N° d'ordre NNT : xxx

THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
l'Université Claude Bernard Lyon 1

École Doctorale 512
InfoMaths

Spécialité de doctorat : Informatique

Soutenue publiquement le 25 novembre 2020, par :
Alice Joffard

Graph domination and reconfiguration problems

Devant le jury composé de :

M. Liedloff Mathieu	Maître de conférences	Université d'Orléans	Rapporteur
M. Togni Olivier	Professeur des Universités	Université de Bourgogne	Rapporteur
Mme. Bonifati Angela	Professeure des Universités	Université Lyon 1	Examinatrice
Mme. Brauner Nadia	Professeure des Universités	Université Grenoble Alpes	Examinatrice
Mme. Meeks Kitty	Chercheure	University of Glasgow	Examinatrice
Mme. Nishimura Naomi	Professeure associée	University of Waterloo	Examinatrice
M. Kheddouci Hamamache	Professeur des Universités	Université Lyon 1	Directeur
M. Bousquet Nicolas	Chargé de Recherche CNRS	CNRS Lyon	Co-directeur

UNIVERSITÉ CLAUDE BERNARD - LYON 1

Président de l'Université	M. Frédéric FLEURY
Président du Conseil Académique	M. Hamda BEN HADID
Vice-président du Conseil d'Administration	M. Didier REVEL
Vice-président du Conseil des Etudes et de la Vie Universitaire	M. Philippe CHEVALIER
Vice-président de la Commission de Recherche	
Directrice Générale des Services	M. Damien VERHAEGHE

COMPOSANTES SANTÉ

Faculté de Médecine Lyon-Est - Claude Bernard	Doyen : M. Gilles RODE
Faculté de Médecine et Maïeutique Lyon Sud Charles. Mérieux	Doyenne : Mme Carole BURILLON
UFR d'Odontologie	Doyenne : Mme Dominique SEUX
Institut des Sciences Pharmaceutiques et Biologiques	Directrice : Mme Christine VINCIGUERRA
Institut des Sciences et Techniques de la Réadaptation	Directeur : M. Xavier PERROT
Département de Formation et Centre de Recherche en Biologie Humaine	Directrice : Mme Anne-Marie SCHOTT

COMPOSANTES ET DEPARTEMENTS DE SCIENCES ET TECHNOLOGIE

UFR Biosciences	Directrice : Mme Kathrin GIESELER
Département Génie Electrique et des Procédés (GEP)	Directrice : Mme Rosaria FERRIGNO
Département Informatique	Directeur : M. Behzad SHARIAT
Département Mécanique	Directeur : M. Marc BUFFAT
UFR - Faculté des Sciences	Administrateur provisoire : M. Bruno ANDRIOLETTI
UFR (STAPS)	Directeur : M. Yannick VANPOULLE
Observatoire de Lyon	Directeur : Mme Isabelle DANIEL
Ecole Polytechnique Universitaire Lyon 1	Directeur : M. Emmanuel PERRIN
Ecole Supérieure de Chimie, Physique, Electronique (CPE Lyon)	Directeur : M. Gérard PIGNAULT
Institut Universitaire de Technologie de Lyon 1	Directeur : M. Christophe VITON
Institut de Science Financière et d'Assurances	Directeur : M. Nicolas LEBOISNE
ESPE	Administrateur provisoire : M. Pierre CHAREYRON

Á Bertrand, merci d'avoir fait naître une vocation

Remerciements

Je voudrais remercier tout d'abord Mathieu Liedloff et Olivier Togni pour la relecture de ma thèse et pour les retours. Je remercie également Angela Bonifati, Nadia Brauner, Kitty Meeks et Naomi Nishimura pour avoir accepté de participer à mon jury de thèse dans ces conditions particulières.

Je voudrais ensuite remercier mes encadrants de thèse, Hamamache Kheddouci et Nicolas Bousquet. Hamamache, merci de m'avoir donné l'opportunité de réaliser cette thèse et d'avoir toujours fait en sorte qu'elle se déroule au mieux. Nicolas, lorsque tu as rejoint mon encadrement, j'avais peur de ne pas me sentir à la hauteur. Mais tu fais partie de ces rares personnes qui savent être grandes sans donner l'impression aux autres qu'ils sont petits. Tu as été encadrant, professeur, coach et psychologue, et j'ai fini cette thèse, en ayant énormément appris, en ayant pris confiance en moi, et en restant saine d'esprit (presque). Merci pour tout.

J'ai eu l'occasion au cours de ma thèse de collaborer avec de nombreux chercheurs qui m'ont également beaucoup appris. Je remercie pour cela Guillaume Bagan, avec qui il était très intéressant de travailler et de discuter, plus généralement. Je remercie également Paul Ouvrard, j'ai énormément apprécié notre collaboration. Je remercie enfin l'équipe GAPCOMB de Barcelone, auprès de laquelle j'ai effectué un séjour enrichissant. En particulier, je remercie Oriol Serra et Guillem Perarnau avec qui j'ai beaucoup échangé. Je remercie plus généralement toutes les personnes avec qui j'ai travaillé lors des nombreux workshops auxquels j'ai eu la chance de participer.

J'ai été très bien entourée durant ma thèse, en partie grâce à l'équipe GOAL. Merci à toute l'équipe pour ces trois années, et merci pour la répétition. En particulier, je remercie Aline Parreau et Eric Duchene, de m'avoir toujours encouragée, de m'avoir intégrée au projet maths à modeler, et d'avoir été des compétiteurs hors pair aux mots croisés.

Je remercie plus généralement le LIRIS et tous mes collègues. Merci Saïd de faire de ce laboratoire un lieu aussi agréable. Merci Isabelle, Brigitte, Sandra, Catherine et Jean-Pierre, d'avoir toujours fait preuve de patience et de bienveillance face à ma phobie administrative. Merci à tous les doctorants de Splitted Solitude et à Remi, je ne me suis pas une seule fois sentie seule en trois ans, même confinée. Maxime, merci, signé Alice. Merci d'être aussi attentif aux autres même à l'autre bout de la planète, se demander si je vais bien parce que j'ai créé une playlist avec une chanson qui s'appelle "la vie ne vaut rien" dedans, c'est quand même un niveau au dessus. Arthur, merci pour les crêpes, pour les parties de scrabble, et pour avoir fait de notre appartement un lieu aussi cool à vivre (je pense que l'un comme l'autre on a eu le loisir de se rendre compte que c'était pas toujours aussi sympa les colocations !). Antoine, merci de ton soutien, tant sur la thèse que sur l'administratif, et merci aussi à Hilbert d'être le deuxième meilleur chat du monde. Simon, merci d'être un asocial aussi gentil. Charles, merci pour tous les pictionary et merci d'être toujours souriant. Thibault, merci d'expliquer tes blagues sinon je les comprends pas... Et merci d'avoir été mon psy et mon coach amoureux. Par contre désolée mais c'est pas fini. Sam, merci d'avoir partagé des bons moments dans ta vie optimisée. Joce, merci pour les bons bouillons thaï, les bonnes bouteilles, et merci d'accueillir des débats politiques sans queue ni tête sans broncher. Alexis, merci de m'avoir fait découvrir des musiques très différentes de ce que je connaissais déjà (vive la K-pop). Agathe, merci pour tous ces random facts sur Pink Floyd, par contre fais semblant d'être moins forte au trivial stp. Julia, merci d'être nulle avec moi au trivial, et merci d'être forte pour réconforter, tu es au top. Remi, merci d'être un adversaire redoutable au blind test, mais de ne m'avoir jamais battue, et merci pour les cours de yoga, c'était fort relaxant. Bea, thank you for the delicious food and the beautiful plants. Valentin, merci de m'avoir donné une vie entière de vidéos et séries à regarder, et merci vraiment pour toute l'aide que tu m'as apportée. Anna, merci pour les vidéos de chats, plus efficaces que le xanax. Matthieu, merci de me faire autant rire, et de m'avoir offert une parenthèse pendant la rédaction. Julia, merci pour cette même parenthèse, et merci pour toutes les fois où je me suis confiée à toi. Marc, pas merci d'avoir placé la barre aussi haut pour ta petite soeur de thèse, mais merci d'être toujours resté humble et agréable. Remy, merci de m'apprendre une centaine de nouvelles informations chaque fois qu'on mange ensemble. Wubu, murcu pur tu lus upru mudu ju (mu tu rugl du rummukub ul su nul).

Je remercie aussi le G-SCOP pour ce début d'année agréable, j'espère que l'on pourra passer d'autres bons moments ensemble prochainement.

Je remercie également tous mes autres amis, qui ont été présents dans les bons moments comme dans la période de rédaction. Camille, merci de m'avoir écoutée me plaindre, et merci de m'avoir accueillie à la fin, merci à Mathieu aussi. Laura, merci pour les conseils et le shitasu. Fanny, merci pour les musiques qui m'ont presque donné l'impression que je pouvais sortir. Nina, merci d'avoir su être un soutien quand tu en avais toi-même besoin, et merci d'avoir vidé mes poubelles (vraiment !). Aurélie, merci pour les bons moments, quand tu veux on passe à la vitesse supérieure. Clara, merci pour cette très bonne soirée, ça fait du bien de te retrouver. Amélie, merci d'avoir choisi les couleurs du Rubik's cube, tu es quasiment autrice de cette thèse. Merci, et merci à Ludo aussi, de m'avoir accueillie quand j'étais déjà dans la folie profonde. Marie, merci d'avoir continué à papoter comme si on était en face à face. Léa, merci d'être toujours rayonnante et merci pour les tongs, ça a été la seule chose attachée à mes pieds pendant les mois de rédaction. Merci aussi à tout Senzala et en particulier à Nuno, pour le parfait remède contre le stress. Iris, merci beaucoup pour ton aide, j'aurais pas pu gérer tout en même temps. Colas, merci des petites parenthèses qui m'ont fait énormément de bien. Et merci à tous mes colocataires à Barcelone, qui m'ont toujours aidée à rester sérieuse. En particulier, merci à Antoine, à Stefanito, et à Coralie, avec qui j'ai pu fêter la fin de la rédaction en beauté (un peu en avance, mais chut). Merci aussi à Alexis, de m'avoir poussée à continuer malgré les doutes.

Je n'aurais probablement jamais été aussi loin sans les encouragements de ma famille. Merci à tous les cousins, oncles et tantes, pour les messages, photos et vidéos pendant le confinement. Papé, Mamé, merci d'être toujours fiers de moi et merci pour les appels qui réchauffent le coeur. Mitié, merci pour cet été et tous les autres, j'espère que tu es heureuse que j'aie choisi la même voie que ton idole Cédric. Nina, merci de m'avoir attachée à une chaise pour que j'apprenne mes tables de multiplication, c'est probablement ce qui a déclenché ma passion. Merci, et merci à Loïc aussi, d'avoir fait de cette fin de rédaction un aussi bon moment. On se retrouve pour un autre weekend spa bientôt ! Merci à ma Nana d'avoir été l'actrice principale de tant de vidéos qui m'ont remonté le moral. Et merci enfin à ceux qui étaient en première ligne et sans qui je crois que je n'aurais pas survécu. Maman, merci déjà pour toutes les boîtes à bijoux et les courses au marché, et merci aussi pour toutes les vraies courses, les vrais repas équilibrés, et tout le soutien moral. Papa, merci de t'être assuré que les repas étaient équilibrés, et merci aussi pour tous les déménagements, tous les conseils, et tout le soutien moral.

Résumé

Cette thèse a pour objet l'étude de la domination de graphes et des problèmes de reconfiguration.

Un ensemble dominant d'un graphe est un sous-ensemble de sommets tel que tous les sommets du graphe sont ou bien dans l'ensemble, ou bien sont voisins d'au moins un sommet dans l'ensemble. Quant aux problèmes de reconfiguration, ils consistent, étant donné un problème source, à étudier si il est possible, et comment, de passer d'une solution de ce problème source à une autre en effectuant une séquence de changements élémentaires suivant une règle donnée qui maintiennent une solution.

Le premier chapitre est une introduction générale, dans laquelle nous présentons de manière informelle les notions principales de cette thèse, et en donnons l'organisation.

Le deuxième chapitre est un chapitre préliminaire, qui a pour but de fournir au lecteur les notions élémentaires de théorie de la complexité, de théorie des graphes, et de domination, dont nous aurons besoin au cours des chapitres suivants.

Le troisième chapitre est entièrement consacré aux problèmes de reconfiguration. Nous donnons les définitions essentielles à la compréhension des problèmes de reconfiguration exposés dans les chapitres suivants. Ces définitions sont illustrées par divers problèmes : le jeu du taquin et ses généralisations, la reconfiguration d'ensembles indépendants et d'ensembles dominants dans un graphe, la reconfiguration de colorations d'un graphe, la reconfiguration du problème de satisfiabilité, ainsi que la reconfiguration de multigraphes connexes ayant la même séquence de degrés.

Le quatrième chapitre se focalise sur ce dernier problème. Nous donnons un algorithme d'approximation polynomial qui renvoit une séquence de reconfiguration entre deux multigraphes connexes ayant la même séquence de degrés, de longueur au plus 2.5 fois la longueur minimale. L'opération élémentaire consiste à échanger une extremité entre deux arêtes disjointes. Le meilleur ratio obtenu jusqu'ici était de 4, et cette amélioration est due à la découverte d'une nouvelle borne supérieure. En revanche, nous gardons la même borne inférieure, et nous montrons que tant qu'une meilleure borne inférieure ne sera pas trouvée, le ratio d'approximation ne pourra pas être drastiquement meilleur que 2.5.

Le cinquième chapitre se recentre sur les problèmes de domination. Nous étudions la connexité et le diamètre du graphe de reconfiguration, lorsque l'opération élémentaire consiste en l'addition ou la suppression d'un sommet de l'ensemble dominant. En d'autres termes, nous cherchons des conditions suffisantes pour qu'il existe toujours une séquence de reconfiguration entre deux ensembles dominants, et donnons une borne supérieure sur la longueur de cette séquence. En particulier, nous nous intéressons à la taille maximale des ensembles dominants autorisés dans la séquence. Nous montrons qu'au delà d'une valeur dépendant du nombre d'indépendance du graphe, le graphe de reconfiguration a un diamètre linéaire. Nous donnons également une autre borne supérieure, qui dépend cette fois ci de la largeur arborescente du graphe, à partir de laquelle le graphe de reconfiguration est connexe et a un diamètre linéaire. Nous donnons également deux autres bornes supérieures pour les graphes planaires, ainsi que les graphes dont le graphe complet K_ℓ n'est pas un mineur.

Dans le sixième chapitre, nous changeons l'opération élémentaire, qui consiste alors en un glissement de jeton le long d'une arête (un sommet est retiré de l'ensemble, et un des ses voisins y est ajouté). Nous étudions la complexité du problème d'atteignabilité, qui consiste à déterminer si il existe une séquence de reconfiguration entre deux ensembles dominants donnés. Nous montrons que le problème est PSPACE-complet pour les graphes planaires bipartis, pour les graphes de disques unité, les circle graphs, et les line graphs. Nous donnons également un algorithme polynomial pour les circular arc graphs.

Le septième chapitre est consacré au problème de domination éternelle. Dans ce problème, deux joueurs s'affrontent sur un graphe : l'attaquant, et le défenseur. Le défenseur commence par choisir un ensemble de sommets, où il place des gardes. Lors de chaque tour, l'attaquant choisit un sommet inoccupé, et le défenseur doit déplacer un de ses gardes sur le sommet attaqué, en le faisant glisser le long d'une arête (un garde doit donc être sur un voisin du sommet attaqué avant le déplacement). Dans une première version, les autres gardes doivent rester immobiles, et dans la deuxième, ils peuvent tous ou bien rester immobile, ou bien glisser le long d'une arête. Le nombre de domination éternelle du graphe correspond alors au nombre minimum de gardes nécessaires pour pouvoir défendre n'importe quelle séquence infinie d'attaques. Nous introduisons une version du problème sur les graphes dirigés, où les gardes doivent suivre la direction des arcs. Nous donnons des bornes sur la valeur des deux paramètres (correspondant aux deux versions), qui généralisent des résultats sur les graphes non dirigés. Nous introduisons ensuite un nouveau problème qui consiste à chercher l'orientation d'un graphe qui minimise ces deux paramètres. Nous montrons que dans la première version, déterminer si le paramètre correspondant est au plus un k donné est un problème co-NP-difficile. Nous étudions également la valeur des deux paramètres sur différentes classes de graphes comme les cycles, les arbres, les graphes complets et complets bipartis, et différents types de grilles. Nous caractérisons enfin les graphes dont la valeur du deuxième paramètre est 2.

Enfin, le huitième chapitre est une conclusion générale, qui rappelle les travaux effectués ainsi que les problèmes ouverts soulevés au cours des chapitres et qui semblent particulièrement intéressants à étudier.

Mots clés : Théorie des graphes, reconfiguration, domination, ensembles dominants, séquence de degrés, domination éternelle.

Abstract

This object of this thesis is to study graph domination and reconfiguration problems.

A dominating set of a graph is a subset of vertices such that every vertex of the graph either is in the set, or is a neighbor of at least one vertex in the set. As for reconfiguration problems, they consist in, given a source problem, determining if it is possible, and how, to go from a solution of this source problem to another by performing a sequence of elementary changes following a given rule and that maintain a solution all along.

The first chapter is a general introduction, in which we present in an informal way the principal notions of this thesis, and give its organization.

The second chapter is a preliminary chapter, whose goal is to give the reader some basic notions of complexity theory, graph theory, and domination, which we will need in the following chapters.

The third chapter is fully devoted to reconfiguration problems. We give the definitions that are essential to understand the reconfiguration problems exposed in the following chapters. These definitions are illustrated with several problems: the 15-puzzle game and its generalizations, the reconfiguration of independent sets and dominating sets in a graph, the reconfiguration of graph colorings, the reconfiguration of the satisfiability problem, as well as the reconfiguration of connected multigraphs with the same degree sequence.

The fourth chapter focuses on this last problem. We give a polynomial approximation algorithm that outputs a reconfiguration sequence between two connected multigraphs that have the same degree sequence, of length at most 2.5 times the minimum length. The elementary operation consists in exchanging an extremity between two disjoint edges. The best ratio known so far was 4, and this improvement is due to the discover of a new upper bound. However, we keep the same lower bound, and we show that as long as a better lower bound is not found, the approximation ratio can not be drastically better than 2.5.

The fifth chapter goes back to domination problems. We study the connectivity and diameter of the reconfiguration graph, when the elementary operation consists in the addition or removal of a vertex of the set. In other words, we search for sufficient conditions that guarantee the existence of a reconfiguration sequence between two dominating sets, and give an upper bound on the length of this sequence. In particular, we are interested in the maximum size of the dominating sets authorized in the sequence. We show that above a certain value, that depends on the independence number of the graph, the reconfiguration graph has linear diameter. We also give another upper bound, that depends on the treewidth of the graph, above which the reconfiguration graph is connected and has linear diameter. We also give two other upper bounds for planar graphs and for K_ℓ-minor-free graphs.

In the sixth chapter, we change the elementary operation, that then consists in sliding a token along an edge (a vertex is removed from the set, and one of its neighbors is added). We study the complexity of the reachability problem, which consists in determining if there exists a reconfiguration sequence between two given dominating sets. We show that the problem is PSPACE-complete in planar bipartite graphs, unit disk graphs, circle graphs and line graphs. We also give a polynomial algorithm for circular arc graphs.

The seventh chapter is devoted to the eternal domination problem. In this problem, two players are playing on a graph: the attacker, and the defender. The defender starts by choosing a set of vertices, where they place some guards. At each turn, the attacker chooses an unoccupied vertex, and the defender must move one of their guards to the attacked vertex, by sliding it along an edge (a guard must therefore be placed on a neighbor of the attacked vertex). In a first version, the other guards must stay on their position, and in a second version, they can all either stay, or slide along an edge. The eternal domination number corresponds to the minimum number of guards necessary to defend any infinite sequence of attacks. We introduce a version of the problem on directed graphs, where the guards must follow the direction of the arcs. We give some bounds on the value of both parameters (corresponding to the two versions), that generalize results on undirected graphs. We then introduce a new problem that consists in searching for the orientation of a graph that minimizes the two parameters. We show that in the first version, determining if the parameter is at most a given k is a co-NP-hard problem. We also study the value of both parameters in several graph classes such as cycles, trees, complete and complete bipartite graphs, and different types of grids. We finally characterize the graphs for which the value of the second parameter is 2.

Finally, the eighth chapter is a general conclusion, that goes back on the presented work and the open problems raised along the previous chapter and that seem particularly interesting to study.

Key words: graph theory, reconfiguration, domination, dominating sets, degree sequence, eternal domination.

Contents

Chapter 1

Introduction

My first encounter with graph theory and combinatorics happened very early. I got used to bothering my mother for her to give me math problems to solve during our hikes, but my father took it to another level by giving me a riddle called the *House of Santa Claus*. He asked me if I could draw a house, with a cross in it as illustrated in Figure 1.1 without raising the pen. He eventually showed me how to do it, but it did not satisfy my curiosity, as I started wondering what could and could not be drawn without raising the pen. A few years later, as I was cheating in Rubik's cube by unsticking the colored stickers, I wondered if I coud "break" it, i.e. make it impossible to solve by mixing up the colors. In school, when the teacher asked the class how many diagonals a pentagon had, I wondered how the number of diagonals depended on the number of vertices. I actually wrote a paper on it in the evening and gave it to my teacher the next day, who had to inform me that, unfortunately, this result was already proven. When I grew up, I had the opportunity to pursue scientific studies, and I was formally introduced to graph theory. All these questions and so many others were answered, but even more were raised. This has been particularly true during my thesis, where I have had the chance to encounter a very large range of problems. In this manuscript, I chose to focus on the reconfiguration problems and graph domination, which both represent an important part of the work completed during this thesis.

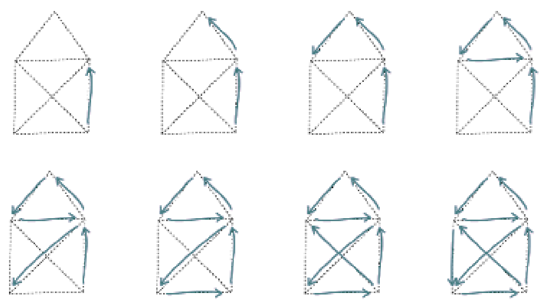


Figure 1.1: The *House of Santa Claus* riddle.
Source: http://planund.com/

Graphs and domination

The reason why I came across graph problems so often in my childhood is simply because graphs are everywhere. They allow to model a very wide range of problems, in various fields such as social sciences, biology, logistics, chemistry or physics. Informally speaking, a graph is a set of points called vertices, and a set of edges that link some pairs of vertices. Typically, in a social network, the vertices can represent the persons, and two persons can share an edge if they know each other. In logistics, a vertex can represent a task to complete, and a task can be linked to another when it needs to be done before the other. Note that this relationship is not symmetrical, and one can add a direction to the edge, from the task that needs to be completed first to the other. The graph then becomes a directed graph, or digraph, and the edges are called arcs. In chemistry, the vertices can represent the atoms of a molecule, and the edges the chemical bonds between them. Note that two atoms can share multiple bonds, and in this case we model the molecule with a multigraph, where two vertices can share multiple edges. This is equivalent to associating an natural number to each edge, and more generally, we can associate any type of data (labels, real numbers, integers...) to the vertices or the edges of the graph. However, in this thesis, unless specified, we work with undirected and simple graphs (i.e. not multigraphs), where no data is attached to the vertices nor the edges. An example of a graph is given in Figure 1.2.

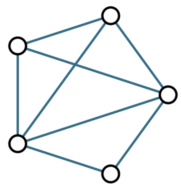


Figure 1.2: A simple undirected graph

We use graphs to model real data and to answer problems on this data. In social sciences, the question can for example be: what is the largest group of people that all know each other ? In logistics, what is the minimum amount of time needed to complete all the tasks ? In chemistry, how many different molecules have the same given formula ? Graphs allow to transform these problems into purely mathematical problems. For instance, the largest group of people that all know each other constitute what we call a clique in the graph, and finding the largest clique is a very commonly studied problem. The minimum amount of time needed to complete a set of tasks can be modeled through a graph problem called coloring. And the number of different molecules that have the same formula corresponds to the number of non isomorphic graphs (i.e. graphs that are not identical up to a permutation of the vertices) that have the same degree sequence (the degree of a vertex is the number of other vertices it is linked to). As for the problems mentionned earlier, the *House of Santa Claus* riddle is a particular case of a famous graph problem called the Eulerian cycle problem, and the great mathematician Leonhard Euler actually showed that it is possible to draw a graph without raising the pen if and only if every vertex has even degree. And the number of diagonals in a polygon on n vertices is given by the number of edges in the complete graph (i.e. the graph that contains all the possible edges) on n vertices, which is $\frac{n(n-1)}{2}$, minus the number of edges in a cycle on n vertices, which is n.

Graph theory is on the border between mathematics and theoretical computer science. In particular, a whole area of graph theory focuses on finding algorithms to solve graph problems. For instance, the problem of finding the largest clique in a graph has been extensively studied from the algorithmic point of view. An algorithm is a sequence of instructions that solves a problem. It can then be implemented in a computer, and used in any application of the problem. But for the algorithm to be useful, we need it to be efficient. The complexity of an algorithm is, informally speaking, the time or the space it takes to run on a computer, and the complexity of a problem is the minimum complexity of an algorithm that solves it. Studying the complexity of a problem is a useful tool when searching for efficient algorithms. In particular, when the complexity is too high, we might want to focus on the search of efficient approximation algorithms, which only give an approximation of the expected result, but whose complexity is lower. For these reasons, the complexity of graph problems is a very interesting and widely investigated problem.

In this thesis, we mostly study a graph problem called domination. A dominating set of a graph is a subset of its vertices, such that each vertex of the graph is either in the set, or shares an edge with a vertex of the set. Dominating sets have many applications, for example in network monitoring, where each vertex in the dominating set corresponds to a monitoring element of the network. It is interesting to minimize the number of vertices in the dominating set (taking the whole vertex set of the graph already gives a dominating set). For instance, the monitoring elements in a network might be costly. We do not know any efficient algorithm (i.e. no algorithm whose running time is polynomial in the number of vertices) that outputs the minimum size of a dominating set of a given graph, and it is strongly believed that such an algorithm does not exist (more precisely, if $\mathrm{P} \neq \mathrm{NP}$, which is a common hypothesis of theoretical computer science, then it does not exist). However, the domination problem can be restricted to graphs that satisfy a given property (the graphs that satisfy this property then form a graph class), and in some cases we obtain polynomial time algorithms. Variants of domination have also been studied, for example the total domination in which a vertex does not dominate itself, or connected domination in which the graph between the vertices of the dominating set must be connected (i.e. there must be a path, from edge to edge, linking any pair of vertices).

Reconfiguration problems

As I was studying Graph Theory, my question about Rubik's cube remained unanswered. Until I was introduced to the reconfiguration framework during my thesis. In reconfiguration problems, we are interested in the solutions of a problem called the source problem. We want to find out if it is possible (and if it is, how) to go from a source solution to a target one by performing what we call a reconfiguration sequence. A reconfiguration sequence is a sequence of solutions of the source problem (which we call configurations), such that two solutions that are consecutive in the sequence are seperated by a move. The definition of the move is given by a rule called the adjacency rule. For example, one can study the reconfiguration of dominating sets in graphs. Then, the source problem is the domination problem, and a move can for example consist in adding or removing a vertex from the set. In the Rubik's cube, the configurations are the possible colorings of the cube with six colors, such that there are nine stickers of each color. A move then corresponds to the rotation of a face of the Rubik's cube. And the player is searching for a reconfiguration sequence from the initial configuration to the target configuration in which the stickers on each face all have the same color. Another famous example is the 15-puzzle. The configurations are the possible positions of fifteen tiles on a 4×4 grid, a move consists in sliding a tile from its position to the hole, and the player needs to find a reconfiguration sequence from the initial configuration to the target one, which creates an image. The last feature that defines a reconfiguration problem is the question we want to answer, i.e. the problem itself. A lot of different problems have been studied in the literature, but the four followings are the most commonly studied.

- Reachability. This is the most studied problem. It asks, given two feasible solutions of the source problem, if there exists a reconfiguration sequence from the first to the second.
- Shortest transformation. Given two feasible solutions such that there exists a reconfiguration sequence from the first to the second, this problem asks what is the minimum length of such a sequence.
- Connectivity. The goal is then to determine if there exists a reconfiguration sequence between any two feasible solutions of the source problem.
- Diameter. This problem asks what is the maximum length of a shortest transformation between any two feasible solutions of the source problem.

In particular, the problem of the Rubik's cube is close to reachability, but in this case we want to find an explicit reconfiguration sequence from the initial configuration of the cube to the target configuration, in which each face has only one color. It is also the problem of the 15-puzzle. However, the question I was wondering about when I was a kid was about the connectivity of the Rubik's cube, as I was asking if the target configuration was reachable from any other configuration. And I finally learned that it is not the case.

Another way to represent a reconfiguration problem is with a graph called the reconfiguration graph. The vertices of the reconfiguration graph are the feasible solutions of the source problem. And two solutions share an edge in the graph if and only if we can go from one to the other in a single move. A portion of the reconfiguration graph of the Rubik's cube is given in Figure 1.3.

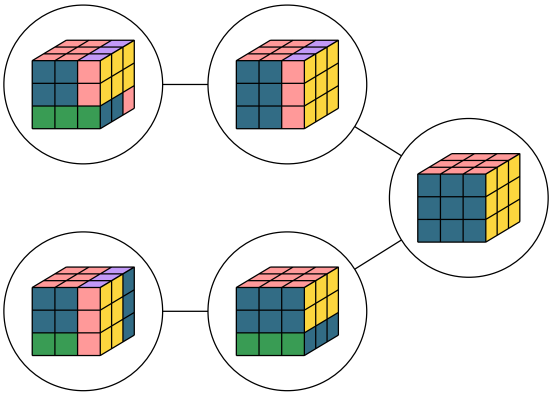


Figure 1.3: A portion of the reconfiguration graph of the Rubik's cube.

Each of the previously mentioned problems can be expressed using the reconfiguration graph. The reachability problem is then equivalent to asking, given two feasible solutions, if there exists a path between them in the reconfiguration graph. The shortest transformation problem asks what is the minimum length of such a path. The connectivity problem is about the connectivity of the reconfiguration graph, and the diameter problem about its diameter.

The relationship between reconfiguration and graph theory is not limitated to the notion of reconfiguration graph. In many cases, the source problem is a graph problem. For example, the reconfiguration of graph colorings has been extensively studied. One of the reasons is that it has important applications, in statistical physics or in communication. Another example is the reconfiguration of connected graphs with the same degree sequence. In this problem, we are given a sequence of integers. The feasible solutions are the connected multigraphs that have this degree sequence. And two multigraphs are adjacent if and only if we can obtain the second by applying a flip to the first, where a flip is an exchange of an endpoint between two disjoint edges. This problem is a generalization of a problem called sorting by reversal, which sorts a permutation of the integers from 1 to n and has many applications, particularly in bioinformatics. Moreover, it has applications in chemistry, as it can be used to enumerate all the molecules with a given formula. The reconfiguration of independent sets and dominating sets are also famous examples of reconfiguration problems in graphs. These two problems can be formulated as the reconfiguration of the position of tokens on the vertices of a graph (where the tokens are placed on the vertices that belong to the set). This generalizes the 15-puzzle, which is equivalent to the reconfiguration of tokens labeled from 1 to 15 on a 4×4 grid graph (although an important difference is that the tokens in the reconfiguration of independent sets and dominating sets are not labeled). For these tokens reconfiguration problems, three adjacency rules have been studied:

- Token Addition-Removal. In this rule, a move can be either the removal or the addition of a token on any vertex.
- Token Jumping. A move under token jumping consists in moving a token from a vertex onto any other one.
- Token Sliding. A move then consists in sliding a token along an edge of the graph.

Note that if a configuration is reachable from another under the token sliding rule, then it also is under token jumping. As we will see, there exist some problems for which the two rules are actually equivalent. Similarly, if a configuration is reachable from another under the token jumping rule, then it also is under token addition-removal. Thus, the token sliding rule is the most constrained one.

For the reconfiguration of dominating sets, the token sliding rule is the one used in another problem recently studied, called eternal domination. It can be described as a game, played on a graph between two players called the defender and the attacker. The defender chooses a set of vertices of the graph where they place some guards. At each turn, the attacker attacks a vertex where there is no guard. To defend, the defender must move one of their guard to the attacked vertex, by sliding it along an edge of the graph. Two versions of this game have mainly been studied, one where no other guard can move, and one where they can all either stay on their vertex or slide along an edge, which is called m-eternal domination. In both cases, the game continues indefinitely, and the defender wins if they can eternally defend against any attacks. At each turn, the guards must be placed on a dominating set of the graph in order to defend against the next attack, and when only one guard can move at a time, the moves that can be performed by the defender correspond to a token sliding move. So in some sense, the game is played on the same reconfiguration graph as in the reconfiguration of dominating sets under token sliding. But in this case, we want to find an infinite walk on this reconfiguration graph, such that at each turn, the chosen configuration contains the attacked vertex. The minimum size of the dominating sets in such a walk, i.e. the minimum number of guards necessary for the defender to win, is called the eternal domination number of the graph, and is the object of many studies.

Like the Rubik's cube or the 15-puzzle, many games can be expressed as reconfiguration problems, and although the reconfiguration framework was introduced recently, reconfiguration problems have been studied for a long time for this reason.There are also many scientific applications. As previously mentioned, the reconfiguration of colorings has applications in statistical physics. The properties of the reconfiguration graph in this case help sampling the set of solutions, which represent the possible values of the spins in a ferromagnetic lattice. More generally, reconfiguration problems are a useful tool for sampling. As for the reconfiguration of connected graphs with the same degree sequence, it has important applications in chemistry, since given a chemical formula, it can be used to enumerate the possible molecules with this formula. Again, reconfiguration problems are a great tool when it comes to enumeration. All these applications make reconfiguration very interesting to study, and since the framework is recent, there remain a lot of open questions.

General Organization

In this thesis, we study both domination problems in graphs, and reconfiguration problems. More precisely, we study the reconfiguration of connected graphs with the same degree sequence, the reconfiguration of dominating sets, and the eternal domination problem.

In Chapter 2, we give the preliminaries necessary to understand the upcoming chapters. We start with some basic definitions of complexity theory. We formally introduce the notions of algorithms, complexity, and we outline some common complexity classes. Then, we properly define graphs, digraphs and multigraphs, and all the vocabulary around it. We also present some usual graph classes and graph problems. Finally, we give more details about the domination problem. We define it formally, we review some fundmental results related to it, and present a few concrete applications and variants of the problem.

In Chapter 3, we present the reconfiguration framework. We first define the components of a reconfiguration problem: the source problem, the instance, the adjacency rule, the reconfiguration graph, and the problem. We illustrate all these definitions with the running example of the 15-puzzle. We then investigate the reconfiguration problems that can be formulated as the reconfiguration of the position of some tokens on the vertices of a graph. In particular, we give some results about several generalizations of the 15-puzzle, about the reconfiguration of independent sets and the reconfiguration of dominating sets. Finally, we present other reconfiguration problems, such as the reconfiguration of graph colorings, the reconfiguration of variable assignments that satisfy a Boolean formula, and the reconfiguration of graphs with the same degree sequence.

In Chapter 4, we present a joint work with Nicolas Bousquet on the reconfiguration of connected multigraphs with the same degree sequence. We already know that the reconfiguration graph is connected for this problem, i.e. we can transform any connected multigraph with a given degree sequence into any other one applying flips. We focus on the shortest transformation problem. We provide a polynomial time algorithm which, given two multigraphs with the same degree sequence, gives a transformation whose length is at most $\frac{5}{2}$ times the length of a shortest transformation. The best ratio known so far was 4. We also show that in order to improve this ratio, we need to change the lower bound we use, i.e. the number of reconfiguration steps that we know are needed.

In Chapter 5, we investigate the reconfiguration of dominating sets under the token addition-removal rule. The results we expose come from a joint work with Nicolas Bousquet and Paul Ouvrard. We study the threshold, which is the maximum size of the dominating sets we take in the reconfiguration graph. More precisely, we investigate the value of this threshold such that above this value, the reconfiguration graph is connected. We give several upper bounds on this value, depending on several graph parameters. In each case, we provide a linear transformation between any two feasible solutions.

In Chapter 6, we change the adjacency rule as we focus on the reconfiguration of dominating sets under the token sliding rule. The results we present come from a joint work with Nicolas Bousquet. In the literature, the reachability problem for the reconfiguration of dominating sets has been studied under the token addition-removal rule, and more recently under the token sliding rule. The problem is hard, and it has been studied in different graph classes. For some of these classes, the problem remains hard, but for others, there exists a polynomial time algorithm. We complete this picture, and answer two open questions on two graph classes. In particular, for one of them, it was believed that the source problem was hard and the reachability problem was easy, but we prove that it is not the case.

In Chapter 7, we present a joint work with Guillaume Bagan and Hamamache Kheddouci on eternal domination. The eternal and m-eternal domination problems have only been studied on undirected graphs. We introduce it on directed graphs, and oriented graphs (directed graphs such that each edge has only one direction). We first generalize a lot of results that are true for undirected graphs to directed graphs. We then introduce the oriented eternal and m-eternal domination, which consist in finding an orientation of a graph that minimizes its eternal or m-eternal domination number. The minimum (m-)eternal domination number is then called the oriented (m-)eternal domination number. We prove that computing the oriented eternal domination number is a hard problem. We also characterize the graphs for which the m-eternal domination number is 2. And we study the value of both parameters in many graph classes.

Chapter 2

Preliminaries

In this first chapter, we introduce the theoretical notions used in this thesis. In Section 1, we give some notions of complexity theory, which we often use in the next chapters. Then, in Section 2, we give some fundamental definitions and results of graph theory, the main field of this thesis. Finally, Section 3 describes in details the DOMINATION problem in a graph, which we study in Chapters 5, 6 and 7.

1 Complexity theory

Complexity theory is a wide field of theoretical computer science, and we can only give an overview in this chapter. Therefore, the definitions we give here are not always rigourous, the idea is to provide the information needed to understand the results we expose in the next chapters. For more details about complexity theory, the reader is referred to the textbooks, like [AB09] or [Sip96].

1.1 Definitions

When confronted to a new problem, the first question that comes is: how to solve it ? A series of instructions that solves this problem is then called an *algorithm*. But finding an algorithm to solve the problem is not always satisfying, we also want it to be efficient. One first has to define formally what efficient means. The efficiency of algorithms, also called complexity, is the object of study of complexity theory. In complexity theory, to describe how efficient an algorithm is, we use two parameters: the time it takes to run it with a computer, which is the *time complexity* of the algorithm, and the space it takes in the computer's memory, which is the *space complexity* of the algorithm. The *time complexity* (resp. *space complexity*) of a problem Π is then the minimum, over all the possible algorithms that can solve Π, of the time complexity of the algorithms. The problem Π is generally posed on a data, which we call the *input* of Π, while the answer is called the *output* of Π. An *instance* of Π is an object that we test as the input. Most of the problems we study in this thesis are *decision problems*, i.e. problems whose output is TRUE or FALSE, or *optimization problems*, i.e. problems whose output is an object that minimizes or maximizes a given function. A *yes-instance* of a decision problem is an instance such that the corresponding output is TRUE, and otherwise it is a *no-instance*. To be able to evaluate the complexity of a new problem, the problems which have similar complexities are grouped into *complexity classes*. We review here some of the most common complexity classes.

1.2 Complexity classes

The P class. The first complexity class we introduce is the class P (where P stands for polynomial). Formally, it is defined using the concept of Turing machines, which are often referred to as the ancestors of computers. The class P is then the class of decision problems that can be solved by a deterministic Turing machine in polynomial time. For more details about Turing machines, the reader is referred to [Sip96]. In this thesis, we simply say that P regroups the decision problems whose time complexity is a polynomial function of the size of the input. If the problems in P are often referred to as "easy", in some cases the degree of the polynomial might be very high or the coefficients very large, thus making the running time of the algorithms very long. For this reason, we sometimes distinguish between the problems in P by the degree of their polynomial. The problems for which the degree of the polynomial is 1 are called *linear*. A classical example of a problem that is in P, and that is actually linear, is the problem 2-SAT. To define it, we need some notions of logic. A *Boolean variable* is a variable that is either TRUE or FALSE. The *negation of a variable* $\overline{x}$ of a variable x is TRUE if and only if x is FALSE. A *literal* is either a variable, or the negation of a variable. The *conjunction* of several variable, denoted by the operator $\wedge$, is TRUE if and only if each of the variables is TRUE. And their *disjunction*, denoted by the operator $\vee$, is TRUE if and only if at least one of them is TRUE. A *Boolean formula* is a formula whose variables are Boolean, and that can be expressed using only conjunctions, disjunctions and negations of these variables. Every Boolean formula F has a *conjunctive normal form* (CNF), which is a conjunction of clauses, where a *clause* is a disjunction of literals.

2-SAT
Input: A CNF Boolean formula F, where each clause contains at most two literals
Output: TRUE if and only if there exists an assignment of the variables of F such that F is TRUE with this assignment.

The NP and co-NP classes. Some problems cannot be decided in polynomial time. But for some of them, it is possible, given an instance $\mathcal{I}$ and an object of this instance which we call a *certificate*, to check that the certificate ensures that $\mathcal{I}$ is a yes-instance of the problem Π, in polynomial time. Typically, in the 2-SAT problem, a certificate could be an assignment of the variables, and it is easy to verify that this assignment satisfies the formula F. The complexity class NP contains the decision problems for which given an instance $\mathcal{I}$ and a certificate of $\mathcal{I}$, we can verify that the certificate makes $\mathcal{I}$ a yes-instance of Π in polynomial time. The ultimate example of a NP problem is the SAT problem, defined as follows.

SAT
Input: A CNF Boolean formula F
Output: TRUE if and only if there exists an assignment of the variables of F such that F is TRUE with this assignment.

More formally, NP is the class of decision problems that can be solved by a non-deterministic Turing machine, with polynomial time. Given that a deterministic Turing machine is also a non-deterministic Turing machine, it implies P $\subseteq$ NP. It is also straightforward when observing that if a problem can be solved in polynomial time, then a certificate of any instance can be checked in polynomial time. However, the following conjecture is still at this day one of the greatest challenge of computer science, although it is commonly used as an hypothesis.

Conjecture. *We have* P $\neq$ NP.

Similarly, there exist some problems for which given an instance and a certificate, it is "easy" (i.e. it can be done in polynomial time) to check that the certificate makes $\mathcal{I}$ a no-instance of the problem. These problems form the *co*-NP class. The *dual problem* $\overline{\mathcal{P}}$ of a decision problem Π in NP, such that every yes-instance of Π is a no-instance of $\overline{\mathcal{P}}$ and conversely, is in co-NP. We have P $\subseteq co -$ NP , but it is also unknown if P $= co -$ NP.

The PSPACE class. In each of the three classes we introduced, the complexity of a problem is expressed through the time it takes to solve it. But in some cases, we might prefer to focus on the space needed to run an algorithm. The complexity class PSPACE gathers the problems whose space complexity is polynomial in the input size. A typical example of a problem in PSPACE is the QBF problem, defined as follows. A *quantifier* is either an *existential quantifier* $\exists$, which is TRUE when there exists an assignment of the variable that makes the formula that follows TRUE, or a *universal quantifier* ∀, which is TRUE when any assignment of the variable makes the formula that follows TRUE. A *quantified Boolean formula* is a formula in which every variable is Boolean and is introduced in the formula by a quantifier, and that only contains conjunctions, disjunctions and negations of these variables after their introduction in the formula. A quantified Boolean formula F can always be written in the form $F = Q_1x_1 \dots Q_nx_n\phi(x_1, x_2, x_3, \dots, x_n)$ where for any i, x_i is a Boolean variable, Q_i is the quantifier ∃ if i is odd and ∀ otherwise, and ϕ is a CNF Boolean formula.

QBF
Input: A quantified Boolean formula F
Output: TRUE if and only if F is TRUE.

More formally, PSPACE is the class of decision problems that can be solved in polynomial space, using a non-deterministic Turing machine. If a certificate of an instance of Π can be checked in polynomial time, then Π can be solved in polynomial space by testing every possible certificate one by one and clearing the memory each time, since the size of a certificate is polynomial. Thus, NP $\subseteq$ PSPACE and $co-$ NP $\subseteq$ PSPACE. However, the following conjecture remains unsolved.

Conjecture. *We have* NP $\neq$ PSPACE.

It is also unknown if $co-$ NP $=$ PSPACE. Some intermediate classes are defined between NP and PSPACE. For each of them, a typical problem is obtained by fixing the number of alternating quantifiers in QBF. These classes form what we call the *polynomial hierarchy*.

The NPSPACE and co-NPSPACE classes. The complexity class NPSPACE is defined similarly to NP, but considering the space complexity instead of the time complexity. More precisely, it is the class of decision problems such that given an instance $\mathcal{I}$ and a certificate of $\mathcal{I}$, one can verify that the certificate makes $\mathcal{I}$ a yes-instance of the problem, with a space complexity polynomial in the size of $\mathcal{I}$. Similarly, the complexity class *co*-NPSPACE is the class of decision problems such that given an instance $\mathcal{I}$ and a certificate of $\mathcal{I}$, we can verify that the certificate makes $\mathcal{I}$ a no-instance of the problem in polynomial space complexity.

It is easily seen that PSPACE $\subseteq$ NPSPACE and PSPACE $\subseteq$ $co-$ NPSPACE, and Savitch actually proved that PSPACE $=$ NPSPACE [Sav70], and Immerman and Szelepcsényi that NPSPACE $=$ $co-$ NPSPACE [Imm88, Sze88]. This means that the information of a certificate does not help with the achievement of a polynomial space complexity. Figure 2.1 illustrates the inclusion relationships between the complexity classes we introduced.

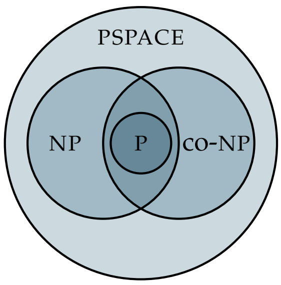


Figure 2.1: The inclusion relationships between the complexity classes P , NP , co-NP , and PSPACE.

Hardness, completeness, and polynomial reductions. Two kinds of relationships can exist between a problem Π and a complexity class C.

Firstly, Π can belong to C. To prove it, the method is generally to give an algorithm with the stated complexity.

The problem Π can also be at least as hard as any problem in C. We say that Π is C-*hard*. To show that Π is C-hard, we use what we call a polynomial reduction. A *polynomial reduction* is a polynomial algorithm that, given an algorithm that solves Π, solves a given C-hard problem Π_2

with a polynomial complexity. The algorithm that solves Π is not explicitly given and is called an *oracle*, or a *black box*. We generally create an instance of Π from any instance of Π_2, then show that the output of Π_2 is TRUE if and only if the output of Π is TRUE . So being able to solve Π makes us able to solve Π_2, and Π is at least as hard as C.

A problem is *C-complete* if it belongs to C and it is C-hard.

As an example, we give a simple polynomial time reduction from SAT to QBF. Let F be an instance of SAT and let $x_1, \ldots, x_n$ be the variables of F. We define the quantified Boolean formula $F' := \exists x_1 \forall y_1 \ldots \exists x_n \forall y_n F(x_1, x_2, \ldots, x_n)$. Since the variables y_i are never used in F, F' is equivalent to $\exists(x_1, x_2, \ldots, x_n) F(x_1, x_2, \ldots, x_n)$. So F' is a yes-instance of QBF if and only if F is a yes-instance of SAT. And an oracle that can solve QBF can be used to solve SAT. Moreover, the formula F' has twice the number of variables of F, so this reduction is polynomial. This implies that QBF is at least as hard as SAT, as we already know from the PSPACE-completeness of QBF, the NP-completeness of SAT, and the fact that NP $\subseteq$ PSPACE.

1.3 Approximation algorithms

Suppose now that we are interested in the solutions of an optimization problem Π, but we know that Π is "hard" (i.e. at least NP-hard). Then, if we are willing to accept solutions that are not necessarily optimal but only close to optimal, a way to get around the hardness of Π is to use an approximation algorithm. A *ρ-approximation algorithm* of Π is an algorithm that outputs a solution for which the value of the function we want to minimize (resp. maximize) is at most (resp. at least) ρ times the value OPT of an optimal solution (note that in the case of a maximization, we have $\rho < 1$). The constant ρ is called the *approximation ratio* of the algorithm. For the approximation algorithm to be interesting, its complexity must be inferior to the complexity of any existing algorithm for Π. Most of the time, we search for polynomial time approximation algorithms. As an example, let us study the optimization problem MAX-3-SAT, defined as follows.

MAX-3-SAT
Input: A CNF Boolean formula F, where each clause contains at most three literals
Output: An assignment of the variables of F such that a maximum number of clauses of F are TRUE with this assignment.

The problem MAX-3-SAT is the canonical problem of a complexity class called MAX-SNP. The decision version of MAX-3-SAT, which consists in determining if there exists an assignment that satisfies at least k clauses for a given k, is NP-complete. So unless P $=$ NP, there exists no polynomial-time algorithm that solves MAX-3-SAT. That being said, outputing the best solution among the one in which the variables are all TRUE and the one in which the variables are all FALSE gives a polynomial time $\frac{1}{2}$-approximation algorithm. Indeed, each clause is TRUE in at least one of these assignments so in one of them at least half the clauses are satisfied. Karloff and Zwick actually designed a polynomial time $\frac{7}{8}$-approximation algorithm for this problem [KZ97].

2 Graph theory

Graph theory is the main field of this thesis, but we only present here a few notions and results, including the ones needed for the understanding of the following chapters. For more details, the reader is referred to the textbooks [Ber01, Bol13, DSS10].

2.1 Definitions

Undirected graphs. A graph is a mathematical structure that represents the existence or the absence of a given relationship between any pair in a group. For instance, in a group of people, two persons may know each other or not. To model it, one can draw the name of each of the persons in this group, and draw a line between any two persons who know each other. This representation constitutes a graph, where each person is a vertex of the graph, and each line is an edge of the graph. An example, which will serve as a toy example through this section, is given in Figure 2.2. More formally, an *undirected graph* G is a pair (V, E) where V is the *vertex set* of G, and E is a set of pairs $\{u, v\}$, where $u, v \in V$, called the *edge set* of G.

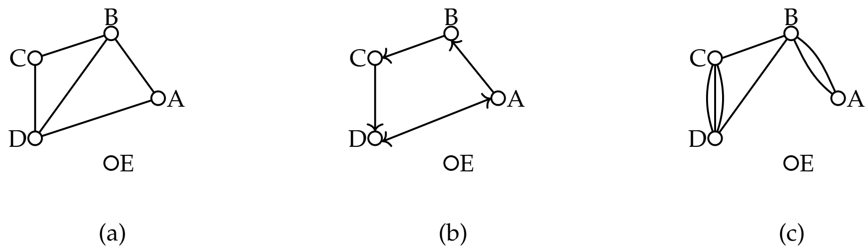


Figure 2.2: The relationships between Alice (A), Bob (B), Claire (C), Denis (D) and Emily (E).
a) The graph modelizing the aquittance relationship. Alice knows Bob and Denis, Bob knows Alice, Claire and Denis, Claire knows Bob and Denis, Denis knows Alice and Bob, and Emily does not know anyone.
b) The digraph modelizing the appreciation relationship. Alice likes Bob and Denis, Bob likes Claire, Claire likes Denis, Denis likes Alice and Emily does not like anyone.
c) The multigraph modelizing the handshakes between Alice, Bob, Claire, Denis and Emily. Alice and Bob shook hands twice, Claire and Denis three times, and Bob and Claire, and Bob and Denis once.

Through this thesis, by abuse of notation, a graph denotes an undirected graph. Moreover, the graphs we work with have no *loop*, i.e. no edge $\{v, v\}$ with $v \in V$. When there is no ambiguity on the graph G, its vertex set will be denoted by V and its edge set by E, otherwise they are respectively denoted by $V(G)$ and $E(G)$. Most of the time, the graphs we work with are up to isomorphism. Intuitively, it means that we do not pay any attention to the names of the vertices, so that in Figure 2.2, it does not matter if we exchange the people who Alice knows with the people who Bob knows. More formally, two graphs G_1 and G_2 are *isomorphic* if their vertex set is $V = \{v_1, \ldots, v_n\}$ and there exists a permutation σ of V such that $v_i v_j \in E(G_1)$ if and only if $\sigma(v_i)\sigma(v_j) \in E(G_2)$.

Let $G = (V, E)$ be a graph. The *order* of G, denoted by n, is its number $|V|$ of vertices, and the *size* of G, denoted by m, is its number $|E|$ of edges. The edge $\{u, v\}$ is also denoted by uv, and two vertices $u, v \in V$ are *adjacent* if $uv \in E$. Let $V = \{v_1, \ldots, v_n\}$. The graph G can be equivalently represented by a $n \times n$ matrix $\mathcal{A}$ such that $\mathcal{A}_{i,j} = 1$ for any integers $i, j \in [1, n]$ such that $v_i, v_j \in E$, and $\mathcal{A}_{i,j} = 0$ otherwise. The matrix $\mathcal{A}$ is called the *adjacency matrix*, or *adjacency list*, of G. Note that it is symmetrical. We say that G is *empty* if $E = \emptyset$. For any $v \in V$, the set of vertices that are adjacent to v is the *neighborhood* of v, denoted by $N(v)$, and the *degree* of v is $d(v) := |N(v)|$. The *closed neighborhood* of v is $N[v] := N(v) \cup \{v\}$. A vertex of degree 0 is called an *isolated vertex*, a vertex of degree 1 is called a *pendant vertex*, and a vertex of degree $n-1$ is called a *universal vertex*. We say that G is *k-regular* if every vertex in V has degree k. The maximum, over all the vertices v of G, of the degrees of v, is called the *maximum degree* of G, denoted by $\Delta(G)$.

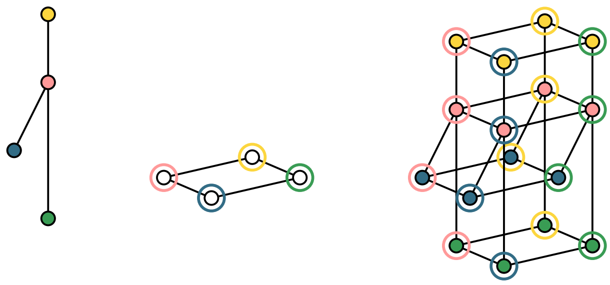
A *subgraph* of G is a graph H such that $V(H) \subseteq V$ and $E(H) \subseteq E$. The *subgraph induced by S*, where $S \subseteq V$, is the subgraph whose vertices are in S, and whose edges have both endpoints in S. A *walk* from $v_1 \in V$ to $v_\ell \in V$ is a sequence $(v_1, v_2, \ldots, v_\ell)$ of vertices of G, such that for any i such that $1 \leq i \leq \ell - 1$, $v_i v_{i+1} \in E$. A *closed walk* is a walk from v_1 to v_1. We say that G is *connected* if for any two vertices $u, v \in V$, there exists a walk from u to v. And for any integer $k \geq 1$, G is *k-connected* if $n \geq k$, and if G remains connected when one removes any set of at most $k-1$ vertices. A *connected component* of G is a connected induced subgraph that is maximal by inclusion with this property. For any two distinct vertices $u, v \in V$ in the same connected component, the *distance* between u and v is the length of a shortest walk from u to v. If G is connected, its *diameter* is the maximum distance between any two vertices. A *minor* of G is a graph that can be constructed from G by deleting edges and vertices and by contracting edges, where the *contraction* of an edge consists in removing the edge and merging its two vertices (the neighborhood of the new vertex is thus the vertices that remain in the union of the two neighborhoods).

The *cycle* C_n is the graph (V, E) such that $V = \{v_0, \ldots, v_{n-1}\}$ and $E = \{v_i v_{i+1 \mod n}, 0 \leq i \leq n-1\}$. In other words, its vertices and edges form a closed walk where each vertex and each edge appears exactly once. The *path* P_n is the graph (V, E) such that $V = \{v_1, \ldots, v_n\}$ and $E = \{v_i v_{i+1}, 1 \leq i \leq n-1\}$. In other words, its vertices and edges form a walk from one vertex to another where each vertex and each edge appear exactly once. The *star* S_n is the graph (V, E) such that $V = \{v_1, \ldots, v_n\}$ and $E = \{v_1 v_i, 2 \leq i \leq n\}$. The *complete graph* of order n, denoted by K_n, is the graph such that every pair of vertices is an edge of K_n. Thus, it is the only graph of order n and size $\frac{n(n-1)}{2}$. A complete induced subgraph is called a *clique*. Figure 2.3 gives an example of a cycle, a path, a star, and a complete graph.

C_5 P_3 S_6 K_5

Figure 2.3: A cycle, a path, a star, and a complete graph.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs. The *complement* of G_1 is the graph whose vertex sex is V_1 and that contains all the edges that are not in G_1 (except for the loops). The *cartesian product* of G_1 and G_2, denoted by $G_1 \square G_2$, is the graph whose vertex set is the cartesian product $V_1 \times V_2$, and where two vertices (u_1, u_2) and (v_1, v_2) are adjacent if and only if either $u_1 = v_1$ and u_2 is adjacent to v_2 in G_2, or $u_2 = v_2$ and u_1 is adjacent to v_1 in G_1. The cartesian product is illustrated by an example in Figure 2.4. Similarly, the *strong product* of G_1 and G_2, denoted by $G_1 \boxtimes G_2$, is the graph whose vertex set is the cartesian product $V_1 \times V_2$, and where two vertices (u_1, u_2) and (v_1, v_2) are adjacent if and only if $u_1 = v_1$ and u_2 is adjacent to v_2 in G_2, or $u_2 = v_2$ and u_1 is adjacent to v_1 in G_1, or u_1 is adjacent to v_1 in G_1 and u_2 is adjacent to v_2 in G_2.



G_1 G_2 $G_1 \square G_2$

Figure 2.4: The cartesian product of two graphs.

Digraphs. Until now, the relationships we wanted to model were mutual, but it might not always be the case. Indeed, we can consider that if Alice knows Bob, then Bob knows Alice, but if we look at the appreciation relationship, it is different: Alice might appreciate Bob even if Bob does not appreciate Alice. To represent these one-sided relationships, we can use arrows instead of lines: if Alice likes Bob, we draw an arrow from Alice to Bob. This representation constitutes a directed graph, or digraph. An illustration is given in Figure 2.2. More formally, a *digraph* G is a pair (V, A) where V is the *vertex set* of G, and A is a set of couples (in contrary to a pair, the order of the two elements matters in a couple) (u, v), where $u, v \in V$, called the *arc set* of G. The *order*, *size* and *adjacency matrix* are defined similarly as for graphs, but in this case, the adjacency matrix is not necessarily symmetrical. An *orientation* of an undirected graph $G = (V, E)$ is a digraph $G' = (V, E')$ such that for any $(u, v) \in E'$, $uv \in E$, and $(v, u) \notin E'$. In other words, it is an affectation of exactly one direction to each edge of G. An *oriented graph* is a digraph $G = (V, E)$ such that for any $(u, v) \in E$, we have $(v, u) \notin E$. In other words, it is the orientation of some undirected graph.

Let $G = (V, E)$ be a digraph. The *outgoing neighborhood* of u is the set $N^+(u) = \{v : (u, v) \in E\}$ and its *closed outgoing neighborhood* is $N^+[u] = N^+(u) \cup \{u\}$. Similarly, the *incoming neighborhood* of u is the set $N^-(u) = \{v : (v, u) \in E\}$ and its *closed incoming neighborhood* is $N^-[u] = N^-(u) \cup \{u\}$. For any $v \in V$, the *outdegree* of v is $|N^+(v)|$ and the *indegree* of v is $|N^-(v)|$. A *strongly connected component* of G is a subgraph H of G such that for any two vertices u and v of H, there exists a walk from u to v in H (in which we follow the direction of the arcs), and H is moreover inclusion-wise maximal with this property. We say that G is *strongly connected* if G has one strongly connected component.

Multigraphs. We might also want to represent relationships that have occurences. For example, if we want to represent how many times the pairs in a group of people shook hands, we can draw an edge for each of these times. In this case, we obtain a multigraph, as illustrated in Figure 2.2. More formally, the difference between a graph and a *multigraph* is that the edge set of a multigraph is a *multiset*, i.e. a set where each element can appear more than once, the number of times it appears being its *multiplicity*. The *order*, *size* and *adjacency matrix* are defined similarly and the adjacency matrix is symmetrical, but it contains the multiplicities of each edge.

Most of the graph theory problems we encounter in this thesis focus on undirected graphs, although we study directed and oriented graphs in Chapter 7, and multigraphs in Chapter 4. The problems we study in graph theory are often hard to solve in the general case (i.e. NP-hard or PSPACE-hard). Thus, we often focus on special cases, by studying the problem only on some graphs that verify a given property. The set of graphs verifying this property then forms a *class* of graphs.

2.2 Graph classes

We present here some graph classes that we encounter in this thesis.

Trees and treewidth. A *forest* is a graph that does not contain any cycle as a subgraph, and a *tree* is a connected forest. A vertex of degree 1 in a tree is called a *leaf*. Note that paths and stars are trees. A lot of graphs that model real data are trees, such as phylogenetic trees or decision trees. Moreover, a lot of graph problems are easy to solve on trees, and this property led to the definition of a graph parameter called the treewidth, whose aim is to describe how close to a tree a graph is. More formally, a *tree decomposition* of a graph $G = (V, E)$ is a pair (X, T) where X is a set of subsets of V called *bags* and T is a tree whose vertices are the bags of X, and that satisfies:

- For any vertex $v \in V$, v belongs to at least one bag of X
- For any edge $uv \in E$, there exists a bag that contains both u and v
- For any vertex $v \in V$, the set of bags containing v forms a connected subgraph of T.

Figure 2.5 gives an example of a tree decomposition of a graph.

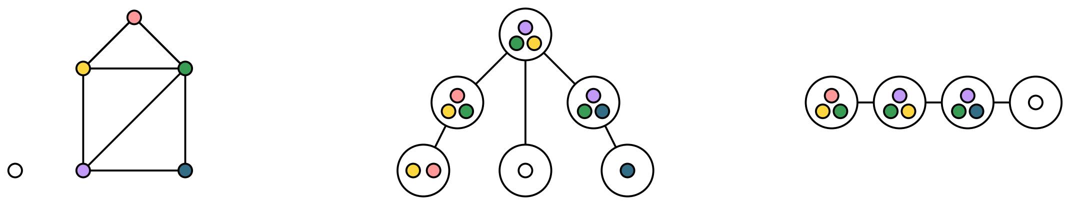


Figure 2.5: On the left, a graph G. In the middle, a tree decomposition of G. On the right, a path decomposition of G.

The minimum, over all the possible tree decompositions of G, of the maximum size of a bag, to which we substract 1, is called the *treewidth* of G and is denoted by $tw(G)$. Note that the graphs of treewidth 1 are exactly the forests. Indeed, by rooting arbitrarily each tree of a forest, and replacing each vertex of the tree by a bag containing the vertex and its parent, we otbain a tree decomposition of treewidth 1. Conversely, if a graph contains a cycle, then any vertex u of this cycle has two neighbors v and w in the cycle. A bag must contain u and v, and a bag must contain u and w. Moreover, if the bags are different, they must be adjacent. By repeatining this argument on every vertex u of the cycle, we obtain a cycle in any tree decomposition of treewidth 1, a contradiction. In general, the treewidth is unbounded. The decision problem that consists in determining whether a given graph has treewidth at most a given k is NP-complete [ACP87]. A tree decomposition of G in which T is a path is called a *path decomposition* of G. The minimum over all path decompositions of G, of the maximum size of a bag, to which we substract 1, is the *pathwidth* of G, denoted by $pw(G)$. Figure 2.5 gives an example of a path decomposition.

