Domination and identification games in graphs

Fionn Mc Inerney

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Jeux de Domination et d'Identification dans les Graphes

Fionn MC INERNEY

Inria Sophia Antipolis - Méditerranée

Présentée en vue de l'obtention du grade de docteur en Informatique **de l'**Université Côte d'Azur
Dirigée par : Nicolas Nisse
Soutenue le : 08 Juillet 2019

Devant le jury, composé de :
Steve Alpern, Professor, The University of Warwick
Victor Chepoi, Professeur, Aix-Marseille Université
Paul Dorbec, Professeur, IUT Caen-Normandy, GREYC
Sylvain Gravier, Directeur de Recherches CNRS, Institut Fourier
Nicolas Nisse, Chargé de Recherches, Inria Sophia Antipolis
Aline Parreau, Chargé de Recherches CNRS, LIRIS

Domination and Identification Games in Graphs

Jury:

PhD Supervisor

Nicolas Nisse, Chargé de Recherches, Inria Sophia Antipolis

Reviewers

Paul Dorbec, Professeur, IUT Caen-Normandy, GREYC
Sylvain Gravier, Directeur de Recherches CNRS, Institut Fourier

Examinators

Steve Alpern, Professor, The University of Warwick
Victor Chepoi, Professeur, Aix-Marseille Université
Aline Parreau, Chargé de Recherches CNRS, LIRIS

Acknowledgements

First, I would like to thank Paul and Sylvain for their time and effort in reviewing my thesis. I also want to thank Steve, Victor, and Aline for accepting to be in my jury.

Next, I want to thank Nicolas, without whom, I could not have completed this PhD. I have learned so much in these 3 years working with you, it is incredible. You are so driven and you pushed me just hard enough so that I could handle it but also excel. Moreover, you have been a good friend over these 3 years and I have enjoyed many great team nights out with you and conference trips as well! Thank you so much!

I would also like to thank all the people I had the pleasure of working with over these 3 years: Dominique, Julien, Anthony, Nancy, Nathann, Danielle, François, Stephen, Alizée, Alex, Jerry, Melissa, Aaron, Nícolas M., Dorian, Andrés, Margaret-Ellen, Nicolas N., Richard, Stéphane, Rudini, Ben, Kasper, and Virgélot.

I want to thank all of the COATI team for such a wonderful time I have spent with you throughout this thesis. All the great discussions, jokes, and nights out were so much fun. I will miss you all a lot and hope to visit often. A special thanks to Christelle, David, Jean-Claude, Michel, and Patricia who are not in the other lists but are very important to me.

I also want to thank all the PhD students in COATI that either had to complain to me about stuff they were doing or had to listen to me complain about stuff I was doing: Adrien, Ali, Andrea, Nicolas H., Guillaume, and William.

I would like to thank all my friends here for all the great times we had together: Adam, Andrea, Dimitra T., Eleni, Emanuele, Giuseppe, Gui, Julien, Lamberto, Lefteris, Natassa, and Vera.

I want to thank all the ping pong people for many "productive" hours spent throughout the course of my thesis but, especially, my main competitors and teammates: Ali, Andrea, Joanna, Julien, Leo, Majus, Minus (mon titulaire), Natassa, and William. I would also like to thank all the other team members I forced to play and all the visitors we forced to play.

I also want to thank the Inria football team which I had a lot of fun playing on and managing. A special thanks to those I played quite often with: Abib, Adam, Amine, Brice, Dorian, Farhood, Federico, Florent, Florian, Fred H., Guillaume, Jef, Jérémy, Kostia, Lamberto, Nicolas L., Pascal, Pierre, and Romain.

Lastly, I would like to thank my parents: Sean and Pascale, my brother: Killian, and all the rest of my family, but especially, Stéphane, Laëtitia, Elisa, Tom and Romane, who took me in for Christmas every year.

Résumé

Dans cette thèse, les jeux à 2 joueurs dans les graphes et leurs aspects algorithmiques et structurels sont étudiés. Nous explorons tout d'abord le jeu de *domination éternelle* ainsi que sa généralisation, le *jeu de l'espion*, deux jeux qui reposent sur les ensembles dominants dynamiques. Dans ces deux jeux, une équipe de gardes poursuit un attaquant ou espion rapide dans un graphe, avec l'objectif de rester près de lui éternellement. Le but est de calculer le *nombre de domination éternelle* (*nombre de gardes* pour le jeu de l'espion) qui est le nombre minimum de gardes nécessaires pour réaliser l'objectif. La *dimension métrique* des digraphes et une version séquentielle de la dimension métrique des graphes sont aussi étudiées. Ces deux problèmes ont pour objectif de trouver un sous-ensemble de sommets de taille minimum tel que tous les sommets du graphe sont identifiés uniquement par leurs distances aux sommets du sous-ensemble. En particulier, dans ce dernier problème, on peut "interroger" un certain nombre de sommets par tour. Les sommets interrogés retournent leurs distances à une cible cachée. Le but est de minimiser le nombre de tours nécessaires pour localiser la cible. Ces jeux et problèmes sont étudiés pour des classes de graphe particulières et leurs complexités temporelles sont aussi étudiées.

Précisément, dans le Chapitre 3, il est démontré que le jeu de l'espion est NP-difficile et les nombres de gardes des chemins et des cycles sont présentés. Ensuite, des résultats sur le jeu de l'espion dans les arbres et les grilles sont présentés. Notamment, nous démontrons une équivalence entre la variante fractionnaire et la variante "intégrale" du jeu de l'espion dans les arbres qui nous a permise d'utiliser la programmation linéaire pour concevoir ce que nous pensons être le premier algorithme exact qui utilise la variante fractionnaire d'un jeu pour résoudre sa variante "intégrale". Dans le Chapitre 4, des bornes asymptotiques sur le nombre de domination éternelle de la grille du roi sont présentées. Dans le Chapitre 5, des résultats sur la NP-complétude du jeu de Localisation sous différentes conditions (et une variante de ce jeu) sont présentés. Notamment, nous démontrons que le problème est NP-complet dans les arbres. Malgré cela, nous concevons un $(+1)$-algorithme d'approximation qui résout le problème en temps polynomial. Autant que nous sachions, il n'existe pas d'autres telles approximations pour les jeux dans les graphes. Finalement, dans le Chapitre 6, des résultats sur la dimension métrique des graphes orientés sont présentés. En particulier, les orientations qui maximisent la dimension métrique sont explorées pour les graphes de degré borné, les tores et les grilles.

Abstract

In this thesis, 2-player games on graphs and their algorithmic and structural aspects are studied. First, we investigate two dynamic dominating set games: the *eternal domination game* and its generalization, the *spy game*. In these two games, a team of guards pursue a fast attacker or spy in a graph with the objective of staying close to him eternally and one wants to calculate the *eternal domination number* (*guard number* in the spy game) which is the minimum number of guards needed to do this. Secondly, the *metric dimension* of digraphs and a sequential version of the metric dimension of graphs are then studied. These two problems are those of finding a minimum subset of vertices that uniquely identify all the vertices of the graph by their distances from the vertices in the subset. In particular, in the latter, one can probe a certain number of vertices per turn which return their distances to a hidden target and the goal is to minimize the number of turns in order to ensure locating the target. These games and problems are studied in particular graph classes and their computational complexities are also studied.

Precisely, in Chapter 3, the NP-hardness of the spy game and the guard numbers of paths and cycles are first presented. Then, results for the spy game on trees and grids are presented. Notably, we show an equivalence between the fractional variant and the "integral" version of the spy game in trees which allowed us to use Linear Programming to come up with what we believe to be the first exact algorithm using the fractional variant of a game to solve the "integral" version. In Chapter 4, asymptotic bounds on the eternal domination number of strong grids are presented. In Chapter 5, results on the NP-completeness of the *Localization game* under different conditions (and a variant of it) and the game in trees are presented. Notably, we show that the problem is NP-complete in trees, but despite this, we come up with a polynomial-time $(+1)$-approximation algorithm in trees. We consider such an approximation to be rare as we are not aware of any other such approximation in games on graphs. Lastly, in Chapter 6, results on the metric dimension of oriented graphs are presented. In particular, the orientations which maximize the metric dimension are investigated for graphs of bounded degree, tori, and grids.

Contents

Part I

Introduction to Games, Definitions, and Notation

Chapter 1

Introduction

1.1 2-Player Combinatorial Games

Games in general have been vastly studied for their applications, entertainment factor, and because the problems therein are usually intriguing and easy to state. The field of the study of games is very large. In this section, I will try to give an overview of the different areas in games and where the results of this thesis fit into this scheme. Before doing so, a general definition of a game must be given. A *game* is a form of play in which the *players* take turns changing the *position* of the game in order to achieve some winning condition with the *rules* of the game defining how the game proceeds, the capabilities of the players, and the winning conditions. Games with more than two players have been studied (see, *e.g.*, [86, 103, 122]), but seeing as the field of 2-player games is already large enough and has been much more extensively studied, I will only focus on 2-player games.

There are two main strands of the study of 2-player games: *combinatorial game theory* and *economic game theory*. Combinatorial game theory typically deals with games of no chance that are *sequential* in nature (the players take turns) and where all players have *perfect information* (both the state of the game and what both players can do is known to all players). Economic game theory typically deals with games of chance that may or may not involve *simultaneous* play (sequential games also exist) and where some or all players may have *imperfect information* (*e.g.*, the moves of one player may not be known to the other). I will again focus on games that fall within combinatorial game theory but note that a gray area exists between both these strands (there are games that can be considered in both strands) since the definitions of both strands are not stringent.

Combinatorial games can be broken down into many subclasses of games and here I will discuss some of these. *Impartial* games are sequential perfect information games in which a play or move available to one player is always available to the other player (they both have the same capabilities of moving on their respective turns). An example of such a game is *Nim*, first studied in [33], in which both players take turns removing objects from piles and the first player who cannot remove an object loses. Other examples of impartial games are *Sprouts* [66], *Kayles* [56], *Cram* [67], etc. In Cram, for

example, the two players take turns placing dominoes vertically or horizontally on a rectangular board (although it is not limited to these boards), and the first player who cannot place a domino, loses. *Partisan* games are like impartial games except a play or move available to one player is not available to the other player. Examples include chess and Go since each player can only play with their tokens or pieces. This thesis will focus on partisan games.

Games can also be played on different surfaces such as boards or graphs. This thesis will focus on partisan games played on graphs. In the next paragraph, we still mention games like Nim, which are not played on graphs.

Partisan games (and impartial ones in most cases) can be broken down into subclasses based on their winning conditions. Five of these subclasses will be defined in this section. The first one is games played under the *normal play convention*. In these games, the first player who cannot move, loses. Examples include Nim, Cram, and *Domineering* [67]. Domineering, for example, is the same game as Cram, except that one of the players can only place their dominoes vertically and the other, only horizontally. The second subclass is *Misère play*, which has the opposite winning condition of normal play, which is that the first player who cannot move, wins. Recently, there has been the development of a theory of *scoring* games, where the winner is the one with the greater score [95, 96, 97]. The fourth one is *maker-breaker* games, where one player wants to maximize a score (often the number of turns that occur) and the other wants to minimize it. Examples include the *Domination game* [38] and the *graph colouring game* [26]. For example, in the first, two players take turns adding vertices to a set such that each new vertex added dominates at least one new vertex, and one player wants to maximize the size of this set while the other wants to minimize it. This game has been vastly studied, see, *e.g.*, [37, 39, 52]. Finally, *pursuit-evasion* games can be seen as a special type of maker-breaker games. Pursuit-evasion games are played on graphs, in which a team of agents (pursuers) collaborate to accomplish a specified task while the evader tries to stop the pursuers achieving their goal.

The principal games in this thesis fit into the category of pursuit-evasion games. The three games that are studied are the *eternal domination game* [72], the *spy game* [j-3], and the *localization game* [c-5]. In the first two, a team of guards pursue a fast attacker or spy in a graph with the objective of staying close to him eternally and the goal is to calculate the *eternal domination number* (*guard number* in the spy game) which is the minimum number of guards needed to do this. In the latter, one can probe a certain number of vertices per turn which return their distances to a hidden immobile target and the goal is to minimize the number of turns in order to ensure locating the target. The latter is more of a 1-player game as the second player (the one that places the target) only has one move at the beginning of the game. The first two games fall into the following categories of games. They are both pursuit-evasion, partisan, sequential, perfect information, and combinatorial games of no chance. The localization game is, roughly, a pursuit-evasion, partisan, sequential, imperfect information, and combinatorial game of no chance.

1.2 Related Work: Pursuit-Evasion Games in Graphs

1.2.1 Games in Graphs & Cops and Robbers

We now focus on games in graphs and specifically, pursuit-evasion games. The full state of the art of the games studied in this thesis as well a lighter state of the art of the problems that motivate the study of these games is given in this chapter. The reader is referred to Chapter 2 for graph theoretic notation and definitions if needed.

Games in graphs have been vastly studied due to their various applications and because the problems therein are usually intriguing and easy to state. This has attracted a lot of interest to the field. In particular, the main focus has been the study of two-player games in which the objective is to minimize the "resources" (*e.g.*, number of agents) of one player while ensuring they can always "win" or achieve their goal regardless of their opponent's strategy. For such games where one player controls a team of agents with the goal of accomplishing a specified task, the combinatorial problem of minimizing the number of agents (resources) to accomplish the task and the algorithmic problem of computing a "winning" strategy for the agents to accomplish the task, have applications in robotics, network security, artificial intelligence, graph theory, logic, routing, telecommunications, etc. (*e.g.*, see [60, 81, 92]).

Game of cops and robbers. The most well-known two-player games of this type are the pursuit-evasion games. In particular, the game of *cops and robbers* [104, 106] has been extensively studied and most of the other pursuit-evasion games have been derived from or have been created as a consequence of this game. In cops and robbers, a team of *cops* place themselves on the vertices of a graph. Then, a single *robber* places himself on a vertex. Turn-by-turn, first each of the cops may move to one of their neighbours or stay put, and then the robber may do the same. The cops win if, after a finite number of turns, a cop *captures* the robber, *i.e.*, moves to the vertex the robber currently occupies. Otherwise, if, for an infinite number of turns, the robber can evade capture, then the robber wins. The objective of the game is to determine the *cop number*, denoted by $c(G)$, of a graph G, which is the minimum number of cops necessary to ensure capturing the robber in G. For example, for any tree T, $c(T) = 1$. Indeed, in any tree T, there is one unique shortest path between any two vertices and if the cop follows the shortest path between himself and the robber at each turn, he will eventually capture the robber as the robber cannot move past or around the cop. Another easy example is the case of cycles of size at least 4, it is easy to see that one cop is not enough but also that 2 are always enough, so $c(C_n) = 2$.

Complexity of cops and robbers. Typically, the method of research for such a game is to first determine its computational complexity and then to solve the game for particular classes of graphs. Deciding whether $c(G) \leq k$ when a graph G is part of the input but k is fixed, is polynomial-time solvable [25]. Deciding whether $c(G) \leq k$ when a graph G and an integer k are part of the input, is NP-hard and W[2]-hard [61]. There is also no polynomial-time algorithm to approximate the cop number to within a multiplicative factor $c \log n$, where $c > 0$ is a constant and n is the size of the graph [61]. It was then proven that cops and robbers is PSPACE-hard [101]. Finally, it was proven that cops and robbers is EXPTIME-complete [87]. As can be seen in the following para-

graph, a typical approach when a problem is shown to be at least NP-hard in general, is to determine under what conditions the problem is computationally tractable. There are many approaches such as designing better exponential-time algorithms, studying approximation algorithms, considering further the parameterized complexity, etc. The common approach, which is the one taken in this thesis, however, is to restrict the problem to particular classes of graphs. The hope being to show that the problem is in P for such classes or, even better, that a closed formula exists to calculate the parameters that we seek in such classes.

With this in mind, the common approach to tackle these games is to try and solve them for particular graph classes by using their structural properties, which is why pursuit-evasion games are often considered to give a better understanding of structural properties of graphs.

Particular graph classes. In cops and robbers, most of the work has been dedicated to particular graph classes. Graphs with cop number equal to one were characterized in [104]. Three cops are sufficient in planar graphs and this bound is sharp [16]. Two cops are sufficient in outerplanar graphs and this bound is sharp [48]. The cop number of intersection graphs was studied in [70] where, among other things, they showed that the cop number of interval filament graphs is at most 2, the cop number of outer-string graphs is at most 4, and the cop number of string graphs is at most 15. For graphs G with genus g, $c(G) \leq \lfloor \frac{3g}{2} \rfloor + 3$ and in the same paper, one of the two main conjectures for cops and robbers was given and that is that $c(G) \leq g + 3$ [112]. The other main conjecture is that of Meyniel, which asks whether $c(G) = O(\sqrt{n})$ for any connected graph G on n vertices. This conjecture is considered the biggest conjecture (since many strong researchers have worked on it) in cops and robbers and is mentioned in [65] as a personal communication between Frankl and Meyniel in 1985. For the bipartite graph $G(P)$ formed from the points and the lines of a projective plane, where the points and the lines are the two partitions, $c(G(P)) = \sqrt{n}$, where n is the number of vertices in $G(P)$. Therefore, if Meyniel's conjecture is true, then the bounds on the cop number (for connected graphs) are asymptotically tight. Several works have been done in regards to Meyniel's conjecture (see, *e.g.*, [46, 65, 99, 113]), yet no one has managed to even prove that $c(G) = O(n^{1-\epsilon})$ for any $\epsilon > 0$. For more on cops and robbers, see the book [29].

In this thesis, the intersection of pursuit-evasion games in graphs and domination and identification in graphs is studied. Once introduced, it will be clear that all these problems are related to distances in graphs. The study of which, from different angles, allows for a better understanding of distance properties of graphs. In the next subsections, some background on domination and identification in graphs is given to provide a basis for the study of the games.

1.2.2 Domination in Graphs

A subset of vertices $S \subseteq V$ is a *dominating* set of $G = (V, E)$ if, for every vertex $v \in V$, either $v \in S$ or $uv \in E$ and $u \in S$. The *domination number* of a graph G, denoted by $\gamma(G)$, is the size of a minimum dominating set of G. Domination in graphs has found its applications in, *e.g.*, facility location problems, designing electrical networks,

and land surveying [55, 80]. Much like games in graphs, the approach to studying domination in graphs has been the same. The problem of deciding whether $\gamma(G) \leq k$ when G and k are part of the input is NP-complete [84]. Moreover, the problem is W[2]-complete [54]. The problem is also $\alpha \ln n$-inapproximable [17]. The fastest exact algorithm (to the best of our knowledge) for finding a minimum dominating set runs in time $O(1.4969^n)$ [121]. Seeing as the problem was most likely not computationally tractable in general graphs, it was studied in particular graph classes. For example, the domination number of planar graphs with diameter at most two (three respectively) is at most 3 (10 respectively) [100]. Goddard and Henning showed that the bound is tight for planar graphs with diameter two and showed that the graph they constructed is the unique graph of diameter two with domination number equal to 3 [73]. MacGillivray and Seyffarth also gave an example of a planar graph with diameter three with domination number equal to 6 [100]. The exact domination number of Cartesian grids was only recently determined in 2011 [75]. The domination number has also been investigated, *e.g.*, for series-parallel graphs [82], the cross product of paths [45], and the cross products of graphs in general [76]. A big conjecture concerning domination in graphs is due to Vizing and it dates back to 1968. It states that, for all finite graphs G and H, $\gamma(G\Box H) \geq \gamma(G)\gamma(H)$ [123]. Significant progress was made by Clark and Suen when they proved that $\gamma(G\Box H) \geq \frac{1}{2}\gamma(G)\gamma(H)$ [47]. For more on progress made on the conjecture, the interested reader is referred to the survey [36]. For more on domination in graphs and its variants see [80].

Domination in graphs is clearly related to the distance properties of a graph. For example, the results on planar graphs with diameters at most 2 and 3 above show that the domination number of a graph depends on the diameter of the graph.

1.2.2.1 Eternal Domination Game

In terms of dominating games, the *all-guards-move* model of the eternal domination game [72] and its generalization, the spy game [j-3], will be considered in this thesis. The eternal domination game was introduced by Burger et al. [40] in 2004. The game is played on a simple undirected graph G. There is a team of *guards* playing against an *attacker*. The guards place themselves on the vertices of G and then, at each turn, the attacker first attacks a vertex $v \in V$ and then only one guard may move to a vertex adjacent to his current position (vertex) and one guard must move to v, otherwise, the attacker wins. If a guard moves to v, then the guards are said to have *defended* against the attack. If the guards can defend against an infinite sequence of attacks, then the guards win. Hence, the guards must always maintain a dominating set. The objective of the game is to determine the *eternal domination number* of G, denoted by $\gamma^\infty(G)$, which is the minimum number of guards necessary in order to ensure winning against the attacker.

Goddard et al. [72] introduced the all-guards-move model of the eternal domination game in 2005. In this variant, each guard may move to a neighbour on their turn, with at least one guard having to move to the attacked vertex v after each attack. Such a problem found its applications in the study of military strategies that date back to the Roman Empire where armies needed to be mobilized to defend the empire but there

were only a limited number of armies [18, 108, 109, 119]. There are two variants of this model, one in which multiple guards may occupy a vertex at any time, and one in which at most one guard may occupy a vertex at any time. Both associated parameters are defined analogously to $\gamma^\infty(G)$, with the former, which will be mainly considered in this thesis, being denoted by $\gamma^\infty_{all}(G)$, and the latter being denoted by $\gamma^{*\infty}_{all}(G)$. The constraint that at most one guard may occupy a vertex is not a stringent one for the original eternal domination game since it is easy to see that the guards do not gain any advantage from this as only one guard may move at each turn [40]. However, there exist graphs in which this constraint is important for the all-guards-move model, *i.e.*, $\gamma^\infty_{all}(G) < \gamma^{*\infty}_{all}(G)$ for any of these graphs G [91].

It is clear that $\gamma^\infty_{all}(G) \leq \gamma^{*\infty}_{all}(G) \leq \gamma^\infty(G)$, however these parameters can also be bounded by well-known graph parameters such as the domination number, independence number, and clique cover number, denoted by $\gamma(G)$, $\alpha(G)$, and $\theta(G)$ respectively. In particular, the following chain of inequalities holds, $\gamma(G) \leq \gamma^\infty_{all}(G) \leq \gamma^{*\infty}_{all}(G) \leq \alpha(G) \leq \gamma^\infty(G) \leq \theta(G)$ [40, 72].

From the variants of the eternal domination game mentioned thus far, seeing as this thesis focuses on the study of γ^∞_{all}, only the state of the art of this variant will be presented. Again, the complexity of the game was studied and the game in particular graph classes. Deciding whether $\gamma^\infty_{all}(G) \leq k$ is NP-hard when G and k are part of the input and this holds for split graphs [21]. While it is not explicitly stated, it can be seen that the problem is also W[2]-hard through the reductions given in [21]. Note that it is not known whether the problem is in NP or in PSPACE and so this leaves the very interesting open problem of determining the exact complexity class, *i.e.*, is it NP-complete? PSPACE-complete? EXPTIME-complete? In the case of particular graph classes, paths and cycles are trivial with $\gamma^\infty_{all}(P_n) = \lceil \frac{n}{2} \rceil$ and $\gamma^\infty_{all}(C_n) = \lceil \frac{n}{3} \rceil$ [72]. A linear-time algorithm to calculate $\gamma^\infty_{all}(T)$ for any tree T was conceived in [89]. In terms of interval graphs, first it was shown that, for any proper interval graph G, $\gamma^\infty_{all}(G) = \alpha(G)$ [34]. Recently, this result was improved, that is, it was shown that, for any interval graph G, $\gamma^\infty_{all}(G) = \alpha(G)$ [110], with the proof being much shorter as well. There are many papers that have focused on determining $\gamma^\infty_{all}(G)$ for Cartesian grids. Exact values have been determined for $2 \times n$ grids [74] and $4 \times n$ grids [22]. It proved to be more difficult for $3 \times n$ grids with asymptotically tight bounds being given in [59] and improved in [49]. Finally, the best upper bound for Cartesian grids in general is $\gamma^\infty_{all}(P_n \Box P_m) = \gamma(P_n \Box P_m) + O(n+m)$ [94]. Note that all the results mentioned in this paragraph also hold for $\gamma^{*\infty}_{all}$.

There are other variants of the eternal domination game but they are just mentioned in passing as they are outside the scope of this thesis. In the *eternal total domination* game, a *total dominating set* must be maintained at each turn, that is, a dominating set in which every vertex in the dominating set is also dominated by another vertex (adjacent to another vertex) in the dominating set [90]. In the *eviction model* of the eternal domination game, a vertex containing a guard is attacked at each turn, and the guard at that vertex must move to an adjacent vertex with the condition that the guards maintain a dominating set each turn [88]. The eternal domination game has also been studied on digraphs [19]. The interested reader is referred to the survey [91]

for more information and results on the eternal domination game and its variants.

Another way of better understanding distances in graphs and how they can be used leads to the study of identification problems in graphs.

1.2.3 Identification in Graphs

Problems where one wants to distinguish the vertices of a graph by their distances from a smallest subset of its vertices are commonly referred to as identifying problems. Many of these problems exist, with *identifying codes* [85], *adaptive identifying codes* [24], and *locating dominating sets* [118] asking for the vertices to be distinguished by their neighbourhood in the subset chosen. *Resolving sets*, in which one wants to distinguish the vertices of a graph by their distances to the vertices in such a set, have been extensively studied [77, 117]. Formally, for a graph G, an ordered subset of vertices $S = \{v_1, \ldots, v_k\} \subseteq V(G)$, and a vertex $u \in V(G)$, let the *distance vector* between S and u be $D(S, u) = (dist(u, v_1), dist(u, v_2), \ldots, dist(u, v_k))$. The set S is a resolving set, if, for any two vertices $u, w \in V(G)$, the distance vectors $D(S, u)$ and $D(S, w)$ are distinct. The size of a minimum resolving set of a graph G is called the *metric dimension* of G and is denoted by $MD(G)$. This problem models, *e.g.*, the detection of an intruder in a facility [62]. Sensors that can detect an intruder at a certain distance can be placed in and around the facility and security wants to be able to know the exact location of the intruder given the distance information provided by the sensors. The sensors may be expensive however, and so one wants to minimize the number of sensors they have to install for the security of their facility.

The associated decision problem, *i.e.*, deciding whether $MD(G) \leq k$ when G and k are part of the input, was first shown to be NP-complete in general graphs in [68]. Thus, this motivated studying the problem in restricted graph classes. The problem was further shown to be NP-complete in planar graphs [50] and in graphs of diameter 2 [64], and W[2]-hard (parameterized by the solution's size) [78]. On the positive side, the problem is FPT when parameterized by the treelength of the graph [23]. The metric dimension of trees can be computed as follows [77, 117]. Contract all vertices of degree 2 and let L be the set of leaves in the remaining tree T' and let S be the set of vertices of degree greater than 1 that are adjacent to at least one leaf in T'. Then, for each vertex in S, taking all adjacent vertices but one that are in L is a resolving set. Bounds on the metric dimension were also shown for interval and permutation graphs [63]. For more on the metric dimension of graphs, the interested reader is referred to the surveys [20, 42].

A variant of resolving sets, called *centroidal bases*, where the vertices of a graph must be distinguished by their relative distances to the probed vertices was introduced in [62]. A formal definition is given later in Chapter 5. However, intuitively, when a set of vertices are probed, it results in the knowledge of which vertex is the closest to the target, second closest, etc., without indicating the exact distances between these vertices and the target. It is also known if two vertices probed are at the same distance from the target. The size of a minimum centroidal basis of a graph G is called the *centroidal dimension* of G and is denoted by $CD(G)$. The associated decision problem, *i.e.*, deciding whether $CD(G) \leq k$ when G and k are part of the input, was shown to

be NP-complete, and almost tight bounds on the centroidal dimension of paths were given in [62].

1.2.3.1 Sequential Identifying Games in Graphs

In terms of identifying games, the LOCALIZATION problem and the RELATIVE-LOCALIZATION problem, introduced in [c-5] are considered in this thesis. In [115], Seager initiated the study of the following sequential locating game: an invisible and immobile target is hidden at some vertex t, and, at every step, one vertex can be probed to retrieve its distance to t, and the objective is to locate t using the minimum number of steps. Seager gave bounds and exact values on this minimum number of steps in particular subclasses of trees (*e.g.*, subdivisions of caterpillars) [115] but left the problem open in trees in general. The LOCALIZATION problem is essentially a generalization of the metric dimension of a graph and this sequential locating game (multiple vertices may be probed each turn instead of just one). Indeed, instead of all the information from probings being given at once, they are given sequentially. This is natural since maybe it is not possible to probe enough sensors all at once in order to ensure locating an intruder in a facility or network, but of course one still wants to locate the intruder. So, one might want to know the minimum number of sequential probings that are necessary to ensure locating an intruder. That is, in the LOCALIZATION problem, an immobile *target* is hidden at a vertex and one probes the vertices of the graph over multiple turns in order to locate the target. For a target hidden at a vertex $u \in V(G)$, probing a vertex $v \in V(G)$ results in the knowledge of the distance between u and v, *i.e.*, $dist(u, v)$. Precisely, given a graph G and two integers $k, \ell \geq 1$, the LOCALIZATION problem asks whether an immobile target hidden at a vertex of G can be located in at most ℓ turns by probing at most k vertices per step.

The first of such sequential localization games studied the case of a moving target, with the first one being proposed by Seager in 2012 [114]. In these games, after each probing, the target may move to one of its neighbours. Sometimes an extra condition on the movement of the target, known as "backtracking", is not allowed, *i.e.*, the target may not move to a neighbour that has just been probed. The goal in these games is to locate the target in a finite number of turns while minimizing the number of vertices that can be probed at each turn.

As with the other related problems and games, the complexity of such games was studied as well as such games in particular graph classes. The number of times all of the edges of a graph must be subdivided in order to guarantee locating a moving target by probing one vertex (k vertices respectively) per step was investigated in [41] ([79] respectively). A *locatable* graph is one in which there exists a strategy, that probes one vertex per step, that locates, in a finite number of steps, a target that may not backtrack. All trees were shown to be locatable and bounds on the number of steps it takes to locate the target in trees were exhibited in [114]. This upper bound was improved in [35]. The case of a target that may backtrack was considered in trees in [116]. Let $\zeta(G)$ be the minimum integer k such that there exists a strategy, that probes k vertices per step, for locating a moving target in G in a finite number of steps. In [32], it was shown that deciding whether $\zeta(G) \leq k$ is NP-hard and that $\zeta(G)$ is

not bounded in the class of graphs G with treewidth 2. Moreover, $\zeta(G) \leq 2$ for any outerplanar graph G [28]. The case where relative distances are returned instead of exact distances was also studied in this context. Let $\zeta^{rel}(G)$ be the minimum integer k such that there exists a strategy, that probes k vertices per step, for locating a moving target in G in a finite number of steps when relative distances are returned instead of exact distances. It was notably shown that $\zeta^{rel}(G) \leq 3$ for any outerplanar graph G and that deciding whether $\zeta^{rel}(G) \leq k$, when G and k are part of the input, is NP-hard [31].

1.2.3.2 Identifying in Oriented Graphs

Motivated by the numerous works on the metric dimension of a graph and its variants, the continued study of the metric dimension of digraphs is presented in this thesis. Chartrand, Rains, and Zhang were the first to study the metric dimension of digraphs in 2000 [43]. In their definition, they require that the vertices of D be distinguished by their distances (in D) to the vertices in $R \subseteq V(D)$. They also require that the distances from each pair of distinct vertices to the vertices in R which distinguish them be defined (not ∞). If both these conditions are met, then R is said to be a resolving set of D. Our definition, which will be the one used in this thesis, requires that the vertices of D be distinguished by their distances (in D) from the vertices in $R \subseteq V(D)$. We also allow for undefined distances (∞) to be used as well. However, if only strong digraphs D are considered, then our definitions are equivalent except that they consider the digraph $\widetilde{D}$ which is obtained by reversing all the arcs of D.

The works on the metric dimension of digraphs that have appeared between theirs and ours are with respect to the original definition. In [43], digraphs with metric dimension 1 were characterized and the open problem of a characterization of digraphs that admit a metric dimension (following their definition) still remains open as far as we know. If G has a Hamiltonian path, then there exists an orientation D of G such that $MD(D) = 1$ (orient all edges of a Hamiltonian path from the first vertex towards the last vertex, and all remaining edges in the opposite direction) [43, 98]. The associated decision problem, *i.e.*, deciding whether $MD(D) \leq k$ when a strong digraph D and k are part of the input, is NP-complete [107]. Bounds on the metric dimension of various digraph families were later exhibited (Cayley digraphs [57], line digraphs [58], tournaments [98], digraphs with cyclic covering [105], De Bruijn and Kautz digraphs [107], etc.). In [44], the worst orientations of G for the metric dimension were considered, *i.e.*, orientations of G with maximum metric dimension.

This motivated our study of $WOMD(G)$ which is the maximum value of $MD(D)$ over all strong orientations D of G. This definition is then extended to graph families as follows. For any family $\mathcal{G}$ of 2-edge-connected graphs (a graph has strong orientations if and only if it is 2-edge-connected), let $WOMD(\mathcal{G}) = \max_{G \in \mathcal{G}} \frac{WOMD(G)}{|V(G)|}$. In [44], the authors proved that, for every positive integer k, there exist infinitely many graphs for which the metric dimension of any of its strongly-connected orientations is exactly k. They have also proved that there is no constant k such that the metric dimension of any tournament is at most k.

The necessary background has been presented now so that the main study of this

thesis can be introduced, that is, the intersection of pursuit-evasion games in graphs and domination and identification in graphs.

1.3 Results of this Thesis: Dominating and Identifying Games

In this thesis, games that depend on domination and identification are studied. Identification problems and games are then extended to the metric dimension of oriented graphs to better understand the role of distances in digraphs in relation to the metric dimension of digraphs. We consider these games as, sometimes, the static nature of dominating sets is not sufficient for applications of problems of a sequential nature like the eternal domination game (as discussed in sections 1.2.2.1 and 1.3.1). Other games, like the spy game (section 1.3.1 and Chapter 3) and the Localization problem (section 1.3.3 and Chapter 5), can be seen as generalizations of well-studied games (parameters respectively) in graphs, like the eternal domination game (metric dimension respectively). All these problems have in common the notion of distances in graphs (and digraphs) and how they can be used and better understood. The main focus is to determine the computational complexity of the decision problems associated with such games and to solve the associated problems (games) in certain graph classes as is the common approach for the individual components that make up these games, as shown above.

1.3.1 Chapter 3: Dominating Games - Spy Game

A generalization of the eternal domination game seemed very natural as maybe a guard just needs to be close enough in order to prevent the attacker from doing something and maybe the attacker has speed restrictions, so that he cannot just attack any vertex but one at most a certain distance from his current position. These generalizations are very natural for a better study of, *e.g.*, military strategies, as was already done for the eternal domination game. This is why the spy game is the other dominating game that will be focused on in this thesis.

The spy game was introduced in [j-3] (joint work with N. Cohen, N. Martins, N. Nisse, S. Pérennes, and R. Sampaio) and further studied in [c-8, j-4] (joint works with N. Cohen, N. Nisse, and S. Pérennes), and the results of these papers (mentioned below) are included in Chapter 3.

In the spy game, similarly to the eternal domination game, a team of guards play against an attacker called the *spy*. The rules of the spy game are defined in terms of two parameters, the *speed* $s \in \mathbb{N}^*$ of the spy and the prescribed distance $d \in \mathbb{N}$. The spy first places itself at a vertex and then the guards do the same. The spy moves at speed s on his turn, that is, he may move to a vertex that is at distance at most s from his current position (vertex). Then, the guards move as they do in the all-guards-move model of the eternal domination game, that is, each guard may move to one of its neighbours and multiple guards may occupy the same vertex. If there is at least one guard at distance at most d from the spy after the guards move, then the guards are said to *control* the spy. The guards must maintain that they control the spy (after they

move) at all times, otherwise, they lose, and if they can do so for an infinite sequence of moves of the spy, then they win. The aim of the game is to determine the *guard number* of G, denoted by $gn_{s,d}(G)$, which is the minimum number of guards necessary in order to ensure winning against the spy. Note that $\gamma^{\infty}_{all}(G) = gn_{s,0}(G)$ if $s \geq diam(G)$.

In light of the research that has been done on the eternal domination game and other pursuit-evasion games, in Chapter 3, it is shown that the spy game is NP-hard, W[2]-hard, and exact values for the guard number of paths and almost tight bounds for cycles are given (our results from [j-3]). This is then followed by the study of the spy game in trees and grids (our results from [c-8, j-4]). Linear Programming is used to calculate the guard number of trees and a corresponding strategy in polynomial time by showing that the fractional guard number equals the classical guard number in trees. As far as we know, this is the first exact algorithm for such combinatorial games using Linear Programming, and we were not able to solve the problem without it. Lastly, motivated by the study of the eternal domination game in Cartesian grids, by considering the fractional relaxation of the spy game again, bounds are obtained on the (fractional) guard number of Cartesian grids and tori.

