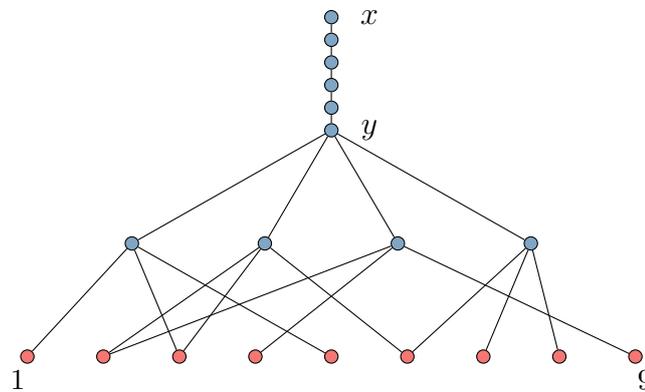


Figure 4.7 – Application de la construction 4.4. Ici l'ensemble est composé de 9 éléments $X = \{1, \dots, 9\}$ et les triplets sont $C = (\{1, 3, 5\}, \{2, 3, 6\}, \{2, 4, 9\}, \{6, 7, 8\})$.

La construction 4.4 est polynomiale. Le graphe obtenu contient $5q + m$ sommets et $4m + 2n - 1$ arêtes. Le lemme 4.7 utilise cette construction pour réduire **Couverture Exacte par Triplets** à **Sous-Graphe Connexe Équilibré** avec sommet obligatoire. La réduction repose sur le fait que pour inclure le sommet obligatoire x (voir figure 4.7) dans un SGCE implique de devoir emprunter un « long » chemin bleu pour atteindre les sommets rouges.

Lemme 4.7

Soit (X, C) une instance de X3C et $G = (V, E)$ le graphe obtenu à partir de (X, C) par la construction 4.4. G a un SGCE contenant x si et seulement si (X, C) est positive. De plus, ledit SGCE est forcément de taille $6q$.

Preuve

⇒ Soit S un SGCE contenant x . S étant équilibré et connexe, il contient le chemin de x à y ($2q$ sommets bleus), puis des sommets-ensembles bleus et enfin des sommets-éléments rouges. Notons l le nombre de sommets-éléments appartenant à S . S contient $2q + l$ sommets bleus. Pour être équilibré, S contient $2q + l$ sommets rouges. Comme chaque sommet-ensemble connecte au plus 3 sommets rouges, S doit contenir au moins $\frac{2q+l}{3}$ sommets-ensembles (pour atteindre $2q+l$ sommets-éléments). On a donc $l \geq \frac{2q+l}{3}$ qui donne $l \geq q$. D'autre part, S contient $2q + l$ sommets rouges, mais le graphe en contient $3q$ on a donc $2q + l \leq 3q$ qui donne $l \leq q$.

On a $l = q$. S contient donc exactement q sommets-ensembles et tous les sommets éléments. Les ensembles correspondants constituent donc une solution à l'instance (X, C) initiale.

⇐ Trivial.

On peut utiliser la construction 4.4 et les mécanismes de la preuve du lemme 4.7 pour établir (par une **gap-reduction**) l'inapproximabilité à facteur constant de **Sous-Graphe Connexe Équilibré** dans les graphes bipartis.

Théorème 4.8

Sous-Graphe Connexe Équilibré dans les graphes bipartis n'est pas approximable à facteur constant, à moins que $P = NP$.

Preuve

Supposons qu'il existe une constante positive $\gamma \in \mathbb{R}^+$, $\gamma > 1$ et un algorithme polynomial γ -approché \mathcal{A} pour **Sous-Graphe Connexe Équilibré**. Pour toute instance de **Sous-Graphe Connexe Équilibré**, si on note K la taille d'une solution retournée par \mathcal{A} et K^* la taille d'une solution optimale, alors on a $K^*/K \leq \gamma$.

Soit p un entier plus grand que γ . Soit (X, C) une instance de **Couverture Exacte par Triplets** et $G = (V, E)$ le graphe obtenu à partir de (X, C) par construction 4.4.

On construit une instance G' de **Sous-Graphe Connexe Équilibré** en

copiant G p fois, et en connectant toutes les copies par un chemin passant par leurs sommets pendants bleus (x), comme illustré en figure 4.3. On note G_1, \dots, G_p les copies de G et x_1, \dots, x_p les sommets extrémités des chemins.

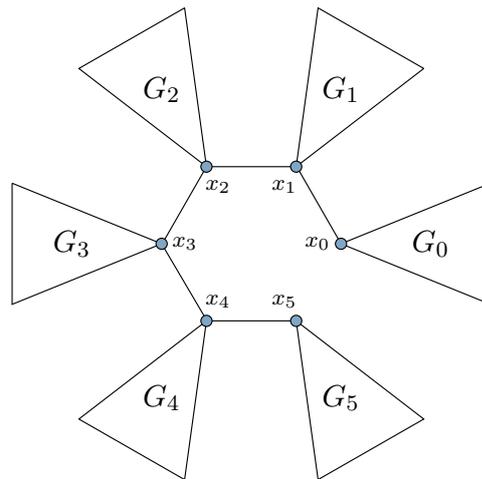


Figure 4.8 – Construction d'une instance arbitrairement grande de **Sous-Graphe Connexe Équilibré** en reliant des copies par un chemin. Pour qu'un SGCE puisse contenir des sommets de deux copies, il doit inclure les sommets x correspondants.

Utilisons maintenant \mathcal{A} sur G' et notons K la taille de la solution retournée. Soit K^* la taille d'une solution optimale. On a deux cas possibles :

- $K^* > 6q$ alors (X, C) est positive. En effet, dans ce cas la solution optimale emprunte des sommets à deux copies différentes G_i et G_j . Pour que la solution soit connexe, il faut qu'elle contienne x_i et x_j . Or, par construction, un sous-graphe connexe de G_i (resp. G_j) contenant x_i (resp. x_j) contient forcément plus de sommets bleus que de rouges. Pour que la solution soit équilibrée, il faut donc qu'elle contienne un SGCE de chaque copie. Enfin, le lemme 4.7 montre que G_i a un SGCE contenant x_i si et seulement si (X, C) est positive.

- $K^* \leq 6q$. Dans ce cas, la solution optimale utilise des sommets dans une unique copie de G , G_i et G_i ne contient pas de SGCE contenant x_i . En utilisant le lemme 4.7, on a (X, C) est négative.

Discutons à présent la taille de la solution retournée par \mathcal{A} .

- Si $K \leq 6q$ alors $K^* \leq 6\gamma q < 6pq$ puisque \mathcal{A} est γ -approché et $p > \gamma$. Dans ce cas, on peut conclure que (X, C) est négative.
- Si $K > 6q$ alors $K^* > 6q$ et (X, C) est positive.

Dans les deux cas, \mathcal{A} résout X3C en temps polynomial, ce qui est absurde si $\mathbf{P} \neq \mathbf{NP}$ puisque X3C est NP-complet. De plus, le graphe G' est biparti par construction, on a donc le théorème.

La construction 4.4 peut être modifiée comme en figure 4.9 pour obtenir un graphe biparti de degré maximum 4. Il suffit pour cela d'ajouter des ensembles et éléments « factices » pour s'assurer que le nombre d'ensembles est une puissance de 2. On remplace ensuite y par un arbre binaire équilibré dont les feuilles sont les sommets-ensembles et la racine est connectée au chemin vers x . On ajoute enfin des sommets pendants rouges aux noeuds internes de l'arbre (voir figure 4.9). Dans ces conditions, si on applique la transformation à des instances de X3C3 (plutôt que de X3C), on peut légèrement modifier la preuve du lemme 4.7 ainsi que du théorème 4.8 pour obtenir le corollaire 4.8.1.

Corollaire 4.8.1

Sous-Graphe Connexe Équilibré dans les graphes bipartis de degré maximum 4 n'est pas approximable à facteur constant, à moins que $\mathbf{P} = \mathbf{NP}$.

4.4 Algorithme polynomial à paramètre fixé

On propose un algorithme FPT pour **Sous-Graphe Connexe Équilibré** paramétré par le nombre de sommets rouges. On utilise pour cela une sous-routine résolvant le problème de l'arbre de Steiner paramétré par le nombre de terminaux. On rappelle l'énoncé du problème de l'arbre de Steiner, qui était l'un des vingt-et-un problèmes NP-complets de Karp [50].

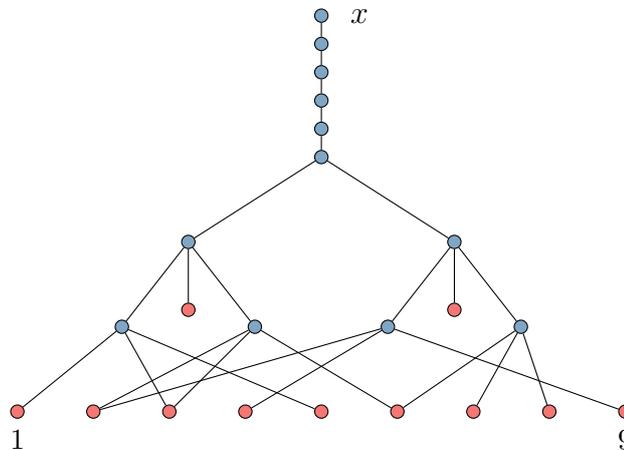


Figure 4.9 – Construction 4.4 adaptée pour obtenir un graphe biparti de degré maximum 4 à partir d'une instance de X3C3. Ici $X = \{1, \dots, 9\}$ et $C = (\{1, 3, 5\}, \{2, 3, 6\}, \{2, 4, 9\}, \{6, 7, 8\})$.

Arbre de Steiner (ST)

Donnée : $G = (V, E)$ un graphe, $T \subset V$, $k \in \mathbb{N}$ un entier.

Question : Existe-t-il $S \subset E$ tel que $|S| \leq k$ et induisant un sous-graphe connexe couvrant T ?

Soit $G = (V, E)$ un graphe dont les sommets sont partitionnés en sommets rouges et sommets bleus. $V = V_r \cup V_b$, et notons $k := |V_r|$. Soit $p \in \mathbb{N}$. L'algorithme suivant décide s'il existe un SGCE de taille $2p$ dans G .

- Pour tout $R \subset V_r$ tel que $|R| \geq p$:
 - Noter G' le graphe obtenu à partir de G en supprimant tous les sommets rouges n'appartenant pas à R .
 - Si R appartient à une composante connexe comportant au moins $|R|$ sommets bleus :
 - Pour toute arête xy de G' , affecter un poids 0 à xy si x et y sont rouges, un poids 1 sinon.
 - S'il existe un arbre de Steiner de terminaux R et de poids au plus $2|R| - 1$ alors répondre **Oui**.
- Répondre **Non**.

Complexité. Dans le pire des cas, l'algorithme énumère tous les sous-ensembles de V_r . Pour chaque sous-ensemble un algorithme FPT résout Steiner paramétré par le nombre de terminaux. Björklund et collab. [7] proposent un algorithme de complexité $O^*(2^k)$, en l'utilisant comme sous-routine notre algorithme aurait une complexité de l'ordre de $\sum_{i=1}^k \binom{k}{i} 2^i = O^*(3^k)$.

Correction. Si l'algorithme répond positivement à une instance, alors il existe un sous-ensemble de sommets rouges, R , appartenant à une composante connexe de G' qui contient au moins $|R|$ sommets bleus. Il existe dans G' un arbre de Steiner couvrant R , de poids au plus $2|R| - 1$. Dans ces conditions, notons B l'ensemble de sommets bleus appartenant à un arbre de Steiner. Comme les arêtes bleu-bleu et rouge-bleu ont un poids de 1, on sait que $|B| \leq |R|$. Si $|B| < |R|$ on peut ajouter à B des sommets bleus connectés à l'arbre (il en existe puisque R appartient à une composante connexe de G' qui contient assez de sommets bleus) pour s'assurer que $|B| = |R|$, et ainsi garantir que $B \cup R$ est un SGCE de G , de taille $2|R|$ donc au moins $2p$.

Complétude. Si une instance positive est fournie à l'algorithme : G a un SGCE $S = B \cup R$ de taille au moins $2p$. Alors R est un sous-ensemble de V_r de taille au moins p . R est donc considéré lors de l'exécution de l'algorithme. Comme $B \cup R$ est un SGCE de G , $B \cup R$ appartient à une composante connexe du graphe G' construit lorsque R est considéré par l'algorithme. Notons qu'un arbre couvrant $B \cup R$ dans G' est de poids $2|R| - 1$ et correspond donc à un arbre de Steiner de poids $2|R| - 1$ qui devrait être trouvé par l'algorithme utilisé en sous-routine pour résoudre le problème de Steiner. En supposant que cette sous-routine est aussi correcte et complète, notre algorithme répondrait **Oui**, d'où la complétude.