Cacti, cographs and chordal graphs. A *cactus* is a connected graph in which any two simple cycles have at most one vertex in common. In other words, it is a connected graph such that every edge belongs to at most one cycle. Cacti can be recognized in linear time. An example of a cactus is given in Figure 2.6.

A *cograph* is a graph that does not contain the path P_4 as an induced subgraph. Cographs are more commonly defined as the graphs that can be generated from a single vertex by using the union operation and the complementation operation. For example, the cograph illustrated in Figure 2.6 can be constructed by taking the union of two graphs: the complement of the union of two single vertices, and the complement of the union of two such graphs. Cographs can be recognized in linear time [CPS85].

A *chordal graph* is a graph G such that every cycle of G on at least 4 vertices has a *chord*, i.e. an edge linking two vertices that are non-adjacent in the cycle. In other words, G has no induced cycle of order other than 3. This is the reason why these graphs are also called *triangulated graphs*. A chordal graph is given in Figure 2.6. Another way to define chordal graphs is with perfect elimination orderings. A *perfect elimination ordering* of a graph $G = (V, E)$ is an ordering of V such that for every $v \in V$, v and the neighbors of v that are after v in the order induce a clique. A graph is chordal if and only if it has a perfect elimination ordering, and Rose, Lueker and Tarjan [RTL76] used this property to show that we can recognize a chordal graph in linear time. If the graph is chordal, their algorithm outputs a perfect elimination ordering.

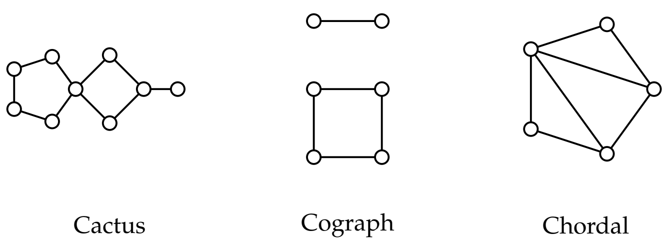
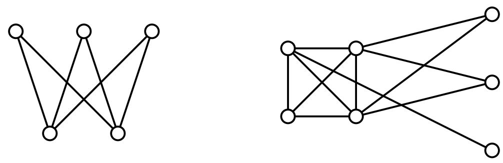


Figure 2.6: A cactus, a cograph and a chordal graph.

Bipartite graphs and split graphs. A *bipartite graph* is a graph (V, E) such that there exists a partition of V into two parts A and B, such that for every edge $e \in E$, one extremity of e is in A and the other is in B. Equivalently, a bipartite graph is a graph that does not contain any induced cycle of odd order. Indeed, any walk in G alternates between vertices of A and vertices of B, so in particular there can be no induced cycle of odd order in G. Conversely, if a graph contains no induced cycle of odd order, then we can create two parts in G by alternating between the two parts in any walk of G. In particular, every forest and cycle of even order is bipartite. Bipartite graphs are often used in modelization, for instance in affiliation networks where the two parts represent two different kind of objects (for example, a part can be a set of people, the other part a set of companies, and a person is adjacent to the companies they worked for). The *complete bipartite graph* $K_{p,q}$ is the bipartite graph of parts A and B where $|A| = p$, $|B| = q$, and such that each vertex of A is adjacent to every vertex of B. An example is given in Figure 2.7.

A *split graph* is a graph whose vertices can be partitioned into a clique and an empty subgraph. Foldes and Hammer proved that a graph G is split if and only if no induced subgraph of G is a cycle of order four or five, or a pair of disjoint edges [FH76]. Hammer and Simeone then provided an algorithm that recognizes split graphs in linear time [HS81]. An example of a split graph is given in Figure 2.7.



$K_{3,2}$ Split

Figure 2.7: A complete bipartite graph and a split graph.

Grid, planar and outerplanar graphs. The *grid of dimensions* $p \times q$, where p and q are integers, is the graph $P_p \square P_q$. An example is given in Figure 2.8. The *toroidal grid of dimensions* $p \times q$ is the graph $C_p \square C_q$. The *king's grid of dimensions* $p \times q$ is the graph $P_p \boxtimes P_q$, and the *toroidal king's grid of dimensions* $p \times q$ is the graph $C_p \boxtimes C_q$. The *rook's graphs of dimensions* $p \times q$ is the graph $K_p \square K_q$. Even if there is only one grid of dimensions $p \times q$ for two given p and q, it is not always straightforward to solve a graph problem for grids, as we will see in Section 3.

A *planar graph* $G = (V, E)$ is a graph that can be drawn in the plane in such a way that no pair of edges intersect. Such a drawing is called a *planar drawing* of G. The decision problem that determines if a given graph is planar is linear, and if it is, a planar drawing can be found in linear time [HT74]. Moreover, there exists a planar drawing of G such that every edge is represented by a straight line [Ist48]. An example of such a planar drawing of K_4 is given in Figure 2.8. From now on, we assume that we are given a drawing of G. A *face* of G is a connected region of the plane bounded by the edges of the graph, including the outer region which is the *outer face*. A very famous result concerning planar graphs is Euler's formula, proven by Leonard Euler in its correspondance (a history of the formula can be found in [Deb10]), which states the following:

Theorem. *For any planar drawing of a graph $G = (V, E)$, we have $n - m + f = 2$, where n is the order of G, m is the size of G, and f is the number of faces in the drawing (including the outer face).*

Another fundamental result about planar graphs is Wagner's theorem [Wag37]. A graph class $\mathcal{C}$ is *minor-closed* if for every graph G of $\mathcal{C}$, every minor of G is also in $\mathcal{C}$. Wagner's theorem states the following:

Theorem. *[Wag37] A graph is planar if and only if it does not have $K_{3,3}$ nor K_5 as a minor.*

This result raised the following question: can every minor-closed class be characterized by a finite number of forbidden minors ? Robertson and Seymour [RS04] finally answered positively this question in 2004, after 10 years, 20 papers and more than 500 pages of proof, thus opening a new area of graph theory which focuses on finding these forbidden minors.

An *outerplanar graph* is a graph that has a planar drawing in which all vertices belong to the outer face of the drawing. Figure 2.8 gives an example of an outerplanar graph. The class of outerplanar graphs is minor-closed, and its two forbidden minors are K_4 and $K_{3,2}$ [DSS10]. The treewidth of outerplanar graphs is bounded by 2, and an optimal tree decomposition is easily found, which makes a lot of graph problems easy to solve with dynamic programming for this class.

Note that forests, cycles and cacti are outerplanar graphs, and grids are planar graphs.

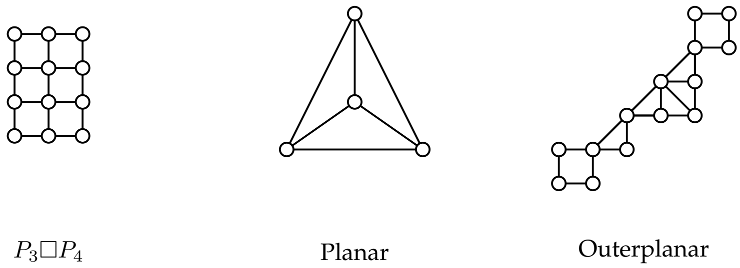


Figure 2.8: A grid, a planar and an outerplanar graph.

Intersection graphs and line graphs. An *intersection graph* is a graph whose vertices are sets, and such that two vertices share an edge if and only if their corresponding sets intersect. Every graph is an intersection graph [EGP66], but we list here some intersection graphs in which the sets verify certain properties.

The most common example of intersection graphs is the interval graphs. An *interval graph* is an intersection graph of intervals of the real line. A *proper interval graph* is an intersection graph of intervals that are not included in each others. In particular, a proper interval graph is an interval graph. Similarly, a *circular arc graph* is an intersection graph of intervals of a circle. In other words, every vertex is associated an arc of the circle and there is an edge between two vertices if their two corresponding arcs intersect. Recognizing an interval graph and giving an interval representation can be done in linear time [BL76]. The same holds for circular arc graphs [McC03].

A *circle graph* is an intersection graph of chords of a circle. In other words, every vertex is defined with two points on a circle and there is an edge between two vertices if the chords leaving their respective pair of points intersect. Note that this is different from a circular arc graph, as two arcs can intersect while the chords between their extremities do not. Actually, an equivalent way to represent a circle graph consists in defining one real interval for each vertex and there is an edge between two intervals if their respective intervals intersect but do not overlap. This second condition makes the class more complicated than interval graphs. Spinrad gave a quadratic algorithm that determines if a graph is a circle graph, and outputs a circle representation if it is [Spi94].

A *unit disk graph* is a graph such that there exists a set C of circles of radius 1 in the plan and a bijection from V to C such that for any $u, v \in V$, $uv \in E$ if and only if the two image circles of u and v intersect. Most of the graph problems are hard on unit-disk graphs. Even determining if a given graph is a unit disk graph is NP-hard [BK98]. Moreover, there exist some unit disk graphs for which it is impossible to output a unit disk representation in polynomial time [MM13].

Let $G = (V, E)$ be a graph. The *line graph* $L(G)$ is the graph such that $V(L) = E$, and for any $e, f \in V(L)$, we have $ef \in E(L)$ if and only if e and f share a vertex in G. A graph L is a *line graph* if there exists a graph G such that $L = L(G)$. Whitney [Whi92] proved that every connected graph G is fully characterized by its line graph, with the exception of the two graphs K_3 and S_4 which both have K_3 as their line graph. Thus, every graph problem has an equivalent formulation in terms of its line graph, and line graphs are useful tools in graph theory. Given a graph G, one can determine if G is a line graph, and output the graph it is the line graph of, in linear time [Rou73].

Figure 2.9 illustrates these definitions with examples.

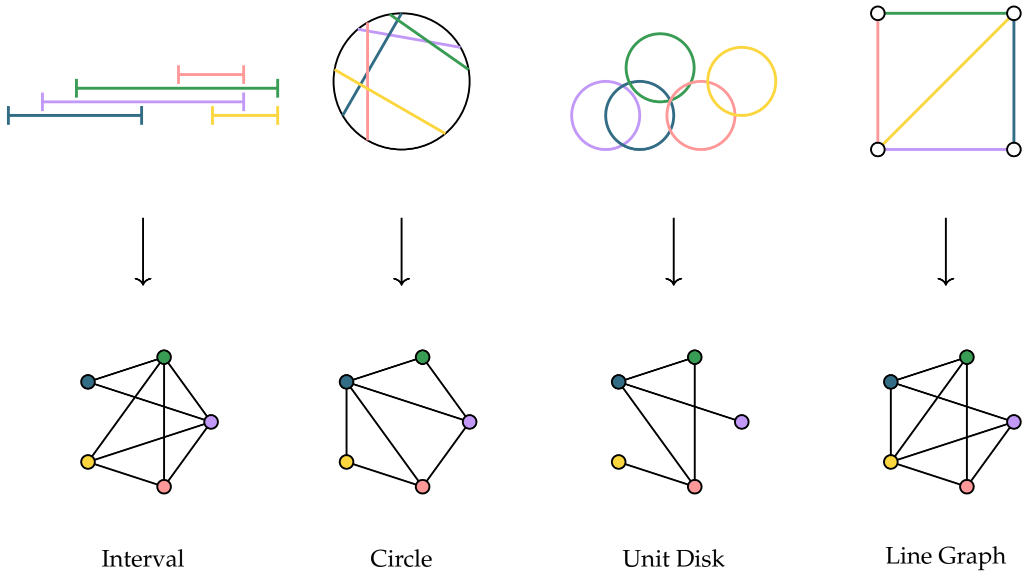


Figure 2.9: An interval, circle, unit disk and a line graph.

2.3 Graph problems and parameters

Since it models binary relationships, graphs can be used to study real networks, such as social networks, physical contacts in a population, interactions between atoms, etc. For this reason, graph theory has applications in social sciences, epidemiology, chemistry, and many other fields. In these applications, the graph represents the data, and the problem on this data then becomes a purely mathematical graph theory problem. One of the earliest and most famous application of graph theory is the one of the seven bridges of Könisberg. Back in 1735, Königsberg was a city in Prussia, that was crossed by the Pregel river, with two islands in the river. It thus contained four lands, linked together by seven bridges, as illustrated in Figure 2.10. The folklore tells that the following question was raised by the king: Is it possible to take a walk in the city, cross each of the seven bridges exactly once, then come back to the starting point ? The great mathematician Leonhard Euler answered this question by proving it was impossible [Eul41].

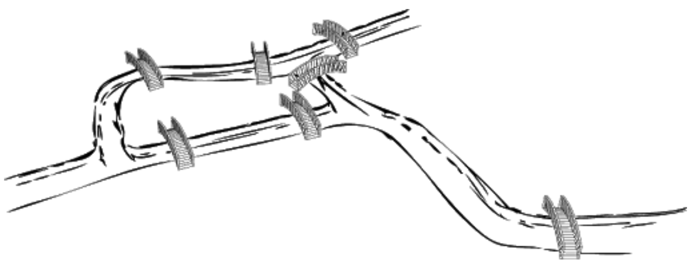


Figure 2.10: The seven bridges of Konigsberg.
Source: http://planund.com/

This problem is often referred to as the first problem of graph theory, and is now known as an instance of the Eulerian cycle problem.

EULERIAN CYCLE
Input: A graph G
Output: TRUE if and only if there exists a closed walk that goes through every edge of G exactly once.

Euler showed that a graph G is a yes-instance of EULERIAN CYCLE if and only if every vertex in G has even degree. But for many graph problems, the yes-instances do not have such a simple characterization. We review here some of these problems, as well as the graph parameters that are associated.

Hamiltonian cycle. The Hamiltonian cycle problem is similar to the Eulerian cycle problem, but the walk has to go through each of the vertices, instead of the edges, exactly once.

HAMILTONIAN CYCLE
Input: A graph G
Output: TRUE if and only if there exists a closed walk that goes through every vertex of G exactly once.

An example of Hamiltonian cycle in a graph is given in Figure 2.11.

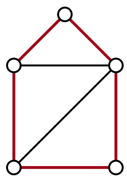


Figure 2.11: An example of a graph G and a Hamiltonian cycle of G.

This problem is a special case of the TRAVELLING SALESMAN problem which, given a list of cities and the distances between each pair of cities, asks what is the shortest closed walk visiting each city. The TRAVELLING SALESMAN problem is a very well known problem that has many applications. It is obviously used in planning and logistics, but also in DNA sequencing (the cities represent DNA fragments and the distance between two fragments are their genetic distance) or even in astronomy. Contrarily to EULERIAN CYCLE, no easy characterization of the yes-instances of HAMILTONIAN CYCLE, called the *Hamiltonian graphs*, is known, and this problem is NP-complete in general [PS03].

Maximum clique. To illustrate the remaining graph problems, we will use the running toy example of the organization of a scientific event. Suppose first that for it to be a successful event, we want everyone to know each other, and we want to invite a maximum number of people. In the group of people of Figure 2.2, for instance, such an event could gather Alice, Bob and Denis, or Alice, Claire and Denis. Using the graph representation, this problem consists in finding a clique of maximum size in the graph, which is an optimization problem known as the MAXIMUM CLIQUE problem. Its associated decision problem is the following.

CLIQUE
Input: A graph G, and integer k
Output: TRUE if and only if there exists a clique of order at least k in G.

The maximum order of a clique in G is the *clique number* of G, denoted by $\omega(G)$, and illustrated in Figure 2.12.

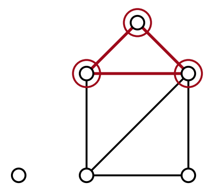


Figure 2.12: An example of a graph G and a maximum clique of G, which gives $\omega(G) = 3$.

This problem has many applications in social sciences, where finding groups of people who all know each other, or share a common interest, can be very useful in the understanding of social behaviours. Note that the term clique actually comes from the term social clique. Deciding if there exists a clique of order k in G is a NP-complete problem [Kar72].

Maximum independent set. Let us go back to our event. A scientific event where everybody knows each other might not be very productive. So let us now consider the problem where we want to organize our event in such a way that nobody knows each other, and with a maximum number of convits. In the group of people of Figure 2.2, the only solution would be to invite Alice, Claire and Emily. A set $S \subseteq V$ is an *independent set* of G if $uv \notin E$ for every u, v in S, and this problem actually consists in finding an independent set of maximum cardinality in the associated graph. This problem is known as the MAXIMUM INDEPENDENT SET problem, and the associated decision problem is defined as follows.

INDEPENDENT SET
Input: A graph G, an integer k
Output: TRUE if and only if there exists an independent set of G of order at least k.

The *independence number* $\alpha(G)$ is the size of a largest independent set of G. The notions of independent set and independence number are illustrated in Figure 2.13.

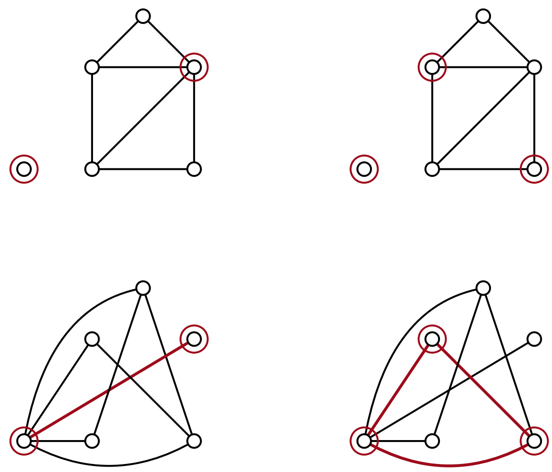


Figure 2.13: An example of a graph G and two independent sets of G. The independent set on the right is maximum, thus giving $\alpha(G) = 3$. The two corresponding cliques in $\overline{G}$ are represented below.

Note that, as illustrated in Figure 2.13, an independent set of G is a clique in the complement $\overline{G}$. So we have $\alpha(G) = \omega(\overline{G})$. It implies that this problem is NP-complete [Kar72], even though the problem of finding an independent set maximal by inclusion can be solved in polynomial time with a greedy algorithm [Lub86].

Graph coloring. Now, even though it is productive to gather people who do not know each other, it is a shame not to be able to invite everyone, and a solution could be to simply gather people in different rooms, such that every room only contains people who do not know each other, and with a minimum number of rooms. For example, in the group of people of Figure 2.2, the optimal solution would be a room with Alice and Claire, a room with Bob, and a room with Denis, and Emily can be in any of these rooms. This problem describes the GRAPH COLORING optimization problem. A *proper coloring* of G is an attribution of a color to each of the vertices of G, such that no adjacent vertices have the same color. In other words, it is a partition of V into independent sets, where each independent set is a color. The decision problem COLORABILITY is defined as follows.

COLORABILITY
Input: A graph G, an integer k
Output: TRUE if and only if there exists a proper coloring of G with at most k colors.

The minimum number of colors in a proper coloring of G is called the *chromatic number* of G, denoted by $\chi(G)$. It is illustrated in Figure 2.14. Note that the bipartite graphs are exactly the graphs of chromatic number 2. The *clique covering number* $\theta(G)$ of G is the chromatic number of the complement of G. In other words, it is the minimum number of cliques in which G can be partitioned.

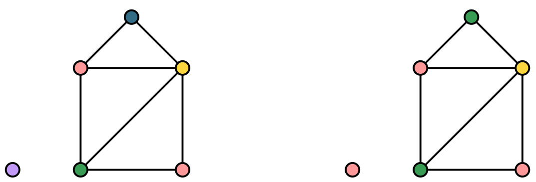


Figure 2.14: An example of a graph G and two proper colorings of G. The number of colors is minimum on the right, thus giving $\chi(G) = 3$.

For any graph G, in a clique of G, since every pair of vertices shares an edge, every vertex must have distinct colors in a proper coloring. Thus, we have $\chi(G) \geq \omega(G)$. The graphs G for which $\chi(H) = \omega(H)$ for any induced subgraph H of G are called the *perfect graphs*. The strong perfect graph theorem, proved by Chudnovsky, Robertson, Seymour, and Thomas in 2006, characterizes the perfect graphs as follows.

Theorem. *[CRST06] A graph is perfect if and only if it contains no hole nor antihole, where a hole is an induced cycle of odd order at least* 5 *and an antihole is the complement of a hole.*

Since each color induces an independent set of G, and the size of an independent set is at most $\alpha(G)$, we also have $\chi(G) \geq \frac{n}{\alpha(G)}$.

The chromatic number is, in the general case, unbounded. However, it is bounded for some graph classes. For instance, the four color theorem [AH76] states that for any planar graph G we have $\chi(G) \leq 4$. It is a famous result concerning graph coloring, partly because it was the first theorem with a computer assisted proof. It implies in particular that every geographic map can be colored with four colors such that two countries that share a border do not have the same color. Graph coloring has many other applications, particularly in logistics and scheduling. In these applications, we are given a set of jobs that each require the same amount of time. For some pairs of jobs, it is not possible to complete the two jobs at the same time (if, for example, they use the same ressource). And we want to minimize the time taken to complete all the jobs. Then, if we represent each job by a vertex and two vertex share an edge if the two jobs cannot be completed at the same time, then minimizing the time corresponds to fiding the minimum number of colors in a proper coloring of the graph.

In the general case, this problem is NP-complete. Actually, it is even NP-complete when restricted to the case where $k = 3$ [Kar72].

Vertex cover. Separating the people who know each other in our event might be a little strict, and we may want to have more details about the relationship between two persons before deciding to seperate them. To do so, we can ask any guest to describe the relationship they have with every guest they know, and we want every relationship to be described, but we want to ask a minimum number of people, not to bother them. For example, in Figure 2.2, the only optimal solution would be to ask Bob and Denis. This problem corresponds to the MINIMUM VERTEX COVER optimization problem, and its associated decision problem is defined as follows, where a set $C \subseteq V$ is a *vertex cover* of G if for any $uv \in E$, $u \in C$ or $v \in C$ (or both).

VERTEX COVER
Input: A graph G, an integer k
Output: TRUE if and only if there exist a vertex cover of G of size at most k.

The vertex cover number $\tau(G)$ is the size of a minimum vertex cover of G. The notions of vertex cover and vertex cover number are illustrated in Figure 2.15.

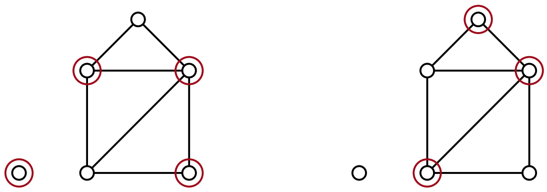


Figure 2.15: An example of a graph G and two vertex covers of G. The vertex cover is minimum on the right, thus giving $\tau(G) = 3$.

Note that C is a vertex cover of G if and only if $V \setminus C$ is an independent set of G. This implies that $\tau(G) = n - \alpha(G)$ for any graph G, and that the VERTEX COVER problem is NP-complete.

Matching. Let us finally consider the following problem. Concerning the accomodation for our event, we need to place people in double and single rooms, but we want to pair up people who know each other only, and we want to maximize the number of double rooms, for budget reasons. Thus, an optimal solution in the example of Figure 2.2 would be to pair up Alice and Denis, and Bob and Claire, and to place Emily in a single room. This optimization problem is the MAXIMUM MATCHING problem. A *matching* of a graph G is a set of edges of G without common vertices. The associated decision problem MATCHING is defined as follows.

MATCHING
Input: A graph G, an integer k
Output: TRUE if and only if there exists a matching of G of size at least k.

The maximum size of a matching of G is called the *matching number* $\mu(G)$ of G, illustrated in Figure 2.16.

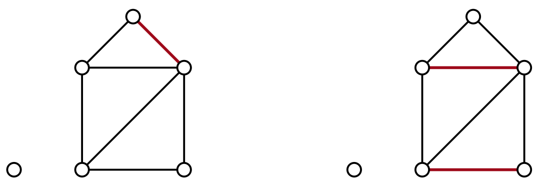


Figure 2.16: An example of a graph G and two matchings of G. The matching is maximum on the right, and thus $\mu(G) = 2$.

A matching of size $\frac{n}{2}$, or in other words such that every vertex is adjacent to exactly one edge of the matching, is called a *perfect matching*. The following theorem, called Hall's marriage Theorem, is a famous result concerning the existence of a perfect matching in a bipartite graph.

Theorem. *[Hal35] Let G be a bipartite graph of parts X and Y, such that $|X| = |Y|$. There exists a perfect matching in G if and only if for every subset W of X, we have $|W| \leq |N(W)|$.*

The MATCHING problem has applications in chemistry, in which a matching between the carbon atoms of an aromatic compound is called a Kekulé structure. Matchings are particularly studied in bipartite graphs. In logistics, it is known as the graduation problem, in which we are given a set of requirements to graduate from a university, and a set of classes that can each complete several requirements, but can each be used to complete only one requirement.

The MATCHING problem can be solved in polynomial time in general graphs, using the so-called blossom algorithm due to Edmonds [Edm65]. The running time of this algorithm is in $\mathcal{O}(\sqrt{n}m)$.

3 The Domination problem in graphs

As the domination problem is one of the main topics of this thesis, we give more details about this particular graph problem. That being said, we only present a few results here, and for a more complete state of the art, the reader is referred to the textbooks [HHS98].

3.1 Definitions

Let us come back to our scientific event. We would like now to have a description of every guest, in order to choose a personalized gift for them. To do so, for each guest, we can obtain this person's description either from themselves, or from another guest who knows them. For example, in the group of people of Figure 2.2, we need to ask Emily to describe herself, and we can ask Alice to describe herself, and ask Claire to describe Bob, Denis and herself. This problem is equivalent to finding a dominating set of the graph. More formally, a *dominating set* of $G = (V, E)$ is a set $D \subseteq V$ such that every vertex of G either is in D or is adjacent to a vertex of D. When the graph G is not empty, and $n \geq 2$, there are more than one dominating set of G. That being said, it might be interesting to search for a dominating set of minimum size. In our group of people, for example, we may want to bother a minimum number of people. Bothering Emily is unevitable, since nobody else knows her, but asking Bob is enough to get all the other descriptions, thus we can ask two people instad of three. A dominating set of a graph G of minimum size is called a *minimum dominating set* of G, and its size is the *domination number* of G, denoted by $\gamma(G)$. These two notions are illustrated in Figure 2.17.

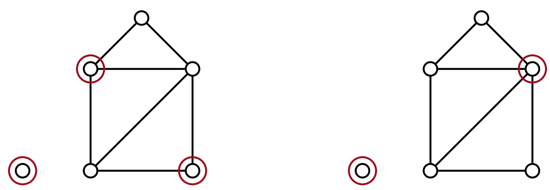


Figure 2.17: An example of a graph G and two dominating sets of G. The dominating set on the right is minimum, thus giving $\gamma(G) = 2$.

The problem of finding a minimum dominating set in a graph is a famous optimization problem called the MINIMUM DOMINATING SET problem, and its associated decision problem DOMINATION is defined as follows.

DOMINATION
Input: A graph G, an integer k
Output: TRUE if and only if there exists a dominating set of G of size at most k.

3.2 State of the art

Domination problems in graphs were first studied in the 50s, but the notion of dominating set was formally introduced by Ore in 1965 [Ore65]. In this same book, Ore proves the following fundamental result.

Theorem. *[Ore65] If $G = (V, E)$ is a connected graph of order $n \geq 2$, then $\gamma(G) \leq \frac{n}{2}$.*

Proof. Let D be a dominating set of G, minimal by inclusion, and let us show that $V \setminus D$ is a dominating set of G. Assume for contradiction that there exists $v \in V$ that is not dominated by $V \setminus D$. Then, by definition, v is not in $V \setminus D$, and no vertex adjacent to v is in $V \setminus D$. Thus, $v \in D$, and every vertex adjacent to v in G is in D. Since G is connected, at least one vertex is adjacent to v in G and thus v is dominated by $D \setminus v$, as well as every vertex adjacent to v in G. So $D \setminus v$ is a dominating set of G, a contradiction with the minimality of D. Therefore, by definition, $\gamma(G) \leq |D|$ and $\gamma(G) \leq n - |D|$, which gives the result. □

To see that this bound is sharp, let us consider, for an even n, a graph constructed from the path $P_{\frac{n}{2}}$ by adding a leaf on each of its vertices. Since each of the $\frac{n}{2}$ leaves is only adjacent to its path vertex, it is easily seen that $\gamma = \frac{n}{2}$. Baogen et al. actually gave a complete characterization of the graphs for which the equality is reached [BCH⁺00].

The domination number can also be compared to other graph parameters such as the independence, or the vertex cover number.

Observation. *For any connected graph G, we have $\gamma(G) \leq \tau(G)$.*

Indeed, by definition, a vertex cover contains one of any two adjacent vertices of G, and since G is connected, every vertex has at least one adjacent vertex, thus the vertex cover is a dominating set. The example of stars, for which $\gamma(S_n) = 1$ and $\tau(S_n) = 1$ for any n, shows that this bound is tight.

Observation. *For any graph G, we have $\gamma(G) \leq \alpha(G)$.*

Indeed, since in a maximum independent set I of G, adding any vertex induces an edge, it means that each vertex of $V \setminus I$ has a neighbor in I. So a maximum independent set is a dominating set. The example of complete graphs, for which $\gamma(K_n) = 1$ and $\alpha(K_n) = 1$ for any n, shows that this bound is tight.

The value of the domination number has been particularly studied in grids, since it is a particular case of the following conjecture by Vizing, which is one of the most studied domination problems.

Conjecture. *[Viz68] For any graphs G and H, we have $\gamma(G \square H) \geq \gamma(G) \cdot \gamma(H)$.*

In 1992, Chang [Cha92] conjectured the following exact value of the domination number in the grid $P_p \square P_q$, which was then proved in 2011 by Gonçales, Pinlou, Rao and Thomassé [GPRT11].

Theorem. *[GPRT11] For any p and q such that $p \geq 16$ and $q \geq 16$, we have $\gamma(P_p \square P_q) = \lceil \frac{(p+2)(q+2)}{5} \rceil - 4$.*

This result, and the computation of γ in the other cases, by Fisher [Fis93] with a computer, confirms Vizing's conjecture for grids.

In 1979, Garey and Johnson [GJ79] proved that the decision problem DOMINATION is NP-complete. Since then, it has been studied in particular classes of graphs. For some graphe classes such as outerplanar graphs [ABF⁺02] or circular arc graphs [HT91], the problem is linear, but it is NP-complete in most of the studied classes such as bipartite graphs [Dew81], line graphs [YG80], circle graphs [Kei93] or unit disks graphs [CCJ91].

3.3 Applications

Domination problems in graphs have been introduced in the early years of graph theory because it is a very natural problem, with many applications, in various fields such as physics, biology, computer science or social sciences.

A common application in these fields is the surveilliance of a network, wheather it is an electrical, biological, computer, or social network. The problem is always the same: we want to be able to monitor every element of the network, by choosing the minimum number of elements to do so, an element being able to monitor itself and its neighbors. It is easily seen, when seing the network as a graph, that this corresponds to the dominating set problem.

Another well-known application is the facility location problem, an optimization problem whose goal is to give the optimal location of facilities (hospitals, for example) in order to make it accessible to the population, while minimizing the number of facilities. In the graph whose vertices are districts, and where two districts share an edge if one is accessible from the other (assuming that being accessible is defined, with a maximum distance for example), this problems corresponds to the dominating set problem.

3.4 Variants

A lot of variants of the domination problem have been studied.

The first one that comes to mind is the *total domination*, more restrictive, since a *total dominating set* is a set $D \subseteq V$ such that every vertex of G is adjacent to a vertex of D. In other words, in a total dominating set, a vertex does not dominate itself. The minimum size of a total dominating set is called the *total domination number* of G, denoted by $\gamma_t(G)$. For example, in network monitoring, we might not want to trust an element to monitor itself, and thus use total domination instead of domination to solve our problem. Note that the graph of Figure 2.17 has no total dominating set. Indeed, such a set only exists in graphs that contain no isolated vertex.

Another variation that has been widely studied is the *connected domination*. A *connected dominating set* of G is a dominating set that has the additional constraint of inducing a connected subgraph of G. The *connected domination number* of G, denoted by $\gamma_c(G)$, is the minimum size of a connected dominating set of G. This problem is particularly useful in network monitoring, when we want to transmit information between the elements of the network through the monitoring elements, that thus have to be connected.

We have already seen that a maximum independent set of a graph G is also a dominating set of G. An *independent dominating set* of G is a set that is both an independent set and a dominating set of G. The *independent domination number* of G, denoted by $i(G)$, is the size of a smallest independent dominating set of G. This parameter introduced a graph class, the *domination-perfect* graphs, for which $\gamma(H) = i(H)$ in every induced subgraph H of G.

Among the other variants of the domination problem, we can also name the fractional domination, or the power domination. The domination problem has also been introduced on digraphs, where for example a vertex u dominates v if and only if uv is an arc of G. A lot of results on graphs have been extended to digraphs. Dynamic versions of the domination problems, where we apply changes to the dominating sets, have also been studied, such as the eternal domination presented in Chapter 7, and the reconfiguration of dominating sets presented in Chapters 5 and 6.

Chapter 3

Reconfiguration problems

In this chapter, we introduce some basic notions and classical results of the reconfiguration framework. In Section 1, we define the features that characterize a reconfiguration problem, and illustrate these definitions with the running example of the 15-puzzle. In Section 2, we review some notable results about the reconfiguration of tokens in a graph, that can be seen as a generalization of the 15-puzzle problem, among which the reconfiguration of INDEPENDENT SET and DOMINATING SET. In Section 3, we present some results on three other problems: the reconfiguration of COLORING and SATISFIABILITY, and the reconfiguration of graphs with the same degree sequence. For more details about reconfiguration problems, the reader is referred to the reviews of Jan Van den Heuvel [vdH13] and Naomi Nishimura [Nis18].

1 Definitions

The reconfiguration framework, although formally introduced recently, regroups a lot of mathematical problems that have been studied for decades in combinatorics and graph theory. A reason is that the questions raised in reconfiguration are quite natural, and they have concrete applications in fields such as physics [Mar99], robotics [PRST94, Sur09], as well as in popular games [JS+79, Cul97]. Reconfiguration problems ask the question of if, and how, we can transform a solution of a problem into another by applying changes which maintain the solution. A classical example is the *15-puzzle*. This popular puzzle consists in a set of 15 tiles, arranged in a 4×4 grid, with one hole. The player has to create an image, by repeatively sliding a tile next to the hole, from its position onto the hole. In the mathematical model of the 15-puzzle, we consider a set of 15 tokens, labeled from 1 to 15, on the vertices $v_1, \ldots, v_{16}$ of the 4×4 grid graph. The vertex that has no token is called the *empty vertex*. The player can apply moves, where a move consists in sliding a token along an edge from its position to the empty vertex. The goal is to find a sequence of moves from the current configuration of tokens, to the configuration where any token labeled i is on the vertex v_i, as illustrated in Figure 3.1.

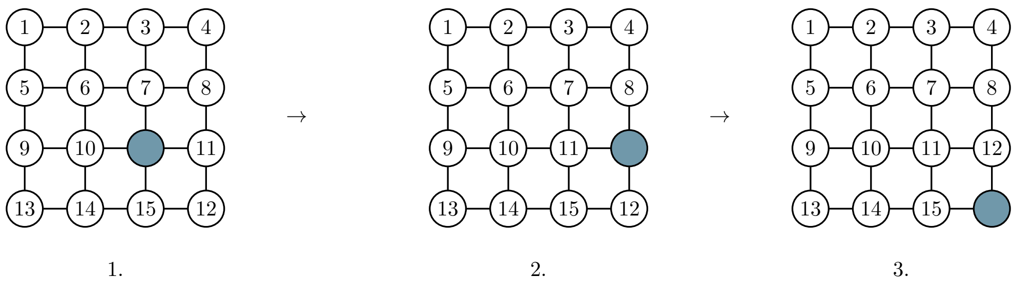


Figure 3.1: A sequence that solves a 15-puzzle in 2 steps.

We use the example of the 15-puzzle to illustrate the classical notions of reconfiguration. As underlined in [Nis18], a reconfiguration problem is defined by 5 features: the source problem, the instance, the feasible solutions, the adjacency rule (these four features characterize the reconfiguration graph), and the problematic. We define here each of these notions.

1.1 Source problem, instance and feasible solutions

The aim of reconfiguration is to transform a solution of a problem into another. The solutions we reconfigurate are then the solutions of the *source problem*. For example, in the 15-puzzle, the source problem can be formulated as follows. What are the possible placements of $n-1$ labeled tokens on a given graph of order n ? The *instance* is the instance of this source problem we are interested in. So in the 15-puzzle, the instance is the 4×4 grid. The solutions of this instance are called the *configurations*, or *feasible solutions*. A variety of source problems has been studied in the literature, such as INDEPENDENT SET [KMM12, IDH+11, HD05], DOMINATING SET [HIM+16, HS14, BDO19], MATCHING [BBH+19], GRAPH COLORING [Cer07, BC09, BH19] or SATISFIABILITY [GKMP09, MNPR17]. We review some of these results in Sections 2 and 3. Additionally to the 15-puzzle, the INDEPENDENT SET reconfiguration problem will often serve as an example in what follows. In this problem, the configurations are independent sets of a graph.

1.2 Adjacency rule

To transform a solution of the source problem into another, one first needs to define the changes that can be applied in such a transformation. The *adjacency rule* is the rule that describes the change we can apply to a solution to go to another in a single step, also called *move*. In the 15-puzzle, the move we can apply to go from one placement of the 15 tokens to another is the sliding of a token along an edge, from its position to the empty vertex. In the framework of SATISFIABILITY reconfiguration, the move consists in changing the value of exactly one Boolean variable. In the reconfiguration of graphs with the same degree sequence, the move consists in flipping two disjoint edges, i.e. exchanging one of their endpoints. Note that in these examples, the adjacency rule is mutual: if we can go from a first configuration to a second, we can go from the second to the first. Most of the works on reconfiguration problems focus on mutual adjacency rules, and it will always be the case in this chapter, but there is no reason why we should restrict to this kind of studies.

1.3 Reconfiguration graph

Let X_s and X_t be two feasible solutions of a source problem Π. A *reconfiguration sequence* from X_s to X_t is a sequence $S =< X_1 := X_s, \dots, X_\ell := X_t >$ such that for any $X_i \in S$, X_i is a feasible solution, and for any two consecutive feasible solutions X_r and X_{r+1} of S, there exists a move (following the adjacency rule) from X_r to X_{r+1}. Thus, solving a 15-puzzle is equivalent to finding a reconfiguration sequence from the initial position of the tokens to the target one (in which every token i is on the vertex v_i). Each element of this sequence is a position of the tokens, reachable from the previous one by the sliding of a token along an edge to the empty vertex.

Another way to represent a reconfiguration sequence from X_s to X_t is with a path on a graph called the *reconfiguration graph* and denoted by $\mathcal{R}$. The vertices of $\mathcal{R}$ are the feasible solutions and two feasible solutions are adjacent in $\mathcal{R}$ if and only if there exists a move from one to the other. A portion of the reconfiguration graph for the 15-puzzle is given in Figure 3.2.

Using this representation, solving a 15-puzzle consists in finding a path in the reconfiguration graph from the initial configuration to the target one, as illustrated in Figure 3.2, where the red path corresponds to the resolution presented in Figure 3.1.

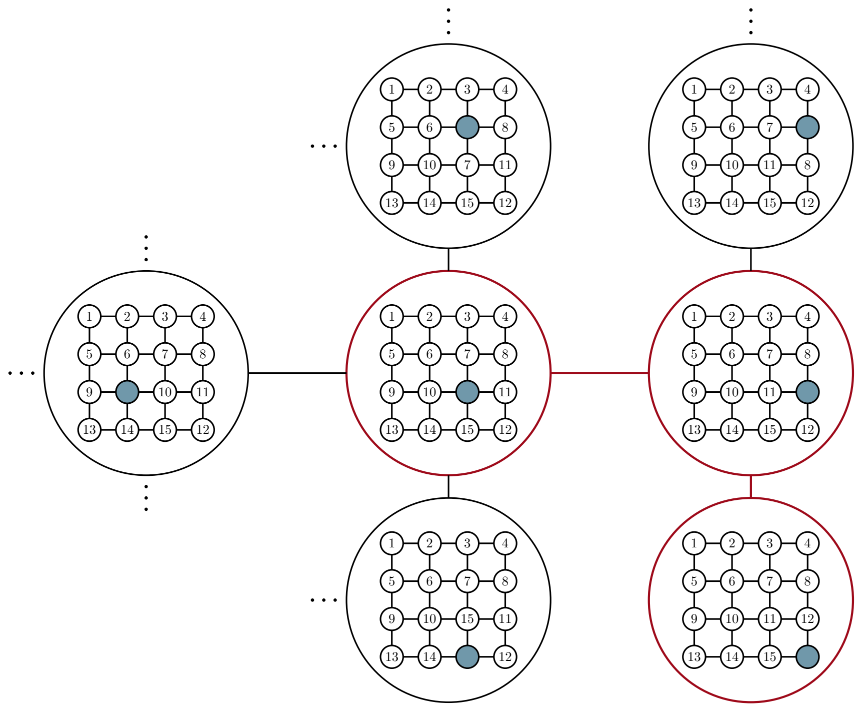


Figure 3.2: A portion of the reconfiguration graph $\mathcal{R}$ for the 15-puzzle.

1.4 Problematic

The last and most important part of the problem is the question we want to solve, i.e. what we want to know given the feasible solutions, the adjacency rule and the instance. In this thesis, we call this question the *problematic*. It can very often be described as a graph problem posed on the reconfiguration graph $\mathcal{R}$. In the 15-puzzle, the problematic is the existence of a reconfiguration sequence from the initial configuration to the target one, or equivalently the existence of a path from the initial to the target configuration in the reconfiguration graph $\mathcal{R}$. Note that this problem is not a structural problem posed on the reconfiguration graph $\mathcal{R}$, it has an input which is the initial configuration, and it is therefore an *algorithmic problematic*, in opposition to a *structural problematic*. Most the algorithmic problematics are easy (i.e. in P) when the reconfiguration graph is given. For instance, since solving a 15-puzzle is equivalent to finding a path in $\mathcal{R}$, if $\mathcal{R}$ is given then the result is straightforward. But the reconfiguration graph $\mathcal{R}$ is not an input of the problem. However, in this thesis, we will always consider reconfiguration problems where we can recognize a feasible solution and find all the feasible solutions adjacent to it in polynomial time. In what follows, we review some of the most common problematics in the reconfiguration framework.

Reachability. The algorithmic problematic *reachability* is the most extensively studied. It is also the one that has the most applications in games. It can be formulated as follows.

REACHABILITY
Input: A source problem Π, an instance $\mathcal{I}$ of Π and two feasible solutions X_s and X_t.
Output: TRUE if and only if there exist a reconfiguration sequence from X_s to X_t (or, equivalently, a path from X_s to X_t in $\mathcal{R}$).

REACHABILITY is a decision problem, and we are generally interested in its complexity. In particular, in Chapter 6, we are interested in the complexity of REACHABILITY for the source problem DOMINATING SET in several graph classes. The REACHABILITY problem is classically

PSPACE-complete [HD05, HIM+16, BC09, GKMP09], although it is sometimes in NP [LM18] or in P [KMM12, CVDHJ11, GKMP09].

Shortest transformation sequence. Another extensively studied algorithmic problematic is *shortest transformation sequence*, which can be formulated as follows.

SHORTEST TRANSFORMATION
Input: A source problem Π, an instance $\mathcal{I}$ of Π and two feasible solutions X_s and X_t such that X_t is reachable from X_s.
Output: The length (i.e. the number of moves) of a shortest reconfiguration sequence from X_s to X_t (or, equivalently, the length of a shortest path from X_s to X_t in $\mathcal{R}$).

Note that in the 15-puzzle game, it is interesting to minimize the number of times we slide a tile before obtaining the target solution. The shortest transformation sequence problematic is an optimization problem, and it is often not solvable in polynomial time. Thus, a lot of work has been done on polynomial time approximation algorithms. To get the approximation ratio, a lower bound can be obtained by noticing that some elements have to be modified from X_s to X_t. These elements belong to the *symmetric difference*, i.e. the set of elements whose state is different in X_s and X_t, denoted by $\Delta(X_s, X_t)$. Typically, in the reconfiguration of independent sets, the symmetric difference is $\Delta(I_s, I_t) = (I_s \setminus I_t) \cup (I_t \setminus I_s)$. In the 15-puzzle, the symmetric difference is the set of tokens that do not have the same position in the initial and target configurations. A trivial lower bound on the length of a transformation is then the size of the symmetric difference, divided by the number of elements that can be removed from the symmetric difference in one move. This is the technique we use to obtain a lower bound on the shortest transformation between connected graphs with the same degree sequence in Chapter 4. While it is obvious that the elements of the symmetric difference need to be changed in a reconfiguration sequence, it is usually not true that there exists a shortest reconfiguration sequence that only changes elements of the symmetric difference, thus this lower bound is usually not tight. Figure 3.3 gives an example of a resolution of the 15-puzzle in 10 steps, where any reconfiguration sequence requires to change the position of tokens that are not in the symmetric difference. In the initial configuration, the only tokens that are placed differently than in the target configuration are 12, 14 and 15, but it is easily seen that if we only slide these tokens, none of the three can reach their final position.

Connectivity. The structural problematic associated to REACHABILITY is *connectivity*, defined as follows.

CONNECTIVITY
Given a source problem Π and an instance $\mathcal{I}$ of Π, does there exist a reconfiguration sequence from X_s to X_t for any two feasible solutions X_s and X_t ? In other words, is $\mathcal{R}$ connected ?

In the 15-puzzle, this problematic can be formulated as follows: can we reach any placement of the 15 tokens from any other one ? Note that up to a permutation on the labels of the tokens, this is equivalent to asking the question: can we reach the target configuration from any initial configuration ? As we will see in Section 2, there actually exist some configurations from which the target configuration is not reachable [JS+79].

A classical method to show connectivity is to prove that any configuration is reachable from a given "canonical" configuration. We use this method in Chapter 5 to show the connectivity of DOMINATING SET reconfiguration under certain conditions. On the other hand, a way to prove that $\mathcal{R}$ is disconnected consists in finding frozen configurations. A configuration is *frozen* if it is an isolated vertex of $\mathcal{R}$. In other words, no move is possible from this configuration.

Diameter. The structural problematic related to SHORTEST TRANSFORMATION is the *diameter* problematic.

DIAMETER
Given a source problem Π and an instance $\mathcal{I}$ of Π, what is the maximum length of a shortest

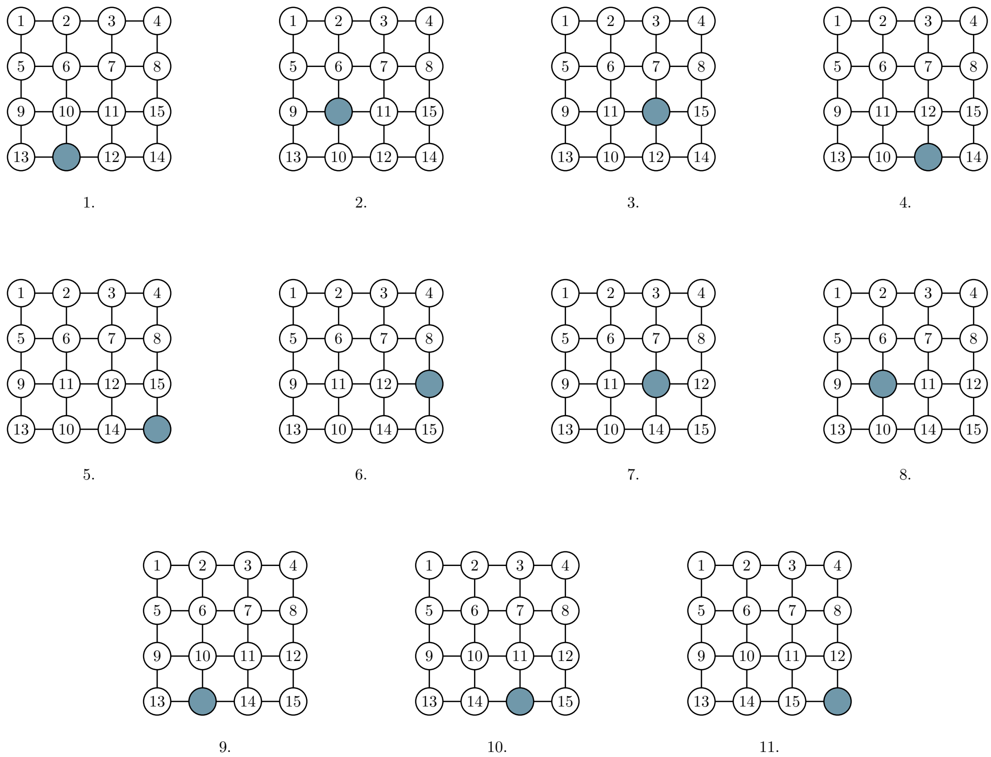


Figure 3.3: A sequence that solves the 15-puzzle in 10 steps.

transformation between any two feasible solutions X_s and X_t ? In other words, what is the diameter of $\mathcal{R}$?

In the 15-puzzle, it is equivalent to asking how many tiles we need to slide at most to reach the target configuration, and the answer is 80 [BMFN99].

Other problematics. Other structural properties of the reconfiguration graph have been studied, such as its Hamiltonicity, which allows to visit each solution of the instance exactly once by performing successive moves on an initial configuration.

Recently, an optimization variant of REACHABILITY has been introduced. Given an optimization source problem and a configuration X_s, it consists in searching for the best solution among the ones reachable from X_s. It has been studied for the source problems INDEPENDENT SET [IMNS18] and DOMINATING SET [BMOS19]. This method allows to search for the best possible solution while only performing local changes on a current solution. This is the principe of local search, a heuristic method that approximates optimization problems by only performing local changes, until a local optimum is reached. For more details about local search, the reader is refered to [LMS03].

2 Reconfiguration of tokens

Similarly to the 15-puzzle, a lot of reconfiguration problems involve the positions of some tokens in a graph $G = (V, E)$. For example, in the reconfiguration of INDEPENDENT SET, a feasible solution is an independent set of G, and we can represent this independent set by placing unlabeled tokens on the vertices that belong to it. Thus, the reconfiguration of the independent sets is equivalent to the reconfiguration of the positions of the tokens in the graph.

2.1 Adjacency rules

We review here the most common adjacency rules among the token reconfiguration problems.

Token Sliding. The first adjacency rule we mention is the one of the 15-puzzle, called *token sliding*, also denoted by TS. A move under TS consists in sliding a token along an edge of the graph. For the INDEPENDENT SET reconfiguration, it means that two configurations I and I' of $\mathcal{R}$ are adjacent if and only if there exist two vertices $u, v \in V$ such that $uv \in E$, and $I' = (I \cup v) \setminus u$. We denote this move by $u \rightsquigarrow v$. Note that the number of tokens remains the same after a move. Thus, we only study the reconfiguration graphs where the number of tokens is a given k. We then denote the reconfiguration graph by $\mathcal{R}_k$. Note that in the reconfiguration of INDEPENDENT SET under TS, the graph G plays a double role, as it both indicates which are the feasible solutions, and which are the allowed moves.

Token Jumping. To get rid of this last constraint, one can allow to move a token without sliding it along an edge of G. This corresponds to the *token jumping* rule, also denoted by TJ, where a move consists in changing the position of one token from a vertex to any other one. Thus, in INDEPENDENT SET reconfiguration, two configurations I and I' are adjacent if and only if there exist two vertices $u, v \in V$ such that $I' = (I \cup v) \setminus u$. Again, the number of tokens remains the same, and we denote by $\mathcal{R}_k$ the reconfiguration graph in which the feasible solutions have size exactly k. Note that in the 15-puzzle, allowing this kind of move is not of great interest. Indeed, the edges of G do not have any impact on the feasible solutions and are only useful to describe the allowed moves. Thus, the 15-puzzle with the token jumping rule is equivalent to the reconfiguration of permutations of the integers from 1 to 16, where the move is a transposition that involves 16 (which in the 15-puzzle represents the absence of token). It is easily seen in this case that REACHABILITY is always TRUE and an optimal algorithm consists in successively applying a transposition between 16 and each misplaced integer (thus, here, applying only changes on the elements of the symmetric difference gives a shortest transformation). Note that TS is a more constrained rule than TJ. Indeed, if a move can be performed under TS then it can also be performed under TJ, and thus if a configuration is reachable from another under TS, it also is under TJ.

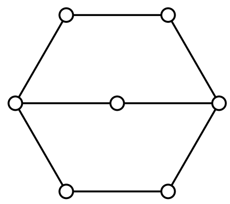
Token Addition-Removal. In the case of INDEPENDENT SET reconfiguration, since the tokens are not labeled, a move under TJ can be described as the removal of a token then the addition of a token on any other vertex. In the *token addition-removal* rule, also denoted by TAR, a move can be any of these two operations. Thus, in INDEPENDENT SET reconfiguration, two configurations I and I' are adjacent if and only if there exists a vertex $v \in V$ such that $I' = I \cup v$ or $I' = I \setminus v$. Note that in this case, the number of tokens is necessarily changing and thus the feasible solutions cannot be restricted to the ones of a given size. But if every independent set of G is a feasible solution, then the reconfiguration problems become trivial. For example, to reach an independent set from another, one can always remove every vertex of the first one then add every vertex of the second, while maintaining an independent set of G. Thus, in general, we are only interested in the independent sets of G of size at least a given k. More generally, when the optimization source problem is a maximization problem, we only consider feasible solutions of size at least a given k, and when it is a minimization problem, such as VERTEX COVER, we consider feasible solutions of size at most k. The reconfiguration graph is then denoted by $\mathcal{R}_k$. The value of k is called the *threshold* and is the object of many studies. For example, it is interesting to determine for which values of k the graph $\mathcal{R}_k$ is connected, as we do in Chapter 5 for the reconfiguration of DOMINATING SET. Note that if a move can be applied under TJ, where the feasible solutions are of size k, then it can be applied under TAR, where the solutions have size at most $k+1$ if the source problem is a minimization problem and at least $k-1$ if it is a maximization problem.

2.2 Generalizations of the 15-puzzle

The 15-puzzle was studied very early by mathematicians. In 1879, Johnson proved that the reconfiguration graph for this problem is not connected [JS+79]. Then, some generalizations were studied, beginning with the replacement of the instance into any $p \times p$ grid. In this case, the reconfiguration graph is disconnected [Wil74], the diameter of the connected components is less than $5p^3$ [Par95], and the shortest transformation problem is NP-complete [RW86]. The 15-puzzle was then generalized to any graph instance, in which the connectivity is characterized, as stated in the following theorem.

Theorem. *[Wil74] Consider the generalization of the 15-puzzle played on any graph G. The reconfiguration graph $\mathcal{R}$ is connected if and only if G is 2-connected, and it is not one of the following exceptions:*

- *bipartite graphs*
- *cycles of order $n \geq 4$*
- *the following graph:*



The fact that if G is disconnected then $\mathcal{R}$ is disconnected is straightforward when noticing that a token cannot change the connected component of G it belongs to by sliding along edges. Similarly, if G is 1-connected but not 2-connected, then there exists a vertex $v \in V$ such that the subgraph of G induced by $V \setminus v$ is disconnected, and one can easily see that a token cannot change the connected component of this subgraph it belongs to. When G is a cycle, only cyclic permutations can be applied to the positions of the tokens on the non-empty vertices, which implies that $\mathcal{R}$ has $(n-2)!$ connected components. The proof that G is disconnected for the other exceptions is less straightforward. For this type of generalization, it is also proved that the diameter is of order $\mathcal{O}(n^3)$ [Spi84], the reachability problem is in P [Spi84], and the shortest transformation problem is still NP-complete [Gol11]. Particular classes of graphs were also studied, such as trees, for which reachability is linear [AMPP99]. Note that the case of complete graphs is simlar to the 15-puzzle under TJ, since we can move any token to the empty vertex. Thus, it is equivalent to the reconfiguration of permutations, where a move consists in a transposition involving a given integer. So $\mathcal{R}$ is connected, and the shortest transformation problem is in P.

Another generalization of the 15-puzzle problem consists in playing with indistinguishable tokens. If the tokens are all pairwise indistinguishable, then solving the 15-puzzle is equivalent to finding a path from the empty vertex of the initial configuration to the one of the target configuration in the grid. Thus, $\mathcal{R}$ is connected, and the shortest sequence between any two configurations corresponds to the shortest path between the two vertices in the grid, and the same property actually holds on any instance that is a connected graph. Thus, we study a generalization where the tokens have labels, but two tokens can have the same label, which we then call colors. Moreover, we can also have several empty vertices, and the instance can be any graph G. We denote such a problem by $puz(G, k_1, \ldots, k_\ell)$, where G is the instance, and where we play with $k_i \geq 1$ tokens of color i for any integer i such that $1 \leq i \leq \ell$, with $k_1 + \ldots + k_\ell \leq n-1$. Brightwell, Van den Heuvel and Trakultraipruk proved that the connectivity problematic is always in P by giving a full characterization of the connected components of $\mathcal{R}$ depending on the values k_i [vdH13], and Goldreich proved that the shortest transformation problem is in P when the tokens all have the same color, i.e. when $\ell = 1$, and that it is NP-complete otherwise [Gol11]. The case where $k_1 = 1$ and $\ell = 2$ has been studied to model robot motion [PRST94], where the token of color 1 represents a robot, the tokens of color 2 represent

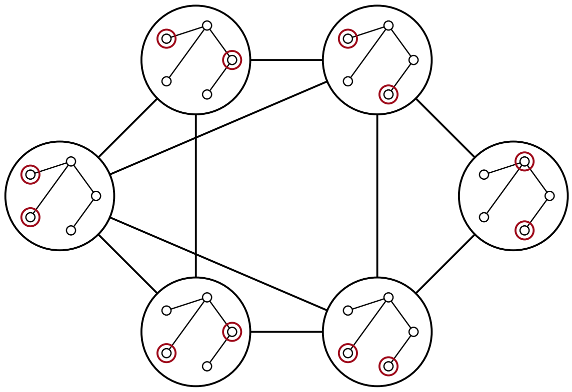


Figure 3.4: The reconfiguration graph $\mathcal{R}_2$ of a graph for the INDEPENDENT SET reconfiguration problem under TJ.

obstacles that can be moved, and the goal is to move the robot from a vertex to a different one with the shortest possible sequence while avoiding the obstacles and eventually moving them. The authors found an efficient algorithm in the case where G is a tree. Many other variants of robot motion were then studied, with for example multiple robots without obstacles [Sur09], or the popular game Sokoban, where the obstacles can only be pushed by the robot, and for which REACHABILITY is PSPACE-complete [Cul97].

2.3 Independent Set Reconfiguration and Vertex Cover Reconfiguration

The INDEPENDENT SET reconfiguration problem can somehow be seen as a generalization of the 15-puzzle, where the tokens are all indistinguishable, the instance is a graph G, and the feasible solutions are independent sets of G. It has been studied under the token sliding, token jumping and token addition-removal rules. Actually, Kaminski et al. proved the following.

Theorem. *[KMM12] There exists an independent set reconfiguration sequence between two configurations of k tokens under TJ if and only if there exists an independent set reconfiguration sequence between them under TAR where the feasible solutions have size at least $k-1$.*

We say that TJ and TAR are *equivalent* for the reconfiguration of independent sets.

In the VERTEX COVER reconfiguration problem, the feasible solutions are the vertex covers of G of size k, or at most k when working under TAR. When it comes to classical complexity result, one can easily check that the reconfigurations of VERTEX COVER and INDEPENDENT SET are actually equivalent. Indeed, since $V \setminus C$ is an independent set of G if and only if C is a vertex cover of G, any reconfiguration sequence of independent sets can be seen as a reconfiguration sequence of vertex covers, where the tokens are placed on the empty vertices.

Token Jumping. We give here some results on the reconfiguration of independent sets under TJ (or, equivalently, under TAR). An example of reconfiguration graph is given in Figure 3.4.

Ito et al. proved that under TJ, REACHABILITY is PSPACE-complete [IDH+11]. They actually studied the case $k = \alpha(G)$, i.e. the case where the independent sets are maximum. Let us define the following decision problems.

MAXIMUM INDEPENDENT SET RECONFIGURATION UNDER TOKEN JUMPING (maxISR$_{\text{TJ}}$)
Input: A graph G, two maximum independent sets I_s and I_t of G
Output: Does there exist an independent set reconfiguration sequence from I_s to I_t with the token jumping rule ?

MINIMUM VERTEX COVER RECONFIGURATION UNDER TOKEN JUMPING (minVCR$_{\text{TJ}}$)
Input: A graph G, two minimum vertex covers C_s and C_t of G
Output: Does there exist a vertex cover reconfiguration sequence from C_s to C_t with the token jumping rule ?

They proved that maxISR$_{\text{TJ}}$, and thus minVCR$_{\text{TJ}}$, is PSPACE-complete. They also restricted this result to planar graphs of maximum degree at most 3.

Theorem. *[IDH+11] max*ISR$_{\text{TJ}}$ *(or, equivalently, min*VCR$_{\text{TJ}}$*) is* PSPACE*-complete in planar graphs of maximum degree at most 3.*

We use this result for several reductions in Chapter 6. Many other complexity results have been obtained for particular graph classes. Some of these results are presented in Table 3.1. Concerning shortest transformation, the problem is PSPACE-complete [KMM12].

Token Sliding. We review now some results with the TS rule. An example of reconfiguration graph is given in Figure 3.5.

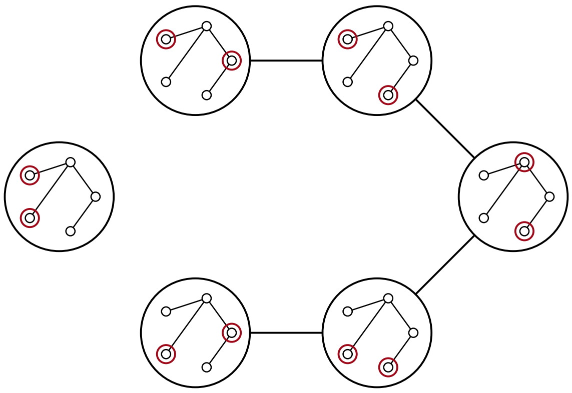


Figure 3.5: The reconfiguration graph $\mathcal{R}_2$ of a graph for the INDEPENDENT SET reconfiguration problem under TS.

Hearn and Demaine proved that REACHABILITY is also PSPACE-complete under TS [HD05], and it is still true when restricted to planar graphs of maximum degree at most 3 [KMM12]. Other results on REACHABILITY concerning particular instances are presented in Table 3.1. The shortest transformation problem is also PSPACE-complete under TS [KMM12], even if polynomial time algorithms exist for particular classes of graphs [YU16].

2.4 Dominating Set Reonfiguration

In the DOMINATING SET reconfiguration problem, the feasible solutions are dominating sets of a graph G. It has been studied mostly under the TAR rule [HIM+16, HS14, SMN15, HS17, MTR19b], although the TS rule was investigated recently [BDO19].

Token Addition-Removal. We present here some results on the reconfiguration of dominating sets under the TAR rule. A reconfiguration graph is given in Figure 3.6.

Class	TJ	TS
Any graph	PSPACE-complete [IDH+11]	PSPACE-complete [HD05]
Planar with $\Delta(G) \geq 3$	PSPACE-complete [KMM12]	PSPACE-complete [KMM12]
Even hole-free	P [KMM12, INZ16, MNR14]	unknown
Tree	P because even-hole free	P [DDFE+15]
Bounded treewidth	PSPACE-complete [Wro18]	PSPACE-complete [Wro18]
Bounded pathwidth	PSPACE-complete [Wro18]	PSPACE-complete [Wro18]
Bipartite	PSPACE-complete [LM18]	NP-complete [LM18]
Chordal	P because even-hole free	PSPACE-complete [BKL+20]
Interval	P because even-hole free	P [BB17]

Table 3.1: The complexity of REACHABILITY in the reconfiguration of independent sets under TJ and TS.