1.3.2 Chapter 4: Dominating Games - Eternal Domination Game

The extensive work done for the eternal domination game on Cartesian grids motivated our (joint work with N. Nisse and S. Pérennes [c-9]) study of the game on strong grids and the general technique that we obtained for grid-like graphs in general which can be seen as Cayley graphs obtained from abelian groups which are truncated. Precisely, in Chapter 4, it is shown that for all $n, m \in \mathbb{N}^*$ such that $m \geq n$, $\lfloor \frac{n}{3} \rfloor \lfloor \frac{m}{3} \rfloor + \Omega(n+m) = \gamma^{\infty}_{all}(P_n \boxtimes P_m) = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil + O(m\sqrt{n})$ and this general technique is presented. This same result is then shown to hold for $\gamma^{*\infty}_{all}$.

1.3.3 Chapter 5: Identifying Games - Localization Problem

In Chapter 5, our results from [c-5] (joint work with J. Bensmail, D. Mazauric, N. Nisse, and S. Pérennes) for both the Localization problem and the Relative-Localization problem are presented. Precisely, motivated by the study of the game with a moving target in trees T, we show that the Localization problem is NP-complete in trees but, despite this, a $(+1)$-approximation algorithm for determining the optimal number of turns to locate the target and a corresponding strategy is presented. As with the associated decision problems of both the parameters $\zeta(G)$ and $\zeta^{rel}(G)$, that were shown to be NP-hard, we show that both the Localization and Relative-Localization problems are NP-complete, even when k or ℓ (but not both) is fixed, but both problems are polynomial when both k and ℓ are fixed.

1.3.4 Chapter 6: Identifying in Oriented Graphs

In Chapter 6, our results from [c-6] (joint work with J. Bensmail and N. Nisse) on the continued study of the metric dimension of digraphs are presented. As in the other

works where specific graph classes were studied, we study $WOMD$ for specific graph families such as graphs with bounded maximum degree, Eulerian tori, and grids.

The next section gives an overview of the results of the thesis, as well as the organization of the thesis. The last section gives a list of the publications included in this thesis and a list of other publications.

1.4 Results and Layout of this Thesis

- In Chapter 2, complexity and graph theoretic notation and definitions used throughout the thesis are introduced.
- In Chapter 3, joint results on the NP-hardness of the spy game and the guard numbers of paths and cycles with N. Cohen, N. Martins, N. Nisse, S. Pérennes, and R. Sampaio from [j-3] are first presented. Then, joint results for the spy game on trees and grids with N. Cohen, N. Nisse, and S. Pérennes from [c-8, j-4] are presented. Notably, we show an equivalence between the fractional variant and the "integral" version of the spy game in trees which allowed us to use Linear Programming to come up with what we believe to be the first exact algorithm using the fractional variant of a game to solve the "integral" version.
- In Chapter 4, joint results on the eternal domination number of strong grids with N. Nisse and S. Pérennes from [c-9] are presented.
- In Chapter 5, joint results on the NP-completeness of the Localization game under different conditions (and a variant of it) and the game in trees with J. Bensmail, D. Mazauric, N. Nisse, and S. Pérennes from [c-5] are presented. Notably, we show that the problem is NP-complete in trees, but despite this, we come up with a polynomial-time $(+1)$-approximation algorithm in trees. We consider such an approximation to be rare as we are not aware of any other such approximation in games on graphs.
- In Chapter 6, joint results on the metric dimension of oriented graphs with J. Bensmail and N. Nisse from [c-6] are presented. In particular, the orientations which maximize the metric dimension are investigated for graphs of bounded degree, tori, and grids.
- Lastly, in Chapter 7, concluding remarks are given and further work is discussed.

1.5 My Publications

1.5.1 List of Publications Included in this Thesis

Journal Publications

[j-3] N. Cohen, N. A. Martins, F. Mc Inerney, N. Nisse, S. Pérennes, and R. Sampaio. Spy-game on graphs: Complexity and simple topologies. *Theoretical Computer Science*, 725:1-15, 2018.

[j-4] N. Cohen, F. Mc Inerney, N. Nisse, and S. Pérennes. Study of a combinatorial game in graphs through linear programming. *To appear in Algorithmica*, 2019.

International Conferences

[c-5] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Sequential metric dimension. In *Proceedings of the 16th Workshop on Approximation and Online Algorithms, WAOA 2018*, 2018. RR, https://hal.inria.fr/hal-01717629.

[c-6] J. Bensmail, F. Mc Inerney, and N. Nisse. Metric dimension: From graphs to digraphs. In *Proceedings of the 10th Latin & American Algorithms, Graphs and Optimization Symposium, LAGOS 2019*, 2019. RR, https://hal.inria.fr/hal-01938290.

[c-8] N. Cohen, F. Mc Inerney, N. Nisse, and S. Pérennes. Study of a combinatorial game in graphs through linear programming. In *Proceedings of the 28th International Symposium on Algorithms and Computation (ISAAC 2017)*, pages 22:1-22:13, 2017. RR, https://hal.archives-ouvertes.fr/hal-01462890.

[c-9] F. Mc Inerney, N. Nisse, and S. Pérennes. Eternal domination in grids. In *Proceedings of the 11th International Conference on Algorithms and Complexity (CIAC 2019)*, pages 311-322, 2019. RR, https://hal.inria.fr/hal-01790322v2.

National Conferences

[c-10] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Localiser une cible dans un graphe. In *20es Rencontres Francophones sur les aspects d'Algorithmiques des Télécommunications (AlgoTel 2018)*, 2018.

[c-11] J. Bensmail, F. Mc Inerney, and N. Nisse. Dimension Métrique des Graphes Orientés. In *21es Rencontres Francophones sur les aspects d'Algorithmiques des Télécommunications (AlgoTel 2019)*, 2019.

[c-12] N. Cohen, N. Martins, F. Mc Inerney, N. Nisse, S. Pérennes, and R. Sampaio. Enquêter dans les graphes. In *19es Rencontres Francophones sur les aspects d'Algorithmiques des Télécommunications (AlgoTel 2017)*, 2017.

1.5.2 List of Other Publications

Journal Publications

[j-1] S. D. Andres, M. Huggan, F. Mc Inerney, and R. J. Nowakowski. The Orthogonal Colouring Game on Graphs. To appear in *Theoretical Computer Science*.

[j-2] A. Bonato, N. Clarke, D. Cox, S. Finbow, F. Mc Inerney, and M. E. Messinger. Hyperopic Cops and Robbers. To appear in *Theoretical Computer Science*.

International Conferences

[c-7] A. Bonato and F. Mc Inerney. The game of wall cops and robbers. In Proceedings of the 2nd International Conference on Computational Intelligence, *ICC3 2015*, volume 412 of Advances in Intelligent Systems and Computing. Springer, 2015.

Submitted

[s-13] S. D. Andres, F. Dross, M. Huggan, F. Mc Inerney, and R. J. Nowakowski. On the Complexity of Orthogonal Colouring Games and the NP-completeness of Recognizing Graphs Admitting a Strictly Matched Involution. Submitted to *Discrete Applied Mathematics*.

[s-14] J. Bensmail, F. Mc Inerney, and Kasper Szabo Lyngsie. On $\{a, b\}$-edge weightings of bipartite graphs with odd a, b. Submitted to *DMTCS*.

[s-15] A. Gagnon, A. Hassler, J. Huang, A. Krim-Yee, F. Mc Inerney, A. Mejía Zacarías, B. Seamone, and V. Virgile. A method for eternally dominating strong grids. Submitted to *DMTCS*.

Chapter 2

Graph Theory and Complexity: Definitions and Notation

2.1 Undirected Graphs

Standard graph theory terminology and notation will be used (see, *e.g.*, [51] for reference). A *graph* $G = (V, E)$ consists of a set of *vertices* V and a set of *edges* E where each edge consists of a pair of vertices in V. The *order* of a graph is the total number of vertices in the graph and is usually denoted by n, *i.e.,* $|V| = n$. The *size* of a graph is the total number of edges in the graph and is usually denoted by m, *i.e.,* $|E| = m$. For any two vertices u, v such that the pair (u, v) is an edge, the edge will be denoted by uv. A graph G is said to be *simple* if between any two vertices $u, v \in V$, there is at most one edge and there are no edges from a vertex to itself (these are called *loops*). All graphs considered in this thesis are simple. A graph is said to be *undirected* if the edges have no directions on them, *e.g.*, the edge uv is equivalent to the edge vu. The following definitions apply to undirected graphs, after which *directed* graphs will be introduced.

A vertex u is said to be *adjacent* to or a *neighbour* of a vertex v if the edge uv exists. The (*open*) *neighbourhood* of a vertex v, denoted by $N(v)$, is the set of all vertices adjacent to v. The *closed* neighbourhood of a vertex v is denoted by $N[v] = N(v) \cup \{v\}$. The *degree* of a vertex $v \in V$, denoted by $d(v)$, is the size of the neighbourhood of v, *i.e.*, $d(v) = |N(v)|$. The minimum degree of a graph G is denoted by $\delta(G) = \min_{v \in V} d(v)$. The maximum degree of a graph G is denoted by $\Delta(G) = \max_{v \in V} d(v)$. A vertex v is said to be *universal* if $N[v] = V$. A vertex v is said to be *isolated* if $d(v) = 0$.

A subset of vertices $S \subseteq V$ is called a *dominating set* if for any vertex $v \in V$, there exists a vertex $u \in S$ (it may be the case that $v = u$) such that $v \in N[u]$. The minimum size of a dominating set of a graph G is called the *domination number* of G and is denoted by $\gamma(G)$. A subset of vertices $S \subseteq V$ is called an *independent set* if for any two vertices $u, v \in S$, $uv \notin E$. The maximum size of an independent set in a graph G is called the *independence number* of G and is denoted by $\alpha(G)$.

A *subgraph* $H = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E \cap (V' \times V')$. An *induced* subgraph $H = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subseteq V$, $E' \subseteq E$, and for all edges $uv \in E$ such that $u, v \in V'$, $uv \in E'$. A

component of a graph G is an induced subgraph of G. A (vertex) *partition* of a graph G is a division of its vertices into vertex-disjoint components.

The Cartesian product of two sets A and B is denoted by $A \times B = \{(a, b) | a \in A, b \in B\}$. The Cartesian product of two graphs $G = (V, E)$ and $H = (V', E')$, denoted by $G \square H$, is a graph such that the vertex set is the Cartesian product $V \times V'$ and two vertices $(a, a'), (b, b') \in G \square H$ are adjacent if $a = b$ and $a'b' \in E'$ or if $a' = b'$ and $ab \in E$. The strong product of two graphs $G = (V, E)$ and $H = (V', E')$, denoted by $G \boxtimes H$, is a graph such that the vertex set is the Cartesian product $V \times V'$ and two vertices $(a, a'), (b, b') \in G \boxtimes H$ are adjacent if $a = b$ and $a'b' \in E'$ or if $a' = b'$ and $ab \in E$ or if $a'b' \in E'$ and $ab \in E$.

A *path* on n vertices, denoted by P_n, is a sequence of vertices $v_1, \ldots, v_n$ where $v_i v_{i+1}$ is an edge for all integers $1 \leq i \leq n - 1$. The *endpoints* of the path are v_1 and v_n. A *cycle* on n vertices, denoted by C_n, is equivalent to a path on n vertices where, in addition, the edge $v_n v_1$ exists. A *tree* is a graph T that contains no cycles. The vertices of a tree are also called *nodes*. A *leaf* is a node of degree 1 in a tree. A *clique* (or *complete graph*) on n vertices, denoted by K_n, is a graph in which for every two vertices $u, v \in V(K_n)$, the edge uv exists. A graph is *bipartite* if its vertices can be partitioned into two components which are independent sets. A *complete bipartite graph*, denoted by $K_{n,m}$, consists of an independent set of size n and one of size m between which all possible edges exist. A *Cartesian grid* is the Cartesian product of two paths P_n and P_m, denoted by $G_{n \times m} = P_n \square P_m$, that is commonly referred to as an $n \times m$ *grid* or simply just a *grid*. A *strong grid* is the strong product of two paths P_n and P_m, denoted by $SG_{n \times m} = P_n \boxtimes P_m$, that is commonly referred to as an $n \times m$ *strong grid*. A *torus* is the Cartesian product of two cycles C_n and C_m, denoted by $T_{n \times m} = C_n \square C_m$, that is commonly referred to as an $n \times m$ *torus*. For a family of intervals S_i, $i = 1, \ldots, n$, an *interval graph* is a graph formed by creating a vertex v_i for each interval S_i and there is an edge between two vertices v_i and v_j $(i \neq j)$ if the intervals S_i and S_j overlap (intersect). A *proper* interval graph is an interval graph in which no interval is properly contained in another interval, *i.e.*, for any two intervals S_i and S_j $(i \neq j)$, it is not the case that $S_i \subsetneq S_j$ nor that $S_j \subsetneq S_i$.

A graph G is *connected* if for any two vertices $u, v \in V$, there exists a subgraph H such that H is a path with u and v as its endpoints, otherwise, the graph is *disconnected*. The *distance* between two vertices $u, v \in G$, denoted by $dist_G(u, v)$ or simply $dist(u, v)$ when no ambiguity is possible, is the length of a shortest path from u to v. If no such path exists, then $dist(u, v) = \infty$. The *diameter* of a graph G, denoted by $diam(G)$, is the length of a longest shortest path in G, *i.e.*, $\max_{u,v \in V} dist(u, v)$. For any graph G, any integer ℓ and $v \in V(G)$, $N_\ell[v]$ is the set of vertices at distance at most ℓ from v in G.

A *clique cover* of a graph is a union of complete subgraphs such that the union of their vertex sets is V. The *clique cover number* of a graph G, denoted by $\theta(G)$, is the minimum number of cliques whose union is a clique cover of G.

An *isomorphism* of graphs G and H is a bijective mapping $\sigma : V(G) \to V(H)$ such that for all $u, v \in V(G)$, $uv \in E(G) \Leftrightarrow \sigma(u)\sigma(v) \in E(H)$. The graphs G and H are said to be *isomorphic* in this case.

2.2 Directed Graphs

Note that directed graphs will only be considered in Chapter 6. A directed graph or *digraph* $D = (V, A)$ is a graph in which each of the edges has a given direction and the edges are called *arcs* instead. If an edge is directed from a vertex u to a vertex v (from a vertex v to a vertex u respectively), then the arc is denoted by (u, v) ((v, u) respectively). An *orientation* D of an undirected graph G is obtained by orienting every edge $uv \in E(G)$ either from u to v, resulting in the arc (u, v) or conversely from v to u, resulting in the arc (v, u). An *oriented* graph is a digraph that is an orientation of a simple graph.

For an arc $(u, v) \in A$, u is an *in-neighbour* of v and v is an *out-neighbour* of u. The set of all in-neighbours (out-neighbours respectively) of a vertex $v \in V$ is denoted by $N^-(v)$ ($N^+(v)$ respectively). The *in-degree* (*out-degree* respectively) of a vertex $v \in D$, denoted by $d^-(v)$ ($d^+(v)$ respectively), is the number of in-neighbours (out-neighbours respectively) of v, *i.e.*, $d^-(v) = |N^-(v)|$ and $d^+(v) = |N^+(v)|$. The *maximum in-degree* (*maximum out-degree* respectively) of a vertex in a digraph D is denoted by $\Delta^-(D)$ ($\Delta^+(D)$ respectively). A *directed* path is an orientation of a path such that all arcs are oriented in the same direction, *i.e.*, all vertices of the path have in-degree (out-degree respectively) at most 1. A digraph D is *strongly connected* if, for every two vertices $u, v \in V$, there exists (as a subgraph) a directed path from u to v and from v to u. The *directed distance* (or simply distance when no ambiguity is possible) between two vertices $u, v \in V$ in D, denoted by $dist_D(u, v)$ (or simply $dist(u, v)$ when no ambiguity is possible), is the length of a shortest directed path from u to v in D.

2.3 Complexity

In this section, we give a brief informal background on complexity theory. The computationally tractable problems are considered to be those in P, which is the class of problems that can be solved in polynomial time. Most "difficult" problems are NP-complete (see [68] for a large catalogue of such problems), which means they are both in NP and NP-hard (at least as hard as any other problem in NP), where NP is the class of problems in which a solution to the problem can be verified in polynomial time. Clearly, $P \subseteq NP$. PSPACE is the class of problems which can be solved using polynomial space in memory. It is known that $NP \subseteq PSPACE$. EXPTIME is the class of problems which can be solved in exponential time and $PSPACE \subseteq EXPTIME$. Lastly, it is known that $P \subset EXPTIME$ and hence, problems which are EXPTIME-hard (at least as hard as any other problem in EXPTIME) are not considered to be computationally tractable.

To further analyse the difference in computational complexity of problems that are in the same complexity class, *e.g.*, two NP-hard problems, the notion of parameterized complexity is used. Formally, for a computational decision problem Π that takes an input x, let Σ^* denote the set of all possible strings over a fixed finite alphabet and let the language $L_\Pi \subseteq \Sigma^*$ denote the set of all strings x that describe an input for which the answer is "yes". A *parameterized language* $L \subseteq \Sigma^* \times \mathbb{N}$ is defined analogously as

the set of all pairs (x, k) of a string and an integer that describe an input for which the answer is "yes". A *parameterized problem* $L \subseteq \Sigma^* \times \mathbb{N}$ is *fixed parameter tractable* (FPT) if, for an input $(x, k) \in \Sigma^* \times \mathbb{N}$, there is an algorithm that decides whether $(x, k) \in L$ in time $f(k)n^{O(1)}$ where $|x| = n$ is the size of the main input. Intuitively, a problem is FPT if, for some fixed parameter k, the problem can be solved in time $f(k) \cdot n^{O(1)}$, where n is the size of the main input (for graph problems, this is the number of vertices of the graph). For example, the vertex cover problem is FPT in the size of the solution (there is an exhaustive search algorithm that solves it in time $2^k n^{O(1)}$), the treewidth problem is FPT in the size of the solution (can be solved in time $O(2^{k^3} n)$) [27], the metric dimension problem is FPT in the treelength [23], etc.

A *kernel* for a problem Π is a polynomial-time algorithm that transforms an instance (x, k) of Π into an instance (x', k') of Π such that:

1. $(x, k) \in L \Leftrightarrow (x', k') \in L$;
2. $|x'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$.

A problem is FPT if and only if it is decidable and it admits a kernel.

There is also a hierarchy of parameterized complexity classes known as the W hierarchy. For all integers $i \geq 0$, a problem is $W[i]$-complete if it is both in $W[i]$ and $W[i]$-hard. Without going into the details, $FPT = W[0]$ and $W[i] \subseteq W[j]$ for all integers $i \leq j$, with the important hypothesis that $FPT \neq W[1]$. To prove $W[i]$-hardness for a problem B, an instance (x, k) of a problem A that is known to be $W[i]$-hard must be reduced to an instance (x', k') of the problem B in time $f(k)|x|^{O(1)}$ and such that:

1. (x, k) is a "yes"-instance of $A \Leftrightarrow (x', k')$ is a "yes"-instance of B;
2. $k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$.

To illustrate this finer analysis of problems in the same complexity class, consider the clique, vertex cover, and dominating set problems which are three NP-hard problems [68]. The vertex cover problem is known to be FPT while the clique problem is $W[1]$-hard [53] and thus, it should be that no FPT algorithm exists for it, and lastly, the dominating set problem is $W[2]$-hard [53], making it even more unlikely to be FPT than the clique problem. Much like for classical complexity where the classical problem to reduce from is SAT, in parameterized complexity, the classical $W[1]$-hard ($W[2]$-hard respectively) problem to reduce from is the clique problem (dominating set problem respectively).

Part II

Domination Games

Chapter 3

Spy Game

3.1 Introduction

In this chapter, the study of the notion of sequential dominating sets is continued with the generalization of the eternal domination game, called the spy game. Specifically, this chapter focuses on results published in the paper that introduced the spy game [j-3], which is joint work with N. Cohen, N. Martins, N. Nisse, S. Pérennes, and R. Sampaio, and results published in [c-8, j-4], which are joint works with N. Cohen, N. Nisse, and S. Pérennes. The spy game is a two-player game played on a graph G as follows. Let $k, d, s \in \mathbb{N}$ be three integers such that $k > 0$ and $s > 0$. One player uses a set of k *guards* occupying some vertices of G while the other player plays with one *spy* initially standing at some node. Note that several guards and even the spy could occupy the same vertex.

Initially, the spy is placed at some vertex of G. Then, the k guards are placed at some vertices of G. Then, the game proceeds turn-by-turn. At each turn, first the spy may move along at most s edges (s is the *speed* of the spy). Then, each guard may move to its neighbour. If there is at least one guard at distance at most d from the spy after the guards' move, then the guards are said to control the spy. The spy wins if, after a finite number of turns (after the guards' move), it reaches a vertex at distance greater than d from every guard. The guards win otherwise, in which case the guards always control the spy.

Given a graph G and two integers $d, s \in \mathbb{N}$, $s > 0$, let the *guard number*, denoted by $gn_{s,d}(G)$, be the minimum number of guards required to control a spy with speed s at distance d for an infinite number of turns, against all the spy's strategies. In what follows, only the case where $s \geq 2$ is considered since if $s = 1$, then the game is trivial as one guard can just follow a shortest path to the spy at each turn. See Figure 3.1 for an example of the spy winning against one guard when $s = 2$ and $d = 1$.

Recently, a new framework was proposed that considers a fractional variant of these combinatorial games (roughly where agents may be split into arbitrarily small entities) and uses Linear Programming to obtain new bounds and algorithms [71]. While this approach seems not to be successful to handle cops and robber games, it has been fruitful in designing approximation algorithms for other combinatorial games. Precisely,

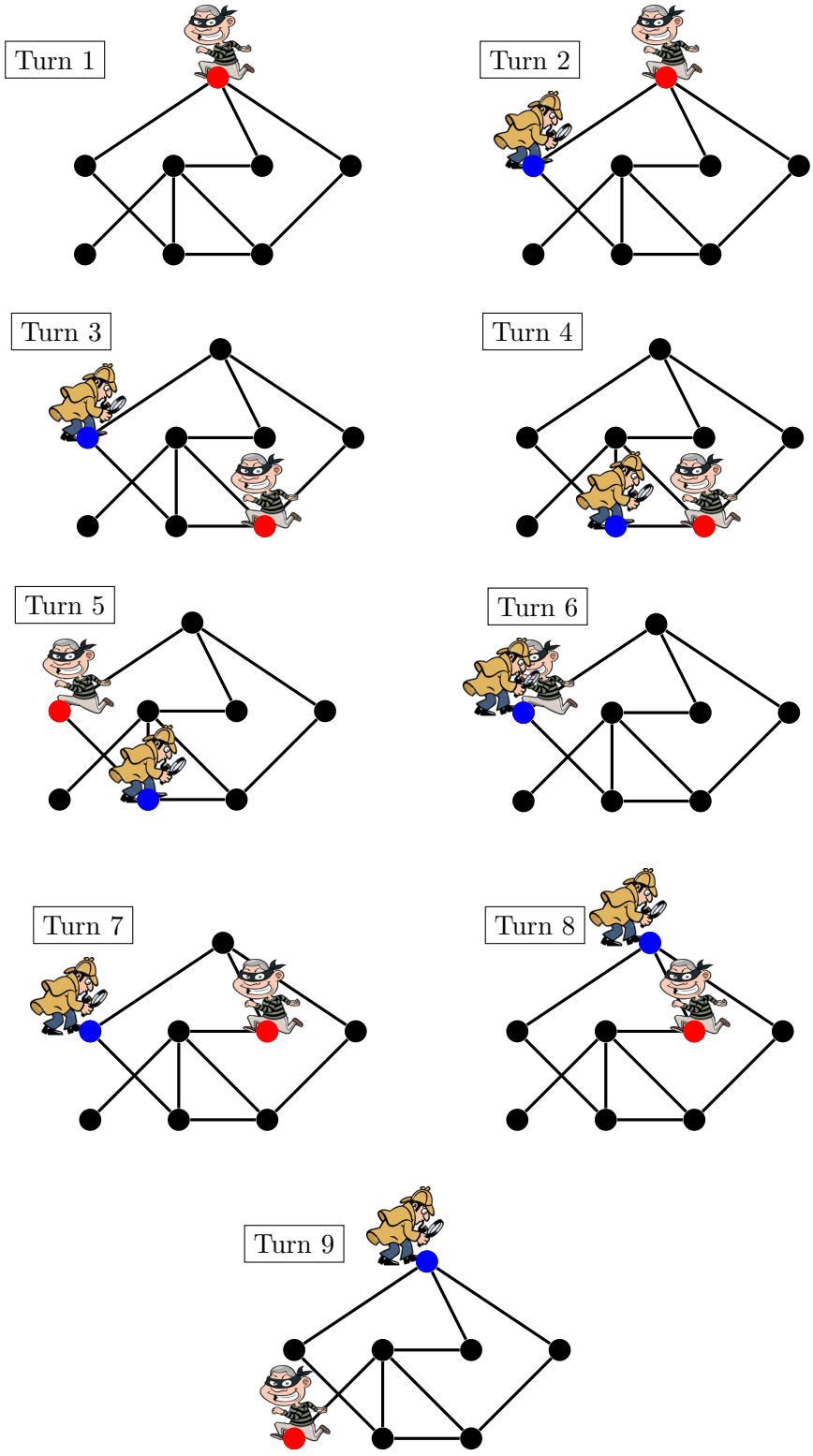


Figure 3.1: Ex: spy (red) wins in the spy game ($s = 2$, $d = 1$) against 1 guard (blue).

it allowed to design polynomial-time approximation algorithms for various (NP-hard) variants of the surveillance game [71]. In this chapter, we present a new successful application of this approach. In particular, we show that the spy game can be solved in polynomial time in trees using this approach. We emphasize that, as far as we know, it is the first exact algorithm for such combinatorial games using a Linear Programming approach and that we were not able to solve it without this technique. Indeed, we show that the techniques used in Section 3.3 and for solving the eternal domination game in trees [89] are not necessarily optimal in trees in general for the spy game. In Section 3.3, for n-node paths, the strategy consists of partitioning the path into $gn_{s,d}(P_n)$ subpaths with one guard assigned to each one. We show that assigning disjoint subtrees to each guard is not necessarily optimal in trees (see Section 3.5). For the eternal domination game in trees T, $\gamma^{\infty}_{all}(T)$ can be computed in linear time [89]. The key property in this simple recursive algorithm is that an optimal strategy consists of partitioning a tree into vertex-disjoint stars, each star being assigned to at most 2 guards. As already mentioned, such a method does not extend to the spy game. We also show the first non-trivial lower bound on the guard number of grids using the fractional variant. We hope that our results will encourage people to use this framework to study combinatorial games and we believe it will enable progress toward solutions of the difficult open problems.

3.1.1 Fractional spy game

Formally, the fractional spy game proceeds as follows in a graph $G = (V, E)$. Let $s \geq 2$, $d \geq 0$ be two integers and let $k \in \mathbb{R}$ such that $k > 0$. First, the spy is placed at a vertex. Then, each vertex v receives some amount $g_v \in \mathbb{R}^+$ (a non-negative real) of guards such that the total amount of guards is $\sum_{v \in V} g_v = k$. Then, on its turn, the spy may first move at distance at most s from its current position. Then, the "fractional" guards move following a flow constrained as follows (see Figure 3.1.1 for an example). For any $v \in V$ and for any $u \in N[v]$[†], there is a flow $f(v, u) \in \mathbb{R}^+$ of guards going from v to $u \in N[v]$ such that $\sum_{u \in N[v]} f(v, u) = g_v$, *i.e.*, the amount of guards leaving v and staying at v is exactly what was at v. Finally, for any vertex $v \in V$, the amount of guards occupying v at the end of the round is $g'_v = \sum_{u \in N[v]} f(u, v)$. We now need to rephrase the fact that the guards control the spy at distance d at the end of each round. This is the case if, after every guards' turn, $\sum_{w \in N_d[x]} g'_w \geq 1$, where x is the vertex occupied by the spy. Let $fgn_{s,d}(G)$ denote the minimum total amount of fractional guards needed to always control at distance d a spy with speed s in a graph G. Note that, by definition, since the fractional game is a relaxation of the "integral" spy game:

Claim 3.1.1. *For any graph G and any $s \geq 2$, $d \geq 0$, $fgn_{s,d}(G) \leq gn_{s,d}(G)$.*

[†]For any graph G, any integer ℓ and $v \in V(G)$, let $N_\ell[v]$ be the set of vertices at distance at most ℓ from v in G and let $N[v] = N_1[v]$.

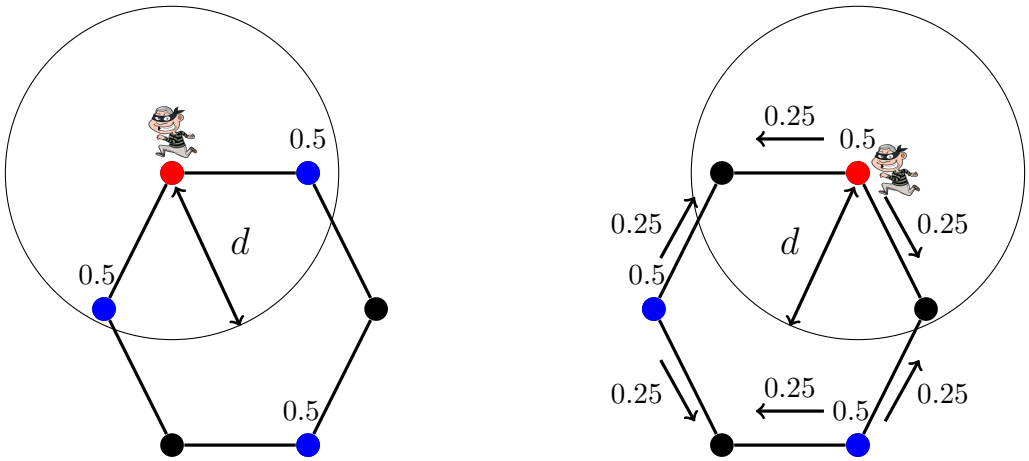


Figure 3.2: Example of the fractional spy game when $s \geq 2$ and $d = 1$. (Left) Initial positions of the spy and the guards, where there are 0.5 guards at each vertex in blue and the spy is at the vertex in red. (Right) The spy moves to the vertex in red and the guards move as illustrated, that is, sending 0.25 guards to each of their neighbours.

3.1.2 Our Results

In this chapter, we initiate the study of the spy game for $s \geq 2$. In Section 3.2, we study the computational complexity of the problem of deciding the guard number of a graph. We prove that computing $gn_{s,d}(G)$ is NP-hard for any $s \geq 2$ and $d \geq 0$ in the class of graphs G with diameter at most $O(d)$ (our result from [j-3]). Then, we consider particular graph classes. In Section 3.3, we precisely characterize the cases of paths and cycles (our results from [j-3]). Precisely, for any $d \geq 0$, $s \geq 2$, we prove that, for any path P_n on n vertices:

$$gn_{s,d}(P_n) = \left\lceil \frac{n}{2d + 2 + \lfloor \frac{2d}{s-1} \rfloor} \right\rceil,$$

and, for any cycle C_n with n vertices:

- $gn_{s,d}(C_n) = \lceil \frac{n}{2d+3} \rceil$ if $0 \leq 2d < s - 1$;
- If $2d \geq s - 1$, then

$$\left\lceil \frac{n + 2\lfloor \frac{2d}{s-1} \rfloor}{2(d + \lfloor \frac{2d}{s-1} \rfloor) + 3} \right\rceil \leq gn_{s,d}(C_n) \leq \left\lceil \frac{n + 2\lfloor \frac{2d}{s-1} \rfloor}{2(d + \lfloor \frac{2d}{s-1} \rfloor) + 1} \right\rceil.$$

In Sections 3.5 and 3.6, we study the spy game in the classes of trees and grids respectively (our results from [c-8, j-4]). We prove that the guard number of any tree can be computed in polynomial time and give non-trivial lower and upper bounds on the fractional guard number of grids. More precisely, for every $s \geq 2$ and $d \geq 0$:

- We design a Linear Program that computes $fgn_{s,d}(T)$ and a corresponding strategy in polynomial time for any tree T. Then, we show that any fractional strategy

(winning for the guards) using k guards in a tree can be turned into a winning (integral) strategy using $\lfloor k \rfloor$ guards. The key argument is that we can restrict the study to what we call *Spy-positional* strategies. Altogether, this shows that, in any tree T, $fgn_{s,d}(T) = gn_{s,d}(T)$, and that $gn_{s,d}(T)$ and a corresponding winning strategy can be computed in polynomial time.

- Then, we show that there is a constant $0 < \beta^* < 1$ such that, for any $n \times n$ grid $G_{n\times n}$ with n large enough, $\Omega(n^{1+\beta^*}) = fgn_{s,d}(G_{n\times n}) \leq gn_{s,d}(G_{n\times n})$. This gives the first non trivial lower bound for the guard number in the class of grids. Finally, for $\alpha = \log_2(1 + \frac{1}{s})$, we show that $fgn_{s,d}(G_{n\times n}) = O(n^{2-\alpha})$. Note that the best known upper bound for $gn_{s,d}(G_{n\times n})$ is $O(n^2)$[†]. A similar bound holds for the $n \times n$ torus.

3.2 Complexity

In this section, we prove that the spy game with speed s and distance d is NP-hard for any $s \geq 2$ and $d \geq 0$. Precisely, we prove the following theorem.

Theorem 3.2.1. *Let $s \geq 2$ and $d \geq 0$ be two fixed integers. The problem that takes an n-node graph G and an integer $k \in \mathbb{N}$ as inputs and aims at deciding whether $gn_{s,d}(G) \leq k$ is NP-hard, W[2]-hard (see [54]) when parameterized by the number of guards, and $\alpha \ln n$-inapproximable in polynomial time for some constant $0 < \alpha < 1$, unless $P = NP$.*

The proof follows from the five Lemmas below. The reduction is from the Set Cover Problem and is divided into three cases: $d + 1 < s < 2d + 2$ (Lemma 3.2.3), $s \geq 2d+2$ (Lemma 3.2.4), and $s \leq d+1$ (Lemmas 3.2.6 and 3.2.7 depending on whether $d \pmod{s-1}$ is greater than $s/2$). The proofs of all the cases are similar but vary slightly depending on the parameters. We present the proofs of all the cases separately for better comprehensibility.

An instance of the Set Cover Problem is a family $\mathcal{S} = \{S_1, \ldots, S_m\}$ of sets and an integer c, and the objective is to decide if there exists a subfamily $\mathcal{C} = \{S_{i_1}, \ldots, S_{i_c}\} \subseteq \mathcal{S}$ such that $|\mathcal{C}| \leq c$ and $S_{i_1} \cup \ldots \cup S_{i_c} = U$, where $U = S_1 \cup \ldots \cup S_m$ (we say that $\mathcal{C}$ is a set cover of U). Given an instance $(\mathcal{S}, c)$ of Set Cover, we construct a graph $G = G_{s,d}(\mathcal{S}, c)$ and an integer $K = K_{s,d}(\mathcal{S}, c)$ such that there exists a cover $\mathcal{C} \subseteq \mathcal{S}$ of U with size at most c if and only if $gn_{s,d}(G) \leq K$. Note that the reductions presented below are actually FPT-reductions and preserve approximation ratio. Therefore, since the Set Cover Problem is W[2]-hard (when parameterized by the size c of the set cover) and has no $\alpha' \ln(n)$ approximation algorithm for some constant $0 < \alpha' < 1$ (unless P=NP) [17], we not only prove the NP-hardness but also the fact that the problem is W[2]-hard (when parameterized by the number of guards) and cannot be approximated in polynomial time up to some logarithmic ratio (unless $P = NP$).

[†]Indeed, $O((n/d)^2)$ vertices are sufficient to dominate every vertex at distance d in $G_{n\times n}$ (tiling the grid with vertex-disjoint balls of radius d).

Definition 3.2.2. Given integers $s \geq 2$ and $d \geq 0$, let $p = p(s,d) = d + \left\lceil \frac{d+1}{s-1} \right\rceil$ and $q = q(s,d)$ be

$$q(s,d) = \begin{cases} 0, & \text{if } d+1 < s < 2d+2, \\ d + \left\lceil \frac{d}{s-1} \right\rceil, & \text{if } s \leq d+1, \\ d, & \text{otherwise.} \end{cases}$$

Let $(\mathcal{S}, c)$ be an instance of Set Cover, where $\mathcal{S} = \{S_1, \ldots, S_m\}$, and let $U = S_1 \cup \ldots \cup S_m = \{u_1, \ldots, u_n\}$. Let $K = K_{s,d}(\mathcal{S}, c)$ be:

$$K_{s,d}(\mathcal{S}, c) = \begin{cases} c, & \text{if } d+1 < s < 2d+2, \\ c+2, & \text{if } s \leq d+1 \text{ and } 1 \leq d \mod (s-1) < \frac{s}{2} - 1, \\ c+1, & \text{otherwise.} \end{cases}$$

Let $G = G_{s,d}(\mathcal{S}, c)$ be the graph defined as follows: for every set $S_j \in \mathcal{S}$, create a new vertex S_j in G and, for every element $u_i \in U$, create a path U_i with p vertices $u_{i,1}, \ldots, u_{i,p}$. Make $\{S_1, \ldots, S_m\}$ a clique in G (add all possible edges). If $u_i \in S_j$, add the edge $u_{i,1}S_j$ in G. Create a new vertex z_0 and add all possible edges between z_0 and $\{S_1, \ldots, S_m\}$ in G. Finally, if $q > 0$, create a path Z with q vertices $z_1, \ldots, z_q$, and add the edge $z_0 z_1$. Moreover, if $s \leq d+1$ and $1 \leq d \mod (s-1) < \frac{s}{2} - 1$, then create a path Z' with q vertices $z'_1, \ldots, z'_q$ and add the edge $z_0 z'_1$.