4.5 Conclusion

Du point de vue de la complexité, nous avons établi la **NP-complétude** de **Sous-Graphe Connexe Équilibré** dans :

- les graphes bipartis cubiques;
- les graphes bipartis de diamètre 4;
- les graphes de diamètre 3.

Le paysage est relativement bien dépeint puisque le cas des graphes de diamètre 2 est polynomial, et on montre que dans le cas des graphes de diamètre 3 le problème est **NP**-difficile. Le problème est aussi polynomial dans les graphes de degré au plus 2 et on montre qu'il est difficile dans les graphes de degré au plus 3. Il reste à déterminer la complexité dans les graphes bipartis de diamètre 3.

Du point de vue de l'approximation, on a montré l'inapproximabilité du problème dans les graphes bipartis de degré au plus 4. Une question intéressante est celle de l'approximabilité du problème dans des cas encore plus restreints comme les graphes cubiques ou bipartis cubiques. Nos tentatives d'adapter la gap-réduction de la section 4.3 au cas des graphes cubiques ont échoué, et on sait déjà que le problème est difficile dans ce cas.

Même si **Sous-Graphe Connexe Équilibré** reste difficile à résoudre dans des cas très limités, nous avons pu proposer un algorithme FPT « efficace » pour le cas général et un paramètre assez naturel. Il est à noter que les résultats de **NP**-complétude dans les graphes de degré maximum et diamètre bornés impliquent que **Sous-Graphe Connexe Équilibré** ne peut être FPT pour ces paramètres.

Dans l'optique de parachever l'étude de **Sous-Graphe Connexe Équilibré**, plusieurs voies pourraient être explorées. Par exemple compléter l'étude de la complexité du problème lorsqu'il est restreint à des familles de graphes (graphes « blocs » ou graphes d'intervalles par exemple). Un autre axe de recherche serait celui des variantes du problème : variante pondérée, cas d'une k -coloration en entrée, restreindre la structure recherchée (sous-graphe 2-connexe, arbre, chemin, etc.).

CHAPITRE 5

MAXIMISATION DE DURÉE DE VIE DE RÉSEAUX DE CAPTEURS SANS FIL

Contenu

5.1. Introduction	87
5.1.1 Notations	90
5.1.2 Organisation du chapitre	91
5.2. Génération de colonnes	91
5.2.1 Formulation	91
5.2.2 Sous-problème de séparation	93
5.3. Tolérance aux pannes	96
5.3.1 Utilisation de 2-couvertures	96
5.3.2 Rôle de suppléant	98
5.3.3 Stratégie de récupération	100
5.4. Résultats expérimentaux	102
5.5. Conclusion, perspectives	106

5.1 Introduction

Les réseaux de capteurs sans-fil sont de nos jours très en vogue. Ces réseaux sont constitués de dispositifs autonomes qui permettent la collecte périodique ou continue d'information et la centralisation de cette dernière par une ou des

bases de communication. Le plus souvent, les dispositifs sont peu onéreux et on privilégie la quantité à la qualité : pour collecter des informations à plusieurs points clés d'une zone étendue, il est souvent très avantageux (en temps et en coût) d'y déployer un grand nombre de capteurs **bon marché**. Bien entendu, jouir d'un coût moindre s'accompagne de limitations matérielles, comme par exemple l'utilisation de communication sans-fil courte portée (ZigBee, WiFi, Bluetooth).

Les applications étant des plus diverses (comme le montre la centaine de références regroupées par Mohamed et collab. [57]), et les moyens techniques pouvant être limités, la recherche d'algorithmes, stratégies et protocoles optimisés a le vent en poupe. Si obtenir des capteurs plus performants et plus durables est coûteux, optimiser leur fonctionnement logiciel l'est beaucoup moins -- ou en tout cas vite amorti.

Un des problèmes soulevés par l'utilisation de batterie et l'autonomie des dispositifs utilisés est celui de la durée de vie. On s'intéresse dans ce chapitre au problème de maximisation de durée de vie d'un réseau de capteurs sans-fil chargé de la surveillance de cibles, avec centralisation de l'information vers une base de communication.

Introduit par Slijepcevic et Potkonjak [63], ce problème de maximisation de durée de vie (MDV) considérait un ensemble de capteurs et une zone (surface) à surveiller (figure 5.1a), et cherchait à planifier des périodes d'activité et de veille pour chaque capteur afin de maximiser la durée totale de surveillance de la zone. Le cas où seulement des points de la zone sont à surveiller (cibles, voir figure 5.1b) a ensuite été étudié par Berman et collab. [5] et ils ont montré que tout problème de surveillance de zone peut être raisonnablement ramené à la surveillance de cibles. Dans ce problème, l'accent est mis sur la surveillance et la durée de vie, on cherche à planifier des couvertures des cibles en respectant les niveaux de batterie en ignorant les problèmes liés à la communication. Le problème de surveillance de cibles a donc ensuite été davantage étudié [9], [65], et de nombreuses variantes introduites : capteurs directionnels avec modules de surveillance pouvant être tournés [1, 8, 39], surveillance des cibles par plusieurs capteurs [20, 66], couverture d'une proportion α des cibles [42], surveillance par plusieurs capteurs et ajustement des portées [17], **etc.**

Le problème a ensuite été repris par Gupta et collab. [46] (puis [67, 69]) en ajoutant une contrainte de connectivité : les capteurs sont chargés de faire remonter l'information qu'ils récoltent à une station de base. Tout comme MDV,

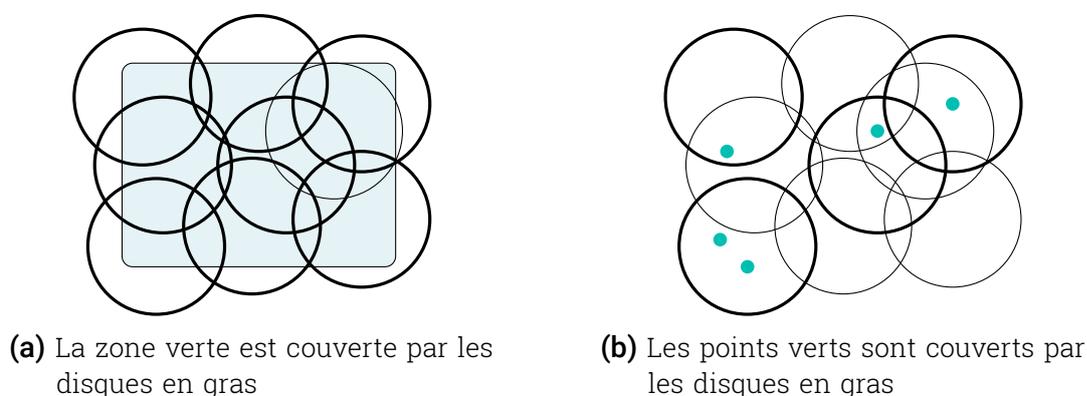


Figure 5.1 – Couvrir des surfaces ou couvrir des points?

ce problème de Maximisation de Durée de Vie avec contrainte de Connectivité (MDVC) a ensuite été largement étudié : surveillance hétérogène (différents types de cibles pour différents types de capteurs) [45], génération de colonnes exacte [29] et approchée [59], portée ajustable [16], ajout de contraintes de bande-passante [60], couverture partielle des cibles [10, 62], surveillance des cibles par plusieurs capteurs [68], **etc.**

La variante étudiée dans ce chapitre est celle de la maximisation de durée de vie avec contrainte de connectivité et affectation de rôles. Introduite par Castaño et collab. [14], cette variante part du principe que la consommation de batterie d'un capteur utilisé pour surveiller une cible n'est pas la même que celle d'un capteur utilisé pour seulement relayer des informations. Le problème consiste donc à planifier les périodes d'activité (surveillance ou relai) et veille des capteurs pour assurer à tout instant la surveillance de toutes les cibles et la remontée de l'information à la base de communication. Une variante avec couverture partielle des cibles [12, 13] a été étudiée, et les techniques de résolution ont été rendues plus rapides [11, 15].

En analysant l'ensemble des solutions apportées à ces variantes avec affectation de rôles, on constate que la robustesse et la tolérance aux pannes est peu considérée. Pourtant, les solutions consistent en général en des couvertures des cibles par des structures minimales qui s'avèrent être des arbres. Or, l'arbre n'est-il pas la structure de réseau la moins tolérante aux pannes qu'il est possible de déployer? Une panne sur un sommet interne d'un arbre utilisé pour une surveillance implique soit une perte de connectivité, soit qu'une cible n'est plus surveillée. Dans les deux cas, la panne invalide la

solution.

La robustesse a pourtant été introduite assez tôt par Cheng et collab. [20], Wang et collab. [66] qui considèrent MDV (sans contrainte de connectivité) avec des contraintes de bande-passante. Dans leur cas, ils souhaitent calculer une planification des surveillances des cibles en garantissant qu'à tout instant une cible est surveillée par au moins k capteurs. Plus proche du cas nous intéressent, Zongheng Zhou et collab. [68] ajoutent à MDVC (avec contrainte de connectivité vers une base, donc) la contrainte que chaque couverture doive garantir la surveillance des cibles par au moins k capteurs. Néanmoins, leur solution n'est tolérante aux pannes qu'au niveau de la surveillance, et non de la connectivité vers la base.

Le but de ce chapitre est de concevoir une nouvelle approche du problème de maximisation de durée des réseaux de capteurs sans fil, avec contrainte de connectivité et affectation de rôles, pour produire des solutions tolérantes aux pannes sous certaines hypothèses.

5.1.1 Notations

On modélise le réseau de capteurs par un graphe orienté $D = (V, A)$, la base est un sommet particulier $r \in V$, aussi appelé **racine** et l'ensemble de cibles à surveiller est un ensemble $T \subset V \setminus \{r\}$ de puits dans le graphe (on peut les atteindre mais jamais en repartir). On peut définir la notion de couverture valide comme il suit.

Définition 5.1 : Couverture valide

Un ensemble de sommets $S \subset V_c$ est une **couverture valide** de T si pour toute cible $t \in T$ il existe un chemin de r à t dans $D[S]$.

À chaque sommet u est assigné un niveau de batterie initial $b_u \in \mathbb{R}^+$. Notons $V_c = V \setminus (T \cup \{r\})$. Le but du problème de maximisation de durée de vie est calculer un ensemble de couvertures valides et les durées associées pour maximiser le temps total de surveillance, tout en respectant les niveaux de batterie des sommets. On ajoute ensuite deux rôles qui correspondent à deux modes d'activité pour les capteurs : surveillance (le capteur utilise son module de surveillance et relaie l'information) ou relai seul (module de surveillance seul). À chacun des rôles on associe un taux de consommation de batterie, α

et β , on peut alors formaliser le problème :

Maximisation de Durée de Vie (connexe, avec rôles)

Donnée : $D = (V, A)$, $r \in V$, $T \subset V$, $b : V \rightarrow \mathbb{R}$, $\alpha, \beta \in \mathbb{R}$, $k \in \mathbb{R}$.

Question : Existe-t-il un ensemble de couvertures valides et durées associées $\mathcal{S} \subset V \times \mathbb{R}$ tel que $\sum_{(S,t) \in \mathcal{S}} t \geq k$ et pour tout $u \in V_c$, $\sum_{(S,t) \in \mathcal{S}} l_S^u t \leq b(u)$, où $l_S^u = \alpha$ si u est surveillant dans S , et β sinon ?

5.1.2 Organisation du chapitre

On commence par rappeler en section 5.2 la formulation en génération de colonnes utilisée dans les nombreux papiers résolvant MDVCR : la formulation du problème en un « grand » problème linéaire, sa décomposition en problème restreint et sous-problème, ainsi que les techniques classiquement utilisées pour résoudre efficacement le sous-problème.

On donne en section 5.3 les éléments nécessaires à l'élaboration de notre approche tolérante aux pannes. D'abord on introduit la notion de 2-couverture; on définit ensuite le rôle de suppléant, un état de basse-consommation qui permet à un capteur d'être sollicité rapidement; on décrit finalement comment notre solution pourrait être déployée.

Enfin, la section 5.4 expose une validation expérimentale de notre approche.