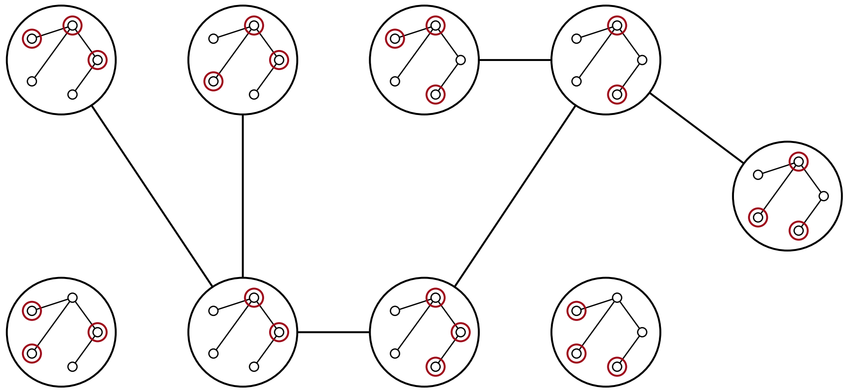


Figure 3.6: The reconfiguration graph $\mathcal{R}_3$ of a graph for the DOMINATING SET reconfiguration problem under TAR.

Haddadan et al. proved that the reachability problem is PSPACE-complete under the addition-removal rule, even when restricted to split graphs and bipartite graphs [HIM+16]. They also give a linear time algorithm in trees and interval graphs.

A lot of attention has been given to the connectivity problem. In particular, several studies have focused on the value d_0 of the threshold such that for any $k \geq d_0$, the reconfiguration graph $\mathcal{R}_k$ is connected. Haas and Seyffarth proved that if G has at least two independent edges, then $d_0 \leq \min\{n-1, \Gamma(G)+\gamma(G)\}$, where $\Gamma(G)$ is the maximum size of an inclusion-wise minimal dominating set of G [HS14]. They also showed that $d_0 \leq \Gamma(G)+1$ for bipartite and chordal graphs, then asked if this is true in any graph. Suzuki et al. proved that it is not the case, by constructing an infinite family of graphs such that $\mathcal{R}_{\Gamma(G)+1}(G)$ is not connected [SMN15]. One of these graphs is planar, thus implying the following result, which we use in Chapter 5:

Theorem. *[SMN15] There exists a planar graph G such that $\mathcal{R}_{\Gamma(G)+1}(G)$ is not connected.*

Mynhardt et al. improved the result of Suzuki et al. by showing that there exists an infinite family of graphs G for which $d_0 = 2\Gamma(G)-1$ [MTR19a]. On the positive side, Haas and Seyffarth proved the following:

Theorem. *[HS17] If $k = \Gamma(G) + \alpha(G) - 1$, then $\mathcal{R}_k(G)$ is connected.*

In Chapter 5, we show that $\mathcal{R}_k(G)$ has moreover a linear diameter in this case. We also give other upper bounds on d_0 for specific graph classes.

Token Sliding. Recently, Bonamy et al. investigated the reconfiguration of dominating sets under TS [BDO19]. An example of reconfiguration graph is given in Figure 3.7.

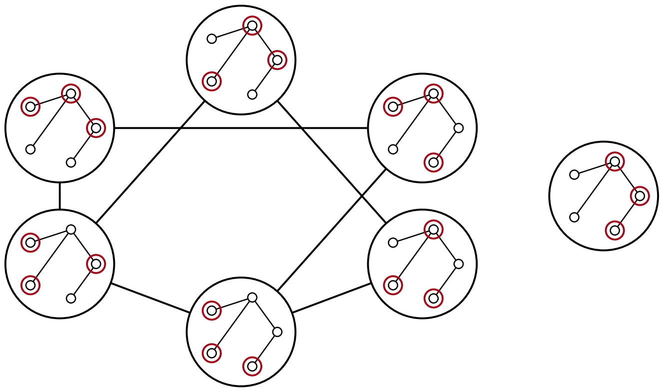


Figure 3.7: The reconfiguration graph $\mathcal{R}_3$ of a graph for the DOMINATING SET reconfiguration problem under TS.

Bonamy et al. studied the complexity of the reachability problem. We denote this problem by DSR_{TS}. In other words, DSR_{TS} is defined as follows.

DOMINATING SET RECONFIGURATION UNDER TOKEN SLIDING (DSR_{TS})
Input: A graph G, two dominating sets D_s and D_t of G
Output: Does there exist a dominating set reconfiguration sequence from D_s to D_t with the token sliding rule ?

Bonamy et al. proved that DSR_{TS} is PSPACE-complete, even when restricted to split, bipartite or bounded treewidth graphs [BDO19]. They also give a polynomial time algorithm for cographs and for dually chordal graphs. In particular, their results implies the following:

Theorem. *[BDO19] Let G be an interval graph, and D_s, D_t be two dominating sets of G of the same size. There always exist a reconfiguration sequence under TS from D_s to D_t.*

We use this result in Chapter 6, as we investigate the complexity of DSR_{TS} in other graph classes such as planar bipartite graphs, unit disk graphs, circle graphs, line graphs and circular arc graphs.

3 Other reconfiguration problems

The formulation of every reconfiguration problem mentioned until now involves the position of tokens on a graph. In this section, we are interested in different kinds of reconfiguration problems, such as COLORING reconfiguration and SATISFIABILITY reconfiguration, as well as the reconfiguration of graphs with the same degree sequence.

3.1 Coloring Reconfiguration

The COLORING reconfiguration problem is one of the most studied reconfiguration problems, partly because it has important applications. In this problem, the instance is a graph G and the feasible solutions are the proper colorings of G with k colors, where k is a given integer. The reconfiguration graph is then denoted by $\mathcal{R}_k$. Two adjacency rules have mostly been studied: the Glauber dynamics, and the Kempe dynamics.

Glauber dynamics. Under the Glauber dynamics, two colorings C_1 and C_2 are adjacent if and only if exactly one vertex has a different color in C_1 and C_2. This kind of dynamics was first used in statistical physics, in an algorithm designed by Roy J. Glauber [Mar99]. In this application, the colorings represent the interacting spins of a ferromagnetic lattice, using the Ising model. Glauber's algorithm represents the possible evolution of the spins through time. It belongs to the family of Markov Chain Monte Carlo algorithm, which are commonly used to sample the possibles states of a variable. To do so, some transitions between the states are allowed, with different probabilities, and we perform a random walk on these states. The *stationnary* distribution then gives the probability to end up on each state after an infinite walk. For the stationnary distribution to be uniform, we must be able to reach any state from any other one in a walk. This is possible if and only if the reconfiguration graph, whose vertices are the states and where two states are adjacent if and only if there is a non zero probability to go from one to another in a walk, is connected. The number of steps needed to reach a distribution close to stationnary is called the *mixing time*. Since we want to sample as efficiently as possible, we want to evaluate this mixing time. The diameter of the reconfiguration graph then gives a lower bound. For more details about sampling through Monte-Carlo Markov chains, the reader is referred to [Jer03].

The reconfiguration graph $\mathcal{R}_3$ under Glauber dynamics for the graph P_3 is illustrated in Figure 3.8.

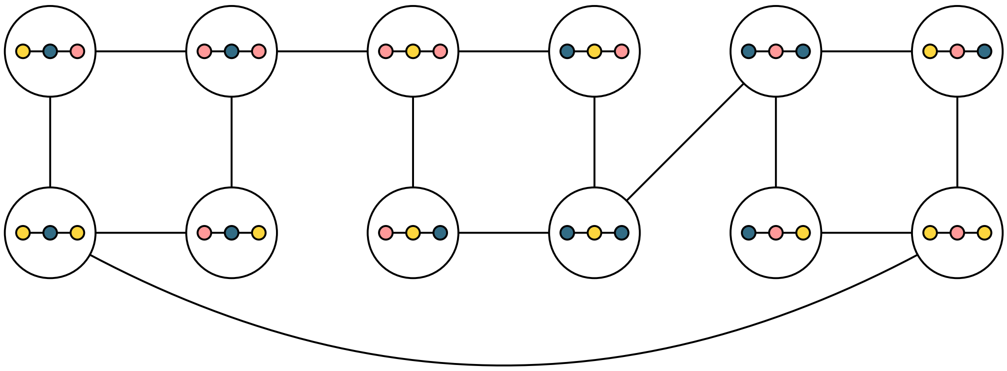


Figure 3.8: The reconfiguration graph $\mathcal{R}_3$ of P_3 under the Glauber dynamics rule.

Similarly to the token addition-removal rule in the reconfiguration of tokens on a graph, the value of k has a great impact. For example, if $k = n$ then we can reach any coloring from any other one by reaching the same canonical coloring from both, where every vertex has a different color. On the other hand, if $k = 2$, only the colors of isolated vertices can be changed while maintaining a proper coloring, and thus a coloring C_2 is reachable from a coloring C_1 if and only if C_1 and C_2 only differ on isolated vertices, and $\mathcal{R}_k$ is disconnected unless G is an empty graph. Thus, most of the work on COLORING reconfiguration studies the properties of $\mathcal{R}_k$ depending on the value of k.

The connectivity problematic has been extensively studied. In relation to the previously mentioned application to Markov Chains, we say that a graph G such that its reconfiguration graph $\mathcal{R}_k$ is connected is *k-mixing*. A lot of studies focus on the relationship between k and the *degeneracy* of G, which is the maximum, over all induced subgraphs H of G, of the minimum degree of H, denoted by $deg(G)$. Firstly, if $k \geq deg(G) + 2$, then G is k-mixing [DFFV06], and Cereceda conjectured the following in his thesis [Cer07].

Conjecture. *(Cereceda, [Cer07]) If $k \geq deg(G) + 2$ then the diameter of $\mathcal{R}_k$ is in $\mathcal{O}(n^2)$.*

To support this conjecture, Cereceda proved it under the weaker assumptions $k \geq \Delta(G)+2$ (note that $deg(G) \leq \Delta(G)$) and $k \geq 2deg(G) + 1$, improved recently by Bousquet and Heinrich into $k \geq \frac{3}{2}(deg(G) + 1)$ [BH19]. In 2011, Cereceda et al. proved that the diameter of each connected component of $\mathcal{R}_3$ is in $\mathcal{O}(n^2)$ [CVDHJ11]. It is also proved on specific graph classes such as chordal graphs [BJL+14], bounded treewidth graphs [BB18] or bipartite planar graphs [BH19]. A more general result towards Cereceda's conjecture is the one of Bousquet and Heinrich who gave the first proof of a polynomial diameter in 2019, proving that if $k \geq deg(G)+2$ then $\mathcal{R}_k$ has diameter $\mathcal{O}(n^{deg(G)+1})$ [BH19]. Note that if Cereceda's conjecture is true, then it is somehow the best possible, as the diameter of $\mathcal{R}_k$ is quadratic when G is a path [BJL+14]. A related conjecture of Bartier and Bousquet states that the diameter of $\mathcal{R}$ is linear if $k \geq deg(G)+3$ [BB19], as it is the case when $k \geq 2deg(G)+2$ [BP16].

Other studies focused on the complexity of CONNECTIVITY and REACHABILITY. As previously mentioned, for $k=2$, REACHABILITY and CONNECTIVITY are in P. Cereceda et al. proved that for $k=3$, REACHABILITY is in P [CVDHJ11], while for $k \geq 4$, it is PSPACE-complete [BC09]. On the other hand, the connectivity problem is co-NP-complete for $k=3$ [CVdHJ09], and the complexity is still unknown for $k \geq 4$, even though the complexity of REACHABILITY gives a hint. Since the problems related to coloring are generally easy on bipartite graphs, for which the chromatic number is 2, these complexity questions have been studied in this graph class, but surprisingly, it does not make the problems easier [CVdHJ09]. On the other hand, in planar graphs, for $k=3$ the connectivity problem is in P[CVdHJ09].

Kempe dynamics. Under Kempe dynamics, a move consists in the exchange of two colours in a set of vertices that are each colored with one of the two colors and that induces a connected subgraph of G, which we call a *Kempe chain*. The idea is that if we want to change the color of a vertex v from i to j but there are neighbors of v that have color j, in order to keep a proper coloring, we can change the color of these neighbors from j to i. But if they have neighbors distinct from v that have color i, we also need to change their color to j, and we continue this process until no color has to change anymore for the coloring to be proper. Note that if we change the color of a single vertex while maintaining a proper coloring, then this vertex is a Kempe chain in which we exchange the former and new color, and thus if a coloring is reachable from another under Glauber dynamics, then it also is under Kempe dynamics. The reconfiguration graph of P_3 under Kempe dynamics for $k=3$ is represented in Figure 3.9, and we can see that every edge in Figure 3.8 is also an edge in Figure 3.9. The concept of Kempe chains was introduced by Kempe in his attempt of proving the Four Color Theorem [Kem79] and used later in the proofs of several important results concerning coloring such as Vizing's edge-colouring Theorem [Viz64].

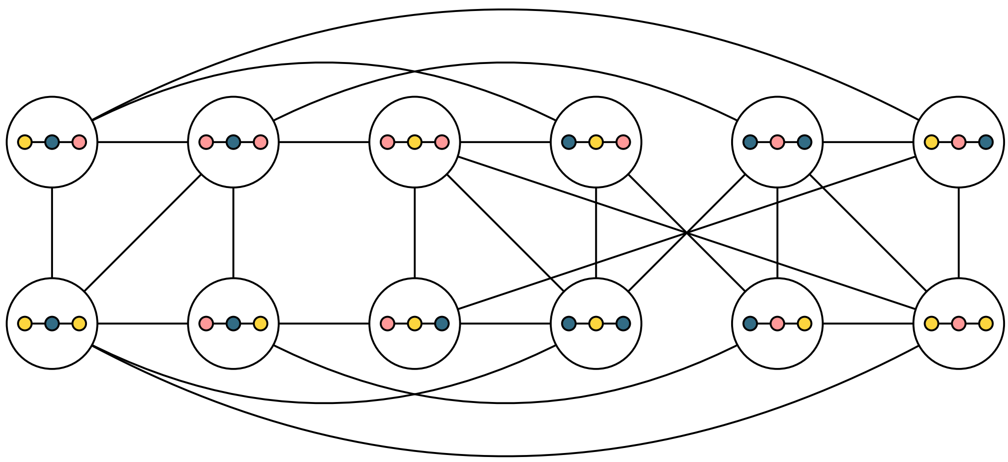


Figure 3.9: The reconfiguration graph $\mathcal{R}_3$ of P_3 under the Kempe dynamics rule

Even though recolorings under Kempe dynamics were studied before recolorings under Glauber dynamics, there are significantly less results about COLORING reconfiguration under Kempe dynamics than under Glauber dynamics. Most of the studies focused on the conditions that ensure that a graph G is k-*Kempe mixing*, i.e. such that the reconfiguration graph $\mathcal{R}_k$ under Kempe dynamics is connected.

Since any move under Glauber dynamics is a move under Kempe dynamics, if G is k-mixing then it is k-Kempe-mixing. Thus, if $k \geq deg(G)+2$ then G is k-Kempe-mixing, and Las Vergnas and Meyniel actually proved it for $k \geq deg(G)+1$ [LVM81]. But if G is k-Kempe mixing then G is not necessarily k-mixing, as proves the example of bipartite graphs that are k-Kempe-mixing for any $k \geq 2$ while there exist bipartite graphs of any order n that are not k-mixing [CVDHJ08]. Other results on connectivity have been obtained for particular graphs, such as planar graphs that are 5-Kempe-mixing [Mey78], or k-regular graphs that are k-Kempe mixing for any $k \geq 3$ except for the triangular prism and the triangle K_3 that are not [FJP15, BBFJ19].

3.2 Satisfiability Reconfiguration

For a given instance F which is a Boolean formula of variables $x_1, \ldots, x_n$, the feasible solutions of SATISFIABILITY reconfiguration are the assignments of the variables that satisfy F, and two assignments are adjacent if and only if they differ by the value of exactly one variable. The reconfiguration graph $\mathcal{R}$ obtained with the formula $F = (\overline{x_1} \vee \overline{x_2} \vee x_3) \wedge x_1 \wedge (x_1 \vee x_2) \wedge (x_1 \vee x_2 \vee \overline{x_4})$ is illustrated in Figure 3.10, where each bit represents in this order the values of x_1, x_2, x_3 and x_4.

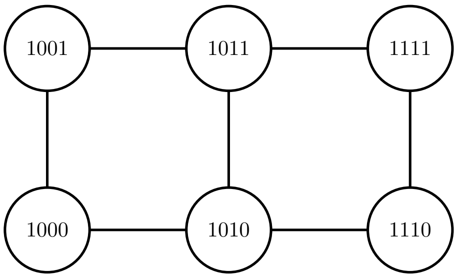


Figure 3.10: An example of reconfiguration graph $\mathcal{R}$ of the SATISFIABILITY reconfiguration problem.

In 1978, Schaefer proved a dichotomy result for the complexity of SATISFIABILITY [Sch78]. He expressed any formula through a Boolean relation and gave necessary and sufficient conditions on this relation for SATISFIABILITY to be in P, while it is NP-complete otherwise. The relations that respect these conditions are now called *Schaefer relations*. We do not detail their definition here but it can be found in [Sch78].

Gopalan et al. studied the reachability problematic for the source problem SATISFIABILITY. It is defined as follows.

SATISFIABILITY RECONFIGURATION (SATR)
Input: A CNF Boolean formula F, two variable assignments A_s and A_t that satisfy F
Output: Does there exist a reconfiguration sequence from A_s to A_t that maintains F satisfied, where the operation consists in a *variable flip*, i.e. a change of the assignment of one variable from $x = 0$ to $x = 1$, or conversely ?

They gave the following complexity result.

Theorem. *[GKMP09] SATR is PSPACE-complete.*

They also obtained a dichotomy result similar to the one of Schaefer, corrected by Schwerdtfeger [Sch12]. They provide necessary and sufficient conditions on the relation for REACHABILITY to be in P. These conditions define the set of *tight relations*. The set of tight relations contains the set of Schaefer relations. They also studied the connectivity and diameter problematics and proved that the connectivity problem is in co-NP for tight relations, with a linear diameter, and PSPACE-complete otherwise with exponential diameter.

The shortest transformation problem was studied by Mouawad et al. in 2017. They obtained a trichotomy by introducing the set of *navigable relations*. The set of tight relations contains the set of navigable relations. They proved that the shortest transformation problem is in P for navigable relations, NP-complete for tight but not navigable relations, and PSPACE-complete otherwise [MNPR17].

3.3 Reconfiguration of graphs with the same degree sequence

The *degree sequence* of a multigraph G is the sequence of the degrees of its vertices in a non-increasing order. Given a non-increasing sequence of integers $S := (d_1, \ldots, d_n)$ and a multugraph $G = (V, E)$ whose vertices are labeled as $V = \{v_1, \ldots, v_n\}$, we say that G *realizes* S if $d(v_i) = d_i$ for all $i \leq n$. Senior gave necessary and sufficient conditions ensuring, given a sequence of integers S, that there exists a connected multigraph realizing S [Sen51b]. Hakimi designed an algorithm that outputs such a multigraph if it exists and returns FALSE otherwise, in polynomial time [Hak62b].

We study here the reconfiguration of multigraphs with the same degree sequence, where the feasible solutions are the multigraphs that realize a given degree sequence (this degree sequence is then the instance of the problem). The adjacency rule we study is called the flip rule. A *flip* on two disjoint edges ab and cd consists in deleting the edges ab and cd and creating the edges ac and bd (or ad and bc). One can easily see that applying a flip on G does not modify its degree sequence. An example of reconfiguration graph is given in Figure 3.11.

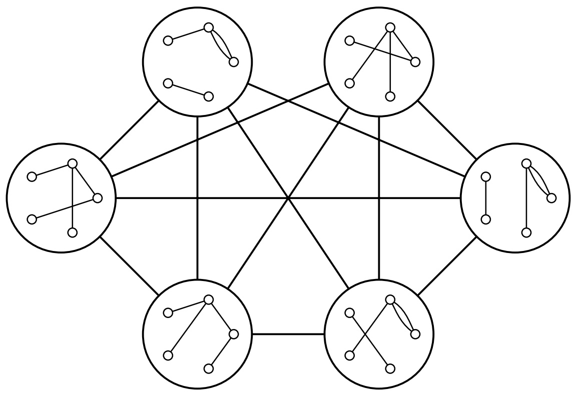


Figure 3.11: The reconfiguration graph $\mathcal{R}$ of the degree sequence $(3, 2, 1, 1, 1)$.

Hakimi proved that for any degree sequence S, if there exists a multigraph realizing S then $\mathcal{R}$ is connected [Hak63b]. Taylor also showed that the reconfiguration graph remains connected when we restrict the feasible solutions to connected multigraphs, simple graphs, and simple connected graphs [Tay81b].

On the other hand, Caprara showed that the shortest transformation problem is NP-hard [Cap97a]. Will however gave the following formula, where $\delta(G, H)$ is the number of edges in the symmetric difference of G and H (i.e. the edges that belong to G but not to H or conversely), and $mnc(G, H)$ is the maximum number of alternating cycles in which this symmetric difference can be partitioned (an alternating cycle is a cycle that alternates between edges of G and edges of H).

Theorem. *[Wil99a] Let G and H be two multigraphs with the same degree sequence. A shortest sequence of flips that transforms G into H has length exactly* $\frac{\delta(G,H)}{2} - mnc(G, H)$.

When restricted to connected multigraphs, the shortest transformation problem can be formulated as follows.

SHORTEST CONNECTED GRAPH TRANSFORMATION (SCGT)
Input: Two connected multigraphs G, H with the same degree sequence.
Output: The minimum number of flips needed to transform G into H while maintaining connectivity.

Bousquet and Mary provide a 4-approximation algorithm for SCGT in [BM18a]. In Chapter 4, we improve this approximation ratio into $\frac{5}{2}$. This problem has applications in chemistry and bioinformatics. We give more details about these applications in Chapter 4.

Chapter 4

Reconfiguration of Graphs with the same Degree Sequence

1 Introduction

In this chapter, we study the reconfiguration of graphs with the same degree sequence. Recall that the degree sequence of a graph G is the sequence of the degrees of its vertices in non-increasing order. Given a non-increasing sequence of integers $S = \{d_1, \ldots, d_n\}$, we say that a graph $G = (V, E)$ whose vertices are labeled as $V = \{v_1, \ldots, v_n\}$ realizes S if $d(v_i) = d_i$ for all $i \leq n$.

The problem of the realization of a given degree sequence has been studied in the early days of graph theory. In 1951, Senior gave necessary and sufficient conditions to guarantee that, given a sequence of integers $S = \{d_1, \ldots, d_n\}$, there exists a connected multigraph realizing S [Sen51a]. Hakimi then proposed a polynomial time algorithm that outputs a connected (multi)graph realizing S if such a graph exists or returns no otherwise [Hak62a]. This algorithm is based on the notion of flips, also called swap or switch in the literature. A flip on two disjoint edges ab and cd consists in deleting the edges ab and cd and creating the edges ac and bd (or ad and bc) [1].

In the framework of reconfiguration of graphs with the same degree sequence, the instance is a degree sequence S, the feasible solutions are the multigraphs realizing S, and the adjacency rule is such that two graphs that realize S are adjacent if they differ by a flip. The reconfiguration of graphs with the same degree sequence has been widely studied. One of the reasons is that it can be used to randomly sample the graphs that realize a given degree sequence [DGKTB10, MR95]. This is extremely useful in applications on real networks, for instance in epidemiology, when the collected data only contains the degree sequence of a graph, and we want to retrieve the graph. As it was the case in the application of the reconfiguration of graph colorings to statistical physics mentionned in Chapter 3, the properties of the reconfiguration graph such as its connectivity and diameter can be useful tools for sampling. In a second paper, Hakimi proved that for any non-increasing sequence S, if the reconfiguration graph is not empty then it is connected [Hak63a], which makes uniform sampling possible.

The shortest transformation problem was then raised. In 1999, Will [Wil99b] gave an explicit formula for the shortest transformation between two graphs that realize S. To do so, he uses the notion of symmetric difference. Since the problem is originally defined for multigraphs, to define the symmetric difference, we need to define the intersection, union, and difference of two multigraphs. The *intersection* of two multigraphs G and H on the same set of vertices V is the graph $G \cap H$ with vertex set V, and such that $e \in E(G \cap H)$, with multiplicity m, if the minimum multiplicity of e in both graphs is m. Their *union*, $G \cup H$, has vertex set V, and $e \in E(G \cup H)$, with multiplicity m, if and only if the maximum multiplicity of e in G and H is m. Finally, the *difference* $G - H$ has vertex set V and $e \in E(G - H)$ with multiplicity m if and

[1] In the case of multigraphs, we simply decrease by one the multiplicities of ab and cd and increase by one the ones of ac and bd.

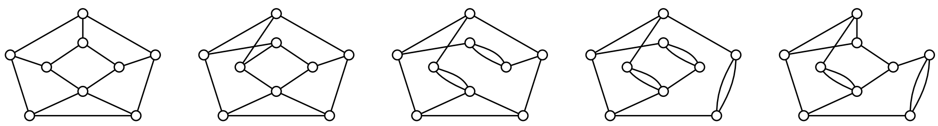
only if the difference between its multiplicities in G and H is $m > 0$. The *symmetric difference* of G and H is $\Delta(G, H) = (G - H) \cup (H - G)$. We denote by $\delta(G, H)$ the number of edges of $\Delta(G, H)$.

Note that each flip removes at most 4 edges of the symmetric difference. Therefore, it is straightforward that the length of a transformation from G to H is at least $\frac{\delta(G,H)}{4}$. To obtain the exact formula of Will, we also need the following notions. A cycle C in $\Delta(G, H)$ is *alternating* if edges of G and H alternate in C. Since the number of edges of G incident to v is equal to the number of edge of H incident to v in $\Delta(G, H)$, the graph $\Delta(G, H)$ can be partitioned into a collection of alternating cycles. We denote by $mnc(G, H)$ the maximal number of cycles in a partition $\mathcal{C}$ of $\Delta(G, H)$ into alternating cycles. The formula of Will is the following.

Theorem 4.1. *[Wil99b] Let G, H be two graphs with the same degree sequence. A shortest sequence of flips that transforms G into H (that does not necessarily maintain the connectivity of the intermediate graphs) has length exactly $\frac{\delta(G,H)}{2} - mnc(G, H)$.*

This formula was reproved in 2013 by Erdős, Király and Miklós [EKM13], then in 2017 by Bereg and Ito [BI17]. Bereg and Ito also proved that computing $mnc(G, H)$ is NP-hard, which implies that the shortest transformation problem is NP-hard for the reconfiguration of graphs with the same degree sequence. They however provide a polynomial time 1.5-approximation algorithm.

Another famous application of the reconfiguration of graphs with the same degree sequence concerns mass spectromerty [Hak62a]. Mass spectrometry is a technique used by chemists in order to obtain the formula of a molecule. It provides the mass-to-charge (m/z) ratio spectrum of the molecule from which we can deduce how many atoms of each element the molecule has. With this formula, we would like to find out the nature of the molecule, i.e. the bonds between the different atoms. But the existence of structural isomers points out that there could exist several solutions for this problem. Thus, we would like to find all of them. Since the valence of each atom is known, this problem actually consists in finding all the connected loop-free multigraphs whose degree sequence is the sequence of the valences of those atoms. The reconfiguration problem we are studying here can be a tool for an enumeration algorithm consisting in visiting the reconfiguration graph. Note that in this case, since we are trying to retrieve the formula of only one molecule, the multigraphs that realize S should have the additional constraint of being connected. In this chapter, we focus on the reconfiguration problem where the feasible solutions are only the connected multigraphs. An example of a reconfiguration sequence is given in Figure 4.1.



G H

Figure 4.1: An example of reconfiguration sequence of connected multigraphs with the same degree sequence.

Taylor proved that the reconfiguration graph remains connected when we restrict the feasible solution to connected multigraphs, and it is also the case when we restrict them to simple graphs, and simple connected graphs [Tay81a].

Since we know that the reconfiguration graph is connected when the feasible solutions are connected multigraphs, we can then ask what is the minimum length of a reconfiguration sequence. This problem is known to be NP-hard, even when restricted to paths [Cap97b].

In the particular case of paths, the problem is equivalent to SORTING BY REVERSALS, which has been widely studied in the last twenty years in genomics. The reversal of a sequence of DNA is a common mutation of a genome, that can lead to major evolutionary events. It consists, given a DNA sequence that can be represented as a labeled path $x_1, \ldots, x_n$ on n vertices, in turning around a part of it. More formally, a *reversal* is a transformation that, given two integers $1 \leq i < j \leq n$, transforms the path $x_1, \ldots, x_n$ into $x_1, \ldots, x_{i-1}, x_j, x_{j-1}, \ldots, x_i, x_{j+1}, \ldots, x_n$. It is easy to prove that, given two paths on the same vertex set (and with the same leaves), there exists a sequence of reversals that transforms the first into the second. But biologists want to find the minimum number of reversals needed to transform a genome (i.e. a path) into another in order to compute the evolutionary distance between different species. An input of SORTING BY REVERSALS consists of two paths P, P' with the same vertex set (and the same leaves) and an integer k. The output is positive if and only if there exists a sequence of at most k reversals that transforms P into P'. Note that a reversal can be equivalently defined as follows: given a path P and two disjoint edges ab and cd of P, a reversal consists in the deletion of the edges ab and cd and the addition of ac and bd, that moreover keeps the connectivity of the graph. Indeed, when we transform $x_1, \ldots, x_n$ into $x_1, \ldots, x_{i-1}, x_j, x_{j-1}, \ldots, x_i, x_{j+1}, \ldots, x_n$, we have deleted the edges $x_{i-1}x_i$ and x_jx_{j+1} and have created the edges $x_{i-1}x_j$ and x_ix_{j+1}. Thus, it is easily seen that sorting by reversals is equivalent to finding a reconfiguration sequence from P to P' where the moves are flips, and the feasible solutions are paths. For this problem, Kececioglu and Sankoff proposed an algorithm that computes a sequence of reversals of size at most twice the length of an optimal solution in polynomial time [KS95]. Then, Christie improved it into a 1.5-approximation algorithm [Chr98]. The best polynomial time algorithm known so far is a 1.375-approximation due to Berman et al. [BHK02].

In this chapter, we study the following problem, which is a generalization of the SORTING BY REVERSALS problem to any connected multigraphs. In other words, it corresponds to the shortest transformation problematic for the reconfiguration of connected multigraphs realizing a given degree sequence.

SHORTEST CONNECTED GRAPH TRANSFORMATION (SCGT)
Input: Two connected multigraphs G, H with the same degree sequence
Output: The minimum number of flips needed to transform G into H while maintaining connectivity.

Since this problem is NP-hard [Cap97b], an approach consists in searching for polynomial time approximation algorithms. Bousquet and Mary proposed a 4-approximation algorithm [BM18b]. In this chapter, we present a result from a joint work with Nicolas Bousquet, in which we provide a 2.5-approximation algorithm. We mainly focus on the SHORTEST TREE TRANSFORMATION problem which is the same as SCGT except that the input consists of trees with the same degree sequence. Informally speaking, it is due to the fact that if an edge of the symmetric difference appears in some cycle, then we can reduce the size of the symmetric difference in one flip, as observed in [BM18b].

2 Preliminaries

Let us first introduce some notions we use in this chapter.

The flip operation that transforms the edges ab and cd into the edges ac and bd is denoted by $(ab, cd) \to (ac, bd)$. When the target edges are not important we simply say that we flip the edges ab and cd.

The *inverse* σ^{-1} of a flip σ is the flip such that $\sigma \circ \sigma^{-1} = id$, i.e. applying σ and then σ^{-1} leaves the initial graph. The *opposite* $-\sigma$ of a flip σ is the unique other flip that can be applied to the two edges of σ. If we consider a flip $\sigma = (ab, cd) \to (ac, bd)$, then $\sigma^{-1} = (ac, bd) \to (ab, cd)$ and $-\sigma = (ab, cd) \to (ad, bc)$. Note that $-\sigma$ is a flip deleting the same edges as σ while σ^{-1} cancels the flip σ. When we transform a graph G into another graph H, we can flip the edges of G or the edges of H. Indeed, applying the sequence of flips $(\sigma_1, \ldots, \sigma_i)$ to transform G into a graph K, and the sequence of flips $(\tau_1, \ldots, \tau_j)$ to transform H into K is equivalent to applying the sequence $(\sigma_1, \ldots, \sigma_i, \tau_j^{-1}, \ldots, \tau_1^{-1})$ to transform G into H.

Let $G = (V, E)$ be a connected graph and let H be a graph with the degree sequence of G. An edge e of G is *good* if it is in $G \cap H$ and is *bad* otherwise. Note that since G and H have the same degree sequence, the graph $\Delta(G, H)$ has even degree on each vertex and the number of edges of G incident to v is equal to the number of edges of H incident to v. A flip is *good* if it flips bad edges and creates at least one good edge. It is *bad* otherwise. A *connected flip* is a flip such that its resulting graph is connected. Otherwise, it is *disconnected*. A *path* from $a \in V$ to $b \in V$ is a sequence of vertices $(v_1, \ldots, v_k)$ such that $a = v_1$, $b = v_k$, for every integer $i \in [k-1]$, $v_i v_{i+1} \in E(G)$ and there is no repetition of vertices. Similarly, a path from e to f with $e, f \in E(G)$ is a path from an endpoint of e to an endpoint of f that does not contain the other endpoint of e and of f. A path between x and y (vertices or edges) is a path from x to y or a path from y to x. The *content* of a path is its set of vertices. We say that an edge e *belongs to* (or *is on*) a path P if both endpoints of e appear consecutively in P. The *intersection* $P_1 \cap P_2$ of two paths P_1 and P_2 is the intersection of their contents. The vertices of a sequence $(v_1, \ldots, v_k)$ are *aligned* in G if there exists a path P which is the concatenation of $k-1$ paths $P_1 P_2 \ldots P_{k-1}$ where P_i is a path from v_i to v_{i+1} for $i \in [k-1]$. Note that we might have $v_i = v_{i+1}$ and then $P_i = v_i$.

Note that, for every connected graph G, if $ab, cd \in E(G), ab \neq cd$, then (a, b, c, d), (a, b, d, c), (b, a, c, d), or (b, a, d, c) are aligned. Moreover, if G is a tree, exactly one of them is aligned. Let G be a connected graph and $a, b, c, d \in V(G)$ such that (a, b, c, d) are aligned. The *in-area* of the two edges ab and cd is the connected component of $G \setminus \{ab, cd\}$ containing the vertices b and c. The other components are called *out-areas*.

We now present a few basic results about alignments and flips. The following lemma links the connectivity of a flip and the alignment of its vertices:

Lemma 4.1. *Let G be a connected graph and $ab, cd \in E(G)$ where a, b, c and d are pairwise distinct vertices of G. If (a, b, c, d) or (b, a, d, c) are aligned in G, then the flip $(ab, cd) \to (ac, bd)$ is connected. If G is a tree, then it is also a necessary condition.*

Proof. The deletion of ab and cd leaves at most three connected components. Let us assume that (a, b, c, d) are aligned, the other case being symmetrical. Let $G_{b,c}$ be the in-area of ab, cd. Let G_a (resp. G_d) be the connected component containing a (resp. d). Note that some of these components might be identical. The addition of ac and bd connects G_a, $G_{b,c}$ and G_d back again. Thus, $(ab, cd) \to (ac, bd)$ is connected.

Assume now that G is a tree. Supposed that (a, b, c, d) and (b, a, d, c) are not aligned. Then, (a, b, d, c) or (b, a, c, d) are. Thus, the deletion of ab and cd splits G into exactly three components G_a, $G_{b,d}$ and G_c, or G_b, $G_{a,c}$ and G_d. In both cases, when we create ac and bd, we create an edge in the in-aera of ab and cd. The resulting graph then contains a cycle, and thus cannot be connected since the total number of edges is still $|V| - 1$. $\square$

Lemma 4.1 ensures that, for trees, exactly one of the two flips σ and $-\sigma$ is connected. For paths, we have seen that applying a connected flip is equivalent to reversing the portion of the path between the two involved edges. A similar statement holds for trees:

Remark 4.1. *Let T be a tree. Let e_1, f_1, e_2, f_2 be four pairwise distinct edges of T, and let σ_1 be the flip of e_1, f_1 such that the resulting tree T' is connected. Let P_1 be the path from e_1 to f_1 in T, P_2 be the path from e_2 to f_2 in T, and P_2' be the path from e_2 to f_2 in T'.*

- *If both e_2 and f_2 are in the in-area of e_1 and f_1, $P_2 = P_2'$.*
- *If both e_2 and f_2 are in the out-areas of e_1 and f_1, the contents of P_2 and P_2' are the same, but the order of the portion of the path that corresponds to P_1 is reversed (if it exists).*
- *If e_2 is in the in-area of e_1 and f_1, and f_2 is in the out-areas (or the converse), the content of P_2' is distinct from the content of P_2. Indeed, the edges that belong to $P_1 \cap P_2$ are changed for the edges of $P_1 \setminus ((P_1 \cap P_2) \cup e_2)$. (See Figure 4.2 for an illustration of this case).*

We can also remark the following:

Remark 4.2. *Let T be a tree and e_1, f_1, e be three pairwise distinct edges. The edge e is in the in-area of e_1, f_1 if and only if it is in the in-area of e_1', f_1' where e_1', f_1' are the edges created by the unique connected* *flip on e_1 and f_1. Moreover e is on the path between e_1 and f_1 in T if and only if e is on the path between e_1' and f_1' in the resulting tree.*

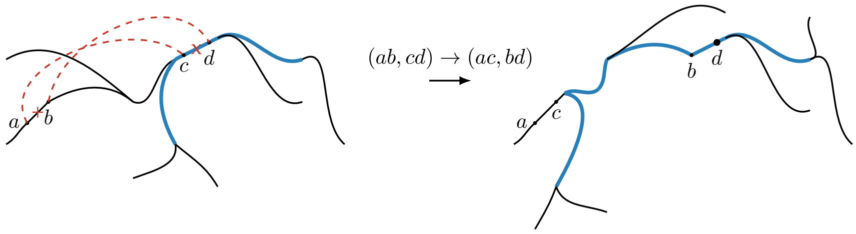


Figure 4.2: The consequences of a connected flip in a tree. The blue thick path goes from an edge of the in-area of ab and cd to an edge of an out-area before the flip, and links the two same edges afterwards.

Let e and f be two vertex-disjoint edges of a tree T, and let σ_2 be a flip in T that does not flip e nor f. The flip σ_2 *depends on e and f* if applying the connected flip on e and f changes the connectivity of σ_2. By abuse of notation, for any two flips σ_1 and σ_2 on pairwise disjoint edges, σ_2 *depends on* σ_1 if σ_2 depends on the edges of σ_1. The flip σ_1 *sees* σ_2 if exactly one of the edges of σ_2 is on the path linking the two edges of σ_1 in G.

The following lemma links the dependency of two flips and the position of their edges in a tree:

Lemma 4.2. *Let T be a tree and σ_1 and σ_2 be two flips on T, whose edges are pairwise distinct. The three following points are equivalent:*

1. *σ_2 depends on σ_1,*
2. *σ_1 depends on σ_2,*
3. *σ_2 sees σ_1 and σ_1 sees σ_2.*

Proof. Since T is a tree, Lemma 4.1 ensures that exactly one flip amongst σ_1 and $-\sigma_1$ is connected. Moreover this connected flip modifies the connectivity of $\sigma_2 : (ab, cd) \to (ac, bd)$ if and only if σ_1 modifies the alignment of a,b,c and d from (a, b, c, d) or (b, a, d, c) to (a, b, d, c) or (b, a, c, d) (or conversely). Equivalently the orientation of one of the two edges ab and cd is modified relatively to the other. Equivalently, by Remark 4.2, one of the edges ab and cd belongs to the path between the two edges e_1 and f_1 of σ_1 in T, and the other is in an out-area of e_1 and f_1. Let us call this property $(1')$. We thus have $(1 \Leftrightarrow 1')$. Let us now show that $(1' \Leftrightarrow 3)$. It will indeed give $(1 \Leftrightarrow 3)$ and, by symmetry, $(2 \Leftrightarrow 3)$.

$(1' \Rightarrow 3)$. If one of the edges ab and cd belongs to the path from e_1 to f_1 and the other is in a out-area of e_1 and f_1, then in particular, one edge is the in-area of e_1 and f_1 and the other is in an out-area. Thus, exactly one edge of σ_1 is on the path from ab to cd, and σ_2 sees σ_1. Moreover, one of the edges ab and cd belongs to the path from e_1 to f_1 and the other does not, so that σ_1 sees σ_2.

$(3 \Rightarrow 1')$. Since σ_1 sees σ_2, exactly one edge of σ_2 is on the path from e_1 to f_1. We can assume without loss of generality that ab is, and cd is not. Moreover, since σ_2 sees σ_1, exactly one edge of σ_1 is on the path from ab to cd, which means that one is in the in-area of e_1 and f_1, and the other is in an out-area. Since ab is on the path from e_1 to f_1, ab is in the in-area of e_1, f_1. And thus bc is in one out-area of e_1, f_1. □

We now give two consequences of applying a connected flip.

Lemma 4.3. *Let T be a tree and σ_1 and σ_2 be two flips on T with pairwise disjoint edges, where σ_1 is connected. Let T' be the tree obtained after applying σ_1 to T. The flip σ_1^{-1} sees σ_2 in T' if and only if σ_1 sees σ_2 in T. And σ_2 sees σ_1^{-1} in T' if and only if σ_2 sees σ_1 in T.*

Proof. The flip σ_1 sees σ_2 in T whenever exactly one edge of σ_2 is on the path between the edges of σ_1 in T. By Remark 4.2, the number of edges of $\{e_2, f_2\}$ between the edges of σ_1 is equal to the number of edges of $\{e_2, f_2\}$ between the edges of σ_1^{-1} in T'. Thus σ_1 sees σ_2 in T if and only if σ_1^{-1} sees σ_2 in T'.

On the other hand, σ_2 sees σ_1 if and only if exactly one edge of σ_2 is in the in-area of the edges of σ_1, and the other is in an out-area. By Remark 4.2, the same holds in T' for the edges of σ_1^{-1}, and thus σ_2 sees σ_1^{-1} in T' if and only if σ_2 sees σ_1 in T.

□

Lemma 4.4. *Let T be a tree and σ_1, σ_2 and σ_3 be three flips on T whose edges are pairwise disjoint and such that σ_1 sees σ_2, σ_2 sees σ_3, and σ_2 is connected. Let T' be the tree obtained by applying the flip σ_2 to T. The flip σ_1 sees σ_3 in T if and only if σ_1 does not see σ_3 in T'.*

Proof. Let e_1 and f_1 (resp. e_2, f_2 and e_3, f_3) be the edges of σ_1 (resp. σ_2 and σ_3). Let P_1 be the path from e_1 to f_1 in T, P_1' be the path from e_1 to f_1 in T', and P_2 be the path from e_2 to f_2 in T.

Since σ_1 sees σ_2 in T, exactly one edge of σ_2 is on the path P_1. Thus, one edge of σ_1 is in the in-area of e_2 and f_2, and the other is in an out-area. We can assume without loss of generality that f_1 is in the in-area. Thus, as described in Remark 4.1, the edges that belong to P_1' differ from the ones that belong to P_1 in the following way: the portion $P_1 \cap P_2$ is replaced by the portion $P_2 \setminus ((P_1 \cap P_2) \cup f_1)$.

Now, since σ_2 sees σ_3, exactly one edge of σ_3 is on the path P_2 in T. Thus, exactly one edge of σ_3 is either on $P_1 \cap P_2$ or on $P_2 \setminus ((P_1 \cap P_2) \cup f_1)$. Therefore, in T', P_1' has either exactly one edge of σ_3 which is added or removed compared to P_1.

Thus, if exactly one edge of σ_3 belongs to P_1, either both or none of the edges of σ_3 belong to P_1', and if both or none of the edges of σ_3 belong to P_1, exactly one edge of σ_3 belongs to P_1'. This concludes the proof.

□

3 Upper bound

In this section, we present the 2.5-approximation algorithm obtained in the joint work with Nicolas Bousquet.

Let us first give a shorter proof of the 4-approximation obtained by Bousquet and Mary [BM18b].

Lemma 4.5. *Let G, H be two connected graphs with the same degree sequence. There exists a sequence of at most two flips that decreases $\delta(G, H)$ by at least 2. Moreover, if there is an alternating C_4 in $\Delta(G, H)$, it can be removed in at most 2 steps, without modifying the rest of the graph.*

Proof. Let $\mathcal{C}$ be a partition of $\Delta(G, H)$ into alternating cycles, and u, v, w, x, y be five consecutive vertices of a cycle C of $\mathcal{C}$, with $uv, wx \in E(G)$ and $vw, xy \in E(H)$. Note that we may have $y = u$, if C is a C_4. At least one of the two flips $\sigma_1 : (uv, wx) \to (uw, vx)$ and $-\sigma_1 : (uv, wx) \to (ux, vw)$ is connected in G. If $-\sigma_1$ is connected, then we can apply it. Since $vw \in E(H)$, $\delta(G, H)$ decreases by at least 2 (resp. 4 if C is a C_4). Similarly, at least one of the two flips $\sigma_2 : (vw, xy) \to (vx, wy)$ and $-\sigma_2 : (vw, xy) \to (vy, wx)$ is connected in H. If $-\sigma_2$ is connected then we can apply it and $\delta(G, H)$ decreases by at least 2 (resp. 4 if C is a C_4). Thus, we can assume that σ_1 and σ_2 are the only flips that are connected. We apply σ_1 to G and σ_2 to H, and reduce $\delta(G, H)$ by 2, since both flips create the edge vx (resp. 4 if C is a C_4 since both flips also create the edge uw). □

It immediately implies the following since, in an optimal solution, the size of the symmetric difference decreases by at most 4 at each step.

Corollary 4.1. SCGT *admits a polynomial time 4-approximation algorithm.*

To improve the approximation ratio, the crucial lemma is the following:

Lemma 4.6. *Let G, H be two trees with the same degree sequence. There exists a sequence of at most 3 flips that decreases $\delta(G, H)$ by at least 4. Moreover, this sequence only flips bad edges.*

Proof. Let G' be the graph whose vertices are the connected components of $G \cap H$ and where two vertices S_1 and S_2 of G' are incident if there exists an edge in G between a vertex of S_1 and a vertex of S_2. In other words, G' is obtained from G by contracting every connected component of $G \cap H$ into a single vertex. Note that the edges of G' are the edges of $G - H$. Moreover, as G is a tree, G' also is. We can similarly define H'. Note that G' and H' have the same degree sequence.

Let S_1 be a leaf of G' and S_2 be its parent in G'. Let us show that S_2 is not a leaf of G'. Indeed, otherwise G' would be reduced to a single edge. In particular, $E(G - H)$ would contain only one edge. Since the degree sequence of $G - H$ and $H - G$ are the same, the edge of $H - G$ would have to be the same, a contradiction. Thus, we can assume that S_2 is not a leaf. Let u_1u_2 be the edge of $G - H$ between $u_1 \in S_1$ and $u_2 \in S_2$. Since $G - H$ and $H - G$ have the same degree sequence and S_1 is a leaf of G', there exists a unique vertex v_1 such that $u_1v_1 \in E(H - G)$. Moreover there exists a vertex v_2 such that $u_2v_2 \in E(H - G)$.

Let us first assume that $v_1 = v_2$. Then there exists a vertex w distinct from u_1 and u_2 such that $v_1w \in E(G - H)$ since v_1 has degree at least 2 in $H - G$. Since S_1 is a leaf of G', $w \notin S_1$ and either (u_1, u_2, v_1, w) or (u_1, u_2, w, v_1) are aligned in G. If (u_1, u_2, v_1, w) are aligned then the flip $(u_1u_2, v_1w) \to (u_1v_1, u_2w)$ in G is connected and creates the edge u_1v_1. If (u_1, u_2, w, v_1) are aligned then $(u_1u_2, v_1w) \to (u_1w, u_2v_1)$ is connected and creates the edge $u_2v_1 = u_2v_2$. In both cases, we reduce the size of the symmetric difference by at least 2 in one flip, and we can conclude with Lemma 4.5.

From now on, we assume that $v_1 \neq v_2$. We focus on the alignment of u_1, v_1, u_2 and v_2 in H. Since S_1 is a leaf of G', it is also a leaf of H'. Thus, v_1 is on the path from u_1 to u_2 and either (u_1, v_1, u_2, v_2) or (u_1, v_1, v_2, u_2) are aligned. If (u_1, v_1, u_2, v_2) are aligned then Lemma 4.1 ensures that $(u_1v_1, u_2v_2) \to (u_1u_2, v_1v_2)$ is connected in H and reduces the size of the symmetric difference by at least 2. We can conclude with Lemma 4.5. Thus, we can assume that (u_1, v_1, v_2, u_2) are aligned in H (see Figure 4.3 for an illustration).

Let us first remark that if u_2 has degree at least 2 in $H - G$ (or equivalently in $G - H$), then we are done. Indeed, if there exists $w \neq v_2$ such that $u_2w \in E(H - G)$ then, since (u_1, v_1, v_2, u_2) are aligned, (u_1, v_1, u_2, w) have to be aligned. Indeed, v_2u_2 is the only edge of $H - G$ on the path from v_1 to u_2 incident to u_2. Thus the flip $(u_1v_1, u_2w) \to (u_1u_2, v_1w)$ is connected in H. Since it reduces $\delta(G, H)$ by at least 2, we can conclude with Lemma 4.5.

From now on, we will assume that u_2 has degree 1 in $H - G$. Let H_3 (resp. H_4) be the connected component of v_1 and v_2 (resp. u_2) in $H \setminus \{u_1v_1, u_2v_2\}$, which exists since (u_1, v_1, v_2, u_2) are aligned. Note that the third component of $H \setminus \{u_1v_1, u_2v_2\}$ is reduced to S_1. By definition, H_3 is the in-area of u_1v_1 and u_2v_2.

We now show that there exists an edge $u_3u_4 \in E(G - H)$, with $u_3 \in H_3$, $u_4 \in H_4$, and such that the connected component S_4 of $G \cap H$ containing u_4 is not a leaf of G'. Indeed, since G is connected, there exists a path P from v_1 to u_2 in G. Since u_1u_2 is the only edge of $G - H$ that has an endpoint in S_1, this path does not contain any vertex of S_1. Thus, it necessarily contains an edge u_3u_4 between a vertex u_3 of H_3 and a vertex u_4 of H_4. Since H_3 and H_4 are anticomplete in $G \cap H$, $u_3u_4 \in E(G - H)$. Moreover, the connected component S_4 of $G \cap H$ containing u_4 is not a leaf of G', as it is either S_2 which is not a leaf, or P has to leave S_4 at some point with an edge of $G - H$ since P ends in $u_2 \in S_2$.

Since u_3 and u_4 have the same degree in $G - H$ and $H - G$, there exist v_3, v_4 such that $u_3v_3, u_4v_4 \in E(H - G)$. Moreover, since S_4 is not a leaf of G' (and thus of H'), there exists an edge of $H - G$ between a vertex $u_5 \in S_4$ and a vertex $v_5 \in V \setminus S_4$ where $u_5v_5 \neq u_4v_4$.

Let us prove that u_3, v_3, u_4 and v_4 are pairwise distinct. By definition, we have $u_3 \neq v_3$, $u_4 \neq v_4$ and $u_3 \neq u_4$. Moreover, since $u_3u_4 \in E(G - H)$, $u_3 \neq v_4$ and $u_4 \neq v_3$. Thus, the only vertices that can be identical are v_3 and v_4. If $v_3 = v_4$, since $u_3 \in H_3$, $u_4 \in H_4$, and v_2u_2 is the only edge of $H - G$ from H_3 to H_4, then either $v_3 = v_4 = v_2$ or $v_3 = v_4 = u_2$. In the first case, $u_4 = u_2$ since v_2u_2 is the only edge of $H - G$ from H_3 to H_4. Thus, u_2 is the endpoint of both u_1u_2 and

u_2u_3 in $G-H$. In the second case, u_2 is the endpoint of both u_2u_3 and u_2u_4 in $H-G$. Thus, in both cases, u_2 has degree at least 2 in $H-G$, a contradiction.

We now focus on the alignment of u_3, u_4, v_3 and v_4 in H. If (v_3, u_3, v_4, u_4) or (u_3, v_3, u_4, v_4) are aligned, then the flip $(u_3v_3, u_4v_4) \to (u_3u_4, v_3v_4)$ is connected in H and reduces the size of the symmetric difference by at least 2, since $u_3u_4 \in E(G-H)$. Note that the flip is well-defined since all the vertices are distinct. Thus, we can conclude with Lemma 4.5. Therefore, we can assume that (u_3, v_3, v_4, u_4) or (v_3, u_3, u_4, v_4) are aligned in H.

We give, in each case, a sequence of three flips that decreases the size of the symmetric difference by at least 4. We first state the three flips that reduce the symmetric difference in each case and then prove that these sequences of flips can be applied.

Case 1. (u_3, v_3, v_4, u_4) are aligned. (See Figure 4.3 for an illustration).
We successively apply the flips $\sigma_1 : (u_2v_2, u_5v_5) \to (u_2v_5, u_5v_2)$, $\sigma_2 : (u_3x, u_4v_4) \to (u_3u_4, xv_4)$ where $x = u_5$ if $u_3 = v_2$ and $v_3 = u_2$, and $x = v_3$ otherwise, and $\sigma_3 : (u_1v_1, u_2v_5) \to (u_1u_2, v_1v_5)$ in H. Since $u_1u_2, u_3u_4 \in E(G-H)$, this sequence of flips indeed reduces $\delta(G,H)$ by at least 4.

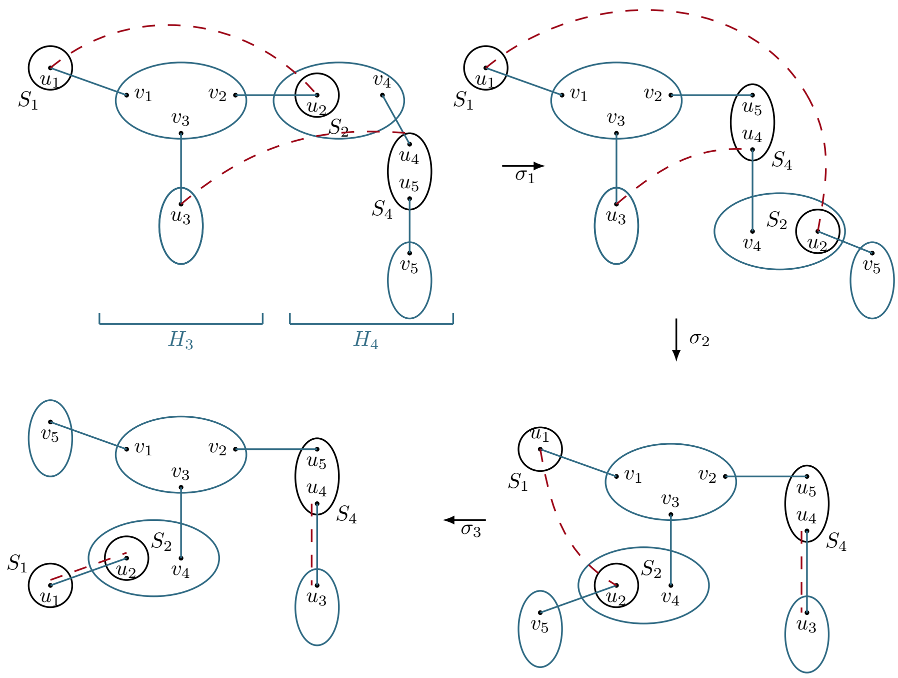


Figure 4.3: The three flips $\sigma_1 : (u_2v_2, u_5v_5) \to (u_2v_5, u_5v_2)$, $\sigma_2 : (u_3v_3, u_4v_4) \to (u_3u_4, v_3v_4)$ and $\sigma_3 : (u_1v_1, u_2v_5) \to (u_1u_2, v_1v_5)$ applied to the graph H where (u_3, v_3, v_4, u_4) are aligned. The blue full edges are in $E(H-G)$ and the red dashed edges are in $E(G-H)$.

Let us now show that this sequence of flips can be applied. We first prove that the flip $\sigma_1 : (u_2v_2, u_5v_5) \to (u_2v_5, u_5v_2)$ is well-defined since the vertices are pairwise distinct. Indeed, by definition, $u_5 \neq v_5$ and $u_2 \neq v_2$. Since $u_5 \in H_4$ and $v_2 \notin H_4$, we have $u_5 \neq v_2$. Similarly, let us show that $v_5 \in H_4$, and thus $v_5 \neq v_2$. Firstly, since $u_4 \neq u_2$ (otherwise the degree of u_2 in $G-H$ is at least 2, a contradiction), we have $v_4 \neq v_2$ and thus $v_4 \in H_4$. Moreover, by hypothesis, (u_3, v_4, u_4) are aligned in H, and since $u_4, u_5 \in S_4$ and $v_4, v_5 \notin S_4$, (u_3, v_4, u_5, v_5) are aligned in H. But $u_3 \in H_3$ and $v_4 \in H_4$ where v_2u_2 is the only edge from H_3 to H_4. This ensures that (v_2, u_2, u_5, v_5) are aligned in H, and since v_2u_2 is the only edge from H_3 to H_4, $v_5 \in H_4$. Thus we can only have $u_2 = u_5$ or $u_2 = v_5$. But, in both cases, u_2 would have degree at least 2 in $H-G$, a contradiction.

We have shown that (v_2, u_2, u_5, v_5) are aligned in H. Thus, Lemma 4.1 ensures that σ_1 is connected.

Let H_{σ_1} be the graph obtained after applying σ_1 to H (which is connected). We apply the flip $\sigma_2 : (u_3x, u_4v_4) \to (u_3u_4, xv_4)$ in H_{σ_1}, where $x = u_5$ if $u_3 = v_2$ and $v_3 = u_2$ and where $x = v_3$ otherwise.

Let us first prove that σ_2 is well-defined. The vertices of σ_2 are pairwise distinct. Indeed, we have previously shown that the vertices u_3, v_3, u_4 and v_4 are pairwise distinct which gives the conclusion in the second case. When $x = u_5$, since $v_4 \notin S_4$ and $u_3 \notin H_4$, we also have $u_5 \neq v_4$ and $u_5 \neq u_3$. Moreover, if $x = u_5$ and $u_5 = u_4$, by definition of x we have $v_2 = u_3$ and σ_1 created the edge $v_2u_5 = u_3u_4 \in G - H$, so that we can conclude with Lemma 4.5. Therefore, all the vertices of σ_2 are distinct.

Let us now show that its two edges, u_3x and u_4v_4, exist in H_{σ_1}. In order to do it, we have to prove that these edges are not the edges of σ_1. By definition, we first have $u_4v_4 \neq u_5v_5$. Moreover, if $u_4v_4 = u_2v_2$, since $v_2 \in H_3$ and $u_4 \notin H_3$, we then have $v_2 = v_4$ and $u_2 = u_4$. Thus, u_2 is the endpoint of both u_1u_2 and u_2u_3 in $G - H$, a contradiction with its degree assumption. Thus we can assume that $u_4v_4 \neq u_2v_2$ and u_4v_4 is not equal to any of the edges of σ_1. Since u_4v_4 is in H, it is in H_{σ_1}. If $x = v_3$ then $u_3v_3 \neq u_2v_2$ by definition of x. And u_3v_3 with both endpoints in H_3 is distinct from u_5v_5 which has both endpoints in H_4. If $x = u_5$, then $u_3 = v_2$ and $v_3 = u_2$. But in this case, $u_3x = v_2u_5$ was created by σ_1 and thus is in H_{σ_1}. So both edges of σ_2 exist in H_{σ_1} and then σ_2 can be applied.

We now show that σ_2 is connected in H_{σ_1}. By hypothesis, (u_3, v_3, v_4, u_4) are aligned in H. Moreover, as $u_4, u_5 \in S_4$ and $v_4, v_5 \notin S_4$, (v_4, u_4, u_5, v_5) are aligned. Therefore, if $u_3 = v_2$ and $v_3 = u_2$, $(u_3, v_3, v_4, u_4, u_5, v_5)$ are aligned in H and $(u_3, u_5, u_4, v_4, v_3, v_5)$ are aligned in H_{σ_1}. In particular, since $x = u_5$ in this case, (u_3, x, u_4, v_4) are aligned and σ_2 is connected. Otherwise, $(u_3, v_3, v_2, u_2, v_4, u_4, u_5, v_5)$ are aligned in H and $(u_3, v_3, v_2, u_5, u_4, v_4, u_2, v_5)$ are aligned in H_{σ_1}. In particular, since $x = v_3$ in this case, (u_3, x, u_4, v_4) are aligned and σ_2 is also connected.

Let H_{σ_2} be the graph obtained after applying σ_2 to H_{σ_1}. We want to apply the flip $\sigma_3 : (u_1v_1, u_2v_5) \to (u_1u_2, v_1v_5)$ in H_{σ_2}. Let us first prove that it is well-defined. By definition, we have $u_1 \neq u_2$ and $u_1 \neq v_1$. Since $v_1 \in H_3$ and $u_2 \notin H_3$, $v_1 \neq u_2$. Since $v_5 \in H_4$ while $u_1, v_1 \notin H_4$, we have $v_5 \neq u_1$ and $v_5 \neq v_1$. Finally, $v_5 \neq u_2$ was proven before applying σ_1. So the vertices of σ_3 are pairwise distinct. Let us now prove that both u_1v_1 and u_2v_5 exist in H_{σ_2}. Since u_1 is the only vertex of S_1 defined in our construction and u_1 does not appear as an endpoint in σ_1 and σ_2, u_1v_1 exits in H_{σ_2}. The edge u_2v_5 is created by σ_1, so $u_2v_5 \in E(H_{\sigma_1})$. Since $u_2, v_5 \in H_4$ and $u_3 \notin H_4$, we have $u_2v_5 \neq u_3x$. Moreover, $v_5 \notin S_4$ and $v_5 \neq v_4$ and then $u_2v_5 \neq u_4v_4$. Thus u_2v_5 is not an edge of σ_2, and $u_2v_5 \in E(H_{\sigma_2})$.

In order to prove that σ_3 is connected, we will use Lemma 4.2. Let us first prove that σ_3 is connected in H_{σ_1}. In H, (u_1, v_1, v_2, u_2) and (v_2, u_2, u_5, v_5) are aligned. Thus, in H_{σ_1}, $(u_1, v_1, v_2, u_5, u_2, v_5)$ are aligned. In particular, (u_1, v_1, u_2, v_5) are aligned and σ_3 is connected in H_{σ_1}.

Finally, we prove that in H_{σ_1}, σ_3 does not depend on σ_2. We claim that σ_2 does not see σ_3, as none of its two edges are on the path from u_3x to u_4v_4. Since S_1 is a leaf of $G \cap H$, u_1v_1 is not on it. If $x = v_3$, since $(u_3, v_3, v_2, u_5, u_4, v_4, u_2, v_5)$ are aligned in H_{σ_1}, u_2v_5 is not on it either, and if $x = u_5$, $(u_3, u_5, u_4, v_4, v_3, v_5)$ are aligned in H_{σ_1} but since $u_2 = v_3$ in this case, u_2v_5 is not on the path either. Thus, by Lemma 4.2, σ_2 and σ_3 are independent. Therefore, σ_3 is still connected in H_{σ_2}.