See Figures 3.3-3.5 for examples.

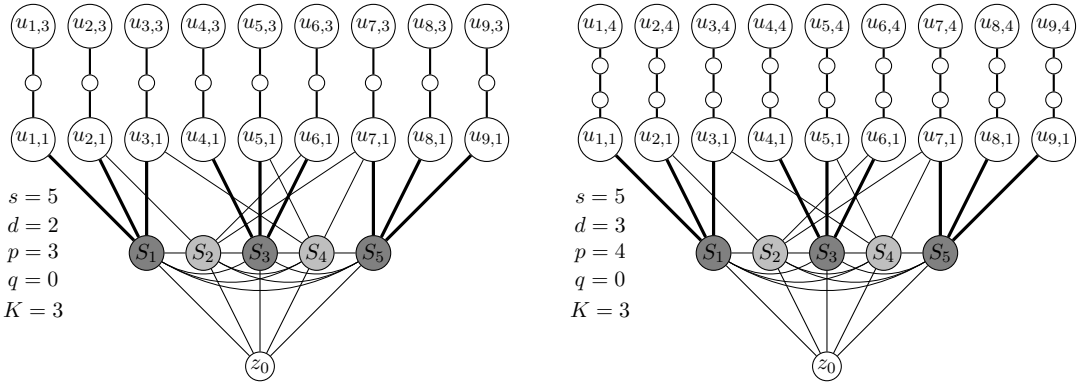


Figure 3.3: Reduction from Set Cover instance $(\mathcal{S}, c)$, where $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1,2,3\}$, $S_2 = \{2,6,7\}$, $S_3 = \{4,5,6\}$, $S_4 = \{3,5,7\}$, $S_5 = \{7,8,9\}$ and $U = \{1,2,3,4,5,6,7,8,9\}$. Cases for speed $s = 5$ and distance $d = 2, 3$. Illustration of the proof of Lemma 3.2.3.

Lemma 3.2.3. *Given a graph G and an integer $K > 0$, deciding if $gn_{s,d}(G) \leq K$ is NP-hard for every $s, d \geq 0$ such that $d+1 < s < 2d+2$.*

Proof. Reduction from Set Cover. Let $(\mathcal{S}, c)$ be an instance of Set Cover. Recall Definition 3.2.2 and let $p = p(s,d) = d+1$, $q = q(s,d) = 0$, $G = G_{s,d}(\mathcal{S}, c)$ and $K = K_{s,d}(\mathcal{S}, c) = c$.

First, suppose that there is no cover $\mathcal{C}$ of U with at most c sets in $\mathcal{S}$. We prove that the spy wins against at most $K = c$ guards. Precisely, the spy starts in z_0 and can win in one step. Indeed, since there are at most K guards and there is no cover of U with c sets in $\mathcal{S}$, then there exists some $1 \leq i \leq n$ such that there is no guard in $N[U_i]$. Thus, the spy goes to $u_{i,p}$ in one step (note that the distance from z_0 to $u_{i,p}$ is $p + 1 = d + 2 \leq s$). During the guards' step, no guard can reach a vertex of U_i, and so the spy remains at distance at least d from all guards. Therefore, the spy wins.

Now, suppose that there is a cover $\mathcal{C} = \{S_{j_1}, \ldots, S_{j_c}\}$ of U with c sets in $\mathcal{S}$. For ease of presentation, we prove that $c = K$ guards win if they are placed first. This is clearly sufficient to prove that $gn_{s,d}(G) \leq K$ since it is a disadvantage for the guards to place themselves first. The strategy of the guards is as follows. Occupy initially the vertices $S_{j_1}, \ldots, S_{j_c}$. Since $\mathcal{C}$ is a cover of U, we can define for any element $u_i \in U$ an index $c(i)$ such that $u_i \in S_{c(i)} \in \mathcal{C}$.

If the spy is not in $\{u_{1,p}, \ldots, u_{n,p}\}$, then the guards occupy the initial vertices and then they control the spy. If the spy is in a vertex $u_{i,p}$, then the guard occupying $S_{c(i)}$ goes to $u_{i,1}$ and controls the spy. Since $s < 2d + 2$, the spy cannot go from $u_{i,p}$ to other vertex $u_{j,p}$ in one step ($j \neq i$). Thus, if the spy leaves $u_{i,p}$, the guards reoccupy the initial vertices. With this strategy, the guards win the game. □

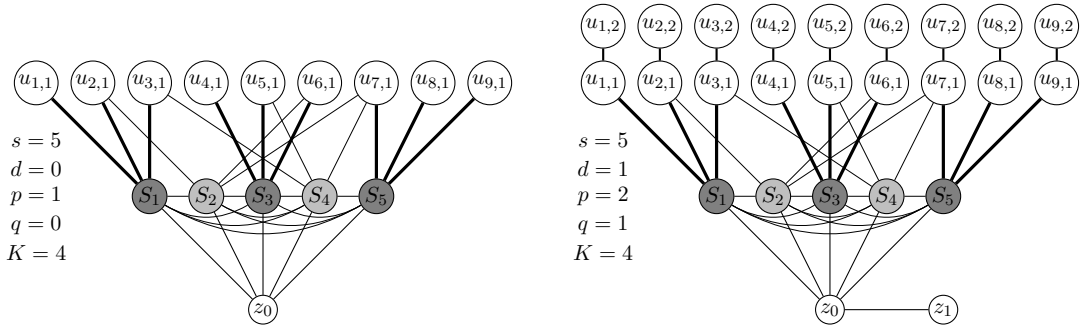


Figure 3.4: Reduction from Set Cover instance $(\mathcal{S}, c)$, where $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ and $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Cases for speed $s = 5$ and distance $d \in \{0, 1\}$. Illustration of the proof of Lemma 3.2.4.

Lemma 3.2.4. *Given a graph G and an integer K, deciding if $gn_{s,d}(G) \leq K$ is NP-hard for every $s, d \geq 0$ such that $s \geq 2d + 2$.*

Proof. Reduction from Set Cover. Let $(\mathcal{S}, c)$ be an instance of Set Cover. Recall Definition 3.2.2 and let $p = p(s, d) = d + 1$, $q = q(s, d) = d$, $G = G_{s,d}(\mathcal{S}, c)$ and $K = K_{s,d}(\mathcal{S}, c) = c + 1$.

First, suppose that there is no cover $\mathcal{C}$ of U with at most c sets in $\mathcal{S}$. We prove that the spy wins against at most $K = c + 1$ guards. Precisely, the spy starts in z_q and can win in one step. Indeed, if initially no guard occupies a vertex in $\{z_0, \ldots, z_q\}$, then the spy wins immediately. Therefore, let us assume that there is at least one guard in $\{z_0, \ldots, z_q\}$. Since there are $c + 1$ guards, then there are at most c guards

outside $\{z_0, \ldots, z_q\}$. Since there is no cover of U with c sets in $\mathcal{S}$, then there exists some $1 \leq i \leq n$ such that there is no guard in $N[U_i]$. Thus, the spy goes to $u_{i,p}$ in one step (note that the distance from z_q to $u_{i,p}$ is $p+q+1 = 2d+2 \leq s$) and wins since no guard can reach a vertex in U_i (*i.e.*, no vertex at distance at most d from $u_{i,p}$) during the next step.

Now, suppose that there is a cover $\mathcal{C} = \{S_{j_1}, \ldots, S_{j_c}\}$ of U with c sets in $\mathcal{S}$. For ease of presentation, we prove that $c+1 = K$ guards win if they are placed first. This is clearly sufficient to prove that $gn_{s,d}(G) \leq K$ since it is a disadvantage for the guards to place themselves first. The strategy of the guards is as follows. Occupy initially the vertices $z_0, S_{j_1}, \ldots, S_{j_c}$. Since $\mathcal{C}$ is a cover of U, we can define, for any element $u_i \in U$, an index $c(i)$ such that $u_i \in S_{c(i)} \in \mathcal{C}$.

If the spy occupies a vertex not in $\{u_{1,p}, \ldots, u_{n,p}\}$, then the guards keep their initial positions and control the spy. If the spy occupies the vertex $u_{i,p}$, then the guard occupying $S_{c(i)}$ goes to $u_{i,1}$ (controlling the spy) and the guard occupying z_0 goes to $S_{c(i)}$. If the spy leaves $u_{i,p}$ and occupies a vertex $u_{j,p}$ with $c(i) = c(j)$, then the guard in $S_{c(i)}$ goes to $u_{j,1}$ (controlling the spy) and the guard in $u_{i,1}$ goes to $S_{c(i)}$. If the spy leaves $u_{i,p}$ and occupies a vertex $u_{j,p}$ with $c(i) \neq c(j)$, then the guard occupying $S_{c(j)}$ goes to $u_{j,1}$ (controlling the spy), the guard in $S_{c(i)}$ goes to $S_{c(j)}$, and the guard in $u_{i,1}$ goes to $S_{c(i)}$. If the spy leaves $u_{i,p}$ to go to some vertex not in $\{u_{1,p}, \ldots, u_{n,p}\}$, then the guards reoccupy the initial vertices: the guard in $S_{c(i)}$ goes to z_0 and the guard in $u_{i,1}$ goes to $S_{c(i)}$. With this strategy, the guards win the game. □

Now consider the case $d+1 \geq s \geq 2$. The next auxiliary lemma will be very useful.

Lemma 3.2.5. *Let $s, d \geq 0$ be two integers such that $d+1 \geq s \geq 2$, let $p = p(s,d) = d + \left\lceil \frac{d+1}{s-1} \right\rceil$, $q = q(s,d) = d + \left\lceil \frac{d}{s-1} \right\rceil$ and $r = d \mod (s-1)$. Note that $p = q+1$ if $r = 0$ and $p = q$ otherwise.*

Let $\ell \in \{p, q\}$, let $P = (x_{-1}, x_0, \cdots, x_\ell)$ be a path and let us consider one guard playing the game in P against a spy with speed s and at distance d.

(a) There is a winning strategy for the guard ensuring that the guard is in x_0 when the spy occupies a vertex in $\{x_{-1}, \cdots, x_r\}$;

(b) If $r > 0$, there are no winning strategies for the guard ensuring that it is in x_0 when the spy is in x_j for $j > r$;

(c) If $\ell = q$, there are no winning strategies for the guard ensuring that it is in x_{-1} when the spy is in x_0.

(d) If $\ell = p$, for every winning strategy for the guard, it must never occupy x_{-1}.

Proof. (a). We first consider the case $\ell = q$. The strategy is defined as follows. If the spy occupies a vertex in $\{x_{-1}, \cdots, x_r\}$, then the guard is at x_0. For any $0 < j \leq \left\lfloor \frac{d}{s-1} \right\rfloor$, if the spy occupies a vertex in $\{x_{r+1+(j-1)s}, \cdots, x_{r+js}\}$, then the guard is at x_j. Note first that the strategy is well-defined: for any move of the spy, the guard either stays idle or moves to a neighbour. Moreover, for any $0 \leq j \leq \left\lfloor \frac{d}{s-1} \right\rfloor$, the distance between them is $r + j(s-1)$. While $j \leq \left\lfloor \frac{d}{s-1} \right\rfloor$, this distance is at most $r + \left\lfloor \frac{d}{s-1} \right\rfloor (s-1) = d$

(by definition of r). It only remains to show that the strategy is defined for all possible positions of the spy. Note that the strategy is well-defined when the spy occupies x_h for all $h \leq r + \left\lfloor \frac{d}{s-1} \right\rfloor s$. If $r = 0$, then $r + \left\lfloor \frac{d}{s-1} \right\rfloor s = d + \left\lceil \frac{d}{s-1} \right\rceil = q = \ell$ and we are done (all positions have been considered). If $r > 0$, then $r + \left\lfloor \frac{d}{s-1} \right\rfloor s = d + \left\lfloor \frac{d}{s-1} \right\rfloor - 1 = q - 1 = \ell - 1$. Therefore, it only remains to define the strategy when the spy is in x_ℓ, in which case, the guard occupies $x_{1+\left\lfloor \frac{d}{s-1} \right\rfloor}$.

Now, let us assume that $\ell = p$. Note that, if $r > 0$, then $p = q$ and therefore, this case has already been treated. Hence, let us consider the case $r = 0$.

The strategy is defined as follows. If the spy is at x_{-1} or x_0, then the guard is at x_0. For any $0 < j \leq \left\lfloor \frac{d}{s-1} \right\rfloor$, if the spy occupies a vertex in $\{x_{(j-1)s+1}, \cdots, x_{js}\}$, then the guard is at x_j. Since $r = 0$, $x_{\left\lfloor \frac{d}{s-1} \right\rfloor s} = x_q = x_{p-1} = x_{\ell-1}$. Therefore, all that remains is to define the position of the guard when the spy occupies x_ℓ, in which case, the guard is at $x_{\left\lfloor \frac{d}{s-1} \right\rfloor + 1}$. Moreover, the distance between the spy and the guard is at most $\ell - \left(\left\lfloor \frac{d}{s-1} \right\rfloor + 1\right) \leq d$.

(b). If $r > 0$ and the spy starts at x_{r+1}, then it goes at full speed toward x_ℓ. After $j = \left\lfloor \frac{d}{s-1} \right\rfloor$ steps, the spy occupies x_h for $h = 1 + r + \left\lfloor \frac{d}{s-1} \right\rfloor s = \ell$ (as shown above when $r > 0$), and the guard can only occupy a vertex in $\{x_{-1}, \cdots, x_j\}$. Therefore, the distance between them is at least $1 + r + \left\lfloor \frac{d}{s-1} \right\rfloor (s-1) = 1 + d$ and the spy wins.

(c). If $r > 0$, the spy first goes to x_{r+1} while the guard can only go to x_0 and the result follows from the previous item. If $r = 0$, then the spy goes at full speed toward x_ℓ. After $j = \left\lfloor \frac{d}{s-1} \right\rfloor$ steps, the spy occupies x_h for $h = \left\lfloor \frac{d}{s-1} \right\rfloor s = \ell$ (as shown in item (a)), and the guard can only occupy a vertex in $\{x_{-1}, \cdots, x_{j-1}\}$. Therefore, the distance between them is at least $1 + \left\lfloor \frac{d}{s-1} \right\rfloor (s-1) = 1 + d$ and the spy wins.

(d). Finally, assume that the spy starts in x_{-1} and goes at full speed to x_ℓ. After $j > 0$ steps, the spy occupies x_{js-1} and the guard occupies x_{j-1}. Therefore, the distance between them is $j(s-1)$ which is at most d if and only if $j \leq \left\lfloor \frac{d}{s-1} \right\rfloor$. Let us set $j_0 = \left\lfloor \frac{d}{s-1} \right\rfloor$ and note that $sj_0 - 1 = s\left\lfloor \frac{d}{s-1} \right\rfloor - 1 = (s-1)\left\lfloor \frac{d}{s-1} \right\rfloor + \left\lfloor \frac{d}{s-1} \right\rfloor - 1 = d - r + \left\lfloor \frac{d}{s-1} \right\rfloor - 1 = d - r + \left\lceil \frac{d+1}{s-1} \right\rceil - 2 = p - 2 - r$. After step j_0, the spy occupies x_{sj_0-1} and is at distance exactly d from the guard. During the step $j_0 + 1$, the spy can progress by at least two edges toward x_p (because $s \geq 2$ and $sj_0 - 1 \leq p - 2$) while the guard can progress of at most one edge. Therefore, the distance between them is at least $d + 1$ and the spy wins. □

Now, let us consider the case when $s \leq d + 1$ and $r = d \mod (s-1) \geq \left\lceil \frac{s}{2} \right\rceil - 1$ or $r = 0$.

Lemma 3.2.6. *Given a graph G and an integer K, deciding if $gn_{s,d}(G) \leq K$ is NP-hard for every $s, d > 0$ such that $2 \leq s \leq d+1$ and $r = d \mod (s-1) \in \{\left\lceil \frac{s}{2} \right\rceil - 1, \ldots, s-2, 0\}$.*

Proof. Reduction from Set Cover. Let $(\mathcal{S}, c)$ be an instance of Set Cover. Recall Definition 3.2.2 and let $p = p(s,d) = d + \left\lceil \frac{d+1}{s-1} \right\rceil$, $q = q(s,d) = d + \left\lceil \frac{d}{s-1} \right\rceil$, $r = d \mod (s-1)$, $G = G_{s,d}(\mathcal{S}, c)$ and $K = K_{s,d}(\mathcal{S}, c) = c + 1$.

First, suppose that there is no cover $\mathcal{C}$ of U with at most c sets in $\mathcal{S}$. We prove that the spy wins against at most $K = c + 1$ guards. Precisely, the spy starts in z_0

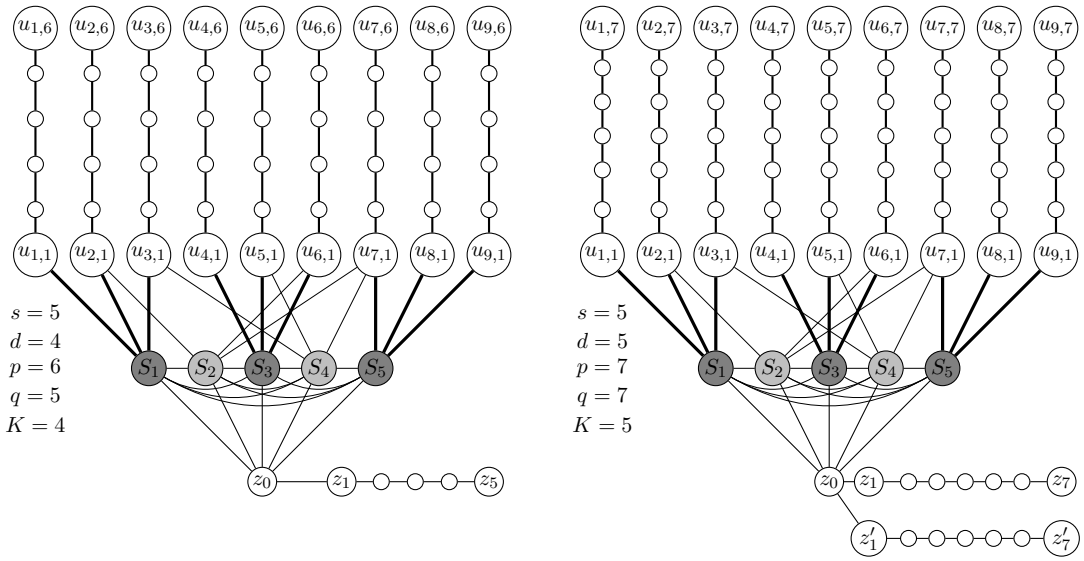


Figure 3.5: Reduction from Set Cover instance $(\mathcal{S}, c)$, where $c = 3$, $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5\}$, $S_1 = \{1, 2, 3\}$, $S_2 = \{2, 6, 7\}$, $S_3 = \{4, 5, 6\}$, $S_4 = \{3, 5, 7\}$, $S_5 = \{7, 8, 9\}$ and $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$. Cases for speed $s = 5$ and distance $d \in \{4, 5\}$. Illustration of the proofs of Lemma 3.2.6 (left) and Lemma 3.2.7 (right).

and can win as follows. If no guards are occupying a vertex in $\{z_0, \ldots, z_q\}$, then by Lemma 3.2.5(c) the spy can move to z_q and win. Therefore, there must be a guard in $\{z_0, \ldots, z_q\}$ and so, at most c guards occupying vertices in $V(G) \setminus \{z_0, \ldots, z_q\}$. Since there is no cover of U with at most c sets in $\mathcal{S}$, then there exists some $1 \leq i \leq n$ such that there is no guard in $N[U_i]$. Thus, the spy goes at full speed s from z_0 to $u_{i,p}$. The conditions are similar to the ones of Lemma 3.2.5(d) where the vertices in $X = N(u_{1,p}) \setminus U_i$ (which are not occupied) play the role of x_0, and the vertices of $N(X) \setminus N[U_i]$ (containing z_0) play the role of x_{-1}. Therefore, the spy eventually wins.

Now, suppose that there is a cover $\mathcal{C} = \{S_{j_1}, \ldots, S_{j_c}\}$ of U with c sets in $\mathcal{S}$. In what follows, we describe a winning strategy for $K = c + 1$ guards. The strategy of the guards will ensure that there is always a guard at every vertex of $\mathcal{C}$. Recall that, since $\mathcal{C}$ is a cover of U, we can define for any element $u_i \in U$ an index $c(i)$ such that $u_i \in S_{c(i)} \in \mathcal{C}$.

The strategy is defined as follows.

- If the spy occupies a vertex in $\{z_0, S_1, \ldots, S_m\}$, then the guards occupy the vertices in $\{z_0, S_{j_1}, \ldots, S_{j_c}\}$.

- If the spy occupies a vertex in U_i for $i \leq n$, let P_i be the path induced by U_i, $S_{c(i)}$ and z_0. Let us apply Lemma 3.2.5(a) on P_i with $\ell = p$, z_0 plays the role of x_{-1} and $S_{c(i)}$ plays the role of x_0. By Lemma 3.2.5(a), there exists a strategy allowing one guard to control the spy and such that the guard occupies $S_{c(i)}$ if the spy occupies a vertex in $\{z_0, S_{c(i)}, u_{i,1}, \cdots, u_{i,r}\}$.

  In that case, one guard, called the *follower*, follows the strategy defined by

Lemma 3.2.5(a). The other guards occupy the vertices in $\{S_{j_1}, \ldots, S_{j_c}\}$ if the follower does not occupy $S_{c(i)}$, and they occupy $\{z_0, S_{j_1}, \ldots, S_{j_c}\} \setminus \{S_{c(i)}\}$ if the follower is at $S_{c(i)}$.

- If the spy occupies a vertex in Z, let Z' be the path induced by Z, z_0 and any vertex S_j. Let us apply Lemma 3.2.5(a) on Z' with $\ell = q$, S_j plays the role of x_{-1} and z_0 plays the role of x_0. By Lemma 3.2.5(a), there exists a strategy allowing one guard to control the spy and such that the guard occupies z_0 if the spy occupies a vertex in $\{z_0, u_{i,1}, \cdots, u_{i,r}\}$ or any vertex S_j.

  In that case, one guard, called the *follower*, follows the strategy defined by Lemma 3.2.5(a). The other guards occupy the vertices in $\{S_{j_1}, \ldots, S_{j_c}\}$.

For any position of the spy, the above strategy ensures that at least one guard controls the spy (by Lemma 3.2.5(a)). Hence, all that remains to be proved is that the strategy is valid, *i.e.*, that, for any move of the spy, the guards can move accordingly. There are several cases to be considered.

- If the spy goes from a vertex in some U_i to another vertex of the same U_i or to a vertex in $\{z_0, S_1, \ldots, S_m\}$, then, the follower moves according to the strategy of Lemma 3.2.5(a). If this move leads the follower to $S_{c(i)}$ (in particular, by the property of the strategy of Lemma 3.2.5(a), it is the case if the spy reaches a vertex in $\{z_0, S_1, \ldots, S_m\}$), then the guard that was occupying $S_{c(i)}$ goes to z_0. Therefore, all guards' moves are valid (if they move, they go to one of their neighbours).

  By symmetry of the strategy (which is positional), the strategy of the guards is also valid if the spy moves from $\{z_0, S_1, \ldots, S_m\}$ to some U_i.

  The case when the spy goes from a vertex of Z to Z, or from Z to $\{z_0, S_1, \ldots, S_m\}$ is similar.

- If the spy goes from a vertex in U_i to a vertex in U_j for some $i \neq j$. Note that, by the property of the strategy of Lemma 3.2.5(a), the follower has to be either in $u_{i,1}$ or in $S_{c(i)}$ after the spy's move (this is because, if the spy is able to go from U_i to U_j, it could also have gone to z_0, and the strategy ensures that, in that case, the follower must be able to reach $S_{c(i)}$).

  If the follower was in $u_{i,1}$ (after the spy's move), then the guard at $S_{c(j)}$ becomes the new follower (recall that all vertices in $\{S_{j_1}, \ldots, S_{j_c}\}$ are always occupied). If the strategy of the follower (in P_j) asks it to move, the new follower moves (in which case, it goes to $u_{j,1}$), then the guard at $u_{i,1}$ goes to $S_{c(i)}$. Finally, if $c(i) \neq c(j)$, the guard that was occupying $S_{c(i)}$ goes to $S_{c(j)}$. If the strategy of the follower is to stay idle, then the guard at $u_{i,1}$ goes to $S_{c(i)}$ and the guard that was at $S_{c(i)}$ goes to z_0.

  Otherwise, the follower was at $S_{c(i)}$, then the guards occupy $\{z_0, S_{j_1}, \ldots, S_{j_c}\}$. In that case, the guard at $S_{c(j)}$ becomes the new follower. If it has to move (to $u_{j,1}$), then the guard at z_0 replaces it at $S_{c(j)}$.

It is important to note that, in all cases, when the spy enters in U_j, the new follower was occupying $S_{c(j)}$ (which plays the role of x_0 in Lemma 3.2.5(a)), and therefore it can apply the strategy described in Lemma 3.2.5(a).

- The last case is when the spy goes from a vertex in U_i to a vertex in Z. If z_0 was occupied by a guard, then it becomes the follower and apply the strategy of Lemma 3.2.5(a)). If z_0 was not occupied, then it means that the guards were occupying the vertices in $\{u_{1,i}, S_{j_1}, \ldots, S_{j_c}\}$. In particular, the follower was occupying $u_{i,1}$ (because, by the property of the strategy of Lemma 3.2.5(a), this guard must be able to go to $S_{c(i)}$ (*i.e.*, x_0) when the spy can reach z_0 (playing the role of x_{-1}). Moreover, if the guard is occupying $u_{i,1}$, it must be because the spy was (before its last move) at $u_{i,h}$ for $h > r$ (otherwise, by the property of the strategy, the guard would be at $S_{c(i)}$).

  There are two cases depending whether $r = 0$ or $r \geq \lceil \frac{s}{2} \rceil - 1$ (the moves are the same, but the reason of their validity is different).

  - If $r = 0$, note that $p = q + 1$. In that case, Lemma 3.2.5(a) can be applied on the path $(u_{1,i}, S_{c(i)}, z_0, \cdots, z_q)$ (playing the role respectively of $(x_{-1}, x_0, x_1, \cdots, x_p)$). Therefore, the guard at $S_{c(i)}$ becomes the follower. It goes to z_0 while the guard at $u_{1,i}$ goes to $S_{c(i)}$.
  - If $r \geq \lceil \frac{s}{2} \rceil - 1$, because the spy was at $u_{i,h}$ for $h > r$, this implies that, after its move, the spy reaches a vertex $z_q \in Z$ for $q \leq r$. In that case, the guard at $S_{c(i)}$ goes to z_0 and becomes the follower (this satisfies the conditions of the strategy of Lemma 3.2.5(a), because $q \leq r$) and the guard at $u_{1,i}$ goes to $S_{c(i)}$.

□

Finally, let us consider the case $s \leq d + 1$ and $1 \leq r = d \mod (s-1) < \frac{s}{2} - 1$. Recall that, in this case, we have added another path Z' to $G_{s,d}(\mathcal{S}, c)$.

Lemma 3.2.7. *Given a graph G and an integer K, deciding if $gn_{s,d}(G) \leq K$ is NP-hard for every $s, d > 0$ such that $s \leq d + 1$ and $1 \leq r = d \mod (s-1) < \frac{s}{2} - 1$.*

Proof. Reduction from Set Cover. Let $(\mathcal{S}, c)$ be an instance of Set Cover. Recall Definition 3.2.2 and let $p = p(s,d) = d + \lceil \frac{d+1}{s-1} \rceil$, $q = q(s,d) = d + \lceil \frac{d}{s-1} \rceil$, $G = G_{s,d}(\mathcal{S}, c)$ and $K = K_{s,d}(\mathcal{S}, c) = c + 2$. Notice that, since $r = d \mod (s-1) \neq 0$, then $p = q$.

Firstly, suppose that there is no cover $\mathcal{C}$ of U with at most c sets in $\mathcal{S}$. We prove that the spy wins against at most $K = c + 2$ guards. Precisely, the spy starts in z'_{r+1} and proceeds as follows. If no guards are occupying a vertex in $\{z'_1, \ldots, z'_q\}$, then, by Lemma 3.2.5(b), the spy can move at full speed to z'_q and win. Moreover, if no guards are occupying a vertex in $\{z_0, \ldots, z_q\}$, then, in one step, the spy goes to z_{r+1} (which is at distance $2r + 2 < s$ by the assumption on r) and, by Lemma 3.2.5(b), the spy will win by moving at full speed to z_q. Therefore, there must be at most c guards at the vertices in $V(G) \setminus \{z_0, z_1, z'_1, \ldots, z_q, z'_q\}$. Since there is no cover of U with c sets in $\mathcal{S}$, then there exists some $1 \leq i \leq n$ such that there is no guard in $N[U_i]$. Thus, in one

step, the spy can go to $u_{i,r+1}$ (at distance $2r+3 \leq s$ by the assumption on r). From Lemma 3.2.5(b), the spy can move to $u_{i,p}$ and wins.

Now, suppose that there is a cover $\mathcal{C} = \{S_{j_1}, \ldots, S_{j_c}\}$ of U with c sets in $\mathcal{S}$. In what follows, we describe a winning strategy for $K = c+2$ guards. Recall that, since $\mathcal{C}$ is a cover of U, we can define for any element $u_i \in U$ an index $c(i)$ such that $u_i \in S_{c(i)} \in \mathcal{C}$. The strategy of the guards will ensure that there is always a guard at every vertex of $\mathcal{C} \cup \{z_0\}$. In addition, the last guard, called the *follower*, follows the strategy described in Lemma 3.2.5(a) in one of the paths U_i, for $1 \leq i \leq n$, Z or Z' depending on the position of the spy.

More precisely, if the spy is occupying a vertex in $\{z_0, S_1, \ldots, S_m\}$, the guards occupy the vertices $z_0, z_0, S_{j_1}, \ldots, S_{j_c}$ (two guards in z_0). When the spy arrives at a vertex in U_i for some $i \leq n$ (resp., in Z or Z'), the guard at $S_{c(i)}$ (resp., at z_0) plays the role of the follower in the corresponding path. The other $c+1$ guards reorganize themselves to occupy the vertices $z_0, S_{j_1}, \ldots, S_{j_c}$.

In particular, when the spy goes from one path U_i (resp., Z, resp., Z') to another path U_j or Z or Z', Lemma 3.2.5(a) ensures that the previous follower was either at $u_{i,1}$ of $S_{c(i)}$ (resp., z_1 or z_0, resp., z'_1 or z_0). Therefore, it is possible for the guards (which are not the new follower) to reorganize themselves to occupy the vertices $z_0, S_{j_1}, \ldots, S_{j_c}$.

The details are similar to the ones provided in the proof of Lemma 3.2.6 and are omitted. □

The question of the complexity of the spy game in undirected graphs is left open. Is it PSPACE-hard, or more probably EXPTIME-complete as Cops and Robber games [87]? The question of parameterized complexity is also open.

3.3 Simple Topologies: Paths and Cycles

In this section, we characterize optimal strategies in the case of paths.

Theorem 3.3.1. *Let $s > 1$ and $d \geq 0$. Let $P = (v_0, \ldots, v_{n-1})$ be any n-node path.*

$$gn_{s,d}(P) = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$$

Proof. Let us set $2d = q(s-1)+r$ where $q = \left\lfloor \frac{2d}{s-1} \right\rfloor$ and $r < s-1$ (note that, if $s > 2d+1$, then $q = 0$ and $r = 2d$). Note also that $2d+2+q = qs+r+2$

Let us first show that the spy can win against at most $\left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$ guards. The spy starts in v_0, so there must be a guard, called Guard 1, at some vertex in $\{v_0, \ldots, v_d\}$ to control the spy. Then, in q steps, the spy goes to v_{qs} while Guard 1 can only reach a vertex in $\{v_0, \ldots, v_{d+q}\}$. Note that the distance between the spy and Guard 1 is then at least $qs - (d+q) = d - r$. During the next step $q+1$, the spy reaches vertex v_{qs+r+2} (note that it is possible since $r+2 \leq s$). Guard 1 can only go to v_{d+q+1} and therefore it is at distance at least $d+1$ from the spy. Therefore, there must be another guard, called Guard 2, occupying a vertex in $\{v_{qs+r+2-d}, \ldots, v_{qs+r+2+d}\}$ to control the

spy. Going on this way, for $0 < j < \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$, after $j(q+1)$ turns, the spy occupies vertex $v_{j(qs+r+2)}$ and there must be a guard, called Guard $j+1$, occupying some vertex in $\{v_{j(qs+r+2)-d}, \ldots, v_{j(qs+r+2)+d}\}$. Moreover, all the j previous guards (Guard 1 to Guard j) are occupying some vertices in $\{v_0, \ldots, v_{j(qs+r+2)-d-1}\}$. In particular, just after $j_0(q+1)$ turns, where $j_0 = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 2$, all the $\left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$ guards are occupying vertices in $\{v_0, \ldots, v_{j_0(qs+r+2)+d}\}$ while the spy is at $v_{j_0(qs+r+2)}$. Therefore, during the next $q+1$ turns, the spy goes to $v_{(j_0+1)(qs+r+2)}$. Note that $(j_0+1)(qs+r+2) = (\left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1)(qs+r+2) = (\left\lceil \frac{n}{2d+2+q} \right\rceil - 1)(2d+2+q) < n$, so the move is possible. During these last $q+1$ steps, all guards can only reach vertices in $\{v_0, \ldots, v_{j_0(qs+r+2)+d+q+1}\}$ and, therefore, are all at distance at least $d+1$ from the spy (indeed, $(j_0+1)(qs+r+2) - (j_0(qs+r+2)+d+q+1) = d+1$). Hence, the spy wins.

Finally, let us describe a winning strategy for $\left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil$ guards. For $0 \leq j < \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$, let $P_j = (v_{j(qs+r+2)}, \ldots, v_{(j+1)(qs+r+2)-1})$. Moreover, for $j_0 = \left\lceil \frac{n}{2d+2+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$, let $P_{j_0} = (v_{j_0(qs+r+2)}, \ldots, v_{n-1})$ (note that $n-1 \leq (j_0+1)(qs+r+2)-1$). The strategy simply uses one guard, called Guard j, for each subpath P_j. Precisely, for any $0 \leq j \leq \left\lceil \frac{n}{2d+1+\left\lfloor \frac{2d}{s-1} \right\rfloor} \right\rceil - 1$,

- for any $0 \leq h < q$, if the spy occupies a vertex in $\{v_{j(qs+r+2)+hs+1}, \ldots, v_{j(qs+r+2)+(h+1)s}\}$, then Guard j occupies $v_{j(qs+r+2)+d+h+1}$;
- if the spy occupies a vertex in $\{v_{j(qs+r+2)+qs+1}, \ldots, v_{j(qs+r+2)+qs+r+1}\}$, then Guard j occupies $v_{j(qs+r+2)+d+q+1}$;
- if the spy occupies $v_{j(qs+r+2)}$ or some subpath P_i, with $i < j$, then Guard j occupies $v_{j(qs+r+2)+d}$;
- finally, if the spy occupies some subpath P_i, with $i > j$, then Guard j occupies $v_{j(qs+r+2)+q+1+d}$.

It can be checked that, following this strategy, the guards always control the spy. Moreover, for any move of the spy, the guards can move according to this strategy. □

Now, we consider the case of cycles. Let us first start with the case $2d < s-1$.

Lemma 3.3.2. *Let $0 \leq 2d < s-1$. For any cycle C_n with n vertices,*

$$gn_{s,d}(C_n) \leq \left\lceil \frac{n}{2d+3} \right\rceil.$$

Proof. Since the number of guards cannot decrease when n increases, we may assume that $\frac{n}{2d+3} = k \in \mathbb{N}$. Let $C_n = (v_0, \ldots, v_{n-1})$. Let us describe a strategy using k guards.

Assume that the spy is initially in v_0. The guards are placed at vertices $v_{d+j(2d+3)}$, for any $0 \leq j < k$. Note that, in particular, the last guard is placed at $v_{d+(k-1)(2d+3)} = v_{n-d-3}$ since $n = (2d+3)k$.

Now, the guards are at distance at most d from all vertices but the vertices $v_{2d+1+j(2d+3)}$ and $v_{2d+2+j(2d+3)}$ for any $0 \leq j < k$. If the spy goes to $v_{2d+1+j(2d+3)}$ for some $0 \leq j < k$, then all guards move clockwise. If the spy goes to $v_{2d+2+j(2d+3)}$ for some $0 \leq j < k$, then all guards move counter-clockwise. Both cases are symmetric to the initial one. In any other case, the guards do not move. Clearly, such a strategy can perpetually ensure that at least one guard controls the spy at distance d. □

Lemma 3.3.3. *Let $2d \geq s - 1 > 0$. For any cycle C_n with n nodes,*

$$gn_{s,d}(C_n) \leq \left\lceil \frac{n + 2\lfloor \frac{2d}{s-1} \rfloor}{2(d + \lfloor \frac{2d}{s-1} \rfloor) + 1} \right\rceil.$$

Proof. Let us set $2d = q(s-1) + r$ where $q = \lfloor \frac{2d}{s-1} \rfloor$, and let $X = 2(d + \lfloor \frac{2d}{s-1} \rfloor) + 1 = q(s+1) + r + 1$. Note that $q > 0$ since $s \leq 2d + 1$ and $d > 0$.