5.2 Génération de colonnes

5.2.1 Formulation

Supposons que l'on dispose d'une énumération de toutes les couvertures valides, $\mathcal{F} = \{f_1, \dots, f_p\}$. On peut alors modéliser le problème de maximisation de durée de vie par le programme linéaire suivant (MP).

$$\begin{aligned} \max \quad & \sum_{j=1}^p t_j \\ \text{s.t.} \quad & \sum_{j=1}^p l_j^u t_j \leq b_u \quad \forall u \in V_c \\ & t_j \geq 0 \quad \forall j \in \{1, \dots, p\} \end{aligned}$$

Où pour tout $j \in \{1, \dots, p\}$, la variable t_j représente la durée d'allumage de la couverture f_j et l_j^u représente la consommation d'énergie du sommet u dans la couverture f_j . Le nombre de couvertures valides est bien entendu très grand aussi le programme linéaire (MP) ne saurait être résolu tel quel. On utilise donc une approche basée sur la génération de colonnes pour espérer le résoudre.

Comme rappelé dans le chapitre 2, la génération de colonnes consiste à résoudre une restriction d'un programme linéaire à un sous-ensemble de variables. On cherche ensuite à déterminer si une variable améliorante existe, puis on résout à nouveau la restriction augmentée d'une variable, on répète ce processus jusqu'à ce qu'aucune variable améliorante ne soit trouvée. Dans notre cas, la restriction de l'ensemble de variables consiste à partir d'une famille $\mathcal{F}' \subset \mathcal{F}$ de couvertures valides. En notant q la taille de \mathcal{F}' , on obtient le programme linéaire (RMP) suivant qui est la restriction de (MP) à q variables.

$$\begin{aligned} \max \quad & \sum_{j=1}^q t_j \\ \text{s.t.} \quad & \sum_{j=1}^q l_j^u t_j \leq b_u \quad \forall u \in V_c \\ & t_j \geq 0 \quad \forall j \in \{1, \dots, q\} \end{aligned}$$

Une fois ce problème résolu, il faut déterminer s'il existe une variable améliorante à ajouter. Pour ce faire, considérons le programme linéaire dual de (RMP).

$$\begin{aligned} \min \quad & \sum_u b_u \pi_u \\ \text{s.t.} \quad & \sum_u l_j^u \pi_u \geq 1 \quad \forall j \in \{1, \dots, q\} \end{aligned}$$

Étant donné une solution t^* de (RMP) et la solution duale π^* associée, une variable améliorante dans (RMP) équivaut à une contrainte non-satisfaite par π^* dans son dual. Dans notre cas, on cherche une couverture valide telle que :

$$\sum_u l_j^u \pi_u^* < 1$$

5.2.2 Sous-problème de séparation

Étant donnée une solution duale π^* , on cherche une couverture valide f_k telle que $\sum_u l_k^u \pi_u^* < 1$. Ce sous-problème est modélisé comme il suit.

Séparation de π^*

Donnée : $D = (V, A)$, $r \in V$, $T \subset V$, $\pi^* : V \rightarrow \mathbb{R}$, $\alpha, \beta \in \mathbb{R}$.

Question : Existe-t-il un ensemble de sommets $S \subset V$ et une affectation $\sigma : S \rightarrow \{a, b\}$ de rôles aux sommets de S tels que :

- pour tout $t \in T$, il existe un chemin de r à t dans $G[S]$;
- pour tout $t \in T$, t a au moins un voisin recevant le rôle a ;
- $\sum_{u \in S} l(u) \pi^*(u) < 1$, où $l(u) = \alpha$ si $\sigma(u) = a$ et β sinon ?

Bien entendu, ce problème de séparation est **NP-complet**, puisque le cas particulier $\alpha = 0$ et $\beta = 1$ correspond au problème de l'arbre de Steiner enraciné sommet-pondéré. On a donc le théorème suivant.

Théorème 5.1

Séparation de π^* est NP-complet.

La différence majeure entre le problème de séparation et l'arbre de Steiner vient de l'affectation de rôles qui est demandée, mais on peut transformer une instance du problème de séparation en une instance de Steiner par la construction suivante.

Construction 5.1

Soit $D = (V, A)$ un graphe orienté, $r \in V$ un sommet racine, $T \subset (V \setminus \{r\})$ des cibles. On note $V_c = V \setminus (T \cup \{r\})$ et on suppose $d^-(r) = 0$ et $d^+(t) = 0$ pour tout terminal t . Soit α une consommation en surveillance, β une consommation en relai, et $\pi^* : V \rightarrow \mathbb{R}^+$ une affectation de poids positifs aux sommets non-terminaux non-racine. On construit un graphe $D' = (V', A')$ en partant de $V' = T \cup \{r\}$, et une pondération ω de ses sommets de la manière suivante.

- Pour tout sommet $u \in V_c$, créer un sommet u^c et un sommet u^s .

- Pour tout terminal $t \in T \cap N^+(u)$ voisin de u , ajouter l'arc $u^s t$.
- Pour tout non-terminal $v \in N^+(u) \setminus T$ voisin de u , ajouter l'arc $u^c v^c$.
- Ajouter l'arc $u^c u^s$.
- $\omega(u^s) = \alpha \pi^*(u)$ et $\omega(u^c) = \beta \pi^*(u)$
- Pour tout $u \in N^+(r)$, ajouter l'arc $r u^c$.
- Pour tout $x \in T \cup r$, $\omega(x) = 0$.

Montrons à présent que la construction 5.1 est une réduction en temps polynomial du problème de séparation au problème de Steiner enraciné.

Théorème 5.2

La construction 5.1 réduit **Séparation de π^*** à **Steiner Enraciné**.

Preuve

Soit $\mathcal{I} = (D = (V, A), r, T, \pi^*)$ une instance du problème de séparation. Soit $D' = (V', A')$ et ω le graphe et la pondération obtenue à partir de D en appliquant la construction 5.1. Montrons qu'il existe une solution à \mathcal{I} si et seulement s'il existe une structure de Steiner enracinée en r , de terminaux T et de poids au plus 1 dans D' .

\Rightarrow Soit (S, σ) une solution au problème de séparation. On la transforme en une structure de Steiner $S' = \{u^c : u \in S\} \cup \{u^s : u \in S \text{ et } \sigma(u) = a\} \cup T \cup \{r\}$. On a alors :

$$\omega(S') = \sum_{u \in S} \beta \pi^*(u) + \sum_{u \in S: \sigma(u)=a} \alpha \pi^*(u) = \sum_{u \in S} l(u) \pi^*(u) < 1$$

S' est donc un ensemble de sommets de poids au plus 1. On peut ensuite aisément vérifier que S' induit bien une structure de Steiner enracinée en r , pour l'ensemble de terminaux T .

\Leftarrow On peut dans ce sens utiliser des arguments similaires pour montrer qu'une structure de Steiner enracinée en r ayant pour terminaux T , de poids au plus 1, dans D' équivaut à une solution au problème de séparation.

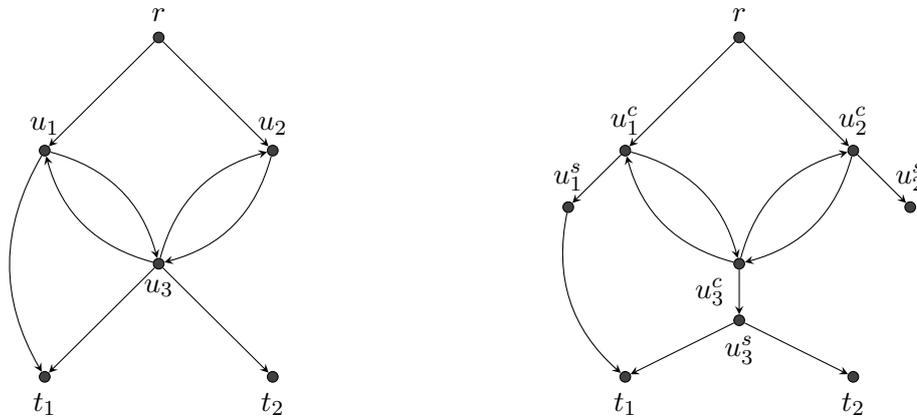


Figure 5.2 – Application de la construction 5.1. À gauche un graphe orienté de racine r et cibles t_1, t_2 . À droite le graphe obtenu, chaque sommet non-terminal a été dupliqué, une copie conservant le voisinage vers les autres sommets non-terminaux, l'autre copie étant éventuellement reliée aux terminaux.

On peut donc utiliser les méthodes exactes permettant de résoudre le problème de Steiner pour résoudre le problème de séparation. On donne une formulation de Steiner en programme linéaire en nombres entiers. À chaque sommet du graphe on associe une variable binaire traduisant la présence ou non du sommet dans la structure de Steiner recherchée. On emploie une formulation similaire à celle exposée dans le chapitre 2 dans laquelle le problème était modélisé avec une variable binaire y_a pour chaque arc a , et une contrainte pour tout ensemble d'arcs déconnectant la racine d'au moins un terminal (une **coupe**), au moins un arc doit appartenir à la structure recherchée. Dans le cas de variables sur les sommets, on remplace les coupes par des **séparateurs** (ensembles de sommets déconnectant la racine d'au moins un terminal).

$$\begin{aligned}
 \min \quad & \sum_{u \in V_c} w_u x_u \\
 \text{s.t.} \quad & \sum_{u \in \Pi} x_u \geq 1 \quad \forall \Pi \subset V_c \text{ séparateur} \\
 & x_u \in \{0, 1\} \quad \forall u \in V_c
 \end{aligned}$$

5.3 Tolérance aux pannes

Force est de constater une certaine fragilité de la solution calculée par la méthode décrite dans la section précédente. En effet, dans une solution optimale chaque couverture assurant la connectivité et la surveillance des cibles a de grandes chances d'être minimale puisque si une couverture s'avérait ne pas l'être, on pourrait en retirer un capteur et donc économiser de la batterie de ce capteur pour potentiellement allonger la durée de vie du réseau. Si on considère que chaque couverture dans une solution optimale est minimale, alors la tolérance aux pannes du réseau est nulle : si une panne survient sur un capteur en fonctionnement, le réseau est déconnecté et la surveillance est interrompue. Pire encore, le capteur défectueux pourrait déconnecter des couvertures ultérieures s'il devait être utilisé plusieurs fois.

On pourrait supposer qu'une vérification de la continuité de surveillance puisse être faite au niveau de la base de communication. Dans ce cas, un protocole de rétablissement pourrait être mis en place. La base pourrait par exemple détecter les pannes assez rapidement, puis retirer de la planification les capteurs tombés en panne et informer le réseau du changement de planification. On aurait dans ce cas une interruption de la surveillance à compter de la panne jusqu'à la distribution de la nouvelle planification aux capteurs.

Ici plusieurs stratégies peuvent être adoptées (voir figure 5.3), la base pourrait rapidement retirer les couvertures invalidées par la panne (figure 5.3b) ou être autorisée à recalculer l'optimal une fois la panne connue (figure 5.3c). Dans le second cas, le temps d'interruption serait plus long, mais la durée de vie pourrait être grandement augmentée par rapport au premier cas.

On propose dans la suite une approche basée sur le recalcul optimal mentionné plus haut, et une stratégie permettant sous certaines hypothèses de ne pas interrompre la surveillance en cas de panne. Les pannes considérées interviennent sur des capteurs et sont définitives.

5.3.1 Utilisation de 2-couvertures

Dans un premier temps, on va étendre la notion de couverture valide à celle de la 2-couverture.

On rappelle qu'on considère un réseau de capteurs : un graphe orienté

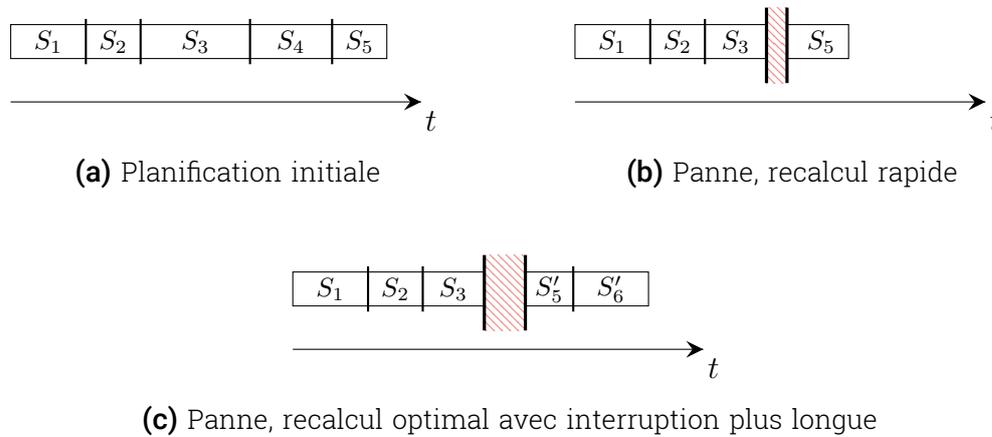


Figure 5.3 – Récupération de panne avec interruption. (a), la planification initiale constituée de cinq couvertures. (b), une panne a lieu durant la couverture S_3 . La surveillance est interrompue durant la période hachurée en rouge. Le capteur défectueux était utilisé dans la couverture S_4 également, la nouvelle planification est donc privée de S_4 . (c), même panne, cette fois l'interruption est plus longue mais une nouvelle planification optimale est calculée, la durée de vie est donc allongée.