Case 2. (v_3, u_3, u_4, v_4) are aligned.
We apply $\sigma_1 : (u_2v_2, u_4v_4) \to (u_2v_4, u_4v_2)$, $\sigma_2 : (u_3v_3, u_4v_2) \to (u_3u_4, v_2v_3)$ then $\sigma_3 : (u_1v_1, u_2v_4) \to (u_1u_2, v_1v_4)$ to H. Again, $u_1u_2, u_3u_4 \in E(G - H)$ and it reduces $\delta(G, H)$ by at least 4.

Let us first prove that we can apply this sequence of flips. We first prove that the vertices of σ_1 are pairwise distinct. Since, by hypothesis, (v_3, u_3, u_4, v_4) are aligned in H, with $u_3 \in H_3$ and $u_4 \in H_4$, we have that (v_2, u_2, u_4, v_4) are aligned in H. By definition, $v_2 \neq u_2$ and $u_4 \neq v_4$.

Thus, the only vertices that might be identical are u_2 and u_4. But if $u_2 = u_4$, then u_2 is both the endpoint of u_1u_2 and u_2u_3 in $G - H$. Thus, u_2 has degree at least 2 in $G - H$, a contradiction.

Moreover, since (v_2, u_2, u_4, v_4) are aligned, σ_1 is connected.

Let H_{σ_1} be the graph obtained after applying σ_1 to H. We apply in H_{σ_1} the flip $\sigma_2 : (u_3v_3, u_4v_2) \to (u_3u_4, v_2v_3)$.

Let us show that its vertices are pairwise distinct. We have previously shown that $u_3, v_3 \neq u_4$. By definition, we have $u_3 \neq v_3$. Since $u_4 \in S_4$ and $v_2 \notin S_4$, $u_4 \neq v_2$. Now, if $v_2 = u_3$, then σ_1 created the edge $u_3u_4 \in E(G - H)$ and we can conclude with Lemma 4.5. Finally, since (v_3, u_3, u_4, v_4) are aligned in H with $v_3, u_3 \in H_3$ and $v_4, u_4 \in H_4$ and since v_2u_2 is the only edge of H from H_3 to H_4, we know that $(v_3, u_3, v_2, u_2, u_4, v_4)$ are aligned in H. As $v_3 \neq u_3$, this gives $v_3 \neq v_2$.

Let us now show that the two edges of σ_2, u_3v_3 and u_4v_2, exist in H_{σ_1}. We know that u_4v_2 is created by σ_1, so that $u_4v_2 \in E(H_{\sigma_1})$. Moreover, $u_3v_3 \in E(H)$. Thus, we only have to show that u_3v_3 is not an edge of σ_1. Since $u_3, v_3 \neq u_4$ and $u_3, v_3 \neq v_2$, it is straightforward.

Finally, let us show that σ_2 is connected in H_{σ_1}. We have seen that $(v_3, u_3, v_2, u_2, u_4, v_4)$ are aligned in H. Thus, in H_{σ_1}, $(v_3, u_3, v_2, u_4, u_2, v_4)$ are aligned. In particular, (v_3, u_3, v_2, u_4) are aligned and, by Lemma 4.1 σ_2 is connected.

Let H_{σ_2} be the graph obtained after applying σ_1 to H_{σ_1}. We apply in H_{σ_2} the flip $\sigma_3 : (u_1v_1, u_2v_4) \to (u_1u_2, v_1v_4)$. Let us prove that the endpoints of its edges are all distinct, that both its edges exist in H_{σ_2}, and that it is connected.

We have seen that (u_1, v_1, v_2, u_2) are aligned in H, and since $v_4 \in H_4$, (u_1, v_1, u_2, v_4) are aligned in H. Moreover, by definition, $u_1 \neq v_1$, we have previously shown that $u_2 \neq v_4$, and since $v_1 \in H_3$ and $u_2 \notin H_3$, $u_2 \neq v_1$. Therefore, the vertices of σ_3 are all pairwise distinct.

Let us now prove that its edges exist in H_{σ_2}. Since u_1 is the only vertex of S_1 we considered, we know that it is distinct from all the other vertices and thus u_1v_1 is distinct from all the other edges. Therefore, it is not an edge of σ_1 nor σ_2 and since $u_1v_1 \in E(H)$, $u_1v_1 \in E(H_{\sigma_2})$. Since u_2v_4 is created by σ_1, we have $u_2v_4 \in E(H_{\sigma_1})$. Since $u_2, v_4 \in H_4$ and $u_3, v_2 \notin H_4$, we have $u_2v_4 \neq u_3v_3$ and $u_2v_4 \neq u_4v_2$. Thus, u_2v_4 is not identical to any edge of σ_2, and $u_2v_4 \in E(H_{\sigma_2})$.

We now show that σ_3 is connected in H_{σ_2}. Since (u_1, v_1, v_2, u_2) and (v_2, u_2, u_4, v_4) are aligned in H, $(u_1, v_1, v_2, u_2, u_4, v_4)$ are aligned in H and $(u_1, v_1, v_2, u_4, u_2, v_4)$ are aligned in H_{σ_1}. In particular, (u_1, v_1, u_2, v_4) are aligned and σ_3 is connected in H_{σ_1}.

Let us prove that in H_{σ_1}, σ_3 does not depend on σ_2. We claim that σ_2 does not see σ_3. Since S_1 is a leaf of $G \cap H$, u_1v_1 is not on the path from u_3v_3 to v_2u_4 in H_{σ_1}. Moreover, since $(v_3, u_3, v_2, u_4, u_2, v_4)$ are aligned in H_{σ_1}, u_2v_4 is not on it either. Thus, by Lemma 4.2, σ_2 and σ_3 are independent. Therefore, σ_3 is still connected in H_{σ_2}.

Therefore, in all the cases, we have found a sequence of three flips whose edges are in the symmetric difference and that reduce $\delta(G, H)$ by at least 4. Moreover, the proof immediately provides a polynomial time algorithm to find such a sequence.

□

Note that Lemma 4.6 allows to obtain a 3-approximation algorithm for SCGT . Indeed, as shown in the proof of Lemma 1 in [BM18b], as long as there exists an edge of the symmetric difference in a cycle of G, one can reduce the size of the symmetric difference by 2 in one step. Afterwards, we can assume that the remaining graphs $G - H$ and $H - G$ are trees. By Lemma 4.6, in three flips, the symmetric difference of the optimal solution decreases by at most 12 while our algorithm decreases it by at least 4. (Note that free to try all the flips, finding these flips is indeed polynomial). But we can actually improve the approximation ratio. The idea consists in treating differently short cycles. A *short* cycle is a C_4, a *long* cycle is a cycle of length at least 6. We now give the main result of this section.

Theorem 4.2. SCGT *admits a* $5/2$*-approximation algorithm running in polynomial time. It becomes a* $9/4$*-approximation algorithm if* $\Delta(G, H)$ *does not contain any short cycle.*

Proof. Let $\mathcal{C}$ be an optimal partition of $\Delta(G, H)$ into alternating cycles, i.e. a partition with $mnc(G, H)$ cycles. Let c be the number of short cycles in $\mathcal{C}$. Bereg and Ito [BI17] provide a polynomial time algorithm to find a partition of $\Delta(G, H)$ into alternating cycles having at least $\frac{c}{2}$ short cycles. Lemma 4.5 ensures that we can remove their $2c$ edges from the symmetric difference in at most c flips. If an edge of the symmetric difference is in a cycle of G or H, then in one step we can reduce the symmetric difference by 2 [BM18b]. Otherwise, by Lemma 4.6, we can remove the remaining $\delta(G, H) - 2c$ edges using at most $\frac{3(\delta(G,H)-2c)}{4}$ flips in polynomial time. Therefore, we can transform G into H with at most $c + \frac{3(\delta(G,H)-2c)}{4}$ flips.

Let us now provide a lower bound on the length of a shortest transformation from G to H. By definition, $\mathcal{C}$ contains c short cycles. Theorem 4.1 ensures that we need at least c steps to remove the short cycles, plus $\ell - 1$ flips to remove each cycle of length 2ℓ. Therefore, we need at least $\frac{\delta(G,H)-4c}{3}$ flips to remove the $\delta(G, H) - 4c$ remaining edges from the symmetric difference.

The ratio between the upper bound and the lower bound is

$$f(c) := \frac{c + \frac{3\delta(G,H)-6c}{4}}{c + \frac{\delta(G,H)-4c}{3}} = \frac{3(3\delta(G, H) - 2c)}{4(\delta(G, H) - c)}.$$

The function f being increasing and since the number of short cycles in $\mathcal{C}$ cannot exceed $\frac{\delta(G,H)}{4}$, we have $f(c) \le f(\frac{\delta(G,H)}{4}) = \frac{5}{2}$. It gives a $\frac{5}{2}$-approximation in polynomial time. Moreover, when there is no alternating short cycle in $\Delta(G, H)$, $c = 0$. Since $f(0) = \frac{9}{4}$, we obtain a $\frac{9}{4}$-approximation. $\square$

4 Discussion on the tightness of the lower bound

In this section, we discuss the quality of the lower bound of Theorem 4.1. We first prove that if we only flip bad edges of the same cycle of the symmetric difference then the length of a shortest transformation can be almost twice longer than the one given by the lower bound of Theorem 4.1. In order to prove it, we generalize several techniques and results of Christie [Chr98], proved for the SORTING BY REVERSALS problem.

Note that the result of Hannenhalli and Pevzner [HP99] actually proves that in the case of paths, when the symmetric difference only contains vertex-disjoint cycles, it is not necessarily optimal to only flip edges of the same cycle. However, studying this restriction gives us a better understanding of the general problem.

We also prove that, if we only flip bad edges (which are not necessarily in the same cycle of the symmetric difference), then the length of a shortest transformation can be almost $3/2$ times longer than the one given by the lower bound. Note that all the existing approximation algorithms for SORTING BY REVERSALS and SCGT only flip bad edges. But again no formal proof guarantees that there always exists a shortest transformation where we only flip bad edges.

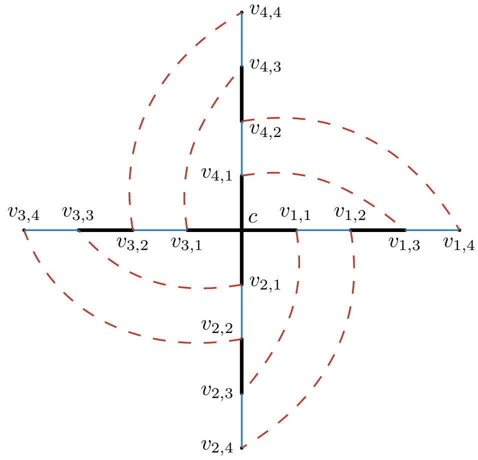


Figure 4.4: The graphs G_4 and H_4. The black thick edges are in $E(G_4 \cap H_4)$, the blue thin edges are in $E(G_4 - H_4)$ and the red dashed edges are in $E(H_4 - G_4)$.

Both results are obtained with the same graphs G_k and H_k represented in Figure 4.4 for $k = 4$. For any $k \geq 2$, let $G_k = (V_k, E(G_k))$ and $H_k = (V_k, E(H_k))$ be the graphs with $V_k = \{v_{i,j}, 1 \leq i \leq k, 1 \leq j \leq 4\} \cup \{c\}$, $E(G_k) = \bigcup_{i \in [k]} \{cv_{i,1}, v_{i,1}v_{i,2}, v_{i,2}v_{i,3}, v_{i,3}v_{i,4}\}$, and $E(H_k) = \bigcup_{i \in [k]} \{cv_{i,1}, v_{i,1}v_{i+1,3}, v_{i,2}v_{i,3}, v_{i,2}v_{i+1,4}\}$, where the additions are defined modulo k. One can easily check that, in this construction, both G_k and H_k are the subdivisions of a star where each branch has 4 vertices. Note that $\Delta(G, H)$ is the disjoint union of k short cycles. Moreover, the partition of $\Delta(G, H)$ into alternating cycles is unique.

4.1 Flipping bad edges of the same cyle

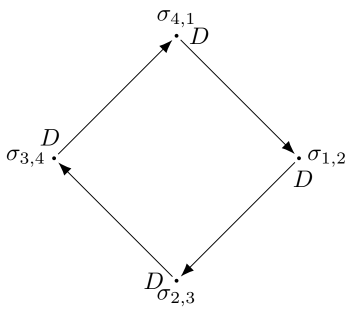


Figure 4.5: The digraph of flips of G_4 and H_4, where $\sigma_{i,j} := (v_{i,1}v_{i,2}, v_{j,3}v_{j,4}) \to (v_{i,1}v_{j,3}, v_{i,2}v_{j,4})$ for any i and j, and the label D stands for disconnected.

Let G and H be two trees with the same degree sequence. The *digraph of flips* $\mathcal{F}(G, H)$ of G and H is the labeled directed graph whose vertices are the good flips in $G - H$ (i.e. the flips that create at least one edge of $G \cap H$, regardless of the fact that they maintain the connectivity of G or not). Every vertex σ is labeled as a connected or non-connected flip. And (σ_1, σ_2) is an arc of $\mathcal{F}(G, H)$ if and only if σ_1 sees σ_2. Note that every vertex of the digraph of flips corresponds to a good flip σ in $G - H$ and thus corresponds to a pair of edges of $G - H$. Since there exists a connected flip between any pair of edges in a tree, if σ is disconnected, then $-\sigma$ is connected and thus any vertex of $\mathcal{F}(G, H)$ can be associated to a connected flip, either itself or its opposite.

If G and H are paths, if exactly one of the two edges of a flip σ_1 is on the path between the two edges of a flip σ_2, then exactly one of the two edges of σ_2 is on the path between the two edges of σ_1. Thus, for paths, the digraph of flips is a non-oriented graph, and it corresponds to the reversal graph introduced by Christie [Chr98]. The reversal graph is related to the interleaving graph introduced in [HP99] for the sorting by reversals of signed permutations (which corresponds to the case where G and H are paths and the partition $\mathcal{C}$ of $\Delta(G, H)$ into alternating cycles is unique). The vertices of the interleaving graph are the cycles of $\mathcal{C}$, labeled as connected if there exists a connected good flip between two edges of C and disconnected otherwise, and there is an edge between two cycles C_1 and C_2 if there exists a connected flip between two edges of C_1 that changes the connectivity of a flip between two edges of C_2. Again, for paths, the converse is also true and the graph is therefore non-directed.

For paths, Christie gives in [Chr98] a characterization of the resulting reversal graph when we apply a flip (of bad edges). Unfortunately, his proof cannot be extended easily to the case of trees for the digraph of flips. Indeed, the arcs and the labels above are not enough to determine the connectivity of the flips in the resulting graph. However, when we restrict to the case where the partition $\mathcal{C}$ of $\Delta(G, H)$ into alternating cycles is unique and only contains short cycles, the model becomes simple enough to be understood. The first part of this section consists in proving that, under these assumptions, we can characterize the resulting digraph of flips when we apply a good flip (or, if the flip is disconnected, its opposite) on the digraph of flips.

Two graphs G and H with the same degree sequence are *close* if $\Delta(G, H)$ admits a unique partition into alternating cycles $\mathcal{C}$ and all the cycles of $\mathcal{C}$ are short. Note that G_k and H_k are close. All along the proofs of the section, we will implicitly use the following remarks:

Remark 4.3. *Let G, H be two close trees. Let $\mathcal{C}$ be the unique partition of $\Delta(G, H)$ into alternating cycles. The digraph of flips contains $|\mathcal{C}|$ flips and each vertex corresponds to the unique good flip that removes some cycle C of the symmetric difference. Moreover, if we flip edges on the same cycle of the symmetric difference, then we either use the flips of the digraph of flips or their opposite.*

Remark 4.4. *Let G, H be two close trees. If we apply any connected flip between edges of the same cycle of $G - H$, then, the resulting graph G' and H are still close.*

Proof. Since G and H are close, the decomposition $\mathcal{C}$ of $\Delta(G, H)$ into alternating cycles is unique and only contains short cycles. Thus, if we apply a connected flip $(ab, cd) \to (ac, bd)$, with ab and cd in the same cycle of $\mathcal{C}$, then (a, b, d, c, a) or (a, b, c, d, a) is an alternating C_4 in $\Delta(G, H)$. In the first case, σ is good and the corresponding C_4 disappears from $\Delta(G, H)$ and in the second case, it is replaced by (a, c, b, d, a). Since the edges of the other cycles are unchanged, in both cases, the partition of $\Delta(G', H)$ into alternating cycles remains unique and still contains only short cycles (where G' is the resulting graph). $\square$

A vertex σ' of a digraph of flips $\mathcal{F}'$ *replaces* a vertex σ of a digraph of flips $\mathcal{F}$ if $V(\mathcal{F}') = (V(\mathcal{F}) \setminus \sigma) \cup \sigma'$, and the label, the incoming neighbors and the outgoing neighbors of σ' in $\mathcal{F}'$ are exactly the label, the incoming neighbors and the outgoing neighbors of σ in $\mathcal{F}$.

Lemma 4.7. *Let G and H be two close trees and let $\mathcal{F} := \mathcal{F}(G, H)$. Let $\sigma \in V(\mathcal{F})$ and G' be the graph obtained from G by applying σ if it is connected and by applying $-\sigma$ otherwise.*

The graph $\mathcal{F}' := \mathcal{F}(G', H)$ is characterized as follows:

1. *If σ is connected, then $V(\mathcal{F}') = V(\mathcal{F}) \setminus \{\sigma\}$. If σ is disconnected, then $-((-\sigma)^{-1})$ replaces σ.*
2. *For every $\sigma_1 \notin N^-_{\mathcal{F}}(\sigma) \cap N^+_{\mathcal{F}}(\sigma)$, σ_1 is connected in $\mathcal{F}$ if and only if σ_1 is connected in $\mathcal{F}'$.*
3. *For every $\sigma_1 \in N^-_{\mathcal{F}}(\sigma) \cap N^+_{\mathcal{F}}(\sigma)$, σ_1 is connected in $\mathcal{F}$ if and only if σ_1 is disconnected in $\mathcal{F}'$.*
4. *For every $\sigma_1, \sigma_2 \in V(\mathcal{F}')$, with $\sigma_1 \notin N^-_{\mathcal{F}}(\sigma)$ or $\sigma_2 \notin N^+_{\mathcal{F}}(\sigma)$, $\sigma_1\sigma_2 \in E(\mathcal{F})$ if and only if $\sigma_1\sigma_2 \in E(\mathcal{F}')$.*
5. *For every $\sigma_1 \in N^-_{\mathcal{F}}(\sigma)$ and every $\sigma_2 \in N^+_{\mathcal{F}}(v)$ such that $\sigma_1 \neq \sigma_2$, $\sigma_1\sigma_2 \in E(\mathcal{F})$ if and only if $\sigma_1\sigma_2 \notin E(\mathcal{F}')$.*

Proof. Let a, b, c and d be the vertices of G such that $\sigma = (ab, cd) \to (ac, bd)$. First, notice that since the partition of $\Delta(G, H)$ is unique and only contains short cycles, and since we only apply good flips, or opposites of good flips, all the different flips we consider here are on disjoint edges.

Proof of (1).
First note that after applying the flip, all the flips of the graph of flips distinct from σ still exist. Indeed, all the cycles of $\mathcal{C}$ distinct from the one of σ are still in $\mathcal{C}'$. Thus, by Remark 4.3, the set of vertices $V(\mathcal{F}) \setminus \sigma$ is in $V(\mathcal{F}')$. Moreover, all the cycles of $\mathcal{C}'$ distinct from the one of the edges created by σ are also in $\mathcal{C}$. Thus, by Remark 4.3, $V(\mathcal{F}') \setminus \{((-\sigma)^{-1}), -((-\sigma)^{-1})\}$ is in $V(\mathcal{F})$.

If σ is connected, then the number of cycles in the partition decreases by one and the vertex is removed. The vertex corresponding to σ disappears but all the other vertices still exist.

Let us now show that if σ is disconnected, then $-((-\sigma)^{-1})$ replaces σ. Firstly, if σ is disconnected in G, then $-\sigma$ is applied to G. Since the partition of $\Delta(G, H)$ into alternating cycles is unique, each vertex of $\Delta(G, H)$ is incident to exactly one edge of $G - H$ and one edge of $H - G$, and as σ is a good flip, $-\sigma$ is not. Thus, the edges created by $-\sigma$ are in $\Delta(G', H)$. That being said, $(-\sigma)^{-1}$ is not a good flip of $\mathcal{F}'$. Indeed, $(-\sigma)^{-1} = (ad, bc) \to (ab, cd)$, and we know that ab and cd are in $\Delta(G, H)$. On the other hand, the flip $-((-\sigma)^{-1}) = (ad, bc) \to (ac, bd)$ is good, as it creates the same edges as σ. Therefore, $-((-\sigma)^{-1}) \in V(\mathcal{F}')$.

Moreover, $-((-\sigma)^{-1})$ is disconnected. Indeed otherwise σ would be connected in $\mathcal{F}$ since σ and $-((-\sigma)^{-1})$ create the same edges. Thus, $-((-\sigma)^{-1})$ has in $\mathcal{F}'$ the label of σ in $\mathcal{F}$.

Finally, by Lemma 4.3, $(-\sigma)^{-1}$ has the same in and out neighborhoods as $-\sigma$. Thus, as $-((-\sigma)^{-1})$ is flipping the same edges as $(-\sigma)^{-1}$, and $-\sigma$ is flipping the same edges as σ, $-((-\sigma)^{-1})$ and σ have the same in and out neighborhoods.

Proof of (2) and (3).
The points 2 and 3 are a direct consequence of Lemma 4.2: the label of σ is considering the fact that the edges, and thus the incoming neighbors and outgoing neighbors, of σ and $-\sigma$ are the same.

Proof of (4).
By Remark 4.1, applying a flip on the edges e and f to a tree T can modify the content of a path P only if one endpoint of P is in the in-area of e and f and the other is in their out-area. When it changes, the portion $P_1 \cap P$ of P is replaced by $P_1 \setminus (P_1 \cap P)$, where P_1 is the path from e to f in T.

Suppose that an arc $\sigma_1\sigma_2$ is in $\mathcal{F}$ but not in $\mathcal{F}'$, or conversely, and let e_1 and f_1 be the edges of σ_1, and e_2 and f_2 be the edges of σ_2. The content of the path P_1 from e_1 to f_1 in G is different from the content of the path P_1' from e_1 to f_1 in G', as either P_1 contains exactly one edge of σ_2 and P_1' contains both or neither edges of σ_2, or conversely. Thus, one of the edges of σ_1 is in the in-area of ab and cd, and the other is in an out-area. Therefore, σ_1 sees σ in G.

Moreover, in P_1', exactly one edge of σ_2 is either added or removed compared to the content of P_1. Thus, either one edge is on the portion of the path from ab to cd that is common to P_1 and none were on the other portion, or the opposite. Thus, exactly one edge of σ_2 is on the path from ab to cd in G, and σ sees σ_2 in G.

Proof of (5).
Finally, point 5 is a consequence of Lemma 4.4. □

Note that Lemma 4.7 generalizes the results of [Chr98] when G, H are paths. Indeed, in that case, the graph is non-directed and then the subgraph induced by the neighborhood of σ in $\mathcal{F}$ is complemented after the flip.

Lemma 4.8. *Let G and H be two close trees. Every disconnected flip of $\mathcal{F}(G, H)$ belongs to an oriented cycle in $\mathcal{F}(G, H)$.*

Proof. Let $\mathcal{F} := \mathcal{F}(G, H)$. Assume by contradiction that there exists a disconnected flip $\sigma \in V(\mathcal{F})$ such that σ does not belong to any oriented cycle of $\mathcal{F}$.

Since the the partition of $\Delta(G, H)$ into alternating cycles only contains short cycles, the proof of Lemma 4.5 ensures that there exists a sequence of flips transforming G into H that only flips edges that are in the same short cycle. In other words, there always exists a sequence of flips using flips of $\mathcal{F}$ that transforms G into H. By Remark 4.4, all the intermediate graphs and H are close, so Lemma 4.7 holds at any step. Moreover, during such a transformation, every vertex of $\mathcal{F}$ has to be removed at some point (since $\mathcal{F}(H, H)$ is empty). By Lemma 4.7.1, a vertex σ_2 can be removed only if σ_2 is connected and we apply σ_2. Thus, the label of σ_2 has to change at some point. Lemma 4.7 ensures that for if the label of σ_2 changes then σ_2 is the in- and outgoing neighborhood of one flip, and thus is in a oriented cycle of $\mathcal{F}$ (of size 2).

Let $\mathcal{F}_1$ be the last step where σ_2 is not in a cycle of the digraph of flips. Assume that we apply a flip σ_1 and let $\mathcal{F}_1'$ be the new digraph of flips. Let us prove by contradiction that σ_2 was in a cycle of $\mathcal{F}_1$.

Let C' be a cycle of $\mathcal{F}_1'$ containing σ_2. If σ_2 does not belong to any oriented cycle in $\mathcal{F}_1$, since no vertices have been added to $\mathcal{F}_1$, there exists an arc $\sigma_3\sigma_4$ of C' that is not in $\mathcal{F}_1$. By Lemma 4.7, $\sigma_3\sigma_1, \sigma_1\sigma_4 \in E(\mathcal{F}_1)$. By replacing every arc $\sigma_3\sigma_4$ of C' that is not in $\mathcal{F}_1$ by the two arcs $\sigma_3\sigma_1$ and $\sigma_1\sigma_4$, we obtain a union of oriented cycles of $\mathcal{F}_1$, one of them containing σ_2. Therefore, σ_2 belongs to an oriented cycle in $\mathcal{F}_1$, a contradiction. □

Note that Lemma 4.8 generalizes a result of Christie [Chr98] which ensures that when G, H are paths, no disconnected flip is isolated in the graph of flips.

Lemma 4.9. *Let G and H be two close trees. Let $\mathcal{F} := \mathcal{F}(G, H)$ and $\mathcal{C}$ be the unique partition of $\Delta(G, H)$ into alternating cycles. If we only flip pairs of edges that are in the same cycle of $\mathcal{C}$, then the shortest transformation from G to H has length at least $|V(\mathcal{F})| + \gamma(\mathcal{F})$, where $\gamma(\mathcal{F})$ is defined as follows:*

- *If there is no oriented cycle in $\mathcal{F}$, or if there exists an oriented cycle in $\mathcal{F}$ that only contains connected flips, $\gamma(\mathcal{F}) = 0$.*
- *Otherwise, $\gamma(\mathcal{F}) = nd(\mathcal{F}) - 1$, where $nd(\mathcal{F})$ is the minimum number of disconnected flip in any oriented cycle of $\mathcal{F}$.*

Proof. We prove Lemma 4.9 by induction on $(|V(\mathcal{F})| + \gamma(\mathcal{F}))$. If $(|V(\mathcal{F})| + \gamma(\mathcal{F})) = 0$, then $\mathcal{F}$ is the empty graph and thus $G = H$. Assume now that $(|V(\mathcal{F})| + \gamma(\mathcal{F})) \geq 1$. Let σ be a good flip between two edges that are in the same cycle of $\mathcal{C}$. Either σ or $-\sigma$ is connected. Let G' be the graph obtained after applying the connected flip, σ or $-\sigma$, and $\mathcal{F}' := \mathcal{F}(G', H)$. We will show that $(|V(\mathcal{F}')| + \gamma(\mathcal{F}')) - (|V(\mathcal{F})| + \gamma(\mathcal{F})) \geq -1$. Since we only flip edges of the same short cycle, Remark 4.4 ensures that G' and H are close and the proof immediately follows.

By Lemma 4.7.1, we have $|V(\mathcal{F}')| - |V(\mathcal{F})| = -1$ if σ is connected and 0 if σ is disconnected. We call this property the property (*). Let us now bound the quantity $\gamma(\mathcal{F}') - \gamma(\mathcal{F})$.

Assume that there exists $\sigma_1 \in V(\mathcal{F})$ such that $\sigma_1 \in N^-_{\mathcal{F}}(\sigma) \cap N^+_{\mathcal{F}}(\sigma)$. Note that σ, σ_1 forms a cycle of size 2. If σ or σ_1 are connected then $\gamma(\mathcal{F}) = 0$. So we have $\gamma(\mathcal{F}') - \gamma(\mathcal{F}) \geq 0$. If both σ and σ_1 are disconnected then $\gamma(\mathcal{F}) \leq 1$. And then $\gamma(\mathcal{F}') - \gamma(\mathcal{F}) \geq -1$. Thus, in both cases, the property (*) gives $(|V(\mathcal{F}')| + \gamma(\mathcal{F}')) - (|V(\mathcal{F})| + \gamma(\mathcal{F})) \geq -1$ and the result is proven.

So we can assume that no vertex σ_1 of $\mathcal{F}$ satisfies $\sigma_1 \in N^-_{\mathcal{F}}(\sigma) \cap N^+_{\mathcal{F}}(\sigma)$. By Lemma 4.7.2 and 4.7.3, the labels of the vertices of $\mathcal{F}'$ and $\mathcal{F}$ are the same. Thus, if $\gamma(\mathcal{F}') < \gamma(\mathcal{F})$, it is because $\mathcal{F}'$ has no oriented cycle, or because at least one cycle of $\mathcal{F}'$ is not in $\mathcal{F}$. Let us consider both cases.

Suppose that $\mathcal{F}'$ has no oriented cycle. Lemma 4.8 ensures that every flip of $\mathcal{F}'$ is connected. Since the labels are the same in $\mathcal{F}$ and $\mathcal{F}'$, only σ can be disconnected in $\mathcal{F}$ if it has been removed in $\mathcal{F}'$, but by Lemma 4.7, if σ has been removed, σ is connected in $\mathcal{F}$. Thus, $\mathcal{F}$ only contains connected flips, and either there are no oriented cycles in $\mathcal{F}$, or the only oriented cycles contain only connected flips. Thus, combining $\gamma(\mathcal{F}) = \gamma(\mathcal{F}') = 0$ with property (*), we have $(|V(\mathcal{F}')| + \gamma(\mathcal{F}')) - (|V(\mathcal{F})| + \gamma(\mathcal{F})) \geq -1$.

So we can assume that at least one oriented cycle C' of $\mathcal{F}'$ is not in $\mathcal{F}$. Since $V(\mathcal{F}') \subseteq V(\mathcal{F})$ and the labels are the same in $\mathcal{F}$ and $\mathcal{F}'$, at least one arc of C' is not in $\mathcal{F}$. Let us prove that there exists a cycle C in $\mathcal{F}$ such that $V(C) \subseteq V(C') \cup \{\sigma\}$.

If exactly one arc $\sigma_1\sigma_2$ of C' is not in $\mathcal{F}$ then, by Lemma 4.7.4 and 4.7.5, $\sigma_1 \in N^-_{\mathcal{F}}(\sigma)$ and $\sigma_2 \in N^+_{\mathcal{F}}(\sigma)$. Moreover, the path P of C' from σ_2 to σ_1 is also in $\mathcal{F}$, so that in $\mathcal{F}$, P plus $\sigma_1\sigma$ and $\sigma\sigma_2$ form an oriented cycle C in $\mathcal{F}$, with $V(C) \subseteq V(C') \cup \{\sigma\}$. If at least two distinct arcs of C' are not in $\mathcal{F}$, let $\sigma_1\sigma_2$ and $\sigma_3\sigma_4$ be two such arcs. We can choose $\sigma_1\sigma_2$ and $\sigma_3\sigma_4$ so that the oriented path P from σ_2 to σ_3 in C' only contains arcs that are in $\mathcal{F}$. Lemma 4.7 ensures that $\sigma_1, \sigma_3 \in N^-_{\mathcal{F}}(\sigma)$ and $\sigma_2, \sigma_4 \in N^+_{\mathcal{F}}(\sigma)$. If $\sigma_2 = \sigma_3$ or $\sigma_4 = \sigma_1$, $N^-_{\mathcal{F}}(\sigma) \cap N^+_{\mathcal{F}}(\sigma)$ is not empty, a contradiction with the assumptions. Thus, $\sigma_1\sigma_2$ and $\sigma_3\sigma_4$ are not consecutive in C'. Since $\sigma_3 \in N^-_{\mathcal{F}}(\sigma)$ and $\sigma_2 \in N^+_{\mathcal{F}}(\sigma)$, in $\mathcal{F}$, P plus $\sigma_3\sigma$ and $\sigma\sigma_2$ forms an oriented cycle C in $\mathcal{F}$, with $V(C) \subseteq V(C') \cup \{\sigma\}$.

Therefore, in both cases, there exists an oriented cycle C in $\mathcal{F}$ such that $V(C) \subseteq V(C') \cup \{\sigma\}$. Since all the vertices have the same label in $\mathcal{F}$ and $\mathcal{F}'$, the minimum number of disconnected flips in an oriented cycle of $\mathcal{F}$ is therefore at most the number of disconnected flips in C if σ is connected, and the number of disconnected flips in C +1 if σ is disconnected. Thus, if σ is connected, $\gamma(\mathcal{F}') - \gamma(\mathcal{F}) \geq 0$, and if σ is disconnected, $\gamma(\mathcal{F}') - \gamma(\mathcal{F}) \geq -1$. Again, in both cases, property (*) ensures that $(|V(\mathcal{F}')| + \gamma(\mathcal{F}')) - (|V(\mathcal{F})| + \gamma(\mathcal{F})) \geq -1$.

□

Note that the lower bound given by Lemma 4.9 corresponds to the upper bound given by Christie [Chr98] for paths when the graph of flips is connected. Indeed, Christie gives an algorithm to transform any path G into another one H by using $|V(\mathcal{F}(G,H))| + s$ good flips, where s is the number of connected components of $\mathcal{F}(G,H)$ that only have disconnected flips. Thus, if the graph of flips is connected, s is equal to 0 if there exists a connected flip in it, and 1 otherwise. As the graph is undirected in this case, s is thus equal to $\gamma(\mathcal{F}(G,H))$.

In our case, the lower bound given by Lemma 4.9 is not necessarily tight when we only flip bad edges of the same cycle. Indeed, let us consider for example the graphs G'_k and H'_k obtained from G_k and H_k by adding a connected and a disconnected C_4, that see each other, on the same branch of the original subdivided star (see Figure 4.6).

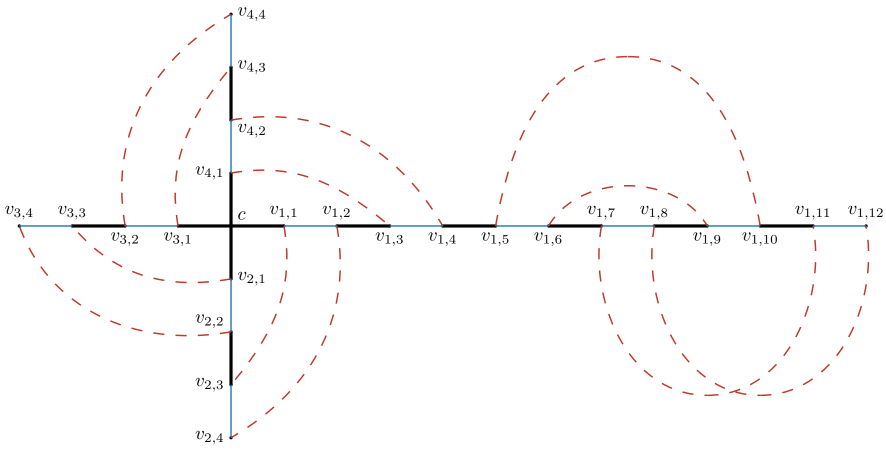


Figure 4.6: G'_4 and H'_4. The black thick edges are in $E(G'_4 \cap H'_4)$, the blue thin edges are in $E(G'_4 - H'_4)$ and the red dashed edges are in $E(H'_4 - G'_4)$.

We claim that a proof similar to the one of Lemma 4.11 can be adapted to prove that the shortest transformation from G'_k to H'_k has length at least $\frac{3k}{2}$. On the other hand, the addition of the two cycles on a leaf of a branch created in $\mathcal{F}(G'_k, H'_k)$ an oriented cycle of length 2 with a connected and a disconnected vertex. Thus, Lemma 4.9 gives the lower bound $k+2$.

We can now apply Lemma 4.9 to prove the following. Recall that G_k and H_k were defined at the beginning of the section.

Lemma 4.10. *If we only flip pairs of edges that are in the same cycle of $\mathcal{C}$, then the shortest transformation from G_k to H_k has length at least $2k-1$.*

Proof. Let us first show that G_k and H_k are close. We thus need to prove that $\Delta(G_k, H_k)$ has a unique partition into alternating cycles, containing only short cycles. In the proof all the indices have to be read modulo k. By construction, the vertex c is not incident to any edge of $\Delta(G_k, H_k)$ and all the other vertices are incident to one edge of $G_k - H_k$ and one edge of $H_k - G_k$. For every i, $v_{i,1}$ is incident to an edge of $G_k - H_k$, namely $v_{i,1}v_{i,2}$. The vertex $v_{i,2}$ is incident to an edge of $H_k - G_k$, namely $v_{i,2}v_{i+1,4}$, which in turn is incident to an edge of $G_k - H_k$, $v_{i+1,4}v_{i+1,3}$. And $v_{i+1,3}$ is incident to an edge of $H_k - G_k$, $v_{i+1,3}v_{i,1}$. This set of four edges create a short cycle, denoted by C_i. Since this property holds for every i, $\Delta(G_k, H_k)$ can be partitioned into k short cycles. The uniqueness of the edges of $G_k - H_k$ and $H_k - G_k$ incident to a vertex $v_{i,j}$ ensures the uniqueness of the decomposition into cycles.

Let us now prove that $\mathcal{F} := \mathcal{F}(G_k, H_k)$ only contains disconnected flips. Since there are $4k$ edges in $\Delta(G_k, H_k)$, there are therefore k vertices in $\mathcal{F}$. Let us prove that, for every i, the good flip $\sigma_i \in \mathcal{F}(G_k, H_k)$ for C_i is disconnected. Since $v_{i,2}v_{i+1,4}$ and $v_{i+1,3}v_{i,1}$ are the edges of $H_k - G_k$ in C_i, the good flip of $\mathcal{F}$ is $\sigma_i : (v_{i,1}v_{i,2}, v_{i+1,4}v_{i+1,3}) \to (v_{i,1}v_{i+1,3}, v_{i,2}v_{i+1,4})$. By construction, $(v_{i,2}, v_{i,1}, v_{i+1,3}, v_{i+1,4})$ are aligned in G_k and then Lemma 4.1 ensures that σ_i is disconnected.

Let us finally show that $\mathcal{F}$ is an oriented cycle of length k. For every i, the path from $v_{i,1}v_{i,2}$ to $v_{i+1,4}v_{i+1,3}$ in G_k is $P_i := (v_{i,1}, c, v_{i+1,1}, v_{i+1,2}, v_{i+1,3})$. The only edge of $G_k - H_k$ that belongs to P_i is $v_{i+1,1}v_{i+1,2}$, which is an edge of σ_{i+1}. So σ_i only sees σ_{i+1}.

Therefore, $\mathcal{F}$ is an oriented cycle of length k containing only disconnected flips.

Since G_k, H_k are close, Lemma 4.9 ensures that a shortest transformation when we only flip pairs of edges that are in the same cycle of $\mathcal{C}$ has length at least $|V(\mathcal{F})| + \gamma(\mathcal{F}) = 2k - 1$. Indeed $|V(\mathcal{F})| = k$ and $\gamma(\mathcal{F}) = k - 1$.

□

We claim that the lower bound given by Lemma 4.10 is also an upper bound. Indeed, by Lemma 4.5, every short cycle can be removed within two steps. We can apply this strategy for $k-1$ cycles which requires $2k-2$ steps. When there only remains one cycle in the symmetric difference, then it is easy to check that the good flip indeed keeps the connectivity of the graph. Thus the last short cycle can be removed in one step.

From Lemma 4.10, we can finally deduce the following corollary:

Corollary 4.2. *There exist some connected graphs G and H with the same degree sequence for which, if we only flip edges of the same cycle of any decomposition of $\Delta(G,H)$ into alternating cycles, the shortest connected transformation from G to H has length at least $2(\frac{\delta(G,H)}{2} - mnc(G,H)) - 1$.*

4.2 Flipping bad edges

The restriction of flipping only edges that are in the same cycle of the partition of $\Delta(G_k, H_k)$ into alternating cycles might seem strong.

That being said, we have also studied the transformation from G_k to H_k under a weaker assumption, which is the one of only flipping bad edges.

Lemma 4.11. *If at any time, we only flip pairs of bad edges, then the shortest transformation from G_k to H_k has length at least $\lceil \frac{3k}{2} \rceil - 1$.*

Proof. Let G_t be a graph obtained after applying t arbitrary flips, whose edges are in $\Delta(G_k, H_k)$, to G_k. The *branch* B_i of G_t is the unique path from $v_{i,1}$ to a leaf of G_t (that is therefore identified as the leaf of the branch) that does not contain the vertex c. Note that since we only flip bad edges, in G_t, one of the two edges incident to $v_{i,1}$ is $cv_{i,1}$. A *core* of a branch B_i is an edge $v_{j,2}v_{j,3}$ that belongs to the branch B_i. Note that a branch might contain no core. An edge of $G_t - H$ is *external* if is incident to a leaf of G_t, and *internal* otherwise. An *inversion* of B_i is a flip whose edges are both on B_i. A *displacement* between two branches B_i and B_j is a flip between one edge of B_i and one edge of B_j. Note that all the flips are either inversions or displacements.

Let S be a sequence of flips that transforms G_k into H_k using only bad edges, and let us show that S has length at least $\lceil \frac{3k}{2} \rceil - 1$.

Let us first show that S contains at least $k-2$ displacements if k is even, and $k-1$ if k is odd. First note that the only way to change the leaf of the branch B_i consists in making a displacement between B_i and another branch B_j. Indeed, an inversion flips two edges of the same branch, and therefore does not change its content. On the other hand, a displacement between the branches B_i and B_j permutes the leaves of the two branches.

Since the leaf of the branch B_i is $v_{i,4}$ in G_k, and $v_{i+2,4}$ in H_k (the addition being modulo k), the leaves associated to the branches $(B_1, \ldots, B_k)$ must be changed from $(v_{1,4}, v_{2,4}, \ldots, v_{k-2,4}, v_{k-1,4}, v_{k,4})$ to $(v_{3,4}, v_{4,4}, \ldots, v_{k,4}, v_{1,4}, v_{2,4})$, only using displacements, i.e. transpositions of the leaves. The canonical notation of the permutation (i.e partition into cycles) from $(1, 2, \ldots, k-2, k-1, k)$ to $(3, 4, \ldots, k, 1, 2)$ is either $(1, 3, \ldots, k-1)(2, 4, \ldots, k)$ if k is even, or $(1, 3, \ldots, k, 2, 4, \ldots, k-1)$ if k is odd. Thus, it is partitioned into 2 orbits if k is even, and 1 orbit if k is odd, and therefore its decomposition into transpositions contains $k-2$ transpositions if k is even, and $k-1$ if k is odd. Therefore, in order to put the leaves $v_{i+2,4}$ on the branches B_i for any i, at least $k-2$ transpositions are needed if k is even, and at least $k-1$ transpositions are needed if k is odd. Therefore, if ℓ is the number of inversions in S, then S has length at least $\ell + k - 2$ if k is even, and $\ell + k - 1$ if k is odd.

Let us now prove that S has length at least $2k - \ell$. Assume that S contains ℓ inversions. First note that, in both G_k and H_k, the vertices $v_{i,2}v_{i,3}$ appear in the same branch, but in G_k, $(c, v_{i,2}, v_{i,3})$ are aligned and in H_k, $(c, v_{i,3}, v_{i,2})$ are aligned. Thus, the order of $v_{i,2}v_{i,3}$ in the branch has to change during the transformation. The only way to change the order of $v_{i,2}$ and $v_{i,3}$ in a branch is to make an inversion of a subpath containing the edge $v_{i,2}v_{i,3}$. Since S contains only ℓ inversions, it means that there exist at least $k - \ell$ indices j for which the edge $v_{j,1}v_{j,2}$ has to belong to a flip before its inversion. Moreover, after each inversion, at most one core belongs to its final branch. Thus, there exist at least $k - \ell$ indices j such that the bad edge incident to $v_{j,2}$ has to belong to a flip after the inversion of $v_{j,1}v_{j,2}$ in order to connect it with $v_{j-1,1}$. So $2k - 2\ell$ internal edges have to be flipped during displacements.

Let us now focus on the leaves. Since S contains ℓ inversions, at most ℓ indices j satisfy that $v_{j,3}$ is incident to a leaf just before the inversion of $v_{j,2}v_{j,3}$. Since all the edges $v_{i,2}v_{i,3}$ have to be inversed during the transformation and since in G_k, every vertex $v_{i,3}$ is incident to a leaf, at least $k - \ell$ external edges have to be belong to a flip before the inversion of the core they are incident to in G_k. Similarly, at most ℓ indices j satisfy that $v_{j,2}$ is incident to a leaf just after the inversion of $v_{j,2}v_{j,3}$. Since in H_k, every vertex $v_{i,2}$ is incident to a leaf, at least $k - \ell$ external edges have to belong to a flip after the inversion of the core they are incident to in H_k. So at least $2k - 2\ell$ external edges have to be flipped during displacements.

Therefore, in total, we need to flip at least $4k - 4\ell$ edges during displacements. So at leat $2k - 2\ell$ displacements are needed in addition to the ℓ inversions, and the total number of flips in S is at least $2k - \ell$.

Thus, if k is even, S has length at least $\max(\ell + k - 2, 2k - \ell)$ and if k is odd, S has length at least $\max(\ell + k - 1, 2k - \ell)$. In both cases, the two lower bounds meet for $\ell = \lfloor \frac{k}{2} \rfloor + 1$ at the value $\lceil \frac{3k}{2} \rceil - 1$. Therefore, S has length at least $\lceil \frac{3k}{2} \rceil - 1$.

□

From Lemma 4.11, we can deduce the following:

Corollary 4.3. *There exist some connected graphs G and H with the same degree sequence for which, if we only flip edges of $\Delta(G, H)$, the shortest connected transformation from G to H has length at least $\frac{3}{2}(\frac{\delta(G,H)}{2} - mnc(G, H)) - 1$.*

5 Conclusion

In this chapter, we presented some results about the reconfiguration of connected multigraphs with the same degree sequence. We mostly focused on the case of trees, which is actually the hard case. We presented a result from a joint work with Nicolas Bousquet, where we study the shortest transformation problem and provide a polynomial time 2.5-approximation algorithm, which improves the former ratio of 4. This improvement is due to a new upper bound on the number of flips needed to transform a tree G into another H. On the other hand, we left the lower bound unchanged, as we used the one provided by Will's theorem [Wil99b] (which is the exact value of the length of a shortest transformation when we do not add the constraint of maintaining connectivity). We then discussed this lower bound. We showed that if we only flip bad edges of the same cycle of the symmetric difference then the length of a shortest transformation can be almost twice longer. To do so, we generalize some results of Christie [Chr98] proved for the particular case of paths, which is equivalent to the SORTING BY REVERSAL problem. We also show that if we only flip edges of the symmetric difference then the length of a shortest transformation can be almost $\frac{3}{2}$ times longer. So in order to drastically improve the approximation ratio, one first needs to improve the lower bound. Despite our efforts, we did not find a better lower bound and leave it as an open question. Similarly, we do not know for sure that flipping edges that are common to G and H cannot lead to a shortest transformation sequence, even if all the existing algorithms for SCGT and SORTING BY REVERSALS only consider flips on edges of the symmetric difference. These two open questions leave room for future improvements.

Chapter 5

Reconfiguration of Dominating Sets under the Token Addition-Removal rule

1 Introduction

In this chapter, we change the source problem as we investigate the reconfiguration of dominating sets. In this problem, the instance is a graph G, and the feasible solutions are dominating sets of G. Here, we focus on the Token Addition-Removal (TAR) rule. Recall that under TAR, two dominating sets D and D' are adjacent in the reconfiguration graph if there exists a vertex u such that $D' = D \cap \{u\}$ or $D' = D \setminus \{u\}$. An example of reconfiguration sequence is given in Figure 5.1.

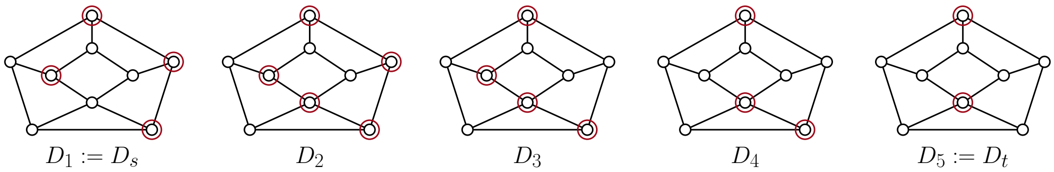


Figure 5.1: An example of reconfiguration sequence of dominating sets under the TAR rule.

Most of the studies concerning the reconfiguration of dominating sets have focused on the TAR rule (see [MN20] for a survey).

Haas and Seyffarth [HS14] first investigated the graphs that are realisable as reconfiguration graphs. They proved that the reconfiguration graph of the star of order n is S_n itself when the dominating sets are the ones of size at most 2. Alikhani et al. [AFK17] showed that this is the only case where a reconfiguration graph is isomoprhic to its instance. Alikhani et al. also showed that C_6, C_8, P_1 and P_3 are the only cycles and paths that are reconfiguration graphs of connected graphs, and that for any graph G, the order of the reconfiguration graph is odd when all the dominating sets of G are feasible solutions.

Haddadan et al.[HIM$^+$16] studied the algorithmic complexity of the REACHABILITY problematic. Recall that this problem consists in determining if there exists a reconfiguration sequence between two given dominating set D_s and D_t. They proved that this problem is PSPACE-complete, even when restricted to bipartite graphs, split graphs and planar graphs. However, they provide a linear algorithm if the input graph is a tree, an interval graph or a cograph.

Blanche et al.[BMOS19] focused on the complexity of an optimization variant of REACHABILITY. It consists in searching for the dominating set of minimum size among the ones that are reachable from a given dominating set D_s. They proved that this problem is PSPACE-complete even when restricted to bipartite graphs, split graphs and bounded pathwidth graphs. They also provide linear-time algorithms for trees, interval graphs and cographs.

Note that the size of the dominating sets is necessarily changing under TAR. Thus, as we stated in Chapter 3, we do not restrict it to a given k in the reconfiguration graph. That being said, we impose a maximum size k, which is called the threshold, and the feasible solutions are all the dominating sets which have size at most k. The reconfiguration graph for the instance G is then denoted by $\mathcal{R}_k(G)$. Note that if $k = n$, then we can always reconfigurate a dominating set D_s into another D_t by adding all the vertices of $D_t \setminus D_s$ to D_s, then removing all the vertices of $D_s \setminus D_t$. So $\mathcal{R}_n$ is connected. It raises the following question. What is the value of k from which it stops being the case ? In other words, what is the minimum value d_0 of k such that $\mathcal{R}_{k'}(G)$ is connected for any $k' \geq k$? This question has been raised by Haas and Seyffarth [HS14], and has been the object of many studies [HS14, SMN15, HS17, MTR19b].

Firstly, one might wonder if d_0 is also the minimum value of k such that $\mathcal{R}_k(G)$ is connected. Haas and Seyffarth actually proved that it is not necessarily the case [HS14]. They showed that being reconfigurable is not a monotone property, which means that if $\mathcal{R}_k(G)$ is connected then $\mathcal{R}_{k+1}(G)$ is not necessarily connected. To see it, let us consider the star S_n of order n. They observed that, for every $n \geq 4$, $\mathcal{R}_k(S_n)$ is connected if $1 \leq k \leq n-2$. But $\mathcal{R}_{n-1}(S_n)$ is not connected since the dominating set of size $n-1$ which contains all the leaves is frozen, i.e. it is an isolated vertex in $\mathcal{R}_{n-1}(S_n)$. However, they also proved the following, where $\Gamma(G)$ is the maximum size of an inclusion-wise minimal dominating set of G (in the rest of this chapter, we simply say minimal, in opposition to minimum).

Lemma 5.1. *[HS14] Let G be a graph. If $k > \Gamma(G)$ and $\mathcal{R}_k(G)$ is connected, then $\mathcal{R}_{k+1}(G)$ is connected.*

Moreover, they proved that if G has at least two independent edges (i.e. two edges that do not share a vertex), then $d_0 \leq \min\{n-1, \Gamma(G) + \gamma(G)\}$ (recall that $\gamma(G)$ is the minimum size of a dominating set of G). They also showed that this value can be lowered to $\Gamma(G) + 1$ if G is bipartite or a chordal graph. This result is tight since S_n is bipartite and chordal and $\mathcal{R}_{n-1}(S_n)$ is not connected, with $\Gamma(S_n) = n-1$. They then asked if this result can be generalized to any graph. Suzuki et al. [SMN15] answered negatively this question by constructing an infinite family of graphs for which $\mathcal{R}_{\Gamma(G)+1}(G)$ is not connected. Mynhardt et al. [MTR19b] improved this result by constructing two infinite families of graphs:

- the first construction provides some graphs G with arbitrary $\Gamma(G) \geq 3$ and arbitrary domination number in the range $2 \leq \gamma(G) \leq \Gamma(G)$ such that $d_0 = \Gamma(G) + \gamma(G) - 1$
- the second one gives some graphs G with arbitrary $\Gamma(G) \geq 3$ and arbitrary domination number in the range $1 \leq \gamma(G) \leq \Gamma(G) - 1$ such that $d_0 = \Gamma(G) + \gamma(G)$.

In particular, this provides an infinite family of graphs for which $d_0 = 2\Gamma(G) - 1$. The value of d_0 obtained by Mynhardt et al. is somehow the best we can hope for in the general case, since as we already stated, if G has at least two independent edges, then $d_0 \leq \min\{n-1, \Gamma(G) + \gamma(G)\}$ [HS14].

Suzuki et al. [SMN14] actually generalized the upper bound of $n-1$ on d_0 when G has at least two independent edges. They gave an upper bound depending on the matching number $\mu(G)$ of G. Recall that this corresponds to the maximum number of independent edges in G. They showed that if $k = n - \mu(G) + 1$, then $\mathcal{R}_k(G)$ is connected and has linear diameter. This result is somehow the best possible since $\mu(P_{2q}) = q$, and $\Gamma(P_{2q}) = q$ (where P_{2q} denotes the path of order $2q$), thus if $k = n - q = q$, then $\mathcal{R}_k(P_{2q})$ is disconnected (the minimal dominating sets are frozen).

Haas and Seyffarth [HS17] gave another upper bound on d_0, depending on the independence number of G. They proved that if $k = \Gamma(G) + \alpha(G) - 1$ (recall that $\alpha(G)$ is the maximum size of an independent set of G), then $\mathcal{R}_k(G)$ is connected. To obtain this result, they show that all the independent dominating sets of G are in the same connected component of $\mathcal{R}_{\Gamma(G)+1}(G)$.

Alikhani, Fatehi, and Mynhardt [AFM17] also studied the value of d_0, in the case where we restrict the solutions to total dominating sets (recall that in such a set, a vertex does not dominate itself). In particular, they characterized the graphs for which $\mathcal{R}_{\Gamma(G)^t}(G)$ is connected, where $\Gamma(G)^t$ is the maximum size of an inclusion-wise minimum total dominating set.

In this chapter, we present a joint work with Nicolas Bousquet and Paul Ouvrard, in which we complete these results by giving other upper bounds on d_0 depending on several graph parameters. In each case, we provide a linear transformation between any pair of dominating sets, thus proving that the reconfiguration graph has linear diameter when $k \geq d_0$.

2 Independence number

In this section, we study the upper bound on d_0 depending on the maximum size of a minimal dominating set of G and the independence number of G. We present a result that improves the following upper bound on d_0, by Haas and Seyffarth.

Lemma 5.2. *[HS17] Let G be a graph. If $k \geq \Gamma(G) + \alpha(G) - 1$, then $\mathcal{R}_k(G)$ is connected.*

In our result, we give a new proof of the same result, but that moreover implies that the reconfiguration graph has linear diameter. The proof is constructive and provides an algorithm that constructs a path between two given dominating sets of size at most k of G.

Firstly, let us show some basic observations about maximal independent sets (i.e. independent sets that are maximal by inclusion) which will be needed for the proof.

As stated in Chapter 2, computing a maximum independent set of a given graph G is a classical NP-complete problem [Kar72], while computing a maximal one can trivially be done in linear time by a greedy algorithm [Lub86]. Moreover, given an independent set S' which is not maximal, one can greedily complete it into a maximal independent set S such that $S' \subseteq S$. In particular, if there exist two vertices u and v such that $uv \notin E$, then there exists a maximal independent set of G which contains both u and v. Obviously, this is also true when S' is reduced to a single vertex. We will use this fact to prove the result we present in this section, as well as for the following well-known observation.

Observation 5.1. *Let $G = (V, E)$ be a graph, and $S \subseteq V$ be a maximal independent set of G. Then, S is an inclusion-wise minimal dominating set.*

Proof. Let $u \in V$ be a vertex. If $u \in S$, u is dominated by itself. Otherwise, there exists $v \in N(u) \cap S$ since S is maximal. Hence, u is dominated by v. Moreover, by definition of an independent set, we have $N(S \setminus u)$ does not contain u for every vertex $u \in S$. Therefore, u is not dominated in $S \setminus \{u\}$ and thus S is a minimal dominating set of G. □

Note that Observation 5.1 implies that any inclusion-wise maximal independent set S of G satisfies $|S| \leq \alpha(G) \leq \Gamma(G)$. We also need the following observation.

Observation 5.2. *Let D be a minimal dominating set of G, and let S be a maximal independent set of G such that $D \cap S \neq \emptyset$. If $k = \Gamma(G) + \alpha(G) - 1$, then there exists a* TAR*(k)-reconfiguration sequence between D and S of length at most $|D| + \alpha(G) - 2$.*

Proof. Recall that since S is a maximal independent set, $|S| \leq \alpha(G) \leq \Gamma(G)$. We first add to D each vertex in $S \setminus D$ one by one. Note that there are at most $\alpha(G) - 1$ such vertices. We thus obtain the set $D' = D \cup S$. We then remove one by one each vertex in $D \setminus S$. There are at most $|D| - 1$ such vertices since $S \cap D \neq \emptyset$. Each intermediate solution is indeed a dominating set since it either contains D or S which are both dominating sets. Moreover, each solution is of size at most $|D'| \leq |D| + |S| - 1 \leq k$. □

We now move on to one of the main results of the work done with Nicolas Bousquet and Paul Ouvrard, which is about the diameter of the reconfiguration graph when $k = \Gamma(G) + \alpha(G) - 1$.

Theorem 5.1. *Let $G = (V, E)$ be a graph on n vertices. If $k = \Gamma(G) + \alpha(G) - 1$ then $\mathcal{R}_k(G)$ has diameter at most $10n$.*

Proof. Let D_1 and D_2 be two dominating sets, both of size at most k. Free to remove at most $2 \cdot (\Gamma(G) + \alpha(G) - 2)$ vertices in total, one can assume without loss of generality that D_1 and D_2 are both inclusion-wise minimal dominating sets of G. Hence $|D_1| \leq \Gamma(G)$ and $|D_2| \leq \Gamma(G)$. We outline a path between D_1 and D_2 in $\mathcal{R}_k(G)$. The next claim deals with the case where D_1 and D_2 have a non-empty intersection.

Claim 5.1. *If $D_1 \cap D_2 \neq \emptyset$ then there exists a reconfiguration sequence from D_1 to D_2 of length at most $2 \cdot (\alpha(G) + \Gamma(G) - 2)$.*

Proof. Let x be a vertex that belongs to both D_1 and D_2. One first constructs greedily (and thus in polynomial-time) a maximal independent set S of G which contains x (which is then of size at most $\alpha(G)$). By Observation 5.2, one can transform D_1 into S under the TAR rule. And the length of the reconfiguration sequence is at most $\Gamma(G) + \alpha(G) - 2$. Similarly, there exists a reconfiguration sequence of length at most $\Gamma(G) + \alpha(G) - 2$ from D_2 to S. By combining these two transformations, we obtain a reconfiguration sequence between D_1 and D_2 of length at most $2 \cdot (\alpha(G) + \Gamma(G) - 2)$, as desired. $\Diamond$

In the remainder of this proof, we assume that $D_1 \cap D_2 = \emptyset$ otherwise we can directly conclude the proof by Claim 5.1. If there exist $u_i \in D_1$ and $v_j \in D_2$ such that the set $D' = (D_1 \setminus \{u_i\}) \cup \{v_j\}$ is a dominating set of G, then we can conclude the proof by Claim 5.1 since $D' \cap D_2 \neq \emptyset$ and D' can be obtained from D_1 in two steps. Suppose now that $D' = (D_1 \setminus \{u_i\}) \cup \{v_j\}$ is not a dominating set of G. This means that u_i is adjacent to a vertex x with no neighbors in $(D_1 \setminus \{u_i\}) \cup \{v_j\}$. Hence, there exists a maximal independent set S_1 of G which contains both x and a vertex $u_k \in D_1 \setminus \{u_i\}$. Similarly, there exists a maximal independent set S_2 which contains both x and v_j. By Observation 5.2, there exists a reconfiguration sequence of length at most $\Gamma(G) + \alpha(G) - 2$ between S_1 (respectively S_2) and D_1 (respectively D_2) under the TAR rule. Finally, since S_1 and S_2 intersect, we can again use Observation 5.2 that ensures that there exists a transformation from S_1 to S_2 of length at most $2\alpha(G) - 2$.

Hence, we obtain a reconfiguration sequence from D_1 to D_2 of length at most $4 \cdot (\Gamma(G) + \alpha(G) - 2) + 2 \cdot (\alpha(G) - 1) < 10n$. $\square$

3 H-minor free graphs

In this section, we present another upper bound on d_0 for minor-free graphs. We say that a graph is *d-minor sparse* if all its bipartite minors have average degree less than d. Note that it is equivalent to say that the ratio between the number of edges and the number of vertices of any bipartite minor of G is strictly less than $\frac{d}{2}$. We first present the following essential lemma.

Lemma 5.3. *Let G be a d-minor sparse graph. Let A and B be two dominating sets of G such that $|A| = |B|$ and $|B \setminus A| \geq d$. Then, there exists a vertex $a \in A \setminus B$ and a set $S \subset B \setminus A$ with $|S| = d - 1$ such that $(A \cup S) \setminus \{a\}$ is a dominating set of G.*

Proof. We prove it by contradiction. For every $a_i \in A \setminus B$, let $S_{i,1}$ be a subset of $B \setminus A$ of size $d - 1$. Let $x_{i,1}$ be a vertex that is only dominated by a_i in A and not dominated by $S_{i,1}$ in B (such a vertex must exist otherwise the conclusion follows). Note that this vertex can be a vertex of A, a vertex of B, a vertex of both or a vertex of neither. Let $b_{i,1}$ be a vertex of $(B \setminus A) \setminus S_{i,1}$ that dominates $x_{i,1}$. This vertex exists since B is a dominating set and $x_{i,1}$ is only dominated by a_i in A. Now, for every $2 \leq j \leq d$, we define recursively the set $S_{i,j}$ as a subset of size $d - 1$ of $B \setminus A$ containing $\{b_{i,1}, \ldots, b_{i,j-1}\}$. We let $x_{i,j}$ be a vertex only dominated by a_i in A that is not dominated by $S_{i,j}$ in B, and $b_{i,j}$ be a vertex of $(B \setminus A) \setminus S_{i,j}$ that dominates $x_{i,j}$. Note that, for every j, since $x_{i,j}$ is incident to $b_{i,j}$ and not to $S_{i,j}$, $b_{i,j} \notin \{b_{i,1}, \ldots, b_{i,j-1}\}$. In particular,

$B_i := \{b_{i,1}, \ldots, b_{i,d}\}$ has size exactly d. Note that $B_i \subseteq B \setminus A$. The construction of the set B_i is illustrated in Figure 5.2.