Since the number of guards cannot decrease when n increases, we may assume that $\frac{n+2\lfloor \frac{2d}{s-1} \rfloor}{2(d+\lfloor \frac{2d}{s-1} \rfloor)+1} = \frac{n+2q}{X} = k \in \mathbb{N}$. Let us describe a strategy using k guards.

Let v_0 be the initial position of the spy, and the cycle is $(v_0, \ldots, v_{n-1})$. The guards are placed at vertices v_{d+jX}, for any $0 \leq j < k$. Let us call the guard at v_{d+jX} as the Guard j, for any $0 \leq j < k$. Note that, in particular, the Guard $k-1$ is placed at $v_{d+(k-1)X} = v_{n-d-1}$ since $n - 2d - 1 = (k-1)X$.

We first consider the case where the spy moves clockwise at the beginning. After, we show that the spy moving counter-clockwise at the beginning results in a case symmetric to the spy moving clockwise at the beginning.

- If the spy goes from v_0 to any vertex in $\{v_0, \ldots, v_{\lfloor s/2 \rfloor}\}$, no guards move. Note that Guard 0 still controls the spy since $s \leq 2d + 1$.

- If the spy goes from a vertex in $\{v_0, \ldots, v_{\lfloor s/2 \rfloor}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor + 1}, \ldots, v_{\lfloor s/2 \rfloor + s}\}$, then Guard 0 also goes clockwise to v_{d+1}. All other guards go counter-clockwise to v_{d+jX-1}, for every $0 < j < k$. The spy is still controlled due to the following.

  Guard 0, who is at v_{d+1}, is within distance at most d of $v_{\lfloor s/2 \rfloor + 1 + y}$ for $0 \leq y < 2d - \lfloor s/2 \rfloor + 1$ since $\lfloor s/2 \rfloor + 1 + y < 2d + 2 \Leftrightarrow y < 2d - \lfloor s/2 \rfloor + 1$ and $\lfloor s/2 \rfloor + 1 \geq 2$ since $s \geq 2$.

  Guard 1, who is at v_{3d+2q}, is not within distance at most d of $\lfloor s/2 \rfloor + 1 + y$ for $0 \leq y < 2d + 2q - 1 - \lfloor s/2 \rfloor$ since $\lfloor s/2 \rfloor + 1 + y < 2d + 2q \Leftrightarrow y < 2d + 2q - 1 - \lfloor s/2 \rfloor$.

  Note that $\lfloor s/2 \rfloor + s > 4d + 2q \Leftrightarrow s > 4d + 2q - \lfloor s/2 \rfloor$, which is not possible since $s \leq 2d + 1$. Therefore, if $2q - 1 \leq 1 \Leftrightarrow q \leq 1 \Leftrightarrow q = 1$ since $q \neq 0$, then the spy

is always controlled when he goes from a vertex in $\{v_0, \ldots, v_{\lfloor s/2 \rfloor}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor+1}, \ldots, v_{\lfloor s/2 \rfloor+s}\}$.

If $q \neq 1$, then $v_{2d-\lfloor s/2 \rfloor+1+1+\lfloor s/2 \rfloor} = v_{2d+2}$ is the first (closest to $v_{\lfloor s/2 \rfloor+1}$ in the clockwise direction) vertex not controlled by Guard 0 or Guard 1. But v_{2d+2} is not in $\{v_{\lfloor s/2 \rfloor+1}, \ldots, v_{\lfloor s/2 \rfloor+s}\}$ since $q > 1 \Leftrightarrow \lfloor \frac{2d}{s-1} \rfloor \geq 2 \Leftrightarrow 2d+2 \geq 2s$ and $2s > \lfloor s/2 \rfloor + s$. Therefore, all the vertices in $\{v_{\lfloor s/2 \rfloor+1}, \ldots, v_{\lfloor s/2 \rfloor+s}\}$ are controlled by Guard 0 or Guard 1.

- For $0 < h < q$, when the spy goes from a vertex in $\{v_{\lfloor s/2 \rfloor+(h-1)s+1}, \ldots, v_{\lfloor s/2 \rfloor+hs}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor+hs+1}, \ldots, v_{\lfloor s/2 \rfloor+(h+1)s}\}$, then Guard 0 also goes clockwise to v_{d+h+1}. All other guards go counter-clockwise to $v_{d+jX-h-1}$, for every $0 < j < k$. The spy is still controlled due to the following.

  Guard 0, who is at v_{d+h+1}, is within distance at most d of $v_{\lfloor s/2 \rfloor+hs+1+y}$ for $0 \leq y < 2d+h-hs-\lfloor s/2 \rfloor+2$ since $\lfloor s/2 \rfloor+hs+1+y < 2d+h+2 \Leftrightarrow y < 2d+h-hs-\lfloor s/2 \rfloor+2$ and $\lfloor s/2 \rfloor + hs + 1 > h \Leftrightarrow h(1-s) < \lfloor s/2 \rfloor + 1 \Leftrightarrow h > \frac{\lfloor s/2 \rfloor+1}{1-s}$ since $\frac{\lfloor s/2 \rfloor+1}{1-s} < 0$ because $s \geq 2$.

  Guard 1, who is at $v_{3d+2q-h}$, is not within distance at most d of $\lfloor s/2 \rfloor + hs + 1 + y$ for $0 \leq y < 2d - h(1+s) + 2q - 1 - \lfloor s/2 \rfloor$ since $\lfloor s/2 \rfloor + hs + 1 + y < 2d + 2q - h \Leftrightarrow y < 2d - h(1+s) + 2q - 1 - \lfloor s/2 \rfloor$.

  Note that $\lfloor s/2 \rfloor + (h+1)s > 4d + 2q - h \Leftrightarrow h(s+1) > 4d + 2q - s - \lfloor s/2 \rfloor$. Since $h < q$, $(q-1)(s+1) \geq h(s+1) > 4d + 2q - s - \lfloor s/2 \rfloor$ and $(q-1)(s+1) > 4d+2q-s-\lfloor s/2 \rfloor \Leftrightarrow qs-s-1 > 4d+q-s-\lfloor s/2 \rfloor \Leftrightarrow q(s-1) > 4d-\lfloor s/2 \rfloor+1 \Leftrightarrow 2d - r > 4d + 1 - \lfloor s/2 \rfloor \Leftrightarrow r < \lfloor s/2 \rfloor - 2d - 1$. But this is not possible since $\lfloor s/2 \rfloor - 2d - 1 < 0$ since $s \leq 2d+1 \Rightarrow \lfloor s/2 \rfloor \leq d$ and r cannot be negative. Therefore, if $2q-1-h(1+s) \leq h(1-s)+2 \Leftrightarrow 2q \leq 2h+3 \Leftrightarrow q \leq h+3/2 \Leftrightarrow h \geq q-2$ since q is an integer, then the spy is always controlled when he goes from a vertex in $\{v_{\lfloor s/2 \rfloor+(h-1)s+1}, \ldots, v_{\lfloor s/2 \rfloor+hs}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor+hs+1}, \ldots, v_{\lfloor s/2 \rfloor+(h+1)s}\}$.

  If $h \not\geq q-2$, then $v_{2d+h(1-s)-\lfloor s/2 \rfloor+2+hs+1+\lfloor s/2 \rfloor} = v_{2d+h+3}$ is the first (closest to $v_{\lfloor s/2 \rfloor+hs+1}$ in the clockwise direction) vertex not controlled by Guard 0 or Guard 1. But v_{2d+h+3} is not in $\{v_{\lfloor s/2 \rfloor+hs+1}, \ldots, v_{\lfloor s/2 \rfloor+(h+1)s}\}$ since $q \geq h+2 \Leftrightarrow \lfloor \frac{2d}{s-1} \rfloor \geq h+2 \Leftrightarrow 2d \geq hs+2s-h-2 \Leftrightarrow 2d+h+3 \geq hs+2s+1$ and clearly $hs+2s+1 > hs+s+\lfloor s/2 \rfloor$. Therefore, all the vertices in $\{v_{\lfloor s/2 \rfloor+hs+1}, \ldots, v_{\lfloor s/2 \rfloor+(h+1)s}\}$ are controlled by Guard 0 or Guard 1.

- For $1 \leq h \leq q$, when the spy goes from a vertex in $\{v_{\lfloor s/2 \rfloor+(h-1)s+1}, \ldots, v_{\lfloor s/2 \rfloor+hs}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor+(h-1)s+1}, \ldots, v_{\lfloor s/2 \rfloor+hs}\}$, no guards move. The proof that the spy is still controlled in the previous case also proves that he is controlled here.

Now, we consider the case of the spy moving counter-clockwise at the start. If the spy goes from v_0 to any vertex in $\{v_0, \ldots, v_{n-\lfloor s/2 \rfloor-1}\}$, no guards move. Note that Guard $k-1$ controls the spy since $s \leq 2d+1$ and the positions of the guards and the spy are symmetrical to the case where the spy had gone clockwise at the start to a vertex in $\{v_0, \ldots, v_{\lfloor s/2 \rfloor}\}$.

If the spy moves from a vertex in $\{v_0, \ldots, v_{n-\lfloor s/2 \rfloor -1}\}$ to a vertex in $\{v_{n-\lfloor s/2 \rfloor -2}, \ldots, v_{n-\lfloor s/2 \rfloor -s-1}\}$, then Guard $k-1$ goes counter-clockwise to v_{n-d-2}. All other guards go clockwise to v_{d+jX+1}. Note that if v_{n-1} is replaced by v_0 and the cycle is relabeled accordingly (one rotation counter-clockwise of the labels), then the positions of the guards and the spy are symmetric to the case where the spy goes clockwise at the start from a vertex in $\{v_0, \ldots, v_{\lfloor s/2 \rfloor}\}$ to a vertex in $\{v_{\lfloor s/2 \rfloor +1}, \ldots, v_{\lfloor s/2 \rfloor +s}\}$.

Seeing as the first move of the spy counter-clockwise results in a symmetric case to the spy's first move clockwise, we only have to consider when the spy moves clockwise.

The following remarks show that the rules above fully describe the strategy of k guards. That is, the behaviour of the guards according to any spy's move can be derived from the rules above by symmetry.

First, all previous moves are reversible. For instance, if the spy goes from $\{v_{\lfloor s/2 \rfloor +1+hs}, \ldots, v_{\lfloor s/2 \rfloor +(h+1)s}\}$ to $\{v_{\lfloor s/2 \rfloor +1+(h-1)s}, \ldots, v_{\lfloor s/2 \rfloor +hs}\}$ (for $1 \leq h < q$), then Guard 0 goes back to v_{d+h}, and all other guards go back to v_{d+jX-h}, for every $0 < j < k$.

Second, let us consider the configuration when the spy arrives in $\{v_{qs+r}, \ldots, v_{\lfloor s/2 \rfloor +qs+r}\}$. At this step, for any $0 < j < k$, Guard j is occupying $v_{d+jX-q} = v_{d+(j-1)X+X-q}$. Since $X - q = q(s+1) + r + 1 - q = qs + r + 1$, this means that, for any $0 < j < k$, Guard j is occupying vertex $v_{qs+r+1+d+(j-1)X}$. Moreover, Guard 0 is occupying vertex $v_{d+q} = v_{qs+r-d}$. Therefore, the situation is symmetric to the initial one up to a rotation and reversal of the labeling of the cycle (where v_{qs+r} replaces v_0). □

Lemma 3.3.4. *Let $s > 1$ and $d \geq 0$. Let $C_n = (v_0, \ldots, v_{n-1})$ be any n-node cycle.*

$$gn_{s,d}(C_n) \geq \left\lceil \frac{n + 2\left\lfloor \frac{2d}{s-1} \right\rfloor}{2(d + \left\lfloor \frac{2d}{s-1} \right\rfloor) + 3} \right\rceil$$

Proof. Let us set $2d = q(s-1) + r$ where $q = \left\lfloor \frac{2d}{s-1} \right\rfloor$ and $r < s-1$ (note that, if $s > 2d+1$, then $q = 0$ and $r = 2d$). Note also that $2d + 2 + q = qs + r + 2$. All integers below must be understood modulo n.

Let us show that the spy can win against a team of $X < \left\lceil \frac{n+2q}{2(d+q)+3} \right\rceil$ guards. If the spy starts in v_0, there must be a guard, called Guard 1, at some vertex in $\{v_{n-d}, \ldots, v_0, \ldots, v_d\}$ to control the spy. Since the spy's speed is greater than the guards' speed, the spy can move clockwise so that he reaches a vertex that is distance $d+2$ from Guard 1 in a finite number of turns (before the guards'turn). Thus, after the guard turn, we may set v_0 (up to renaming the vertices) to be the new position of the spy and so, Guard 1 is at a vertex in $\{v_{n-d-3}, v_{n-d-2}, v_{n-d-1}\}$.

Since Guard 1 is at distance at least $d+1$ from the spy, there must be another guard, called Guard 2, occupying a vertex in $\{v_{n-d}, \ldots, v_0, \ldots, v_d\}$ to control the spy. The spy goes at full speed clockwise and Guard 1 may go at full speed counterclockwise.

Then, after step q, the spy occupies v_{qs} while Guard 2 occupies a vertex in $\{v_{qs-d}, \ldots, v_{d+q}\}$. During the next step (Step $q+1$) the spy goes to v_{qs+r+2} (note that it is possible since $r + 2 \leq s$). In this case, Guard 2 can only go to a vertex in $\{v_{qs-d-1}, \ldots, v_{d+q+1}\}$ and therefore it is at distance at least $d+1$ from the spy and cannot control it anymore.

Therefore, there must be another guard, called Guard 3, occupying a vertex in $\{v_{qs+r+2-d}, \ldots, v_{qs+r+2+d}\}$ to control the spy. Going on this way after $(X-1)(q+1)$ steps, the spy is at $v_{(X-1)(qs+r+2)} = v_\alpha$ while there are X guards occupying vertices in $\{v_{n-(d+3+(X-1)(q+1))}, \ldots, v_{(X-2)(qs+r+2)+d+q+1}\} = \{v_\beta, \ldots, v_\gamma\}$.

Note that $\alpha - \gamma = qs + r + 2 - d - q - 1 = 2d + 2 + q - d - q - 1 = d + 1$. Therefore, the distance between the spy and v_γ is at least $d+1$ and the spy can only be controlled from a guard in v_β. The distance between v_β and v_α is:

$n - (d + 3 + (X-1)(q+1) + (X-1)(qs+r+2)) = n + d + 2q - X(2q + 2d + 3)$.

Moreover, $n + d + 2q - X(2q + 2d + 3) > d$ if and only if $\frac{n+2q}{2(d+q)+3} > X$.

Therefore, the distance between v_α and v_β is at least $d+1$ for $X < \left\lceil \frac{n+2q}{2(d+q)+3} \right\rceil$ since X is an integer and thus, no guard controls the spy which wins. □

The above lemmas can be summarized with the following theorem.

Theorem 3.3.5. *Let $s > 1$ and $d \geq 0$ be two integers. For any cycle C_n with n nodes,*

- $gn_{s,d}(C_n) = \left\lceil \frac{n}{2d+3} \right\rceil$ *if* $0 \leq 2d < s - 1$;
- *If* $2d \geq s - 1$, *then*

$$\left\lceil \frac{n + 2\lfloor \frac{2d}{s-1} \rfloor}{2(d + \lfloor \frac{2d}{s-1} \rfloor) + 3} \right\rceil \leq gn_{s,d}(C_n) \leq \left\lceil \frac{n + 2\lfloor \frac{2d}{s-1} \rfloor}{2(d + \lfloor \frac{2d}{s-1} \rfloor) + 1} \right\rceil.$$

Before continuing with the study of the spy game in trees and grids, some particular strategies must be defined, which is done in the following section.

3.4 Toward Fractional Games

A strategy for the guards is a function describing the moves of the guards in each round. A strategy is *winning* if it allows the guards to perpetually control the spy. Note that there is always an optimal winning strategy (using the minimum number of guards) which is *positional*, *i.e.*, such that the next move is only determined by the current position of both the spy and the guards, and not by the history of the game. This is because the spy game is a parity game (by considering the directed graph of configurations) and parity games always admit positional strategies for the winning player [93].

In other words, there is always an optimal winning strategy which is a function that takes the current positions of the spy and of the guards and returns the new positions of the guards (and so, their moves).

Representation of (fractional) guards' strategies. Let $G = (V, E)$ be an n-node graph, $s \geq 2$ and $d \geq 0$ be two integers. Let $V = \{v_1, \ldots, v_n\}$. A winning strategy σ using $k \in \mathbb{R}^+$ guards is defined as a set $\sigma = \{\mathcal{C}_v\}_{v \in V}$ of sets of configurations. That is, for any $v \in V$ (a possible position for the spy), $\mathcal{C}_v$ is a non-empty set of functions, called *configurations*, that represent the possible positions of the guards when the spy

is at v. More precisely, any $\omega \in \mathcal{C}_v$ is a function $\omega : V \to \mathbb{R}^+$, where $\omega(u) \in \mathbb{R}^+$ represents the amount of guards at vertex $u \in V$ when the spy occupies v, that must satisfy $\sum_{u \in V} \omega(u) = k$ and $\sum_{u \in N_d[v]} \omega(u) \geq 1$. Finally, for any $v \in V$, any $\omega \in \mathcal{C}_v$, and any $v' \in N_s[v]$, there must exist $\omega' \in \mathcal{C}_{v'}$ such that the guards can go from ω to ω' in one round. That is, for any possible move of the spy (from v to v'), there must exist a valid flow from ω to ω' (the guards must be able to reach a configuration controlling the spy in v'). A strategy is *integral* if $k \in \mathbb{N}^+$, each of its configurations is a function $V \to \mathbb{N}$, and every move is an integral flow. The *size* of a strategy is the number of different configurations necessary to describe the strategy, *i.e.*, the size of σ is $\sum_{v \in V} |\mathcal{C}_v|$. Note that, a single position for the spy may correspond to many different positions of the guards. Therefore, the size of an integral strategy using k guards in an n-node graph is $n^{O(k)}$. Moreover, the size of a fractional strategy is *a priori* unbounded.

Spy-positional strategies. In this chapter, we will also consider more constrained strategies. A winning strategy is said to be *spy-positional* if it depends only on the position of the spy. That is, in a spy-positional strategy $\sigma = \{\mathcal{C}_v\}_{v \in V}$, the positions of the guards are only determined by the position of the spy. In particular, every time the spy occupies some vertex v, the set of vertices occupied by the guards is defined by a unique function $\sigma_v : V(G) \to \mathbb{N}$ such that, for every $u \in V$, $\sigma_v(u)$ is the number of guards occupying u when the spy is occupying v. That is, $\mathcal{C}_v = \{\sigma_v\}$ and $|\mathcal{C}_v| = 1$ for every $v \in V$. An important consequence for our purpose is that any (fractional or integral) spy-positional strategy has size $O(n)$.

Let us remark that, in a spy-positional strategy, it is not required that the same guards occupy the same vertices when the spy is at some vertex. That is, assume that, at the end of some round, the spy occupies some vertex v, some Guard A occupies a vertex a and a guard B occupies a vertex b. It may happen that, after some rounds and at the end of such a round, the spy goes back to v and now Guard A is at b and Guard B is at a (however, the set of vertices occupied by the guards is the same).

Second, there does not always exist an optimal strategy (using the minimum number of guards) that is spy-positional. As an example, consider the cycle C_5 with 5 vertices $\{a, b, c, d, e\}$. It is easy to show that $gn_{2,1}(C_5) = 1$. However, consider the following execution. It is easy to see that we may assume that, initially, the guard occupies a neighbour of the spy. W.l.o.g., the spy starts at b while the guard is at a. Then, the spy goes to c and the guard has to go to b. The spy goes to d and the guard has to go to c. Finally, the spy goes back to b and the guard either stays at c or goes to b. Hence, every spy-positional strategy in C_5 needs 2 guards. One of our main results is to show that, in trees, there always exists an optimal strategy which is spy-positional.

Let $fgn^*_{s,d}(G)$ be the minimum total amount of fractional guards needed to always control at distance d a spy with speed s in a graph G, when the guards are constrained to play spy-positional strategies. By definition, for any graph G and any $s \geq 2$, $d \geq 0$,

$$fgn_{s,d}(G) \leq \min\{fgn^*_{s,d}(G), gn_{s,d}(G)\}.$$

Now that fractional spy-positional strategies have been defined, we present a polynomial-time algorithm that computes optimal spy-positional fractional strategies in general graphs. Here, optimal means using the minimum total amount of guards with the

extra constraint that guards are restricted to play spy-positional strategies. In other words, we prove that, for any graph G, $s \geq 2$, and $d \geq 0$, $fgn^*_{s,d}(G)$ and a corresponding strategy can be computed in polynomial time. We prove this result by describing a Linear Program with polynomial size that computes such strategies.

In Section 3.5, we will show that in any tree T, $gn_{s,d}(T) = fgn^*_{s,d}(T)$. More precisely, we will show that in trees, the Linear Program below can be used to compute optimal (integral) strategies in polynomial time.

We describe a Linear Program for computing an optimal fractional spy-positional strategy.

Variables. Let $G = (V, E)$ be a connected n-node graph. Recall that a spy-positional strategy is defined by, for each position of the spy, the amount of guards that must occupy each vertex. Therefore, for any two vertices $u, v \in V$, let $\sigma_v(u) \in \mathbb{R}^+$ be the non-negative real variable representing the amount of guards occupying vertex u when the spy is at v.

Moreover, for any $x \in V$, $y \in N_s[x]$ and for any $u \in V$ and $v \in N[u]$, let $f_{x,y,u,v} \in \mathbb{R}^+$ be the non-negative real variable representing the amount of guards going from vertex u to $v \in N[u]$ when the spy goes from x to $y \in N_s[x]$. Finally, a variable k will represent the total amount of guards. Overall, there are $O((|E|+n+1)n^2) = O(n^4)$ real variables.

These variables fully describe a strategy since σ encodes a distribution of guards for every position of the spy and f describes a feasible transition between two successive distributions.

Objective function. We aim at minimizing the total amount of guards.

$$\texttt{Minimize } k. \tag{3.1}$$

Constraints.

The first family of constraints states that, for every position $v \in V$ of the spy, the total amount of guards is at most k.

$$\forall v \in V, \quad \sum_{u \in V} \sigma_v(u) \leq k. \tag{3.2}$$

The second family of constraints states that, for every position $v \in V$ of the spy, the amount of guards at distance at most d from the spy is at least 1, *i.e.*, the guards always control the spy at distance d.

$$\forall v \in V, \quad \sum_{u \in N_d[v]} \sigma_v(u) \geq 1. \tag{3.3}$$

The third family of constraints states that, for any move of the spy (from x to $y \in N_s[x]$), the corresponding moves of the guards ensure that the amount of guards leaving a vertex $v \in V$ plus what remains at v equals the amount of guards that was at v before the move.

$$\forall x \in V\,,\ y \in N_s[x]\,,\ v \in V\,, \quad \sum_{w \in N[v]} f_{x,y,v,w} = \sigma_x(v). \tag{3.4}$$

The fourth family of constraints states that, for any move of the spy (from x to $y \in N_s[x]$), the corresponding moves of the guards ensure that the amount of guards

that are at a vertex $w \in V$ after the moves equals the amount of guards arriving in w plus what remains at w.

$$\forall x \in V\,,\ y \in N_s[x]\,,\ w \in V\,,\ \sum_{v \in N[w]} f_{x,y,v,w} = \sigma_y(w). \tag{3.5}$$

Finally, the last family of constraints expresses the definition domain of the variables:

$$k \geq 0 \tag{3.6}$$

$$\forall u, v \in V\,,\ \sigma_u(v) \geq 0 \tag{3.7}$$

$$\forall x \in V\,,\ y \in N_s[x]\,,\ v \in V\,,\ w \in N[v]\,,\ f_{x,y,v,w} \geq 0 \tag{3.8}$$

There are $O(n^4)$ constraints and the above Linear Program has polynomial size and clearly computes an optimal spy-positional fractional strategy. Hence:

Theorem 3.4.1. *For any connected graph G, and any two integers $s \geq 2$ and $d \geq 0$, the above Linear Program computes $fgn^*_{s,d}(G)$ and a corresponding spy-positional strategy in polynomial time.*

3.5 Spy game is Polynomial in Trees

This section is devoted to the study of the spy game in trees (Theorem 3.5.11). Before going into the details, we would like to emphasize one difficulty when dealing with guards' strategies.

A natural idea would be to partition the tree into smaller subtrees (with bounded diameter) with a constant number of guards assigned to each of them. That is, each guard would be assigned (possibly with other guards) a subtree S and would move only when the spy is in S (in particular, the guard would only occupy some vertices of S). As already mentioned, there exist such strategies that are optimal in paths or in trees when $d = 0$ and s is large (Eternal Domination) [89]. We show that we cannot expect such strategies for the spy game (for any $s \geq 2$ and $d > 0$) in trees and hence, optimal guards' strategies seem difficult to be described in trees. We present an example in the case $s = 2$ and $d = 1$ but it can be generalized to any $s \geq 2$ and $d > 0$ (by increasing the branches of the star S defined below).

Lemma 3.5.1. *Let $s = 2$ and $d = 1$. There exists a family of trees with unbounded guard number such that, for each of these trees, there is a strategy of the spy that forces every guard to occupy each non-leaf vertex infinitely often, whatever be the optimal strategy followed by the guards.*

Proof. Let S be the tree obtained from a star with three leaves by subdividing each edge exactly twice (*i.e.*, S has 10 vertices). Let $(S_i)_{i \leq k}$ be k disjoint copies of S and let c_i be the unique vertex of degree 3 of S_i. Finally, let T be the tree obtained by adding one vertex c and making it adjacent to every c_i, $i \leq k$. Note that $|V(T)| = 10k+1 = n$.

First, let us show that $gn_{2,1}(T) = k + 1 = \Theta(n)$ and that, when the spy is in c, the guards have to occupy the vertices $c, c_1, \ldots, c_k$. We label the vertices as follows where $1 \leq j \leq 3k$: let v_{3j} be a leaf, let v_{3j-1} be the vertex adjacent to the leaf v_{3j}, and let v_{3j-2} be the vertex adjacent to v_{3j-1} and c_i for $i = \lceil \frac{j}{3} \rceil$ (see Figure 3.6).

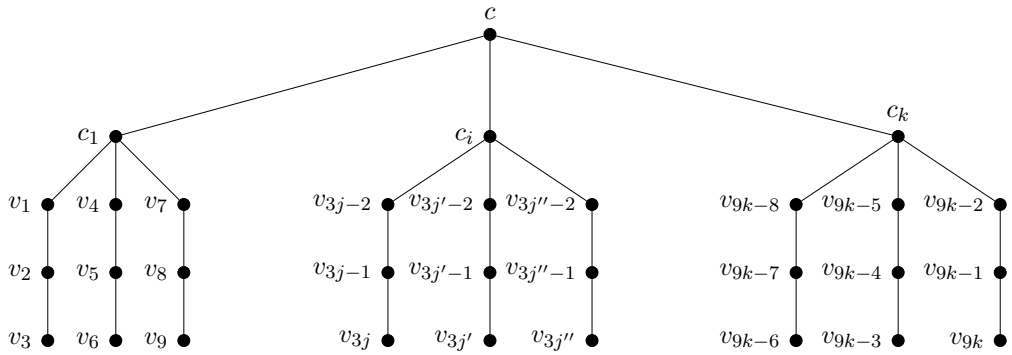


Figure 3.6: Scheme of the tree T (in the proof of Lemma 3.5.1) where only the subtrees S_1, S_i and S_k have been depicted (for some i such that $2 \leq i < k$). In the subtree S_i, $\{j, j', j''\} = \{3i-2, 3i-1, 3i\}$, *i.e.*, $i = \lceil \frac{z}{3} \rceil$ for every $z \in \{j, j', j''\}$.

- A strategy using $k+1$ guards proceeds as follows. In any round, the vertices $c_1, \ldots, c_k$ are occupied (not necessarily by the same guards). Then, if the spy occupies c or one of the c_i's, one guard occupies c. If the spy occupies v_{3j-1} or v_{3j-2} for some $j \leq 3k$, then v_{3j-2} must be occupied by a guard. Finally, if the spy occupies v_{3j}, then v_{3j-1} must be occupied by a guard. It is easy to see that, whatever be the strategy of the spy, the guards may move (at most 2 guards per round) as to ensure the desired positions.

- Now we prove the lower bound. We now suppose the game is being played with at most k guards. The spy starts at a leaf v_{3j} for some $1 \leq j \leq 3k$ and moves at full speed to another leaf v_{3l} for some $1 \leq l \leq 3k$ such that $l \neq j$. Then, there must be a guard at either v_{3j} or v_{3j-1} initially and when the spy reaches v_{3j-2}, there must be at least one other guard at a vertex in the same subtree S_i for $i = \lceil \frac{j}{3} \rceil$ as otherwise, the spy could move to one of $\{v_{3(j+1)}, v_{3(j+2)}\}$ and win. Since there are two guards in the same subtree as the spy, then the spy moves to c. Neither of the two previous guards can reach any of the other $k-1$ vertices c_i in this round. There must be at least one guard in each of the other $k-1$ subtrees S_i as otherwise, the spy moves to a leaf in one of these subtrees and wins since it would take him two rounds but a guard at c could only be at distance at least 2 from the spy after two rounds. It also follows that, when the spy is at c, the guards must occupy the vertices $c, c_1, \ldots, c_k$. Therefore, the spy wins against k guards.

Now, we can prove the main statement of the lemma. For any $i \leq k$ and any vertex $v \in V(T)$ of degree two, there is a strategy of the spy that brings the guard initially at c_i to v and thus, to any non-leaf vertex. For this purpose, let $j \neq i$ be such that $v \in V(S_j)$. The spy first goes (at full speed) to a leaf of S_i, then to another leaf of S_i, then it goes to a leaf of S_j that is not the neighbour of v and finally the spy goes to the leaf neighbour of v. It can be verified that the guard that was initially at c_i must occupy v. Repeating infinitely this strategy (for any v and i) gives the strategy announced in the statement. □

To overcome this difficulty, we use the power of Linear Programming. Precisely, we prove that, in any tree T and for any $s \geq 2$, $d \geq 0$, $gn_{s,d}(T) = fgn^*_{s,d}(T)$. Therefore, using the Linear Program of Section 3.4, it follows that computing $gn_{s,d}(T)$ can be done in polynomial time in trees. The proof is twofold. First, we prove that $gn_{s,d}(T) = fgn_{s,d}(T)$ for any $s \geq 2$ and $d \geq 0$ (*i.e.*, the integrality gap is null in trees), and then that $fgn_{s,d}(T) = fgn^*_{s,d}(T)$.

Theorem 3.5.2. *For any tree T and for any $s \geq 2$, $d \geq 0$, $gn_{s,d}(T) = fgn_{s,d}(T)$. More precisely, any fractional winning strategy using a total amount of $k \in \mathbb{R}^+$ guards can be transformed into an integral winning strategy using $\lfloor k \rfloor$ guards. Moreover, such a transformation can be done in polynomial time in the size of the fractional strategy.*

Proof. Let $\sigma = \{\mathcal{C}_v\}_{v \in V}$ be any fractional winning strategy using a total amount of $k \in \mathbb{R}^+$ guards to control a spy with speed $s \geq 2$, at distance $d \geq 0$, and in an n-node tree $T = (V, E)$.

We build a winning integral strategy σ^r using $\lfloor k \rfloor$ guards by "rounding" all configurations of σ. For any configuration ω of σ, we will define an integral configuration ω^r (which we call a *rounding* of ω) using $\lfloor k \rfloor$ guards (Claim 3.5.3), such that if the spy is controlled in ω then it is also controlled in ω^r (Claim 3.5.4). Moreover, for any two configurations ω_1 and ω_2 such that there is a feasible flow from ω_1 to ω_2, we show that there is feasible integral flow from ω_1^r to ω_2^r (Claim 3.5.5). Altogether, this shows that σ^r is a winning integral strategy using $\lfloor k \rfloor$ guards, which proves the theorem.

From now on, let us consider T to be rooted at some vertex $r \in V$.

Notations. For any $u \in V$, let T_u be the subtree of T rooted in u (*i.e.*, the subtree that consists of u and all its descendants) and let $Children(u)$ be the set of children of u. For any configuration $\omega : V \to \mathbb{R}^+$, let $\omega(T_u) = \sum_{v \in V(T_u)} \omega(v)$ and let $\omega(T) = \omega(T_r)$. By definition, $\omega(T_u) \geq \omega(u)$ for every $u \in V$. Finally, let $cont(T, \omega) = \{u \in V : \sum_{v \in N_d[u]} \omega(v) \geq 1\}$ (*i.e.*, $cont(T, \omega)$ is the set of vertices u such that the spy on u is controlled at distance d by the guards in the configuration ω).

Let us define the *rounded* configuration $\omega^r : V \mapsto \mathbb{N}$ as, for every $u \in V$,

$$\omega^r(u) = \left\lfloor \omega(u) + \sum_{v \in Children(u)} (\omega(T_v) - \lfloor \omega(T_v) \rfloor) \right\rfloor$$

Intuitively, the fractional part of guards that are in each of the subtrees rooted in the children of u is "pushed" to u. Then, u "keeps" only the integral part of the sum of what it had plus what it received from its children.

We first prove that rounding a configuration using k guards provides an integral configuration using $\lfloor k \rfloor$ guards.

Claim 3.5.3. *For any configuration $\omega : V(T) \to \mathbb{R}^+$, $\omega^r(T) = \lfloor \omega(T) \rfloor$.*

Proof of the claim. The proof is by induction on $|V|$. It clearly holds if $|V| = 1$. Let

$T_1, \ldots, T_h$ be the subtrees of $T \backslash r$ rooted in the children of r. By definition,

$$\begin{aligned}\omega^r(T) &= \omega^r(r) + \sum_{1 \le i \le h} \omega^r(T_i) \\ &= \lfloor \omega(r) + \sum_{1 \le i \le h} (\omega(T_i) - \lfloor \omega(T_i) \rfloor) \rfloor + \sum_{1 \le i \le h} \omega^r(T_i)\end{aligned}$$

By induction, $w^r(T_i) = \lfloor \omega(T_i) \rfloor$ for every $1 \le i \le h$, and so:

$$\begin{aligned}\omega^r(T) &= \lfloor \omega(r) + \sum_{1 \le i \le h} (\omega(T_i) - \lfloor \omega(T_i) \rfloor) \rfloor + \sum_{1 \le i \le h} \lfloor \omega(T_i) \rfloor \\ &= \lfloor \omega(r) + \sum_{1 \le i \le h} \omega(T_i) \rfloor \\ &= \lfloor \omega(T) \rfloor\end{aligned}$$

◇

Then, Claim 3.5.4 proves that every position of the spy that is controlled by the guards in a configuration ω is also controlled by the guards in the configuration ω^r.

Claim 3.5.4. *For any configuration $\omega : V(T) \to \mathbb{R}^+$, $cont(T, \omega) \subseteq cont(T, \omega^r)$.*

Proof of the claim. Let $u \in cont(T, \omega)$. By definition, $\sum_{v \in N_d[u]} \omega(v) \ge 1$. Let r' be the vertex in $N_d[u]$ that is closest to the root r, and let T' be the subtree of T rooted in r'. Finally, let $T'_1, \ldots, T'_h$ be the subtrees of $T' \setminus N_d[u]$. By Claim 3.5.3, $\omega^r(T') = \lfloor \omega(T') \rfloor$ and $\omega^r(T'_i) = \lfloor \omega(T'_i) \rfloor$ for any $1 \le i \le h$. Hence,

$$\omega^r(T') = \sum_{v \in N_d[u]} \omega^r(v) + \sum_{1 \le i \le h} \omega^r(T'_i) = \sum_{v \in N_d[u]} \omega^r(v) + \sum_{1 \le i \le h} \lfloor \omega(T'_i) \rfloor$$

and,

$$\omega^r(T') = \lfloor \omega(T') \rfloor = \left\lfloor \sum_{v \in N_d[u]} \omega(v) + \sum_{1 \le i \le h} \omega(T'_i) \right\rfloor$$

Since $\sum_{v \in N_d[u]} \omega(v) \ge 1$, it follows that

$$\left\lfloor \sum_{v \in N_d[u]} \omega(v) + \sum_{1 \le i \le h} \omega(T'_i) \right\rfloor \ge 1 + \left\lfloor \sum_{1 \le i \le h} \omega(T'_i) \right\rfloor \ge 1 + \sum_{1 \le i \le h} \lfloor \omega(T'_i) \rfloor$$

Altogether, $1 + \sum_{1 \le i \le h} \lfloor \omega(T'_i) \rfloor \le \omega^r(T') = \sum_{v \in N_d[u]} \omega^r(v) + \sum_{1 \le i \le h} \lfloor \omega(T'_i) \rfloor$. Therefore, $\sum_{v \in N_d[u]} \omega^r(v) \ge 1$ and $u \in cont(T, \omega^r)$. ◇

Finally, Claim 3.5.5 shows that the moves that were valid in σ still hold in the "rounded" strategy.