$D = (V, A)$, un sommet particulier $r \in V$, un ensemble de cibles $T \subset V \setminus \{r\}$, et un niveau de batterie initial $b_v \in \mathbb{R}^+$ pour chaque sommet v . On rappelle que l'on note $V_c = V \setminus (T \cup \{r\})$. On dit qu'un sous-ensemble de sommets $S \subset V_c$ est une **2-couverture valide** si pour tout $t \in T$, il existe deux chemins sommets-disjoints de r à t . On peut remarquer que si on remplace les couvertures valides par des 2-couvertures valides dans une solution, la récupération d'une panne n'implique plus d'interrompre la surveillance, en supposant que les pannes soient raisonnablement espacées. En revanche, avoir recours à des 2-couvertures occasionne une surconsommation de batterie considérable puisque, par exemple, dans une 2-couverture chaque cible doit être voisine d'au moins deux capteurs. On peut construire des instances dans lesquelles la durée de vie du réseau est divisée par deux lorsque des 2-couvertures sont utilisées, comme illustré en figure 5.4.

En revanche, remplacer les couvertures par des 2-couvertures lors de la résolution du problème de planification est aisé. On peut conserver la même formulation en génération de colonne, sauf que les colonnes représentent maintenant des 2-couvertures. On peut ensuite modifier une valeur dans la formulation en programme linéaire du problème de Steiner employé pour

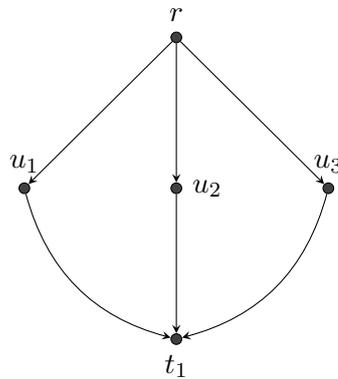


Figure 5.4 – Ici, en supposant que u_1, u_2 et u_3 aient une autonomie de surveillance d'une heure, une planification optimale avec des couvertures simples consiste à utiliser u_1 puis u_2 puis u_3 , une heure à chaque fois, pour surveiller t_1 durant trois heures. Dans le cas où des 2-couvertures sont requises, une solution optimale serait d'utiliser les ensembles $\{u_1, u_2\}$, $\{u_1, u_3\}$, $\{u_2, u_3\}$ une demie-heure chacun, pour une durée de vie totale d'une heure trente -- deux fois moins.

résoudre le sous-problème de séparation (défini en section 5.2) pour traduire 2-connexité recherchée.

$$\begin{aligned} \min \quad & \sum_{u \in V_c} w_u x_u \\ \text{s.t.} \quad & \sum_{u \in \Pi} x_u \geq \mathbf{2} \quad \forall \Pi \subset V_c \text{ séparateur} \\ & x_u \in \{0, 1\} \quad \forall u \in V_c \end{aligned}$$

Une solution optimale doit contenir au moins deux sommets par ensemble séparant la racine des terminaux.

5.3.2 Rôle de suppléant

Pour palier au problème de surconsommation engendré par l'utilisation de 2-couvertures, on introduit d'un troisième rôle, le rôle de **suppléant**, dont le taux de consommation serait largement réduit. Une 2-couverture serait alors décomposable en un ensemble de capteurs formant une couverture valide, et un ensemble de suppléants garantissant l'existence de deux chemins sommet-disjoints de la racine à toute cible.

Reprenons l'exemple exposé en figure 5.4. Supposons que le taux de consommation du rôle de suppléant soit de 25% par rapport à la consommation en surveillance, ainsi les capteurs consomment 100% de leur batterie en surveillant une heure, alors qu'ils n'en consomment que 25% en suppléance. La planification optimale de 2-couvertures sans le rôle de suppléant était $\{u_1, u_2\}$, $\{u_1, u_3\}$ et $\{u_2, u_3\}$ trente minutes chacun. En ajoutant le rôle de suppléant, la solution optimale devient $\{u_1, u_2^*\}$, $\{u_1^*, u_3\}$ et $\{u_2, u_3^*\}$ où les sommets marqués d'une astérisque sont suppléants, on peut alors utiliser chaque couverture quarante-huit minutes sans excéder l'autonomie de chaque capteurs. La durée de vie passe d'une heure trente à deux heures vingt-quatre.

Pour résoudre le problème de maximisation de durée de vie avec utilisation de 2-couvertures et suppléance, il suffit, là encore, de modifier le sous-problème. La formulation en génération de colonnes reste valide, mais l'ensemble \mathcal{F} des couvertures devient l'ensemble des 2-couvertures avec affectation (possible) du rôle de suppléant. On rappelle la formulation décrite en section 5.2 :

$$\begin{aligned} \max \quad & \sum_{j=1}^q t_j \\ \text{s.t.} \quad & \sum_{j=1}^q l_j^u t_j \leq b_u \quad \forall u \in V_c \end{aligned}$$

t_j étant le temps d'utilisation de la couverture numérotée j , étant donnée une énumération de l'ensemble des 2-couvertures possibles, et l_j^u le taux de consommation du capteur u dans la couverture j . Une fois le programme linéaire résolu et un vecteur dual π^* obtenu, le problème de séparation s'exprime comme il suit, en notant γ le taux de consommation en suppléance.

Séparation de π^* (avec suppléance)

Donnée : $D = (V, A)$ un graphe orienté, $r \in V$, $T \subset V$, $\pi^* : V \rightarrow \mathbb{R}$, $\alpha, \beta, \gamma \in \mathbb{R}$.

Question : Existe-t-il un ensemble de sommets $S \subset V$ et une affectation

$\sigma : S \rightarrow \{a, b, c\}$ de rôles à S tels qu'en notant $S_x = \{u \in S : \sigma(u) = x\}$

- $\forall t \in T$, il existe un chemin de r à t passant par $S_a \cup S_b$;
- $\forall t \in T$, il existe 2 chemins disjoints de r à t passant par S ;
- $\forall t \in T$, t a un voisin entrant recevant le rôle a ;
- $\sum_{u \in S} l_S^u \pi^*(u) < 1$.

Pour résoudre ce problème de séparation, on transforme le graphe en utilisant la construction 5.1. Cette transformation permet de s'affranchir de l'affectation des rôles de surveillance et relai. On reprend la formulation en programme linéaire en nombres entiers décrite en section 5.2.2, à laquelle on va ajouter une famille de variable z telle que z_u vaille 1 si le sommet u est utilisé comme suppléant et 0 sinon. Une variable x_u vaut, quant à elle, 1 si le sommet u est actif dans la couverture.

$$\begin{aligned} \min \quad & \sum_{u \in V_c} w_u x_u + \gamma w_u z_u \\ \text{s.t.} \quad & \sum_{u \in \Pi} x_u \geq 1 & \forall \Pi \subset V_c \text{ séparateur} & (1) \\ & \sum_{u \in \Pi} x_u + z_u \geq 2 & \forall \Pi \subset V_c \text{ séparateur} & (2) \\ & x_u + z_u \leq 1 & \forall u \in V_c & (3) \\ & x_u, z_u \in \{0, 1\} & \forall u \in V_c & \end{aligned}$$

- (1) Au moins un sommet actif par ensemble de sommets séparant la racine d'au moins un terminal.
- (2) Au moins deux sommets (actifs ou suppléants) par ensemble de sommets séparant la racine d'au moins un terminal.
- (3) Un sommet ne peut pas être actif et suppléant à la fois.

5.3.3 Stratégie de récupération

Les planifications calculées sont maintenant des ensembles de 2-couvertures et les temps d'allumage associés. Pour chaque 2-couverture, l'ensemble des sommets surveillants et relais forme une couverture valide et le reste des sommets garantit l'existence de deux chemins sommet-disjoints de la racine à tout terminal. Pour autant, la tolérance aux pannes est toujours nulle. En effet, la planification respecte les niveaux de batterie uniquement dans le cas où aucune panne ne survient. Reprenons l'exemple illustré en figure 5.4. La planification optimale avec 2-couvertures et suppléance est $\{u_1, u_2^*\}$, $\{u_1^*, u_3\}$ et $\{u_2, u_3^*\}$ pour quarante-huit minutes chacun. Si le sommet u_2 tombe en panne à la centième minute (durant la troisième couverture), le sommet u_3 doit devenir actif pour assurer la continuité de la surveillance. Seulement, il reste quarante-quatre minutes à tenir dans cette couverture et le sommet u_3 a déjà été utilisé quarante-huit minutes dans la couverture précédente;

il ne reste donc que douze minutes d'autonomie à u_3 et si rien n'est fait, la surveillance sera interrompue bien avant la fin planifiée.

En ne sachant pas, a priori, quand et où les pannes surviendront, il n'est pas possible de produire une planification « statique » qui permettra de garantir la non-discontinuité d'une surveillance. On propose donc de dynamiser cette planification en recalculant une nouvelle solution à chaque panne. On introduit le **temps de récupération** d'un réseau de capteurs qui correspond au temps nécessaire pour détecter une panne, basculer sur les sommets suppléants nécessaires, faire remonter l'information de la panne à la base, recalculer une planification et enfin la redistribuer. On formule deux hypothèses pour élaborer notre stratégie de récupération.

1. Lorsque le réseau est en récupération, aucune autre panne n'a lieu.
2. Les pannes ne touchent pas les capteurs éteints.

La première hypothèse permet de garantir que les 2-couvertures suffisent à assurer la continuité de la surveillance pendant la récupération. Elle reste réaliste si la durée de vie totale du réseau est grande et le temps de récupération court en rapport. Si on considère un réseau dont les capteurs ont une autonomie de plusieurs semaines et que le temps de récupération du réseau est de, par exemple, dix minutes.

La seconde hypothèse n'est pas indispensable au bien fondé de notre approche, elle permet de s'affranchir des problématiques liées à la détection de panne.

La figure 5.5 schématise la stratégie élaborée autour du temps de récupération. Une fois qu'une panne est détectée, l'information remonte à la base et celle-ci peut mettre à jour ses données sur les niveaux de batterie des capteurs du réseau. Elle dispose ensuite d'un temps ε pour recalculer une solution et la redistribuer au réseau.

Il ne reste qu'à apporter la dernière modification à notre génération de colonne pour prendre en compte le temps de récupération. On souhaite qu'un capteur en suppléance puisse tenir en surveillance au moins ε pour que la surveillance soit maintenue jusqu'au changement de planification. Il suffit pour cela de considérer que le niveau de batterie initial de chaque capteur est raccourci d'un temps ε , (RMP) devient alors :

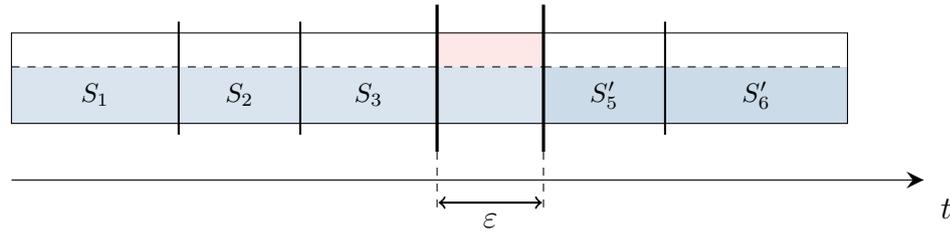


Figure 5.5 – Récupération du réseau en cas de panne. Dans le cas de la suppléance, les couvertures ont une partie active (schématiquement en bleu, sous les pointillés), et une partie suppléante (au dessus des pointillés). Une panne survient sur un capteur actif durant la couverture S_3 , il y a alors allumage des capteurs suppléants nécessaires (en rouge) durant le temps de récupération ε . À l'issue de la récupération, une nouvelle planification a été distribuée au réseau et la couverture S'_5 débute.

$$\begin{aligned} \max \quad & \sum_{j=1}^q t_j \\ \text{s.t.} \quad & \sum_{j=1}^q l_j^u t_j \leq b_u - \varepsilon \quad \forall u \in V_c \end{aligned}$$

En prenant en compte toutes les modifications décrites dans les sections précédentes, les planifications calculées sont constituées de 2-couvertures avec affectation de suppléants. Si aucune panne n'a lieu, la surveillance se termine avec ε de batterie restante dans chaque capteur. En supposant, là encore, que ε est petit par rapport à la durée de vie totale, ce reste n'est pas gênant.