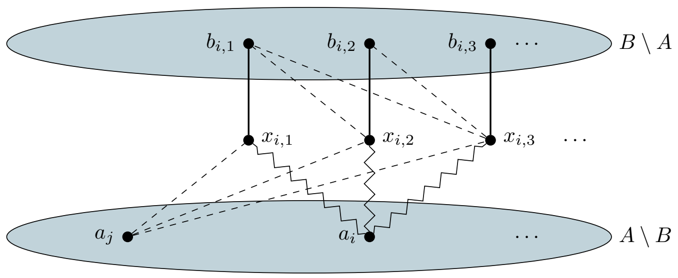


Figure 5.2: The set B_i. The dotted lines represent the non-edges, and the zigzags represent the edges that are contracted in G'.

Let us construct a minor G' of G of density at least d. In this minor, every vertex a_i in $A \setminus B$ will be adjacent to every vertex of B_i. To that end, for every $a_i \in A \setminus B$, we contract the edges $a_i x_{i,j}$ for any j such that $x_{i,j} \notin B \setminus A$ and $x_{i,j} \notin A \setminus B$. If $x_{i,j} \in A \setminus B$, then $x_{i,j} = a_i$ and a_i is already adjacent to $b_{i,j}$, so no contraction is needed. If $x_{i,j} \in B \setminus A$, then by construction $x_{i,j} = b_{i,j}$ and no contraction is needed. By abuse of notations, we still denote by a_i the vertex resulting from the contractions involving a_i. Note that the vertices $x_{i,j}$ are pairwise disjoint. If $x_{i,j} = x_{i',j'}$ then, since $x_{i,j}$ is only dominated by a_i and $x_{i',j'}$ by a'_i, we must have $a_i = a'_i$. And by construction in the previous paragraph, $x_{i,j} \neq x_{i,j'}$ if $j \neq j'$. So the contractions above are well defined. Moreover, the size of $A \setminus B$ is left unchanged. Similarly the size of $B \setminus A$ is not modified. We finally remove from the graph any vertex which is not in $(A \setminus B) \cup (B \setminus A)$, and any edge internal to $A \setminus B$ or to $B \setminus A$. The resulting graph G' is a minor of G and is bipartite.

For every i and every vertex v in B_i, there exists a j such that v is adjacent to $x_{i,j}$ or a_i in G. Thus, a_i is adjacent to every vertex of B_i in G'. Therefore, for any $a_i \in A \setminus B$, a_i has degree at least d in G'. Thus, there are at least $d \cdot |A \setminus B|$ edges in G'. Since G' has $|A \setminus B| + |B \setminus A| = 2|A \setminus B|$ vertices, it contradicts the fact that G is a d-minor sparse graph. □

Lemma 5.4. *Let G be a d-minor sparse graph. If $k = \Gamma(G) + d - 1$, then $\mathcal{R}_k(G)$ is connected and the diameter of $\mathcal{R}_k(G)$ is at most $2\Gamma(G) \cdot (d-1) + 2 \cdot \max(\Gamma(G) - 1, d - 1)$.*

Proof. Firstly, if $d > \Gamma(G)$, then the result follows from Theorem 5.1. So we assume $d \leq \Gamma(G)$. We proceed by induction on $|D_t \setminus D_s|$. Let D_s and D_t be two dominating sets of G of size at most k. Since $\Gamma(G)$ is the maximum size of a dominating set minimal by inclusion, we can add or remove vertices from D_s and D_t so that D_s and D_t both have size exactly $\Gamma(G)$, while still remaining dominating sets. Note that by assumption, we need to remove or add at most $2(\Gamma(G) - 1)$ vertices in total. So from now on, we assume that $|D_s| = |D_t| = \Gamma(G)$. Let us show that there is a path from D_s to D_t in $\mathcal{R}_k(G)$ of length at most $2|D_t \setminus D_s| \cdot (d-1)$. Since $|D_t \setminus D_s| \leq \Gamma(G)$, and by taking into account the $2(\Gamma(G) - 1)$ vertices eventually added or removed at the beginning, this will give the desired result. We proceed by induction on $|D_t \setminus D_s|$.

If $|D_t \setminus D_s| \leq d - 1$ then, since $|D_s| = \Gamma(G)$, we have $|D_s \cup D_t| \leq \Gamma(G) + d - 1$. Thus, we can simply add all the vertices of $D_t \setminus D_s$ to D_s and then remove the vertices of $D_s \setminus D_t$. We thus obtain a path from D_s to D_t in $\mathcal{R}_k(G)$ of length at most $2d - 2 \leq 2|D_t \setminus D_s| \cdot (d-1)$.

Assume now that $|D_t \setminus D_s| \geq d$. By Lemma 5.3, there exists a vertex $v \in D_s \setminus D_t$ and a set $S \subset D_t \setminus D_s$ with $|S| = d - 1$ such that $D'_s := (D_s \cup S) \setminus \{v\}$ is a dominating set of G. Let D''_s be any dominating set of size exactly $\Gamma(G)$ obtained by removing vertices of D'_s, i.e. such that $D''_s \subseteq D'_s$. Since $|S| = d - 1$ and $|D_s| = \Gamma(G)$, the transformation that consists in adding every vertex of S to D_s and then removing v and every vertex of $D'_s \setminus D''_s$ is a path from D_s to D''_s in $\mathcal{R}_k(G)$. Moreover, $|D'_s| = \Gamma(G) + d - 2$. Thus, this path has length $2d - 2$.

We have $D'_s := (D_s \cup S) \setminus \{v\}$ where $v \in D_s \setminus D_t$ and $S \subset D_t \setminus D_s$ with $|S| = d - 1$. Thus, $|D_t \setminus D'_s| = |D_t \setminus D_s| - d + 1$. Since $D''_s \subseteq D'_s$ and $|D'_s \setminus D''_s| \leq d - 2$, it gives $|D_t \setminus D''_s| \leq |D_t \setminus D_s| - 1$. By the induction hypothesis, there exists a path from D''_s to D_t in $\mathcal{R}_k(G)$ of length at most $|D_t \setminus D''_s| \cdot (2d - 2)$. The concatenation of the two paths gives a path from D_s to D_t in $\mathcal{R}_k(G)$ of length at most $2|D_t \setminus D_s| \cdot (d - 1)$. This concludes the proof. □

Let us now state two of our main results, which are immediate corollaries of Lemma 5.4:

Corollary 5.1. *Let G be a graph. Then, we have the following:*

- *if G is planar, then $\mathcal{R}_k(G)$ is connected and has linear diameter for every $k \geq \Gamma(G) + 3$.*
- *if G is K_ℓ-minor free, then there exists a constant C such that $\mathcal{R}_k(G)$ is connected and has linear diameter for every $k \geq \Gamma(G) + C\ell\sqrt{\log_2 \ell}$.*

Proof. Every minor of a planar graph is planar. Moreover every bipartite planar graph has at most $2n - 4$ edges. Thus every planar graph is a 4-minor sparse graph and the first point follows from Lemma 5.4.

A result of Thomason[Tho84] (improving a result of Mader[Mad68]) ensures that the average degree of a K_ℓ-minor free graph is at most $0.265 \cdot \ell\sqrt{\log_2 \ell}(1 + o(1))$. In particular, there exists a constant C such that, for every ℓ and every K_ℓ-minor free graph G, the average degree of G is at most $C\ell\sqrt{\log_2 \ell}$. Thus G is $C\ell\sqrt{\log_2 \ell}$-minor sparse and the second point follows from Lemma 5.4. □

Concerning planar graphs, the result we give is almost tight. Indeed, we know that there exist planar graphs for which $\mathcal{R}_k(G)$ is not connected if $k = \Gamma(G) + 1$. Suzuki et al [SMN15] gave an example of such a planar graph G. The graph G is given in Figure 5.3.

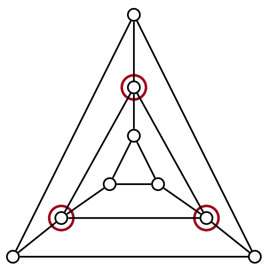


Figure 5.3: The planar graph G such that $\mathcal{R}_{\Gamma(G)+1}$ is not connected.

It is easily seen that $\Gamma(G) = 3$. Moreover, if we consider the circled dominating set, in order to remove a vertex, we must add the other vertices it is adjacent to, thus reaching a dominating set of size $\Gamma(G) + 2$. Despite our efforts, we were not able to find an example where $\Gamma(G) + 3$ is needed for the reconfiguration graph to be connected. Thus, we conjecture the following.

Conjecture 5.1. *If G is planar and if $k \geq \Gamma(G) + 2$, then $\mathcal{R}_k(G)$ is connected.*

4 Bounded treewidth graphs

In this section, we present our last result, which provides an upper bound on d_0 depending on $\Gamma(G)$ and the treewidth of G.

Theorem 5.2. *Let $G = (V, E)$ be a graph. If $k = \Gamma(G) + tw(G) + 1$, then $\mathcal{R}_k(G)$ is connected. Moreover, the diameter of $\mathcal{R}_k(G)$ is at most $4(n + 1) \cdot (tw(G) + 1)$.*

Proof. Let (X, T) be a tree decomposition of G such that the maximum size of a bag of X is $tw(G) + 1$. Let $b = |X|$. We root the tree T in an arbitrary bag, then set $X := \{X_1, \ldots, X_b\}$, where for any X_i, X_j such that X_i is a child of X_j, we have $i < j$. In other words, $X_1, \ldots, X_b$ is an elimination ordering of the (rooted) tree T where at each step we remove a leaf of the remaining tree. We say that a bag X_i is a *descendant* of X_j if X_j is on the path from the root to X_i (in other words, X_i belongs to the subtree rooted in X_j in T). Note that, free to contract edges if a bag is included in another, we can assume $b \leq n$. We denote by V_i the set of vertices that do not appear in the set of bags $\cup_{j=i+1}^{b} X_j$. We set $V_0 := \emptyset$.

Let D_s and D_t be two dominating sets. Free to first remove vertices from D_s and D_t if possible (which can be done in at most $2(tw(G) + 1)$ operations in total), we can assume that D_s and D_t have size at most $\Gamma(G)$. Let D be a minimum dominating set of G. Instead of proving directly that there exists a reconfiguration sequence from D_s to D_t, we will prove that that there exists a reconfiguration sequence from D_s to D and from D_t to D of length at most $2n \cdot (tw(G) + 1)$ each. Since the reverse of a reconfiguration sequence also is reconfiguration sequence, that will give the conclusion, that gives a reconfiguration sequence of the desired length. So the rest of the proof is devoted to prove the following:

Lemma 5.5. *Let $G = (V, E)$ be a graph and let D_s be a dominating set of G of size at most $\Gamma(G)$ and D be a minimum dominating set of G. If $k = \Gamma(G) + tw(G) + 1$, then there is a reconfiguration sequence from D_s to D. Moreover, the length of this reconfiguration sequence is at most $2n \cdot (tw(G) + 1)$.*

In order to prove Lemma 5.5, we prove that there exists a sequence $\langle D_1 := D_s, D_2, \ldots, D_b\rangle$ of dominating sets such that, for every j, D_j satisfies the following property $\mathcal{P}$:

- **(i)** D_j is a dominating set of G of size at most $\Gamma(G)$,
- **(ii)** For every $j > 1$, there exists a transformation sequence of length at most $2(tw(G) + 1)$ from D_{j-1} to D_j in $\mathcal{R}_k(G)$,
- **(iii)** $D_j \cap V_{j-1} \subseteq D$. In other words, the vertices of D_j that only belong to bags in $X_1 \cup \ldots \cup X_{j-1}$ are also in D.

So that will provide a reconfiguration sequence in $\mathcal{R}_k(G)$ from D_s to a dominating set D_b sufficiently close to D to ensure the existence of a transformation from D_b to D of length at most $2n \cdot (tw(G) + 1)$. To prove the existence of the sequence, we use induction on j.

First note that since D_s is a dominating set of G of size at most $\Gamma(G)$ and V_0 is empty, D_s satisfies property $\mathcal{P}$. Let us now show that if D_j satisfies property $\mathcal{P}$, then there exists a set D_{j+1} that satisfies property $\mathcal{P}$. A vertex v is a *left vertex* (for X_j) if v only appears in bags that are descendant of X_j. Note that by definition, X_j is a descendant of itself. Otherwise, we say that v is a *right vertex*. When no confusion is possible, we will omit the mention of X_j.

Claim 5.2. *If a left vertex u (for X_j) is adjacent to a right vertex v (for X_j), then $v \in X_j$.*

Proof. Since u and v are adjacent in G, there exists a bag X_i which contains both u and v. Note that since u is a left vertex, X_i is a descendant of X_j. Besides, since v is a right vertex, there exists a bag $X_{i'}$ that contains v and which is not a descendant of X_i. Since the set of bags that contain v induces a connected tree, v must belong to each bag on the path from X_i to $X_{i'}$. In particular, $v \in X_j$. $\Diamond$

To construct D_{j+1}, we define several subsets of vertices (see Figure 5.4 for an illustration).

- A is the set of left vertices of $X_j \cap (D_j \setminus D)$. In other words, A is the set of left vertices of X_j that are in D_j but not in D.
- B is the set of right vertices of X_j. In other words, B is the set of vertices of X_j that also appear in a bag $X_{j'}$ with $j' > j$.
- C is the set of left vertices of $D \setminus D_j$. In other words, C is the set of vertices of D at the left of X_j that are missing in D_j.

We partition again B into three parts:

- B_1 is the set of vertices of $B \setminus D$ that are dominated by C

- $B_2 = B \cap D_j$
- $B_3 = B \setminus (B_1 \cup B_2)$.

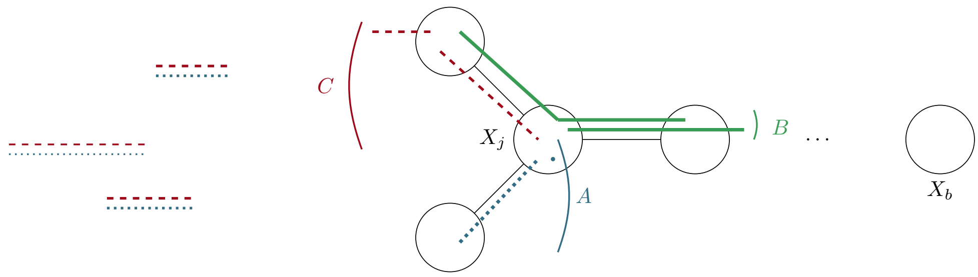


Figure 5.4: The tree decomposition of G, and the sets A, B and C. The circles represent the bags of the tree decomposition. The vertices are represented by lines, or dots, that go through the bags they belong to. The thick full lines represent the vertices of B, the dashed lines represent the vertices of D, and the dotted lines represent the vertices of D_j. By the induction hypothesis, the left vertices of D_j that do not belong to X_j belong to D.

We set $D'_j = (D_j \setminus A) \cup C \cup B_3$. Let us first prove that D'_j is a dominating set of G.

Claim 5.3. *The set D'_j is a dominating set of G.*

Proof. Since D_j is a dominating set of G and $D_j \setminus A \subseteq D'_j$, the only vertices that can be undominated in D'_j are the ones dominated only by vertices of A in D_j. Let $N_r(A)$ (resp. $N_l(A)$)) be the right vertices (resp. left vertices) that are only dominated by A in D_j. Note that $N_l(A)$ might contain vertices of A, while $N_r(A)$ does not, since by definition the vertices of A are left vertices. Let us show that all the vertices in $N_r(A) \cup N_l(A)$ are dominated by D'_j.

We start with $N_r(A)$. Since the vertices of A are left vertices and the vertices of $N_r(A)$ are right vertices, by Claim 5.2, we have $N_r(A) \subseteq X_j$. Since the vertices in $N_r(A)$ are right vertices, we have $N_r(A) \subseteq B$. Moreover, since every vertex of $N_r(A)$ is only dominated by A in D_j but does not belong to A, it is not in D_j and thus not in B_2. Thus, the vertices of $N_r(A)$ either belong to B_1 (and are by definition dominated by C), or they belong to B_3. Therefore, $N_r(A)$ is dominated by $C \cup B_3$ and thus by D'_j.

Let us now focus on $N_l(A)$. In D, $N_l(A)$ is dominated by vertices that we partition into two sets: the right vertices Y and the left vertices Z. We show that both Y and Z are included in D'_j, which implies that D'_j dominates $N_l(A)$. Since the vertices of $N_l(A)$ are left vertices and the vertices of Y are right vertices, Claim 5.2 gives $Y \subseteq X_j$. Thus, by definition, $Y \subseteq B$. Moreover, the vertices of Y that belong to D_j do not belong to A as they are right vertices, and thus belong to $D_j \setminus A$, and the vertices of Y that do not belong to D_j belong by definition to $B \cap (D \setminus D_j) \subseteq B_3$. Thus, $Y \subseteq (D_j \setminus A) \cup B_3 \subseteq D'_j$. Finally, the vertices of Z either belong to D_j and thus by definition to $D_j \cap D \subseteq D_j \setminus A$, or they do not belong to D_j and by definition they thus belong to C. Therefore, $Z \subseteq (D_j \setminus A) \cup C \subseteq D'_j$. Therefore, $N_l(A)$ is dominated by D'_j, which concludes the proof of this claim. $\Diamond$

Let us now prove the following:

Claim 5.4. $|D_j \cup C \cup B_3| \leq \Gamma(G) + tw(G) + 1$.

Proof. Let us first show that the set $D' := (D \setminus C) \cup A \cup B_1 \cup B_2$ is a dominating set of G. We will then explain how to exploit this property to prove that $|D_j \cup C \cup B_3| \leq \Gamma(G) + tw(G) + 1$.

Since D is a dominating set, the only vertices that can be undominated in $(D \setminus C) \cup A \cup B_1 \cup B_2$ are vertices that are only dominated by C in D. Let $N_r(C)$ (resp. $N_l(C)$) be the subset of right (resp. left) vertices that are only dominated by C in D. Note that $N_l(C)$ might contain vertices of C and $N_r(C)$ does not, since the vertices of C are left vertices. We prove that $N_r(C)$ and $N_l(C)$ are dominated by D'.

We first prove that the vertices of $N_r(C)$ are dominated in D'. Since C only contains left vertices and $N_r(C)$ only contains right vertices, Claim 5.2 ensures that $N_r(C) \subseteq X_j$. Thus, by definition of B, $N_r(C) \subseteq B$. Since the vertices of $N_r(C)$ are only dominated by C in D, $N_r(C) \subseteq B_1$. Therefore $(D \setminus C) \cup A \cup B_1 \cup B_2$ dominates $N_r(C)$.

Let us now prove that $N_l(C)$ is dominated in D'. Every vertex $v \in N_l(C)$ is dominated in D_j by either a right vertex or a left vertex. Assume that v is dominated in D_j by a right vertex w. Since v is a left vertex and w a right vertex, Claim 5.2 ensures that $w \in X_j$ and thus $w \in B$. Since $w \in D_j$, $w \in B_2 \subseteq D'$. Assume now that v is dominated in D_j by a left vertex u. If u belongs to D, it is in $D \cap D_j \subseteq D \setminus C \subseteq D'$. So we can assume that $u \notin D$. By the induction hypothesis, D_j satisfies (iii) and since $u \notin D$, the vertex u necessarily belongs to X_j. So we finally have $u \in A$. Thus, $u \in (D \setminus C) \cup A \subseteq D'$. So $N_l(C)$ is dominated in D'. And then D' is a dominating set of G.

We can now show that $|D_j \cup C \cup B_3| \leq \Gamma(G) + tw(G) + 1$. Since D is a minimum dominating set of G and $D' = (D \setminus C) \cup (A \cup B_1 \cup B_2)$ also is a dominating set of G, we have $|C| \leq |A \cup B_1 \cup B_2|$. Thus, $|C \cup B_3| \leq |A| + |B_1 \cup B_2| + |B_3|$. But A, $B_1 \cup B_2$ and B_3 are pairwise disjoint subsets of X_j. Thus, $|A| + |B_1 \cup B_2| + |B_3| \leq |X_j| \leq tw(G) + 1$, and $|C \cup B_3| \leq tw(G) + 1$. Since, by the induction hypothesis, D_j has size at most $\Gamma(G)$, this gives $|D_j \cup C \cup B_3| \leq \Gamma(G) + tw(G) + 1$. $\Diamond$

We now have a reconfiguration sequence of length at most $tw(G) + 1$ from D_j to D'_j by simply adding all the vertices of $C \cup B_3$ and then removing all the vertices of A. All along the sequence, the corresponding set is dominating. Indeed, it contains D_j during the first part and D'_j during the second one. One is dominating by assumption and the other is dominating by Claim 5.3. By Claim 5.4, this reconfiguration sequence exists in $\mathcal{R}_{\Gamma(G)+tw(G)+1}(G)$.

The dominating set D_{j+1} will be any dominating set of size at most $\Gamma(G)$ obtained from D'_j by removing vertices, i.e. any dominating set D_{j+1} satisfying $D_{j+1} \subseteq D'_j$ and $|D_{j+1}| = \Gamma(G)$, which necessarily exist by definition of $\Gamma(G)$. This can be done in at most $tw(G) + 1$ deletions. Thus, there exist a sequence in $\mathcal{R}_{\Gamma(G)+tw(G)+1}(G)$ from D_j to D_{j+1} of length at most $2(tw(G)+1)$, and D_{j+1} thus satisfies (i) and (ii). Let us now justify why D_{j+1} satisfies (iii).

Since D_{j+1} is a subset of D'_j, if (iii) holds for D'_j it holds for D_{j+1}. We have $D'_j = (D_j \setminus A) \cup C \cup B_3$. Since $C \subseteq D$, if a left vertex v (for X_j) appears in D'_j but not in D, it is either in $D_j \setminus A$ or in B_3. Since B_3 only contains right vertices, it must be in $D_j \setminus A$. Since A contains the left vertices of $X_j \cap (D_j \setminus D)$, it means that v should be in V_{j-1}. But, by the induction hypothesis, the vertices of D_j that belong to V_{j-1} belong to D. So v does not exists and D'_j satisfies (iii). Thus, D_{j+1} satisfies property $\mathcal{P}$, and by induction, there exists a set D_b that satisfies property $\mathcal{P}$. Moreover, since for any i such that $2 \leq i \leq b$, there is a path of length at most $2(tw(G) + 1)$ from D_{i-1} to D_i in $\mathcal{R}_k(G)$, there is transformation of length at most $2(b - 1) \cdot (tw(G) + 1)$ from D_s to D_b in $\mathcal{R}_k(G)$.

To complete the construction of a path from D_s to D in $\mathcal{R}_k(G)$, we show that there exists a transformation from D_b to D in $\mathcal{R}_k(G)$ of length at most $2(tw(G) + 1)$. Let $A' = D_b \setminus D$, and $C' = D \setminus D_b$. We have $D = (D_b \cup C') \setminus A'$. Let S'_1 be the reconfiguration sequence from D_b to $D_b \cup C'$ which consists in adding one by one every vertex of C'. Since each of the sets of S'_1 contains D_b, they are all dominating sets of G. Note that S'_1 has length $|C'|$. Let S'_2 be the reconfiguration sequence from $D_b \cup C'$ to D which consists in removing one by one each vertex of A'. Since each of the sets of S'_2 contains D, they all are dominating sets. Note that S'_2 has length $|A'|$. Thus, applying S'_1 then S'_2 gives a reconfiguration sequence from D_b to D of length $|C'| + |A'|$. Moreover, the maximum size of a dominating set reached in this sequence is $|D_b \cup C'|$. Let us show that $|D_b \cup C'| \leq \Gamma(G) + tw(G) + 1$. We have $D_b = (D \setminus C') \cup A'$. Thus, since D is a minimum dominating set, $|C'| \leq |A'|$. Since D_b satisfies (iii), every vertex of D_b that does not belong to X_b also belongs to D. Thus, $A' \subseteq X_b$, and $|A'| \leq tw(G) + 1$, which gives $|C'| \leq tw(G) + 1$, as well as $|C'| + |A'| \leq 2(tw(G) + 1)$. Since D_b is a minimal dominating set of G, we have therefore $|D_b \cup C'| \leq \Gamma(G) + tw(G) + 1$. Thus, there is a path of length at most $2(tw(G) + 1)$ from D_b to D in $\mathcal{R}_k(G)$ which completes the transformation of length at most $2b \cdot (tw(G) + 1)$ from D_s to D in $\mathcal{R}_k(G)$. Since $b \leq n$, the conclusion follows. $\square$

The upper bound given by Theorem 5.2 is tight up to an additive constant factor. Indeed, Mynhardt et al. [MTR19b] constructed an infinite family of graphs $G_{\ell,r}$ (with $\ell \geq 3$ and $1 \leq r \leq \ell - 1$) for which $2\Gamma(G) - 1$ tokens are necessary to guarantee the connectivity of the reconfiguration graph. Let us describe their construction when $r = \ell - 1$. The graph $G_{\ell,\ell-1}$ contains $\ell - 1$ cliques $C_1, C_2, \dots, C_{\ell-1}$ called *inner cliques*, each of size ℓ. We denote by c_i^j the j-th vertex of the clique C_i. We then add a new clique C_0 of size ℓ, called the *outer clique* and we add a new vertex u_0 adjacent to all the vertices of C_0 (hence, C_0 can be seen as a clique of size $\ell + 1$). For every $1 \leq i \leq \ell - 1$ and for every $1 \leq j \leq \ell$, we add an edge between c_i^j and c_0^j. This completes the construction of $G_{\ell,\ell-1}$ (see Figure 5.5 for an example). Mynhardt et al. [MTR19b] showed that $\Gamma(G_{\ell,\ell-1}) = \ell$.

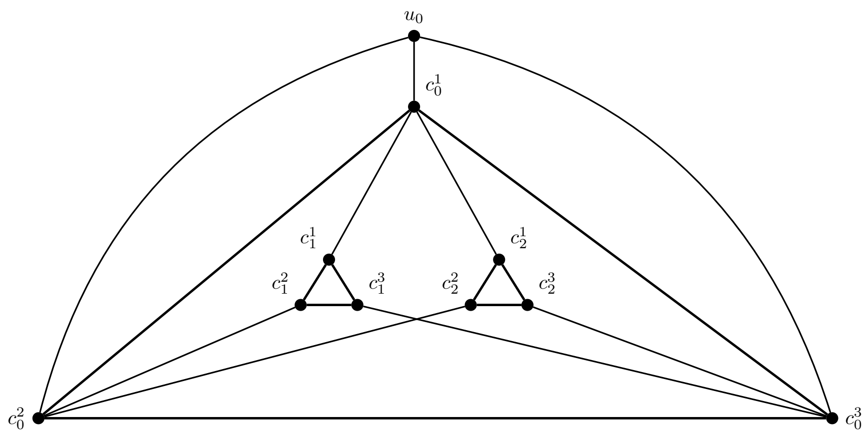


Figure 5.5: The graph $G_{3,2}$

They moreover show that $\mathcal{R}_{2\ell-2}(G_{\ell,\ell-1})$ is not connected. Let us prove that $G_{\ell,\ell-1}$ has treewidth ℓ. This implies that $\mathcal{R}_{\Gamma(G)+tw(G)-2}$ is not necessarily connected, and that our function of the treewidth is tight up to an additive constant factor.

Claim 5.5. *The graph $G_{\ell,\ell-1}$ has treewidth ℓ.*

Proof. First, observe that $tw(G_{\ell,\ell-1}) \geq \ell$ since $G[C_0 \cup \{u_0\}]$ is a clique of size $\ell + 1$.

Let us now give a tree decomposition of $G_{\ell,\ell-1}$ of width ℓ. We first create a "central" bag B_0 containing all the vertices of C_0 and the vertex u_0. For each inner clique C_i with $1 \leq i \leq \ell - 1$, we attach to B_0 a path $B_i^1 B_i^2 \dots B_i^\ell$ where B_i^j contains the vertices $(C_0 \setminus \bigcup_{k=0}^{j-1} c_0^k) \cup \bigcup_{k=1}^{j} c_i^k$ (see Figure 5.6 for an example). Observe that for any $1 \leq i \leq \ell - 1$, the bag B_i^ℓ contains all the vertices of C_i. And the bag B_i^j contains both c_0^j and c_i^j. Hence, each edge is contained in at least one bag. For every $1 \leq j \leq \ell$, the vertex c_0^j is contained in the bags $B_0 \cup \bigcup_{i=1}^{\ell-1} \bigcup_{k=1}^{j} B_i^k$. And for every $1 \leq i \leq \ell - 1$ and every $1 \leq j \leq \ell$, the vertex c_i^j is contained in $B_i^1, B_i^2, \dots, B_i^j$. It follows that for every vertex $u \in V(G_{\ell,\ell-1})$ the set of bag containing u induces a connected subtree. Finally, one can easily check that each bag contains exactly $\ell + 1$ vertices. Hence, this decomposition indeed is a tree decomposition of $G_{\ell,\ell-1}$ of width ℓ and the conclusions follows. $\Diamond$

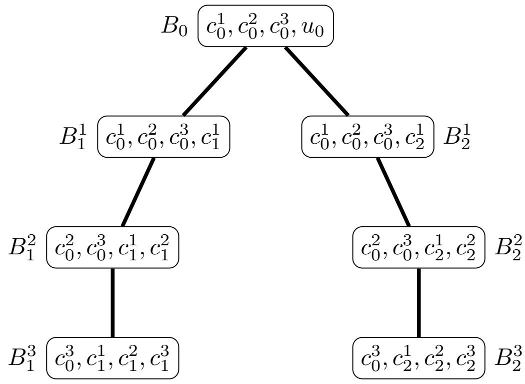


Figure 5.6: Tree decomposition of $G_{3,2}$ of width $tw(G_{3,2})$.

On the other hand, concerning the pathwidth of $G_{\ell,\ell-1}$, we have the following.

Claim 5.6. *The pathwidth of $G_{\ell,\ell-1}$ is at most $2\ell - 1$.*

Proof. We give a path decomposition of width at most $2\ell - 1$ of $G_{\ell,\ell-1}$. We first create a bag B_0 which contains $C_0 \cup \{u_0\}$. For every $1 \leq i \leq \ell - 1$, we create a bag $B_i = C_0 \cup C_i$ such that $B_1 B_2 \dots B_{\ell-1}$ induces a path. One can easily check that it is a path decomposition of width $2\ell - 1$ of $G_{\ell,\ell-1}$. $\diamondsuit$

However, it is not clear if and how we can obtain a better upper bound for bounded pathwidth graphs. To summarize, $\mathcal{R}_k(G)$ is not necessarily connected if $k < \Gamma(G) + \frac{pw(G)}{2} + \mathcal{O}(1)$ and is connected if $k \geq \Gamma(G) + pw(G) + 1$. We were not able to close this gap and leave it as an open problem.

5 Conclusion

In this chapter, we studied the reconfiguration of dominating sets under the Token Addition-Removal rule. More precisely, we investigated the values of the thresholds k such that the reconfiguration graph $\mathcal{R}_k(G)$ is connected. We presented several upper bounds on the minimum threshold d_0 such that above this value, all the reconfiguration graphs $\mathcal{R}_k(G)$ are connected. Previously existing upper bounds included $\Gamma(G) + \gamma(G)$ [HS14], $n - \mu(G) + 1$ [SMN14], and $\Gamma(G) + \alpha(G) - 1$ [HS17]. We presented a joint work with Nicolas Bousquet and Paul Ouvrard, in which we improve this last result, by proving that the reconfiguration graph moreover has a linear diameter when $k \geq \Gamma(G) + \alpha(G) - 1$. With the same authors, we also proved that if G is K_ℓ-minor free, then there exists a constant C such that $\mathcal{R}_k(G)$ is connected and has linear diameter for every $k \geq \Gamma(G) + C\ell\sqrt{\log_2 \ell}$. For planar graphs, we showed that $\mathcal{R}_k(G)$ is connected and has linear diameter for any $k \geq \Gamma(G) + 3$. We underlined that this result is almost tight since Suzuki et al. proved that there exist some planar graphs for which if $k = \Gamma(G) + 1$ then $\mathcal{R}_k(G)$ is disconnected [SMN14]. Thus, only the connectivity of $\mathcal{R}_{\Gamma(G)+2}(G)$ remains unknown and we leave it as an open question, although we strongly believe that the reconfiguration graph is also connected in this case. We finally presented another upper bound depending on the treewidth of G, by proving that if $k \geq \Gamma(G) + tw(G) + 1$, then $\mathcal{R}_k(G)$ is connected, with a linear diameter. We underlined that this result is almost tight, since $\mathcal{R}_{\Gamma(G)+tw(G)-2}$ is not necessarily connected. The gap is left as an open question, as well as finding a better upper bound depending on the pathwidth.

More generally, studying the dependency between d_0 and other graph parameters can lead to different upper bounds, that will lead to a better understanding of the connectivity of the reconfiguration graph. There are also a lot of open questions on other problems related to the reconfiguration of dominating sets under token addition-removal, such as the reachability problem in many graph classes (other than the ones studied by Haddadan et al. [HIM+16]). And there remains a lot of work to do on other adjacency rules for the reconfiguration of dominating sets. In particular, we present some results on the reconfiguration of dominating sets under the token sliding rule in the next chapter.

Chapter 6

Reconfiguration of Dominating Sets under the Token Sliding rule

1 Introduction

In this chapter, we switch the adjacency rule, as we study the reconfiguration of dominating sets under Token Sliding (TS). The instance is still a graph G, and the feasible solutions are the dominating sets of G of a given size k. Recall that under TS, two dominating sets D and D' are adjacent in the reconfiguration graph if there exists an edge $uv \in E$ such that $D' = (D \cap \{v\}) \setminus \{u\}$. An example of reconfiguration sequence is given in Figure 6.1.

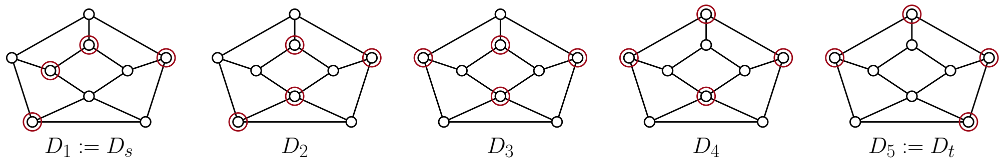


Figure 6.1: An example of reconfiguration sequence of dominating sets under the TS rule.

The reconfiguration of dominating sets under the token sliding rule has been first studied in 2011, in the particular case where $k = \gamma(G)$, i.e. the dominating sets are minimum. Fricke, Hedetniemi, Hedetniemi, and Hutson [FHHH11] gave the reconfiguration graph when G is a complete graph, a complete bipartite graph, a path or a cycle. Connelly, Hedetniemi and Hutson [CHHH11] then showed that very graph is realisable as the reconfiguration graph of infinitely many graphs G. They also proved that if $n \leq 5$, then the reconfiguration graph is connected. More recently, Edwards et al. [EMN18] investigated the particular case where G is a tree. They were then able to determine the order, the diameter, and the maximum degree of the reconfiguration graph. In particular, the diameter is linear. More details about the state of the art on the reconfiguration of minimum dominating sets are given in [MN20].

Bonamy, Dorbec and Ouvrard were the first to investigate the general case, where the dominating sets are not necessarily minimum, in 2019. They studied the complexity of the reachability problem in several graph classes. They proved that it is PSPACE-complete, even when restricted to split, bipartite graphs, and bounded treewidth graphs [BDO19]. On the other hand, they provide polynomial time algorithms for cographs and dually chordal graphs. In particular, since dually chordal graphs contain circular interval graphs, the problem is in P for circular interval graphs, and they actually prove the following.

Theorem 6.1. *[BDO19] Let G be a connected interval graph, and D_s, D_t be two dominating sets of G of the same size. There always exist a TS-reconfiguration sequence from D_s to D_t.*

They also raised two open questions, on the complexity of the reachability problem in circle graphs and circular arc graphs. A more general open question asks if there exist some graph classes for which the source problem is NP-complete, but the reachability problem is P, or for which the source problem is in P while the reachability problem is PSPACE-complete. Circle graphs and circular arc graphs are possible leads to answer this question.

In their paper, Bonamy et al. authorize the dominating sets to be multisets. In other words, they authorize several tokens on a same vertex. This choice can have an impact on the output of the reachability problem. They illustrate it with the example of the reconfiguration of dominating sets of size 2 of the star S_n of order $n \geq 3$. Indeed, any dominating set of size 2 contains the center and a leaf, and it is not possible to go from one dominating set to another by sliding tokens, if we do not authorize the token on the leaf to first slide to the center (which is already occupied), then to the other leaf.

We follow the same choice in this chapter. Thus, a *dominating set* D of G is defined here as a multiset of elements of V, such that for any $v \in V$, $v \in D$ or there exists $u \in D$ such that $uv \in E$. Similarly, an *edge-dominating set* D' of G is defined as a multiset of elements of E, such that for any $uv \in E$, $uv \in D'$ or there exists uw or $vw \in D'$. A *vertex cover* C of G is a multiset of elements of V such that for any $uv \in E$, $u \in C$ or $v \in C$. An *independent set* I of G is a multiset of elements of V such that for any $uv \in E$, $u \notin I$ or $v \notin I$. By abuse of language, all along this chapter we may refer to multisets as sets.

In this chapter, we present a joint work with Nicolas Bousquet where we continue the investigation on the complexity of the reachability problem in several graph classes. In other words, we study the complexity of the following problem.

DOMINATING SET RECONFIGURATION UNDER TOKEN SLIDING (DSR$_{\text{TS}}$)
Input: A graph G, two dominating sets D_s and D_t of G
Output: Does there exist a dominating set reconfiguration sequence from D_s to D_t with the token sliding rule ?

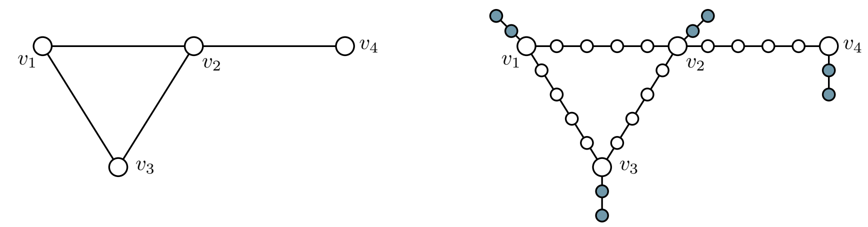
2 Planar Bipartite Graphs and Unit Disk Graphs

We first investigate the complexity of DSR$_{\text{TS}}$ in planar bipartite graphs and unit disk graphs. We prove that in both these classes, DSR$_{\text{TS}}$ is PSPACE-complete. We use the same polynomial-time reduction from minVCR$_{\text{TS}}$ in planar graphs of degree at most 3 to DSR$_{\text{TS}}$ in both proofs. Recall that the minVCR$_{\text{TS}}$ problem is defined as follows.

MINIMUM VERTEX COVER RECONFIGURATION UNDER TOKEN SLIDING (minVCR$_{\text{TS}}$)
Input: A graph G, two minimum vertex covers C_s and C_t of G
Output: Does there exist a vertex cover reconfiguration sequence from C_s to C_t with the token sliding rule ?

The reduction uses a combination of the ideas used in [BDO19] for bipartite graphs and in [Ale82] for independent sets.

Let $G = (V, E)$ be a planar graph of maximum degree at most 3 and $V = \{v_1, \dots, v_n\}$. Let $k = 3q + 1$ for some integer q. We create the graph G_k as follows. We subdivide every edge of G into paths containing $k + 1$ edges (i.e. We create k new vertices called the *subdivided vertices*). The vertices already there before the subdivision of the edges are called the *original vertices*. For every original vertex v, we finally add a pendant path only attached to v containing two new vertices. The vertices of these pendant paths are called *pendant vertices*. The construction is illustrated in Figure 6.2.



G $\qquad$ G_k

Figure 6.2: A planar graph G of maximum degree 3 and its associated graph G_k with $k = 4$. The small white vertices are the subdivided vertices, the small blue ones are the pendant vertices, and the big vertices are the original vertices.

More formally, we have $V(G_k) = V \cup V' \cup V'' \bigcup_{v_iv_j \in E} V_{i,j}$ and $E(G_k) = \bigcup_{v_iv_j \in E} E_{i,j} \bigcup_{v_i \in V} E_i$, where:

- $V' = \{v'_1, \dots v'_n\}$
- $V'' = \{v''_1, \dots v''_n\}$
- for every $v_iv_j \in E$, $V_{i,j} = \bigcup_{1 \leq p \leq k} \{v^p_{i,j}\}$
- for every $v_iv_j \in E$, $E_{i,j} = \{v_iv^1_{i,j}, v^k_{i,j}v_j\} \bigcup_{1 \leq p \leq k-1} \{v^p_{i,j}v^{p+1}_{i,j}\}$
- for every $v_i \in V$, $E_i = \{v_iv'_i, v'_iv''_i\}$.

By abuse of notation, $V_{i,j}$ and $V_{j,i}$ will denote the same sets and the vertex $v^p_{i,j}$ can equivalently be denoted by $v^{k-p+1}_{j,i}$.

The following remark is straightforward.

Remark 6.1. *If G is planar then the graph G_k is planar. Moreover, if k is odd then the graph G_k is bipartite.*

For any vertex cover C of G, let us create a dominating set $D(C)$ of G_k called the *dominating set associated to C*. We let $D(C) := C \bigcup_{v_iv_j \in E} D_{i,j} \bigcup_{v_i \in V} \{v'_i\}$ where for any i and j such that $i < j$, $D_{i,j} = \{v^p_{i,j} : p = 0 \mod 3\}$ if $v_i \in C$, and $D_{i,j} = \{v^p_{i,j} : p = 2 \mod 3\}$ otherwise.

Lemma 6.1. *Let $G = (V, E)$ be a planar graph of maximum degree at most 3, C be a minimum vertex cover of G and $k = 3q + 1$ for some integer q. Then $D(C)$ is a dominating set of G_k.*

Proof. For every $v_i \in V$, v'_i dominates v_i, v'_i and v''_i. Thus, the original vertices and the pending vertices are all dominated in $D(C)$. Now, for any $v_iv_j \in E$ such that $i < j$ and $v_i \in C$, v_i dominates $v^1_{i,j}$, and $v^p_{i,j}$ dominates $v^{p-1}_{i,j}$, $v^p_{i,j}$ and $v^{p+1}_{i,j}$ for any $p = 0 \mod 3$. On the other hand, for any $v_iv_j \in E$ such that $i < j$ and $v_i \notin C$, we have $v_j \in C$ and thus v_j dominates $v^k_{i,j}$, and $v^p_{i,j}$ dominates $v^{p-1}_{i,j}$, $v^p_{i,j}$ and $v^{p+1}_{i,j}$ for any $p = 2 \mod 3$. Thus, the subdivided vertices are all dominated. □

We can now show that the reduction is safe.

Lemma 6.2. *Let G be a planar graph of degree at most 3, C_s and C_t be two minimum vertex covers of G. Let $D_s := D(C_s)$ and $D_t := D(C_t)$. If $k = 3q + 1$ for some integer q, then (G_k, D_s, D_t) is a yes-instance of* DSR$_{\text{TS}}$ *if and only if (G, C_s, C_t) is a yes-instance of min*VCR$_{\text{TS}}$.

Proof. ($\Leftarrow$) Let (G, C_s, C_t) be a yes-instance of VCR$_{\text{TS}}$, and let $S =< C_1 := C_s, \dots, C_\ell := C_t >$ be the associated reconfiguration sequence. We construct a reconfiguration sequence $S' =< D_1 := D_s, \dots, D_{\ell'} := D_t >$ in G_k by replacing every move $v_i \rightsquigarrow v_j$ from C_r to C_{r+1} of S by the following sequence of moves from $D(C_r)$ to $D(C_{r+1})$. All along the sequence, for every i, the token initially on v'_i will never move. Thus the pendant vertices as well as the original vertices are dominated all along the transformation. So we just have to prove that subdivided vertices are dominated.

- We apply the sequence of moves $(v_{x,i}^2 \rightsquigarrow v_{x,i}^3, v_{x,i}^5 \rightsquigarrow v_{x,i}^6, \ldots, v_{x,i}^{k-2} \rightsquigarrow v_{x,i}^{k-1})$ for every $v_x \in N(v_i)$ such that $i < x$ and $v_x \neq v_j$. Since all the vertex covers of S are minimum (since C_s is), $v_i \notin C_{r+1}$. So all the neighbors v_x of v_i distinct from v_j are in C_r and thus in $D(C_r)$. Therefore, $v_{x,i}^1$ is dominated by v_x in $D(C_r)$ and we can apply the first move. Similarly, by induction, for any $p = 2 \mod 3$, $v_{x,i}^{p-1}$ is dominated by $v_{x,i}^{p-2}$ and we can apply the move $v_{x,i}^p \rightsquigarrow v_{x,i}^{p+1}$. So all the intermediate sets are dominating sets of G_k. Let D' be the resulting dominating set.
- We then apply $(v_i \rightsquigarrow v_{i,j}^1 \rightsquigarrow v_{i,j}^2, v_{i,j}^3 \rightsquigarrow v_{i,j}^4 \rightsquigarrow v_{i,j}^5, \ldots, v_{i,j}^{k-1} \rightsquigarrow v_{i,j}^k \rightsquigarrow v_j)$. We have seen that for any neighbor $v_x \neq v_j$ of v_i in G, $v_x \in C_r$. Thus, either $x < i$ and $v_{x,i}^{k-1} \in D(C_r)$, or $i < x$ and the previous sequence ensures that $v_{x,i}^{k-1} \in D'$. So for any neighbor $v_x \neq v_j$ of v_i in G, $v_{x,i}^k$ is dominated by $v_{x,i}^{k-1}$. So $v_i \rightsquigarrow v_{i,j}^1 \rightsquigarrow v_{i,j}^2$ keeps G_k dominated. Then, by induction, for any $p = 0 \mod 3$, $v_{i,j}^p$ is dominated by $v_{i,j}^{p-1}$ and thus $v_{i,j}^p \rightsquigarrow v_{i,j}^{p+1} \rightsquigarrow v_{i,j}^{p+2}$, or $v_{i,j}^p \rightsquigarrow v_{i,j}^{p+1} \rightsquigarrow v_j$ if $p = k-1$ keeps a dominating set. Let D'' be the resulting dominating set.
- We finally apply the sequence $(v_{x,j}^{k-1} \rightsquigarrow v_{x,j}^{k-2}, \ldots, v_{x,j}^6 \rightsquigarrow v_{x,j}^5, v_{x,j}^3 \rightsquigarrow v_{x,j}^2)$ for every $v_x \in N(v_j)$ such that $j < x$ and $v_x \neq v_i$. Since $v_j \in D''$, $v_{x,j}^k$ is dominated by v_j in D'' and we can apply the first move. And by induction, for any $p = 0 \mod 3$, $v_{x,j}^{p+1}$ is dominated by $v_{x,j}^{p+2}$ and we can apply the move $v_{x,i}^p \rightsquigarrow v_{x,i}^{p-1}$. Thus all the intermediate sets are dominating sets of G_k.

After these moves, the obtained dominating set is $D(C_{r+1})$, which concludes this direction of the proof.

($\Rightarrow$) Let (G_k, D_s, D_t) be a yes-instance of DSR$_{\text{TS}}$, and let us prove that (G, C_s, C_t) is a yes-instance of minVCR$_{\text{TS}}$. There exists a reconfiguration sequence $S' =< D_1 := D_s, \ldots, D_\ell := D_t >$ in G_k. Let D be a dominating set of S'. We define the following parameters of D. For any $v_i \in V$, let $\alpha(v_i)$ be the multiplicity of v_i in D, $\beta(v_i)$ be the sum of the multiplicities of v_i' and v_i'' minus 1, and for any $v_j \in V$ such that $v_iv_j \in E$, let $\gamma_j(v_i)$ be 0 if $i > j$, and the sum of the multiplicities of the subdivided vertices $v_{i,j}^p$ in D minus q if $i < j$. All these values are non negative. Indeed, for every $v_i \in V$, we have $\alpha(v_i) \geq 0$, and since v_i'' is dominated in D, $\beta(v_i) \geq 0$. Moreover, every vertex $v_{i,j}^p$ with $p = 2 \mod 3$ has to be dominated by a different vertex in D. So at least q subdivided vertices $v_{i,j}^r$ are in D, and $\gamma_j(v_i) \geq 0$.

We define, for every i, $\phi(v_i) = \alpha(v_i) + \beta(v_i) + \sum_{v_iv_j \in E} \gamma_j(v_i)$. Note that $\phi(v_i) \geq 0$ and that any token move does not modify $\sum_{v_i \in V} \phi(v_i)$. Let us define the set $C(D)$ of vertices of G *associated to* D, where $v_i \in C(D)$ with multiplicity $\phi(v_i)$ for every i. We construct the sequence S of subsets of V by replacing any dominating set D of S' by $C(D)$. To conclude, we simply have to prove that: (i) the sets associated to D_s and D_t are C_s and C_t and; (ii) that for every dominating set D, $C(D)$ is a vertex cover; and (iii) that a token slide in S' corresponds to a token slide in S.

Proof of (i). Firstly, $D_s = D(C_s)$. So $\alpha(v_i) = 1$ for any $v_i \in C$ and $\alpha(v_i) = 0$ otherwise. Moreover, for any v_i, $\beta(v_i) = 0$ and for any $v_iv_j \in E$, $\gamma_j(v_i) = 0$. So if $v_i \in C_s$, $\phi(v_i) = 1$, otherwise $\phi(v_i) = 0$, which gives $C(D_s) = C_s$. Similarly, $C(D_t) = C_t$.

Proof of (ii). For any $v_iv_j \in E$, if $v_i \in D$ then $\alpha(v_i) \geq 1$. So $\phi(v_i) \geq 1$ and $v_i \in C(D)$. If $v_i \notin D$, to dominate $v_{i,j}^1$ either $v_{i,j}^1 \in D$ or $v_{i,j}^2 \in D$. Moreover, for any p such that $p = 1 \mod 3$ and $4 \leq p \leq k-3$, to dominate $v_{i,j}^p$ in D we have $v_{i,j}^{p-1} \in D$, $v_{i,j}^p \in D$ or $v_{i,j}^{p+1} \in D$, and to dominate $v_{i,j}^k$ we have $v_{i,j}^{k-1} \in D$, $v_{i,j}^k \in D$ or $v_j \in D$. So either $v_j \in D$, in which case $v_j \in C(D)$, or there are at least $q+1$ subdivided vertices $v_{i,j}^r$ in D, in which case either $i < j$ and $\gamma_j(v_i) \geq 1$, or $i > j$ and $\gamma_i(v_j) \geq 1$. This gives either $\phi(v_i) \geq 1$ so $v_i \in C(D)$, or $\phi(v_j) \geq 1$ so $v_j \in C(D)$. Therefore, for any edge $v_iv_j \in E$, $v_i \in C(D)$ or $v_j \in C(D)$. So $C(D)$ is a vertex cover of G.

Proof of (iii). Let D_r and D_{r+1} be two adjacent sets of S'. There exist two adjacent vertices u and v in G_k such that $D_{r+1} = D_r \cup v \setminus u$. The pairs of vertices that are adjacent in G_k are two pendant vertices, or a pendant vertex and an original vertex, or two subdivided vertices, or a subdivided vertex and an original vertex. If both u and v are pendant vertices, then they are pendant on the same original vertex. In that case, $\phi(x)$ is left unchanged for every x and $C(D_{r+1}) = C(D_r)$. Similarly, if u and v are subdivided vertices, they belong to the same original edge and $\phi(v)$ is left unchanged for every v so $C(D_{r+1}) = C(D_r)$. If one vertex is pendant and the other is original then, free to permute D_r and D_{r+1}, we can assume that u is a pendant vertex and v is an original vertex. So there exists i such that $v = v_i$ and $u = v_i'$. Thus $\alpha(v_i)$ increases by 1 from D_r to D_{r+1} but $\beta(v_i)$ decreases by 1, which gives $C(D_{r+1}) = C(D_r)$. Finally, if one is a subdivided vertex and the other an original vertex, we can assume u is subdivided and v original. So there exist i and j such that $v = v_i$ and $u = v_{i,j}^1$. So $\alpha(v_i)$ increases by 1 from D_r to D_{r+1}, if $i < j$ then $\gamma_j(v_i)$ decreases by 1, and if $i > j$ then $\gamma_i(v_j)$ decreases by 1. Thus, if $i < j$ then $C(D_{r+1}) = C(D_r)$, and if $i > j$ then $C(D_{r+1}) = C(D_r) \cup v_i \setminus v_j$, which corresponds to the move $v_j \rightsquigarrow v_i$. □

Let us now explain how to use Lemma 6.2 to show that DSR$_{\text{TS}}$ is PSPACE-complete in planar bipartite graphs and in unit disk graphs.

2.1 Planar Bipartite Graphs

The first result we show concerns planar bipartite graphs. We prove the following.

Theorem 6.2. *DSR$_{\text{TS}}$ is PSPACE-complete in planar bipartite graphs.*

Proof. Let G be a planar graph of maximum degree at most 3. Since G is planar, Remark 6.1 ensures that the graph G_1 of the reduction of Lemma 6.2 is a bipartite planar graph. So if minVCR$_{\text{TS}}$ is PSPACE-complete in planar graphs of maximum degree 3 the same holds for DSR$_{\text{TS}}$ in planar bipartite graphs by Lemma 6.2.

The TAR-MAXIMUM INDEPENDENT SET RECONFIGURATION problem is PSPACE-complete in planar graphs of maximum degree 3 [IDH+11]. For INDEPENDENT SET RECONFIGURATION, TAR and TJ models are equivalent [KMM12]. Moreover, for maximum independent sets, TS and TJ models are equivalent (since a token can only be moved on one of its neighbors because of the maximality). Thus MAXIMUM INDEPENDENT SET RECONFIGURATION is PSPACE-complete in planar graphs of maximum degree 3 with the token sliding rule. Since the complement of a maximum independent set of a graph H is a minimum vertex cover, MINIMUM VERTEX COVER RECONFIGURATION is PSPACE-complete in planar graphs of maximum degree 3 with the token sliding rule. Thus, by Lemma 6.2, DSR$_{\text{TS}}$ is PSPACE-complete in planar bipartite graphs. □

2.2 Unit Disk Graphs

We now show that DSR$_{\text{TS}}$ is PSPACE-complete in unit disk graphs. To do so, we will use the reduction presented in Section 2. But first, we need to prove that there exists a polynomial k for which G_k is a unit disk graph. We use the following lemma, which refines of a result of Valiant [Val81].

Lemma 6.3. *Any planar graph G of maximum degree at most 4 can be embedded in the plane in such a way that the vertices are at integer coordinates, and the edges are non-crossing paths composed of unions of vertical and horizontal segments with integer coordinates and all have the same polynomial length λ. Such an embedding can be obtained in polynomial time. We can moreover ensure that $\lambda = 2 \mod 3$.*

Proof. Valiant [Val81] proved that any planar graph G can be embedded on a 2-dimensional grid with integral coordinates, in such a way that every edge is associated a path in the grid, such that the paths are non-crossing, i.e. if two paths share a common point, it is an extremity of both paths. This embedding can be obtained in polynomial time [IPS82], and it has an area in $\mathcal{O}(n)$. We strengthen this result by showing that we can modify this embedding in order to ensure that the paths all have the same (polynomial in n) length λ.

Let ℓ_{max} be the maximum length of an edge in the embedding given by [Val81]. Note that ℓ_{max} is at most the area, which is in $\mathcal{O}(n)$. Let $L := \ell_{max}$ or, if we want to ensure that $\lambda = 2 \mod 3$, $L := 3\ell_{max} + 1$. Let $u := \frac{1}{8L}$. For every edge $e \in E$ of length $\ell < L$, we change the embedding of e by replacing exactly one line segment of length 1 of the path associated to e by the curve $\mathcal{Z}$ represented in Figure 6.3.

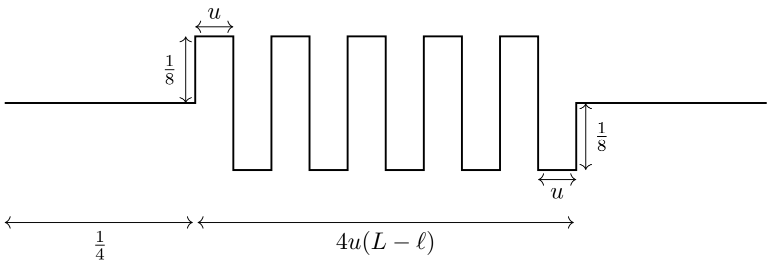


Figure 6.3: The curve $\mathcal{Z}$.

The sum of the lengths of the horizontal segments in $\mathcal{Z}$ is 1, and there are $8(L-\ell)$ vertical segments of length $\frac{1}{8}$, thus the length of the curve $\mathcal{Z}$ is $1 + L - \ell$. So by changing exactly one segments of length 1 in the path associated to e and leaving the $\ell - 1$ other ones unchanged, the length of the path associated to e is now L. So every edge now has the same length L.

Since $u = \frac{1}{8L}$ and $L - \ell < L$, we have $4u(L-\ell) < \frac{1}{2}$. So in $\mathcal{Z}$, the vertical lines are at distance at least $\frac{1}{4}$ from each extremity of the original straight line. Since the height of the rectangles is $\frac{1}{8}$, the curves $\mathcal{Z}$ of different line segments therefore only intersect where the original line segments intersect. So the new paths representing different edges remain non-crossing.

Let us now rescale the embedding. Note that all the coordinate of the points defined in the reduction are of the form $\frac{a}{8L}$. So if we rescale by $8L$, all the vertices will now have integral coordinates. So this rescaling ensures that the vertical and horizontal lines of each path are edges of an integral grid. Finally, the path associated to each edge has length $\lambda = \frac{L}{u} = 8L^2$, which is polynomial. And if we took $L := 3\ell_{max} + 1$, we have $L = 1 \mod 3$, so $\lambda = 2 \mod 3$. This concludes the proof. $\square$

Lemma 6.4. *For any planar graph G of maximum degree at most 4, there exists an integer k with $k = 1 \mod 3$ such that the k-subdivision of G is a unit disk graph. Moreover, the value of k is polynomial.*

Proof. By Lemma 6.3, we can compute in polynomial time an embedding of G in the plane such that the vertices of G are at integer coordinates, and such that the edges are unions of vertical and horizontal segments with integer coordinates, and all have the same polynomial length λ, with $\lambda = 2 \mod 3$. Let us draw disks of radius $\frac{1}{8}$ on this embedding, such that there is a center of a disk on every integer coordinates that belongs to an edge, as well as on every $\frac{1}{4}$ of integer coordinates that belongs to an edge (thus placing 3 centers between two consecutive integer coordinates on an edge). Since the angle between two line segments in the drawing is more than $\frac{\pi}{3}$, two non consecutive disks on an edge cannot intersect, while consecutive ones indeed intersect. Thus, the unit disk graph obtained with this set of disks is a k-subdivision of G, where $k = 4\lambda - 1$ and thus $k = 1 \mod 3$. $\square$

Theorem 6.3. DSR$_{\text{TS}}$ *is* PSPACE*-complete in unit disk graphs.*

Proof. Let G be a planar graph of maximum degree at most 3, and let us add a pendant vertex to any vertex of G. The resulting graph G' is planar and has degree at most 4. By Lemma 6.4, there exists a polynomial function k with $k = 1 \mod 3$ such that the k-subdivision G'' of G' is a unit disk graph. One can remark that the graph G_k of the construction of Lemma 6.2 is a subgraph of G'' (the paths pending on G'' are too long, we simply have to cut them into paths with two edges). So G_k is a unit disk graph. Since minVCR$_{\text{TS}}$ is PSPACE-complete in planar graphs of degree at most 3 [IDH+11], Lemma 6.2 ensures that DSR$_{\text{TS}}$ is PSPACE-complete in unit disk graphs. $\square$

3 Circle Graphs

Recall that a way to reprsent a circle graph consists in defining one real interval for each vertex and there is an edge between two intervals if their relative intervals intersect but do not overlap. In this section, we use this representation of circle graphs. Moreover, for every interval I, $\ell(I)$ denotes the *left extremity* of I, and $r(I)$ the *right extremity* of I.

We show that DSR$_{\text{TS}}$ is PSPACE-complete in circle graphs. We provide a polynomial time reduction from SATR to DSR$_{\text{TS}}$. This reduction is very similar to the one used in [Kei93] to show that the minimum dominating set problem is NP-complete on circle graphs. Recall that the SATR problem is defined as follows:

SATISFIABILITY RECONFIGURATION (SATR)
Input: A Boolean formula F in conjunctive normal form, two variable assignments A_s and A_t that satisfy F.
Output: Does there exist a reconfiguration sequence from A_s to A_t that maintains F satisfied, where the operation consists in a variable flip, i.e. the change of the assignment of exactly one variable from $x = 0$ to $x = 1$, or conversely ?

Let (F, A_s, A_t) be an instance of the SATR problem. Let $x_1 \dots, x_n$ be the variables of the Boolean formula F. Since F is in conjunctive normal form, it is by definition a conjunction of clauses $c_1, \dots, c_m$ which are disjunctions of literals. Recall that a literal is a variable or the negation of a variable, and we denote by $x_i \in c_j$ the fact that x_i is a literal of c_j, and $\overline{x_i} \in c_j$ the fact that the negation of x_i is a literal of c_j. Since duplicating clauses does not modify the satisfiability of a formula, we can assume without loss of generality that m is a multiple of 4. We can also assume that for every $i \leq n$ and $j \leq m$, x_i or $\overline{x_i}$ is not in c_j since otherwise the clause is indeed satisfied.

The reduction. Let us construct an instance $(G_F, D_F(A_s), D_F(A_t))$ of the DSR$_{\text{TS}}$ problem from (F, A_s, A_t). We start by constructing the circle graph G_F from F. All along this construction, we repeatedly refer to real number as *points*. We say that a point p is *at the left* of a point q (or q is *at the right* of p) if $p < q$. We say that p is *just at the left* of q, (or q is *just at the right* of p) if p is at the left of q, and no interval defined so far has an extremity in $[p, q]$.

Finally, we say that an interval I *frames* a set of points P if $\ell(I)$ is just at the left of the minimum of P and $r(I)$ is just at the right of the maximum of P.

One can easily check that by adding an interval that frames one extremity of the interval of a vertex u of a graph H, we add one vertex to H which is only connected to u. So:

Remark 6.2. *If H is a circle graph and u is a vertex of H, then the graph H plus a new vertex only connected to u is circle graph.*

We construct G_F step by step. Each step consists in creating new intervals, giving their positions regarding to the previously constructed intervals. We also outline some of the edges and non edges in G_F that have an impact on the upcoming proofs. Figures 6.4, 6.5 and 6.6 illustrate the positions of most of the intervals of G_F.

For each variable x_i, we create m *base intervals* B^i_j where $1 \leq j \leq m$. The base intervals B^i_j are pairwise disjoint for any i and j, and are ordered by increasing i, then increasing j for a same i.

For each variable x_i, we then create $\frac{m}{2}$ intervals X^i_j called the *positive bridge intervals* of x_i, and $\frac{m}{2}$ intervals $\overline{X}^i_j$ called the *negative bridge intervals* of x_i, where $1 \leq j \leq \frac{m}{2}$. A *bridge interval* is a positive or a negative bridge interval. Let us give the positions of these intervals. They are illustrated in Figure 6.4.

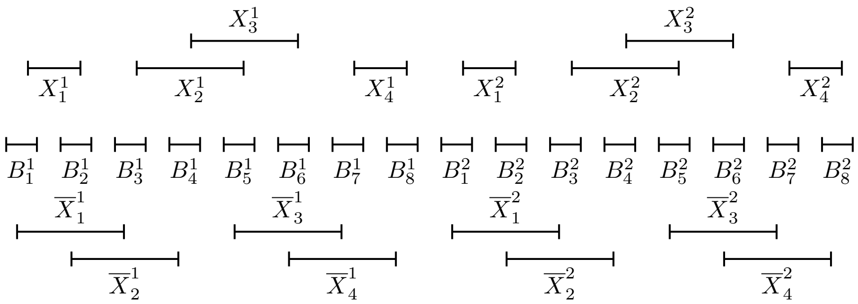


Figure 6.4: The base, positive and negative bridge intervals obtained with $n = 2$ and $m = 8$.