Claim 3.5.5. *Let $\omega_1, \omega_2 : V(T) \mapsto \mathbb{R}^+$ be two configurations such that the guards can go from ω_1 to ω_2 in one round (there is feasible flow from ω_1 to ω_2). Then, the guards can go from ω_1^r to ω_2^r in one round (there is feasible integral flow from ω_1^r to ω_2^r).*

Proof of the claim. The proof is by induction on k, the result being trivial when $k = 0$ (note that $k = \omega_1^r(T) = \omega_2^r(T)$). Let f be the flow representing the move of the guards from ω_1 to ω_2. Clearly, we may assume that, $\forall u, v \in V$, at most one of $f(u, v)$ and $f(v, u)$ is non-null.

Among all vertices $v \in V$ such that $\omega_1(T_v) \geq 1$ or $\omega_2(T_v) \geq 1$, let x be a lowest one (such a vertex furthest from the root). By symmetry (there is a feasible flow from ω_1 to ω_2 if and only if there is a feasible flow from ω_2 to ω_1), up to exchanging ω_1 and ω_2, we may assume that $\omega_1(T_x) \geq 1$. Note that, by the minimality of x, for every descendant $u \in V(T_x) \setminus \{x\}$ of x, $\omega_1(T_u) < 1$ and $\omega_2(T_u) < 1$.

Now, let γ_1 be the function defined by $\gamma_1(x) = \omega_1(T_x) - 1$, $\gamma_1(u) = 0$ for every descendant u of x, and $\gamma_1(v) = \omega_1(v)$ for every $v \in V \setminus V(T_x)$. Note that $\gamma_1^r(v) = \omega_1^r(v)$ for every $v \in V \setminus \{x\}$ and $\gamma_1^r(x) = \omega_1^r(x) - 1$. Now, there are two cases to be considered.

- First, assume that $\omega_2(T_x) \geq 1$. In this case, let γ_2 be the function defined by $\gamma_2(x) = \omega_2(T_x) - 1$, $\gamma_2(u) = 0$ for every descendant u of x, and $\gamma_2(v) = \omega_2(v)$ for every $v \in V \setminus V(T_x)$. Note that there is a feasible flow f' from γ_1 to γ_2: for any $u, v \in V(T_x)$, $f'(u, v) = 0$ and for any $u \in V$, $v \in V \setminus V(T_x)$, $f'(u, v) = f(u, v)$. Note also that $\gamma_2^r(v) = \omega_2^r(v)$ for every $v \in V \setminus \{x\}$ and $\gamma_2^r(x) = \omega_2^r(x) - 1$.

  By induction (since $\gamma_1^r(T) = \gamma_2^r(T) = \omega_1^r(T) - 1$), there is a feasible integral flow f^* from γ_1^r to γ_2^r. Since ω_1^r (resp., ω_2^r) is obtained from γ_1^r (resp., γ_2^r) by adding 1 guard in x, this flow f^* is also a feasible integral flow from ω_1^r to ω_2^r.

- Second, $\omega_2(T_x) < 1$. Let p be the parent of x (x cannot be the root since $\omega_2^r(T) \geq 1$). Note that, because there is flow from ω_1 to ω_2, then $\omega_2(p) + \omega_2(T_x) \geq \omega_1(T_x) \geq 1$.

  In this case, let γ_2 be the function defined by $\gamma_2(u) = 0$ for every $u \in V(T_x)$, $\gamma_2(v) = \omega_2(v)$ for every $v \in V \setminus (V(T_x) \cup \{p\})$ and $\gamma_2(p) = \omega_2(p) + \omega_2(T_x) - 1 \geq 0$.

  Note that there is a feasible flow f' from γ_1 to γ_2: for any $u, v \in V(T_x)$, $f'(u, v) = 0$, for any $u, v \in V \setminus V(T_x)$, $f'(u, v) = f(u, v)$, and $f'(x, p) = \gamma_1(x)$. Note also that $\gamma_2^r(v) = \omega_2^r(v)$ for every $v \in V \setminus \{p\}$ and $\gamma_2^r(p) = \omega_2^r(p) - 1$.

  By induction (since $\gamma_1^r(T) = \gamma_2^r(T) = \omega_1^r(T) - 1$), there is a feasible integral flow f^* from γ_1^r to γ_2^r. Since ω_1^r (resp., ω_2^r) is obtained from γ_1^r (resp., γ_2^r) by adding 1 guard in x (resp., in p), there is a feasible integral flow from ω_1^r to ω_2^r that can be obtained from f^* by adding to it one unit of flow from x to p.

◇

This concludes the proof of Theorem 3.5.2. □

The second step in this section is to show that there is always an optimal fractional strategy which is spy-positional. For this purpose, we prove the following theorem.

Theorem 3.5.6. *For any tree T and for any $s \geq 2$, $d \geq 0$, $fgn^*_{s,d}(T) = fgn_{s,d}(T)$. More precisely, any fractional winning strategy using a total amount of $k \in \mathbb{R}^+$ guards can be transformed into a spy-positional winning strategy using k guards.*

Proof. Let $\sigma = \{\mathcal{C}_v\}_{v \in V}$ be any fractional winning strategy using a total amount of $k \in \mathbb{R}^+$ guards to control a spy with speed $s \geq 2$, at distance $d \geq 0$, and in an n-node tree $T = (V, E)$. Recall that, for any vertex $v \in V$, $\mathcal{C}_v$ is the set of possible configurations $\omega : V \to \mathbb{R}^+$ for the guards when the spy is at v.

The proof consists of defining a spy-positional strategy σ^{min} that is a winning strategy using k guards. For any $v \in V$, we will define the function $\omega_v^{min} : V \to \mathbb{R}^+$ to be the (unique) configuration of σ^{min} when the spy is at v, *i.e.*, $\sigma^{min} = \{\omega_v^{min}\}_{v \in V}$. We first prove that σ^{min} is a strategy using k guards (Claims 3.5.7-3.5.8), then that the spy at $v \in V$ is controlled at distance d by the guards in the configuration ω_v^{min} (Claim 3.5.9). Finally, we prove that, for any move of the spy from v to $v' \in V$, the guards can move from ω_v^{min} to $\omega_{v'}^{min}$ (Claim 3.5.10).

From now on, T is rooted in an arbitrary vertex $r \in V$.

Notations. For any weight function $\omega : V \to \mathbb{R}^+$, let $\omega^+ : V \to \mathbb{R}^+$ be the *cumulative* function of ω, defined by, for every $u \in V$, $\omega^+(u) = \sum_{v \in V(T_u)} \omega(v) = \omega(T_u)$. Let $v \in V$ and $\mathcal{C}_v = \{\omega_1, \ldots, \omega_h\} \in \sigma$ be the set of configurations of the guards, when the spy is in v. Let $\alpha_v : V \to \mathbb{R}^+$ be such that, for every $u \in V$, $\alpha_v(u) = \min_{1 \leq i \leq h} \omega_i^+(u)$. Now, ω_v^{min} is defined as the (unique) function such that α_v is its cumulative function, *i.e.*, $\alpha_v = (\omega_v^{min})^+$. Formally, for every $u \in V$: $\omega_v^{min}(u) = \alpha_v(u) - \sum_{x \in Children(u)} \alpha_v(x)$.

Claim 3.5.7 proves that, for every $v \in V$, $\omega_v^{min} : V \to \mathbb{R}^+$ is a configuration.

Claim 3.5.7. *For every $u \in V$, $\omega_v^{min}(u) \geq 0$.*

Proof of the claim. Let $1 \leq i \leq h$ be an integer such that $\alpha_v(u) = \min_{1 \leq j \leq h} \omega_j^+(u) = \omega_i^+(u)$. By definition of α_v, for every $x \in Children(u)$, $\alpha_v(x) = \min_{1 \leq j \leq h} \omega_j^+(x) \leq \omega_i^+(x)$. Hence, $\omega_v^{min}(u) \geq \omega_i^+(u) - \sum_{x \in Children(u)} \omega_i^+(x) = \omega_i(u) \geq 0$. ◇

Claim 3.5.8 proves that, for every $v \in V$, the configuration ω_v^{min} uses k guards.

Claim 3.5.8. *For every $v \in V$, $\sum_{u \in V} \omega_v^{min}(u) = k$.*

Proof of the claim. For every $1 \leq i \leq h$, $\omega_i^+(r) = k$. Hence, $\alpha_v(r) = \min_{1 \leq i \leq h} \omega_i^+(r) = k$. $\sum_{u \in V} \omega_v^{min}(u) = (\omega_v^{min})^+(r) = \alpha_v(r) = k$ (since α_v is the cumulative function of ω_v^{min}). ◇

Claim 3.5.9 proves that the guards in the configuration ω_v^{min} control a spy located at v.

Claim 3.5.9. *For every $v \in V$, $\sum_{u \in N_d[v]} \omega_v^{min}(u) \geq 1$.*

Proof of the claim. Let v^* be the vertex of $N_d[v]$ that is closest to the root r. Let $v_1, \ldots, v_p$ be the descendants of v^* that are at distance exactly $d+1$ from v. Since α_v is the cumulative function of ω_v^{min}, we have that $\sum_{u \in N_d[v]} \omega_v^{min}(u) = \alpha_v(v^*) - \sum_{1 \leq j \leq p} \alpha_v(v_j)$. Let $1 \leq i \leq h$ be an integer such that $\alpha_v(v^*) = \min_{1 \leq j \leq h} \omega_j^+(v^*) = \omega_i^+(v^*)$. Since the guards in configuration ω_i control the spy in v at distance d, we have

that $\sum_{u\in N_d[v]}\omega_i(u)=\omega_i^+(v^*)-\sum_{1\le j\le p}\omega_i^+(v_j)\ge 1$. Hence, $\sum_{u\in N_d[v]}\omega_v^{min}(u)=\alpha_v(v^*)-\sum_{1\le j\le p}\alpha_v(v_j)=\omega_i^+(v^*)-\sum_{1\le j\le p}\min_{1\le j'\le h}\omega_{j'}^+(v_j)\ge\omega_i^+(v^*)-\sum_{1\le j\le p}\omega_i^+(v_j)\ge 1$. $\diamond$

Finally, Claim 3.5.10 shows that the moves that were valid in σ still hold for σ^{min}.

Claim 3.5.10. *For every $v\in V$ and $v'\in N_s[v]$, there is a feasible flow from ω_v^{min} to $\omega_{v'}^{min}$.*

Proof of the claim. Let $\mathcal{C}_v=\{\omega_1,\ldots,\omega_h\}\in\sigma$ (the configurations of σ when the spy is at v) and $\mathcal{C}_{v'}=\{\omega'_1,\ldots,\omega'_{h'}\}\in\sigma$ (the configurations of σ when the spy is at v'). Since σ is a winning strategy, it means that, for every $1\le i\le h$, there is $1\le\delta(i)\le h'$, such that there is a feasible flow from $\omega_i\in\mathcal{C}_v$ to $\omega'_{\delta(i)}\in\mathcal{C}_{v'}$. That is, there is a function $f^i: V\times V\to\mathbb{R}^+$ such that, for every $u\in V$, $\omega'_{\delta(i)}(u)=\omega_i(u)+\sum_{w\in N(u)}(f^i(w,u)-f^i(u,w))$ and $\sum_{w\in N(u)}f^i(u,w)\le\omega_i(u)$. Note that, such a function f^i can be defined as, for every $u\in V$ and $p\in V$, the parent of u in T rooted in r (if $u\neq r$), $f^i(u,p)=\max\{\omega_i^+(u)-(\omega'_{\delta(i)})^+(u),0\}$ and $f^i(p,u)=\max\{(\omega'_{\delta(i)})^+(u)-\omega_i^+(u),0\}$.

Let $u\in V$, $X\subseteq Children(u)$ be any subset of the children of u, and $1\le i\le h$. Because of the existence of the flow f^i, $\sum_{w\in X}(\omega'_{\delta(i)})^+(w)\le\omega_i(u)+\sum_{w\in X}\omega_i^+(w)$, hence:

$$\begin{aligned}\omega_i^+(u)&=\omega_i(u)+\sum_{w\in X}\omega_i^+(w)+\sum_{w\in Children(u)\setminus X}\omega_i^+(w)\\&\ge\sum_{w\in X}(\omega'_{\delta(i)})^+(w)+\sum_{w\in Children(u)\setminus X}\omega_i^+(w).\end{aligned}$$

So, since for every $w\in V$, $\alpha_{v'}(w)=\min_{1\le j\le h'}(\omega'_j)^+(w)$ and $\alpha_v(w)=\min_{1\le j\le h}\omega_j^+(w)$:

$$\omega_i^+(u)\ge\sum_{w\in X}\alpha_{v'}(w)+\sum_{w\in Children(u)\setminus X}\alpha_v(w)$$

The above inequality holds for every $1\le i\le h$. Since $\alpha_v(u)=\min_{1\le i\le h}\omega_i^+(u)$, it follows that:

$$\alpha_v(u)\ge\sum_{w\in X}\alpha_{v'}(w)+\sum_{w\in Children(u)\setminus X}\alpha_v(w)$$

By similar arguments (because, by symmetry, there is a flow from ω'_j to some $\omega_{j'}$ for every $1\le j\le h'$), we get

$$\alpha_{v'}(u)\ge\sum_{w\in X}\alpha_{v'}(w)+\sum_{w\in Children(u)\setminus X}\alpha_v(w)$$

We need to prove that there exists a function $f: V\times V\to\mathbb{R}^+$ such that, for every $u\in V$, $\omega_{v'}^{min}(u)=\omega_v^{min}(u)+\sum_{w\in N(u)}(f(w,u)-f(u,w))$ and $\sum_{w\in N(u)}f(u,w)\le\omega_v^{min}(u)$.

For every $u \in V$, let $p \in V$ be the parent of u in T rooted in r (if $u \neq r$). Let $f^{min}(u,p) = \max\{\alpha_v(u) - \alpha_{v'}(u), 0\}$ and let $f^{min}(p,u) = \max\{\alpha_{v'}(u) - \alpha_v(u), 0\}$.

By definition, f preserves the amount of guards in every subtree. Consequently, it also preserves it at every node and so for every $u \in V$, $\omega^{min}_{v'}(u) = \omega^{min}_v(u) + \sum_{w \in N(u)}(f^{min}(w,u) - f^{min}(u,w))$.

Therefore, we only need to prove that $\sum_{w \in N(u)} f^{min}(u,w) \leq \omega^{min}_v(u)$.

Let $u \in V$, p its parent (if $u \neq r$), and let $X \subseteq Children(u)$ be the set of vertices such that, for every $w \in X$, $f^{min}(u,w) = \alpha_{v'}(w) - \alpha_v(w) > 0$. There are two cases to be considered.

- First, let us assume that $f^{min}(u,p) = 0$.

$$\begin{aligned}
\omega^{min}_v(u) &= \alpha_v(u) - \sum_{w \in Children(u)} \alpha_v(w) \\
&= (\alpha_v(u) - \sum_{w \in Children(u) \setminus X} \alpha_v(w)) - \sum_{w \in X} \alpha_v(w) \\
&\geq \sum_{w \in X} (\alpha_{v'}(w) - \alpha_v(w)) = \sum_{w \in N(u)} f^{min}(u,w)
\end{aligned}$$

- Second, assume that $f^{min}(u,p) = \delta > 0$.

$$\begin{aligned}
\omega^{min}_v(u) &= \alpha_v(u) - \sum_{w \in Children(u)} \alpha_v(w) \\
&= \alpha_{v'}(u) + \delta - \sum_{w \in Children(u)} \alpha_v(w) \\
&= \delta + (\alpha_{v'}(u) - \sum_{w \in Children(u) \setminus X} \alpha_v(w)) - \sum_{w \in X} \alpha_v(w) \\
&\geq \delta + \sum_{w \in X} (\alpha_{v'}(w) - \alpha_v(w)) = \sum_{w \in N(u)} f^{min}(u,w)
\end{aligned}$$

$\diamond$

This concludes the proof of Theorem 3.5.6. $\square$

We can now prove the main theorem of this section.

Theorem 3.5.11. *Let $s \geq 2$ and $d \geq 0$ be two integers. There is a polynomial-time algorithm that computes an integral winning strategy using $gn_{s,d}(T)$ guards to control a spy with speed s at distance d in any tree T.*

Proof. By Theorem 3.5.6, there exists an optimal (fractional) winning strategy that is spy-positional for T. By Theorem 3.4.1, such a strategy can be computed in polynomial time. From that strategy, Theorem 3.5.2 can compute in polynomial time an optimal integral winning strategy for T. $\square$

3.6 Spy game in Grid and Torus

It is clear that, for any $n \times n$ grid G, $gn_{s,d}(G) \leq |V(G)| = O(n^2)$. However, the exact order of magnitude of $gn_{s,d}(G)$ is not known. In this section, we prove that there exists $\beta^* > 0$, such that $\Omega(n^{1+\beta^*})$ guards are necessary to win against one spy in an $n \times n$-grid. Our lower bound actually holds for the fractional relaxation of the game. Precisely, we prove that $fgn_{s,d}(G)$ is super-linear and sub-quadratic (in the side n).

Let $n, m \geq 2$ be two integers. We consider the $n \times m$ toroidal grid $T_{n\times m} = (V, E)$, *i.e.*, the graph with vertices $v_{i,j} = (i, j)$ and edges $\{(i, j), (i+1 \mod n, j)\}$ and $\{(i, j), (i, j+1 \mod m)\}$, for all $0 \leq i < n$ and $0 \leq j < m$. The $n \times m$ grid $G_{n\times m}$ is obtained from $T_{n\times m}$ by removing the edges $\{\{(i, m-1), (i, 0)\}; \{(n-1, j), (0, j)\} \mid \forall 0 \leq i < n, 0 \leq j < m\}$.

First, we show that the number of fractional (resp., integral) guards required in the grid and in the torus have the same order of magnitude. Precisely:

Lemma 3.6.1. *For all $n, m \geq 2$, $s \geq 2$, $d \geq 0$, and for all $f \in \{gn_{s,d}, fgn_{s,d}, fgn^*_{s,d}\}$:*

$$f(T_{n\times m})/4 \leq f(G_{n\times m}) \leq 4 \cdot f(T_{n\times m}).$$

Proof. Let us present the proof in the integral case, *i.e.*, when $f = gn_{s,d}$, the other two cases are similar.

Let σ be a winning strategy using k guards in $T_{n\times m}$. We define a winning strategy using $4k$ guards in $G_{n\times m}$. For this purpose, let us label the guards used by σ as $G_1, \ldots, G_k$. In $G_{n\times m}$, the behavior of Guard G_i ($1 \leq i \leq k$) is "simulated" by four guards as follows. The guard G_i being at $(x, y) \in V(T_{n\times m})$ is simulated by one guard at each of the four vertices: (x, y), $(n-1-x, y)$, $(x, m-1-y)$ and $(n-1-x, m-1-y)$. That is, one of the four guards occupies (x, y) while the other three guards occupy its images with respect to the horizontal, vertical, and diagonal axes. Hence, $gn_{s,d}(G_{n\times m}) \leq 4 \cdot gn_{s,d}(T_{n\times m})$.

Let σ be a winning strategy using k guards in $G_{n\times m}$. We define a winning strategy using $4k$ guards in $T_{n\times m}$. Our strategy actually allows to control four spies whose moves are correlated. Precisely, assume that when one spy occupies vertex (x, y), the three other spies occupy respectively $(n-1-x, y)$, $(x, m-1-y)$, and $(n-1-x, m-1-y)$. We divide the $4k$ guards into four teams, each of which uses the strategy σ (*i.e.*, they all act as if they were in the grid) to control one of the four spies. When some spies cross an edge of $E(T_{n\times m}) \setminus E(G_{n\times m})$, some teams will exchange their target. Hence, $gn_{s,d}(T_{n\times m}) \leq 4 \cdot gn_{s,d}(G_{n\times m})$. □

The first of the two main results of this section is:

Theorem 3.6.2. *There exists $\beta^* > 0$ such that, for every $s \geq 2$, $d \geq 0$,*

$$fgn_{s,d}(G_{n\times n}) = \Omega(n^{1+\beta^*}).$$

Corollary 3.6.3. *There exists $\beta^* > 0$ such that, for every $s \geq 2$, $d \geq 0$, $gn_{s,d}(G_{n\times n}) = \Omega(n^{1+\beta^*})$.*

Section 3.6.1 is devoted to prove Theorem 3.6.2, and Section 3.6.2 will be devoted to prove the second main result of this section which is that $fgn_{s,d}(G_{n\times n})$ and $fgn_{s,d}(T_{n\times n})$ are sub-quadratic (in the side n) when s is a constant.

3.6.1 Lower bound in Grids

The goal of this section is to prove that there exists $\beta^* > 0$ such that $fgn_{s,d}(G_{n\times n}) = \Omega(n^{1+\beta^*})$, *i.e.*, the number of guards required in any $n \times n$-grid is super-linear in the side n of the grid. We first make the following observation, which we will need for the proof of Theorem 3.6.13:

Observation 3.6.4. *In the class of $n\times n$ grids, $gn_{s,d}$ and $fgn_{s,d}$ are increasing functions of n.*

Proof. For every $n' \geq n$, any strategy played by the guards on the $n' \times n'$ grid can be transformed into a strategy on the $n \times n$ grid. Indeed, it is sufficient to project any quantity of guards located on a vertex $v \in G_{n'\times n'}$ onto the vertex of $G_{n\times n}$ that is the closest to it, with respect to the Hamming distance (*i.e.*, L^1 distance). Note that this vertex is unique, and that this transformation is compatible with moves between two consecutive positions. □

To prove this section's lower bound on grids, let us define (yet) another parameter. For any $s \geq 2$, $d \geq 0$, $t \geq 0$, $q \geq 1$, and any graph G (note that t may be a function of $|V(G)|$), let $gn^{q,t}_{s,d}(G)$ be the minimum number k of guards such that there is an integral strategy using k guards that ensures that at least q guards are at distance at most d from a spy with speed s during at least the first t rounds. Note that, by definition, $\sup_t gn^{1,t}_{s,d}(G) \leq gn_{s,d}(G)$.

The first step of the proof is that $gn^{q,2n}_{s,d}(G_{n\times n}) = \Omega(q \cdot n \log n)$ in any $n \times n$-grid and then we "extend" this result to the fractional strategies. The latter result will then be used as a "bootstrap" in the induction proof for the main result. Let $H : \mathbb{R}^+ \to \mathbb{R}^+$, $H(x) = \sum_{1\leq i\leq x} 1/i$ for every $x \in \mathbb{R}^+$.

Lemma 3.6.5. *$\exists \beta \geq 1/48$ such that for any $n \in \mathbb{N}, s \geq 2, d \geq 0$ (possibly d depends on n), $q > 0$,*

$$gn^{q,2n}_{s,d}(G_{n\times n}) \geq \beta \cdot q \cdot \frac{n}{d+1} H(\frac{n}{d+1}).$$

Proof. The proof is for $s = 2$ since $gn^{q,2n}_{s,d}(G_{n\times n}) \geq gn^{q,2n}_{2,d}(G_{n\times n})$.

In order to prove the result, we will consider a *family* of strategies for the spy. For every $0 \leq r < n$, the spy starts at position $(0, 0)$ and runs at full speed toward $(r, 0)$. Once there, it continues at full speed toward $(r, n-1)$. We name P_r the strategy defined by that path (which is completed in $\lceil \frac{1}{2}(r+n-1) \rceil$ rounds) and sometimes use the same notation to denote the path itself. Note that the guards may be aware of the family of strategies played by the spy but do not know r in advance.

Let us assume that there exists a strategy using an amount k of guards that maintains at least q guards at distance at most d from the spy during at least $2n$ rounds. Moreover, the spy only plays the strategies described above.

Assuming that the guards are labeled with integers in $\{1, \ldots, k\}$, we can name at any time of strategy P_r the labels of q guards that are at distance at most d from the spy. In this way, let $c(i, j)$ denote this set of q guards that are at distance at most d from the spy, when the spy is at position (i, j). Observe that this definition does not depend on the choice of r, since all strategies of our family that eventually reach position (i, j) are indistinguishable up to that round. Precisely, for any $i \leq n$, when the spy reaches vertex $(i, 0)$, it must be playing some strategy P_r, for $r \geq i$, but the guards do not know which one, and so $c(i, 0)$ is independent of $r \geq i$. Moreover, for any $j > 0$ and $r \leq n$, the spy reaches vertex (r, j) only when it is playing the strategy P_r. Hence, there is no ambiguity.

Claim 3.6.6. *Let $j_1, j_2 \in \mathbb{N}^*$ such that $2j_1 < n$ and $2j_2 < n$. If $|j_2 - j_1| > 2d$, then $c(r, 2j_1)$ and $c(r, 2j_2)$ are disjoint.*

Proof of the claim. Assuming $j_1 < j_2$, it takes $j_2 - j_1$ rounds for the spy in strategy P_r to go from $(r, 2j_1)$ to $(r, 2j_2)$. A guard cannot be at distance at most d from $(r, 2j_1)$ and, $j_2 - j_1$ rounds later, at distance at most d from $(r, 2j_2)$. Indeed, to do so its speed must be at least $2(j_2 - j_1 - d)/(j_2 - j_1) > 1$, a contradiction. ◇

Claim 3.6.7. *Let $j_1, j_2, r_1, r_2 \in \mathbb{N}^*$ such that $2z < n$ for every $z \in \{j_1, j_2, r_1, r_2\}$. If $|r_2 - r_1| > 2d + 2\min(j_1, j_2)$, then $c(2r_1, 2j_1)$ and $c(2r_2, 2j_2)$ are disjoint.*

Proof of the claim. Assuming $r_1 < r_2$, note that strategies P_{2r_1} and P_{2r_2} are identical for the first r_1 rounds. The spy is at position $(2r_1, 0)$ when they split. If $c(2r_1, 2j_1)$ intersects $c(2r_2, 2j_2)$ on a given guard, it means that at this instant (*i.e.*, at round r_1) that guard is simultaneously at distance at most $d + j_1$ from $(2r_1, 2j_1)$[†] and at distance at most $d + |r_2 - r_1| + j_2$ from $(2r_2, 2j_2)$[‡] since the guard belongs to both sets. As those two points are at distance $2|r_2 - r_1| + 2|j_2 - j_1|$ from each other, we have:

$$\begin{aligned} 2|r_2 - r_1| + 2|j_2 - j_1| &\leq (d + j_1) + (d + |r_2 - r_1| + j_2) \\ |r_2 - r_1| + 2|j_2 - j_1| &\leq 2d + j_1 + j_2 \\ |r_2 - r_1| &\leq 2d + 2\min(j_1, j_2) \end{aligned}$$

◇

We can now proceed to prove that the number of guards is sufficiently large. To do so, we define a graph $\mathcal{H}$ on a subset of $V(G_{n \times n})$ and relate the distribution of the guards (as described by c) with the independent sets of $\mathcal{H}$ by treating every guard as a *colour*. Intuitively, an independent set I in $\mathcal{H}$ will consist of a set of sets $c(i, j)$ of guards that must be pairwise disjoint. It is defined over $V(\mathcal{H}) = \{(2r, 4dj) : 0 \leq 2r < n, 0 \leq 4dj < n\}$, where:

- $(2r, 4dj_1)$ is adjacent to $(2r, 4dj_2)$ for $j_1 \neq j_2$ (see Claim 3.6.6).
- $(2r_1, 4dj_1)$ is adjacent to $(2r_2, 4dj_2)$ if $|r_2 - r_1| > 4d(1 + \min(j_1, j_2))$ (see Claim 3.6.7).

[†]In strategy P_{2r_1}, that guard must be at distance $\leq d$ from $(2r_1, 2j_1)$ when the spy visits it.
[‡]Similarly, for strategy P_{2r_2}.

By definition, c gives q colours (represented by the q guards) to each vertex of $\mathcal{H}$ and any set of vertices of $\mathcal{H}$ receiving a common colour is an independent set of $\mathcal{H}$. If we denote by $\#c^{-1}(x)$ the number of vertices which received colour x, and by $\alpha_{(2r,4dj)}(\mathcal{H})$ the maximum size of an independent set of $\mathcal{H}$ containing $(2r, 4dj)$, we have:

$$\begin{aligned} k &= \sum_{x\in\{1,\dots,k\}} 1 = \sum_{x\in\{1,\dots,k\}} \frac{\#c^{-1}(x)}{\#c^{-1}(x)} = \sum_{v\in V(H)} \sum_{x\in c(v)} \frac{1}{\#c^{-1}(x)} \\ &= \sum_{(2r,4dj)\in V(H)} \sum_{x\in c(2r,4dj)} \frac{1}{\#c^{-1}(x)} \\ &\geq \sum_{(2r,4dj)\in V(H)} \frac{q}{\alpha_{(2r,4dj)}(\mathcal{H})} \end{aligned}$$

It is easy, however, to approximate this lower bound.

Claim 3.6.8. $\alpha_{(2r,4dj)}(\mathcal{H}) \leq 4d(j+1)+1$

Proof of the claim. An independent set $S \subseteq V(\mathcal{H})$ containing $(2r, 4dj)$ cannot contain two vertices with the same first coordinate. Furthermore, $(2r, 4dj)$ is adjacent with any vertex $(2r', 4dj')$ if $|r'-r| > 4d(1+j)$. ◇

We can now finish the proof:

$$\begin{aligned} k &\geq \sum_{(2r,4dj)\in V(\mathcal{H})} \frac{q}{\alpha_{(2r,4dj)}(\mathcal{H})} \\ &\geq \sum_{(2r,4dj)\in V(\mathcal{H})} \frac{q}{4d(j+1)+1} \\ &\geq \frac{n}{2} \sum_{j\in\{0,\dots,\lfloor\frac{n}{4d+1}\rfloor-1\}} \frac{q}{4d(j+1)+1} \\ &= \frac{n}{2} \sum_{j\in\{1,\dots,\lfloor\frac{n}{4d+1}\rfloor\}} \frac{q}{4dj+1} \\ &\geq \frac{n}{2} \sum_{j\in\{1,\dots,\lfloor\frac{n}{4d+1}\rfloor\}} \frac{q}{4(d+1)j} \\ &= \frac{qn}{8(d+1)} \sum_{j\in\{1,\dots,\lfloor\frac{n}{4d+1}\rfloor\}} \frac{1}{j} \\ &\geq \frac{qn}{48(d+1)} \Big(\sum_{j\in\{1,\dots,\lfloor\frac{n}{4d+1}\rfloor\}} \frac{1}{j} + \sum_{j\in\{\lfloor\frac{n}{4d+1}\rfloor+1,\dots,\lfloor\frac{n}{d+1}\rfloor\}} \frac{1}{j} \Big) \\ &\quad \Big(\text{since } \sum_{j\in\{\lfloor\frac{n}{4d+1}\rfloor+1,\dots,\lfloor\frac{n}{d+1}\rfloor\}} \frac{1}{j} \leq \Big(\frac{n(4d+1)-n(d+1)}{(4d+1)(d+1)}\Big)\Big(\frac{4d+1}{n}\Big) = \frac{3d}{d+1} \leq 3\Big) \\ &= \frac{qn}{48(d+1)} H\Big(\frac{n}{d+1}\Big) \end{aligned}$$

where H is the harmonic function. □

Next, we aim at transposing Lemma 3.6.5 in the case of fractional strategies.

Lemma 3.6.9. *Let $n \in \mathbb{N}^*$ and $d \in \mathbb{N}$ (possibly depending on n). There exists $\beta > 0$ (the one of Lemma 3.6.5) such that $fgn_{s,d}(G_{n\times n}) \geq \beta \frac{n}{d+1} H(\frac{n}{d+1})$, where H is the harmonic function. Moreover, against a smaller amount of guards, the spy wins after at most $2n$ rounds starting from a corner of $G_{n\times n}$.*

Proof. Let us start by the following claim.

Claim 3.6.10. *Let G be any graph with n vertices and $d, s, t, q \in \mathbb{N}$. Then,*

$$gn^{q,t}_{s,d}(G) \leq q \cdot fgn_{s,d}(G) + (t+1)n^2$$

Proof of the claim. From a fractional strategy using a total amount c of guards, let us define an integral strategy keeping at least q guards at distance at most d from the spy during at least t rounds. Initially, each vertex v which has an amount x_v of guards in the fractional strategy receives $\lfloor x_v q \rfloor + (t+1)n$ guards in the integral strategy. That is, our integral strategy uses at most $\sum_{v\in V(G)}((\lfloor x_v q \rfloor + (t+1)n) \leq (t+1)n^2 + \sum_{v\in V(G)} x_v q \leq (t+1)n^2 + cq$ guards.

We then ensure that, in each round $t' \in \{1, ..., t\}$, a vertex v occupied by an amount of x_v guards in the fractional strategy is occupied by at least $\lfloor x_v q \rfloor + (t-t')n$ guards in the integral strategy. To this aim, whenever an amount x_{uv} of guards is to be moved from u to v in the fractional strategy, we move $\lfloor x_{uv} q \rfloor + 1$ in the integral strategy.

Precisely, let x_v (resp., x'_v) be the amount of guards at v in round t' (resp., at $t'+1$). Let $A \subseteq N(v)$ be the set of neighbours of v sending it a positive amount of flow and let $B \subseteq N(v)$ be the set of neighbours of v that receive a positive amount of flow from v. We have $x_v + \sum_{u\in A} x_{uv} - \sum_{u\in B} x_{vu} = x'_v$.

In the integral strategy, by induction on t', we get that, after round $t'+1$, the number of guards at v is at least

$$\begin{aligned}
&\lfloor x_v q \rfloor + (t-t')n + \sum_{u\in A}(\lfloor x_{uv} q \rfloor + 1) - \sum_{u\in B}(\lfloor x_{vu} q \rfloor + 1) \\
&\geq x_v q - 1 + (t-t')n + \sum_{u\in A}(x_{uv} q + 1 - 1) - \sum_{u\in B}(x_{vu} q + 1) \\
&\geq q(x_v + \sum_{u\in A} x_{uv} - \sum_{u\in B} x_{vu}) + (t-t')n - 1 - |B| \\
&= q x'_v + (t-t')n - 1 - |B|.
\end{aligned}$$

Since $B \subseteq N(v)$, $|B| < n$ and so, the number of guards at v in round $t'+1$ is at least $q x'_v + (t-t'-1)n$.

As our invariant is preserved throughout the t rounds, the spy which had an amount of at least 1 guard within distance d in the fractional strategy now has at least q guards

around it, which proves the result. Indeed, the number of guards at distance at most d from the spy (occupying vertex y in round $t' \leq t$) is at least

$$\begin{aligned}
&\sum_{v \in N_d(y)} (\lfloor x_v q \rfloor + (t - t' + 1)n) \\
&\geq \sum_{v \in N_d(y)} (x_v q - 1 + (t - t' + 1)n) \\
&\geq q \sum_{v \in N_d(y)} x_v \\
&\geq q.
\end{aligned}$$

$\diamond$

The previous claim holds for every $q \in \mathbb{N}$. Therefore, $\limsup_{q \to \infty} \frac{gn_{s,d}^{q,t}(G)}{q} \leq fgn_{s,d}(G)$.

Finally, by Lemma 3.6.5, there exists $\beta > 0$ such that $gn_{s,d}^{q,2n}(G_{n \times n}) \geq \beta \cdot q \cdot \frac{n}{d+1} H(\frac{n}{d+1})$.

Altogether, $fgn_{s,d}(G_{n \times n}) \geq \beta \frac{n}{d+1} H(\frac{n}{d+1})$.

Moreover, Lemma 3.6.5 shows that against strictly less than $\beta \cdot q \cdot \frac{n}{d+1} H(\frac{n}{d+1})$ integral guards, the spy will win in $2n$ rounds, starting from the corner. By the claim, this result implies that the spy will win in $2n$ rounds, starting from the corner, against less than $\beta \frac{n}{d+1} H(\frac{n}{d+1})$ fractional guards. $\square$

The next lemma is a key argument for our purpose. While it holds for any graph and its proof is very simple, we have not been able to prove a similar lemma in the classical (*i.e.*, non-fractional) case. Note that this is the only part in this section where we really need to consider the fractional variant of the spy game.

Lemma 3.6.11. *Let $G = (V, E)$ be any graph and $s \geq 2$, $d \geq 0$ be two integers with $fgn_{s,d}(G) > c \in \mathbb{Q}^*$ where the spy wins in at most t rounds against c guards starting from $v \in V(G)$.*

For any fractional strategy using a total amount $k > 0$ of guards, there exists a strategy for the spy (with speed s) starting from v such that after at most t rounds, the amount of guards at distance at most d from the spy is less than k/c.