En cas de panne sur un sommet v , la base passe le niveau de batterie de v à 0 et met à jour les autres niveaux de batterie avec leur consommation réelle. La génération de colonnes est alors relancée, et on voit ici qu'elle peut tout à fait repartir de l'ensemble de colonnes générées lors du calcul de la planification avant la panne. La figure 5.6 (à la fin de ce chapitre) montre un exemple d'utilisation de 2-couvertures et temps de récupération.

5.4 Résultats expérimentaux

L'algorithme de génération de colonnes avec 2-couvertures et suppléance ainsi qu'une simulation des pannes/recalcul ont pu être implémentées. On

propose une preuve de concept en comparant la durée de vie obtenue en simulant notre approche, qu'on appellera 2COSUP, avec celle obtenue par une stratégie peu réaliste : on calcule une planification de couvertures simples, sans suppléance, optimale ; à chaque panne, on suppose que le temps de récupération du réseau est instantané et une nouvelle planification optimale est calculée. On appellera cette stratégie, par la suite, RUR (Récupération Ultra Rapide).

L'implémentation utilise le langage **C++** et sa bibliothèque de graphes **Boost**¹ développée par Siek et collab. [61]. Les générations de colonnes utilisées pour les stratégies RUR et 2COSUP utilisent le solveur CPLEX 12.9.

Les caractéristiques des instances utilisées pour les comparatifs sont regroupées dans la table 5.1.

Les instances sont des graphes de disques unitaires. Pour générer une instance, des points sont générés aléatoirement, de coordonnées allant de 0 à 400. Les portées de communication (R_c) et de surveillance (R_s) sont supposées égales à 100. Si les unités considérées sont des mètres, cela correspond à des communications sans fil sur une distance de 100 mètres (raisonnable pour du Bluetooth ou WiFi), et une surveillance à distance de 100 mètres (raisonnable pour une surveillance d'activité sismique), sur une superficie de 160000 mètres carrés (une vingtaine d'hectares).

On suppose que tous les sommets ont la même batterie initiale, on peut alors normaliser : $b_u = 1$ pour tout sommet u -- chaque sommet a une autonomie de 1 unité de temps à consommation maximale. On suppose que la consommation maximale d'un sommet survient en surveillance, on pose donc $\alpha = 1$. On exprime ensuite la consommation en relai comme étant 80% de la consommation en surveillance, et la consommation en suppléance 10% : $\beta = 0.8$ et $\gamma = 0.1$. Enfin le temps de récupération du réseau est supposé égal à 0.01. Pour mettre ces valeurs en perspective, supposons que l'unité de temps soit de 1000 heures (une quarantaine de jours), les sommets auraient une autonomie de 1000 heures en surveillance, 1250 heures en relai et 10000 heures en suppléance ; le temps de récupération du réseau serait de 10 heures.

Pour comparer 2COSUP et RUR, des scénarios de panne ont du être générés. On a défini la probabilité de panne en fonction du temps t comme étant :

$$P(t) = 1 - e^{-t}$$

1. https://www.boost.org/doc/libs/1_66_0/libs/graph/doc/index.html

Table 5.1 – Classes d'instances considérées. Dans tous les cas, on a $R_s = R_c = 100$, $\varepsilon = 0.01$, $\alpha = 1$, $\beta = 0.8$, $\gamma = 0.1$, et $b_v = 1$ pour tout sommet v .

Instances	#sommets	#cibles
r_CMLP_1 à r_CMLP_5	25	3
r_CMLP_6 à r_CMLP_10	30	3
r_CMLP_11 à r_CMLP_15	35	4
r_CMLP_16 à r_CMLP_20	40	4
r_CMLP_41 à r_CMLP_45	25	5
r_CMLP_46 à r_CMLP_50	30	6
r_CMLP_51 à r_CMLP_55	35	7
r_CMLP_56 à r_CMLP_60	40	8

On a ensuite mis en place le processus suivant pour la génération de panne :

1. Calculer une planification $[(S_1, t_1), (S_2, t_2), \dots, (S_k, t_k)]$.
2. Générer une date de panne t_p en utilisant $P(t)$.
3. Identifier la couverture S_j active à t_p , si elle existe, sélectionner aléatoirement un capteur actif dans S_j .
4. Mettre à jour les batteries et retourner au 1 si une panne a été générée.

Pour réaliser une comparaison « équitable », 2COSUP et RUR ont été comparés sur les mêmes scénarios de panne. Pour chaque instance, 10 scénarios ont été générés en appliquant le processus ci-dessus à 2COSUP et en conservant les paires de date/capteurs de chaque panne; RUR a ensuite été confronté aux mêmes pannes. De la même manière, 10 scénarios ont été générés en partant de RUR. On a récupéré pour chaque scénario la durée de vie calculée par 2SUPCO et celle calculée par RUR.

La table 5.2 présente les résultats obtenus en moyenne sur chaque groupe d'instances dont les paramètres sont similaires. Les colonnes 2COSUP et RUR donnent les durées de vie moyennes obtenues pour chacune des stratégies, la dernière colonne donne l'écart relatif $(RUR - 2COSUP)/RUR$. Le détail par instance est donné dans la table 5.4 (en fin de chapitre), qui renseigne également le nombre de pannes moyen.

Table 5.2 – Résultats par groupe d'instances similaires

Instances	2COSUP	RUR	Écart relatif
r_CMLP_1-5	1.61	2.12	26.0%
r_CMLP_6-10	2.38	2.85	17.6%
r_CMLP_11-15	2.48	3.01	18.1%
r_CMLP_16-20	3.24	3.93	16.4%
r_CMLP_41-45	1.63	2.10	23.2%
r_CMLP_46-50	2.16	2.52	14.2%
r_CMLP_51-55	2.51	2.96	15.8%
r_CMLP_56-60	2.67	3.27	18.0%

On constate une durée de vie avec 2COSUP d'en moyenne 19% plus courte qu'avec RUR. Ces résultats sont très encourageants puisqu'on peut s'attendre à perdre environ 10% de durée de vie ne serait-ce que par la surconsommation occasionnée par le rôle de suppléant ($\gamma = 0.1$) dans les voisinages des cibles, et il faut rappeler que RUR est une stratégie loin d'être réaliste.

La table 5.3 oppose les durées de vie moyennes, en l'absence de pannes, obtenues par 2COSUP aux durées de vie optimales (colonne OPT). On observe que la durée de vie obtenue par 2COSUP en l'absence de panne est environ 10% plus courte que la durée de vie optimale et cet écart varie peu.

On peut expliquer cette régularité en raisonnant sur la consommation des séparateurs d'une instance : pour tout ensemble de sommets séparant une cible de la base, une couverture valide contient au moins un sommet en relai dans cet ensemble. On peut dire que la consommation totale des sommets du séparateur est d'au moins $\beta = 0.8$ par unité de temps. Si maintenant le plus petit séparateur est de taille 3, on ne peut espérer une durée de vie totale de plus de 3.75 unités de temps puisque le total de batterie consommé par le séparateur serait $3.75 * 0.8 = 3$, qui égale le total de batterie (1 par sommet) du séparateur.

En utilisant 2COSUP, toute 2-couverture devra avoir un sommet relai et, au moins, un sommet suppléant par séparateur. On peut dire, dans ce cas, que la consommation par séparateur est d'au moins $\beta + \gamma = 0.9$ par unité de temps. En reprenant l'exemple du plus petit séparateur de taille 3, la durée de vie

Table 5.3 – Résultats par groupe d'instances similaires en l'absence de panne

Instances	2COSUP	OPT	Écart relatif
r_CMLP_1-5	2.33	2.61	10.69%
r_CMLP_6-10	3.12	3.49	10.43%
r_CMLP_11-15	3.26	3.62	10.00%
r_CMLP_16-20	4.47	4.98	10.13%
r_CMLP_41-45	2.38	2.63	9.49%
r_CMLP_46-50	2.65	2.98	10.92%
r_CMLP_51-55	3.27	3.67	10.83%
r_CMLP_56-60	3.41	3.76	9.41%

totale ne pourra excéder $\frac{10}{3}$ en utilisant 2COSUP puisque le total de batterie consommé au sein du séparateur serait $\frac{10}{3} * 0.9 = 3$.

Si ces deux bornes étaient atteintes, on aurait un écart relatif d'environ 11%. Un raisonnement similaire pourrait être fait sur les voisinages des cibles, pour donner des écarts relatifs d'environ 9%. On retrouve bien l'écart relatif moyen d'une dizaine de pourcents mis en exergue par la table 5.3.

5.5 Conclusion, perspectives

On a conçu une nouvelle approche de la maximisation de durée de vie dans les réseaux de capteurs. Pour rendre la planification plus tolérante aux pannes, on a proposé de remplacer les classiques couvertures valides par des 2-couvertures. Ces 2-couvertures engendraient une forte surconsommation dans la réseau pouvant diviser par deux la durée de vie. Pour palier à cette surconsommation, on a introduit un troisième mode pour les capteurs : le mode suppléant. Dans ce mode, on suppose qu'un capteur consomme peu de batterie mais est prêt à être utilisé en cas de panne. Enfin, pour modéliser le problème on émis l'hypothèse que chaque réseau avait un temps de récupération ϵ durant lequel une panne peut être détectée, centralisée et un nouveau planning calculé.

On a comparé notre approche avec une stratégie de récupération de panne

peu réaliste : des couvertures (simples) sont planifiées, à chaque panne une planification de couverture est calculée et redistribuée instantanément. On a vu que rendre réaliste cette approche en utilisant la notre « coûtait » environ 19% de durée de vie.

De nombreuses perspectives sont à explorer. Dans un premier temps la performance des algorithmes peut être largement améliorée. Ensuite, les variations classiques proposées dans la littérature peuvent aussi être adaptées au problème : couvertures partielles, capteurs hétérogènes, portée ajustable, **etc.**

La nature des pannes constitue un axe de recherche à part entière. Dans notre cas les pannes ont lieu sur des sommets, et un sommet en panne l'est définitivement. On pourrait supposer que les pannes sont temporaires ou toucher des arêtes (apparition d'obstacles). Pour chaque type de panne, il faudrait adapter notre algorithme de résolution dynamique du problème, ou en concevoir un nouveau.

Enfin, une multitude d'expériences supplémentaires pourraient être réalisées en changeant les paramètres (consommation en suppléance, temps de récupération, densité du graphe, etc.).

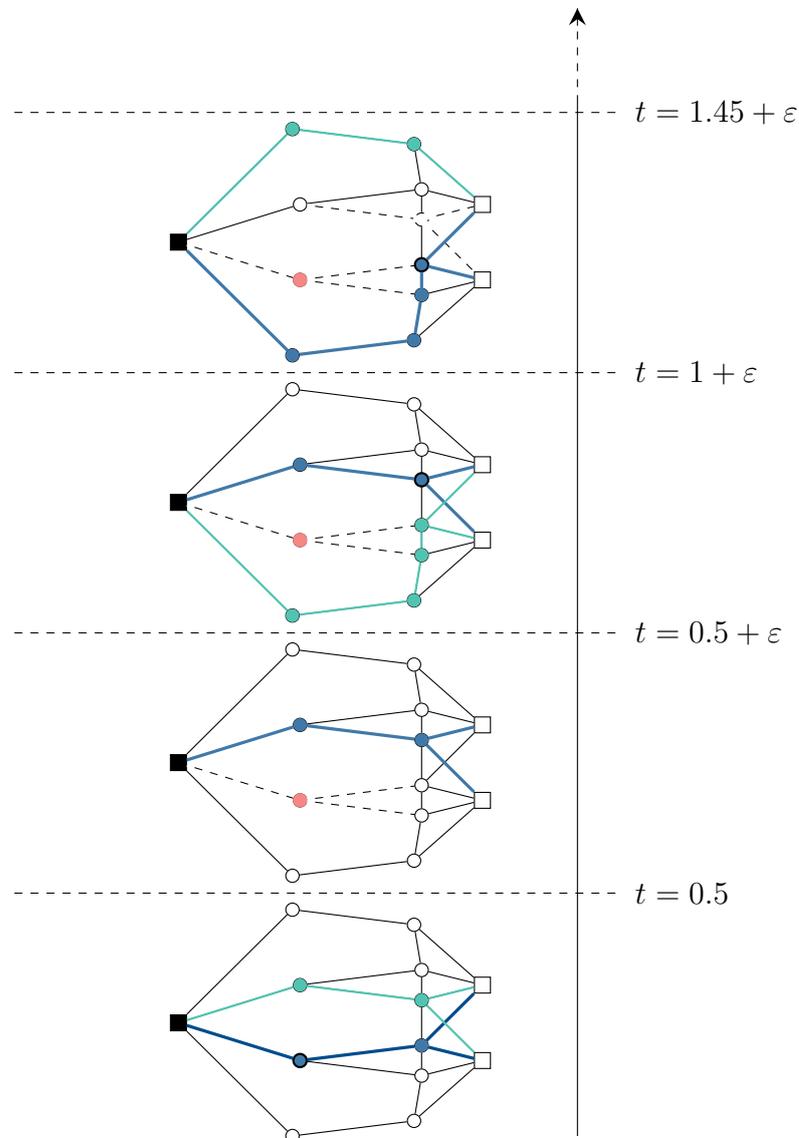


Figure 5.6 – Exemple d'utilisation de 2-couvertures avec suppléance et temps de récupération. Sur chaque représentation du graphe, les sommets bleus sont actifs et les verts sont suppléants. À $t = 0.5$ le sommet en gras subit une panne (il devient rouge par la suite), s'en suit une période de récupération d'une durée ϵ durant laquelle les sommets suppléants sont utilisés pour surveiller et relayer l'information. Une fois la récupération terminée, une nouvelle planification est calculée et mise en place. À $t = 1 + \epsilon$ une nouvelle couverture doit être utilisée car le nouveau sommet en gras a épuisé sa batterie lors des couvertures précédentes, de même à $t = 1.45 + \epsilon$.