Let q be such that $m = 4q$. For every i and every $0 \leq r < q$, the interval $\overline{X}_{2r+1}^i$ starts just at the right of $\ell(B_{4r+1}^i)$ and ends just at the right of $\ell(B_{4r+3}^i)$, and $\overline{X}_{2r+2}^i$ starts just at the right of $\ell(B_{4r+2}^i)$ and ends just at the right of $\ell(B_{4r+4}^i)$. The interval X_1^i starts just at the left of $r(B_1^i)$ and ends just at the left of $r(B_2^i)$. For every $1 \leq r < q$, the interval X_{2r}^i starts just at the left of $r(B_{4r-1}^i)$ and ends just at the left of $r(B_{4r+1}^i)$, and X_{2r+1}^i starts just at the left of $r(B_{4r}^i)$ and ends just at the left of $r(B_{4r+2}^i)$. Finally, $X_{\frac{m}{2}}^i$ starts just at the left of $r(B_{m-1}^i)$ and ends just at the left of $r(B_m^i)$.

Let us outline some of the edges induced by these intervals. Base intervals are pairwise non adjacent. Moreover, every positive (resp. negative) bridge interval is incident to exactly two base intervals. And all the positive (resp. negative) bridge intervals of x_i are incident to pairwise distinct base intervals. In particular, the positive (resp. negative) bridge intervals dominate the base intervals. Thus, every base interval is adjacent to exactly one positive and one negative bridge interval. All the positive (resp. negative) bridge intervals but X_1^i and $X_{\frac{m}{2}}^i$ have exactly one other positive (resp. negative) bridge interval neighbor. Finally, for every i, every negative bridge interval $\overline{X}_j^i$ has exactly two positive bridge interval neighbors which are X_{j-1}^i and X_j^i except for $\overline{X}_1^i$ which does not have any for any i. Note that a bridge interval of x_i is not adjacent to a bridge interval or a base interval of x_j for $j \neq i$.

Now for any clause c_j, we create two identical *clause intervals* C_j and C'_j. The clause intervals C_j are pairwise disjoint and ordered by increasing j, and we have $\ell(C_1) > r(B_m^n)$. Thus, they are not adjacent to any intervals constructed so far.

For any j such that x_i is in the clause c_j, we also create four intervals T_j^i, U_j^i, V_j^i and W_j^i, called the *positive path intervals* of x_i, and for any j such that $\overline{x_i}$ is in the clause c_j, we create four intervals $\overline{T}_j^i$, $\overline{U}_j^i$, $\overline{V}_j^i$ and $\overline{W}_j^i$, called the *negative path intervals* of x_i. These intervals are represented in Figure 6.5. To better see the relative position of the extremities, a zoom is given in Figure 6.6. The interval T_j^i frames the right extremity of B_j^i and the extremity of the positive bridge interval that belongs to B_j^i. The interval $\overline{T}_j^i$ frames the left extremity of B_j^i and the extremity of the negative bridge interval that belongs to B_j^i. The interval U_j^i starts just at the left of $r(T_j^i)$, the interval $\overline{U}_j^i$ starts just at the right of $l(\overline{T}_j^i)$, and they both end between the right of the last base interval of the variable x_i and the left of the next base or clause interval. We moreover construct the intervals U_j^i (resp. $\overline{U}_j^i$) in such a way $r(U_j^i)$ (resp. $r(\overline{U}_j^i)$) is increasing when j is increasing. In other words, the U_j^i (resp. $\overline{U}_j^i$) are pairwise adjacent. The interval V_j^i (resp. $\overline{V_j^i}$) frames the right extremity of U_j^i (resp. $\overline{U_j^i}$). And the interval W_j^i (resp. $\overline{W_j^i}$) starts just at the left of $r(V_j^i)$ (resp. $r(\overline{V_j^i})$) and ends in an arbitrary point of C_j. Moreover, for any $i \neq i'$, W_j^i (resp. $\overline{W_j^i}$) and $W_j^{i'}$ (resp. $\overline{W_j^{i'}}$) end on the same point of C_j. This ensures that they overlap and are therefore not adjacent.

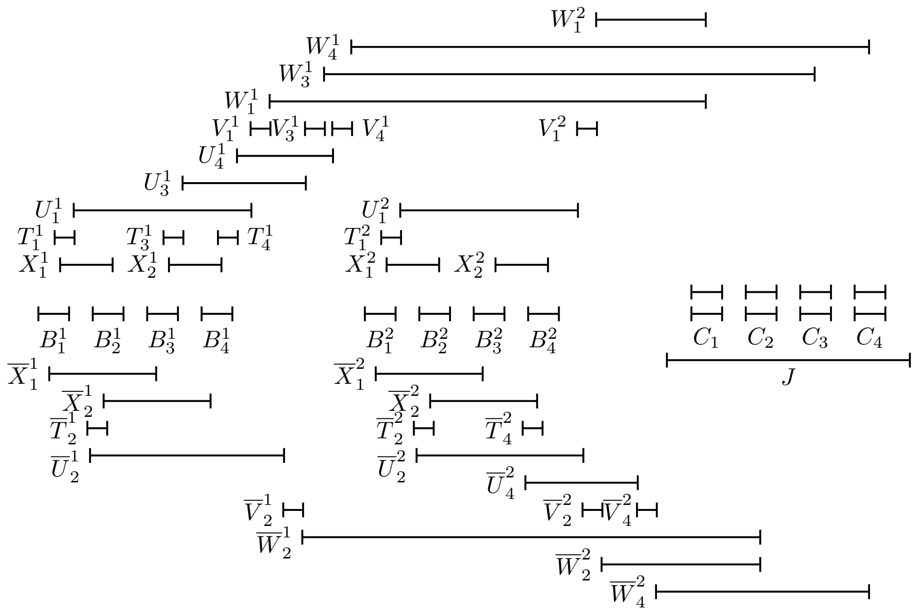


Figure 6.5: The intervals obtained for the formula $F = (x_1 \vee x_2) \wedge (\overline{x_1} \vee \overline{x_2}) \wedge (x_1) \wedge (\overline{x_2} \vee x_1)$ with $m = 4$ clauses and $n = 2$ variables. The dead-end intervals and the pending intervals are not represented here.

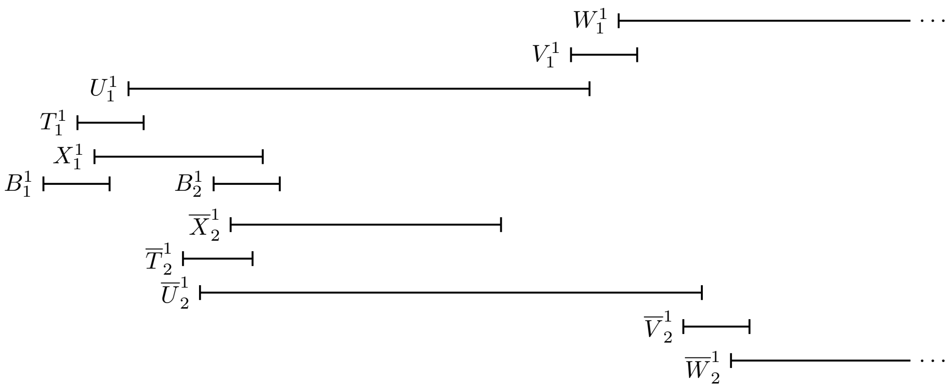


Figure 6.6: A zoom on some intervals of the variable x_1.

A *path interval* is a positive or a negative path interval. The *intervals of* x_i are the base, bridge and path intervals of x_i. The T intervals of x_i refers to the intervals T_j^i for any j. The $\overline{T}$, U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$ intervals of x_i are defined similarly.

Let us outline some neighbors of the path intervals. The neighborhood of every clause interval C_j is the set of intervals W_j^i with $x_i \in c_j$ and intervals $\overline{W}_j^i$ with $\overline{x_i} \in c_j$. Since V_j^i spans the left extremity of W_j^i and the right extremity of U_j^i and since no interval starts or ends between these two points, the interval V_j^i is only adjacent to U_j^i and W_j^i. Similarly $\overline{V}_j^i$ is only adjacent to $\overline{U}_j^i$ and $\overline{W}_j^i$. Moreover, T_j^i is only adjacent to B_j^i, U_j^i and one positive bridge interval (the same one that is adjacent to B_j^i), and $\overline{T}_j^i$ is only adjacent to B_j^i, $\overline{U}_j^i$ and one negative bridge interval (the same one that is adjacent to B_j^i). Moreover, since U_j^i and W_j^i are not adjacent, B_j^i, T_j^i, U_j^i, V_j^i, W_j^i and C_j induce a path, and since $\overline{U}_j^i$ and $\overline{W}_j^i$ are not adjacent, B_j^i, $\overline{T}_j^i$, $\overline{U}_j^i$, $\overline{V}_j^i$, $\overline{W}_j^i$ and C_j induce a path. Finally, for any two variables x_i and x_i' such that $x_i \neq x_i'$, the only path intervals of respectively x_i and x_i' that can be adjacent are the W and $\overline{W}$ intervals adjacent to different clause intervals.

Now, for every bridge interval and every U, $\overline{U}$, W and $\overline{W}$ interval, we create a *dead-end interval*, that is only adjacent to it. Remark 6.2 ensures that it can be done while keeping a circle graph. Then, for any dead-end interval, we create $6mn$ *pending intervals* that are each only adjacent to it. Again, Remark 6.2 ensures that the resulting graph is a circle graph. Informally speaking, since the dead-end intervals have a lot of pending intervals, they will be forced to be in any dominating set. Thus, in any dominating set, we will know that bridge, U, $\overline{U}$, W and $\overline{W}$ intervals (as well as dead-end an pending vertices) are already dominated. So the other vertices in the dominating set will only be there to dominate the other vertices of the graph, which are called the *important vertices*.

Finally, we create a *junction interval* J, that frames $\ell(C_1)$ and $r(C_m)$. By construction, it is adjacent to every W or $\overline{W}$ interval, and to no other interval. This completes the construction of the graph G_F.

Basic properties of G_F. Let us now give a couple of properties satisfied by G_F.

Lemma 6.5. *The graph G_F is connected.*

Proof. Let x_i be a variable. Let us first prove that the intervals of x_i are in the same connected component of G_F. (Recall that they are the base, bridge and path intervals of x_i). Firstly, for any j such that $x_i \in C_j$ (resp. $\overline{x_i} \in C_j$), $B^i_j T^i_j U^i_j V^i_j W^i_j$ (resp. $B^i_j \overline{T}^i_j \overline{U}^i_j \overline{V}^i_j \overline{W}^i_j$) is a path of G_F. Since every base interval of x_i is adjacent to a positive and a negative bridge interval of x_i, it is enough to show that all the bridge intervals of x_i are in the same connected component. Since for every $j \geq 2$, $\overline{X}^i_j$ is adjacent to X^i_{j-1} and X^i_j, we know that $X^i_1 \overline{X}^i_2, X^i_2 \ldots \overline{X}^i_{\frac{m}{2}} X^i_{\frac{m}{2}}$ is a path of G_F. Moreover, $\overline{X}^i_1$ is adjacent to $\overline{X}^i_2$. So all the intervals of x_i are in the same connected component of G_F..

Now, since the junction interval J is adjacent to every W and $\overline{W}$ interval (and that each variable appears in at least one clause), J is in the connected component of every path variable, so the intervals of x_i and $x_{i'}$ are in the same connected component for every $i \neq i'$. Since each clause contains at least one variable, C_j is adjacent to at least one interval W^i_j or $\overline{W}^i_j$. Finally, each dead-end interval is adjacent to a bridge interval or a U, $\overline{U}$, W or $\overline{W}$ interval, and each pendant interval is adjacent to a dead-end interval. Therefore, G_F is connected. □

For any variable assignment A of F, let $D_F(A)$ be the set of intervals of G_F defined as follows. The junction interval J belongs to $D_F(A)$ and all the dead-end intervals belong to $D_F(A)$. For any variable x_i such that $x_i = 1$ in A, the positive bridge, W and $\overline{U}$ intervals of x_i belong to $D_F(A)$. Finally, for any variable x_i such that $x_i = 0$ in A, the negative bridge, $\overline{W}$ and U intervals of x_i belong to $D_F(A)$. The multiplicity of each of these intervals in $D_F(A)$ is one. Thus, we have $|D_F(A)| = \frac{3mn}{2} + 3\sum_{i=1}^{n} \ell_i + 1$ where for any variable x_i, ℓ_i is the number of clauses that contain x_i or $\overline{x_i}$.

Lemma 6.6. *If A satisfies F, then $D_F(A) \setminus J$ is a dominating set of G_F.*

Proof. Since every dead-end interval belongs to $D_F(A) \setminus J$, every pending and dead-end interval is dominated, as well as every bridge, U, $\overline{U}$, W and $\overline{W}$ interval. Since for each variable x_i, the positive (resp. negative) bridge intervals of x_i dominate the base intervals of x_i, the base intervals are dominated. Moreover, the positive (resp. negative) bridge intervals of x_i and the U (resp. $\overline{U}$) intervals of x_i both dominate the T (resp. $\overline{T}$) intervals of x_i. Thus, the T and $\overline{T}$ intervals are all dominated. Moreover, for any variable x_i, the U and W (resp. $\overline{U}$ and $\overline{W}$) intervals of x_i both dominate the V (resp. $\overline{V}$) intervals of x_i. Thus, the V and $\overline{V}$ intervals are all dominated. Finally, since A satisfies F, each clause has at least one of its literal in A. Thus, each C_j and C'_j has at least one adjacent interval W^i_j or $\overline{W}^i_j$ in $D_F(A) \setminus J$ and are therefore dominated by it, as well as the junction interval. □

Before continuing further, let us prove a few results that are of importance in our proof. Let $K := \frac{3mn}{2} + 3\sum_{i=1}^{n} \ell_i + 1$. Since the number $6mn$ of leaves attached on each dead-end interval is strictly more than K (as $\ell_i \leq m$), the following holds.

Remark 6.3. *Any dominating set of size at most K contains all the $(mn+2\sum_{i=1}^{n}\ell_i)$ dead-end intervals.*

So in particular, if we have a dominating set of size K, it must contain the dead-end vertices. And then all the pending, dead-end, bridge, U, $\overline{U}$, W and $\overline{W}$ intervals are dominated. So we will simply have to focus on the domination of base, T, $\overline{T}$, V, $\overline{V}$ and junction intervals (i.e. the important intervals).

Lemma 6.7. *If D is a dominating set of G_F, then for any variable x_i, D contains at least ℓ_i intervals that dominate the V and $\overline{V}$ intervals of x_i, and at least $\frac{m}{2}$ intervals that dominate the base intervals of x_i. Moreover, these two sets of intervals are disjoint. And for every $i \neq j$, the set of vertices that dominates the base, V and $\overline{V}$ intervals of x_i and x_j are disjoint.*

Proof. For any variable x_i, each interval V_j^i (resp. $\overline{V}_j^i$) can only be dominated by U_j^i, V_j^i or W_j^i (resp. $\overline{U}_j^i$, $\overline{V}_j^i$ or $\overline{W}_j^i$). Indeed V_j^i spans the left extremity of W_j^i and the right extremity of U_j^i and since no interval starts or ends between these two points, the interval V_j^i is only adjacent to U_j^i and W_j^i. And similarly $\overline{V}_j^i$ is only adjacent to $\overline{U}_j^i$ and $\overline{W}_j^i$. Thus, at least ℓ_i intervals dominate the V and $\overline{V}$ intervals of x_i. Moreover, only the base, bridge, T and $\overline{T}$ intervals of x_i are adjacent to the base intervals. Since each bridge interval is adjacent to two base intervals, and each T and $\overline{T}$ interval of x_i is adjacent to one base interval of x_i, D must contain at least $\frac{m}{2}$ of such intervals to dominate the m base intervals. □

By combining Remark 6.3 and Lemma 6.7, we obtain that any dominating set D of size K contains $(mn+2\sum_{i=1}^{n}\ell_i)$ dead-end intervals, as well as $(\ell_i+\frac{m}{2})$ intervals of x_i for any variable x_i. Since $K=\frac{3mn}{2}+3\sum_{i=1}^{n}\ell_i+1$, this leaves only one remaining token in D. Thus, for any variable x_i but at most one, there are exactly $(\ell_i+\frac{m}{2})$ intervals of x_i in D. If there exists a variable x_k for which it is not the case, then there are exactly $(\ell_k+\frac{m}{2}+1)$ intervals of x_k in D, and we call this variable the *moving variable* of D, denoted by $\sigma(D)$.

For any variable x_i, we denote by X_i the set of positive bridge variables of x_i and by $\overline{X_i}$ the set of negative bridge variables of x_i. Similarly, we denote by W_i the set of W variables of x_i and by $\overline{W_i}$ the set of $\overline{W}$ variables of x_i. Let us now give some precision about the intervals of x_i that belong to D.

Lemma 6.8. *If D is a dominating set of size K, then for any variable $x_i \neq \sigma(D)$, either $X_i \subset D$ and $\overline{X_i}\cap D=\emptyset$, or $\overline{X_i}\subset D$ and $X_i\cap D=\emptyset$.*

Proof. Since $x_i \neq \sigma(D)$, there are exactly $\ell_i+\frac{m}{2}$ variables of x_i in D. Thus, by Lemma 6.7, exactly $\frac{m}{2}$ intervals of x_i in D dominate the bridge intervals of x_i. Only the bridge, T and $\overline{T}$ intervals of x_i are adjacent to the base intervals. Moreover, bridge intervals are adjacent to two base intervals and T or $\overline{T}$ intervals are adjacent to only one. Since there are m base intervals of x_i, each interval of D must dominate a pair of base intervals (or none of them). So these intervals of D should be some bridge intervals of x_i.

Note that, by cardinality, each pair of bridge intervals of D must dominate pairwise disjoint base intervals. Let us now show by induction that these bridge intervals are either all the positive bridge intervals, or all the negative bridge intervals. We study two cases: either $X_1^i\in D$, or $X_1^i\notin D$.

Assume that $X_1^i\in D$. In D, X_1^i dominates B_1^i and B_2^i. Thus, since $\overline{X}_1^i$ dominates B_1^i and $\overline{X}_2^i$ dominates B_2^i, none of $\overline{X}_1^i,\overline{X}_2^i$ are in D (since their neighborhood in the set of base intervals is not disjoint with X_1^i). But B_3^i (resp B_4^i) is only adjacent to $\overline{X}_1^i$ and X_2^i (resp. $\overline{X}_2^i$ and X_3^i). Thus both X_2^i, X_3^i are in D. Suppose now that for a given j such that j is even and $j\leq\frac{m}{2}-2$, we have $X_j^i, X_{j+1}^i\in D$. Then, since a base interval dominated by X_j^i (resp. X_{j+1}^i) also is dominated by $\overline{X}_{j+1}^i$ (resp. $\overline{X}_{j+2}^i$), the intervals $\overline{X}_{j+1}^i,\overline{X}_{j+2}^i$ are not in D. But there is a base interval adjacent only to $\overline{X}_{j+1}^i$ and X_{j+2}^i (resp. $\overline{X}_{j+2}^i$ and X_{j+3}^i if $j\neq\frac{m}{2}-2$, or $\overline{X}_{j+2}^i$ and X_{j+2}^i if $j=\frac{m}{2}-2$). Therefore, if $j+2<\frac{m}{2}$ we have $X_{j+2}^i, X_{j+3}^i\in D$, and $X_{\frac{m}{2}}^i\in D$. By induction, if $X_1^i\in D$ then each of the $\frac{m}{2}$ positive bridge intervals belong to D and thus none of the negative bridge intervals do.

Assume now that $X_1^i \notin D$. Then, to dominate B_1^i and B_2^i, we must have $\overline{X}_1^i, \overline{X}_2^i \in D$. Let us show that if for a given odd j such that $j \leq \frac{m}{2} - 3$ we have $\overline{X}_j^i, \overline{X}_{j+1}^i \in D$, then $\overline{X}_{j+2}^i, \overline{X}_{j+3}^i \in D$. Since $\overline{X}_j^i$ (resp. $\overline{X}_{j+1}^i$) dominates base intervals also dominated by X_{j+1}^i (resp. X_{j+2}^i), we have $X_{j+1}^i, X_{j+2}^i \notin D$. But there exists a base interval only adjacent to X_{j+1}^i and $\overline{X}_{j+2}^i$ (resp. X_{j+2}^i and $\overline{X}_{j+3}^i$). Thus, $\overline{X}_{j+2}^i, \overline{X}_{j+3}^i \in D$. By induction, if $X_1^i \notin D$ then each of the $\frac{m}{2}$ negative bridge intervals belong to D. Thus, none of the positive bridge intervals belong to D. □

Lemma 6.9. *If D is a dominating set of size K, then for any variable $x_i \neq \sigma(D)$, if $X_i \subset D$ then $\overline{W_i} \cap D = \emptyset$, otherwise $W_i \cap D = \emptyset$.*

Proof. By Lemma 6.8, D either contains X_i or contains $\overline{X_i}$.

If $X_i \subset D$, Lemma 6.8 ensures that $\overline{X}_i \cap D = \emptyset$. So the intervals $\overline{T}_j^i$ have to be dominated by other intervals.

By Lemma 6.7, ℓ_i intervals must dominate the V and $\overline{V}$ intervals of x_i. Since no interval dominates two of them, each $\overline{T}_j^i$ has to be dominated by an interval that is also dominating a V or $\overline{V}$ interval. The only interval that dominates both $\overline{T}_j^i$ and a V or $\overline{V}$ interval is $\overline{U}_j^i$. So all the $\overline{U}$ intervals are in D and $\overline{W} \cap D = \emptyset$ (since the only V or $\overline{V}$ interval dominated by a $\overline{W}$ interval is a $\overline{V}$ interval, which is already dominated).

Similarly if $\overline{X_i} \subset D$, Lemma 6.8 ensures that $X_i \cap D = \emptyset$. So the intervals T_j^i have to be dominated by other intervals. And one can prove similarly that these intervals should be the U intervals and then the W intervals are not in D. □

Safeness of the reduction. Let (F, A_s, A_t) be an instance of SATR, and let $D_s = D_F(A_s)$ and $D_t = D_F(A_t)$. By Lemma 6.6, (G_F, D_s, D_t) is an instance of DSR_{TS}. We can now show the first direction of our reduction.

Lemma 6.10. *If (F, A_s, A_t) is a yes-instance of* SATR, *then (G_F, D_s, D_t) is a yes-instance of* DSR_{TS}.

Proof. Let (F, A_s, A_t) be a yes-instance of SATR, and let $S =< A_1 := A_s, \ldots, A_\ell := A_t >$ be the reconfiguration sequence from A_s to A_t. We construct a reconfiguration sequence S' from D_s to D_t by replacing any flip of variable $x_i \rightsquigarrow \overline{x_i}$ of S from A_r to A_{r+1} by the following sequence of token slides from $D_F(A_r)$ to $D_F(A_{r+1})$ (and a $\overline{x_i} \rightsquigarrow x_i$ move is replaced by the converse of this sequence).

- We perform a sequence of slides that moves the token on J to $\overline{X}_1^i$. By Lemma 6.5, G_F is connected, and by Lemma 6.6, $D_F(A_r) \setminus J$ is a dominating set. So any sequence of moves from J to $\overline{X}_1^i$ keeps a dominating set.
- For any j such that $x_i \in C_j$, we first move the token from W_j^i to V_j^i then from V_j^i to U_j^i. Let us show that this keeps G_F dominated. The important intervals that can be dominated by W_j^i are V_j^i, C_j, and J. The vertex V_j^i is dominated anyway during the sequence since it is also dominated by V_j^i and U_j^i. Moreover, since $x_i \rightsquigarrow \overline{x_i}$ keeps F satisfied, each clause containing x_i has a literal different from x_i that also satisfies the clause. Thus, for each C_j such that $x_i \in C_j$, there exists an interval $W_j^{i'}$ or $\overline{W}_j^{i'}$, with $i' \neq i$, that belongs to $D_F(A_r)$, and then dominates both C_j and J during these two moves.
- For j from 1 to $\frac{m}{2} - 1$, we apply the move $X_j^i \rightsquigarrow \overline{X}_{j+1}^i$. This move is possible since X_j^i and $\overline{X}_{j+1}^i$ are neighbors in G_F. Let us show that this move keeps a dominating set. For $j = 1$, the important intervals that are dominated by X_1^i are B_1^i, B_2^i, and T_1^i. Since U_1^i is in the current dominating set (by the second point), T_1^i is dominated. Moreover B_1^i is dominated by $\overline{X}_1^i$, and B_2^i is a neighbor of $\overline{X}_2^i$. Thus, $X_1^i \rightsquigarrow \overline{X}_2^i$ maintains a dominating set. For $2 \leq j \leq \frac{m}{2} - 1$, the important intervals that are dominated by X_j^i are B_k^i, B_{k-2}^i and T_j^i where $k = 2j + 1$ if j is even and $k = 2j$ otherwise. Again T_j^i is dominated by the

U intervals. Moreover B^i_{k-2} is dominated by $\overline{X}^i_{j-1}$ (on which there is a token since we perform this sequence for increasing j), and B^i_k is also dominated by $\overline{X}^i_{j+1}$.

- For any j such that $\overline{x_i} \in C_j$, we move the token from $\overline{U}^i_j$ to $\overline{V}^i_j$ and then from $\overline{V}^i_j$ to $\overline{W}^i_j$. The important intervals dominated by $\overline{U}^i_j$ are the intervals $\overline{T}^i_j$, $\overline{V}^i_j$. But $\overline{T}^i_j$ is dominated by a negative bridge interval, and $\overline{V}^i_j$ stays dominated by $\overline{V}^i_j$ then $\overline{W}^i_j$.
- The previous moves lead to the dominating set $(D_F(A_{r+1}) \setminus J) \cup X^i_{\frac{m}{2}}$. We finally perform a sequence of moves that slide the token on $X^i_{\frac{m}{2}}$ to J. It can be done since Lemma 6.5 ensures that G_F is connected. And all along the transformation, we keep a dominating set by Lemma 6.6. As wanted, it leads to the dominating set $D_F(A_{r+1})$.

□

We now prove the other direction of the reduction. Let us prove the following lemma.

Lemma 6.11. *If there exists a reconfiguration sequence S from D_s to D_t, then there exists a reconfiguration sequence S' from D_s to D_t such that for any two adjacent dominating sets D_r and D_{r+1} of S', if both D_r and D_{r+1} have a moving variable, then it is the same one.*

Proof. Assume that, in S, there exist two adjacent dominating sets D_r and D_{r+1} such that both D_r and D_{r+1} have a moving variable, and $\sigma(D_r) \neq \sigma(D_{r+1})$. Let us modify slightly the sequence in order to avoid this move.

Since D_r and D_{r+1} are adjacent in S, we have $D_{r+1} = D_r \cup v \setminus u$, where uv is an edge of G_F. Since $\sigma(D_r) \neq \sigma(D_{r+1})$, u is an interval of $\sigma(D_r)$, and v an interval of $\sigma(D_{r+1})$. By construction, the only edges of G_F between intervals of different variables are between their $\{W, \overline{W}\}$ intervals. Thus, both u and v are W or $\overline{W}$ intervals and, in particular they are adjacent to the junction interval J. Moreover, the only important intervals that are adjacent to u (resp. v) are the V or $\overline{V}$ intervals of the same variable as u, W or $\overline{W}$ intervals, clause intervals, or the junction interval J. Since u and v are adjacent, and since they are both W or $\overline{W}$ intervals, they cannot be adjacent to the same clause interval. But the only intervals that are potentially not dominated by $D_r \setminus u = D_{r+1} \setminus v$ should be dominated both by u in D_r and by v in D_{r+1}. So these intervals are included in the set of W or $\overline{W}$ intervals and the junction interval, which are all dominated by J. Thus, $D_r \cup J \setminus u$ is a dominating set of G_F. Therefore, we can add in S the dominating set $D_r \cup J \setminus u$ between D_r and D_{r+1}. This intermediate dominating set has no moving variable. By repeating this procedure while there are adjacent dominating sets in S with different moving variables, we obtain the desired reconfiguration sequence S'. □

Lemma 6.12. *If (G_F, D_s, D_t) is a yes-instance of* DSR$_{\text{TS}}$*, then (F, A_s, A_t) is a yes-instance of* SATR.

Proof. Let (G_F, D_s, D_t) be a yes-instance of DSR$_{\text{TS}}$. There exists a reconfiguration sequence S' from D_s to D_t. Moreover, by Lemma 6.11, we can assume that for any two adjacent dominating sets D_r and D_{r+1} of S', if both D_r and D_{r+1} have a moving variable, then it is the same one.

Let us construct a reconfiguration sequence S from A_s to A_t. To any dominating set D of G_F, we associate a variable assignment $A(D)$ of F defined as follows. For any variable $x_i \neq \sigma(D)$, either $X_i \subset D$ or $\overline{X}_i \subset D$ by Lemma 6.8. If $X_i \subset D$ then we set $x_i = 1$. Otherwise, we set $x_i = 0$. Let x_k be such that $\sigma(D) = x_k$ if it exists. If there exists a clause interval C_j such that $W^k_j \in D$, and if for any $x_i \neq x_k$ with $x_i \in c_j$, we have $\overline{X_i} \subset D$, and for any $x_i \neq x_k$ with $\overline{x_i} \in c_j$, we have $X_i \subset D$, then we set $x_k = 1$. Otherwise $x_k = 0$.

Let S be the sequence of assignments obtained by replacing in S' any dominating set D by the assignment $A(D)$. In order to conclude, we must show that the assignments associated to D_s and D_t are precisely A_s and A_t. Moreover, for every dominating set D, the assignment associated to D has to satisfy F. Finally, for every move in G_F, we must be able to associate a (possibly empty) variable flip. Let us first show a useful claim, then proceed with the end of the proof.

Claim 6.1. *For any consecutive dominating sets D_r and D_{r+1} and any variable x_i that is not the moving variable of D_r nor D_{r+1}, the value of x_i is identical in $A(D_r)$ and $A(D_{r+1})$.*

Proof. Lemma 6.8 ensures that for any x_i such that $x_i \neq \sigma(D_r)$ and $x_i \neq \sigma(D_{r+1})$, either $X_i \subset D_r$ and $\overline{X_i} \cap D_r = \emptyset$ or $\overline{X_i} \subset D_r$ and $X_i \cap D_r = \emptyset$, and the same holds in D_{r+1}. Since the number of positive and negative bridge intervals is at least 2 (since by assumption m is a multiple of 4), and D_{r+1} is reachable from D_r in a single step, either both D_r and D_{r+1} contain X_i, or both contain $\overline{X_i}$. Thus, by definition of $A(D)$, for any variable x_i such that $x_i \neq \sigma(D_r)$ and $x_i \neq \sigma(D_{r+1})$, x_i has the same value in $A(D_r)$ and $A(D_{r+1})$. $\Diamond$

Claim 6.2. *We have $A(D_s) = A_s$ and $A(D_t) = A_t$.*

Proof. By definition, $D_s = D_F(A_s)$ and thus D_s contains the junction interval, which means that it does not have any moving variable. Moreover, D_s contains X_i for any variable x_i such that $x_i = 1$ in A_s and $\overline{X_i}$ for any variable x_i such that $x_i = 0$ in A_s. Therefore, for any variable x_i, $x_i = 1$ in A_s if and only if $x_i = 1$ in $A(D_s)$. Similarly, $A(D_t) = A_t$. $\Diamond$

Claim 6.3. *For any dominating set D of S', $A(D)$ satisfies F.*

Proof. Since the clause intervals are only adjacent to W and $\overline{W}$ intervals, they are dominated by them, or by themselves in D. But only one clause interval can belong to D. Thus, for any clause interval C_j, if $C_j \in D$, then C'_j must be dominated by a W or a $\overline{W}$ interval, that also dominates C_j. So in any case, C_j is dominated by a W or a $\overline{W}$ interval. We study four possible cases and show that in each case, c_j is satisfied by $A(D)$.

If C_j is dominated in D by an interval W^i_j, where $x_i \neq \sigma(D)$, then by Lemmas 6.8 and 6.9, $X_i \subset D$ and by definition of $A(D)$, $x_i = 1$. Since W^i_j exists, it means that $x_i \in c_j$, thus c_j is satisfied by $A(D)$.

Similarly, if C_j is dominated in D by an interval $\overline{W}^i_j$, where $x_i \neq \sigma(D)$, then by Lemmas 6.8 and 6.9, $\overline{X_i} \subset D$. So $x_i = 0$. Since $\overline{W}^i_j$ exists, $\overline{x_i} \in c_j$, and therefore c_j is satisfied by $A(D)$.

If C_j is only dominated by W^k_j in D, where $x_k = \sigma(D)$. Then, if there exists $x_i \neq x_k$ with $x_i \in c_j$ and $X_i \subset D$ (resp. $\overline{x_i} \in c_j$ and $\overline{X_i} \subset D$), then $x_i = 1$ (resp. $x_i = 0$) and c_j is satisfied by $A(D)$. So we can assume that, for any $x_i \neq x_k$ with $x_i \in c_j$ we have $X_i \not\subset D$. By Lemma 6.8, $\overline{X_i} \subset D$. And for any $x_i \neq x_k$ such that $\overline{x_i} \in c_j$ we have $\overline{X_i} \not\subset D$, and thus $X_i \subset D$. So, by definition of $A(D)$, we have $x_k = 1$. Since $x_k \in c_j$ (since W^k_j exists), c_j is satisfied by $A(D)$.

Finally, assume that C_j is only dominated by $\overline{W}^k_j$ in D, where $x_k = \sigma(D)$. If there exists $x_i \neq x_k$ such that $x_i \in c_j$ and $X_i \subset D$ (resp. $\overline{x_i} \in c_j$ and $\overline{X_i} \subset D$), then $x_i = 1$ (respectively $x_i = 0$) so c_j is satisfied by $A(D)$. Thus, by Lemma 6.8, we can assume that for any $x_i \neq x_k$ such that $x_i \in c_j$ (resp. $\overline{x_i} \in c_j$), we have $\overline{X_i} \subset D$ (resp. $X_i \subset D$). Let us show that there is no clause interval $C_{j'}$ dominated by a W^k_i interval of x_k in D and that satisfies, for any $x_i \neq x_k$, if $x_i \in c_{j'}$ then $\overline{X_i} \subset D$, and if $\overline{x_i} \in c_{j'}$ then $X_i \subset D$. This will imply $x_k = 0$ by construction and then the fact that c_j is satisfied.

Since D_s has no moving variable, there exists a dominating set before D in S' with no moving variable. Let D_r be the the latest in S' amongst such dominating sets. By assumption, $\sigma(D_q) = x_k$ for any set D_q that comes earlier than D but later than D_r. Thus, by Claim 6.1, for any variable $x_i \neq x_k$, x_i has the same value in $A(D_r)$ and $A(D)$.

Now, by assumption, for any $x_i \neq x_k$ with $x_i \in c_j$ (resp. $\overline{x_i} \in c_j$) we have $\overline{X_i} \subset D$ (resp. $X_i \subset D$). Thus, since x_i has the same value in D and D_r, if $x_i \in c_j$ (resp. $\overline{x_i} \in c_j$) then $\overline{X_i} \subset D_r$ (resp. $X_i \subset D_r$) and then, by Lemma 6.9, $W^i_j \notin D_r$ (resp. $\overline{W}^i_j \notin D_r$). Therefore, C_j is only dominated by $\overline{W}^k_j$ in D_r. But since D_r has no moving variable, $\overline{X_k} \subset D_r$ by Lemma 6.8 and Lemma 6.9. Thus, by Lemma 6.9, for any $j' \neq j$, $W^k_{j'} \notin D_r$. So for any $j' \neq j$ such that $x_k \in c_{j'}$, $C_{j'}$ is dominated by at least one interval $W^i_{j'}$ or $\overline{W}^i_{j'}$ in D_r, where $x_i \neq x_k$. Lemma 6.9 ensures that if $C_{j'}$ is dominated by $W^i_{j'}$ (resp. $\overline{W}^i_{j'}$) in D_r then $X_i \subset D_r$ (resp. $\overline{X_i} \subset D_r$), and since x_i has the same value in D and D_r, it gives $X_i \subset D$ (resp. $\overline{X_i} \subset D$). Therefore, by Lemma 6.8, if a clause interval $C_{j'}$ is dominated by a W interval of x_k in D, then either there exists $x_i \neq x_k$ such that $x_i \in c_{j'}$ and $D(\overline{x_i}) \not\subset D$, or there exists $x_i \neq x_k$ such that $\overline{x_i} \in c'_j$ and $D(x_i) \not\subset D$. By definition of $A(D)$, this implies that $x_k = 0$ in $A(D)$. Since $\overline{W}_j^k$ exists, $\overline{x_k} \in c_j$ thus c_j is satisfied by $A(D)$.

Therefore, every clause of F is satisfied by $A(D)$, which concludes the proof. $\Diamond$

Claim 6.4. *For any two dominating sets D_r and D_{r+1} of S', either $A(D_{r+1}) = A(D_r)$, or $A(D_{r+1})$ is reachable from $A(D_r)$ with a variable flip move.*

Proof. By Claim 6.1, for any variable x_i such that $x_i \neq \sigma(D_r)$ and $x_i \neq \sigma(D_{r+1})$, x_i has the same value in $A(D_r)$ and $A(D_{r+1})$. Moreover, by definition of S', if both D_r and D_{r+1} have a moving variable then $\sigma(D_r) = \sigma(D_{r+1})$. Therefore, at most one variable change its value between $A(D_r)$ and $A(D_{r+1})$, which concludes the proof. $\Diamond$ □

Theorem 6.4. DSR$_{\text{TS}}$ *is* PSPACE*-complete in circle graphs.*

Proof. Let $D_s = D_F(A_s)$ and $D_t = D_F(A_t)$. Lemma 6.10 and 6.12 ensure that (G_F, D_s, D_t) is a yes-instance of DSR$_{\text{TS}}$if and only if (F, A_s, A_t) is a yes-instance of SATR. Since SATR is PSPACE-complete [GKMP09], it gives the result. □

4 Line Graphs

We are interested here in the complexity of DSR$_{\text{TS}}$ in line graphs. An equivalent way to define the DSR$_{\text{TS}}$ problem in line graphs is to use edge-dominating sets. Two edge-dominating sets F, F' are TS-adjacent if $F' = (F \setminus e) \cup f$ and e and f share an endpoint. The DSR$_{\text{TS}}$ problem on line graphs is indeed equivalent to the EDSR$_{\text{TS}}$ problem defined as follows.

EDGE DOMINATING SET RECONFIGURATION UNDER TOKEN SLIDING (EDSR$_{\text{TS}}$)
Input: A graph G, two edge-dominating sets D'_s and D'_t of G
Output: Does there exist an edge-dominating set reconfiguration sequence from D'_s to D'_t in which any two consecutive dominating sets are TS-adjacent ?

We prove here the following.

Theorem 6.5. DSR$_{\text{TS}}$ *is* PSPACE*-complete in line graphs. In other words* EDSR$_{\text{TS}}$ *is* PSPACE*-complete.*

To do so, we use a polynomial time reduction from minVCR$_{\text{TS}}$ in graphs of maximum degree 3 to EDSR$_{\text{TS}}$. As discussed in Section 2.1, minVCR$_{\text{TS}}$ is PSPACE-complete in graphs of maximum degree 3, even when restricted to planar graphs [IDH+11], which gives the result.

The reduction. From any graph $G = (V, E)$ of maximum degree 3, we construct the *associated graph* G' as follows. For every vertex $u \in V$, we create the vertex-gadget Γ_u represented in Figure 6.7. The vertices u_1, u_2 and u_3 of Γ_u are called the *exit vertices* of Γ_u as they will be the only ones (possibly) connected with the rest of the graph. Let us arbitrarily order the edges of G and obtain, for every vertex $u \in V$, a natural ordering of the (at most three) edges incident to it in G.

For every edge $uv \in E$, we create the edge-gadget Γ_{uv} represented in Figure 6.7. If uv is the i-th edge incident to u and the j-th edge incident to v then Γ_{uv} contains the exit vertices u_i of Γ_u and v_j of Γ_v, as well as three new vertices. Note that each exit vertex belongs to at most one edge-gadget of G'. By abuse of notations, all along the proof, Γ_{uv} and Γ_{vu} will refer to the same edge-gadget. The construction is illustrated in Figure 6.8.

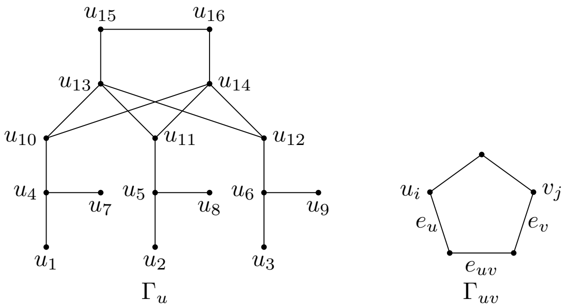


Figure 6.7: The gadgets Γ_u and Γ_{uv}

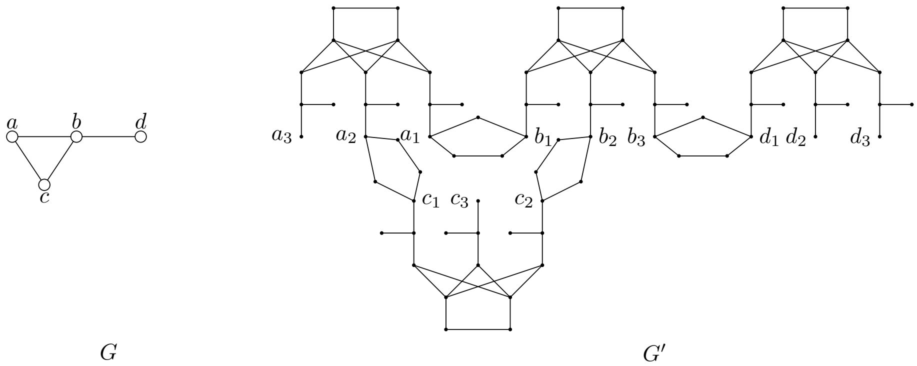


Figure 6.8: An example of a graph G and its associated graph G' obtained after the reduction. Note that since d has degree one, only one of its exit vertices is connected to the rest of the graph.

Let C be a minimum vertex cover of G. We construct a set $D(C)$ of edges of G' as follows.

- For any vertex $u \in C$, the set of edges of Γ_u in $D(C)$ is exactly the set E_u, with $E_u = \{u_1u_4, u_2u_5, u_3u_6, u_{13}u_{15}, u_{14}u_{16}\}$, represented in Figure 6.9. Each of these edges has multiplicity 1 in C.

- For any vertex $u \notin C$, the set of edges of Γ_u in $D(C)$ is exactly the set F_u, with $F_u = \{u_4u_{10}, u_5u_{11}, u_6u_{12}, u_{15}u_{16}\}$, represented in Figure 6.9. Each of these edges has multiplicity 1 in $D(C)$.

- For every edge uv such that $u, v \in C$, the only edge of Γ_{uv} in $D(C)$ is e_{uv}, with multiplicity 1.

- For any edge uv such that $u \notin C$, the only edge of Γ_{uv} in $D(C)$ is e_u, with multiplicity 1.

- For any edge uv such that $v \notin C$, the only edge of Γ_{uv} in $D(C)$ is e_v, with multiplicity 1.

Since C is a vertex cover of G, exactly one of the three last conditions holds. One can easily check that for any $u \in V$, E_u (resp. F_u) dominates Γ_u. Moreover, for any $uv \in E$, if $u \in C$ then the two edges of Γ_{uv} incident to an exit vertex u_i are dominated by an edge of E_u, and the three others are dominated either by e_v, or by e_{uv} and an edge of Γ_v, and similarly if $v \in C$. Thus, since C is a vertex cover of G, $D(C)$ is an edge-dominating set of G'. We say that $D(C)$ is the edge-dominating set *associated to* C.

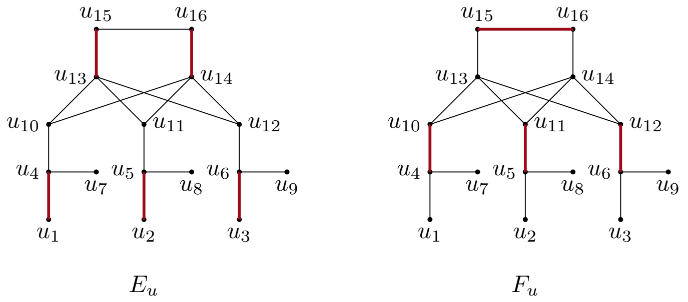


Figure 6.9: The edge sets E_u and F_u.

Theorem 6.5 is a direct consequence of the following lemma.

Lemma 6.13. *Let G be a graph of maximum degree 3. Let C_s, C_t be two minimum vertex covers of G and let $D_s = D(C_s)$ and $D_t = D(C_t)$. Then (G', D_s, D_t) is a yes-instance of* EDSR$_{\text{TS}}$ *if and only if (G, C_s, C_t) is a yes-instance of min*VCR$_{\text{TS}}$.

Proof. ($\Leftarrow$) Let (G, C_s, C_t) be a yes-instance of minVCR$_{\text{TS}}$. There exists a reconfiguration sequence $S =< C_1 := C_s, C_2 \ldots, C_\ell := C_t >$ in G. Let us prove that, for every $r \leq \ell - 1$, there exists a reconfiguration sequence S' from $D(C_r)$ to $D(C_{r+1})$, which gives the conclusion.

We perform the following sequence of edge slides:

- For every edge ux with $x \neq v$, we move e_{ux} onto e_u. Note that there is a token on e_{ux} since $C_r \setminus u$ is a vertex cover of $G[E \setminus \{uv\}]$ and thus $x \in C_r$. Note moreover that all the edges of Γ_{ux} are and will be dominated regardless of the modifications we will perform in Γ_u (since an edge incident to x_j is in $D(C_r)$ as $x \in C_r$).
- If u_i is the exit vertex of Γ_u such that $u_i \in \Gamma_{uv}$, then we make the following moves. The first two moves are $u_p u_{p+3} \rightsquigarrow u_{p+3} u_{p+9}$ and $u_q u_{q+3} \rightsquigarrow u_{q+3} u_{q+9}$ in Γ_u, where u_p and u_q are the two exit vertices of Γ_u different from u_i. The next moves are $u_{13} u_{15} \rightsquigarrow u_{13} u_{i+9}$, $u_{14} u_{16} \rightsquigarrow u_{15} u_{16}$, $u_{13} u_{i+9} \rightsquigarrow u_{i+3} u_{i+9}$, then $u_i u_{i+3} \rightsquigarrow e_u$ where e_u is the edge of Γ_{uv} incident to u_i. Note that these moves are indeed possible since $u \in C_r$. Note that the tokens of Γ_u are the ones desired in $D(C_{r+1})$. Moreover, all along the transformation, all the edges in Γ_u are dominated and so all the intermediate sets are edge-dominating.
- In Γ_u, we currently have tokens on the edges $u_{15}u_{16}, u_4 u_{10}, u_5 u_{11}$ and $u_6 u_{12}$ and the edge e_u on Γ_{uv}. Note that we have exactly the same in Γ_v. So we can perform on Γ_v the converse of the sequence we just applied in Γ_u in order to get the tokens on $\{v_1 v_4, v_2 v_5, v_3 v_6, v_{13} v_{15}, v_{14} v_{16}\}$ in Γ_v. Note that there is a token on e_v in Γ_{vx} for every $x \neq v$ since $v \notin C_r$ by minimality of C_{r+1}. Thus all the edges of Γ_{vx} are dominated all along the transformation.
- Finally, for every edge vx with $x \neq u$, we move e_v onto e_{vx}.

One can easily notice that the resulting set is $D(C_{r+1})$, which completes this part of the proof.

($\Rightarrow$) Let (G', D_s, D_t) be a yes-instance of EDSR$_{\text{TS}}$, and let us prove that (G, C_s, C_t) is a yes-instance of minVCR$_{\text{TS}}$. To do so let us prove that we can associate a vertex cover of G to each edge-dominating set of G' and that every move of a token either does not modify the associated vertex cover or corresponds to a token slide.

For any $uv \in E$, we define an arbitrary order such that either $u < v$ or $v < u$. Let D be an edge-dominating set of G'. For any vertex-gadget Γ_u, let $p(u)$ be the number of edges of Γ_u in D. For $k \in \{1, 2, 3\}$, let $q_k(u)$ be 0 if u_k does not belong to any edge-gadget (i.e. u has degree less than three and the exit vertex is unused) or if $v > u$ where v is such that the edge gadget Γ_{uv} containing u_k, and let $q_k(u)$ be the number of tokens in Γ_{uv} containing u_k, minus one, otherwise. To the edge-dominating set D of G', we associate a set $C(D)$ of vertices of G, such that for any $u \in V$, if $p(u) + \sum_{k=1}^{3} q_k(u) \geq 5$ then u belongs to $C(D)$ with multiplicity $p(u) + \sum_{k=1}^{3} q_k(u) - 4$, and $u \notin C(D)$ otherwise. In order to conclude, we have to show that the sets associated to D_s and D_t are precisely the initial and target vertex covers of G. Moreover, for every edge-dominating set D, the set associated to D has to be a vertex cover of G. And finally, for every move in G', we must be able to associate a (possibly empty) move in G.

Firstly, $D_s = D(C_s)$ and thus by definition D_s contains exactly one edge in each edge-gadget, four edges in each vertex-gadget Γ_u such that $u \notin C_s$, and five edges in each vertex-gadget Γ_u such that $u \in C_s$. Therefore, if $u \notin C_s$ then in D_s, $p(u) + \sum_{k=1}^{3} q_k(u) = 4$, thus $u \notin C(D_s)$, and if $u \in C_s$ (necessarily with multiplicity 1 since C_s is minimum) then in D_s, $p(u) + \sum_{k=1}^{3} q_k(u) = 5$ and then $u \in C(D_s)$ with multiplicity 1. The same arguments holds for C_t. Thus, $C(D_s) = C_s$ and $C(D_t) = C_t$.

Let us show that for every edge-dominating set D of G', $C(D)$ is a vertex cover of G. By construction, for any vertex $u \in V$, in the vertex-gadget Γ_u, only the edges u_1u_4, u_2u_5 and u_3u_6 can be dominated by edges outside of Γ_u. But even if we remove these edges, the only minimum edge-dominating set of Γ_u is F_u (see Figure 6.9 for an illustration), which contains four edges. Thus, for any $uv \in E$, if D contains only 4 edges of Γ_u and 4 edges of Γ_v, then D contains F_u and F_v, which implies that D contains at least two edges of Γ_{uv} to dominate Γ_{uv}. Thus, $q_i(u) \geq 1$ or $q_j(v) \geq 1$, where u_i and v_j are the exit vertices that belong to Γ_{uv}, and therefore $p(u) + \sum_{k=1}^{3} q_k(u) \geq 5$ or $p(v) + \sum_{k=1}^{3} q_k(u) \geq 5$, which implies $u \in C(D)$ or $v \in C(D)$. Therefore, for any edge $uv \in E$, $u \in C(D)$ or $v \in C(D)$ and by definition, $C(D)$ is a vertex cover of G.

Finally, let us show that for any two adjacent edge-dominating sets D_r and D_{r+1} of G', either $C(D_r) = C(D_{r+1})$, or there exists $uv \in E(G)$ such that $u \in C_r$ and $C_{r+1} = C_r \cup v \setminus u$. We have $D_{r+1} = D_r \cup e \setminus f$, where $e, f \in E$ and e and f share a vertex. Note that if e belongs to the vertex gadget Γ_u then f can only belong to Γ_u or to Γ_{uv} with $uv \in E$. Similarly, if $e \in \Gamma_{uv}$ either f also belongs to it or it belongs to Γ_u or Γ_v. If u and v belong to the same gadget, neither the p nor the q functions are modified and thus $C(D_{r+1}) = C(D_r)$. So, free to permute D_r and D_{r+1} we can assume that $e \in \Gamma_u$ and $f \in \Gamma_{uv}$. Note that $p(u)$ decreases by 1 and $p(v)$ is not modified. If $u > v$ then $q_i(u)$ increases by one and $q_j(v)$ is not modified, where u_i and v_j are the exit vertices that belong to Γ_{uv}. So $C(D_{r+1}) = C(D_r)$. If $u < v$, $q_j(v)$ increases by one and $q_i(u)$ is not modified, so the multiplicity of u decreases by one and the multiplicity of v increases by one, which corresponds to a move $u \rightsquigarrow v$ in G. So any move in G' corresponds to a move in G, which completes the proof. □

5 Circular Arc Graphs

As stated in Theorem 6.1, there always exists a transformation between two dominating sets of identical size in interval graphs. Since the class of circular arc graphs strictly contain the class of interval graphs (since a long cycle is a circular arc graph while it is not an interval graph), one can naturally wonder if this result can be extended to circular arc graphs. The answer is negative since, for every k, the cycle C_{3k} of length $3k$ is a circular arc graph and it contains three isolated dominating sets of size exactly k (the ones containing vertices i mod 3 for $i \in \{0, 1, 2\}$). However, we show here the following:

Theorem 6.6. DSR_{TS} *is polynomial in circular arc graphs.*

The rest of this section is devoted to prove Theorem 6.6. Let $G = (V, E)$ be a circular arc graph and D_s, D_t be two dominating sets of G of the same size.

Assume first that an arc $A \in V$ contains the whole circle. So A dominates G and then for any two dominating sets D_s and D_t of G, we can move a token from D_s to A, then move every other other token of D_s to a vertex of D_t (in at most two steps passing through A), and finally move A to the last vertex of D_t. Since a token is on A all along the transformation, we indeed have a dominating set. Thus, there exists a reconfiguration sequence from D_s to D_t. From now on we assume that no arc of V contains the whole circle.

For any arc $A_i \in V$, we call *left extremity* of I_i the first extremity of A_i we meet when we follow the circle clockwise, starting from a point outside of A_i. The other extremity of A_i is called the *right extremity* of A_i.

Let us first prove the following straightforward lemma.

Lemma 6.14. *Let G be a graph, and let u and v be two vertices of G such that $N(u) \subseteq N(v)$. If S is a dominating set reconfiguration sequence in G, and S' is obtained from S by replacing every occurrence of u by v in the dominating sets of S, then S' is also a dominating set reconfiguration sequence in G.*

Proof. Every neighbor of u also is a neighbor of v. Thus, replacing u by v in a dominating set keeps the domination of G. Moreover, any move that involves u can be applied if we replace it by v, which gives the result. □

For the proof of Theorem 6.6, we need to define a graph G_u which is an interval graph. We construct it in the following way. Let u be an interval of G that is maximal by inclusion. For any arc $v \in V$ such that $v \neq u$, replace v by $v' := (v \setminus u) \cup P$ where P is the set of extremities of u that belong to v (if it exists). Since, by maximality of u, v does not contain u, v' is indeed an arc. Let G'_u be the resulting circular arc graph. Note that the set of edges in G'_u might be smaller than the one of G but any dominating set of G containing u is still dominating in G'_u. Now remove from G'_u the vertex u, as well as any vertex whose interval is now empty (in other words, the intervals of G that are strictly included in u). And create two new vertices, u' and u'', that correspond to each extremity of u. Since no arc intersects u in G'_u but on its extremities, we can also create $(n+2)$ new vertices which are only adjacent to u' and $(n+2)$ new vertices that are only adjacent to u'' (these vertices are called *leaves*). The resulting graph is the interval graph G_u. Note that we can assume that G_u is connected (otherwise G is an interval graph and we can conclude with Theorem 6.1). The construction is illustrated on Figure 6.10.

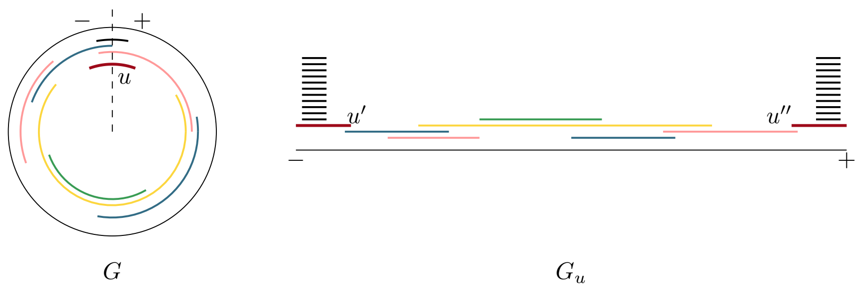


Figure 6.10: The linear interval graph G_u obtained from the circular arc graph G.

Let us prove three useful lemmas.

Lemma 6.15. *Let D be a dominating set of G such that $u \in D$, and let D_u be the set $D \cup \{u', u''\} \setminus \{u\}$. The set D_u is a dominating set of G_u.*

Proof. Every vertex of $N(u)$ in the original graph G is either not in G_u, or is dominated by u' or u''. The neighborhood of all the other vertices have not been modified. Moreover, all the new vertices are dominated since they are all adjacent to u' or u''. □

Note that D_u has size $|D|+1$.

Lemma 6.16. *The following holds:*

- ***(i)*** *All the dominating sets of G_u of size $|D|+1$ contain u' and u''.*
- ***(ii)*** *For every dominating set X of G_u of size $|D|+1$, $(X \cap V) \cup \{u\}$ is a dominating set of G of size at most $|D|$.*
- ***(iii)*** *Every reconfiguration sequence in G_u between two dominating sets D_s, D_t of G_u of size at most $|D|+1$ and that does not contain any leaf can be adapted into a reconfiguration sequence in G between $(D_s \setminus \{u', u''\}) \cup \{u\}$ and $(D_t \setminus \{u', u''\}) \cup \{u\}$.*

Proof. The point (i) holds since there are $n+2$ pending vertices attached to each of u' and u'' and $|D| \leq n$.

The point (ii) is indeed true since we have not created any edge between vertices of V in the construction of G_u, since u' and u'' only dominate vertices of V dominated by u in G, and since u' and u'' are in any dominating set of size at most $|D|+1$ of G_u.

The point (iii) follows since we simply have to slide the tokens along the same edge (no edge is created) and move the token on (resp. from) u if a token is slid on (resp. from) u' or u'' (and do nothing is a token is slid on or from a leaf attached to u' or u''). $\square$

By Lemma 6.16 and Theorem 6.1, we immediately obtain the following corollary:

Corollary 6.1. *Let G be a circular interval graph, $u \in V(G)$, and k be an integer. All the k-dominating sets of G containing u are in the same connected component of the reconfiguration graph.*

Let us now prove Theorem 6.6. Let $G=(V,E)$ be a circular arc graph, and let D_s and D_t be two dominating sets of G. Free to slide tokens, we can assume that all the intervals of D_s and D_t are maximal by inclusion. Moreover, by Lemma 6.14, the vertices of all the dominating sets we will consider from now on will be maximal by inclusion. By abuse of notation, we will say that in G, an arc v is the first arc *on the left* (resp. *on the right*) of another arc u if the first left extremity of an inclusion-wise maximal arc (of G, or of the stated dominating set) we encounter when browsing the circle counter clockwise (resp. clockwise) from the left extremity of u is the one of v. On the other hand, in interval graphs, we say that an interval v is at the left (resp. at the right) of an interval u if the left extremity of v is inferior (resp. superior) to the one of u. Note that since the intervals we consider here are maximal by inclusion, it implies that it is also the case for their right extremities.

Let u_1 be a vertex of D_s. Let v be the first vertex at the right of u_1 in D_t. We perform the following algorithm, called the Right Sliding Algorithm. By Lemma 6.16, all the dominating sets of size $|D_s|+1$ in G_{u_1} contain u_1'. Let D_2' be a dominating set of the interval graph G_{u_1} of size $|D_s|+1$, such that the first vertex at the right of u_1' has the lowest left extremity (we can indeed find such a dominating set in polynomial time). By Theorem 6.1, there exists a transformation from $(D_s \cup \{u_1', u_1''\}) \setminus \{u_1\}$ to D_2' in G_{u_1}. And thus by Lemma 6.16, there exists a transformation from D_s to $D_2 := (D_2' \cup \{u_1\}) \setminus \{u_1', u_1''\}$ in G. We apply this transformation. Informally speaking, this ensures that the first vertex on the left of u_1 in D_2 is as close to u_1 as possible. Now, we fix all the vertices of D_2 but u_1, and we try to slide the token on u_1 onto the first arc on the right of u_1 in G. Let u_2 be this arc. We now repeat these operations with u_2 instead of u_1, i.e. we apply a reconfiguration sequence towards a dominating set of G in which the first vertex on the left of u_2 is the closest to u_2, then try to slide u_2 to the right, onto u_3. We repeat these operations until $u_i = u_{i+1}$ (i.e. we cannot move to the right anymore) or until $u_i = v$. Let $u_1, \ldots, u_\ell$ be the resulting sequence of vertices. Note that this algorithm is indeed polynomial since after at most n steps we arrive to v or we cannot move to the right anymore.

We can similarly define the Left Sliding Algorithm by replacing the leftmost dominating set of G_{u_i} by the rightmost, and then slide u_i to the left for any i. We stop when we cannot slide to the left anymore, or when $u_i = v'$, where v' is the first vertex at the left of u_1 in D_t. Let u_ℓ' be the last vertex of the sequence of vertices given by the Left Sliding Algorithm.

Let us prove that there exists a transformation from D_s to D_t if and only if $u_\ell = v$ or $u_\ell' = v'$. Firstly, if $u_\ell = v$, then Corollary 6.1 ensures that there exists a transformation from D_ℓ to D_t and thus from D_s to D_t, and similarly if $u_\ell' = v'$. Now, if $u_\ell \neq v$ and $u_\ell' \neq v'$, assume for contradiction that there exists a transformation sequence S from D_s to D_t. By Lemma 6.14 we can assume that all the vertices in any dominating set of S are maximal by inclusion. Moreover, free to decompose the moves in S, we can assume that in S, the moves are always from a vertex to the first vertex at its right or its left in G.

Let us consider the first dominating set C of S where the token initially on u_1 is on the first vertex on the right of u_ℓ in G, or on the first vertex on the left of u_ℓ' in G. Such a dominating set exists since v is the first vertex on the right of u in D_t and v' the first vertex on the left of u in D_t, and since the Right Sliding Algorithm and Left Sliding Algorithm stopped before reaching a vertex of D_t. Without loss of generality, we can assume that it is on the right of u_ℓ (the case where it is on the left of u_ℓ' is symmetrical). Let C' be the dominating set just before C in S. Note that $u_\ell \in C'$. Let x be the first vertex on the right of u_ℓ in G, i.e. the vertex on which the token on u_ℓ slides from C' to C. Let y be the first vertex of C' at the left of u_ℓ.

The dominating set D'_ℓ of the interval graph G_{u_ℓ} has size $|D_s|+1$ and is such that the first vertex at the right of u'_ℓ has the lowest left extremity. We denote by z this vertex. In particular, y is at the right of z in G_{u_ℓ}. We claim that, in D_ℓ, u_ℓ can be slid on x. Indeed, since the token on u_ℓ slides to the first vertex on the right, if a vertex is not dominated, it is a vertex that contains the left extremity of u_ℓ. But any such vertex is dominated by y in C and thus by z in D_ℓ, since z is at the left of y in G_{u_ℓ}. Thus sliding u_ℓ on x is possible. It gives a contradiction with $u_\ell = u_{\ell+1}$.