Proof. For purpose of contradiction, assume that there is a strategy $\mathcal{S}$ using $k > 0$ guards that contradicts the lemma. Then, consider the strategy $\mathcal{S}'$ obtained from $\mathcal{S}$ by multiplying the number of guards by c/k. That is, if $w \in V$ is initially occupied by $q > 0$ guards in $\mathcal{S}$, then $\mathcal{S}'$ places qc/k guards at w initially (note that $\mathcal{S}'$ uses a total amount of $kc/k = c$ guards). Then, when $\mathcal{S}$ moves an amount q of guards along an edge $e \in E$, $\mathcal{S}'$ moves qc/k guards along e. Since $\mathcal{S}$ contradicts the lemma, in any round $\leq t$, at least an amount k/c of guards is at distance at most d from the spy, whatever be the strategy of the spy. Therefore, $\mathcal{S}'$ ensures that an amount of at least 1 guard is at distance at most d from the spy during at least t rounds. This contradicts that the spy wins after at most t rounds against a total amount of c guards. $\square$

From Lemmas 3.6.9 and 3.6.11, we get

Corollary 3.6.12. *Let $n \in \mathbb{N}^*$ and $d \in \mathbb{N}$ (possibly depending on n). There exists $\beta > 0$ (the one of Lemma 3.6.5) such that for any strategy using a total amount of $k > 0$ guards in $G_{n\times n}$, there exists a strategy for the spy (with speed s, starting from a corner of $G_{n\times n}$) such that after at most $2n$ rounds, the amount of guards at distance at most d from the spy is less than $\frac{k}{\beta\frac{n}{d+1}H(\frac{n}{d+1})}$.*

The next Theorem is the main result of this section. Before going into the details, we aim at providing informal guidelines of its proof.

In a few words, we use Corollary 3.6.12 recursively to show that if $G_{n\times n}$ requires $\Theta(n\log(n))$ guards, then a spy traveling in a larger $G_{an\times an}$ grid (for a large integer a) must have, at all times, at least $\Theta(n\log(n))$ guards in any $G_{n\times n}$ grid that contains the spy. This provides a new lower bound, on which we recurse until the claimed lower bound is obtained.

Theorem 3.6.13. *Let $s \geq 2$ and $d \in \mathbb{N}^*$. There exist $\beta^* > 0$ and $n_0 \in \mathbb{N}$ such that, for every $n \geq n_0$, $fgn_{s,d}(G_{n\times n}) = \Omega(n^{1+\beta^*})$.*

Proof. Let $a \geq 2$ be an integer and let $\beta > 0$ be the constant defined in Lemma 3.6.5. Note that, as $fgn_{s,d}$ is monotone by Observation 3.6.4, we only need to prove the theorem for grids of side $n \in \{a^c \mid c \in \mathbb{N}\}$. Let us then define $n = a^c$ for some $c \geq 1$ and $d_p = \frac{4n}{a^p}$ for every $p \in \mathbb{N}^*$, where $4n/a^p \geq d$. The following claim is a recursive application of Corollary 3.6.12:

Claim 3.6.14. *Let $p \in \mathbb{N}^*$ such that $d_p = 4n/a^p \geq 1$. For any $k > 0$, there is a spy-strategy (against an amount of k fractional guards) in $G_{n\times n}$ such that, after $\sum_{i=1}^{p}\frac{2n}{a^{i-1}}$ rounds, the amount of guards at distance at most d_p from the spy is less than $k_p = \frac{k}{\Pi_{i=1}^{p}(\beta\frac{n/a^{i-1}}{d_i+1}H(\frac{n/a^{i-1}}{d_i+1}))}$.*

Proof of the claim. The proof is by induction on $p \geq 1$.

By Corollary 3.6.12 applied for d_1, there is a spy-strategy (against fractional guards) such that, after $2n$ rounds, the amount of guards at distance at most d_1 from the spy is less than $\frac{k}{\beta\frac{n}{d_1+1}H(\frac{n}{d_1+1})}$. Therefore, the induction hypothesis holds for $p = 1$. Let us assume by induction that it holds for $p-1 \geq 1$. That is, there is a spy-strategy $\mathcal{S}_{p-1}$ such that, after $t_{p-1} = \sum_{i=1}^{p-1}\frac{2n}{a^{i-1}}$ rounds, the amount of guards at distance at most d_{p-1} from the spy is less than $k_{p-1} = \frac{k}{\Pi_{i=1}^{p-1}(\beta\frac{n/a^{i-1}}{d_i+1}H(\frac{n/a^{i-1}}{d_i+1}))}$.

Let v_{p-1} be the vertex reached by the spy after the $(t_{p-1})^{th}$ round of the spy-strategy $\mathcal{S}_{p-1}$. Let F_{p-1} be the set of vertices at distances at most $d_{p-1} = \frac{4n}{a^{p-1}}$ from v_{p-1} and let H_p be any subgrid of $G_{n\times n}$ with v_{p-1} as corner and side $\frac{n}{a^{p-1}}$. Note that any vertex in $V(G_{n\times n}) \setminus F_{p-1}$ is at distance at least $\frac{2n}{a^{p-1}}$ from any vertex of H_p (since $a \geq 2$). Therefore, after following the spy-strategy $\mathcal{S}_{p-1}$ during t_{p-1} rounds, whatever the spy does during the next $\frac{2n}{a^{p-1}}$ rounds, there can be at most $k_{p-1} = \frac{k}{\Pi_{i=1}^{p-1}(\beta\frac{n/a^{i-1}}{d_i+1}H(\frac{n/a^{i-1}}{d_i+1}))}$ guards in H_p.

Starting from the configuration reached after the $(t_{p-1})^{th}$ round of the spy-strategy $\mathcal{S}_{p-1}$, and applying Corollary 3.6.12 in H_p (of side n/a^{p-1}) and for d_p, there is a spy-strategy $\mathcal{S}'$ (against fractional guards) such that, after the next $\frac{2n}{a^{p-1}}$ rounds, the amount of guards at distance at most d_p from the spy is less than $k_p = \frac{k_{p-1}}{\beta \frac{n/a^{p-1}}{d_p+1} H(\frac{n/a^{p-1}}{d_p+1})}$. Hence, concatenating $\mathcal{S}_{p-1}$ and $\mathcal{S}'$, we get a spy-strategy $\mathcal{S}_p$ (against fractional guards) such that, after $\sum_{i=1}^{p} \frac{2n}{a^{i-1}}$ rounds, the amount of guards at distance at most d_p from the spy is less than $k_p = \frac{k}{\Pi_{i=1}^{p}(\beta \frac{n/a^{i-1}}{d_i+1} H(\frac{n/a^{i-1}}{d_i+1}))}$. $\diamond$

Let us now simplify the bound provided in the above claim. We have:

$$\begin{aligned}
X &:= \Pi_{i=1}^{p} \Big[\beta \frac{n/a^{i-1}}{d_i+1} H\Big(\frac{n/a^{i-1}}{d_i+1}\Big)\Big] \\
&= \Pi_{i=1}^{p} \Big[\beta \frac{n/a^{i-1}}{\frac{4n}{a^i}+1} H\Big(\frac{n/a^{i-1}}{\frac{4n}{a^i}+1}\Big)\Big] \\
&\geq \Pi_{i=1}^{p} \Big[\beta \frac{n/a^{i-1}}{\frac{5n}{a^i}} H\Big(\frac{n/a^{i-1}}{\frac{5n}{a^i}}\Big)\Big] \\
&= \Pi_{i=1}^{p} \beta \frac{a}{5} H(\frac{a}{5}) \\
&= \Big(\beta \frac{a}{5} H(\frac{a}{5})\Big)^p
\end{aligned}$$

Hence, the above claim implies that, for any $k > 0$, there is a spy-strategy (against an amount of k fractional guards) in $G_{n\times n}$ such that, after $\sum_{i=1}^{p} \frac{2n}{a^{i-1}}$ rounds, the amount of guards at distance at most d_p from the spy is less than $k_p \leq \frac{k}{(\beta \frac{a}{5} H(\frac{a}{5}))^p}$.

Note that, for the guards' strategy to be winning, it must be ensured that $k_p \geq 1$ whenever $d_p \geq d$.

Let us now define a^* to be any integer such that $\beta \frac{a^*}{5} H(\frac{a^*}{5}) > a^*$: it exists since this holds whenever $H(\frac{a^*}{5}) > \frac{5}{\beta}$, which is asymptotically true for a^* (since β is a constant).

Applying the previous claim with $a = a^*, n = (a^*)^c$, and with $p \leq c = \log_{a^*} n$ such that $d_p \geq d$, we get that $k_p \geq 1$ implies:

$$\begin{aligned}
k &\geq \Big(\beta \frac{a^*}{5} H(\frac{a^*}{5})\Big)^p \\
&= (a^* + \epsilon)^p \text{ (for some } \epsilon > 0) \\
&\geq (a^* + \epsilon)^{\log_{a^*}(\frac{4n}{d})} \text{ (Since } d_p = \frac{4n}{(a^*)^p} \geq d \Leftrightarrow \log_{a^*}(\frac{4n}{d}) \geq p) \\
&= \big((a^*)^{\log_{a^*}(a^*+\epsilon)}\big)^{\log_{a^*}(\frac{4n}{d})} = (a^*)^{\log_{a^*}(\frac{4n}{d}) \cdot \log_{a^*}(a^*+\epsilon)} \\
&= (\frac{4n}{d})^{\log_{a^*}(a^*+\epsilon)} \\
&= c' n^{\log_{a^*}(a^*+\epsilon)} \text{ (for the constant } c' = (\frac{4}{d})^{\log_{a^*}(a^*+\epsilon)}) \\
&= \Omega(n^{1+\beta^*}) \text{ (for } \beta^* = \log_{a^*}(1 + \frac{\epsilon}{a^*}) > 0)
\end{aligned}$$

□

The above theorem proves Theorem 3.6.2 and Corollary 3.6.3.

3.6.2 Upper bound in Torus

The second of the two main results of this section is:

Theorem 3.6.15. *Let $\alpha = \log_2(1 + \frac{1}{s})$. Then, for every $d \geq 0$ and every constant $s \geq 2$,*

$$fgn^*_{s,d}(T_{n\times n}) = O(n^{2-\alpha}).$$

To prove Theorem 3.6.15, we make use of the Linear Program (LP) of Section 3.4. Recall that, in a spy-positional strategy, the positions of the guards (configuration) only depend on the position of the spy. Also, note that $T_{n\times n}$ is a vertex-transitive graph, that is, given any two vertices $u, v \in T_{n\times n}$, there exists an automorphism $f : V(T_{n\times n}) \to V(T_{n\times n})$ such that $f(u) = v$. Roughly, every vertex in a vertex-transitive graph is the same. Then, in any vertex-transitive graph (so in $T_{n\times n}$), there is actually a unique configuration to be considered (where the spy is occupying the vertex $(0, 0)$). Therefore, the LP of Section 3.4 can be reformulated as follows.

Throughout subsection 3.6.2, all coordinates are assumed to be taken modulo n whenever appropriate. We are looking for a function $\omega : \{0, \ldots, n-1\}^2 \to \mathbb{R}^+$ such that $\omega(i, j)$ is the amount of guards occupying the vertex (i, j) when the spy is occupying the vertex $(0, 0)$. This function must be defined such as to minimize the number of guards, *i.e.*, $\sum_{0\leq i,j<n} \omega(i, j)$ must be minimum, subject to the following constraints. The spy must be controlled, *i.e.*, $\sum_{(i,j)\in N_d[(0,0)]} \omega(i, j) \geq 1$. Moreover, for any move of the spy from $(0, 0)$ to $(x, y) \in N_s[(0, 0)]$, there must be a feasible flow from the configuration $(\omega(i, j))_{(i,j)\in V(T_{n\times n})}$ to $(\omega(i - x, j - y))_{(i,j)\in V(T_{n\times n})}$. Before going further, let us simplify the latter constraint. Indeed, instead of considering every possible move of the spy in $N_s[(0, 0)]$, we only consider the *extremal moves* from $(0, 0)$ to one of the vertices in $\{(0, s), (s, 0), (-s, 0), (0, -s)\}$, *i.e.*, we weaken the spy by allowing it to move only "horizontally" or "vertically" at full speed. We prove in Lemma 3.6.16 that it does not change the order of magnitude of an optimal solution.

Lemma 3.6.16. *Let $n, s \geq 2$, and $d \geq 0$ be integers. Assume that there exists a (fractional or integral) winning strategy using k guards to control a spy, with speed s and restricted moves, at distance d in the $n\times n$-torus. Then, there exists a (fractional or integral) winning strategy using $O(s^2k)$ guards to control a spy, with speed s, at distance d in the $n \times n$-torus.*

Proof. The proof is written in the integral case. The fractional case is similar.

For any strategy of a (non-restricted) spy, we will define a strategy for a restricted spy, called the *spy's shadow*, that ensures that the shadow is always at distance at most $2s$ from the non-restricted spy. To control the non-restricted spy, the strategy consists of applying the strategy σ against its shadow (*i.e.*, using k guards) and replacing each

guard γ of σ by $O(s^2)$ guards, one at every vertex at distance at most $2s$ from the position of γ.

The shadow starts at the same vertex as the spy and "follows" it but only using restricted moves. The shadow can easily stay at distance $< 2s$ from the spy if the spy moves from a vertex at distance $< 2s$ from the shadow to a vertex at distance at least $2s$ (but $< 3s$ since the spy has speed s) from the shadow. This means, then, that the shadow is at a position such that one of its coordinates differs by at least s from one of the spy's coordinates. So it can decrease its distance to the spy by exactly s using a restricted move. This means that after the shadow moves, the distance is still $< 2s$. □

The above LP, restricted to vertex-transitive graphs, is more efficient than the one presented in Section 3.4 since there is only one configuration to be considered and fewer flow constraints (and so, fewer variables and constraints). In particular, it gives interesting experimental results as presented in the conclusion. In what follows, we present and analyze a function using a sub-quadratic (in n) number of guards that satisfies the above LP.

Precisely, let $0 < \alpha < 1$ and let $d(v)$ (resp., $d(i,j)$) denote the distance (length of a shortest path) between vertex v (resp., (i,j)) and vertex $(0,0)$ in $T_{n\times n}$.

Definition 3.6.17 (Strategies ω_α)**.** Let us consider the spy-positional strategy ω_α of the form $\omega_\alpha(i,j) = \frac{B}{(d(i,j)+1)^\alpha}$ for every $(i,j) \in V(T_{n\times n})$ and for some constant B defined later.

Note that ω_α is *symmetric*, *i.e.*, $\omega_\alpha(i,j) = \omega_\alpha(n-i,j) = \omega_\alpha(i,n-j) = \omega_\alpha(n-i,n-j)$. Therefore, by symmetry, we only need to check that there is a feasible flow from the configuration $(\omega_\alpha(i,j))_{(i,j)\in V(T_{n\times n})}$ to the one $(\omega_\alpha(i-s,j))_{(i,j)\in V(T_{n\times n})}$, *i.e.*, when the spy goes from $(0,0)$ to $(s,0)$.

Equivalently, the flow constraints can be defined as a flow problem in a *transportation bipartite auxiliary network* H defined as follows (*i.e.*, the constraints are satisfied if and only if there is feasible flow in H). Let $H = (V_1 \cup V_2, E(H))$ be the graph such that V_1 and V_2 are two copies of $V(T_{n\times n})$. There is an arc from $u \in V_1$ to $v \in V_2$ if $\{u,v\} \in E(T_{n\times n})$. Each vertex $(i,j) \in V_1$ has a supply $\omega_\alpha(i,j)$ and every vertex $(i',j') \in V_2$ has a demand $\omega_\alpha(i'-s,j')$. By Hall's Theorem [30], there is a feasible flow in H if and only if, for every $A \subseteq V_1$, the total supply in $N[A]$ is at least the demand in $A \subseteq V_2$, *i.e.*, at least $\sum_{(i,j)\in A} \omega_\alpha(i-s,j)$.

To summarize, the flow constraints can be stated as:

$$\forall A \subseteq V(T_{n\times n}),\ \sum_{(i,j)\in N[A]} \omega_\alpha(i,j) \geq \sum_{(i,j)\in A} \omega_\alpha(i-s,j). \tag{3.9}$$

We aim at deciding the range of α such that the function ω_α satisfies constraint 3.9. For this purpose, we first aim at finding a set $\mathcal{H}_s \subseteq V(T_{n\times n})$ such that $\kappa_\alpha(\mathcal{H}_s) = \sum_{(i,j)\in N[\mathcal{H}_s]} \omega_\alpha(i,j) - \sum_{(i,j)\in \mathcal{H}_s} \omega_\alpha(i-s,j)$ is minimum. For such a set $\mathcal{H}_s$, if $\kappa_\alpha(\mathcal{H}_s) \geq 0$, it implies that ω_α satisfies constraint 3.9.

Let $\mathcal{H}_s$ be the set of vertices $(i,j) \in V(T_{n\times n})$ defined by:

$$\mathcal{H}_\mathbf{s} = \{(i,j) \mid s/2 \leq i \leq (n+s)/2,\ 0 \leq j < n\}.$$

Lemma 3.6.18. *Let $\alpha > 0$ and $s \geq 2$ be a constant. For every $A \subseteq V(T_{n\times n})$, $\kappa_\alpha(A) \geq \kappa_\alpha(\mathcal{H}_s)$.*

Proof. For simplicity of calculations, let us assume that both s and n are even. For any $0 \leq j < n$, the *column* C_j equals $\{(i,j) \mid 0 \leq i < n\}$.

For any integer $0 \leq \ell < n$ and some constant B defined later, let $f_\ell : V \to \mathbb{R}^+$ be the function such that, for any $(i,j) \in V(T_{n\times n})$,

$$f_\ell(i,j) = \frac{B}{(d((i,j),(\ell,0))+1)^\alpha}$$

where $d(x,y)$ denotes the distance between x and y in $T_{n\times n}$.

Note that

Claim 3.6.19. *For any i,j, $f_s(i,j) = f_0(i-s,j)$.*

For any $A \subseteq V(T_{n\times n})$, let us define the *border* $\delta(A)$ of A as $\delta(A) = \{w \notin A \mid \exists v \in A, \{v,w\} \in E\}$, *i.e.*, the set of vertices not in A that have a neighbour in A.

Note that:

$$\kappa_\alpha(A) = \sum_{v\in N[A]} f_0(v) - \sum_{v\in A} f_s(v) = \sum_{v\in A}(f_0(v) - f_s(v)) + \sum_{v\in\delta(A)} f_0(v).$$

To find a vertex-set minimizing the above function, we actually define another function lower bounding the previous one. We identify a set A_{min} minimizing this second function such that both functions achieve the same value for A_{min}. Therefore, A_{min} also minimizes the first function.

The *vertical border* $\mu(A)$ equals $\{(i,j) \notin A \mid (i+1,j) \in A \text{ or } (i-1,j) \in A\}$, *i.e.*, the set of vertices not in A that have a neighbour in A and in the same column. Note that $\mu(A) \subseteq \delta(A)$ for any $A \subseteq V$.

Let us set

$$\gamma(A) = \sum_{v\in A}(f_0(v) - f_s(v)) + \sum_{v\in\mu(A)} f_0(v).$$

Since f_0 is positive and $\mu(A) \subseteq \delta(A)$,

Claim 3.6.20. *$\kappa_\alpha(A) \geq \gamma(A)$ for any $A \subseteq V$.*

A useful property of γ is that:

Claim 3.6.21. *$\gamma(A) = \sum_{0\leq j<n} \gamma(A \cap C_j)$.*

Note that $\mathcal{H}_s = \{(i,j) \mid s/2 \leq i \leq (n+s)/2,\ 0 \leq j < n\}$ is the set of vertices v such that $f_0(v) - f_s(v) \leq 0$. Moreover, note that $\mu(\mathcal{H}_s) = \delta(\mathcal{H}_s)$ (since $\mathcal{H}_s$ consists of "full" rows) and so:

Claim 3.6.22. *$\gamma(\mathcal{H}_s) = \kappa_\alpha(\mathcal{H}_s)$.*

Another useful property is that, by the first claim (and telescopical sum),

Claim 3.6.23. *For any* $0 \leq j < n$,

$$\gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) = \sum_{-s/2+1 \leq i \leq s/2+1} f_0(n/2+i, j) - \sum_{-s/2 \leq i \leq s/2-2} f_0(i, j).$$

Proof of the claim.

$$\begin{aligned}
\gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) &= \left[\sum_{i=s/2}^{(n+s)/2} f_0(i,j) - f_s(i,j) \right] + f_0(s/2-1, j) + f_0((n+s)/2+1, j) \\
&= \left[\sum_{i=s/2}^{(n+s)/2} f_0(i,j) - f_0(i-s,j) \right] + f_0(s/2-1, j) + f_0((n+s)/2+1, j) \\
&= \sum_{i=(n-s)/2+1}^{(n+s)/2+1} f_0(i,j) - \sum_{i=-s/2}^{s/2-2} f_0(i,j)
\end{aligned}$$

◇

The above proof actually extends to the following. Let

$$H(a,b) \cap C_j = \{(i,j) \mid a \leq i \leq b\}$$

Claim 3.6.24. *For any* $|a-b| > 1$,

$$\gamma(H(a,b) \cap C_j) = \sum_{-s/2+1 \leq i \leq s/2+1} f_0(b-s/2+i, j) - \sum_{-s/2 \leq i \leq s/2-2} f_0(a-s/2+i, j).$$

The remaining part of this section is devoted to prove that $\mathcal{H}_{\mathbf{s}}$ minimizes κ_α. Precisely, let us prove that $\gamma(\mathcal{H}_{\mathbf{s}}) = \min_{A \subseteq V} \gamma(A)$. This follows from the two following claims and previous claims.

Claim 3.6.25. *Let* X *be such that* $\gamma(X) = \min_{A \subseteq V} \gamma(A)$. *Then, for any* $0 \leq j < n$, $X \cap C_j$ *is connected.*

Proof of the claim. First, assume that there exists a vertex $v \in C_j \setminus X$ such that its two neighbours in C_j are in X. Then, $\gamma(X \cup \{v\}) = \gamma(X) - f_s(v) < \gamma(X)$. Therefore, by minimality of $\gamma(X)$, there are no such vertices.

Suppose there is $(n+s)/2 < i < n+s/2$ such that $u = (i,j) \in X$, $w = (i+1, j) \notin X$, and $(i-1, j) \in X$. Note that, by the previous paragraph, $(i+2, j) \notin X$. Therefore, $\gamma(X \setminus u) = \gamma(X) - f_0(w) + f_s(u) < \gamma(X)$. The last inequality is because $f_0(w) > f_s(u)$ because of the choice of i. This contradicts the minimality of $\gamma(X)$. If on the other hand, $(i-1, j) \notin X$, then $\gamma(X \setminus u) = \gamma(X) - f_0(w) - f_0(u) + f_s(u) < \gamma(X)$ which contradicts the minimality of $\gamma(X)$.

"Symmetrically", suppose there is $s/2 \leq i \leq (n+s)/2$ such that $u = (i,j) \notin X$, $w = (i-1, j) \in X$, and $(i+2, j) \notin X$. Note that, by the first paragraph, $(i+1, j) \notin X$. Therefore, $\gamma(X \cup \{u\}) = \gamma(X) - f_s(u) + f_0(i+1, j) < \gamma(X)$. The last inequality is because $f_0(i+1, j) < f_s(u)$ because of the choice of i. This contradicts the minimality

of $\gamma(X)$. If on the other hand, $(i+2,j) \in X$, then $\gamma(X \cup \{u\}) = \gamma(X) - f_s(u) < \gamma(X)$ which contradicts the minimality of $\gamma(X)$.

If $X \cap C_j$ would not be connected, one of the cases of the two previous paragraphs should occur. Therefore, $X \cap C_j$ is connected. $\diamond$

Claim 3.6.26. *Let $0 \leq j < m$. For any $X \subseteq V$ such that $X \cap C_j$ is connected, $\gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) \leq \gamma(X \cap C_j)$.*

Proof of the claim. Since $X \cap C_j$ is connected, it has the form $H(a,b) \cap C_j$ for some a and b. We assume that $|a-b| > 1$ (the other case can be done similarly). Therefore, by previous claims, it remains to prove that, for any a and b, $\gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) \leq \gamma(H(a,b) \cap C_j)$.

$$\gamma(H(a,b) \cap C_j) - \gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) =$$

$$\sum_{-s/2+1 \leq i \leq s/2+1} (f_0(b-s/2+i,j) - f_0(n/2+i,j)) - \sum_{-s/2 \leq i \leq s/2-2} (f_0(a-s/2+i,j) - f_0(i,j)).$$

Since the function f_0 is maximum around $i = 0$ and minimum around $i = n/2$, it is easy to check that, for any a and b:

$$\sum_{-s/2+1 \leq i \leq s/2+1} (f_0(b - s/2 + i, j) - f_0(n/2 + i, j)) \geq 0$$

and

$$\sum_{-s/2 \leq i \leq s/2-2} (f_0(a - s/2 + i, j) - f_0(i, j)) \leq 0.$$

Hence, $\gamma(H(a,b) \cap C_j) - \gamma(\mathcal{H}_{\mathbf{s}} \cap C_j) \geq 0$. $\diamond$

By previous claims, $\kappa_\alpha(\mathcal{H}_{\mathbf{s}}) = \gamma(\mathcal{H}_{\mathbf{s}}) = \min_{A \subseteq V} \gamma(A) \leq \min_{A \subseteq V} \kappa_\alpha(A)$.
Hence, $\kappa_\alpha(\mathcal{H}_{\mathbf{s}}) = \min_{A \subseteq V} \kappa_\alpha(A)$. $\square$

Finally, we are ready to present a winning strategy in the $n \times n$ torus which proves Theorem 3.6.15.

Lemma 3.6.27. *Let $n, s \geq 2$ where s is a constant, $d \geq 0$ and, $\alpha = \log_2(1 + \frac{1}{s})$. There exists a constant $B > 0$ (independent of n) such that the function $\omega_\alpha : V(T_{n\times n}) \to \mathbb{R}^+$ where $\omega_\alpha(v) = \frac{B}{(d(v)+1)^\alpha}$ for every $v \in V(T_{n\times n})$ is a spy-positional winning fractional strategy that uses $O(n^{2-\alpha})$ guards to control a spy with speed s at distance d in $T_{n\times n}$.*

Proof. To verify that ω_α is a winning strategy, we need to prove that it satisfies constraints 3.3 and 3.9. Let B_d be the set of vertices at distance at most d from $(0,0)$ and let $B = 1/\sum_{v \in B_d} \frac{1}{(d(v)+1)^\alpha}$.

The total amount of guards used by the strategy is:

$$\begin{aligned}
\sum_{v \in V(T_{n\times n})} \frac{B}{(d(v)+1)^\alpha} &\leq B\Big(1 + \sum_{1\leq i<n} \frac{4i}{(i+1)^\alpha}\Big) \text{(At most } 4i \text{ vertices distance } i>0 \text{ from } (0,0))\\
&\leq 4B \sum_{0\leq i<n} \frac{i+1}{(i+1)^\alpha}\\
&= 4B \sum_{0\leq i<n} (i+1)^{1-\alpha} = 4B \sum_{1\leq i\leq n} i^{1-\alpha}\\
&\leq 4B \cdot n \cdot n^{1-\alpha} \text{ (since } i^{1-\alpha} \leq n^{1-\alpha})\\
&= 4Bn^{2-\alpha}\\
&= O(n^{2-\alpha})
\end{aligned}$$

Constraint 3.3 states that $\sum_{v\in B_d} \omega_\alpha(v) \geq 1$ which is satisfied by the choice of B since

$$\begin{aligned}
\sum_{v\in B_d} \omega_\alpha(v) &= \sum_{v\in B_d} \frac{B}{(d(v)+1)^\alpha}\\
&= B \sum_{v\in B_d} \frac{1}{(d(v)+1)^\alpha}\\
&= \frac{1}{\sum_{v\in B_d} \frac{1}{(d(v)+1)^\alpha}} \sum_{v\in B_d} \frac{1}{(d(v)+1)^\alpha}\\
&= 1.
\end{aligned}$$

Constraint 3.9 states that, $\forall A \subseteq V(T_{n\times n})$, $\sum_{(i,j)\in N[A]} \omega_\alpha(i,j) \geq \sum_{(i,j)\in A} \omega_\alpha(i-s,j)$.

By Lemma 3.6.18, we know that $\kappa_\alpha(A) = \sum_{(i,j)\in N[A]} \omega_\alpha(i,j) - \sum_{(i,j)\in A} \omega_\alpha(i-s,j)$ is minimum for $A = \mathcal{H}_s$, where $\mathcal{H}_s = \{(i,j) \mid s/2 \leq i \leq (n+s)/2,\ 0 \leq j < n\}$. Hence, it is sufficient to show that $\kappa_\alpha(\mathcal{H}_s) \geq 0$.

Again, for ease of presentation, let us assume that s and n are even.

$$\kappa_\alpha(\mathcal{H}_s) = \sum_{s/2-1\leq i\leq (n+s)/2+1, 0\leq j<n} \frac{B}{(d(i,j)+1)^\alpha} - \sum_{s/2\leq i\leq (n+s)/2, 0\leq j<n} \frac{B}{(d(i-s,j)+1)^\alpha}$$

Because s is a constant, this can be simplified to:

$$\begin{aligned}
\kappa_\alpha(\mathcal{H}_s) &= \sum_{(n-s)/2+1\le i\le (n+s)/2+1, 0\le j<n} \frac{B}{(d(i,j)+1)^\alpha} - \sum_{-s/2\le i\le s/2-2, 0\le j<n} \frac{B}{(d(i,j)+1)^\alpha} \\
&\ge (s+1)\sum_{0\le j<n} \frac{B}{(d(n/2,j)+1)^\alpha} - (s-1)\sum_{0\le j<n} \frac{B}{(d(0,j)+1)^\alpha} \\
&= 2(s+1)\sum_{0\le j\le n/2} \frac{B}{(d(n/2,j)+1)^\alpha} - \frac{(s+1)\cdot B}{(d(n/2,0)+1)^\alpha} - \frac{(s+1)\cdot B}{(d(n/2,n/2)+1)^\alpha} \\
&- 2(s-1)\sum_{0\le j\le n/2} \frac{B}{(d(0,j)+1)^\alpha} + \frac{(s-1)B}{(d(0,0)+1)^\alpha} + \frac{(s-1)B}{(d(0,n/2)+1)^\alpha} \\
&= 2(s+1)\sum_{1\le j\le n/2+1} \frac{B}{(n/2+j)^\alpha} - 2(s-1)\sum_{1\le j\le n/2+1} \frac{B}{j^\alpha} \\
&- \frac{2B}{(n/2+1)^\alpha} - \frac{(s+1)\cdot B}{(n+1)^\alpha} + (s-1)B \\
&= 2(s+1)\sum_{0\le j\le n/2-1} \frac{B}{(n/2+j)^\alpha} - 2(s-1)\sum_{1\le j\le n/2} \frac{B}{j^\alpha} \\
&- \frac{2(s+1)\cdot B}{(n/2)^\alpha} + \frac{2(s+1)\cdot B}{n^\alpha} + \frac{2(s+1)\cdot B}{(n+1)^\alpha} - \frac{2(s-1)\cdot B}{(n/2+1)^\alpha} \\
&- \frac{2B}{(n/2+1)^\alpha} - \frac{(s+1)\cdot B}{(n+1)^\alpha} + (s-1)B \\
&\ge 2(s+1)\sum_{0\le j\le n/2-1} \frac{B}{(n/2+j)^\alpha} - 2(s-1)\sum_{1\le j\le n/2} \frac{B}{j^\alpha} + O(1/n^\alpha)
\end{aligned}$$

And so:

$$\kappa_\alpha(\mathcal{H}_s) \ge 2(s+1)\sum_{0\le j\le n/2-1} \frac{B}{(n/2+j)^\alpha} - 2(s-1)\sum_{1\le j\le n/2} \frac{B}{j^\alpha} + O(1/n^\alpha)$$

Since $0<\alpha$, then $p(x)=\frac{1}{x^\alpha}$ is decreasing, and $\int_a^{b+1} p(t)\,\mathrm{d}t \le \sum_{x=a}^{b} p(x) \le \int_{a-1}^{b} p(t)\,\mathrm{d}t$. Hence,

$$\begin{aligned}
\kappa_\alpha(\mathcal{H}_s)/(2B) &\ge (s+1)\int_0^{n/2} \frac{1}{(n/2+t)^\alpha}\,\mathrm{d}t - (s-1)\int_0^{n/2} \frac{1}{t^\alpha}\,\mathrm{d}t + O(1/n^\alpha) \\
&= \frac{1}{1-\alpha}[(s+1)((n)^{1-\alpha} - (n/2)^{1-\alpha}) - (s-1)(n/2)^{1-\alpha}] + O(1/n^\alpha) \\
&= \frac{n^{1-\alpha}}{1-\alpha}[(s+1)(1-(1/2)^{1-\alpha}) - (s-1)(1/2)^{1-\alpha}] + O(1/n^\alpha)
\end{aligned}$$

Hence, $\kappa_\alpha(\mathcal{H}_s) \geq 0$ if $0 \leq (s+1)(1-(1/2)^{1-\alpha}) - (s-1)(1/2)^{1-\alpha}$. In other words, $\kappa_\alpha(\mathcal{H}_s) \geq 0$ if $2^\alpha \leq \frac{s+1}{s}$, that is, if $0 < \alpha \leq \log_2(1+\frac{1}{s})$, which is the case since $\alpha = \log_2(1+\frac{1}{s})$. □

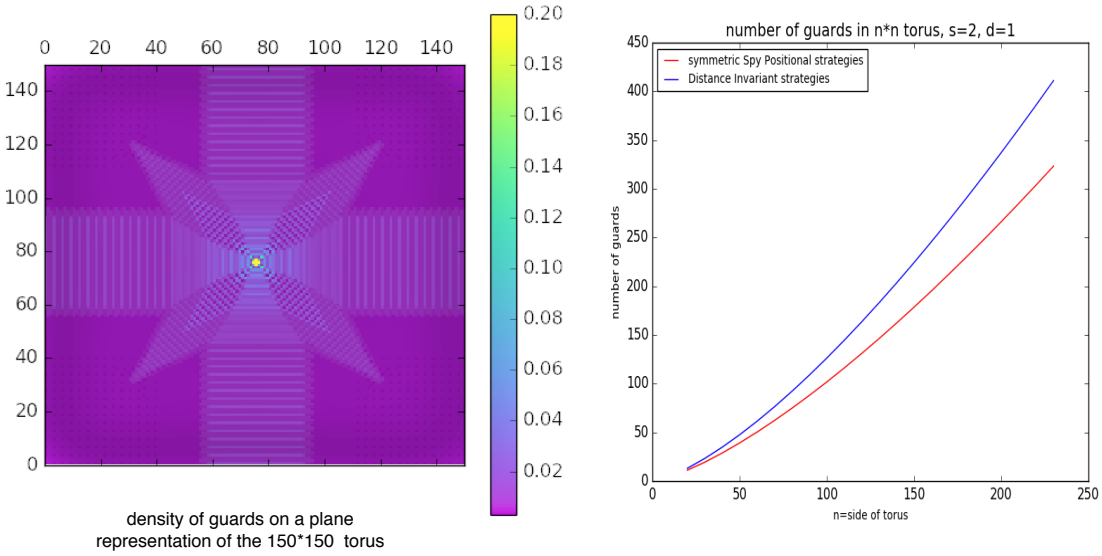


Figure 3.7: Experimental results, $s = 2$ and $d = 1$. (Left) Density of guards on a plane representation of the 150×150 torus in an optimal symmetrical Spy-positional configuration. (Right) Minimum total amount of guards for symmetrical (red) and distance-invariant (blue) Spy-Positional strategies.

3.7 Further Work

Many open questions remain such as the characterization of the guard number in other graph classes, *e.g.*, in planar graphs or interval graphs since the eternal domination game was solved for the latter. As the eternal domination game is studied in strong grids in Chapter 4, it would be interesting to study the spy game in this class of graphs as well. The exact complexity of the associated decision problem is also still open.

Determining the exact value of $gn_{s,d}(G_{n\times n})$ in any $n \times n$ grid $G_{n\times n}$ (or torus) is a very interesting problem. A first step towards such a result would be to prove that $gn_{s,d}(G_{n\times n}) = O(gn_{s',d'}(G_{n\times n}))$ for any $s, s' \geq 2$ and $d, d' \geq 0$. To get more intuition on optimal strategies for guards, we used Cplex to solve the LP described in Section 3.4 with additional constraints of symmetry. The left drawing in Fig. 3.7 represents the density of guards in the torus of side 150 (where the central vertex is the position of the spy) for $s = 2$ and $d = 1$. It shows that optimal symmetric Spy-positional (SSP)

strategies may be much more intricate than the strategy ω_α we studied. For instance, it is not monotone when the distance to the spy's position increases. On the right, we plotted the number of guards used by optimal SSP (in red) which is much less than $n^{2-\log_2(3/2)}$ for $n \leq 250$ (it is difficult to extrapolate further intuition from such small values of n)[†]. Even the optimal distance-invariant strategies (*i.e.*, the density of guards is only a function of the distance to the spy's position) computed using the LP (plotted in blue) use much less guards than $n^{2-\log_2(3/2)}$ (we did not plot the function $n^{2-\log_2(3/2)}$ for more readability, indeed, $50^{2-\log_2(3/2)} > 500$ and $250^{2-\log_2(3/2)} > 6600$).

In trees, it would be interesting to design a combinatorial algorithm (*i.e.*, not relying on the solution of a Linear Program) that computes optimal strategies for controlling a spy with speed s at distance d.