Table 5.4 – Détail des résultats par instance

Instance	2COSUP	RUR	Écart relatif	#pannes
r_CMLP_1	2.207	2.478	10.94%	2.722
r_CMLP_2	0.821	1.63	49.63%	1.444
r_CMLP_3	1.391	1.927	27.82%	1.833
r_CMLP_4	2.13	2.595	17.92%	3.056
r_CMLP_5	1.507	1.974	23.66%	2.056
r_CMLP_6	1.912	2.292	16.58%	2.556
r_CMLP_7	4.464	5.007	10.84%	6.556
r_CMLP_8	1.204	1.642	26.67%	1.444
r_CMLP_9	1.687	1.885	10.50%	1.722
r_CMLP_10	2.614	3.418	23.52%	4
r_CMLP_11	2.302	2.627	12.37%	2.944
r_CMLP_12	2.412	2.743	12.07%	3.278
r_CMLP_13	1.426	1.944	26.65%	2.278
r_CMLP_14	2.355	3.109	24.25%	3
r_CMLP_15	3.903	4.609	15.32%	6.556
r_CMLP_16	2.649	3.32	20.21%	4.389
r_CMLP_17	3.988	5.472	27.12%	7
r_CMLP_18	3.105	3.468	10.47%	5.278
r_CMLP_19	4.067	4.726	13.94%	6.333
r_CMLP_20	2.379	2.657	10.46%	3.889
r_CMLP_41	1.431	2.055	30.36%	2.056
r_CMLP_42	1.857	2.293	19.01%	3.333
r_CMLP_43	1.249	1.828	31.67%	2.111
r_CMLP_44	2.159	2.385	9.48%	2.444
r_CMLP_45	1.444	1.935	25.37%	1.889
r_CMLP_46	2.653	3.098	14.36%	3.278
r_CMLP_47	1.761	2.075	15.13%	2.389
r_CMLP_48	2.632	3.045	13.56%	4.611
r_CMLP_49	1.656	1.908	13.21%	2.278
r_CMLP_50	2.115	2.476	14.58%	3.722
r_CMLP_51	1.656	2.284	27.50%	2.278
r_CMLP_52	2.462	2.854	13.74%	4.444
r_CMLP_53	2.511	2.831	11.30%	4.4
r_CMLP_54	2.468	2.827	12.70%	3.944
r_CMLP_55	3.463	4.011	13.66%	5.111
r_CMLP_56	3.088	3.696	16.45%	5.167
r_CMLP_57	2.497	2.824	11.58%	3.389
r_CMLP_58	2.684	3.108	13.64%	4.667
r_CMLP_59	2.506	3.118	19.63%	3.778
r_CMLP_60	2.553	3.58	28.69%	5.111

CHAPITRE 6

CONCLUSION

6.1 De la diffusion à distance 2

6.1.1 Rappel du problème

Le problème de diffusion à distance 2 consiste à planifier les émissions des noeuds d'un réseau pour faire parvenir une information détenue par une source à tous les noeuds à distance 2 (ou moins) d'elle. Nous avons étudié ce problème dans les réseaux sans-fil sans communication point à point (un noeud voulant communiquer « crie » et tous les noeuds à sa portée peuvent « l'entendre ») sous les hypothèses de communication synchrone (temps découpé en étapes) et d'interférences à la réception (un noeud ne peut pas recevoir deux messages à la même étape). On a supposé, de plus, que les noeuds à distance 2 de la source ne pouvaient pas informer d'autres noeuds à distance 2. En remarquant qu'à la première étape, le noeud détenteur de l'information peut émettre seul et informer tous les noeuds dans sa portée, le problème peut être formalisé comme il suit.

Diffusion à Distance 2

Donnée : $G = (X, Y, E)$ un graphe biparti de diffusion, $k \in \mathbb{N}$

Question : Existe-t-il k ensembles d'émetteurs $X_1, \dots, X_k \subset X$ tels que :

$$\forall y \in Y, \exists X_i : |N(y) \cap X_i| = 1 ?$$

De ce problème de diffusion à distance 2, on a étudié plusieurs aspects : la complexité du problème et les bornes sur le nombre d'étapes nécessaires pour compléter la diffusion. On a également défini une variante consistant à

maximiser le nombre de sommets prévenus en une étape :

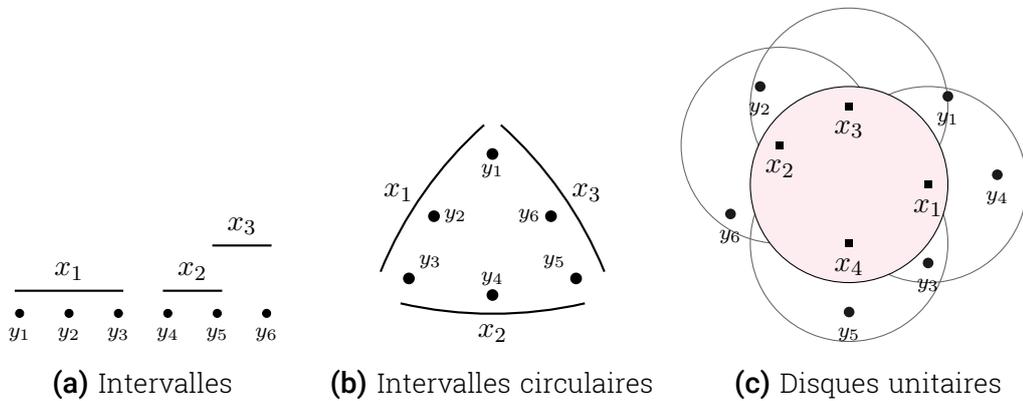
Diffusion Maximum

Donnée : $G = (X, Y, E)$ un graphe biparti de diffusion, $p \in \mathbb{N}$

Question : Existe-t-il un ensemble d'émetteurs S tel que :

$$|\{y \in Y : |N(y) \cap S = 1|\}| \geq p ?$$

Là encore, deux aspects nous intéressaient : la complexité et les bornes sur la proportion de récepteurs informés. Pour les deux problèmes, on a étudié les deux aspects dans trois cas particuliers :



6.1.2 Contributions

La table 6.1 regroupe les résultats obtenus dans chacun des cas. Les résultats n'étant pas des contributions sont grisés.

Table 6.1 – Résultats obtenus pour la diffusion à distance 2

Classe	# étapes	Min étapes	% informés	Max informés
Intervalles	2	Poly.	$\frac{2}{3}$	Poly.
Intervalles circ.	2	Poly.	$\frac{2}{3}$	Poly.
Disques unit.	$4 + 4 \log n$	Poly.?	Non-constant	Poly. [18]

6.1.3 Perspectives

À court terme On conjecture que le problème de minimiser le nombre d'étapes est polynomial dans le cas des disques unitaires. Prouver ce résultat, ou prouver qu'il est faux, concluerait l'étude des trois cas largement entamée durant cette thèse.

À plus long terme Des travaux futurs pourraient inclure d'étendre notre « classification topologique » à d'autres classes de réseaux (disques non unitaires, pseudo-disques, rectangle, etc.). On pourra aussi introduire de nouvelles variantes sur les critères à optimiser : minimiser le nombre total de noeuds devant émettre, diffusion maximum en 2, 3, k étapes, etc. Pour chacune des variantes on pourra établir des bornes et propriétés en fonction de la topologie du réseau.

6.2 De la recherche de sous-graphe connexe équilibré

6.2.1 Rappel du problème

Étant donné un graphe dont les sommets sont colorés en bleu ou rouge, on dit qu'un sous-graphe induit est **équilibré** s'il contient autant de sommets rouges que de sommets bleus. Le problème du sous-graphe connexe équilibré consiste à trouver le plus grand de ces sous-graphes équilibrés qui soit aussi connexe.

Sous-Graphe Connexe Équilibré

Donnée : $G = (V, E)$ un graphe 2-coloré, $k \in \mathbb{N}$ un entier.

Question : Existe-t-il un sous-graphe connexe de G de taille au moins $2k$ contenant autant de sommets de chaque couleur ?

Nous avons étudié ce problème du point de vue de la complexité, et présenté des résultats qui relèvent d'un travail de classification : dans quelles classes de graphe ce problème est-il difficile ou facile ?

6.2.2 Contributions

La table 6.2 regroupe les résultats déjà connus (en noir) et ceux qui ont été montrés dans cette thèse (en bleu). On rappelle que Δ désigne le degré maximum et D le diamètre.

Table 6.2 – Récapitulatif des résultats connus et des contributions

Classe de graphes	Complexité	Référence
Général	NP-complet, FPT	Bhore et collab. [6], Section 4.4
Général , $\Delta = 2$	P	Bhore et collab. [6]
Général , $D = 2$	P	Bhore et collab. [6]
Général , $D = 3$	NP-complet	Section 4.2.2
Biparti , $D = 4$	NP-complet	Section 4.2.1
Biparti , $\Delta = 4$	NP-complet, \notin APX	Bhore et collab. [6], Section 4.4
Biparti (sous)-cubique	NP-complet	Section 4.2.3

Nos résultats apportent des précisions au paysage de la complexité pour le problème. On connaît plus précisément certaines limites au-delà desquelles le problème devient difficile à résoudre, par exemple $D = 3$ et $\Delta = 3$.

6.2.3 Perspectives

À court terme Certaines zones d'ombre demeurent, comme le cas des graphes bipartis de diamètre 3 et il serait intéressant d'en déterminer la complexité pour délimiter encore plus précisément les cas où le problème devient difficile à résoudre.

À plus long terme Une des raisons qui nous a poussé à étudier le problème est la facilité avec laquelle on peut en dériver des variantes. Que se passe-t-il lorsque la coloration en entrée utilise trois couleurs? Ou quatre? Peut-on espérer que le problème soit plus simple si on ne cherche plus un sous-graphe connexe quelconque mais plutôt un chemin équilibré ou un arbre équilibré? Ou une clique? Et si le graphe en entrée est k -coloré proprement?

Toutes ces variantes pourront faire l'objet d'une étude approfondie pour enrichir la classification des problèmes au sens de la complexité. On pourrait même imaginer une approche de « méta »-classification qui irait jusqu'à donner des résultats très généraux de type : trouver le plus grand sous-graphe, vérifiant la propriété Π , équilibré dans les graphes de la famille \mathcal{F} est NP-difficile; avec Π une propriété de graphe comme « être connexe » ou « être un arbre ».

6.3 De la maximisation de durée de vie

6.3.1 Rappel du problème

Les réseaux de capteurs sans-fil consistent en une multitude de dispositifs autonomes déployés sur de larges zones pour en assurer la surveillance. Les capteurs sont munis de batteries, ainsi que de modules de surveillance et de communication à portées limitées. On s'est intéressé au problème de la planification de l'activité (surveillance, relai ou veille) de chaque capteur pour assurer la surveillance de points d'intérêts (des cibles) et la remontée d'information à une base de communication.

Dans les travaux existants, le problème de maximisation de durée de vie consiste à planifier des **couvertures** qui sont des ensembles de capteurs actifs surveillant toutes les cibles et assurant la remontée d'information à la base. Dans une couverture, chaque capteur reçoit un rôle selon qu'il soit en mode surveillance ou relai simple. En fonction du mode d'activité, la consommation du capteur diffère.