6 Conclusion

In this chapter, we investigated the reconfiguration of dominating sets under the token sliding rule. We studied the complexity of the reachability problem in several classes of graphs. With the same reduction from minVCR$_{\text{TS}}$ in planar graphs of maximum degree at most 3, we proved that the problem is PSPACE-complete in both planar bipartite graphs and unit disk graphs. Note that Bonamy, Dorbec and Ouvrard [BDO19] already proved that it was PSPACE-complete in planar graphs and bipartite graphs. We then showed that it is also PSPACE-complete in circle graphs, answering an open question raised by Bonamy, Dorbec and Ouvrard [BDO19]. To do so, we used an adaptation of the proof of the NP-completeness of the source problem [Kei93]. We then focused on line graphs, and proved that the problem is also PSPACE-complete for this class, using a reduction from minVCR$_{\text{TS}}$ in graphs of maximum degree at most 3. This is equivalent to saying that the reachability problem is PSPACE-complete in general for the reconfiguration of edge-dominating sets under the token sliding rule. On the positive side, we provide a polynomial time algorithm for circular arc graphs, using a result of Bonamy et al. stating that there always exists a reconfiguration sequence in interval graphs. Note that the class of circular arc graphs contains the class of circular interval graphs, and this result thus answers another question raised by Bonamy, Dorbec and Ouvrard [BDO19]. All these results complete their work on the complexity of the reachability problem in other graph classes, and both are illustrated in Figure 6.11.

The questions raised by Bonamy, Dorbec and Ouvrard about circle graphs was a possible lead to answer the following question: does there exist a graph class for which computing a minimum dominating set is NP-complete but the reachability problem is polynomial ? In this chapter, we proved that the reachability problem is PSPACE-complete in circle graphs, with an adaptation of the proof of the NP-completeness of the source problem. Therefore, the question remains open, and the complexity of the reachability problem needs to be studied in other graph classes. In particular, for outerplanar graphs, which is a maximal subclass of circle graphs, the complexity of the reachability problem is open, even if finding a minimum dominating set is polynomial. We strongly believe that the problem is in P, but we were not able to prove it.

Since the reconfiguration of dominating sets under the token sliding rule has been studied only recently, there remains a lot of work to be done. For example, one could study the complexity of the shortest transformation problem, or the diameter when the reconfiguration graph is connected (for interval graphs, for example). A version where only one token can be on each vertex could also be investigated. This version has somehow been undirectly studied through eternal domination. In this problem, the goal is to start from a dominating set and to go to another that has to contain a given vertex, then to another containing another vertex, etc. The next chapter is devoted to the eternal domination problem and its variants.

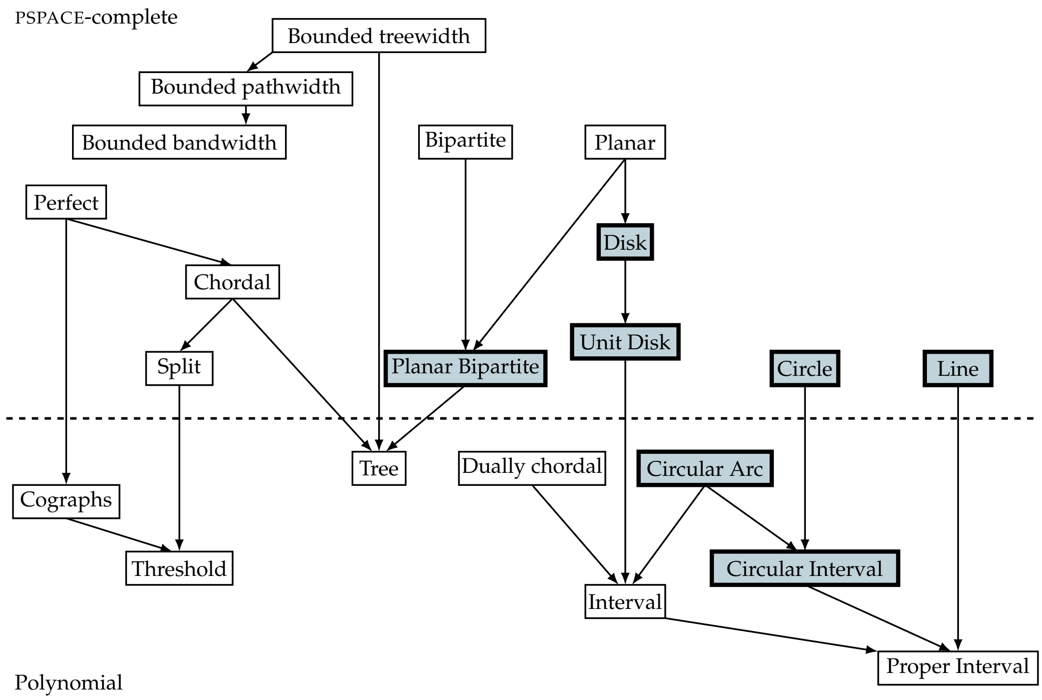


Figure 6.11: The complexity of DSR$_{TS}$ in several graph classes. The rectangles in blue are the results we show in this chapter and the other ones are previously known results.

Chapter 7

Eternal Domination

1 Introduction

In this chapter, we are interested in the eternal domination problem. It can be seen as an infinite game played on a graph G between two players: the defender and the attacker. The defender starts by choosing a set D_0 of vertices of G, where they place some guards. Then, at each turn i, the attacker chooses a vertex r_i called the attack in $V \setminus D_{i-1}$, and the defender must defend against the attack, by moving to r_i a guard on a vertex v_i adjacent to r_i (thus sliding exactly one guard along an edge of G). The new guards configuration is therefore $D_i = (D_{i-1} \cup \{r_i\}) \setminus \{v_i\}$. The defender wins the game if it can defend against any infinite sequence of attacks. The eternal domination number, denoted by $\gamma^\infty(G)$, is the minimum number of guards necessary for the defender to win. An eternal dominating set is a set that can initially be chosen by the defender in a winning strategy.

Let us give a more formal definition of these notions. Let $G = (V, E)$ be a graph. The set $EDS(G)$ of *eternal dominating sets* of G is the greatest set of subsets of V such that for every $S \in EDS(G)$ and every $r \in V \setminus S$, there exists $v \in S$ such that $vr \in E$ and $(S \cup \{r\}) \setminus \{v\} \in EDS(G)$. The *eternal domination number* of G is defined as $\gamma^\infty(G) = \min\{|S| : S \in EDS(G)\}$.

Another variant of the eternal domination problem is the m-eternal domination problem. Note that the m of m-eternal does not represent a parameter, but stands for multiple, as in multiple guards can move at a time. In this variant, every turn, the defender is authorized to move any number of guards at a time. Each guard thus either stays on its current vertex, or moves to a neighbor of its current vertex, provided that the vertices occupied by the guards are all distinct. The attack is defended against if and only if one guard moves to the attacked vertex.

More formally, we give the following definitions. Given two sets $S_1, S_2 \subseteq V$, a *multimove* f from S_1 to S_2 is a one-to-one mapping from S_1 to S_2 such that for every $x \in S_1$, we have $f(x) = x$ or $xf(x) \in E$. The set $MEDS(G)$ of *m-eternal dominating sets* of G is the greatest set of subsets of V such that for every $S \in MEDS(G)$ and every $r \in V(G) \setminus S$, there is a multimove f such that $r \in f(S)$ and $f(S) \in MEDS(G)$. The *m-eternal domination number* of G is $\gamma_m^\infty(G) = \min\{|S| : S \in MEDS(G)\}$.

Figure 7.1 illustrates the game played between the defender and the attacker in both eternal domination and m-eternal domination.

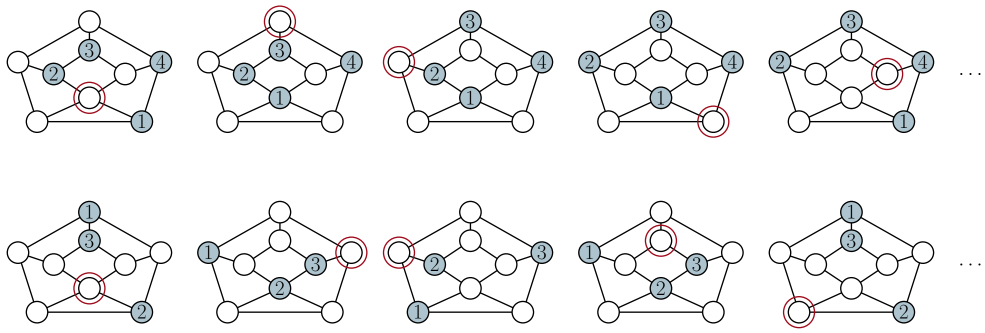


Figure 7.1: Above, an example of eternal domination of a graph. Below, an example of m-eternal domination of a graph. Each number represents a guard, and the attack is circled.

The study of eternal domination finds its origins in its application to military defense introduced in the 90's. The goal was to investigate the military strategy of Emperor Constantine used to defend the Roman Empire [AF95, ReV97, RR00, Ste99]. It was then formally introduced by Burger et al. [BCG+04] in 2004, and has been widely studied ever since. The first studies mainly focused on the search of general bounds on the eternal domination number, depending on usual graph parameters. Burger et al. made the following observation in their paper. Recall that $\theta(G)$ denotes the minimum number of cliques in which G can be partitioned.

Observation 7.1. *[BCG+04] Given a graph G, we have $\alpha(G) \leq \gamma^\infty(G) \leq \theta(G)$.*

Indeed, if the attacker attacks consecutively all the vertices of a maximum independent set I of G, and if the defender is able to defend G, then each vertex of I is occupied by a guard when the attacker is done. And the second inequality is straightforward when considering that each clique of G can be defended by one guard. However, Klostermeyer and MacGillivray proved that none of these two bounds are tight, as stated in the following theorem.

Theorem 7.1. *[KM09] For any integers k and k', there exists a graph G such that $\gamma^\infty(G) \geq \alpha(G) + k$ and $\theta(G) \geq \gamma^\infty(G) + k'$.*

Goddard et al. gave the first example of a graph G such that $\alpha(G) < \gamma^\infty(G) < \theta(G)$ [GHH05]. On the other hand, for any $k \geq 3$, Klostermeyer and MacGillivray [KM05] proved the existence of graphs with $\gamma^\infty(G) = \alpha(G)$ and $\theta(G) = k$.

Observation 7.1 is particularly interesting when G is a perfect graph. Then, since the complement $\overline{G}$ of G is also perfect [LC05], by definition, $\omega(\overline{G}) = \chi(\overline{G})$, which is equivalent to $\alpha(G) = \theta(G)$. Thus, it gives $\gamma^\infty(G) = \alpha(G) = \theta(G)$. Finding the graphs for which $\gamma^\infty(G) = \theta(G)$ have been the object of many studies and several necessary conditions have been found [ABB+07, BCG+04, KM09, Reg07]. In particular, circular arc graphs [Reg07] and K_4-minor free graphs [ABB+07] verify this property.

Another famous bound on $\gamma^\infty(G)$ is given by the following theorem, proved by Klostermeyer and MacGillivray [KM07].

Theorem 7.2. *[KM07] For any graph G, we have $\gamma^\infty(G) \leq \binom{\alpha(G)+1}{2}$.*

Goldwasser and Klostermeyer gave some graphs for which the equality is reached [GK08].

In the paper of Burger et al. [BCG+04], the exact value of $\gamma^\infty(G)$ is given for paths, cycles, multipartite graphs, grids and rook's graphs. In toroidal grids, they obtain the following bounds.

Theorem 7.3. *[BCG+04] For any integers p and q, we have* $\frac{7nm}{23} \leq \gamma^\infty(P_n \Box P_m) \leq \lceil \frac{nm}{2} \rceil$.

More generally, in cartesian product of graphs, a Vizing-like question asks if it is always true that $\gamma^\infty(G \Box H) \geq \gamma^\infty(G) \cdot \gamma^\infty(H)$. No counter example is known to this day. This type of question was studied in [DKK+20]. For the strong products, the authors proved that for any graphs G and H, we have $\gamma^\infty(G \boxtimes H) \geq \alpha(G) \cdot \gamma^\infty(H)$. They also provided some families of graphs attaining the strict inequality $\gamma^\infty(G \Box H) > \gamma^\infty(G) \cdot \gamma^\infty(H)$ for the cartesian product.

The m-eternal domination problem was introduced a year later than the eternal domination one. The first paper to mention it is the one of Goddard et al. [GHH05]. In the same paper, they give the following bounds on the value of $\gamma_m^\infty(G)$.

Theorem 7.4. *[GHH05] For any graph G, we have* $\gamma(G) \leq \gamma_m^\infty(G) \leq \alpha(G)$.

The upper bound $\alpha(G)$ is not always reached, as proves the example of the star S_n for which $\gamma_m^\infty(S_n) = 2$ and $\alpha(S_n) = n - 1$. That being said, some graphs reach the equality, for instance the complete graphs, for which $\gamma_m^\infty(K_n) = \alpha(K_n) = 1$.

With Obervation 7.1, Theorem 7.2 and Theorem 7.4, we get the following nice sequence of inequalities.

Theorem 7.5. *[BCG+04, GHH05, KM07] For any graph G, we have*

$$\gamma(G) \leq \gamma_m^\infty(G) \leq \alpha(G) \leq \gamma^\infty(G) \leq \binom{\alpha(G)+1}{2}.$$

Other upper bounds on $\gamma_m^\infty(G)$ include $\lfloor \frac{n}{2} \rfloor$ [CKP06], $2\gamma(G)$ [KM16], and $2\tau(G)$ [KM12b] (recall that $\tau(G)$ denotes the size of a minimum vertex cover of G).

In the paper that introduces the m-eternal domination problem, Goddard et al. studied the value of $\gamma_m^\infty(G)$ in several graph classes. They obtained the exact value for cycles, paths, complete graphs and complete bipartite graphs [GHH05]. In 2015, Braga, de Souza and Lee also solved the problem in proper interval graphs [BdSL15], generalized in 2019 to interval graphs by Rinemberg and Soulignac [RS19]. The value of the m-eternal domination number is also known in split graphs [BDE+17], and upper bounded in cacti [BKV19].

As it is often the case in domination problems, a lot attention has been given to grid graphs. Godwasser, Klostermeyer and Mynhardt obtained the value of $\gamma_m^\infty(G)$ for grids of dimensions $2 \times n$ [GKM13]. They also provided an upper bound for grids of dimensions $3 \times n$, improved later by Finbow, Messinger and van Bommel, who also gave a lower bound [FMvB15]. Beaton, Finbow and MacDonald [BFM13] continued this investigation and gave the exact value for the dimensions $6 \times n$, and lower and upper bounds for the dimensions $4 \times n$. The $5 \times n$ grid was studied by Van Bommel and Van Bommel who found lower and upper bounds [vBvB16]. Recently, the asymptotic value of $\gamma_m^\infty(G)$ was obtained for any dimensions. Indeed, Lamprou, Martin and Schewe proved that $\gamma_m^\infty(P_n \Box P_m) = \lceil \frac{mn}{5} \rceil + \mathcal{O}(n+m)$ [LMS19]. Concerning king's grids (i.e. strong products of paths), McInerney, Nisse and Pérennes proved in 2019 that $\gamma_m^\infty(P_n \boxtimes P_m) = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil + \mathcal{O}(m\sqrt{n})$ [MINP19], and Virgile et al. that $\gamma_m^\infty(P_n \boxtimes P_m) \leq \frac{mn}{7} + \mathcal{O}(m+n)$ [VSZ+20].

The complexity of deciding whether the m-eternal domination number of a graph is at most a given k has been recently studied. It was proven to be NP-complete when restricted to Hamiltonian split graphs [BDE+17]. In the general case, we do not know if the problem is in NP. For other graph classes such as strongly chordal split graphs [BDE+17], the problem is in P, and even linear in interval graphs [RS19] and cacti [BKV19].

Several variants of the eternal and m-eternal domination problems have been studied. For instance, Klostermeyer and Mynhardt [KM12a] introduced the eternal total domination and eternal connected domination problems. More recenty, an eviction model was introduced, where a guarded vertex is attacked at each turn and the guards have to avoid it [KMAA17]. It was also recently proved [FGMO18] that in the m-eternal domination problem, contrarily to what stated a conjecture of Goddard in [GHH05], allowing multiple guards to occupy a same vertex is an advantage for the defender. This led to several studies in which such guards configurations are authorized [B+19, BKV19]. For a more complete state of the art concerning eternal domination and its variants, the reader is referred to the review of Klostermeyer and Mynhardt [KM16]. In this chapter, we present a joint work with Guillaume Bagan and Hamamache Kheddouci, where we investigate a directed and an oriented version of the eternal and m-eternal domination problems.

2 Eternal domination on digraphs

We present here the eternal domination and m-eternal domination problems on directed graphs. The definitions of $EDS(G)$, $\gamma^\infty(G)$, $MEDS(G)$ and $\gamma_m^\infty(G)$ are straightforward, the only difference is that whenever an edge is considered, we consider an arc instead, and the movements of the guards must follow the direction of the arcs.

Note that the notion of domination in digraphs has been defined in multiple ways in the literature, also, we use the definition where a vertex can only be dominated by itself or its incoming neighbors, also called out-domination.

It is straightforward that the (m-)eternal domination number of a graph is the sum of the (m-)eternal domination number of each of its connected components. We prove that this result can be extended to the strongly connected components (s.c.c) of a digraph.

Lemma 7.1. *Let G be a digraph with strongly connected components $S_1, \dots S_l$. Then*

$$\gamma^\infty(G) = \sum_{i=1}^{l} \gamma^\infty(G[S_i])$$

and

$$\gamma_m^\infty(G) = \sum_{i=1}^{l} \gamma_m^\infty(G[S_i]).$$

Proof. The statement $\gamma^\infty(G) \leq \sum_{i=1}^{l} \gamma^\infty(G[S_i])$ (resp. $\gamma_m^\infty(G) \leq \sum_{i=1}^{l} \gamma_m^\infty(G[S_i])$) is straightforward, as the defender can always defend independently each strongly connected components S_i, using $\gamma^\infty(G[S_i])$ (resp. $\gamma_m^\infty(G[S_i])$) guards. Let us prove the other two inequalities.

We define the following relation $\leq_r$ on the vertices of G: $x \leq_r y$ if there exists a directed path from y to x in G. Since G is finite, $\leq_r$ is a wqo. Let us represent a guards configuration of k guards by the tuple of the k vertices they are placed on, and extend the relation $\leq_r$ on (ordered) guards configurations: $(x_1, \dots, x_k) \leq_r (y_1, \dots, y_k)$ if and only if $x_i \leq_r y_i$ for every $i \in [1, k]$. This relation is a subset of the direct product of k wqos, where a wqo (for well-quasi-ordering) is a preorder such that any infinite sequence of elements $x_0, x_1, x_2, \dots$ contains an increasing pair $x_i \leq_P x_j$ with $i < j$. Thus, it also is a wqo. For both eternal domination and m-eternal domination, if the guards configuration c_2 is reachable from the guards configuration c_1, then $c_1 \geq_r c_2$.

Assume now that the defender plays with less than $\sum_{i=1}^{l} \gamma^\infty(G[S_i])$ (resp. $\sum_{i=1}^{l} \gamma_m^\infty(G[S_i])$) guards. Let D_0 be the initial configuration. There exists a s.c.c. S_i such that $|S_i \cap D_0| < \gamma^\infty(G[S_i])$ (resp. $|S_i \cap D_0| < \gamma_m^\infty(G[S_i])$). So the attacker can apply a strategy in $G[S_i]$ such that they can either win, or the defender moves a guard from outside of S_i into S_i. In the second case, we obtain a configuration D_j such that $D_0 \not\leq_r D_j$, and there exists in D_j a s.c.c. $S_{i'}$ with $|S_{i'} \cap D_j| < \gamma^\infty(G[S_{i'}])$ (resp. $|S_{i'} \cap D_j| < \gamma_m^\infty(G[S_{i'}])$). The attacker can apply the same strategy over again. Since $\leq_r$ is a wqo, any infinite sequence of elements contains an increasing pair, and thus the attacker eventually wins. □

This leads to the value of the two parameters for directed acyclic graphs.

Corollary 7.1. *If G is an acyclic digraph with n vertices, then $\gamma^\infty(G) = \gamma_m^\infty(G) = n$.*

A useful result concerns the monotonicity of γ^∞.

Lemma 7.2. *Let G be a digraph and H be an induced subgraph of G. Then, $\gamma^\infty(H) \leq \gamma^\infty(G)$.*

Proof. Let D_0 be an eternal dominating set of size $\gamma^\infty(G)$ of G, such that the intersection $|D_0 \cap V(H)|$ is maximum with this property. Consider any infinite sequence of attacks in H, and defend it in G, by taking D_0 as the initial guards configuration. At each turn, only one guard is moved, towards the attacked vertex, which belongs to H. Thus, no guard ever leaves H. Moreover, by maximality of $|D_0 \cap V(H)|$, no guard ever arrives in H from outside of H. So any movement of the guards in this defense can be reproduced in H to defend against the sequence of attacks (more precisely, at each turn, the guards configuration in H to defend again the attack is the intersection between $V(H)$ and the guards configuration in G), which gives $\gamma^\infty(H) \leq \gamma^\infty(G)$. □

Note that it is not true for γ_m^∞. Indeed, consider the m-eternal domination of the star S_n of order $n \geq 4$, where each edge is replaced by a double arc. It is easily seen that it can not be defended with only one guard, as attacking two leaves in a row gives a winning strategy for the attacker. But it can be defended with two guards, if the defender always move a guard from a leaf to the center, and the other guard from the center to a leaf (precisely the attacked vertex). So $\gamma_m^\infty(S_n) = 2$. But the set L of leaves of S_n is an independent set, so by Lemma 7.1, $\gamma_m^\infty(S_n[L]) = n-1$, which gives $\gamma_m^\infty(S_n[L]) > \gamma_m^\infty(S_n)$.

The main result of this section is the generalization of Theorem 7.5 to directed graphs.

We define α for digraphs as the order of the greatest induced acyclic subgraph of G. This definition is illustrated in Figure 7.2. Note that this definition of α is, in some sense, a generalization of the independence number to undirected graphs. Indeed, if we replace every edge of a graph G by two arcs, thus creating the digraph $\overleftrightarrow{G}$, we have the equality $\alpha(G) = \alpha(\overleftrightarrow{G})$.

Theorem 7.6. *Given a digraph G, we have*

$$\gamma(G) \leq \gamma_m^\infty(G) \leq \alpha(G).$$

Proof. To see that $\gamma(G) \leq \gamma_m^\infty(G)$, note that in order to defend against the first attack, D_0 must be a dominating set of G.

The proof of the second inequality $\gamma_m^\infty(G) \leq \alpha(G)$ is similar to the one of Goddard et al. [GHH05] for undirected graphs. We consider two cases.

Case 1. Assume that for any $v \in V$, v belongs to an acyclic subgraph of G of order $\alpha(G)$. We prove that for any $A, B \subseteq V$ such that A and B both induce acyclic subgraphs of order $\alpha(G)$ (i.e. maximum induced acyclic subgraphs), there exists a multimove from A to B. This will imply that we can always defend against an attacked vertex r by going from a guards configuration that induces an acyclic subgraph of order $\alpha(G)$ to another, that contains r. Let us construct a bipartite undirected graph G' in the following way. Its parts are the sets A' and B', where A' (resp. B') contains a copy of each of the vertices of A (resp. B). Note that the vertices of $A \cap B$ have a copy in both A' and B'. By abuse of notation, we sometimes refer to a vertex of A' or B' by the vertex of V it is a copy of. Thus, we for example consider that $A' \cap B' = A \cap B$. The edge set of G' is $E' = \{ab : a \in A', b \in B', (a, b) \in E\} \cup \{ab : a \in A', b \in B', a = b\}$. This construction is illustrated in Figure 7.2.

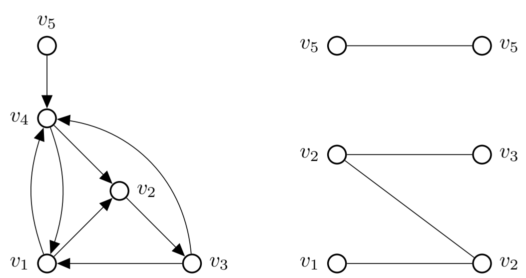


Figure 7.2: On the left, a graph G for which $G[\{v_1, v_2, v_5\}]$ and $G[\{v_2, v_3, v_5\}]$ are maximum induced acyclic subgraphs. Thus, $\alpha(G) = 3$. On the right, the bipartite graph constructed in the proof of Theorem 7.6.

We show that for any set $S \subseteq A'$, $|N(S)| \geq |S|$, where $N(S)$ is the set of neighbors of vertices of S in G'. We will then be able to use Hall's marriage theorem [Hal35] to prove that there exists a perfect matching between A' and B' in G', and therefore a multimove from A to B in G (since every vertex in A is therefore associated to a vertex in B that is either identical, or that is its neighbor in G).

Assume for contradiction that there exists $S \subseteq A'$ such that $|N(S)| < |S|$. Let $X := S \setminus B'$, $Y := N(S) \setminus A'$, and $C := (B' \cup X) \setminus Y$. By construction, $S \cap B' \subseteq N(S) \cap A'$ (the vertices that are in S and in B' are both in A' and B', thus they are neighbors with their copy in B'). So $|S \cap B'| \leq |N(S) \cap A'|$. Thus, since $|S| > |N(S)|$ by assumption, and since $|X| = |S| - |S \cap B'|$ and $|Y| = |N(S)| - |N(S) \cap A'|$, this gives $|X| > |Y|$. Since $X \cap B' = \emptyset$, $C \geq |B'| + |X| - |Y|$ so $|C| > |B'|$, i.e. $|C| > \alpha(G)$.

Let us now prove that $G[C]$ is acyclic. Since $G[A]$ and $G[B]$ are acyclic, it is enough to show that for any $a \in C \cap (A \setminus B), b \in C \cap (B \setminus A)$, we have $(a, b) \notin E$. The only vertices of $A \setminus B$ in C are the ones of $S \setminus B$ (i.e. of X). But the vertices of $C \cap (B \setminus A)$ that are neighbors of vertices of $S \setminus B$ in G' are in $N(S) \setminus A$ (i.e. in Y) and thus are not in C. Thus, $G[C]$ is an induced acyclic subgraph of order greater than $\alpha(G)$, a contradiction. So Hall's condition is verified. Therefore, there exists a perfect matching between A' and B' in G', and a multimove from A to B in G.

Case 2. Assume now that there exists $v \in V$ such that v does not belong to any acyclic subgraph of G of order $\alpha(G)$. We proceed by induction on n.

For $n = 1$, we have $\gamma_m^\infty(G) = \alpha(G) = 1$, so $\gamma_m^\infty(G) \leq \alpha(G)$. Assume now that $n > 1$. We define the graph G' as follows: $G' = (V', E')$ with $V' = V \setminus \{v\}$ and $E' = E(G[V']) \cup \{(u, w) \in V'^2 : u \neq w, (u, v), (v, w) \in E\}$.

By induction hypothesis, we have $\gamma_m^\infty(G') \leq \alpha(G')$.

Let us show that $\alpha(G') \leq \alpha(G) - 1$. Let $S \subseteq V'$ be a set of cardinality $\alpha(G)$. We prove that $G'[S]$ is cyclic. Firstly, if $G[S]$ is cyclic then $G'[S]$ is cyclic too. Now, if $G[S]$ is acyclic, since $|S| = \alpha(G)$, $G[S \cup \{v\}]$ is cyclic. Thus, there exist $u, w \in S$ such that $(u, v), (v, w) \in E$, and (u, v) and (v, w) belong to a cycle of $G[S]$. If $w = u$, since by assumption v does not belong to any of the greatest induced acyclic subgraphs of G, then $G[(S \setminus \{u\}) \cup \{v\}]$ is cyclic and so there exists $u', w' \in S$ such that $u' \neq u$, $w' \neq u$ and $(u', v) \in E$ and (v, w') belong to a cycle in $G[S]$. If $u' = w'$, then (u, u') and (u', u) form a cycle in G'. Otherwise (u', w') belongs to a cycle of G', and similarly, if $w \neq u$, (u, w) belongs to a cycle of G'. In any case, $G'[S]$ is cyclic. So there exists no induced acyclic subgraph of G' of order $\alpha(G)$, and $\alpha(G') \leq \alpha(G) - 1$.

Moreover, $\gamma_m^\infty(G) \leq \gamma_m^\infty(G') + 1$. Indeed, if G' can be defended with k guards, then G can be defended with $k + 1$ guards by adding a guard on v and copying the strategy in G. If the defender of G' moves a guard from u to w using an edge (u, w) that has been added in G' (and, thus, not present in G), then the two edges (u, v) and (v, w) belong to G. Thus the defender of G moves a guard from v to w and another one from u to v.

Combining $\gamma_m^\infty(G') \leq \alpha(G')$ with $\alpha(G') \leq \alpha(G) - 1$ and $\gamma_m^\infty(G) \leq \gamma_m^\infty(G') + 1$ finally gives $\gamma_m^\infty(G) \leq \alpha(G)$. □

To prove the second part of the analogue of Theorem 7.5, we need the following lemma.

Lemma 7.3. *Let G be a digraph. Any set $S \subseteq V$ that induces a maximum acyclic subgraph of G is a dominating set of G.*

Proof. Let S induce a maximum acyclic subgraph of G. Assume for contradiction that there exists $v \in V$ such that $N[v] \cap S = \emptyset$. Then, there is no incoming neighbor of v in S, thus $G[S \cup \{v\}]$ is acyclic. Since $v \notin S$, $|S \cup \{v\}| = |S| + 1$, a contradiction with the maximality of $G[S]$. □

Theorem 7.7. *Given a digraph G, we have*

$$\alpha(G) \leq \gamma^\infty(G) \leq \binom{\alpha(G) + 1}{2}.$$

Proof. The inequality $\alpha(G) \leq \gamma^\infty(G)$ follows from Corollary 7.1 and Lemma 7.2.

The proof of the second inequality $\gamma^\infty(G) \leq \binom{\alpha(G)+1}{2}$ is similar to the proof given for undirected graphs in [KM07], in which α denotes the independence number. Firstly, if $n \leq \binom{\alpha(G)+1}{2}$, since $\gamma^\infty(G) \leq n$, we are done. So we assume $n > \binom{\alpha(G)+1}{2}$. We show that we can defend against any attack $v \in V$ with a strategy that preserves the following invariants for any guards configuration D:

(i) we can partition D into $\alpha(G)$ pairwise disjoint sets $S_{\alpha(G)}, S_{\alpha(G)-1}, \ldots, S_1$ such that:

- $S_{\alpha(G)}$ induces an acyclic subgraph of order $\alpha(G)$
- For any $i < \alpha(G)$, $S_i = \emptyset$ or S_i induces an acyclic subgraph of order i

(ii) $|D|$ is maximum with respect to these constraints.

Note that $\sum_{i=1}^{\alpha(G)} |S_i| \leq \binom{\alpha(G)+1}{2}$, and thus this will imply the result. Moreover, since $|D|$ is maximum, since any vertex set of size 1 induces an acyclic subgraph, and since $n > \binom{\alpha(G)+1}{2}$, we necessarily have $S_1 \neq \emptyset$.

We start with any set D_0 that verifies the invariant. We know that such a set exists since the set $S_{\alpha(G)}$ exists and we can always take $S_i = \emptyset$ for any $i < \alpha(G)$. Let D_t verify the invariant, and let $v \notin D_t$ be the next attack. Let S_j be the smallest set of the partition of D_t such that there exists $u \in S_j$ where $(u, v) \in E$. We know that such a set exists, since $S_{\alpha(G)}$ induces a maximum acyclic subgraph and by Lemma 7.3 it is therefore a dominating set of G. We defend against v by moving to v a guard from the vertex u of S_j. Let us show that $D_{t+1} = (D_t \setminus \{u\}) \cup \{v\}$ verifies the invariant. As $|(D_t \setminus \{u\}) \cup \{v\}| = |D_t|$, we only have to show that it verifies Property (i).

Firstly, if $(S_j \setminus \{u\}) \cup \{v\}$ induces an acyclic subgraph, we just replace S_j by $(S_j \setminus \{u\}) \cup \{v\}$ and obtain a partition of D_{t+1} that satisfies Property (i). Otherwise, v necessarily has at least two distinct incoming neighbors in S_j (including u), and thus $j > 1$. Let S_k be the greatest set of the partition of D_t such that $k < j$. Since $S_1 \neq \emptyset$, we have $S_k \neq \emptyset$. By definition of S_j, since $k < j$, there exists no arc from a vertex of S_k to v and thus $S_k \cup \{v\}$ induces an acyclic subgraph of order $k+1$. Assume for contradiction that $k < j-1$. Then, $S_{k+1} = \emptyset$, and taking $S_{k+1} = S_k \cup \{v\}$ and $S_k = \emptyset$ would create a set of size $|D_t| + 1$ respecting Property (i) and therefore contradict the maximality of $|D_t|$. Thus, $k = j-1$, $S_{j-1} \neq \emptyset$, we can replace in our partition S_j by $S_{j-1} \cup \{v\}$ and S_{j-1} by $S_j \setminus \{u\}$ and still verify Property (i).

□

By Combining Theorems 7.6 and 7.7, we get $\gamma(G) \leq \gamma_m^\infty(G) \leq \alpha(G) \leq \gamma^\infty(G) \leq \binom{\alpha(G)+1}{2}$, which is the analogue of Theorem 7.5 for undirected graphs.

3 Eternal domination on orientations of graphs

We now introduce a new problem that consists in orientating an undirected graph in order to minimize its eternal domination number, or its m-eternal domination number. Recall that an orientation of an undirected graph G is an assignment of exactly one direction to each of the edges of G. This leads to the introduction of three new parameters for undirected graphs:

Given an undirected graph G:

- $\overrightarrow{\gamma^\infty}(G) = \min\{\gamma^\infty(H) : H \text{ is an orientation of } G\}$
- $\overrightarrow{\gamma_m^\infty}(G) = \min\{\gamma_m^\infty(H) : H \text{ is an orientation of } G\}$
- $\overrightarrow{\alpha}(G) = \min\{\alpha(H) : H \text{ is an orientation of } G\}$.

We call $\overrightarrow{\gamma^\infty}(G)$ the *oriented eternal domination number* of G, and $\overrightarrow{\gamma_m^\infty}(G)$ the *oriented m-eternal domination number* of G.

3.1 Oriented eternal domination

For now, we only focus on the oriented eternal domination problem, i.e. only one guard can move every turn. We first show that for non-trivial graphs, $\overrightarrow{\gamma^{\infty}}$ can never be equal to γ, as implied by the following proposition:

Proposition 7.1. *Let G be a graph with at least one edge. Then, $\gamma(G) \leq \alpha(G) < \overrightarrow{\alpha}(G) \leq \overrightarrow{\gamma^{\infty}}(G)$.*

Proof. The inequality $\gamma(G) \leq \alpha(G)$ follows from Theorem 7.5. To see the inequality $\overrightarrow{\alpha}(G) \leq \overrightarrow{\gamma^{\infty}}(G)$, observe that, by Theorem 7.7, the inequality $\alpha(H) \leq \gamma^{\infty}(H)$ is true for any orientation H of G. So we only need to prove that $\alpha(G) < \overrightarrow{\alpha}(G)$. Let H be an orientation of G and S be a maximum independent set of G. Let $S' = S \cup \{v\}$ where v is an arbitrary vertex in $V \setminus S$. The graph $G[S']$ is a union of stars and isolated vertices. Thus, $H[S']$ is acyclic. □

We conjecture a stronger result.

Conjecture 7.1. *Let G be a graph with at least one edge. Then, $\theta(G) < \overrightarrow{\alpha}(G)$.*

This conjecture is true if G is a perfect graph, since $\alpha(G) = \theta(G)$ (as explained in Section **??**), and since $\alpha(G) < \overrightarrow{\alpha}(G)$ by Proposition 7.1.

Note that if true, Conjecture **??** would imply that there is no non-trivial graph G with $\gamma^{\infty}(G) = \overrightarrow{\gamma^{\infty}}(G)$ (since, by Theorem 7.1, $\gamma^{\infty}(G) \leq \theta(G)$).

Hardness results

We give here some hardness results for the computation of $\overrightarrow{\gamma^{\infty}}(G)$.

Let G be an undirected graph. Let $C(G)$ be the graph constructed from G by adding one vertex per edge and connecting each new vertex to the extremities of the associated edge. In $C(G)$, we call $v_1, \ldots v_n$ the vertices of G, and $u_{i,j}$, for any $i < j$, the vertex added from the edge v_iv_j. We call P the set of all added vertices $u_{i,j}$ in $C(G)$. See Figure 7.3 for an example.

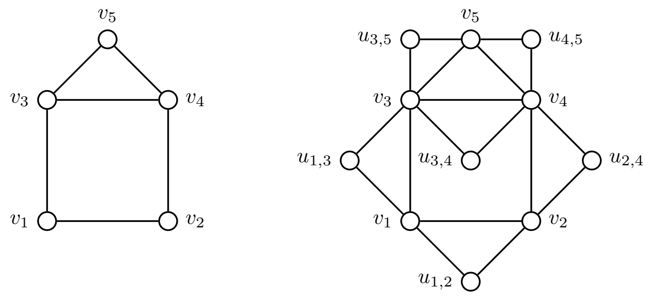


Figure 7.3: The house graph G on the left and $C(G)$ on the right.

Lemma 7.4. *Let G be an undirected graph with m edges. Then, $\overrightarrow{\alpha}(C(G)) = \alpha(G) + m$.*

Proof. Let us first prove that $\alpha(G) \geq \overrightarrow{\alpha}(C(G)) - m$. Let H be an orientation of $C(G)$ such that $\alpha(H) = \overrightarrow{\alpha}(C(G))$. Let us show that we can assume without loss of generality that all triplets $\{u_{i,j}, v_i, v_j\}$ in H induce an oriented triangle. Assume that it is not the case, and let H' be an orientation in which all the triplets $\{u_{i,j}, v_i, v_j\}$ induce an oriented triangle, obtained from H by only changing the orientation of some edges $\{u_{i,j}, v_i\}$ and $\{u_{i,j}, v_j\}$. All the minimal induced cycles of H are also in H'. So for any set of vertices I, if $H[I]$ is acyclic, then $H'[I]$ also is. Thus $\alpha(H') \geq \alpha(H)$. Since $\alpha(H) = \overrightarrow{\alpha}(C(G))$, it gives $\alpha(H') = \alpha(H)$, and we can take H' instead of H. So from now on, we assume that in H, all triplets $\{u_{i,j}, v_i, v_j\}$ induce an oriented triangle. Thus, at most two of the three vertices of each triplet $\{v_i, v_j, u_{i,j}\}$ belong to S. Moreover, if $v_i, v_j \in S$ then $H[(S \setminus \{v_i\}) \cup \{u_{i,j}\}]$ is also acyclic, and if $v_i \notin S$ then $H[S \cup \{u_{i,j}\}]$ is also acyclic. So we assume without loss of generality that $P \subseteq S$. Thus, $|S \cap V| = \overrightarrow{\alpha}(C(G)) - m$. Moreover, by assumption we do not have $v_i, v_j \in S$, and $|S \cap V|$ is an independent set of G. Therefore, $\alpha(G) \geq \overrightarrow{\alpha}(C(G)) - m$.

To see that $\overrightarrow{\alpha}(C(G)) \geq \alpha(G) + m$, consider a maximum independent set S of G. Take $S' = S \cup P$. We have $|S'| = \alpha(G) + m$. Let us show that $C(G)[S']$ is acyclic. There are no cycles in $C(G)[S] = G[S]$, and an added vertex $u_{i,j}$ of P cannot create any cycle since it is only linked to v_i and v_j, and v_i or v_j is not in S. Thus, for any orientation H of $C(G)$, $H[S']$ is acyclic. □

Lemma 7.5. *Let G be an undirected graph with m edges. Then, $\overrightarrow{\gamma^{\infty}}(C(G)) = \gamma^{\infty}(G) + m$.*

Proof. Let us first prove that $\overrightarrow{\gamma^{\infty}}(C(G)) \leq \gamma^{\infty}(G) + m$. We show that there exists an orientation H of $C(G)$ that can be defended by $\gamma^{\infty}(G) + m$ guards. Let us orientate H as follows: every edge $v_iv_j \in E$ is from v_i to v_j where $i < j$, then v_i, v_j, and $u_{i,j}$ induce an oriented triangle. We consider a strategy that preserves the following invariant: we can partition the guards configuration D into a set A of m guards defending each one of the m added vertices $u_{i,j}$ (i.e. are either on $u_{i,j}$ or on v_j) and a set $B \subseteq V$ which is an eternal dominating set of G.

We start with $D_0 = A_0 \cup B_0$ where B_0 is a minimum eternal dominating set of G and $A_0 = P$. Consider a guards configuration $D = A \cup B$ that verifies the invariant and let r be the attack. If $r = u_{i,j}$, then the defender moves the guard from v_j to $u_{i,j}$. If $r = v_i$, then there is a vertex $v_j \in B$ adjacent to r such that $(B \cup \{v_i\}) \setminus \{v_j\}$ is an eternal dominating set of G. We have two possible cases depending on the orientation of v_iv_j. If $v_jv_i \in E(H)$, then the defender moves the guard from v_j to v_i. If $v_iv_j \in E(H)$, then the defender moves the guard from $u_{i,j}$ to v_i. By choosing $B' = (B \cup \{v_i\}) \setminus \{v_j\}$, and $A' = (A \cup \{v_j\}) \setminus \{u_{i,j}\}$, it is easily seen that $D' = A' \cup B'$ satisfies the invariant.

Let us now prove that $\overrightarrow{\gamma^{\infty}}(C(G)) \geq \gamma^{\infty}(G) + m$. We prove that there exists an eternal dominating set X of G such that $|X| = \overrightarrow{\gamma^{\infty}}(C(G)) - m$. Let H be an orientation of $C(G)$ such that $\gamma^{\infty}(H) = \overrightarrow{\gamma^{\infty}}(C(G))$. We call a *clean configuration of* H any eternal dominating set S of H such that $P \subseteq S$. In H, the $\overrightarrow{\gamma^{\infty}}(C(G))$ guards can always be brought to a clean configuration. Indeed, the attacker can successively attack every vertex of P and be sure that they will all be occupied since P is an independent set. We prove that if a set S is a clean configuration of H, then $S \cap V$ is an eternal dominating set of G.

We first prove that $S \cap V$ is a dominating set of G. Firstly, S is a dominating set of H. Any vertex of G dominated by $S \cap V(G)$ in H is still dominated by it in G. Moreover, a vertex $u_{i,j} \in P$ can only dominate v_i and v_j. If $u_{i,j}$ is the only vertex dominating v_i in S, then if v_i is attacked, a guard has to move from $u_{i,j}$ to v_i. So $u_{i,j}$ is dominated by v_j in H, and $v_j \in S$. Since $v_iv_j \in E$, v_j dominates v_i in G, so $S \cap V$ is a dominating set of G.

Now for every attack r in G, let us attack r in H and then $u_{i,j}$ if $u_{i,j}$ became unoccupied. If a guard moves from $v \in V$ to r in H, the guard can do the same in G, and we obtain the clean configuration $S' = (S \setminus \{v\}) \cup \{r\}$. If a guard moves from $u_{i,j}$ to $r = v_i$ in H, then v_j is the only vertex dominating $u_{i,j}$, so when $u_{i,j}$ is attacked, a guard must move from v_j to $u_{i,j}$. We therefore obtain a clean configuration $S' = (S \setminus \{v_j\}) \cup \{v_i\}$, and a guard can directly move from v_j to $r = v_i$ in G to obtain the configuration $S' \cap V$.

Therefore, if we take a clean configuration S of H such that $|S| = \overrightarrow{\gamma^{\infty}}(C(G))$, we have that $S \cap V$ is an eternal dominating set of G, and since $|S \cap V| = \overrightarrow{\gamma^{\infty}}(C(G)) - m$, it gives $\overrightarrow{\gamma^{\infty}}(C(G)) \geq \gamma^{\infty}(G) + m$. □

Note that Lemma 7.5 is not true if we replace $\overrightarrow{\gamma^{\infty}}$ with $\overrightarrow{\gamma^{\infty}_m}$. Indeed, $\overrightarrow{\gamma^{\infty}_m}(C(G)) \leq \gamma^{\infty}_m(G) + m$ remains true but $\overrightarrow{\gamma^{\infty}_m}(C(G)) \geq \gamma^{\infty}_m(G) + m$ is not necessarily true. For example, if we consider the graph P_3, we let the reader verify that $\gamma^{\infty}_m(G) = 2$ and $\overrightarrow{\gamma^{\infty}_m}(C(G)) = 3 < \gamma^{\infty}_m(G) + m$.

The consequences of Lemma 7.4 and Lemma 7.5 are particularly interesting, leading to complexity results. The first consequence is about the coNP-hardness of computing $\overrightarrow{\gamma^{\infty}}(G)$. To our knowledge, there is no known hardness result about the complexity of computing $\gamma^{\infty}(G)$. However, given a graph G and a set S, deciding whether S is an eternal dominating set of G is a Π_2^P-hard problem [Klo07].

We first prove that given a graph G and an integer k, deciding whether $\gamma^\infty(G) \leq k$ is coNP-hard. We use a reformulation of a theorem saying that $\alpha(G)$ is hard to approximate with a polynomial ratio.

Theorem 7.8. *[Zuc06] Let $\epsilon > 0$ and Π be a problem with a graph G and an integer $k > 0$ as the input and such that:*

- *every instance (G, k) with $\alpha(G) < k$ is negative;*
- *every instance (G, k) with $\alpha(G) \geq k^{1/\epsilon}$ is positive.*

Then, Π is NP-hard.

Theorem 7.9. *Given an undirected graph G and an integer k, deciding whether $\gamma^\infty(G) \leq k$ is coNP-hard.*

Proof. We use Theorem 7.8 with $\epsilon = \frac{1}{2}$. We consider the problem Π : given G and k, do we have $\gamma^\infty(G) > \binom{k+1}{2}$?

Clearly, there is a polynomial reduction from the complement of Π to the stated problem. Thus, it suffices to prove that Π satisfies the conditions of Theorem 7.8. If $\alpha(G) < k$, then $\gamma^\infty(G) \leq \binom{\alpha(G)+1}{2} < \binom{k+1}{2}$ (by Theorem 7.5), and thus (G, k) is a negative instance of Π . If $\alpha(G) \geq k^2$, then $\gamma^\infty(G) \geq k^2 > \binom{k+1}{2}$ (by Theorem 7.5). Thus (G, k) is a positive instance of Π . □

From Lemma 7.5 and Theorem 7.9, we obtain:

Corollary 7.2. *Deciding whether $\overrightarrow{\gamma^\infty}(G) \leq k$ is coNP-hard.*

Since deciding whether $\alpha(G) \geq k$ is NP-hard [Kar72], from Lemma 7.4, we also obtain:

Corollary 7.3. *Deciding whether $\overrightarrow{\alpha}(G) \geq k$ is NP-hard.*

Theorem 7.1 states that there can exist an arbitrary gap between α and γ^∞. From this and as a consequence of Lemma 7.4 and Lemma 7.5, we show the same result between $\overrightarrow{\alpha}$ and $\overrightarrow{\gamma^\infty}$.

Corollary 7.4. *For every integer $k > 0$, there exists a graph G such that $\overrightarrow{\gamma^\infty}(G) \geq \overrightarrow{\alpha}(G) + k$.*

Results on some classes of graphs

We now study the value of $\overrightarrow{\gamma^\infty}$ for particular classes of graphs.

Cycles and forests. The first class we consider are cycles and forests, for which the results are straightforward.

Corollary 7.5. *For any $n \geq 3$, we have $\overrightarrow{\gamma^\infty}(C_n) = n - 1$.*

Proof. By Corollary 7.1, for any acyclic orientation H of C_n, we have $\gamma^\infty(H) = n$. Consider now the cyclic orientation H of C_n.

Since $\alpha(H) = n - 1$, by Lemma 7.7 we have $\gamma^\infty(H) \geq n - 1$.

To see that $\gamma^\infty(H) \leq n - 1$, consider the following strategy with $n - 1$ guards: every time a vertex is attacked, the guard on its unique incoming neighbor moves to it. Since only the attacked vertex is unoccupied, we know that the neighbor is occupied so this defense is always possible, and leads to an identical guards configuration. □

Corollary 7.6. *Let G be a graph of order n. Then, $\overrightarrow{\gamma^\infty}(G) = n$ if and only if G is a forest.*

Proof. If G is a forest, then every orientation of G is acyclic and thus $\overrightarrow{\gamma^\infty}(G) = n$ by Corollary 7.1. If G is not a forest, then G admits a cycle C of k vertices. Consider an orientation H of G where the edges of C form an oriented cycle. One can protect C with at most $k - 1$ guards and $G - C$ can be protected by at most $n - k$ guards. Thus, $\gamma^\infty(H) \leq n - 1$. □

Complete graphs. We now consider complete graphs. Surprisingly, the exact value of $\overrightarrow{\gamma^\infty}$ for complete graphs seems hard to find. However, we can obtain lower and upper bounds using a result from Erdös and Moser concerning $\overrightarrow{\alpha}$.

Theorem 7.10. *[EM64] For any $n > 0$, $\lfloor \log_2 n \rfloor + 1 \leq \overrightarrow{\alpha}(K_n) \leq 2\lfloor \log_2 n \rfloor + 1$.*

By combining this theorem with Theorem 7.7, we obtain:

Corollary 7.7. *For any $n > 0$, $\lfloor \log_2 n \rfloor + 1 \leq \overrightarrow{\gamma^\infty}(K_n) \leq \binom{2\lfloor \log_2 n \rfloor + 2}{2}$.*

Complete bipartite graphs. For complete bipartite graphs, finding the exact value of the oriented eternal domination number is however much easier.

Theorem 7.11. *For any $n, m \geq 1$, we have $\overrightarrow{\gamma^\infty}(K_{n,m}) = \max\{n, m\} + 1$.*

Proof. Denote by A and B the two parts of $K_{n,m}$. First, we prove that $\overrightarrow{\gamma^\infty}(K_{n,m}) \geq \max\{n, m\} + 1$. Without loss of generality, we assume that $|A| \geq |B|$. Let $T := G[A \cup \{v\}]$ where v is a vertex of B. It is easily seen that T is a tree. Thus, any orientation of T is acyclic. So $\overrightarrow{\alpha}(K_{n,m}) \geq |A| + 1$ and Proposition 7.1 gives the result.

Let us now prove that $\overrightarrow{\gamma^\infty}(K_{n,m}) \leq \max\{n, m\} + 1$. Without loss of generality, we can assume that $n \geq m$. Let us show that $\overrightarrow{\gamma^\infty}(K_{n,n}) \leq n + 1$. Since $K_{n,m}$ is an induced subgraph of $K_{n,n}$, by Lemma 7.2, it will imply the result. Let M be a perfect matching of $K_{n,n}$. We construct an orientation H of $K_{n,n}$ as follows: let $u \in A$, and $v \in B$. If $\{u, v\} \in M$ then $(u, v) \in E(H)$. Otherwise $(v, u) \in E(H)$. We start by putting a guard on every vertex of A and one guard on an arbitrary vertex of B. In the strategy, we preserve the following invariant: there is at least one guard in every edge of the matching M, and exactly one edge e^* of M has a guard on its two extremities. We denote by v^* the extremity of e^* in B. Suppose that a vertex v of B is attacked. Let u be the vertex such that $uv \in M$. Then, we move the guard on u to v. Suppose now that a vertex u of A is attacked. Then we move the guard on v^* to u. It is easily seen that the invariant is preserved. □

Grids and products of graphs. We now investigate the value of $\overrightarrow{\gamma^\infty}$ on grids. We think that the exact value of $\overrightarrow{\gamma^\infty}$ cannot be expressed by a simple formula. We give here lower and upper bounds.

The following proposition has been verified using a greedy algorithm. We tested all the possible orientations H of $P_3 \square P_3$ (up to isomorphism), and computed the value of the eternal domination number of H each time, by testing all the possible attacks for all the possible guards configurations.

Proposition 7.2. *$\overrightarrow{\gamma^\infty}(P_3 \square P_3) = 7$.*

The unique orientation H with $\gamma^\infty(H) = 7$ (up to isomorphism) is shown in Figure 7.4.

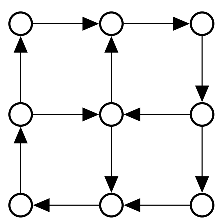


Figure 7.4: An orientation H of $P_3 \square P_3$ with $\gamma^\infty(H) = 7$.

We show that $\frac{2nm}{3} \leq \overrightarrow{\gamma^\infty}(P_n \square P_m) \leq \frac{7nm}{9} + O(n+m)$. The next two theorems give more precise bounds [1].

[1]Note that Theorem 7.12 is not necessarily optimized according to the respective values of n and m, but, since $\overrightarrow{\gamma^\infty}(P_n \square P_m) = \overrightarrow{\gamma^\infty}(P_m \square P_n)$, we can always take the maximum of the two obtained lower bounds.

Theorem 7.12.
$$\left\lceil \frac{n}{2} \right\rceil m + \left\lfloor \frac{n}{2} \right\rfloor \left\lceil \frac{m}{3} \right\rceil \leq \overrightarrow{\gamma^{\infty}}(P_n \Box P_m).$$

Proof. Consider the graph $G = P_n \Box P_m$ with n lines and m columns. We denote by $v_{i,j}$ the vertex at line i and column j. We show that $\alpha(H) \geq \lceil \frac{n}{2} \rceil m + \lfloor \frac{n}{2} \rfloor \lceil \frac{m}{3} \rceil$ for any orientation H of G. This will give $\overrightarrow{\alpha}(P_n \Box P_m) \geq \lceil \frac{n}{2} \rceil m + \lfloor \frac{n}{2} \rfloor \lceil \frac{m}{3} \rceil$, which by Proposition 7.1 gives the result. We construct a set S such that $H[S]$ is acyclic as follows. First, we put odd lines vertices into S. Obviously, $H[S]$ is acyclic and $|S| = \lceil \frac{n}{2} \rceil m$. Now, let us show that one can add $\lceil \frac{m}{3} \rceil$ vertices of each even line and keep $H[S]$ acyclic. Let $i \in [1, n]$ be even. If $i = n$, clearly, we can add to S the vertices $v_{i,j}$ with j odd without creating cycles. Assume that $i < n$. We partition the vertices of the line i into 3 sets: A is the set of vertices $v_{i,j}$ such that $(v_{i-1,j}, v_{i,j})$ and $(v_{i,j}, v_{i+1,j})$ are arcs of H; B is the set of vertices $v_{i,j}$ such that $(v_{i+1,j}, v_{i,j})$ and $(v_{i,j}, v_{i-1,j})$ are arcs of H; and C is the set of vertices $v_{i,j}$ that do not belong to A nor B.

Three cases can occur.
Case 1: $|A| \geq \lceil \frac{m}{3} \rceil$. We add all vertices of A to S and we do not create any cycle.
Case 2: $|B| \geq \lceil \frac{m}{3} \rceil$. Similarly, we add all vertices of B to S and we do not create any cycle.
Case 3: $|A| < \lceil \frac{m}{3} \rceil$ and $|B| < \lceil \frac{m}{3} \rceil$. Without loss of generality, we assume that $|A| \geq |B|$. We construct a set D from $A \cup C$ by picking one vertex out of two in the ordered sequence of vertices of $A \cup C$ (ordered by increasing j). Since $|A \cup C| \geq \lceil \frac{2m}{3} \rceil$, we have $D \geq \lceil \frac{m}{3} \rceil$. We add every vertex of D in S. The only possible way to create cycles with elements of $A \cup C$ is to choose two consecutive vertices $v_{i,j}$ and $v_{i,j+1}$, which is not possible in D by construction. $\square$

Theorem 7.13. *For $m = 3p + 2x$ and $n = 3q + 2y$ with $p, q \in \mathbb{N}$ and $x, y \in \{0, 1, 2\}$, we have:*

$$\overrightarrow{\gamma^{\infty}}(P_n \Box P_m) \leq 7pq + \left\lceil \frac{9p}{2} \right\rceil y + \left\lceil \frac{9q}{2} \right\rceil x + 3xy$$

Proof. We divide the grid into 4 parts of size $3p \times 3q$, $3p \times 2y$, $2x \times 3q$ and $2x \times 2y$ respectively. We already know from Proposition 7.2 that $\overrightarrow{\gamma^{\infty}}(P_3 \Box P_3) = 7$ and from Corollary 7.5 that $\overrightarrow{\gamma^{\infty}}(P_2 \Box P_2) = 3$. Thus, the grid of size $3p \times 3q$ can be protected with $7pq$ guards by dividing it into squares of size 3×3. Similarly, the grid of size $2x \times 2y$ can be protected with $3xy$ guards. The two remaining parts can be covered by squares of size 2×2 then 1×1 and can therefore be protected with $\lceil \frac{9p}{2} \rceil y$ and $\lceil \frac{9q}{2} \rceil x$ guards respectively. $\square$

For grids of size $2 \times n$, $3 \times n$ and $4 \times n$, the lower bound of Theorem 7.12 and the upper bound of Theorem 7.13 coincide and we have the exact value of $\overrightarrow{\gamma^{\infty}}$.

Corollary 7.8. *Let $n \geq 2$. Then,*
$\overrightarrow{\gamma^{\infty}}(P_2 \Box P_n) = \lceil \frac{3n}{2} \rceil$,
$\overrightarrow{\gamma^{\infty}}(P_3 \Box P_n) = \lceil \frac{7n}{3} \rceil$,
$\overrightarrow{\gamma^{\infty}}(P_4 \Box P_n) = 2 \lceil \frac{3n}{2} \rceil$.

We don't know the value of $\overrightarrow{\gamma^{\infty}}(P_5 \Box P_5)$. Using Theorems 7.12 and 7.13, we obtain $19 \leq \overrightarrow{\gamma^{\infty}}(P_5 \Box P_5) \leq 20$.

3.2 Oriented m-Eternal domination

We now switch to oriented m-eternal domination, where any number of guards can move every turn, providing that one of them defends against the attack, and that they are all on different vertices. We study the value of $\overrightarrow{\gamma_m^{\infty}}$ in several classes of graphs.

Cycles and forests. The case of cycles and forests is also straightforward for the m-eternal domination problem.

Corollary 7.9. *For any $n \geq 3$, we have $\overrightarrow{\gamma_m^{\infty}}(C_n) = \lceil \frac{n}{2} \rceil$.*

Proof. We know from Corollary 7.1 that for any acyclic orientation H of C_n, $\gamma_m^\infty(H) = n$. Consider now the cyclic orientation H of C_n.

Assume for contradiction that $\gamma_m^\infty(H) < \lceil \frac{n}{2} \rceil$. Let $D \subseteq V(H)$ be a dominating set of size $\gamma_m^\infty(H)$. Since $\gamma_m^\infty(H) < \lceil \frac{n}{2} \rceil$, there exist $u, v \in V(H) \setminus D$ such that $(u, v) \in E(H)$. The only vertex dominating v in H is u, and $u \notin D$. So D is not a dominating set of H, a contradiction.

To see that $\gamma_m^\infty(H) \leq \lceil \frac{n}{2} \rceil$, consider a strategy with $\lceil \frac{n}{2} \rceil$ guards respecting the following invariant : there is one guard every two vertices on H (with two consecutive vertices if n is odd). When a vertex is attacked, move every guard to the unique outgoing neighbor of their current vertex. Since the attacked vertex is unoccupied, we know that its incoming neighbor is occupied so we defend against the attack, and the new guards configuration verifies the invariant. □

Corollary 7.10. *Let G be a graph of order n. Then, $\overrightarrow{\gamma_m^\infty}(G) = n$ if and only if G is a forest.*

Proof. If G is a forest then by Corollary 7.1, $\overrightarrow{\gamma_m^\infty}(G) = n$. If G is not a forest, G admits a cycle C. Let k be the number of vertices in C and consider an orientation H of G where C is an oriented cycle. We can defend C with $k - 1$ guards and $G - C$ with $n - k$ guards, which gives the result. □

Graphs of oriented m-eternal domination number 2. We now characterize the graphs G such that $\overrightarrow{\gamma_m^\infty}(G) = 2$. To that end, we prove two simple but essential lemmas.

Lemma 7.6. *Let $G = (V, E)$ be a graph of order $n \geq 2$. Then, $\overrightarrow{\gamma_m^\infty}(G) \geq 2$.*

Proof. Let $H = (V, E')$ be an orientation of G, and assume for contradiction that $\gamma_m^\infty(H) = 1$. Let the guard be on the vertex u and the attack on the vertex v. For the defender to defend against v, we must have $(u, v) \in E'$. But then if the next attack is on u, the defender cannot defend since there is no arc (v, u) in E', a contradiction. □

Lemma 7.7. *Let $H = (V, E)$ be the orientation of a graph of order $n \geq 3$, with $\gamma_m^\infty(H) = 2$. Let $u, v \in V$. If neither (u, v) nor (v, u) are arcs of H, then $\{u, v\}$ is a dominating set of H. Moreover, the attacker can force the defender to put their two guards on $\{u, v\}$.*

Proof. Let D be a m-eternal dominating set of H with $|D| = 2$. Consider an attack on u and let D' be the guards configuration that follows. We have $u \in D'$. Now, consider an attack on v. Since $(u, v) \notin E$, the guard on u cannot move to v, so the second guard must do it. If the defender moves the guard on u to a neighbor u^+, then $\{v, u^+\}$ is not a dominating set since neither of u^+ nor v dominates u. So $D' = \{u, v\}$ and $\{u, v\}$ is a dominating set. □

Theorem 7.14. *Let $G = (V, E)$ be a graph of order $n \geq 3$. Then, $\overrightarrow{\gamma_m^\infty}(G) = 2$ if and only if either:*

- *$n = 2k$ and G is a complete graph from which at most k disjoint edges are removed*
- *$n = 2k + 1$ and G is a complete graph from which at most $k - 1$ disjoint edges are removed.*

Proof. ($\Rightarrow$) We first prove that for $n = 2k$ or $n = 2k+1$, if $\overrightarrow{\gamma_m^\infty}(G) = 2$ then G is a complete graph from which a matching, i.e. at most k disjoint edges, are removed. Let H be an orientation of G such that $\gamma_m^\infty(H) = 2$. Assume for contradiction that there exists u,v and w such that $uv, uw \notin E$. By Lemma 7.7, $\{u, v\}$ and $\{u, w\}$ are dominating sets of H. Since $uw \notin E$ and $uv \notin E$, this gives $(v, w) \in E(H)$ and $(w, v) \in E(H)$, a contradiction.

It remains to show that for $n = 2k + 1$, at most $k - 1$ edges are removed. Let G be the complete graph from which exactly k disjoint edges are removed, where $n = 2k + 1$. Let $x_1, \ldots, x_k, y_1, \ldots, y_k, z$ be the vertices of G, such that the non-edges of G are the pairs $x_i y_i$ for $i \in [k]$. We show that $\overrightarrow{\gamma_m^\infty}(G) > 2$. Assume for contradiction that $\overrightarrow{\gamma_m^\infty}(G) = 2$, and let H be an orientation of G such that $\gamma_m^\infty(H) = 2$. Consider an attack on the vertex z. Free to exchange the labels of x_i and y_i, we can assume that the guards configuration that follows is $\{z, x_i\}$. So y_i is dominated by $\{z, x_i\}$ in H, but $x_i y_i \notin E$, thus z dominates y_i in H. Moreover, since by Lemma 7.7, $\{x_i, y_i\}$ is a dominating set of H, x_i dominates z. Now by Lemma 7.7, the attacker can force the guards to go on $\{x_i, y_i\}$. Then, if they attack z again, only the guard on x_i can go to z since z dominates y_i. Moreover, the guard on y_i cannot move to x_i. So the new guards configuration is $\{z, v\}$ where $v \neq x_i$. Free to exchange the labels of x_j and y_j if $j \neq i$, we can assume that $v = y_j$ (note that we might have $v = y_i$). Since $x_j y_j \notin E$, z dominates x_j in H. And since, by Lemma 7.7, $\{x_j, y_j\}$ is a dominating set of H, y_j therefore dominates z. So both x_i and y_j dominate z, with $x_i \neq y_j$. But $\{z, x_i\}$ dominates y_j and $\{z, y_j\}$ dominates x_i,so x_i dominates y_j and y_j dominates x_i, a contradiction.