More importantly, using the fractional framework to obtain new results in two-player combinatorial games in graphs seems promising.

[†]Solving the LP for $n \geq 150$ takes more than one hour on a basic laptop.

Chapter 4

Eternal Domination

4.1 Introduction

In this chapter, the all-guards move model of the eternal domination game is considered. Recall that this is a 2-player game on graphs introduced in [72] and defined as follows. Initially, k guards are placed on some vertices of a graph $G = (V, E)$. Turn-by-turn, an *attacker* first chooses a vertex $v \in V$ to attack. Then, if no guard is occupying v or a vertex adjacent to v, then the attacker wins. Otherwise, every guard may move to a neighbour of its position and at least one guard must move to v if it is not already occupied, and the next turn starts. If the attacker never wins whatever be its sequence of attacks, then the guards win. So, clearly, there is no point in the attacker attacking an occupied vertex. The aim in eternal domination is to minimize the number of guards that must be used in order to win. Hence, let $\gamma_{all}^{\infty}(G)$ be the minimum integer k such that there exists a strategy allowing k guards to win, regardless of what the attacker does [72].

Variants of the eternal domination game also differ in the fact that one or more guards may simultaneously occupy the same vertex. In the case of the all-guards move model of the eternal domination game, there are some graphs where this constraint increases the number of guards [91]. Let $\gamma_{all}^{*\infty}(G)$ be the minimum number of guards to win in G, moving several guards per turn, and in such a way that a vertex cannot be occupied by several guards.

Deciding whether $\gamma_{all}^{\infty}(G) \leq k$ is NP-hard when G and k are part of the input and this holds for split graphs [21]. Particular graph classes have been studied such as paths and cycles [72], trees [89], and interval graphs [34, 110]. In particular, the class of grids and graph products has been widely studied [22, 49, 59, 74, 91, 94, 120]. The best known upper bound for $\gamma_{all}^{\infty}(P_n \Box P_m)$ was determined recently in [94], where it was shown that $\gamma_{all}^{\infty}(P_n \Box P_m) = \gamma(P_n \Box P_m) + O(n + m)$. The study of this game in grids is the motivation for our work on this game in strong grids and "grid-like" graphs. Specifically, this chapter focuses on results published in the paper [c-9], which is joint work with N. Nisse and S. Pérennes.

In this chapter, we focus on the class of *strong grids* SG ($P_n \boxtimes P_m$) and provide an almost tight asymptotical value for $\gamma_{all}^{\infty}(SG)$. Our result also holds for $\gamma_{all}^{*\infty}(SG)$. Our

main result is a new technique to prove upper bounds that we believe can be generalized to many other "grid-like" graphs.

4.1.1 Our Results

The main result of this chapter is that, for all $n, m \in \mathbb{N}^*$ such that $m \geq n$,

$$\left\lfloor \frac{n}{3} \right\rfloor \left\lfloor \frac{m}{3} \right\rfloor + \Omega(n+m) = \gamma_{all}^{\infty}(P_n \boxtimes P_m) = \left\lceil \frac{n}{3} \right\rceil \left\lceil \frac{m}{3} \right\rceil + O(m\sqrt{n}).$$

We prove that this result also holds in the case when at most one guard may occupy each vertex (see Section 4.5).

Note that, in toroidal strong grids $C_n \boxtimes C_m$, the problem becomes trivial and $\gamma_{all}^{\infty}(C_n \boxtimes C_m) = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil$ for any n and m. However, in strong grids, border-effects make the problem much harder. The upper bound is proven by defining a set of specific configurations that each dominate the grid and are "invariant" to the movements required by the defined strategy to defend against attacks. That is, the attacks are separated into three types of attacks: horizontal, vertical, and diagonal, and the strategy defined gives the movement of the guards based on the type of attack. It is shown that in each of the three cases of attacks, the guards are able to move from their current configuration to another configuration in the set of configurations (so, it does not matter which configuration was the initial one and which new configuration the guards reach after their moves) and hence, the guards can defend against an infinite sequence of attacks.

The lower bound is proven by showing that, in any winning configuration in eternal domination, there are some vertices that are dominated by more than one guard, and/or some guards dominate at most 6 vertices. By double counting, this leads to the necessity of having $\Omega(n+m)$ extra guards compared to the classical domination (when $n \equiv 0 \pmod 3$ and $m \equiv 0 \pmod 3$).

4.2 Preliminaries

Let $n, m \in \mathbb{N}^*$ be such that $m \geq n$ and let the $n \times m$ strong grid, denoted by $SG_{n\times m}$, be the strong product $P_n \boxtimes P_m$ of an n-node path with an m-node path. Precisely, $SG_{n\times m}$ is the graph with the set of vertices $\{(i,j) \mid 1 \leq i \leq n, 1 \leq j \leq m\}$, and two vertices (i_1, j_1) and (i_2, j_2) are adjacent if and only if $\max\{|i_2 - i_1|, |j_2 - j_1|\} = 1$. That is, the vertices are identified by their Cartesian coordinates, *i.e.*, the vertex (i,j) is the vertex in *row* i and *column* j. The vertex $(1,1)$ is in the *bottom-left corner* and the vertex (n,m) is in the *top-right corner*.

Definition 4.2.1. The set of *border* vertices of $SG_{n\times m}$ is the set

$$B = \bigcup_{1\leq i\leq n, 1\leq j\leq m} \{(1,j),(n,j),(i,1),(i,m)\} \text{ of vertices of degree} \leq 5.$$

The set of *pre-border* vertices of $SG_{n\times m}$ is the set $PB = N(B)$.

Equivalently, PB is the set of border vertices of the strong grid induced by $V(SG_{n\times m}) \setminus B$.

We consider the all-guards move model of the eternal domination game. Each *turn*, each vertex of a graph $G = (V, E)$ may be occupied by one or more guards. Let $k \in \mathbb{N}^*$ be the total number of guards. The positions of the guards are formally defined by a multi-set C of vertices, called a *configuration*, where the number of occurrences of a vertex $v \in C$ corresponds to the number of guards at $v \in V$ and $k = |C|$. Each turn, given a current configuration $C = \{v_i \mid 1 \leq i \leq k\}$ of k guards, Player 1, the *attacker*, *attacks* a vertex $v \in V$. Then, Player 2 (the *defender*) may move each of its *guards* to a neighbour of their current position, thereby, achieving a new configuration $C' = \{w_i \mid 1 \leq i \leq k\}$ such that $w_i \in N[v_i]$ for every $1 \leq i \leq k$ (we then say that C' is *compatible* with C, which is clearly a symmetric relation). If $v \notin C'$, then the attacker *wins*, otherwise, the game goes on with a next turn (given the new configuration C').

A *strategy* for k guards is defined by an initial configuration of size k and by a function that, for every current configuration C and every attacked vertex $v \in V$, specifies a new configuration C' compatible with C. A strategy $\mathcal{S}$ for the guards is *winning* if, for every sequence of attacked vertices, the attacker never wins when the defender plays according to $\mathcal{S}$.

Our main contribution is a winning strategy for $\gamma(SG_{n\times m}) + o(\gamma(SG_{n\times m}))$ guards in $SG_{n\times m}$, where $\gamma(SG_{n\times m}) = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil$ is the *domination number* of $SG_{n\times m}$. The next lemma is key for this winning strategy.

In our strategy, it will often be useful to move a guard from a node $u \in PB$ of the pre-border to another node $v \in PB$ such that u and v are not necessarily adjacent. For this purpose, the idea is to place a sufficient number of guards on the vertices of the border such that a "flow" of the guards on the border vertices will simulate the move of the guard from u to v in one turn.

Precisely, given a configuration C and $u, v \in V(SG_{n\times m})$ with $u \in C$, a guard is said to *jump* from u to v if the configuration $(C \setminus \{u\}) \cup \{v\}$ is compatible with C, *i.e.*, the guards, in one turn, can move to achieve the same configuration as C except that there is one guard less on u and one guard more on v. More generally, given $U \subset C$ and $W \subset V(SG_{n\times m})$, a set of guards is said to *jump* from U to W if the configuration $(C \setminus U) \cup W$ is compatible with the configuration C.

Lemma 4.2.2. *Let $\alpha, \beta \in \mathbb{N}^*$ such that $\beta \leq \alpha$. Let $U, W \subseteq PB$ be two subsets of pre-border vertices such that $|U| = |W| = \beta$. In any configuration C such that $U \subseteq C$ and C contains at least α occurrences of each vertex in B (i.e., each border vertex is occupied by at least α guards), then β guards may "jump" from U to W in one turn. Moreover, only guards in $U \cup B$ move.*

Proof. The proof is by induction on β. The inductive hypothesis is that if each vertex in B contains α guards, then $\beta \leq \alpha$ guards may "jump" from U to W in one turn such that at most β guards move off of each vertex $w \in B$ in this turn. For the base case, let us assume that $U = \{u\}$ and $W = \{w\}$. Let us show how 1 guard can "jump" from u to w in one turn. If $u = w$, the result trivially holds, so let $u \neq w$. Let $u' \in B$ (resp., w') be a neighbour of u (of w) that shares one coordinate with u (with w). Let

$Q = (u' = v_0, v_1, \ldots, v_\ell = w')$ be a path from u' to w' induced by the border vertices. In one turn, a guard at u moves to u', for every $0 \leq i < \ell$, a guard at v_i moves to v_{i+1}, and a guard at v_ℓ moves to w.

Now, assume the inductive hypothesis holds for $\beta \geq 1$. If $\beta = \alpha$, we are done, so assume $\beta < \alpha$. Let $|U| = |W| = \beta + 1 \leq \alpha$ and let $u \in U$ and $w \in W$. By the inductive hypothesis, β guards may jump from $U \setminus \{u\}$ to $W \setminus \{w\}$ in one turn in such a way that, for every vertex $b \in B$, at most β guards move off of b during this turn. Since every vertex of B is occupied by $\alpha > \beta$ guards, at least one guard is unused on every vertex of B. Thus, it possible to use the same strategy as in the base case to make one guard jump from u to w on this same turn. □

4.3 Upper Bound Strategy

This section is devoted to proving that for all $n, m \in \mathbb{N}^*$ such that $m \geq n$, $\gamma^\infty_{all}(SG_{n\times m}) = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil + O(m\sqrt{n})$.

Before considering the general case, let us first assume that $n - 2 \equiv 0 \pmod 3$ and that there exists $k \in \mathbb{N}^*$ such that $k - 2 \equiv 0 \pmod 3$, and $m \equiv 0 \pmod k$. The $n \times m$ strong grid will be partitioned into *blocks* which are subgrids of size $n \times k$. More precisely, for all $1 \leq q \leq \frac{m}{k}$, the q^{th} block contains columns $(q-1)k + 1$ through qk of $SG_{n\times m}$.

4.3.1 Horizontal Attacks

In this section, we only consider one block of $SG_{n\times m}$. W.l.o.g., let us consider the block $SG_{n\times k}$ induced by $\{(i, j) \mid 1 \leq i \leq n, 1 \leq j \leq k\}$. Let us first define a family of parameterized configurations for this block.

Let $\mathcal{X} = \{(b, a_1, \ldots, a_{\frac{n-2}{3}}) \mid b \in \{1, 2, 3\}, a_i \in \{1, 2, 3\} \text{ for } i = 1, \ldots, \frac{n-2}{3}\}$.

Given $X = (b, a_1, \ldots, a_{\frac{n-2}{3}}) \in \mathcal{X}$, let $x_i(X) = 3(i-1) + b + 1$, and $y_{j,i}(X) = 3(j - 1) + a_i + 1$ for every $1 \leq i \leq \frac{n-2}{3}$ and $1 \leq j \leq \frac{k-2}{3}$. We set $x_i = x_i(X)$ and $y_{j,i} = y_{j,i}(X)$ when there is no ambiguity. Intuitively, b will represent the *vertical shift* of the positions of the guards in configuration X. Similarly, for every $1 \leq i \leq \frac{n-2}{3}$, a_i represents the *horizontal shift* of the positions of the guards in row $x_i(X)$ in configuration X (see Figure 4.1).

Horizontal Configurations. Let us define the set $\mathcal{C}_H$ of configurations as follows. For every $X \in \mathcal{X}$, let $C_H(X) = B \cup \{(x_i(X), y_{j,i}(X)) \mid 1 \leq i \leq \frac{n-2}{3}, 1 \leq j \leq \frac{k-2}{3}\}$ be the configuration where there is one guard at every vertex of B and one guard at each vertex $(x_i(X), y_{j,i}(X)) = (3(i-1) + b + 1, 3(j-1) + a_i + 1)$ for every $1 \leq i \leq \frac{n-2}{3}$ and $1 \leq j \leq \frac{k-2}{3}$. See an example in Figure 4.1. Then,

$$\mathcal{C}_H = \{C_H(X) \mid X \in \mathcal{X}\}.$$

Note that $|C_H(X)| = \frac{(n-2)(k-2)}{9} + 2(n + k) - 4 = \kappa_H$ for every $X \in \mathcal{X}$. That is, any horizontal configuration uses κ_H guards.

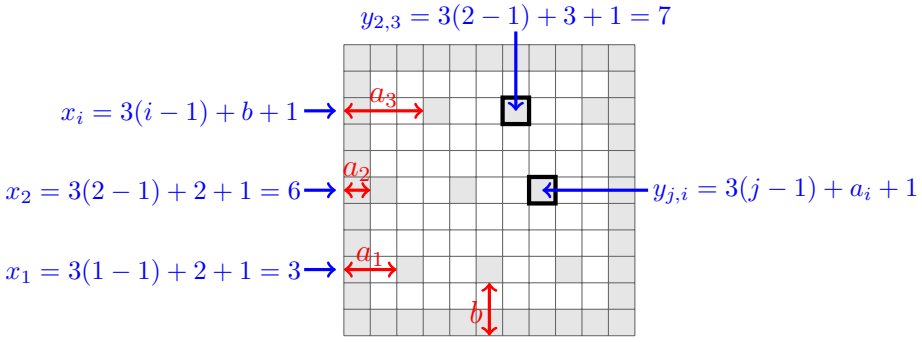


Figure 4.1: $P_{11} \boxtimes P_{11}$ where the squares are vertices and two squares sharing a side and/or a corner are adjacent. Example of a configuration $C_H(X)$ where $X = (b = 2, a_1 = 2, a_2 = 1, a_3 = a_{\frac{n-2}{3}} = 3)$, there is one guard at each square in gray, and the white squares contain no guards.

Lemma 4.3.1. *Every configuration $C_H(X) \in \mathcal{C}_H$ is a dominating set of the block $SG_{n\times k}$.*

Proof. The pre-border and border vertices are dominated by the guards on the border vertices. For all $i, j \in \mathbb{N}^*$ such that $1 \leq i \leq \frac{n-2}{3}$ and $1 \leq j \leq \frac{k-2}{3}$, the guards on the vertices $(x_i, y_{j,i})$ dominate the vertices $\{(x_i+1, y_{j,i}), (x_i-1, y_{j,i}), (x_i, y_{j,i}-1), (x_i, y_{j,i}+1), (x_i+1, y_{j,i}+1), (x_i+1, y_{j,i}-1), (x_i-1, y_{j,i}-1), (x_i-1, y_{j,i}+1)\}$. □

In this subsection, we limit the power of the attacker by allowing it to attack only some predefined vertices (this kind of attack will be referred to as a *horizontal attack*). For every configuration $C_H(X) \in \mathcal{C}_H$ and for any such attack, we show that the guards may be moved (in one turn) in such a way to defend the attacked vertex and reach a new configuration in $\mathcal{C}_H$.

Horizontal Attacks. Let $X = (b, a_1, \ldots, a_{\frac{n-2}{3}}) \in \mathcal{X}$ and $C_H(X) \in \mathcal{C}_H$. Let

$$A_H(X) = \{(x_i, y) \mid 1 \leq i \leq \frac{n-2}{3}, 1 \leq y \leq k\}.$$

A *horizontal attack with respect to* X is an attack at any vertex in $A_H(X)$, *i.e.*, an attack at any vertex of a row where some non-border vertex is occupied by a guard. Note that, for every vertex $v \in A_H(X)$, either v is occupied by a guard or there is a guard on the vertex to the left or to the right of v. In Figure 4.2, red squares represent the vertices of $A_H(X) \setminus C_H(X)$.

The next lemma proves that, from any horizontal configuration and against any horizontal attack (with respect to this current configuration), there is a possible strategy for the guards that defends against this attack and leads to a (new) horizontal configuration. Therefore, starting from any horizontal configuration, there is a strategy of the guards that wins against any sequence of horizontal attacks.

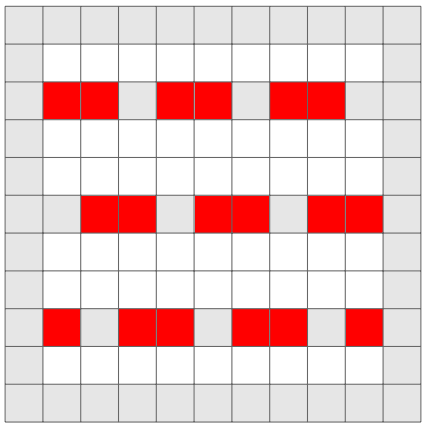


Figure 4.2: $P_{11} \boxtimes P_{11}$ where the squares are vertices. Example of the non-occupied attackable vertices in red when only horizontal attacks are considered. The guards occupy a configuration $C_H(X)$ where $X = (b = 2, a_1 = 2, a_2 = 1, a_3 = a_{\frac{n-2}{3}} = 3)$, there is one guard at each square in gray, and the white squares contain no guards.

Lemma 4.3.2. *For any $X \in \mathcal{X}$ and any $v \in A_H(X)$, there exists $X' \in \mathcal{X}$ such that $v \in C_H(X')$ and configurations $C_H(X)$ and $C_H(X')$ are compatible. That is, in one turn, the guards may move from $C_H(X)$ to $C_H(X')$ and defend against an attack at v.*

Proof. Initially, κ_H guards are in a configuration $C_H(X)$ (see Figure 4.1). Consider an attack at some vertex $v \in A_H(X)$. If $v \in C_H(X)$, all guards may remain idle. Hence, let us assume that $v \in A_H(X) \setminus C_H(X)$.

Let us assume that $v = (x_\ell(X), y_{w,\ell}(X) - 1) = (x_\ell, y_{w,\ell} - 1)$ for some integers $1 \leq \ell \leq \frac{n-2}{3}$ and $1 \leq w \leq \frac{k-2}{3}$ (note that if $a_\ell = 1$ then $w > 1$ since v is not a border vertex), that is v is to the left of the vertex $(x_\ell, y_{w,\ell})$ that is occupied by a guard. The cases of attacks at $(x_\ell(X), y_{w,\ell}(X) + 1)$ (v is to the right of an occupied vertex) or $(x_\ell(X), 2)$, or $(x_\ell(X), k - 1)$ (the attacked vertex v is adjacent to a border vertex), are similar, by symmetry, to at least one of the two cases below.

The guards will move from the configuration $C_H(X)$ to a configuration $C_H(X')$ that defends against the attack at v, *i.e.*, $v \in C_H(X')$, where $X' = \{b', a'_1, \ldots, a'_{\frac{n-2}{3}}\}$ as defined below.

Intuitively, for the guards to move from the configuration $C_H(X)$ to a configuration $C_H(X')$ that defends against this attack at v, all the guards in row x_ℓ will shift left except for perhaps the guards on the border vertices (it depends on the value of a_ℓ). Hence, the only difference between X and X' will be the value of the horizontal shift related to row x_ℓ.

Precisely, by the definition of $C_H(X)$, there is a guard at $(x_\ell, y_{w,\ell})$. There are two cases of how the guards will move in response to the attack, depending on the three possible values of $a_\ell \in \{1, 2, 3\}$.

Case i) $a_\ell \in \{2, 3\}$**.** To defend against the attack, all the guards in row x_ℓ except those that occupy border vertices, shift one vertex to the left. That is, the guard at $(x_\ell, y_{j,\ell})$ moves to $(x_\ell, y_{j,\ell} - 1)$ for all $j \in \mathbb{N}^*$ such that $1 \leq j \leq \frac{k-2}{3}$. Since the

positions of the other guards did not change, the guards occupy a configuration $C_H(X')$ where $b' = b$, $a'_i = a_i$ for all $1 \leq i \leq \frac{n-2}{3}$ such that $i \neq \ell$, but $a'_\ell = a_\ell - 1$.

Case ii) $a_\ell = 1$**.** To defend against the attack, all the guards in row x_ℓ except the one at $(x_\ell, 1)$, shift one vertex to the left. That is, the guard at $(x_\ell, y_{j,\ell})$ moves to $(x_\ell, y_{j,\ell} - 1)$ for all $j \in \mathbb{N}^*$ such that $1 < j \leq \frac{k-2}{3}$. Also, the guard at $(x_\ell, 2)$ jumps to $(x_\ell, k-1)$ which is possible by Lemma 4.2.2 and since none of the border guards have to move for any other purpose. Since the positions of the other guards did not change, the guards occupy a configuration $C_H(X')$ where $b' = b$, $a'_i = a_i$ for all $1 \leq i \leq \frac{n-2}{3}$ such that $i \neq \ell$, but $a'_\ell = 3$. See Figure 4.3.

□

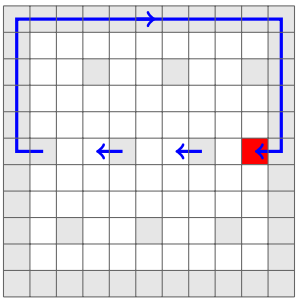


Figure 4.3: $P_{11} \boxtimes P_{11}$ where the squares are vertices. Example of an attack in Case ii) at the red square. The guards occupy a configuration $C_H(X)$ where $X = (b = 2, a_1 = 2, a_2 = 1, a_3 = a_{\frac{n-2}{3}} = 3)$, there is one guard at each square in gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack.

4.3.2 Vertical Attacks

In this section, we consider the entire strong grid $SG_{n\times m}$ partitioned into $\frac{m}{k}$ blocks $SG_{n\times k}$ with block q, for $1 \leq q \leq \frac{m}{k}$, being induced by $\{(i, j + (q-1)k) \mid 1 \leq i \leq n, 1 \leq j \leq k\}$. Let us first define a family of parameterized configurations for this graph.

A configuration for the whole grid will be defined as the union of some configurations for each of the q blocks. Formally, for every $1 \leq q \leq \frac{m}{k}$, let us first define:

$$\mathcal{X}^q = \{(b^q, a^q_1, \ldots, a^q_{\frac{n-2}{3}}) \mid b^q, a^q_i \in \{1,2,3\} \text{ for } i = 1, \ldots, \frac{n-2}{3} \text{ and } q = 1, \ldots, \frac{m}{k}\}.$$

Given $X^q = (b^q, a^q_1, \ldots, a^q_{\frac{n-2}{3}}) \in \mathcal{X}^q$, let $x^q_i(X^q) = 3(i-1) + b^q + 1$, and $y^q_{j,i}(X^q) = (q-1)k + 3(j-1) + a^q_i + 1$ for every $1 \leq i \leq \frac{n-2}{3}$, $1 \leq j \leq \frac{k-2}{3}$, and $1 \leq q \leq \frac{m}{k}$. We set $x^q_i = x^q_i(X^q)$ and $y^q_{j,i} = y^q_{j,i}(X^q)$ when there is no ambiguity.

That is, intuitively, b^q will represent the *vertical shift* of the positions of the guards in configuration X^q in the q^{th} block. Similarly, for every $1 \leq i \leq \frac{n-2}{3}$, a^q_i represents the *horizontal shift* of the positions of the guards in row $x_i(X)$ in configuration X^q in the q^{th} block.

Finally, let $\mathcal{Y} = \{(X^1, \ldots, X^{\frac{m}{k}}) \mid X^q \in \mathcal{X}^q \text{ for } q = 1, \ldots, \frac{m}{k}\}$.

Vertical Configurations. In order to properly define the following set of configurations, we require the following notation. For a set S of vertices in a configuration $\mathcal{C}$ and an integer $x > 0$, let $S^{[x]}$ be the multi-set of vertices that consists of x copies of each vertex in S. Intuitively, $S^{[x]}$ will be used to define a configuration where x guards occupy each vertex of S. Let us now define the set $\mathcal{C}_V$ of configurations as follows.

For every $Y = (X^1, \ldots, X^{\frac{m}{k}}) \in \mathcal{Y}$, let $C_V(Y) = B^{[\frac{k-2}{3}]} \cup \bigcup_{q=1}^{\frac{m}{k}} C_H(X^q)$ be the configuration obtained as follows. First, for any $1 \leq q \leq \frac{m}{k}$, guards are placed in configuration $C_H(X^q)$ in the q^{th} block. Then, $\frac{k-2}{3}$ guards are added to every border vertex. Note that overall, there are $\frac{k-2}{3}+1$ guards at each vertex of B. See an example in Figure 4.4. Then,

$$\mathcal{C}_V = \{C_V(Y) \mid Y \in \mathcal{Y}\}.$$

Note that $|C_V(Y)| = \frac{m}{k}\kappa_H + 2(\frac{k-2}{3})(n+m-2) = \kappa_V$ for every $Y \in \mathcal{Y}$. That is, any vertical configuration uses κ_V guards.

Lemma 4.3.3. *Every configuration $C_V(Y) \in \mathcal{C}_V$ is a dominating set of $SG_{n\times m}$.*

Proof. Since $C_V(Y) \in \mathcal{C}_V$, by definition, for all $1 \leq q \leq \frac{m}{k}$, there exists $X^q \in \mathcal{X}^q$ such that the vertices of $C_H(X^q)$ are occupied by guards. Therefore, each of the $\frac{m}{k}$ blocks $SG_{n\times k}$ is dominated by the guards within it by Lemma 4.3.1. □

In this subsection, we limit the power of the attacker by allowing it to attack only some predefined vertices (this kind of attack will be referred to as a *vertical* attack). For every configuration $C_V(X) \in \mathcal{C}_V$ and for any such attack, we show that the guards may be moved (in one turn) in such a way to defend the attacked vertex and reach a new configuration in $\mathcal{C}_V$.

Vertical Attacks. Let $Y = (X^1, \ldots, X^{\frac{m}{k}}) \in \mathcal{Y}$ and $C_V(Y) \in \mathcal{C}_V$. Let

$$\begin{aligned} A_V(Y) =& \{(x_i^q - 1, y_{j,i}^q), (x_i^q + 1, y_{j,i}^q) \mid 1 \leq i \leq \frac{n-2}{3}, 1 \leq j \leq \frac{k-2}{3}, 1 \leq q \leq \frac{m}{k}\} \\ & \cup \{(2, y_{j,n-1}^q) \mid 1 \leq j \leq \frac{k-2}{3}, 1 \leq q \leq \frac{m}{k} \text{ and } b^q = 3\} \\ & \cup \{(n-1, y_{j,2}^q) \mid 1 \leq j \leq \frac{k-2}{3}, 1 \leq q \leq \frac{m}{k} \text{ and } b^q = 1\} \end{aligned}$$

A *vertical attack with respect to* Y is an attack at any vertex in $A_V(Y)$, *i.e.*, an attack at any non-border vertex above or below a guard not on a border vertex. Moreover, if the vertical shift b^q of the q^{th} block equals 3, then some vertices of the second row of the q^{th} block may also be attacked (depending on the horizontal shift a_{n-1}^q). Finally, if the vertical shift b^q of the q^{th} block equals 1, then some vertices of the $(n-1)^{th}$ row of the q^{th} block may also be attacked (depending on the horizontal shift a_2^q).

Note that $A_V(Y) \cap C_V(Y) = \emptyset$, and $A_V(Y) \cap A_H(X^q) = \emptyset$ for any $X^q \in Y$, *i.e.*, any vertical attack with respect to Y is not a horizontal attack with respect to $X^q \in Y$ and vice versa. In Figure 4.5, red squares represent the vertices of $A_V(Y)$.

The next lemma proves that, from any vertical configuration and against any vertical attack (with respect to this current configuration), there is a possible strategy for the guards that defends against this attack and leads to a (new) vertical configuration. Therefore, starting from any vertical configuration, there is a strategy of the guards that wins against any sequence of vertical attacks.

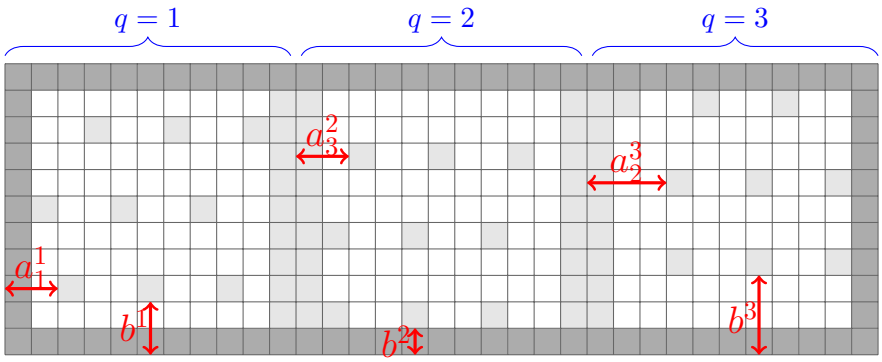


Figure 4.4: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 1, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards.

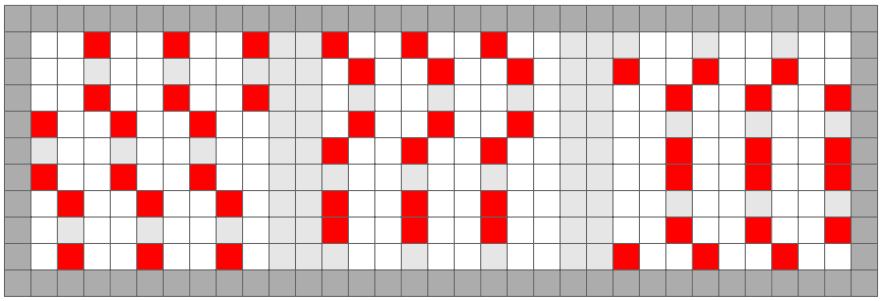


Figure 4.5: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of the non-occupied attackable vertices in red when only vertical attacks are considered. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 1, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards.

Lemma 4.3.4. *For any $Y \in \mathcal{Y}$ and any $v \in A_V(Y)$, there exists $Y' \in \mathcal{Y}$ such that $v \in C_V(Y')$ and configurations $C_V(Y)$ and $C_V(Y')$ are compatible. That is, in one turn, the guards may move from $C_V(Y)$ to $C_V(Y')$ and defend against an attack at v.*

Proof. Let $Y = (X^1, \ldots, X^{\frac{m}{k}})$. Initially, κ_V guards are in a configuration $C_V(Y)$ (see Figure 4.4).

Consider an attack at some vertex $v \in A_V(Y)$. Let us assume that $v = (x_\ell^z(X^z) - 1, y_{w,\ell}^z(X^z))$ for some $1 \leq z \leq \frac{m}{k}$, $1 \leq \ell \leq \frac{n-2}{3}$, and $1 \leq w \leq \frac{k-2}{3}$ (note that if $b^z = 1$,

then $\ell > 1$ since v is not a border vertex). That is, v is a vertex of the z^{th} block that is below the vertex $(x_\ell^z(X^z), y_{w,\ell}^z(X^z))$ which is occupied by a guard.

The cases of attacks at $(x_\ell^z(X^z)+1, y_{w,\ell}^z(X^z))$ (v is above an occupied vertex), $(2, y_{w,n-1}^z(X^z))$ (v is above a border vertex), and $(n-1, y_{w,2}^z(X^z))$ (v is below a border vertex), are similar, by symmetry, to at least one of the two cases below.

The guards will move from the configuration $C_V(Y)$ to a configuration $C_V(Y')$ that defends against the attack at v, *i.e.*, $v \in C_V(Y')$, where $Y' = \{X'^1, \ldots, X'^{\frac{m}{k}}\}$ as defined below.

Intuitively, for the guards to move from the configuration $C_V(Y)$ to a configuration $C_V(Y')$ that defends against this attack at v, all the guards in the block z will shift down except for perhaps the guards on the border vertices (it depends on the value of b^z).

Precisely, by the definition of $C_V(Y)$, there is a guard at $(x_\ell^z, y_{w,\ell}^z)$. There are two cases of how the guards will move in response to the attack, depending on the three possible values of $b^z \in \{1,2,3\}$.

Case i) $b^z \in \{2,3\}$**.** To defend against the attack, all the guards in the block z that contains the attacked vertex except those that occupy border vertices of the block z, shift one vertex downwards. That is, for all $i, j \in \mathbb{N}$ such that $1 \le i \le \frac{n-2}{3}$ and $1 \le j \le \frac{k-2}{3}$, the guard at $(x_i^z, y_{j,i}^z)$ moves to $(x_i^z - 1, y_{j,i}^z)$.

Since the positions of the other guards did not change, the guards occupy a configuration $C_V(Y')$ where $X^p = X'^p$ for all $1 \le p \le \frac{m}{k}$ such that $p \ne z$, and $X'^z = (b'^z, a_1'^z, \ldots, a_{\frac{n-2}{3}}'^z)$ with $a_i'^z = a_i^z$ for all $1 \le i \le \frac{n-2}{3}$, but $b'^z = b^z - 1$.

Case ii) $b^z = 1$**.** To defend against the attack, all the guards in the block z shift one vertex downwards, except those that occupy the vertices of the border of the block z and the guards just above the bottom border of the block. Using the guards on the border of the (whole) grid, the guards just above the bottom border of the block jump to the row just below the top border of the block z.

That is, for all $i, j \in \mathbb{N}$ such that $1 < i \le \frac{n-2}{3}$ and $1 \le j \le \frac{k-2}{3}$, the guard at $(x_i^z, y_{j,i}^z)$ moves to $(x_i^z - 1, y_{j,i}^z)$. Also, the guard at $(2, y_{j,i}^z)$ jumps to $(n-1, y_{j,i}^z)$ which is possible by Lemma 4.2.2 since a total of $\frac{k-2}{3}$ guards jump, $\frac{k-2}{3}+1$ guards occupy each vertex of the border of the grid, and since none of the border guards have to move for any other purpose. Since the positions of the other guards did not change, the guards occupy a configuration $C_V(Y')$ where $X^p = X'^p$ for all $1 \le p \le \frac{m}{k}$ such that $p \ne z$, and $X'^z = (b'^z, a_1'^z, \ldots, a_{\frac{n-2}{3}}'^z)$ with $a_i'^z = a_i^z$ for all $1 \le i \le \frac{n-2}{3}$, but $b'^z = 3$. See Figure 4.6.

□

4.3.3 Diagonal Attacks

The same $n \times m$ strong grid $SG_{n\times m}$, notations, and configurations for the guards used in subsection 4.3.2 will be used here.

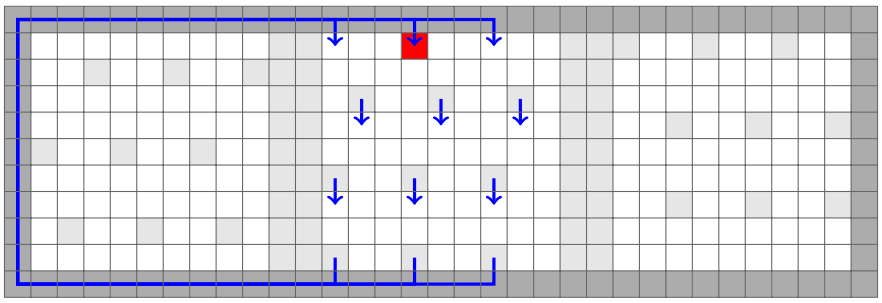


Figure 4.6: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of an attack in Case iii) at the red square. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 1, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack.

In this subsection, we limit the power of the attacker by allowing it to attack only some *diagonal* vertices. For every configuration $C_V(X) \in \mathcal{C}_V$ and for any such attack, we show that the guards may be moved (in one turn) in such a way to defend the attacked vertex and reach a new configuration in $\mathcal{C}_V$.

Diagonal Attacks. Let $Y = (X^1, \ldots, X^{\frac{m}{k}}) \in \mathcal{Y}$ and $C_V(Y) \in \mathcal{C}_V$. Let $A_D(Y) = V(SG_{n\times m}) \setminus (B \cup A_H(Y) \cup A_V(Y))$. That is, $A_D(Y)$ covers all possible attacks that are neither horizontal nor vertical.

A *diagonal attack with respect to* Y is an attack at any vertex in $A_D(Y)$. Note that, for every vertex $v \in A_D(Y)$, there is a guard on a vertex adjacent to v and neither in the same column nor in the same row as v. In Figure 4.7, red squares represent the vertices of $A_D(Y)$.

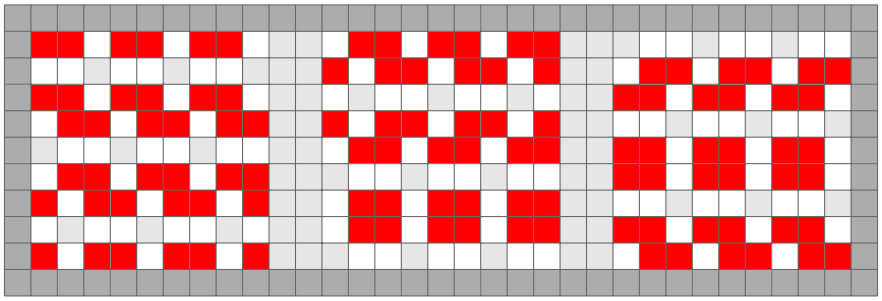


Figure 4.7: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of the non-occupied attackable vertices in red when only diagonal attacks are considered. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 1, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards.