6.3.2 Contributions

Notre contribution au problème est une nouvelle approche de la tolérance aux pannes, une sorte de protocole de récupération se basant sur des structures et mécanismes que nous avons imaginé. On a ensuite simulé et comparé une gestion peu réaliste des pannes avec la notre. Les résultats sont encourageants dans la mesure où notre approche ne réduit que « peu » la durée de vie (une vingtaine de pourcents) et garantit la continuité de la surveillance.

6.3.3 Perspectives

À court terme De nombreuses expériences restent à réaliser pour achever la validation expérimentale de notre approche. On pourrait par exemple la comparer à d'autres approches réalistes n'utilisant pas les concepts de 2-couvertures, de suppléance ou de temps de récupération que nous avons introduits.

À long terme De nombreuses pistes restent à explorer. La littérature est riche en variantes pour le problème de maximisation de durée de vie, et notre approche pourrait être adaptée dans chacun de ces cas. Une variante qui serait intéressante de développer est celle de la planification cyclique lorsque les capteurs disposent de panneaux solaires qu'ils peuvent utiliser pour remonter leur niveau de batterie.

Le mot de la fin

On a donc présenté des résultats divers pour trois problèmes bien différents. Comme annoncé en introduction de cette thèse, les motivations poussant à étudier un problème conditionne bien souvent le type de résultat que l'on va rechercher et donc obtenir.

Pour la diffusion à distance 2, l'intérêt porté est surtout d'ordre théorique mais la proximité du problème avec des problématiques réelles de réseau nous a orienté vers des bornes sur des nombres d'étapes nécessaires ou des proportions de sommets possiblement informés. En effet, le nombre d'étapes pour compléter une diffusion correspond par exemple au temps nécessaire pour compléter cette diffusion; ainsi donner une borne sur le nombre d'étapes nécessaires revient à donner un délai maximum entre l'émission et la réception d'un message dans un réseau, et cela fait sens.

À l'inverse, lorsque nous avons entrepris d'étudier le problème de sous-graphe connexe équilibré, nous n'avions pas en tête d'application réelle et la simple curiosité scientifique nous a poussé à nous demander pourquoi ce problème était si difficile à résoudre. Pourtant la recherche de sous-graphe connexe équilibré est en fort lien avec le problème de recherche de motif, qui, à l'origine, répondait à une problématique « réelle ». À force de variantes et

considérations théoriques cette motivation pratique s'est trouvée amoindrie et a laissé place à la curiosité scientifique.

À l'opposé des deux précédents, le troisième problème est profondément inspiré par des problématiques réelles : optimisation de consommation de batterie, communication, surveillance. L'approche que l'on propose pourrait être intégrée à la couche application de réseaux de capteurs réels, et c'est pour cela que les méthodes de résolution exacte ont été employées.

Bien que les problèmes soient différents de part leur origine et leur formulation, les techniques utilisées pour obtenir les résultats proposés dans cette thèse ont des points communs. Par exemple, on a montré la polynomialité de certains sous-cas de la diffusion à distance 2 au moyen de réductions polynomiales vers des problèmes « faciles ». D'autres réductions polynomiales ont été utilisées pour montrer que la recherche de sous-graphe connexe équilibrée est NP-complète dans des classes de graphes particulières. Encore d'autres réductions polynomiales ont été utilisées dans la maximisation de durée de vie pour ramener la résolution du sous-problème de séparation à la résolution du problème de Steiner et donc utiliser les méthodes exactes très efficaces déjà existantes pour ce problème.

On a, au travers de ces contributions, indirectement aperçu le cercle vertueux de la classification au sens de la NP-complétude : pour classifier un problème A, on réduit un autre problème B vers A; la réduction est un algorithme polynomial qui peut enrichir l'éventail de techniques disponibles pour résoudre et réduire d'autres problèmes; une fois le problème classifié il s'ajoute aux problèmes déjà connus qui servent de point de départ pour les réductions.

TABLE DES FIGURES

1.1	$P \neq NP$ ou $P = NP$?	5
2.1	Exemple de sous-graphe induit	12
2.2	Exemple de chemins et distance.	13
2.3	Exemple de voisinages et degrés	14
2.4	Exemple de coloration	15
2.5	Représentation d'un programme linéaire et de son dual sous la forme de matrices/vecteurs.	18
3.1	Biparti de diffusion dans le cas des intervalles.	32
3.2	Instance atteignant le ratio de $\frac{2}{3}$ récepteurs informés en une étape.	34
3.3	Exemple d'application de la construction 3.1	35
3.4	Exemple d'application de la construction 3.2	37
3.5	Biparti de diffusion dans le cas des intervalles circulaires	38
3.6	Limiter le nombre d'émissions change l'optimal	39
3.7	Exemple d'application de la construction 3.3	42
3.8	Instance dans le cas des disques unitaires	45
3.9	Exemple de zones de réceptions et signatures	46
3.10	Instance critique dans le cas des disques	47
3.11	Aide à la preuve du lemme 3.7	50
3.12	Ordre sur les disques par angle à leur zone propre.	51
3.13	Aide à la preuve du lemme 3.8	53
3.14	Matrice de signature lacunaire	54
3.15	Construction d'une famille d'instance ne permettant pas de couvrir une proportion constante en une étape.	55
3.16	Aide à la preuve du lemme 3.10	56
4.1	Exemple de recherche de motif	63
4.2	Exemple de sous-graphe connexe équilibré	64
4.3	Exemple d'application de la construction 4.1	67
4.4	Exemple d'application de la construction 4.2	71

4.5	Exemple d'application de la construction 4.3	74
4.6	Utilisation des gadgets dans un SGCE	77
4.7	Exemple d'application de la construction 4.4	80
4.8	Gap-reduction de X3C à SGCE	82
4.9	Adaptation de la construction 4.4 pour les graphes bipartis degré maximum 4	84
5.1	Couvrir des surfaces ou couvrir des points?	89
5.2	Exemple d'application de la construction 5.1	95
5.3	Récupération de panne avec interruption	97
5.4	Employer des 2-couvertures est coûteux	98
5.5	Récupération du réseau en cas de panne, avec suppléance	102
5.6	Planification dynamique	108

BIBLIOGRAPHIE

- [1] Ai, J. et A. A. Abouzeid. 2006, «Coverage by directional sensors in randomly deployed wireless sensor networks», **Journal of Combinatorial Optimization**, vol. 11, n° 1, doi :10.1007/s10878-006-5975-x, p. 21--41, ISSN 1573-2886.
- [2] Alon, N., A. Bar-Noy, N. Linial et D. Peleg. 1991, «A lower bound for radio broadcast», **Journal of Computer and System Sciences**, vol. 43, n° 2, doi :10.1016/0022-0000(91)90015-W, p. 290 -- 298, ISSN 0022-0000.
- [3] Bar-Yehuda, R., O. Goldreich et A. Itai. 1992, «On the time-complexity of broadcast in multi-hop radio networks : An exponential gap between determinism and randomization», **Journal of Computer and System Sciences**, vol. 45, n° 1, doi :10.1016/0022-0000(92)90042-H, p. 104 -- 126, ISSN 0022-0000.
- [4] Bärtschi, A. et F. Grandoni. 2015, «On conflict-free multi-coloring», dans **Algorithms and Data Structures**, édité par F. Dehne, J.-R. Sack et U. Stege, Springer International Publishing, Cham, ISBN 978-3-319-21840-3, p. 103--114, doi :10.1007/978-3-319-21840-3_9.
- [5] Berman, P., G. Calinescu, C. Shah et A. Zelikovsky. 2004, «Power efficient monitoring management in sensor networks», dans **2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)**, vol. 4, ISSN 1525-3511, p. 2329--2334 Vol.4, doi :10.1109/WCNC.2004.1311452.
- [6] Bhore, S., S. Chakraborty, S. Jana, J. S. B. Mitchell, S. Pandit et S. Roy. 2018, «The balanced connected subgraph problem», **CoRR**, vol. abs/1809.08856, doi :10.1007/978-3-030-11509-8_17.
- [7] Björklund, A., T. Husfeldt, P. Kaski et M. Koivisto. 2007, «Fourier meets möbius : Fast subset convolution», dans **Proceedings of the Thirty-ninth**

- Annual ACM Symposium on Theory of Computing**, STOC '07, ACM, New York, NY, USA, ISBN 978-1-59593-631-8, p. 67--74.
- [8] Cao, X., X. Jia et G. Chen. 2010, «Maximizing lifetime of sensor surveillance systems with directional sensors», dans **2010 Sixth International Conference on Mobile Ad-hoc and Sensor Networks**, p. 110--115, doi :10.1109/MSN.2010.23.
- [9] Cardei, M., M. T. Thai, Yingshu Li et Weili Wu. 2005, «Energy-efficient target coverage in wireless sensor networks», dans **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.**, vol. 3, ISSN 0743-166X, p. 1976--1984 vol. 3, doi :10.1109/INFCOM.2005.1498475.
- [10] Carrabs, F., R. Cerulli, C. D'Ambrosio et A. Raiconi. 2015, «A hybrid exact approach for maximizing lifetime in sensor networks with complete and partial coverage constraints», **Journal of Network and Computer Applications**, vol. 58, doi :<https://doi.org/10.1016/j.jnca.2015.08.018>, p. 12 -- 22, ISSN 1084-8045.
- [11] Carrabs, F., R. Cerulli, C. D'Ambrosio et A. Raiconi. 2017, «An exact algorithm to extend lifetime through roles allocation in sensor networks with connectivity constraints», **Optimization Letters**, vol. 11, n° 7, doi :10.1007/s11590-016-1072-y, p. 1341--1356, ISSN 1862-4480.
- [12] Carrabs, F., R. Cerulli, C. D'Ambrosio et A. Raiconi. 2016, «Extending lifetime through partial coverage and roles allocation in connectivity-constrained sensor networks», **IFAC-PapersOnLine**, vol. 49, n° 12, doi :<https://doi.org/10.1016/j.ifacol.2016.07.902>, p. 973 -- 978, ISSN 2405-8963. 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016.
- [13] Castaño, F., E. Bourreau, A. Rossi, M. Sevaux et N. Velasco. 2016, «Partial target coverage to extend the lifetime in wireless multi-role sensor networks», **Networks**, vol. 68, n° 1, doi :10.1002/net.21682, p. 34--53.
- [14] Castaño, F., E. Bourreau, N. Velasco, A. Rossi et M. Sevaux. 2015, «Exact approaches for lifetime maximization in connectivity constrained wireless multi-role sensor networks», **European Journal of Operational Research**, vol. 241, n° 1, doi :<https://doi.org/10.1016/j.ejor.2014.08.013>, p. 28 -- 38, ISSN 0377-2217.

-
- [15] Castaño, F., A. Rossi, M. Sevaux et N. Velasco. 2018, «An exact approach to extend network lifetime in a general class of wireless sensor networks», **Information Sciences**, vol. 433-434, doi :<https://doi.org/10.1016/j.ins.2017.12.028>, p. 274 -- 291, ISSN 0020-0255.
- [16] Cerulli, R., R. D. Donato et A. Raiconi. 2012, «Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges», **European Journal of Operational Research**, vol. 220, n° 1, doi :<https://doi.org/10.1016/j.ejor.2012.01.046>, p. 58 -- 66, ISSN 0377-2217.
- [17] Cerulli, R., M. Gentili et A. Raiconi. 2014, «Maximizing lifetime and handling reliability in wireless sensor networks», **Networks**, vol. 64, n° 4, doi : [10.1002/net.21578](https://doi.org/10.1002/net.21578), p. 321--338.
- [18] Chan, T. et E. Grant. 2014, «Exact algorithms and apx-hardness results for geometric packing and covering problems», **Computational Geometry**, vol. 47, n° 2, doi :[10.1016/j.comgeo.2012.04.001](https://doi.org/10.1016/j.comgeo.2012.04.001), p. 112 -- 124, ISSN 0925-7721. Special Issue : 23rd Canadian Conference on Computational Geometry (CCCG11).
- [19] Cheilaris, P. et S. Smorodinsky. 2012, «Conflict-free coloring with respect to a subset of intervals», .
- [20] Cheng, M. X., L. Ruan et W. Wu. 2005, «Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks», dans **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.**, vol. 4, ISSN 0743-166X, p. 2638--2645 vol. 4, doi :[10.1109/INFCOM.2005.1498547](https://doi.org/10.1109/INFCOM.2005.1498547).
- [21] Chimani, M., P. Mutzel et B. Zey. 2012, «Improved steiner tree algorithms for bounded treewidth», **Journal of Discrete Algorithms**, vol. 16, doi : [10.1007/978-3-642-25011-8_30](https://doi.org/10.1007/978-3-642-25011-8_30), p. 67 -- 78, ISSN 1570-8667. Selected papers from the 22nd International Workshop on Combinatorial Algorithms (IWOCA 2011).
- [22] Chlamtac et Kutten. 1987, «Tree-based broadcasting in multihop radio networks», **IEEE Transactions on Computers**, vol. C-36, n° 10, doi :[10.1109/TC.1987.1676861](https://doi.org/10.1109/TC.1987.1676861), p. 1209--1223, ISSN 0018-9340.