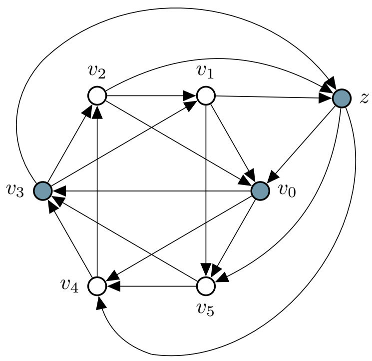


Figure 7.5: The orientation H of the complete graph K_n from which $k-1$ edges are removed, with $n = 2k+1$ and $k = 3$.

($\Leftarrow$) Let us first prove that if $n = 2k+1$ and G is the complete graph of order n from which at most $k-1$ disjoint edges are removed, then $\overrightarrow{\gamma_m^\infty}(G) = 2$. Since adding edges to G cannot make $\overrightarrow{\gamma_m^\infty}(G)$ increase, and since, by Lemma 7.6, $\overrightarrow{\gamma_m^\infty}(G) \geq 2$, we can assume without loss of generality that exactly $k-1$ disjoint edges are removed. So G has exactly 3 universal vertices. Let z be one of them and let $v_0 \dots v_{2k-1}$ be the vertices in $V \setminus \{z\}$, where v_0 and v_k are the two other universal vertices, and $v_i v_{i+k}$ is not an edge of E for every $i \in [k-1]$. We create an orientation H of G as follows. For any i and j such that $i < j$, except for $(i, j) = (0, k)$, we take $(v_j, v_i) \in E(H)$ if and only if $j - i < k$, and $(v_i, v_j) \in E(H)$ if and only if $j - i > k$. Then, for any i, except for $i = k$ and $i = 0$, we take $(v_i, z) \in E(H)$ if and only if $i < k$, and $(v_j, z) \in E(H)$ if and only if $i > k$. Finally, $(v_0, v_k), (v_k, z), (z, v_0) \in E(H)$. The orientation H is represented in Figure 7.5.

We use a strategy that preserves the following invariant: either there exists $i \in [0, k-1]$ such that $D = \{v_i, v_{i+k}\}$, or $D = \{z, v_k\}$. We start with $D_0 = \{v_0, v_k\}$. Let D be a guards configuration that verifies the invariant. There are five possible cases. In each case, we let the reader check that the moves we make are allowed in H.

- if $D = \{v_i, v_{i+k}\}$ and the attack is v_r, if $i < r < i+k$, then we move a guard from v_{i+k} to v_r and the other from v_i to $v_{r'}$ with $r' = r + k \mod 2k$, otherwise we move a guard from v_i to v_r and the other from v_{i+k} to $v_{r'}$.
- if $D = \{v_i, v_{i+k}\}$ with $i \in [1, k-1]$ and the attack is z, we move a guard from v_i to z and the other from v_{i+k} to v_k.
- if $D = \{v_0, v_k\}$ and the attack is z, we move the guard on v_0 to v_k and the guard on v_k to z.
- if $D = \{z, v_k\}$ and the attack is v_r with $r > 0$, if $0 < r < k$, then we move a guard from v_k to v_r and the other from z to $v_{r'}$ with $r' = r + k \mod 2k$. Otherwise we move a guard from z to v_r and the other from v_k to $v_{r'}$.
- if $D = \{z, v_k\}$ and the attack is v_0 we only move the guard on z to v_0.

In each case, we defend against the attack and the new guards configuration respects the invariant, which gives the expected result.

Let us now show that if $n = 2k$ and G is the complete graph of order n from which at most k disjoint edges are removed, then $\overrightarrow{\gamma_m^\infty}(G) = 2$. Without loss of generality, we can assume that exactly k disjoint edges are removed. Observe that in the previously described orientation H, removing z gives an orientation of the complete graph of order $n = 2k$ from which exactly $k-1$ disjoint edges are removed, and the only disjoint edge remaining is v_0v_k. Moreover, in the described defense with 2 guards, if the attacker never attacks z, then the guards configuration never contains z, and the arc (v_0, v_k) is never used by the guards (only the first case occurs). So $H[V \setminus \{z\}]$, to which we remove the arc (v_0, v_k), is an orientation of G that can be defended with 2 guards with this strategy. This concludes the proof.

□

Complete bipartite graphs. The exact value of the parameter is also given for complete bipartite graphs. We prove the following.

Theorem 7.15. *For every $n \geq 2$, and $m \geq 4$, we have $\overrightarrow{\gamma_m^\infty}(K_{n,m}) = 4$. Moreover, $\overrightarrow{\gamma_m^\infty}(K_{2,2}) = 2$ and $\overrightarrow{\gamma_m^\infty}(K_{2,3}) = \overrightarrow{\gamma_m^\infty}(K_{3,3}) = 3$.*

In order to prove it, we split this result into four lemmas.

Lemma 7.8. *We have $\overrightarrow{\gamma_m^\infty}(K_{2,2}) = 2$ and $\overrightarrow{\gamma_m^\infty}(K_{2,3}) = \overrightarrow{\gamma_m^\infty}(K_{3,3}) = 3$.*

Proof. The graph $K_{2,2}$ is isomorphic to C_4 so, by Corollary 7.9, $\overrightarrow{\gamma_m^\infty}(K_{2,2}) = 2$. It is easily seen that $K_{2,3}$ and $K_{3,3}$ don't satisfy the conditions of Theorem 7.14, so $\overrightarrow{\gamma_m^\infty}(K_{2,3}) \geq 3$ and $\overrightarrow{\gamma_m^\infty}(K_{3,3}) \geq 3$. Let us show that $\overrightarrow{\gamma_m^\infty}(K_{2,3}) \leq 3$. Denote by A and B the two parts of $K_{2,3}$, $A = \{a_1, a_2, a_3\}$, $B = \{b_1, b_2\}$. We consider the following defense: we make one guard stay on a_3, and the subgraph induced by the other vertices is isomorphic to $K_{2,2}$, and can thus be defended with the two other guards. Finally, C_6 is a spanning subgraph of $\overrightarrow{\gamma_m^\infty}(K_{3,3})$ so by Corollary 7.9, $\overrightarrow{\gamma_m^\infty}(K_{3,3}) \leq 3$. □

Lemma 7.9. *Let G be a complete bipartite graph of parts A and B, with $|A| \geq 4$ and $|B| \geq 2$. If $\overrightarrow{\gamma_m^\infty}(G) \leq 3$, then the attacker can always force the defender to reach a guards configuration D such that $|D \cap A| = 2$.*

Proof. Let D be a guards configuration. We have four possibilities:

- If $|D \cap A| = 3$, then since $|A| \geq 4$, there exists a vertex in A that is not dominated by D, a contradiction.
- If $|D \cap A| = 2$, then we are done.
- If $|D \cap A| = 1$, let $b_1, b_2 \in B$ and $a_1, a_2, a_3 \in A$. The vertices and arcs we outline in this proof are represented in Figure 7.6.

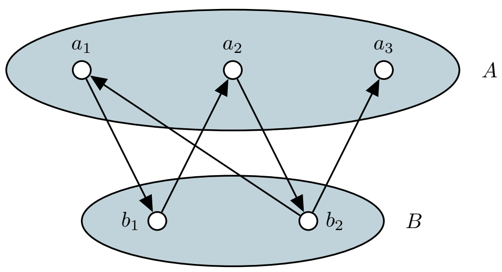


Figure 7.6: The vertices b_1, b_2, a_1, a_2, a_3, and some of their induced arcs in H.

We assume without loss of generality that $D = \{b_1, b_2, a_1\}$. Consider an attack on a_2. Only the guards on b_1 or b_2 can move to a_2. We can assume without loss of generality, that the guard on b_1 does (thus, b_1 dominates a_2). Now the guards on b_2 and a_1 can either stay on their vertex or go to an outgoing neighbor. So we can obtain four guards configurations: $\{a_2, b_2, a_1\}$, $\{a_2, b_2^+, a_1\}$, $\{a_2, b_2, a_1^+\}$ or $\{a_2, b_2^+, a_1^+\}$, where b_2^+ (resp. a_1^+) is an outgoing neighbor of b_2 (resp. a_1). Note that $b_2^+ \in A$ and $a_1^+ \in B$, and they can be identical to previously introduced vertices.

In the first and last configurations, $|D \cap A| = 2$ and we are done. Assume for contradiction that it is not the case. In the second configuration, since $a_2, b_2^+, a_1 \in A$, and $|A| \geq 4$, there exists a vertex in A that is not dominated. So $D = \{a_2, b_2, a_1^+\}$. But since $a_1^+, b_2 \in B$ and b_1 dominates a_2, to dominate b_1 we must have $a_1^+ = b_1$ and $D = \{a_2, b_2, b_1\}$, where a_1 dominates b_1.

Now similarly, if the next attack is a_3, we must reach $\{b_1, b_2, a_3\}$, and either b_1 or b_2 dominates a_3 and is dominated by a_2. Since b_1 dominates a_2, it must be b_2. Then similarly, attacking a_1 ensures that either b_1 or b_2 dominates a_1 and is dominated by a_3. Since b_2 dominates a_3, and a_1 dominates b_1, we obtain a contradiction.

- If $|D \cap A| = 0$, then if the next attack is on a vertex of A, the following guards configuration contains a vertex of A and we obtain one of the previous cases.

□

Lemma 7.10. *For every $n \geq 2$, and $m \geq 4$, we have $\overrightarrow{\gamma_m^\infty}(K_{n,m}) \geq 4$.*

Proof. Assume for contradiction that $\overrightarrow{\gamma_m^\infty}(K_{n,m}) \leq 3$, i.e. there exists an orientation H of $K_{n,m}$ that can be defended by 3 guards. Denote by A and B the two parts of $K_{n,m}$, $|A| = m$, $|B| = n$. Note that a vertex of A (resp. B) cannot dominate another vertex of A (resp. B).

By Lemma 7.9, the attacker can force the defender to reach a guards configuration D where $|D \cap A| = 2$ and $|D \cap B| = 1$. Let $a_1, a_2, a_3, a_4 \in A$ and $b_1 \in B$. All the vertices and the arcs we outline during the proof are represented in Figure 7.7.

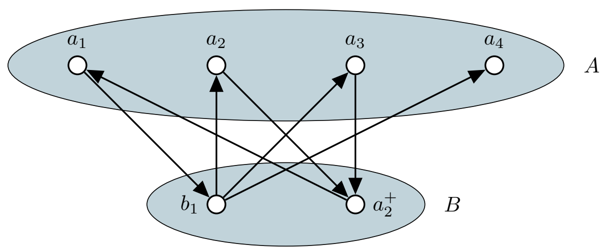


Figure 7.7: The vertices $a_1, a_2, a_3, a_4, b_1, a_2^+$, and some of their induced arcs in H.

Without loss of generality, assume that $D = \{a_1, a_2, b_1\}$. Suppose that the attack is a_3. Then, only the guard on b_1 can move to a_3, so b_1 dominates a_3 in H. The other guards can either stay on a_1 and a_2, or go to one of their outgoing neighbors. We can therefore obtain four different guards configurations: $\{a_1, a_2, a_3\}$, $\{a_1^+, a_2, a_3\}$, $\{a_1, a_2^+, a_3\}$ or $\{a_1^+, a_2^+, a_3\}$, where a_1^+ (resp. a_2^+) is an outgoing neighbor of a_1 (resp. a_2). Note that a_1^+ and a_2^+ belong to B, and they might be identical to b_1.

In the first case, no vertex dominates a_4, in the second, none dominates a_1, and in the third one, none dominates a_2. So we necessarily obtain $\{a_1^+, a_2^+, a_3\}$. But since $a_1^+, a_2^+, b_1 \in B$ and b_1 dominates a_3, to dominate b_1 we must have $a_1^+ = b_1$ or $a_2^+ = b_1$. We consider without loss of generality that $a_1^+ = b_1$. So a_1 dominates b_1, and since $\{b_1, a_2^+, a_3\}$ is a dominating set, where $a_3 \in A$, then a_2^+ must dominate a_1 and b_1 must dominate a_2.

Assume for contradiction that a_2^+ dominates a_3, and attack a_1. Only a_2^+ can go to a_1, so we can obtain four different guards configurations: $\{b_1, a_1, a_3\}$, $\{b_1^+, a_1, a_3\}$, $\{b_1, a_1, a_3^+\}$ or $\{b_1^+, a_1, a_3^+\}$, where b_1^+ (resp. a_3^+) is an outgoing neighbor of b_1 (resp. a_3). In the first and third configurations, no vertex dominates a_2^+ and in the second, none dominates either a_2 or a_4 (if $b_1^+ = a_2$). The only possibility for the last one to dominate a_3 is if $b_1^+ = a_3$, but then a_2^+ is not dominated, a contradiction. So a_3 dominates a_2^+, as represented in Figure 7.7.

Since $D = \{a_1, a_2, b_1\}$ dominates H, b_1 dominates $A \setminus \{a_1, a_2\}$. And since b_1 dominates a_2, only a_1 dominates b_1 in H. So for any guards configuration D', $a_1 \in D'$ or $b_1 \in D'$. Similarly, since $\{b_1, a_2^+, a_3\}$ is a dominating set, a_3 dominates $B \setminus \{b_1, a_2^+\}$. And since a_3 dominates a_2^+, only b_1

dominates a_3 in H. So for any guards configuration D', $b_1 \in D'$ or $a_3 \in D'$. Similarly, since we could have attacked a_4 instead of a_3 in the first place, $b_1 \in D'$ or $a_4 \in D'$. So for any guards configuration D', either $D' = \{a_1, a_3, a_4\}$ or $b_1 \in D'$. But in $\{a_1, a_3, a_4\}$, no vertex dominates a_2. So $b_1 \in D'$. But from our second guards configuration $\{b_1, a_2^+, a_3\}$, if the attack is a_2, only b_1 can move to a_2, and since neither a_3 nor a_2^+ dominates b_1, b_1 cannot be in the next guards configuration, a contradiction. □

Lemma 7.11. *For every $n \geq 2$, and $m \geq 4$, we have $\overrightarrow{\gamma_m^\infty}(K_{n,m}) \leq 4$.*

Proof. Consider the following strategy with 4 guards: partition A into two non-empty sets A_1 and A_2, and B into two non-empty sets B_1 and B_2. Orientate the edges of $K_{n,m}$ from A_1 to B_1, from B_1 to A_2, from A_2 to B_2 and from B_2 to A_1. Start with one guard in each of the four sets. We preserve the following invariant: exactly one guard dominates one set. Every time a vertex is attacked, move the guard who dominates its corresponding set to the attack, and move the three other guards to the set they dominate. □

By combining Lemma 7.8, Lemma 7.10 and Lemma 7.11, we get the result of Theorem 7.15.

Grids and products of graphs. We now investigate the value of $\overrightarrow{\gamma_m^\infty}$ on various kinds of grids.

Theorem 7.16. *For every $n \geq 2$ and $m \geq 2$, we have*

$$\overrightarrow{\gamma_m^\infty}(P_n \square P_m) \leq \left\lceil \frac{nm}{2} \right\rceil .$$

Proof. If n or m are even, $P_n \square P_m$ is Hamiltonian, so $\overrightarrow{\gamma_m^\infty}(P_n \square P_m) \leq \frac{nm}{2}$ by Corollary 7.9. Otherwise, $P_n \square P_m$ is Hamiltonian if we remove a corner vertex. The corresponding Hamiltonian cycles are illustrated in Figure 7.8. So if we keep a guard on the corner vertex and defend the remaining vertices with $\frac{nm-1}{2}$ vertices, we obtain the desired bound. □

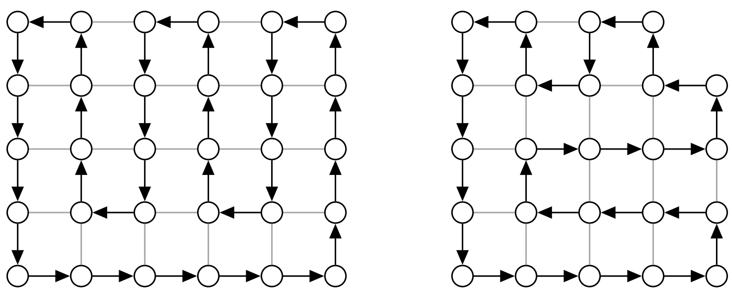


Figure 7.8: On the left, the Hamiltonian cycle of $P_n \square P_m$ when n is even. On the right, the Hamiltonian cycle of $P_n \square P_m$ to which we remove a corner, when both n and m are odd.

We do not have lower bounds except for the straightforward bound $\left\lceil \frac{nm}{4} \right\rceil$ (obtained by observing that any vertex either has indegree 0 and a guard must always be on it, or dominates at most 4 vertices). On the other hand, the upper bound seems loose but we have verified, using a computer, that $\overrightarrow{\gamma_m^\infty}(P_n \square P_m) = \left\lceil \frac{nm}{2} \right\rceil$ for every n and m between 2 and 5. No counter example has been found for other values. We lack tools to find tight lower bounds.

We now consider upper bounds on $\overrightarrow{\gamma_m^\infty}$ for toroidal grids, rook's graphs, toroidal king's grids and toroidal hypergrids. We present a general method based on the neighborhood-equitable coloring (NE-coloring), a notion we introduce. Let k and ℓ be two integers and G be a $\ell \cdot (k-1)$-regular graph. A (k, ℓ)-*NE coloring* of G is a proper coloring $(V_1, \ldots, V_k)$ of G with k colors such that for every vertex v and color $i \in [1, k]$ such that $v \notin V_i$, we have $|N(v) \cap V_i| = \ell$, i.e. every vertex has exactly ℓ neighbors of each color but its own.

Lemma 7.12. *Let $G = (V, E)$ be a graph that admits a $(k, 2\ell)$-NE coloring. Then $\overrightarrow{\gamma_m^\infty}(G) \leq \frac{n}{k}$.*

Proof. Consider a $(k, 2\ell)$-NE coloring $(V_1, \dots V_k)$ of G. Let G_{ij} be the subgraph of G induced by $V_i \cup V_j$. By construction, G_{ij} is a 2ℓ-regular bipartite graph. We orientate G_{ij} such that the indegree and outdegree of every vertex is ℓ. Indeed, by Euler's condition [Eul41], each component G_{ij} is Eulerian. So we can orientate each component to obtain Eulerian orientations (we orientate each edge with the direction used in the Eulerian walk). We do this for every distinct i and j in $[1, k]$ and obtain an orientation H of G. Let us prove that $\gamma_m^\infty(H) \leq \frac{n}{k}$. We initially put the guards on all the vertices of an arbitrary color class V_i. If a vertex $v \in V_j$ is attacked, we move all guards from V_i to V_j. Indeed, consider the graph B_{ij} with vertices $V_i \cup V_j$ and where we put an edge between two vertices $u \in V_i$ and $v \in V_j$ if and only if $(u, v) \in E(H)$. The graph B_{ij} is l-regular by construction. Thus, by application of Hall's marriage theorem [Hal35], B_{ij} admits a perfect matching. So there is a multimove from V_i to V_j in H. And for any i, we have $|V_i| = \frac{n}{k}$. Indeed, the number of edges between a vertex of color i and a vertex of color $j \neq i$ is $|V_i| \cdot 2\ell$, or, equivalently, $|V_j| \cdot 2\ell$. Thus, $|V_i| = |V_j|$ for any pair of colors i and j, which concludes the proof. □

Products of graphs admit the following nice property.

Lemma 7.13. *Let G_1 be a graph that admits a (k, ℓ_1)-NE coloring and G_2 be a graph that admits a (k, ℓ_2)-NE coloring. Then, $G_1 \Box G_2$ admits a $(k, \ell_1 + \ell_2)$-NE coloring.*

Proof. We assume that the vertices of G_1 and G_2 are colored with integers chosen in the set $[1, k-1]$. Let $v_1, \dots, v_n$ be the vertices of G_1, and $u_1, \dots, u_m$ be the vertices of G_2. For a vertex v_i of G_1 and a vertex u_j of G_2, we denote by $w_{i,j}$ the vertex associated to (v_i, u_j) in $G_1 \Box G_2$. Let p be the color of v_i and q be the color of u_j. Then, we assign to $w_{i,j}$ the color $r = p + q \mod k$. Let r' be a color different from r. Then $w_{i,j}$ has exactly ℓ_1 (resp. ℓ_2) neighbors $w_{i',j'}$ of color r' with $i = i'$ (resp. $j = j'$) and thus $\ell_1 + \ell_2$ neighbors of color r'. □

This notion of NE-coloring has direct consequences on toroidal grids and rook's graphs.

Theorem 7.17. *When m and n are both multiples of 3, we have:*

$$\overrightarrow{\gamma_m^\infty}(C_n \Box C_m) \leq \frac{nm}{3}.$$

In general, we have:

$$\overrightarrow{\gamma_m^\infty}(C_n \Box C_m) \leq \left\lceil \frac{nm}{3} \right\rceil + O(n+m).$$

Proof. A cycle of order n, where n is a multiple of 3, admits a $(3, 1)$-NE coloring (we alternate between the three colors). So the first inequality follows from Lemma 7.12 and Lemma 7.13.

If n or m is not a multiple of 3, consider the grid $C_{3n+x} \Box C_{3m+y}$ with $x, y \in \{0, 1, 2\}$ and $x > 0$ or $y > 0$. Let H' be an orientation of $C_{3n} \Box C_{3m}$ as described in the previous case. We construct an orientation H of $C_{3n+x} \Box C_{3m+y}$ as follows. We orientate each edge between the vertices of coordinates (i, j) and $(i, j+1)$ for $i \in [1, 3n]$, $j \in [1, 3m-1]$, and between the ones of coordinates (i, j) and $(i+1, j)$ for $i \in [1, 3n-1]$ and $j \in [1, 3m]$, in the same direction as in H'. For every $i \in [1, 3n]$, if $(v_{i,3m}, v_{i,1}) \in E(H')$, then we orientate H such that $(v_{i,3m}, v_{i,3m+1}, \dots, v_{i,3m+y}, v_{i,1})$ is an oriented path. Otherwise, we orientate H such that $(v_{i,1}, v_{i,3m+y}, v_{i,3m+y-1}, \dots, v_{i,3m})$ is an oriented path. We do the same for every edge $v_{3n,j}v_{1,j}$ with $j \in [1, 3m]$. We arbitrarily orientate the remaining edges. An example of an orientation is given in Figure 7.9.

Consider the set of vertices S of H including an m-eternal dominating set of H', as described in the previous case, and containing every vertex (i, j) with $i > 3n$ or $j > 3m$. Then, S is an m-eternal dominating set of H. Indeed, we mimic the strategy of the defender for H'. The only difference is when a guard in H' goes from a "border" of the grid to the opposite border. For example, if a guard goes from a vertex $(i, 3m)$ to the vertex $(i, 1)$. Then, we push every guard, except for the last one, in the path $(v_{i,3m}, v_{i,3m+1}, \dots, v_{i,3m+y}, v_{i,1})$ to the next vertex. One can easily generalize to the other borders. □

For square rook's graphs, we obtain the exact value of $\overrightarrow{\gamma_m^\infty}$.

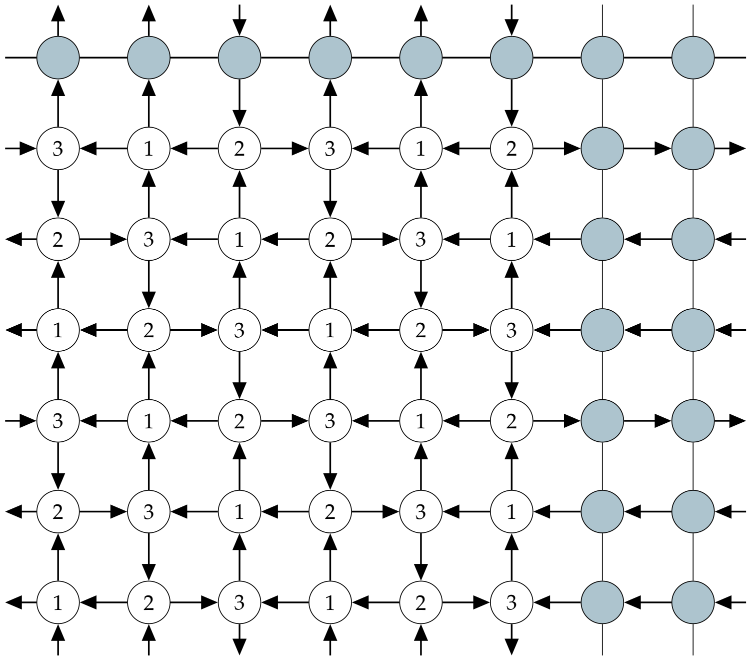


Figure 7.9: orientation of the toroidal grid $C_7 \Box C_8$.

Theorem 7.18. *For every $n \geq 1$, we have $\overrightarrow{\gamma_m^\infty}(K_n \Box K_n) = \gamma(K_n \Box K_n) = n$.*

Proof. It easily seen that $\gamma(K_n \Box K_n) \geq n$. Indeed, a set of size lower than n does not dominate at least a line and a column. Thus, it does not dominate the vertex which is at the intersection of this line and this column. The upper bound is a direct consequence of Lemma 7.12 and Lemma 7.13 with the fact that a complete graph of order n admits a $(n, 1)$-NE coloring. □

Toroidal king's grids are the strong product of two cycles. For this class of graphs, we obtain the following result.

Theorem 7.19. *Let m and n be two multiples of 5. Then, we have:*

$$\overrightarrow{\gamma_m^\infty}(C_n \boxtimes C_m) \leq \frac{nm}{5}.$$

Proof. If we split the grid in squares of size 5×5, each square can be colored as in Figure 7.10. We let the reader check that we obtain a $(5, 2)$-NE coloring. So Lemma 7.12 gives the result.

□

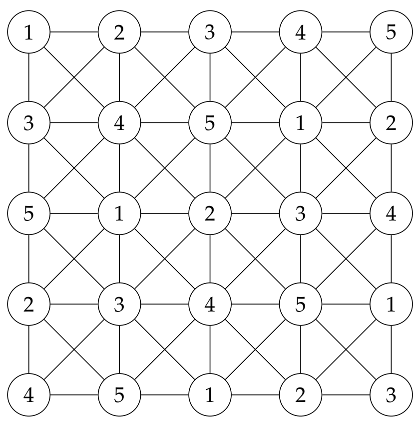


Figure 7.10: $(5, 2)$-NE coloring of a square of a king's grid.

We can obtain an upper bound $\frac{nm}{5} + O(n+m)$ when there is no condition on n and m. The idea is similar to the proof of Theorem 7.17. However, the proof is quite complicated and the result not essential so we omit it.

We also generalize the first statement of Theorem 7.17 to toroidal hypergrids.

Theorem 7.20. *If for any i, n_i is a multiple of $k+1$, then $\overrightarrow{\gamma_m^\infty}(C_{n_1}\square \dots \square C_{n_k}) \leq \frac{n}{k+1}$ where n is the order of the graph.*

Proof. Let v be a vertex at position $(i_1, \dots, i_k)$ in the hypergrid. We affect to v the color

$$(\sum_{j=1}^{k} j \cdot i_j) \mod (k+1).$$

It is easily seen that this coloring is proper. Moreover, for every distinct colors i, j, and vertex v of color i, v has exactly two neighbors of color j. Indeed, if v is at position $(i_1, \dots, i_k)$, then the two neighbors are at positions $(i_1, \dots, i_{p-1}, i_p + 1, i_{p+1}, \dots, i_k)$ where $p = (j-i) \mod (k+1)$ and $(i_1, \dots, i_{q-1}, i_q - 1, i_{q+1}, \dots, i_k)$ where $q = (i-j) \mod (k+1)$. Thus, we obtain a $(k+1, 2)$-NE coloring. So Lemma 7.12 gives the result. ☐

We conjecture that the upper bounds in Theorems 7.19, 7.20, and 7.17 correspond to the exact value of $\overrightarrow{\gamma_m^\infty}$. A way to prove this would be to show that any orientation that minimizes $\overrightarrow{\gamma_m^\infty}$ is Eulerian. More generally, we think that the following is true.

Conjecture 7.2. *Let G be a graph that admits a $(k,2)$-NE coloring. Then $\overrightarrow{\gamma_m^\infty}(G) = \frac{n}{k}$.*

This conjecture is verified for even cycles (Theorem 7.9) and for square rook's grids (Theorem 7.18).

4 Conclusion

In this chapter, we presented the eternal domination framework. We introduced a version on digraphs, in which the guards must follow the direction of the arcs. We gave several bounds on the eternal and m-eternal domination number of digraphs. These results are analogues of results already obtained for undirected graphs. We then introduced a new problem that consists in finding an orientation of an undirected graph which minimizes the eternal, or m-eternal domination number. These minimum values allowed to define new graph parameters, $\overrightarrow{\gamma^\infty}$ and $\overrightarrow{\gamma_m^\infty}$. We first got interested in the complexity of determining if $\overrightarrow{\gamma^\infty}(G) \leq k$ for a given k. We proved that it is a co-NP-hard problem, using a simple reduction from the unoriented version of the problem. On the other hand, the complexity of deciding whether $\overrightarrow{\gamma_m^\infty}(G) \leq k$ for a given k remains open. We then studied the value of the parameters $\overrightarrow{\gamma^\infty}$ and $\overrightarrow{\gamma_m^\infty}$ in various graph classes. We obtained the exact value of $\overrightarrow{\gamma^\infty}$ in trees, cycles, and complete bipartite graphs. Despite our efforts, we did not find its value in complete graphs. We gave an upper and a lower bound but the gap between them is quite large, and even finding better bounds would be a significant improvement. We also gave several bounds for different kinds of grids, and we would like to improve these bounds. In particular, we proved that $\overrightarrow{\gamma^\infty} = \overrightarrow{\alpha}$ for the $2\times n$, $3\times n$, and $4\times n$ grids, so one might wonder if it is always the case in rectangular grids. It is also true in other graphs such as trees, cycles and complete bipartite graphs, and one might wonder if it is the case in other graph classes, such as complete graphs. Similarly, we obtained the exact value of $\overrightarrow{\gamma_m^\infty}$ in trees, cycles, complete graphs, complete bipartite graphs, and we obtained upper and lower bounds in different kinds of grid, using in particular the notion of neighborhood-equitable coloring. To improve these bounds, a lead would consist in proving that for any graph G that admits a $(k,2)$-NE coloring, we have $\overrightarrow{\gamma_m^\infty}(G) = \frac{n}{k}$, as we conjectured. We also gave a complete characterization of the graphs for which the oriented m-eternal domination number is 2. In particular, the non-complete graphs with $\overrightarrow{\gamma_m^\infty} = 2$ verify $\overrightarrow{\gamma_m^\infty} = \gamma$. It would be interesting to characterize the graphs that verify this property. Other examples are rook's graphs.

More generally, a lot of problems remain open concerning eternal domination. For instance, it would be interesting to fully characterize the graphs for which $\gamma^{\infty}(G) = \alpha(G)$ or $\gamma^{\infty}(G) = \theta(G)$. The Vizing-like conjecture $\gamma^{\infty}(G \Box H) \geq \gamma^{\infty}(G) \cdot \gamma^{\infty}(H)$ also remains open. As for m-eternal domination, characterizing the graphs for which $\gamma_m^{\infty}(G) = \gamma(G)$, $\gamma_m^{\infty}(G) = \gamma^{\infty}(G)$, or $\gamma_m^{\infty}(G) = \alpha(G)$ could be a lead for future work. Finally, other generalizations of the problems can be considered. For instance, one could allow a given number k of guards to move at each turn, and a given number k' of guards to be on the same vertex at each turn. The reconfiguration of eternal dominating sets could also be studied from the perspective of classical reconfiguration problems. For example, one could study the connectivity problems for the reconfiguration of eternal dominating sets, under the token sliding rule or even the token addition-removal rule.

Chapter 8

Conclusion

In this thesis, we focused mainly on graph domination, and reconfiguration problems.

The first reconfiguration problem we focused on concerned the reconfiguration of connected multigraphs with the same degree sequence. We obtained a polynomial time 2.5-approximation algorithm for the shortest transformation problem. The best algorithm known so far was a 4-approximation, and the improvement is due the upper bound we found. On the other hand, we also showed that under several reasonable hypothesis (in particular, if we only perform flips on the elements of the symmetric difference), we cannot drastically improve the approximation ratio unless we change the lower bound.

We then moved on to the reconfiguration of dominating sets. We first focused on the token addition-removal rule, and the connectivity problem. More precisely, we studied the maximum size k of the dominating sets allowed in the reconfiguration graph. We investigated the minimum value above which the reconfiguration graph is always connected. We first proved that the reconfiguration graph has a linear diameter when $k \geq \Gamma(G) + \alpha(G) - 1$ (it was already known that the reconfiguration graph is connected in this case). We then proved that if G is K_ℓ-minor free, then there exists a constant C such that the reconfiguration graph is connected and has linear diameter for every $k \geq \Gamma(G) + C\ell\sqrt{\log_2 \ell}$. We also showed that it is connected and has linear diameter for any $k \geq \Gamma(G) + 3$ when G is a planar graph. Finally, we showed that for any graph G, if $k \geq \Gamma(G) + tw(G) + 1$, then the reconfiguration graph is connected and has a linear diameter.

Then, we investigated the reconfiguration of dominating sets under the token sliding rule, and studied the complexity of the reachability problem in several graph classes. We showed that the problem is PSPACE-complete in planar bipartite graphs, unit disk graphs, circle graphs and line graphs. On the other hand, we gave a polynomial time algorithm for circular arc graphs, by using the connectivity of the reconfiguration graph for interval graphs.

Finally, we studied the eternal domination problem. We introduced a version on digraphs, where the guards follow the direction of the arcs. We obtained several bounds on the eternal and m-eternal domination number of digraphs, that generalized results on undirected graphs. We then introduced the oriented and m-oriented eternal domination problems, which consist in finding an orientation of an undirected graph that minimizes its eternal or m-eternal domination number. These numbers then define the oriented and m-oriented eternal domination number of the graph. We studied the complexity of determining if the oriented eternal domination number is at most a given k and proved that it is a co-NP-hard problem. We then studied the value of the parameters in various graph classes such as trees, cycles, complete graphs, complete bipartite graphs and different kinds of grids. We also gave a complete characterization of the graphs for which the oriented m-eternal domination number is 2.

Along these chapters, a lot of questions remained open.

Concerning the reconfiguration of connected multigraphs with the same degree sequence, we spent some time trying to find a better lower bound on the shortest transformation but did not succeed. When discussing the lower bound, we also made the hypothesis that there always exists a shortest sequence that only flips edges of the symmetric difference, but we have no proof of this statement, although the best algorithms always flip such edges. Both these questions leave room for future improvements.

On the reconfiguration of dominating sets under the token addition-removal rule, there are several gaps that we would like to close. In particular, we conjectured that for planar graphs, the reconfiguration graph is connected as long as $k \geq \Gamma(G)+2$, and only the value $k = \Gamma(G)+2$ is missing. Similarly, we would like to find out if the reconfiguration graph is connected when $k = \Gamma(G) + tw(G)$ and $k = \Gamma(G) + tw(G) - 1$.

Under the token sliding rule, it remains to find out if there exists a graph class for which computing a minimum dominating set is NP-complete but the reachability problem is polynomial, or for which computing a minimum dominating set is in P but the reachability problem is PSPACE-complete. Any results on the reachability problem in specific graph classes would be an improvement, and we suggest outerplanar graphs, for instance.

Finally, mainy questions remain open on the oriented and m-eternal oriented eternal domination problem. In particular, we would like to know the complexity of deciding whether the oriented m-eternal domination number is at most a given k. We would also like to find the value of the oriented eternal domination number in complete graphs.

More generally, the reconfiguration framework is very rich and interesting, and after my thesis, I intend to work on a larger range of source problems, such as graph colorings, graph matchings, or the token swapping problem.

Bibliography

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[ABB+07] Mark Anderson, Christian Barrientos, Robert C Brigham, Julie R Carrington, Richard P Vitray, and Jay Yellen. Maximum-demand graphs for eternal security. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 61:111, 2007.

[ABF+02] Jochen Alber, Hans L Bodlaender, Henning Fernau, Ton Kloks, and Rolf Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.

[ACP87] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in ak-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[AF95] John Arquilla and Hal Fredricksen. " graphing" an optimal grand strategy. *Military Operations Research*, pages 3–17, 1995.

[AFK17] Saeid Alikhani, Davood Fatehi, and Sandi Klavžar. On the structure of dominating graphs. *Graphs and Combinatorics*, 33(4):665–672, 2017.

[AFM17] Saeid Alikhani, Davood Fatehi, and Kieka Mynhardt. On k-total dominating graphs. *arXiv preprint arXiv:1711.04363*, 2017.

[AH76] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. *Bulletin of the American mathematical Society*, 82(5):711–712, 1976.

[Ale82] V. E. Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. *Combinatorial-Algebraic Methods in Applied Mathematics*, pages 3–13, 1982. in Russian.

[AMPP99] Vincenzo Auletta, Angelo Monti, Mimmo Parente, and Pino Persiano. A linear-time algorithm for the feasibility of pebble motion on trees. *Algorithmica*, 23(3):223–245, 1999.

[B+19] Andrei de Almeida Sampaio Braga et al. An eternal domination problem: graph classes, solving methods, and practical standpoint. 2019.

[BB17] Marthe Bonamy and Nicolas Bousquet. Token sliding on chordal graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 127–139. Springer, 2017.

[BB18] Marthe Bonamy and Nicolas Bousquet. Recoloring graphs via tree decompositions. *European Journal of Combinatorics*, 69:200–213, 2018.

[BB19] Nicolas Bousquet and Valentin Bartier. Linear transformations between colorings in chordal graphs. *arXiv preprint arXiv:1907.01863*, 2019.

[BBFJ19] Marthe Bonamy, Nicolas Bousquet, Carl Feghali, and Matthew Johnson. On a conjecture of mohar concerning kempe equivalence of regular graphs. *Journal of Combinatorial Theory, Series B*, 135:179–199, 2019.

[BBH+19] Marthe Bonamy, Nicolas Bousquet, Marc Heinrich, Takehiro Ito, Yusuke Kobayashi, Arnaud Mary, Moritz Mühlenthaler, and Kunihiro Wasa. The perfect matching reconfiguration problem. *arXiv preprint arXiv:1904.06184*, 2019.

[BC09] Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: Pspace-completeness and superpolynomial distances. *Theoretical Computer Science*, 410(50):5215–5226, 2009.

[BCG+04] AP Burger, EJ Cockayne, WR Grundlingh, CM Mynhardt, JH Van Vuuren, and W Winterbach. Infinite order domination in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 50:179–194, 2004.

[BCH+00] Xu Baogen, Ernest J Cockayne, Teresa W Haynes, Stephen T Hedetniemi, and Zhou Shangchao. Extremal graphs for inequalities involving domination parameters. *Discrete Mathematics*, 216(1-3):1–10, 2000.

[BDE+17] STEFAN Bard, CHRIS Duffy, MICHELLE Edwards, GARY Macgillivray, and FEIRAN Yang. Eternal domination in split graphs. *J. Comb. Math. Comb. Comput*, 101:121–130, 2017.

[BDO19] Marthe Bonamy, Paul Dorbec, and Paul Ouvrard. Dominating sets reconfiguration under token sliding. *arXiv preprint arXiv:1912.03127*, 2019.

[BdSL15] Andrei Braga, Cid C de Souza, and Orlando Lee. The eternal dominating set problem for proper interval graphs. *Information Processing Letters*, 115(6-8):582–587, 2015.

[Ber01] Claude Berge. *The theory of graphs*. Courier Corporation, 2001.

[BFM13] I Beaton, S Finbow, and JA MacDonald. Eternal domination numbers of 4× n grid graphs. *J. Combin. Math. Combin. Comput*, 85:33–48, 2013.

[BH19] Nicolas Bousquet and Marc Heinrich. A polynomial version of cereceda's conjecture. *arXiv preprint arXiv:1903.05619*, 2019.

[BHK02] Piotr Berman, Sridhar Hannenhalli, and Marek Karpinski. 1.375-Approximation Algorithm for Sorting by Reversals. In Rolf Möhring and Rajeev Raman, editors, *Algorithms — ESA 2002*, page 200–210, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[BI17] Sergey Bereg and Hiro Ito. Transforming graphs with the same graphic sequence. *Journal of Information Processing*, 25:627–633, 2017.

[BJL+14] Marthe Bonamy, Matthew Johnson, Ioannis Lignos, Viresh Patel, and Daniël Paulusma. Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs. *Journal of Combinatorial Optimization*, 27(1):132–143, 2014.

[BK98] Heinz Breu and David G Kirkpatrick. Unit disk graph recognition is np-hard. *Computational Geometry*, 9(1-2):3–24, 1998.

[BKL+20] Rémy Belmonte, Eun Jung Kim, Michael Lampis, Valia Mitsou, Yota Otachi, and Florian Sikora. Token sliding on split graphs. *Theory of Computing Systems*, pages 1–25, 2020.

[BKV19] Václav Blažej, Jan Matyáš Křišt'an, and Tomáš Valla. On the m-eternal domination number of cactus graphs. In *International Conference on Reachability Problems*, pages 33–47. Springer, 2019.

[BL76] Kellogg S Booth and George S Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of computer and system sciences*, 13(3):335–379, 1976.

[BM18a] Nicolas Bousquet and Arnaud Mary. Reconfiguration of graphs with connectivity constraints. In *International Workshop on Approximation and Online Algorithms*, pages 295–309. Springer, 2018.

[BM18b] Nicolas Bousquet and Arnaud Mary. Reconfiguration of graphs with connectivity constraints. In *Approximation and Online Algorithms - WAOA 2018*, pages 295–309, 2018.

[BMFN99] Adrian Brüngger, Ambros Marzetta, Komei Fukuda, and Jürg Nievergelt. The parallel search bench zram and its applications. *Annals of Operations Research*, 90:45–63, 1999.

[BMOS19] Alexandre Blanché, Haruka Mizuta, Paul Ouvrard, and Akira Suzuki. Decremental optimization of dominating sets under reachability constraints. *arXiv preprint arXiv:1906.05163*, 2019.

[Bol13] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[BP16] Nicolas Bousquet and Guillem Perarnau. Fast recoloring of sparse graphs. *European Journal of Combinatorics*, 52:1–11, 2016.

[Cap97a] Alberto Caprara. Sorting by reversals is difficult. In *Proceedings of the first annual international conference on Computational molecular biology*, pages 75–83, 1997.

[Cap97b] Alberto Caprara. Sorting by reversals is difficult. In *Proceedings of the First Annual International Conference on Computational Molecular Biology*, RECOMB '97, pages 75–83. ACM, 1997.

[CCJ91] Brent N Clark, Charles J Colbourn, and David S Johnson. Unit disk graphs. In *Annals of Discrete Mathematics*, volume 48, pages 165–177. Elsevier, 1991.

[Cer07] Luis Cereceda. *Mixing graph colourings*. PhD thesis, The London School of Economics and Political Science (LSE), 2007.

[Cha92] Tony Yu Chang. *Domination numbers of grid graphs*. PhD thesis, University of South Florida, 1992.

[CHHH11] Elizabeth Connelly, Kevin R Hutson, Stephen T Hedetniemi, and TW Haynes. A note on γ-graphs. *AKCE International Journal of Graphs and Combinatorics*, 8(1):23–31, 2011.

[Chr98] David A. Christie. A 3/2-approximation algorithm for sorting by reversals. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '98, pages 244–252, 1998.

[CKP06] E Chambers, W Kinnersly, and Noah Prince. Mobile eternal security in graphs. *Manuscript*, 2006.

[CPS85] Derek G. Corneil, Yehoshua Perl, and Lorna K Stewart. A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14(4):926–934, 1985.

[CRST06] Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.

[Cul97] Joseph Culberson. Sokoban is pspace-complete. 1997.

[CVDHJ08] Luis Cereceda, Jan Van Den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Mathematics*, 308(5-6):913–919, 2008.

[CVdHJ09] Luis Cereceda, Jan Van den Heuvel, and Matthew Johnson. Mixing 3-colourings in bipartite graphs. *European Journal of Combinatorics*, 30(7):1593–1606, 2009.

[CVDHJ11] Luis Cereceda, Jan Van Den Heuvel, and Matthew Johnson. Finding paths between 3-colorings. *Journal of graph theory*, 67(1):69–82, 2011.

[DDFE$^+$15] Erik D Demaine, Martin L Demaine, Eli Fox-Epstein, Duc A Hoang, Takehiro Ito, Hirotaka Ono, Yota Otachi, Ryuhei Uehara, and Takeshi Yamada. Linear-time algorithm for sliding tokens on trees. *Theoretical Computer Science*, 600:132–142, 2015.

[Deb10] Lokenath Debnath. A brief historical introduction to euler's formula for polyhedra, topology, graph theory and networks. *International Journal of Mathematical Education in Science and Technology*, 41(6):769–785, 2010.

[Dew81] Alexander Keewatin Dewdney. *Fast Turing Reductions Between Problems in NP: Chapter 4: Reductions Between NP-complete Problems*. Department of Computer Science, University of Western Ontario, 1981.

[DFFV06] Martin Dyer, Abraham D Flaxman, Alan M Frieze, and Eric Vigoda. Randomly coloring sparse random graphs with fewer colors than the maximum degree. *Random Structures & Algorithms*, 29(4):450–465, 2006.

[DGKTB10] Charo I Del Genio, Hyunju Kim, Zoltán Toroczkai, and Kevin E Bassler. Efficient and exact sampling of simple graphs with given arbitrary degree sequence. *PloS one*, 5(4):e10012, 2010.

[DKK$^+$20] Keith Driscoll, William F Klostermeyer, Elliot Krop, Colton Magnant, and Patrick Taylor. On eternal domination and vizing-type inequalities. *AKCE International Journal of Graphs and Combinatorics*, pages 1–5, 2020.

[DSS10] Reinhard Diestel, Alexander Schrijver, and Paul Seymour. Graph theory. *Oberwolfach Reports*, 7(1):521–580, 2010.

[Edm65] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.

[EGP66] Paul Erdös, Adolph W Goodman, and Louis Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.

[EKM13] Péter L. Erdős, Zoltán Király, and István Miklós. On the swap-distances of different realizations of a graphical degree sequence. *Combinatorics, Probability and Computing*, 22(3):366–383, 2013.

[EM64] Paul Erdös and Leo Moser. On the representation of directed graphs as unions of orderings. *Math. Inst. Hung. Acad. Sci*, 9:125–132, 1964.

[EMN18] Michelle Edwards, Gary MacGillivray, and Shahla Nasserasr. Reconfiguring minimum dominating sets: the γ-graph of a tree. *Discussiones Mathematicae Graph Theory*, 38(3):703–716, 2018.

[Eul41] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, pages 128–140, 1741.

[FGMO18] Stephen Finbow, Serge Gaspers, Margaret-Ellen Messinger, and Paul Ottaway. A note on the eternal dominating set problem. *International Journal of Game Theory*, 47(2):543–555, 2018.

[FH76] Stéphane Foldes and Peter L Hammer. *Split graphs*. Universität Bonn. Institut für Ökonometrie und Operations Research, 1976.

[FHHH11] Gerd Fricke, Sandra Hedetniemi, Stephen Hedetniemi, and Kevin Hutson. γ-graphs of graphs. *Discussiones Mathematicae Graph Theory*, 31(3):517–531, 2011.

[Fis93] David C Fisher. The domination number of complete grid graphs. *preprint*, 1993.

[FJP15] Carl Feghali, Matthew Johnson, and Daniël Paulusma. Kempe equivalence of colourings of cubic graphs. *Electronic notes in discrete mathematics*, 49:243–249, 2015.

[FMvB15] S Finbow, ME Messinger, and MF van Bommel. Eternal domination on $3\times$ n grid graphs. *Australian Journal of Combinatorics*, 61(2):156–174, 2015.

[GHH05] Wayne Goddard, Sandra M Hedetniemi, and Stephen T Hedetniemi. Eternal security in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 52:169–180, 2005.

[GJ79] Michael R Garey and David S Johnson. *Computers and intractability*, volume 174. freeman San Francisco, 1979.

[GK08] John L Goldwasser and William F Klostermeyer. Tight bounds for eternal dominating sets in graphs. *Discrete Mathematics*, 308(12):2589–2593, 2008.

[GKM13] John L Goldwasser, William F Klostermeyer, and CM Mynhardt. Eternal protection in grid graphs. *Util. Math*, 91:47–64, 2013.

[GKMP09] Parikshit Gopalan, Phokion G Kolaitis, Elitza Maneva, and Christos H Papadimitriou. The connectivity of boolean satisfiability: computational and structural dichotomies. *SIAM Journal on Computing*, 38(6):2330–2355, 2009.

[Gol11] Oded Goldreich. Finding the shortest move-sequence in the graph-generalized 15-puzzle is np-hard. In *Studies in complexity and cryptography. Miscellanea on the interplay between randomness and computation*, pages 1–5. Springer, 2011.

[GPRT11] Daniel Gonçalves, Alexandre Pinlou, Michaël Rao, and Stéphan Thomassé. The domination number of grids. *SIAM Journal on Discrete Mathematics*, 25(3):1443–1453, 2011.

[Hak62a] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.

[Hak62b] S Louis Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph. i. *Journal of the Society for Industrial and Applied Mathematics*, 10(3):496–506, 1962.

[Hak63a] S. L. Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph ii. uniqueness. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):135–147, 1963.

[Hak63b] S Louis Hakimi. On realizability of a set of integers as degrees of the vertices of a linear graph ii. uniqueness. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):135–147, 1963.

[Hal35] Philip Hall. On representatives of subsets. *Journal of the London Mathematical Society*, 1(1):26–30, 1935.

[HD05] R. A. Hearn and E. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theor. Comput. Sci.*, 343(1-2), October 2005.

[HHS98] Teresa W Haynes, Stephen Hedetniemi, and Peter Slater. *Fundamentals of domination in graphs*. CRC press, 1998.

[HIM$^+$16] Arash Haddadan, Takehiro Ito, Amer E Mouawad, Naomi Nishimura, Hirotaka Ono, Akira Suzuki, and Youcef Tebbal. The complexity of dominating set reconfiguration. *Theoretical Computer Science*, 651:37–49, 2016.

[HP99] Sridhar Hannenhalli and Pavel A Pevzner. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM (JACM)*, 46(1):1–27, 1999.

[HS81] Peter L Hammer and Bruno Simeone. The splittance of a graph. *Combinatorica*, 1(3):275–284, 1981.

[HS14] Ruth Haas and Karen Seyffarth. The k-dominating graph. *Graphs and Combinatorics*, 30(3):609–617, 2014.

[HS17] R Haas and Karen Seyffarth. Reconfiguring dominating sets in some well-covered and other classes of graphs. *Discrete Mathematics*, 340(8):1802–1817, 2017.

[HT74] John Hopcroft and Robert Tarjan. Efficient planarity testing. *Journal of the ACM (JACM)*, 21(4):549–568, 1974.

[HT91] Wen-Lian Hsu and Kuo-Hui Tsai. Linear time algorithms on circular-arc graphs. *Information Processing Letters*, 40(3):123–129, 1991.

[IDH+11] Takehiro Ito, Erik D Demaine, Nicholas JA Harvey, Christos H Papadimitriou, Martha Sideri, Ryuhei Uehara, and Yushi Uno. On the complexity of reconfiguration problems. *Theoretical Computer Science*, 412(12-14):1054–1065, 2011.

[Imm88] Neil Immerman. Nondeterministic space is closed under complementation. *SIAM Journal on computing*, 17(5):935–938, 1988.

[IMNS18] Takehiro Ito, Haruka Mizuta, Naomi Nishimura, and Akira Suzuki. Incremental optimization of independent sets under reachability constraints. *arXiv preprint arXiv:1804.09422*, 2018.

[INZ16] Takehiro Ito, Hiroyuki Nooka, and Xiao Zhou. Reconfiguration of vertex covers in a graph. *IEICE TRANSACTIONS on Information and Systems*, 99(3):598–606, 2016.

[IPS82] Alon Itai, Christos H Papadimitriou, and Jayme Luiz Szwarcfiter. Hamilton paths in grid graphs. *SIAM Journal on Computing*, 11(4):676–686, 1982.

[Ist48] Fáry István. On straight-line representation of planar graphs. *Acta scientiarum mathematicarum*, 11(229-233):2, 1948.

[Jer03] Mark Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Springer Science & Business Media, 2003.

[JS+79] Wm Woolsey Johnson, William Edward Story, et al. Notes on the" 15" puzzle. *American Journal of Mathematics*, 2(4):397–404, 1879.

[Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[Kei93] J Mark Keil. The complexity of domination problems in circle graphs. *Discrete Applied Mathematics*, 42(1):51–63, 1993.

[Kem79] Alfred B Kempe. On the geographical problem of the four colours. *American journal of mathematics*, 2(3):193–200, 1879.

[Klo07] William F Klostermeyer. Complexity of eternal security. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 61:135, 2007.

[KM05] William F Klostermeyer and Gary MacGillivray. Eternally secure sets, independence sets and cliques. *AKCE Int. J. Graphs Comb*, 2:119–122, 2005.

[KM07] William Klostermeyer and Gary MacGillivray. Eternal security in graphs of fixed independence number. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 63, 2007.

[KM09] William Klostermeyer and Gary MacGillivray. Eternal dominating sets in graphs. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 68:97—111, 2009.

[KM12a] William Klostermeyer and Christina M Mynhardt. Eternal total domination in graphs. *Ars Comb.*, 107:473–492, 2012.

[KM12b] William F Klostermeyer and Christina M Mynhardt. Vertex covers and eternal dominating sets. *Discrete Applied Mathematics*, 160(7):1183–1190, 2012.

[KM16] William F Klostermeyer and C. M. Mynhardt. Protecting a graph with mobile guards. *Applicable Analysis and Discrete Mathematics*, 10(1):1–29, 2016.

[KMAA17] William Klostermeyer, Margaret-Ellen Messinger, and Alejandro Angeli Ayello. An eternal domination problem in grids. *Theory and Applications of Graphs*, 4(1):2, 2017.

[KMM12] Marcin Kamiński, Paul Medvedev, and Martin Milanič. Complexity of independent set reconfigurability problems. *Theoretical computer science*, 439:9–15, 2012.

[KS95] John D. Kececioglu and David Sankoff. Exact and approximation algorithms for sorting by reversals, with application to genome rearrangement. *Algorithmica*, 13(1/2):180–210, 1995.

[KZ97] Howard Karloff and Uri Zwick. A 7/8-approximation algorithm for max 3sat? In *Proceedings 38th Annual Symposium on Foundations of Computer Science*, pages 406–415. IEEE, 1997.

[LC05] Linda Lesniak and G Chartrand. *Graphs & digraphs*. Chapman & Hall/CRC, 2005.

[LM18] Daniel Lokshtanov and Amer E Mouawad. The complexity of independent set reconfiguration on bipartite graphs. *ACM Transactions on Algorithms (TALG)*, 15(1):1–19, 2018.

[LMS03] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search. In *Handbook of metaheuristics*, pages 320–353. Springer, 2003.

[LMS19] Ioannis Lamprou, Russell Martin, and Sven Schewe. Eternally dominating large grids. *Theoretical Computer Science*, 794:27–46, 2019.

[Lub86] Michael Luby. A simple parallel algorithm for the maximal independent set problem. *SIAM journal on computing*, 15(4):1036–1053, 1986.

[LVM81] Michel Las Vergnas and Henri Meyniel. Kempe classes and the hadwiger conjecture. *Journal of Combinatorial Theory, Series B*, 31(1):95–104, 1981.

[Mad68] W. Mader. Homomorphiesätze für graphen. *Math. Ann.*, 178:154–168, 1968.

[Mar99] Fabio Martinelli. Lectures on glauber dynamics for discrete spin models. In *Lectures on probability theory and statistics*, pages 93–191. Springer, 1999.

[McC03] Ross M McConnell. Linear-time recognition of circular-arc graphs. *Algorithmica*, 37(2):93–147, 2003.

[Mey78] Henry Meyniel. Les 5-colorations d'un graphe planaire forment une classe de commutation unique. *Journal of Combinatorial Theory, Series B*, 24(3):251–257, 1978.

[MINP19] Fionn Mc Inerney, Nicolas Nisse, and Stéphane Pérennes. Eternal domination in grids. In *International Conference on Algorithms and Complexity*, pages 311–322. Springer, 2019.

[MM13] Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013.

[MN20] CM Mynhardt and S Nasserasr. Reconfiguration of colourings and dominating sets in graphs: a survey. *arXiv preprint arXiv:2003.05956*, 2020.

[MNPR17] Amer E Mouawad, Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman. Shortest reconfiguration paths in the solution space of boolean formulas. *SIAM Journal on Discrete Mathematics*, 31(3):2185–2200, 2017.

[MNR14] Amer E Mouawad, Naomi Nishimura, and Venkatesh Raman. Vertex cover reconfiguration and beyond. In *International Symposium on Algorithms and Computation*, pages 452–463. Springer, 2014.

[MR95] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

[MTR19a] Christina M Mynhardt, LE Teshima, and A Roux. Connected k-dominating graphs. *Discrete Mathematics*, 342(1):145–151, 2019.

[MTR19b] C.M. Mynhardt, L.E. Teshima, and A. Roux. Connected k-dominating graphs. *Discrete Mathematics*, 342(1):145–151, January 2019.

[Nis18] Naomi Nishimura. Introduction to reconfiguration. *Algorithms*, 11(4):52, 2018.

[Ore65] Oystein Ore. *Theory of graphs*, volume 38. American Mathematical Soc., 1965.

[Par95] Ian Parberry. A real-time algorithm for the (n2- 1)-puzzle. *Information Processing Letters*, 56(1):23–28, 1995.

[PRST94] Christos H Papadimitriou, Prabhakar Raghavan, Madhu Sudan, and Hisao Tamaki. Motion planning on a graph. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 511–520. IEEE, 1994.

[PS03] Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*. Cambridge university press, 2003.

[Reg07] F Regan. Dynamic variants of domination and independence in graphs. *Graduate Thesis, Rheinischen Friedrich-Wilhlems University, Bonn*, 2007.

[ReV97] Charles S ReVelle. Can you protect the roman empire. *Johns Hopkins Magazine*, 49(2):40, 1997.

[Rou73] Nicholas D Roussopoulos. A max {m, n} algorithm for determining the graph h from its line graph g. *Information Processing Letters*, 2(4):108–112, 1973.

[RR00] Charles S ReVelle and Kenneth E Rosing. Defendens imperium romanum: a classical problem in military strategy. *The American Mathematical Monthly*, 107(7):585–594, 2000.

[RS04] Neil Robertson and Paul D Seymour. Graph minors. xx. wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

[RS19] Martín Rinemberg and Francisco J Soulignac. The eternal dominating set problem for interval graphs. *Information Processing Letters*, 146:27–29, 2019.

[RTL76] Donald J Rose, R Endre Tarjan, and George S Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on computing*, 5(2):266–283, 1976.

[RW86] Daniel Ratner and Manfred K Warmuth. Finding a shortest solution for the $n \times n$ extension of the 15-puzzle is intractable. In *AAAI*, pages 168–172, 1986.

[Sav70] Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970.

[Sch78] Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226, 1978.

[Sch12] Konrad W Schwerdtfeger. A computational trichotomy for connectivity of boolean satisfiability. *Journal on Satisfiability, Boolean Modeling and Computation*, 8(3-4):173–195, 2012.

[Sen51a] J. Senior. Partitions and their representative graphs. *American Journal of Mathematics*, 73(3):663–689, 1951.

[Sen51b] James K Senior. Partitions and their representative graphs. *American Journal of Mathematics*, 73(3):663–689, 1951.

[Sip96] Michael Sipser. Introduction to the theory of computation. *ACM Sigact News*, 27(1):27–29, 1996.

[SMN14] A. Suzuki, A. Mouawad, and N. Nishimura. Reconfiguration of Dominating Sets. *CoRR*, 1401.5714, 2014.

[SMN15] Akira Suzuki, Amer E. Mouawad, and Naomi Nishimura. Reconfiguration of dominating sets. *Journal of Combinatorial Optimization*, 32(4):1182–1195, August 2015.

[Spi84] Daniel Kornhauser Gary Miller Paul Spiralris. Coordinating pebble motion o_n graphs, the diameter of permutation groups, and applications. 1984.

[Spi94] Jeremy Spinrad. Recognition of circle graphs. *Journal of Algorithms*, 16(2):264–282, 1994.

[Ste99] Ian Stewart. Defend the roman empire! *Scientific American*, 281(6):136–138, 1999.

[Sur09] Pavel Surynek. A novel approach to path planning for multiple robots in biconnected graphs. In *2009 IEEE International Conference on Robotics and Automation*, pages 3613–3619. IEEE, 2009.

[Sze88] Róbert Szelepcsényi. The method of forced enumeration for nondeterministic automata. *Acta Informatica*, 26(3):279–284, 1988.

[Tay81a] R. Taylor. Contrained switchings in graphs. In Kevin L. McAvaney, editor, *Combinatorial Mathematics VIII*, pages 314–336, Berlin, Heidelberg, 1981. Springer Berlin Heidelberg.

[Tay81b] Richard Taylor. Contrained switchings in graphs. In *Combinatorial Mathematics VIII*, pages 314–336. Springer, 1981.

[Tho84] Andrew Thomason. An extremal function for contractions of graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 95(2):261–265, 1984.

[Val81] Leslie G Valiant. Universality considerations in vlsi circuits. *IEEE Transactions on Computers*, 100(2):135–140, 1981.

[vBvB16] Christopher M van Bommel and Martin F van Bommel. Eternal domination numbers of $5\times$ n grid graphs. *J. Comb. Math. Comb. Comput*, 97:83–102, 2016.

[vdH13] Jan van den Heuvel. The complexity of change. *Surveys in combinatorics*, 409(2013):127–160, 2013.

[Viz64] Vadim G Vizing. On an estimate of the chromatic class of a p-graph. *Discret Analiz*, 3:25–30, 1964.

[Viz68] Vadim G Vizing. Some unsolved problems in graph theory. *Russian Mathematical Surveys*, 23(6):125, 1968.

[VSZ+20] Virgélot Virgile, Ben Seamone, Andrés Zacarías, Fionn Mc Inerney, Aaron Krim-Yee, Jerry Huang, Alexander Hassler, and Alizée Gagnon. A method for eternally dominating strong grids. *Discrete Mathematics & Theoretical Computer Science*, 22, 2020.

[Wag37] Klaus Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937.

[Whi92] Hassler Whitney. Congruent graphs and the connectivity of graphs. In *Hassler Whitney Collected Papers*, pages 61–79. Springer, 1992.

[Wil74] Richard M Wilson. Graph puzzles, homotopy, and the alternating group. *Journal of Combinatorial Theory, Series B*, 16(1):86–96, 1974.

[Wil99a] Todd G Will. Switching distance between graphs with the same degrees. *SIAM Journal on Discrete Mathematics*, 12(3):298–306, 1999.

[Wil99b] Todd G. Will. Switching distance between graphs with the same degrees. *SIAM Journal on Discrete Mathematics*, 12(3):298–306, 1999.

[Wro18] Marcin Wrochna. Reconfiguration in bounded bandwidth and tree-depth. *Journal of Computer and System Sciences*, 93:1–10, 2018.

[YG80] Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.

[YU16] Takeshi Yamada and Ryuhei Uehara. Shortest reconfiguration of sliding tokens on a caterpillar. In *International Workshop on Algorithms and Computation*, pages 236–248. Springer, 2016.

[Zuc06] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 681–690. ACM, 2006.

Index