- [23] Chlamtac, I. 1991, «The wave expansion approach to broadcasting in multihop radio networks», **IEEE Transactions on Communications**, vol. 39, n° 3, doi :10.1109/26.79285, p. 426--433, ISSN 0090-6778.
- [24] Chlamtac, I. et S. Kutten. 1985, «On broadcasting in radio networks - problem analysis and protocol design», **IEEE Transactions on Communications**, vol. 33, n° 12, doi :10.1109/TCOM.1985.1096245, p. 1240--1246, ISSN 0090-6778.
- [25] Chopra, S., E. R. Gorres et M. Rao. 1992, «Solving the steiner tree problem on a graph using branch and cut», **INFORMS Journal on Computing**, vol. 4, doi :10.1287/ijoc.4.3.320, p. 320--335.
- [26] Cogis, O., B. Darties, S. Durand, J. König et G. Simonet. 2008, «The mv-decomposition : definition and application to the distance-2 broadcast problem in multi-hops radio networks», dans **Fifth IFIP-TCS, 2008**, p. 115--126, doi :10.1007/978-0-387-09680-3_8.
- [27] Cook, S. A. 1971, «The complexity of theorem-proving procedures», dans **Proceedings of the third annual ACM symposium on Theory of computing**, ACM, p. 151--158, doi :10.1145/800157.805047.
- [28] Dantzig, G. B. 1990, «A history of scientific computing», chap. Origins of the Simplex Method, ACM, New York, NY, USA, ISBN 0-201-50814-1, p. 141--151, doi :10.1145/87252.88081.
- [29] Deschinkel, K. 2011, «A column generation based heuristic for maximum lifetime coverage in wireless sensor networks», dans **SENSORCOMM'11, 5-th Int. Conf. on Sensor Technologies and Applications**, Nice, France, p. 209--214.
- [30] Dessmark, A. et A. Pelc. 2007, «Broadcasting in geometric radio networks», **Journal of Discrete Algorithms**, vol. 5, n° 1, doi :https://doi.org/10.1016/j.jda.2006.07.001, p. 187 -- 201, ISSN 1570-8667.
- [31] Dhannya, S. M. et N. S. Narayanaswamy. 2018, «Conflict-free colouring using maximum independent set and minimum colouring», **CoRR**, vol. abs/1812.01459.

-
- [32] Dondi, R., G. Fertin et S. Vialette. 2009, «Maximum motif problem in vertex-colored graphs», dans **Combinatorial Pattern Matching**, édité par G. Kucherov et E. Ukkonen, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-02441-2, p. 221--235, doi :10.1007/978-3-642-02441-2_20.
- [33] Dreyfus, S. E. et R. A. Wagner. 1971, «The steiner problem in graphs», **Networks**, vol. 1, n° 3, doi :10.1002/net.3230010302, p. 195--207.
- [34] Even, G., Z. Lotker, D. Ron et S. Smorodinsky. 2002, «Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks», dans **The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.**, ISSN 0272-5428, p. 691--700, doi :10.1109/SFCS.2002.1181994.
- [35] Fellows, M. R., G. Fertin, D. Hermelin et S. Vialette. 2007, «Sharp tractability borderlines for finding connected motifs in vertex-colored graphs», dans **Automata, Languages and Programming**, édité par L. Arge, C. Cachin, T. Jurdziński et A. Tarlecki, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-73420-8, p. 340--351, doi :10.1007/978-3-540-73420-8_31.
- [36] Fellows, M. R., G. Fertin, D. Hermelin et S. Vialette. 2011, «Upper and lower bounds for finding connected motifs in vertex-colored graphs», **Journal of Computer and System Sciences**, vol. 77, n° 4, doi :https://doi.org/10.1016/j.jcss.2010.07.003, p. 799 -- 811, ISSN 0022-0000. JCSS IEEE AINA 2009.
- [37] Ford, L. et D. R. Fulkerson. 1956, «Maximal flow through a network», **Canadian Journal of Mathematics**, vol. 8, doi :10.1007/978-0-8176-4842-8_15, p. 399--404.
- [38] Fuchs, B., W. Kern, D. Molle, S. Richter, P. Rossmanith et X. Wang. 2007, «Dynamic programming for minimum steiner trees», **Theory of Computing Systems**, vol. 41, n° 3, doi :10.1007/s00224-007-1324-4, p. 493--500, ISSN 1433-0490.
- [39] Fusco, G. et H. Gupta. 2009, «Selection and orientation of directional sensors for coverage maximization», dans **2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks**, ISSN 2155-5486, p. 1--9, doi :10.1109/SAHCN.2009.5168968.

- [40] Garey, M. R. et D. S. Johnson. 1979, **Computers and Intractability : A Guide to the Theory of NP-Completeness**, W. H. Freeman, ISBN 0-7167-1044-7.
- [41] Garey, M. R. et D. S. Johnson. 1990, **Computers and Intractability; A Guide to the Theory of NP-Completeness**, W. H. Freeman & Co., New York, NY, USA, ISBN 0716710455.
- [42] Gentili, M. et A. Raiconi. 2013, « α -coverage to extend network lifetime on wireless sensor networks», **Optimization Letters**, vol. 7, n° 1, doi : 10.1007/s11590-011-0405-0, p. 157--172, ISSN 1862-4480.
- [43] Gonzalez, T. F. 1985, «Clustering to minimize the maximum intercluster distance», **Theoretical Computer Science**, vol. 38, doi :10.1016/0304-3975(85)90224-5, p. 293--306.
- [44] Grötschel, M., L. Lovász et A. Schrijver. 1981, «The ellipsoid method and its consequences in combinatorial optimization», **Combinatorica**, vol. 1, n° 2, doi :10.1007/BF02579273, p. 169--197.
- [45] Gu, Y., Y. Ji et B. Zhao. 2009, «Maximize lifetime of heterogeneous wireless sensor networks with joint coverage and connectivity requirement», dans **2009 International Conference on Scalable Computing and Communications; Eighth International Conference on Embedded Computing**, p. 226--231, doi :10.1109/EmbeddedCom-ScalCom.2009.48.
- [46] Gupta, H., S. R. Das et Q. Gu. 2003, «Connected sensor cover : self-organization of sensor networks for efficient query execution», dans **Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing**, ACM, p. 189--200, doi :10.1145/778415.778438.
- [47] Har-Peled, S. et S. Smorodinsky. 2005, «Conflict-free coloring of points and simple regions in the plane», **Discrete & Computational Geometry**, vol. 34, n° 1, doi :10.1007/s00454-005-1162-6, p. 47--70, ISSN 1432-0444.
- [48] Johnson, D. S. 1982, «The np-completeness column:an ongoing guide», **J. Algorithms**, vol. 3, p. 182--195.
- [49] Johnson, D. S. 1985, «The np-completeness column : An ongoing guide», **Journal of Algorithms**, vol. 6, n° 1, doi :10.1016/0196-6774(87)90043-5, p. 145 -- 159, ISSN 0196-6774.

-
- [50] Karp, R. M. 1972, «Reducibility among combinatorial problems», dans **Proceedings of a symposium on the Complexity of Computer Computations, IBM Thomas J. Watson Research Center**, p. 85--103, doi : 10.1007/978-3-540-68279-0_8.
- [51] Katz, M. J., N. Lev-Tov et G. Morgenstern. 2012, «Conflict-free coloring of points on a line with respect to a set of intervals», **Computational Geometry**, vol. 45, n° 9, doi :<https://doi.org/10.1016/j.comgeo.2012.01.013>, p. 508 -- 514, ISSN 0925-7721. The 19th Canadian Conference on Computational Geometry (CCCG2007) held in Ottawa, Canada on August 20-22, 2007.
- [52] Kushilevitz, E. et Y. Mansour. 1998, «A lower bound for broadcast in radio networks», **SIAM Journal on Computing**, vol. 27, n° 3, doi :10.1137/S0097539794279109, p. 702--712.
- [53] Lacroix, V., C. G. Fernandes et M.-F. Sagot. 2006, «Motif search in graphs : Application to metabolic networks», **IEEE/ACM Trans. Comput. Biol. Bioinformatics**, vol. 3, n° 4, doi :10.1109/TCBB.2006.55, p. 360--368, ISSN 1545-5963.
- [54] Ladner, R. E. 1975, «On the structure of polynomial time reducibility», **J. ACM**, vol. 22, n° 1, doi :10.1145/321864.321877, p. 155--171, ISSN 0004-5411.
- [55] Lawler, E. L. et D. E. Wood. 1966, «Branch-and-bound methods : A survey», **Operations Research**, vol. 14, n° 4, doi :10.1287/opre.14.4.699, p. 699--719.
- [56] Lokshтанov, D., N. Misra, G. Philip, M. S. Ramanujan et S. Saurabh. 2013, «Hardness of r -dominating set on graphs of diameter $(r + 1)$ », dans **Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers**, p. 255--267, doi :10.1007/978-3-319-03898-8_22.
- [57] Mohamed, R. E., A. I. Saleh, M. Abdelrazzak et A. S. Samra. 2018, «Survey on wireless sensor network applications and energy efficient routing protocols», **Wireless Personal Communications**, vol. 101, n° 2, doi :10.1007/s11277-018-5747-9, p. 1019--1055, ISSN 1572-834X.
- [58] Niedermeier, R. 2002, **Invitation to fixed-parameter algorithms**, Citeseer, doi :10.1093/acprof:oso/9780198566076.001.0001.

- [59] Raiconi, A. et M. Gentili. 2011, «Exact and metaheuristic approaches to extend lifetime and maintain connectivity in wireless sensors networks», dans **Network Optimization**, édité par J. Pahl, T. Reiners et S. Voß, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-21527-8, p. 607--619, doi :10.1007/978-3-642-21527-8_68.
- [60] Rossi, A., A. Singh et M. Sevaux. 2012, «Column generation algorithm for sensor coverage scheduling under bandwidth constraints», **Networks**, vol. 60, n° 3, doi :10.1002/net.20466, p. 141--154.
- [61] Siek, J., A. Lumsdaine et L.-Q. Lee. 2002, **The boost graph library : user guide and reference manual**, Addison-Wesley.
- [62] Singh, A., A. Rossi et M. Sevaux. 2013, «Matheuristic approaches for q-coverage problem versions in wireless sensor networks», **Engineering Optimization**, vol. 45, n° 5, doi :10.1080/0305215X.2012.687732, p. 609--626.
- [63] Slijepcevic, S. et M. Potkonjak. 2001, «Power efficient organization of wireless sensor networks», dans **ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No.01CH37240)**, vol. 2, p. 472--476 vol.2, doi :10.1109/ICC.2001.936985.
- [64] Smorodinsky, S. 2013, **Conflict-Free Coloring and its Applications**, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-41498-5, p. 331--389, doi :10.1007/978-3-642-41498-5_12.
- [65] Thai, M. T., F. Wang, D. H. Du et X. Jia. 2008, «Coverage problems in wireless sensor networks; designs and analysis», **Int. J. Sen. Netw.**, vol. 3, n° 3, doi :10.1504/IJSNET.2008.018485, p. 191--200, ISSN 1748-1279.
- [66] Wang, C., M. T. Thai, Y. Li, F. Wang et W. Wu. 2007, «Minimum coverage breach and maximum network lifetime in wireless sensor networks», dans **IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference**, ISSN 1930-529X, p. 1118--1123, doi :10.1109/GLOCOM.2007.215.
- [67] Zhao, Q. et M. Gurusamy. 2008, «Lifetime maximization for connected target coverage in wireless sensor networks», **IEEE/ACM Transactions on Networking**, vol. 16, n° 6, doi :10.1109/TNET.2007.911432, p. 1378--1391, ISSN 1063-6692.

-
- [68] Zongheng Zhou, S. Das et H. Gupta. 2004, «Connected k-coverage problem in sensor networks», dans **Proceedings. 13th International Conference on Computer Communications and Networks (IEEE Cat. No.04EX969)**, ISSN 1095-2055, p. 373--378, doi :10.1109/ICCCN.2004.1401672.
- [69] Zorbas, D. et C. Douligeris. 2010, «Power efficient target coverage in wireless sensor networks», dans **Sustainable Wireless Sensor Networks**, IntechOpen, doi :10.5772/13377.