The next lemma proves that, from any vertical configuration and against any diagonal attack (with respect to this current configuration), there is a possible strategy for the guards that defends against this attack and leads to a (new) vertical configuration.

Therefore, starting from any vertical configuration, there is a strategy of the guards that wins against any sequence of diagonal attacks.

Lemma 4.3.5. *For any $Y \in \mathcal{Y}$ and any $v \in A_D(Y)$, there exists $Y' \in \mathcal{Y}$ such that $v \in C_V(Y')$ and configurations $C_V(Y)$ and $C_V(Y')$ are compatible. That is, in one turn, the guards may move from $C_V(Y)$ to $C_V(Y')$ and defend against an attack at v.*

Proof. Let $Y = (X^1, \ldots, X^{\frac{m}{k}})$. Initially, κ_V guards are in a configuration $C_V(Y)$ (see Figure 4.4).

Consider an attack at some vertex $v \in A_D(Y)$. Let us assume that $v = (x^z_\ell(X^z) - 1, y^z_{w,\ell}(X^z) + 1)$ for some $1 \leq \ell \leq \frac{n-2}{3}$, $1 \leq w \leq \frac{k-2}{3}$, and $1 \leq z \leq \frac{m}{k}$ (Note that, if $b^z = 1$, then $\ell > 1$ and if $a^z_\ell = 3$, then $w < \frac{k-2}{3}$ since v is not a border vertex). All other cases are similar by symmetry (see Figures 4.9 and 4.10).

The guards will move from a configuration $C_V(Y)$ to a configuration $C_V(Y')$ that defends against the attack at v, *i.e.*, $v \in C_V(Y')$, where $Y' = \{X'^1, \ldots, X'^{\frac{m}{k}}\}$ as defined below.

Intuitively, for the guards to move from a configuration $C_V(Y)$ to a configuration $C_V(Y')$ that defends against this attack at v, in the block z that contains the attacked vertex, the guards in row x^z_ℓ will move as they would in response to a horizontal attack and a vertical attack but simultaneously, so moving diagonally down and to the right, and the remainder of the guards in the block z will move as they would in response to a vertical attack, so moving down.

In particular, if $b^z = 1$ (there are guards in the row above the bottom border of the block q), the guards in row 2 in the block z will jump to the row below the top border of the block z using the border of the grid (as specified in Lemma 4.3.4). Moreover, if $a^z_\ell = 3$, the guard on vertex $(x^z_\ell, zq - 1)$ jumps to vertex $(x^z_\ell - 1, z(q-1) + 2)$ using the border of the block z. So, a total of at most $\frac{k-2}{3} + 1$ guards jump which is possible (by Lemma 4.2.2) since enough guards are present on each vertex of the border of the grid.

Precisely, after their moves, the guards occupy a configuration $C_V(Y')$ where $X^p = X'^p$ for all $1 \leq p \leq \frac{m}{k}$ such that $p \neq z$, and $X'^z = (b'^z, a'^z_1, \ldots, a'^z_{\frac{n-2}{3}})$ with $a'^z_i = a^z_i$ for all $1 \leq i \leq \frac{n-2}{3}$ such that $i \neq \ell$, but

Case $b^z \in \{2, 3\}$ and $a^z_\ell \in \{1, 2\}$. $a'^z_\ell = a^z_\ell + 1$ and $b'^z = b^z - 1$.

Case $b^z \in \{2, 3\}$ and $a^z_\ell = 3$. $a'^z_\ell = 1$ and $b'^z = b^z - 1$.

Case $b^z = 1$ and $a^z_\ell \in \{1, 2\}$. $a'^z_\ell = a^z_\ell + 1$ and $b'^z = 3$. See Figure 4.8.

Case $b^z = 1$ and $a^z_\ell = 3$. $a'^z_\ell = 1$ and $b'^z = 3$.

□

4.3.4 Upper Bound in Strong Grids

Note that, for any $Y = (X^1, \ldots, X^{\frac{m}{k}}) \in \mathcal{Y}$, $A_D(Y) \cup A_V(Y) \cup \bigcup_{q=1}^{\frac{m}{k}} A_H(X^q) \cup B = V(SG_{n\times m})$. That is, any attack by the attacker in $SG_{n\times m}$ is either an attack at an occupied vertex or a horizontal, vertical or diagonal attack.

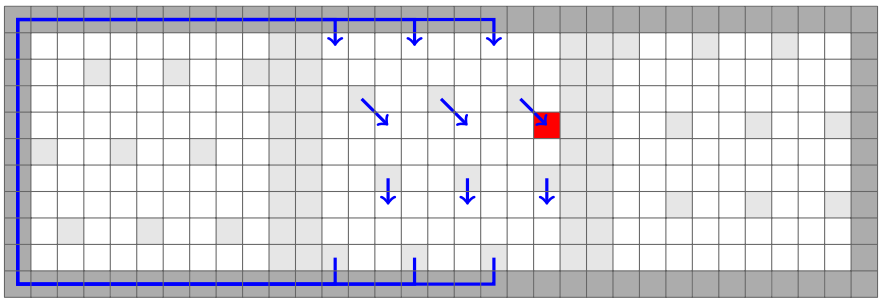


Figure 4.8: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of a diagonal attack at the red square. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 3, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack.

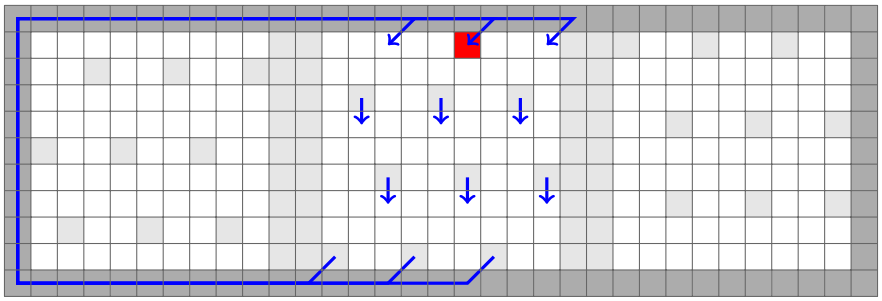


Figure 4.9: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of a diagonal attack at the red square. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 3, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack. To be consistent with the strategy described, the guards in the second row of the middle block move diagonally down and to the left when entering and leaving the border but they may clearly just move vertically down in both instances if they move far enough along the border.

Hence, lemmas 4.3.2, 4.3.4, and 4.3.5 hold for any possible attack, which leads to our main theorem.

Theorem 4.3.6. *For all $n, m \in \mathbb{N}^*$ such that $m \geq n$,*

$$\gamma^{\infty}_{all}(SG_{n\times m}) = \left\lceil \frac{n}{3} \right\rceil \left\lceil \frac{m}{3} \right\rceil + O(m\sqrt{n}) = (1 + o(1))\gamma(SG_{n\times m}).$$

Proof. Let k be the integer closest to $\sqrt{n}$ such that $k - 2 \equiv 0 \pmod 3$.

First, we prove that we can restrict our study to the case when n, m, and k satisfy the hypothesis of the previous lemmas, *i.e.*, $n - 2 \equiv 0 \pmod 3$ and $m \equiv 0 \pmod k$. For this purpose, some guards are placed at each of the vertices of a few columns and rows (and these guards will never move) such that what remains to be protected is an $a \times b$ subgrid H such that $a - 2 \equiv 0 \pmod 3$ and $b \equiv 0 \pmod k$.

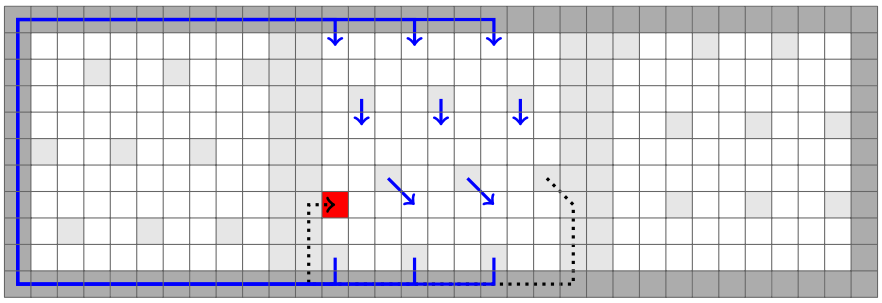


Figure 4.10: $P_{11} \boxtimes P_{33}$ where the squares are vertices. Example of a diagonal attack at the red square. The guards occupy a configuration $C_V(Y)$ where $k = 11$, $Y = (X^1, X^2, X^3)$, $X^1 = (2, 2, 1, 3)$, $X^2 = (1, 1, 3, 2)$, $X^3 = (3, 3, 3, 1)$, there are $\frac{k-2}{3} + 1 = 4$ guards at each square in dark gray, 1 guard at each square in light gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack. The arrow in black is to differentiate between the different guards jumping.

If $n-2 \equiv 0 \pmod 3$, then $a = n$. Otherwise, if $n-2 \equiv 1 \pmod 3$ (resp., 2) then, place one guard at every vertex of the first (resp., the first two) row(s) of $SG_{n\times m}$ and $a = n-1$ (resp., $a = n-2$). Then, place one guard at every vertex of the $x < k$ first columns of $SG_{n\times m}$, such that $b = m - x$ and $b \equiv 0 \pmod k$. Overall, $O(m + kn) = O(m + n\sqrt{n})$ guards have been placed, so proving that $\gamma^{\infty}_{all}(H) = \lceil \frac{a}{3} \rceil \lceil \frac{b}{3} \rceil + O(b\sqrt{a})$ will be sufficient to prove the theorem.

Hence, from now on, let us assume that n and m satisfy $n - 2 \equiv 0 \pmod 3$ and $m \equiv 0 \pmod k$.

Let $Y \in \mathcal{Y}$ be any configuration. The guards initially occupy the configuration $C_V(Y)$. By Lemma 4.3.3, the guards occupy a dominating set. We show that, for an attack at any vertex v, there is $Y' \in \mathcal{Y}$ such that $v \in C_V(Y')$ and $C_V(Y')$ is compatible with $C_V(Y)$.

Let the attacker attack some unoccupied vertex $v \in V(SG_{n\times m})$. As mentioned in subsection 4.3.3, the vertex v is in $A_H(Y)$ or $A_V(Y)$ or $A_D(Y)$ (or already contains a guard since every border vertex contains at least one guard). If $v \in C_V(Y)$, all guards remain idle. Hence, let us assume that $v \notin C_V(Y)$. If $v \in A_H(X^q)$ for some $X^q \in Y$, then the guards in the block q that contains v will respond as in Lemma 4.3.2 (only the guards in the same block and in the same row as v will move, plus some guards on the border of this block if some jump is needed). If $v \in A_V(Y)$, then the guards in the block q that contains v will respond as in Lemma 4.3.4. If $v \in A_D(Y)$, then the guards in the block q that contains v will respond as in Lemma 4.3.5. By Lemma 4.3.2, Lemma 4.3.4, and Lemma 4.3.5, after the attack, the guards occupy a configuration $C_V(Y')$ for some $Y' \in \mathcal{Y}$ and thus, can defend against an infinite sequence of attacks.

The above strategy uses $\kappa_V = \frac{m}{k}(\kappa_H) + 2(\frac{k-2}{3})(m + n - 2)$ guards (see Subsection 4.3.2). Since $\kappa_H = \frac{(n-2)(k-2)}{9} + 2(n+k) - 4$ (see Subsection 4.3.1) and $k = \Theta(\sqrt{n})$, the strategy uses $\kappa_V = \lceil \frac{n}{3} \rceil \lceil \frac{m}{3} \rceil + O(m\sqrt{n})$ guards, which concludes the proof of the theorem. □

4.4 Lower Bound in Strong Grids

So far, the best lower bound for $\gamma^\infty_{all}(SG_{n\times m})$ was the trivial lower bound $\gamma(SG_{n\times m})$. In this section, we slightly increase this lower bound, reducing the gap with the new upper bound of the previous section.

Theorem 4.4.1. *For all $n, m \in \mathbb{N}^*$, $\gamma^\infty_{all}(SG_{n\times m}) = \lfloor \frac{n}{3} \rfloor \lfloor \frac{m}{3} \rfloor + \Omega(n+m)$.*

Proof. $\gamma^\infty_{all}(SG_{n\times m})$ is clearly increasing with n and m, thus, it is sufficient to prove the theorem for $n \equiv 0 \pmod 3$ and $m \equiv 0 \pmod 3$. Hence, let us assume that $n \equiv 0 \pmod 3$ and $m \equiv 0 \pmod 3$.

Note that, if n and m are divisible by 3, there is a unique minimum dominating set of $SG_{n\times m}$ and, in this dominating set, each vertex is dominated by exactly one guard. The idea of the proof is that, in any winning configuration in eternal domination, there are some vertices that are dominated by more than one guard, and/or some guards dominate at most 6 vertices. By double counting, this leads to the necessity of having $\Omega(n+m)$ extra guards compared to the classical domination.

The following claim shows that, whatever be the guards' strategy, at every step, every 4×5 subgrid that includes 5 border vertices must have at least two guards in it or else the attacker wins.

Claim 4.4.2. *Consider any configuration of the guards in $SG_{n\times m}$. If there is a 4×5 subgrid that includes 5 border vertices with only one guard in it, the attacker can win in at most two turns.*

Proof of the claim. W.l.o.g. let the 4×5 subgrid include border vertices from the left column of $SG_{n\times m}$. Also, for some integer $1 \leq x \leq n-4$, let $\{(x,1), \ldots, (x+4,1)\}$ be the 5 border vertices. If there is only one guard in this subgrid, then the guard must be at $(x+2,2)$ in order to prevent the attacker from winning in one turn as otherwise, it is not possible to dominate all the vertices of the subgrid. Then, the attacker attacks $(x+2,3)$ which forces the guard at $(x+2,2)$ to move to $(x+2,3)$ as he is the only guard adjacent to that vertex since, initially, there was only one guard in the 4×5 subgrid. Now the attacker attacks $(x+2,1)$ and wins since every guard is at distance at least 2 from this vertex after the previous moves of the guards since, initially, there was only one guard in the 4×5 subgrid. ◇

In any configuration C, let $x = x(C)$ be the number of 4×5 subgrids with at least one vertex dominated by two guards and $y = y(C)$ be the number of 4×5 subgrids where one guard dominates exactly 6 vertices.

Using the previous claim, it can be proved that:

Claim 4.4.3. *There is $\delta > 0$ such that, for any configuration C of the guards in $SG_{n\times m}$ in any winning strategy for the guards, $x + y = \delta(n+m)$ where $x = x(C)$ and $y = y(C)$ are defined as above.*

Proof of the claim. Consider the subgraph induced by rows 1 through 4 and columns 6 through $m-5$ of $SG_{n\times m}$.

Considering columns 6 through 13, there must exist a 4×5 subgrid that includes 5 border vertices and has a guard in its center column as otherwise, there are no guards in the four center columns of the 8 considered (columns 9 through 12 in this case) which means that $SG_{n\times m}$ is not dominated and hence, this configuration is not part of any winning strategy for the guards. Therefore, by considering the columns eight by eight from the first to the last column in rows 1 through 4 of the subgraph described above, there are at least $\lfloor \frac{m-10}{8} \rfloor$ 4×5 subgrids that fit the profile of the subgrid in Claim 4.4.2. Hence, there are at least two guards in each of these subgrids as otherwise, the attacker wins by Claim 4.4.2. Moreover, since there is a guard in the center column of each of these subgrids, there is at least one vertex in each of these subgrids that is dominated by two guards, unless there is a guard on the border in the center column and the other guard(s) are in row 4. However, in the latter case, the guard on the border in the center column only dominates 6 vertices. By symmetry, this is true for the first and last 4 columns and the topmost 4 rows as well. Therefore, there are at least $2\lfloor \frac{m-10}{8} \rfloor + 2\lfloor \frac{n-10}{8} \rfloor$ subgrids in $SG_{n\times m}$ that fit the profile of the subgrid in Claim 4.4.2. Then, $2\lfloor \frac{m-10}{8} \rfloor + 2\lfloor \frac{n-10}{8} \rfloor \leq x + y$. ◇

Let us consider any winning strategy using k guards. Let x and y be the same as in Claim 4.4.3. At every step, these k guards dominate at most $9k - 3y$ vertices (with multiplicity, *i.e.*, a vertex is counted once for each guard that dominates it). By the definition of x, at least $nm + x$ vertices (with multiplicity) must be dominated. Hence, $9k - 3y \geq nm + x$. It follows that $k \geq \frac{nm}{9} + \frac{x}{9} + \frac{y}{3}$. By Claim 4.4.3, $\frac{x}{9} + \frac{y}{3} = \delta'(n + m)$ for some $\delta' > 0$ and so $k = \frac{nm}{9} + \Omega(n + m)$. □

4.5 At Most One Guard at each Vertex

This section is devoted to proving that the two main results presented thus far are also true for the variant of the eternal domination game where at most one guard may occupy a vertex. The corresponding eternal domination number for this variant will be denoted by $\gamma^{*\infty}_{all}$. This variant is also considered in, *e.g.*, [34, 89, 94].

A generalization of Lemma 4.2.2 will be the key to generalizing Theorem 4.3.6 to this variant of the game. The following definitions are required to properly state this new lemma.

For $t \in \mathbb{N}^*$, the set of vertices of the *t-thick border* of $SG_{n\times m}$ is the set

$$TB_t = \bigcup_{1\leq i\leq n, 1\leq j\leq m, 1\leq \ell\leq t} \{(\ell, j), (n + 1 - \ell, j), (i, \ell), (i, m + 1 - \ell)\}$$

and $TB_0 = \emptyset$. In other words, $TB_1 = B(SG_{n\times m})$ is the border of $SG_{n\times m}$, and $TB_t = TB_{t-1} \cup B(SG_{n\times m} \setminus TB_{t-1})$ for any $t \geq 1$. Essentially, the t-thick border vertices are the vertices of the t leftmost and rightmost columns and the t top and bottom rows of $SG_{n\times m}$.

Recall that $PB = TB_2 \setminus TB_1$ is the set of pre-border vertices. Two vertex-disjoint sets $U, W \subseteq PB$ are said to be *non-overlapping*, if there is a path Q induced only by vertices of PB such that $U \subseteq V(Q)$ and $V(Q) \cap W = \emptyset$.

Let $PB_\alpha = TB_{\alpha+1} \setminus TB_\alpha$ be the pre-border vertices of $SG_{n\times m} \setminus (TB_{\alpha-1})$.

Lemma 4.5.1. *Let $\alpha, \beta \in \mathbb{N}^*$ such that $\beta \leq \alpha$. Let $U, W \subseteq PB_\alpha$ be two non-overlapping subsets of pre-border vertices of $SG_{n\times m} \setminus (TB_{\alpha-1})$ such that $|U| = |W| = \beta$. In any configuration C such that $U \subseteq C$, if the α-thick border of $SG_{n\times m}$ contains one guard at each of its vertices, then β guards may "jump" from U to W in one turn.*

Proof. Let $U = \{u_1, \ldots, u_\beta\}$ and $W = \{w_1, \ldots, w_\beta\}$ where the vertices of U and W are ordered according to the order in which they appear when going clockwise along the cycle induced by PB_α. Because $\alpha \geq \beta$, and U and W are non-overlapping, there exist vertex-disjoint paths $P_1, \ldots, P_\beta$ such that, for any $1 \leq i \leq \beta$, P_i is a path from u_i to $w_{\beta-i+1}$ whose internal vertices are in TB_α (see Figure 4.11 for an example with $\alpha = \beta = 4$). Since each vertex in TB_α contains one guard, there is a guard at each vertex of the paths $P_1, \ldots, P_\beta$ except for at the end vertices $w_1, \ldots, w_\beta$. For the guards to jump from U to W, in one turn, for all $1 \leq i \leq \beta$, each guard on each of the vertices of the path P_i moves to its neighbour in the direction of $w_{\beta-i+1}$. □

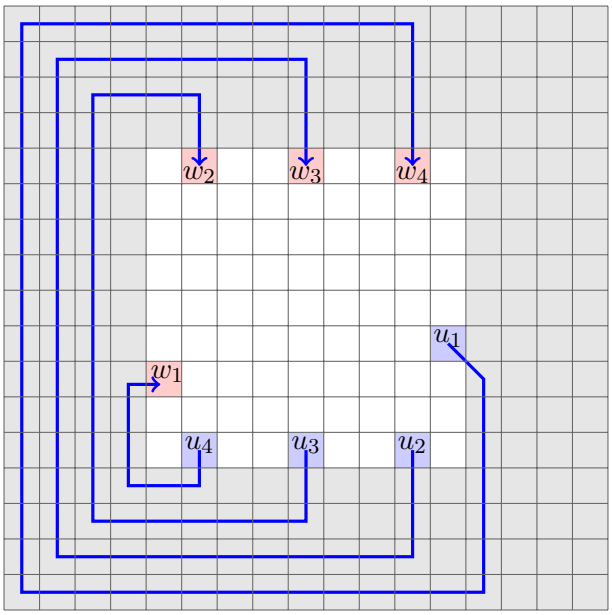


Figure 4.11: $P_{17} \boxtimes P_{17}$ where the squares are vertices. Example of how the guards jump in Lemma 4.5.1. The vertices of $U = \{u_1, u_2, u_3, u_4\}$ are in blue and the vertices of $W = \{w_1, w_2, w_3, w_4\}$ are in red. Note that when $|U| = |W| = \beta = \alpha$. The arrows (in blue) show the vertex-disjoint paths $P_1, \ldots, P_\beta$ that allow the guards to jump from U to W. There is 1 guard at each square in light gray and each vertex of U (in blue), and the white squares contain no guards.

Theorem 4.5.2. *For all $n, m \in \mathbb{N}^*$ such that $m \geq n$,*

$$\gamma(SG_{n\times m}) + \Omega(n+m) = \gamma^{*\infty}_{all}(SG_{n\times m}) = \gamma(SG_{n\times m}) + O(m\sqrt{n}).$$

Proof. The lower bound simply follows from Theorem 4.4.1 and the fact that $\gamma^{*\infty}_{all}(G) \geq \gamma^{\infty}_{all}(G)$ for any graph G.

Let us prove the upper bound. The strategy that we propose follows the same principles as the one of Theorem 4.3.6 but the border vertices occupied by several guards are replaced by several layers of vertices, each one occupied by a single guard.

Let k be the integer closest to $\sqrt{n}$ such that $k-2 \equiv 0 \pmod 3$. Let $SG_{n^* \times m^*}$ be the remaining subgrid that excludes the first and last (topmost and bottommost resp.) $\frac{k-2}{3}$ columns (rows resp.). As in Theorem 4.3.6, we may assume that $n = n^* + 2(\frac{k-2}{3})$ and $m = m^* + 2(\frac{k-2}{3})$ are such that $n^* - 2 \equiv 0 \pmod 3$ and $m^* \equiv 0 \pmod k$. Indeed, otherwise, it is sufficient to "fill" (place one guard at every vertex) at most two rows and at most $k = O(\sqrt{n})$ columns with one guard per vertex (see proof of Theorem 4.3.6).

Hence, from now on, let us assume that n and m satisfy $n - 2(\frac{k-2}{3}) - 2 \equiv 0 \pmod 3$ and $m - 2(\frac{k-2}{3}) \equiv 0 \pmod k$.

Instead of there being $\frac{k-2}{3} + 1$ guards occupying each of the border vertices of the grid like in Theorem 4.3.6, there is one guard at each vertex of the first $\frac{k-2}{3} + 1$ and last $\frac{k-2}{3} + 1$ columns and rows.

The strategy for the guards remains the same as the strategy used in Theorem 4.3.6 except for in the case when a guard or guards have to jump from one vertex to another in which case they move as in Lemma 4.5.1 with a small exception. The exception is that one of the paths between a vertex being jumped from and a vertex being jumped to in a block z, may consist of vertices in one of the columns that forms a border of block z. Figure 4.12 shows an example of a response to a diagonal attack that forces guards to jump and shows that this exception is trivial to deal with. □

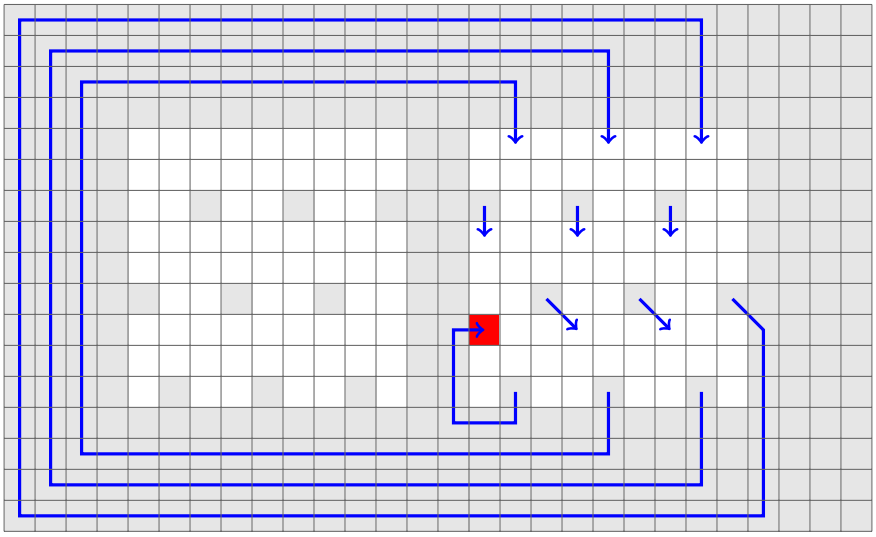


Figure 4.12: $P_{17} \boxtimes P_{28}$ where the squares are vertices. Example of a diagonal attack at the red square when at most one guard may occupy a vertex. There is 1 guard at each square in light gray, and the white squares contain no guards. The arrows (in blue) show the movements of the guards in response to the attack.

4.6 Further Work

Our results in the strong grid leave the open problem of tightening the bounds. Also, for which other grid graphs can our techniques used in obtaining the upper bound be applied? The technique of considering subgrids where only certain attacks are permitted and packing the borders of these subgrids as well as the entire grid with guards should allow to prove that $\gamma_{all}^{\infty}(G) = \gamma(G)+o(nm)$ for many types of $n\times m$ grids. This should be true since, for all Cayley graphs H obtainable from abelian groups, $\gamma_{all}^{\infty}(H) = \gamma(H)$ [72], and many grid graphs can be represented as Cayley graphs obtained from abelian groups which are truncated. This truncation may increase the number of guards needed but our technique should permit the additional $o(nm)$ guards to suffice. Lastly, as mentioned in the introduction, it is known that given a graph G and an integer k as inputs and asking whether $\gamma_{all}^{\infty}(G) \leq k$ is NP-hard in general [21] but the exact complexity of the decision problem is open.

Part III

Identification Games

Chapter 5

Localization Game

5.1 Introduction

In this chapter, the games that generalize the metric and centroidal dimensions of graphs are studied. Recall that Localization (or Identification) problems consist of distinguishing the vertices of a graph $G = (V, E)$ using a smallest subset $R \subseteq V$ of its vertices. For resolving sets, one aims at distinguishing the vertices of a graph by their distances to such a set. Given a graph G, the main problem is to compute a resolving set with minimum size, this minimum being called the *metric dimension* of G [77, 117]. The corresponding decision problem (first shown to be NP-complete in [68]) is NP-complete in planar graphs [50] and in graphs of diameter 2 [64], and W[2]-hard (parameterized by the solution's size) [78]. On the positive side, the problem is FPT in the class of graphs with bounded treelength [23]. Bounds on the metric dimension have also been determined for various graph classes [63].

In this chapter, we address a *sequential variant* of this problem (introduced by Seager in [115] and generalized by us in [c-5], which is joint work with J. Bensmail, D. Mazauric, N. Nisse, and S. Pérennes), which we deal with through the following terminology. Let us consider a graph $G = (V, E)$ where an unknown vertex $t \in V$ hosts a hidden (invisible) and immobile target. *Probing* one vertex $v \in V$ results in the knowledge of the distance between t and v, denoted by $dist_G(v, t)$, which is the length of a shortest path from t to v. Probing a set $R \subseteq V$ of vertices results in the *distance vector* $(dist_G(v, t))_{v \in R}$ and R is *resolving* if no two vertices of G get the same distance vector (by R). The *metric dimension* of G, denoted by $MD(G)$, is then the minimum number of vertices that must be probed simultaneously to immediately (in one step) determine the location t of the target (wherever it is). For instance, in the case of a path, probing one of its ends is sufficient to locate the target, *i.e.*, $MD(P) = 1$ for every path P. Another example is the case of a star (tree with a universal node) with n leaves, denoted by S_n, for which it is necessary and sufficient to probe every leaf but one, *i.e.*, $MD(S_n) = n - 1$.

If less than $MD(G)$ vertices can be probed at once, then it is impossible to locate a target in one step, in which case it is natural to allow more than one probing step. Obviously, if at most $1 \leq k < MD(G)$ vertices can be probed at once, then it is always

feasible to locate an immobile target in $\lceil MD(G)/k \rceil$ steps, simply by considering a smallest resolving set R of G, and probing all vertices of R through successive steps (probing at most k vertices each step). However, there are graphs for which the target can be located much faster (see Section 5.2 or Lemma 5.4.1). In [115], Seager initiated the study of the following sequential locating game: an invisible and immobile target is hidden at some vertex t, and, at every step, one vertex can be probed to retrieve its distance to t, and the objective is to locate t using the minimum number of steps. Seager gave bounds and exact values on this minimum number of steps in particular subclasses of trees (*e.g.*, subdivisions of caterpillars) [115] but left the problem open in trees in general. In this chapter, we study the generalization of this game where $k \geq 1$ vertices can be probed at every step and notably solve the problem for trees.

Precisely, let $k \geq 1$ be an integer and let $G = (V, E)$ be a graph hosting an invisible and immobile target hidden at $t \in V$. A *k-strategy* is a sequence of probing steps, where, at each step, at most k vertices are probed, and at the end of which t is uniquely determined. Note that, in a k-strategy, the choice of the vertices to be probed at some step obviously depends on the result of the previous steps. Let $\lambda_k(G)$ denote the minimum integer h such that there exists a k-strategy for locating the target in G in at most h steps, whatever be the location of the target. Given G and $k, \ell \geq 1$, the Localization problem asks whether $\lambda_k(G) \leq \ell$. We also consider the dual parameter $\kappa_\ell(G)$ defined as the minimum integer h such that there exists an h-strategy for locating the target in G in at most ℓ steps. Note that, for every graph G, the parameter $\kappa_1(G)$ is exactly the metric dimension $MD(G)$ of G, and $\lambda_k(G) \leq \ell$ if and only if $\kappa_\ell(G) \leq k$. We are interested in the complexity of the Localization problem in general graphs and particularly in trees.

Relative distances and centroidal dimension Foucaud *et al.* defined a variant of resolving sets, called *centroidal bases*, where the vertices of a graph must be distinguished by their *relative distances* to the probed vertices [62]. In this setting, given an integer $k \geq 2$, probing a set $B = \{v_1, \ldots, v_k\}$ of vertices results in the *relative-distance vector* $(\delta_{i,j}(t))_{1 \leq i < j \leq k}$ where, for every $1 \leq i < j \leq k$, $\delta_{i,j}(t) = 0$ if $dist_G(t, v_i) = dist_G(t, v_j)$, $\delta_{i,j}(t) = 1$ if $dist_G(t, v_i) > dist_G(t, v_j)$, and $\delta_{i,j}(t) = -1$ otherwise. Intuitively speaking, the relative-distance vector of t indicates which vertices of B are the closest to t, which vertices are the second closest, etc., without indicating the exact distances between v and these vertices. The set B is a *centroidal basis* of G if the relative-distance vectors are distinct for every two vertices of G. The *centroidal dimension* of G, denoted by $CD(G)$, is the minimum size of a centroidal basis of G [62]. Note that $CD(G) \geq 2$ unless G has only one vertex, and that $CD(G)$ is well defined since, clearly, V is a centroidal basis of G. The decision problem associated to the centroidal dimension was shown to be NP-complete, and almost tight bounds on the centroidal dimension of paths have been computed (see [62]). Note that this problem, even for paths, is much more complicated.

Again, sequential variants of the centroidal basis can naturally be defined. The variant where the target is allowed to move was considered in [31]. In this chapter, we also initiate the study of the variant where the target is immobile (introduced by us

in [c-5], which is joint work with J. Bensmail, D. Mazauric, N. Nisse, and S. Pérennes), which, to the best of our knowledge, has not been considered yet. Let $k \geq 2$ be an integer and G be a graph. Let $\lambda_k^{rel}(G)$ denote the minimum integer h such that there exists a k-strategy for locating, through the relative-distance vectors, a hidden immobile target in G in at most h steps, whatever be its location. Given G, k, ℓ, the Relative-Localization problem asks whether $\lambda_k^{rel}(G) \leq \ell$. The dual parameter $\kappa_\ell^{rel}(G)$ is defined as the minimum integer h such that there exists an h-strategy for locating, through the relative-distance vectors, the target in G in at most ℓ steps. Note that, for every graph G, the parameter $\kappa_1^{rel}(G)$ is exactly the centroidal dimension $CD(G)$ of G, and $\lambda_k^{rel}(G) \leq \ell$ if and only if $\kappa_\ell^{rel}(G) \leq k$.

5.1.1 Our Results

All results of this chapter feature in [c-5], which is joint work with J. Bensmail, D. Mazauric, N. Nisse, and S. Pérennes. This chapter is dedicated to the computational complexity of the Localization problem, where one aims at locating an invisible and immobile target in a graph through successive probing steps where the distance vectors are retrieved. To give a first intuition for this problem, we start, in Section 5.2, by providing first, some observations. In Section 5.3, we then show that the Localization problem is polynomial-time solvable when both k and ℓ are fixed parameters but that, in general, the Localization problem is NP-complete when only one of k and ℓ is a fixed parameter. Precisely:

- Let $k \geq 1$ and $\ell \geq 1$ be two fixed integers. Given a graph G as an input, the problem of deciding whether $\lambda_k(G) \leq \ell$ is polynomial-time solvable (in time $n^{O(k\ell)}$) (Theorem 5.3.1).
- Let $k \geq 1$ be a fixed integer. Given a graph G with a universal vertex and an integer $\ell \geq 1$ as inputs, the problem of deciding whether $\lambda_k(G) \leq \ell$ is NP-complete (Theorem 5.3.3).
- Let $\ell \geq 1$ be a fixed integer. Given a graph G with a universal vertex and an integer $k \geq 1$ as inputs, the problem of deciding whether $\kappa_\ell(G) \leq k$ is NP-complete (Theorem 5.3.7).

The proof of Theorem 5.3.1 also yields that the Relative-Localization problem is polynomial-time solvable when $k \geq 2$ and $\ell \geq 1$ are fixed integers. Through modifications, our proofs also yield that the Relative-Localization problem is NP-complete for any fixed $k \geq 2$ (Theorem 5.3.6) or any fixed $\ell \geq 1$ (Theorem 5.3.9).

In Section 5.4, we then focus on the Localization problem in the class of trees. Although we prove that the problem remains NP-complete in the class of trees, surprisingly we show that this hardness only comes from the first probing step. More precisely, we show that, in a tree, the Localization problem becomes polynomial-time solvable after the first step. As a consequence, we design a polynomial-time $(+1)$-approximation algorithm for the problem. To summarize:

- deciding whether $\lambda_k(T) \leq \ell$ is NP-complete for a tree T when both k and ℓ are part of the input (Theorem 5.4.2);
- there exists an algorithm that computes, in time $O(n \log n)$ (independent of k), a k-strategy for locating a target in at most $\lambda_k(T) + 1$ steps in any (possibly edge-weighted) n-node tree T (Theorem 5.4.12);
- deciding whether $\lambda_k(T) \leq \ell$ for any (possibly edge-weighted) n-node tree T can be solved in time $O(n^{k+2} \log n)$ (independent of ℓ) (Theorem 5.4.13).

5.2 Preliminaries

Assuming a vertex of G hosts an invisible and immobile target, recall that a *k-strategy* Φ is a sequence of steps where at most k vertices are probed per step, resulting in the exact localization of the target. As we mainly focus on the Localization problem in this chapter, unless stated otherwise, such a strategy will always deal with the exact distances between the target and the probed vertices. After the s^{th} step of Φ, we denote by $H_s \subseteq V$ the set of vertices that remain as possible locations for the target, *i.e.*, that have not been eliminated at step s. Unless stated otherwise, we thus have $H_0 = V$.

Let us precisely describe the (already mentioned) case of stars because the simple arguments occurring in this case will be used as basic tools for several of the proofs in this chapter. Given a star S_n with n leaves, $\lambda_k(S_n) = \lceil \frac{n-1}{k} \rceil$ and any optimal strategy to locate the target consists of probing every leaf but one. Indeed, if the target is at distance 1 of a leaf, then it is located at the center of the star. Otherwise, if the target is at distance 2 from each of the probed leaves, it must be located in the single unprobed leaf. On the other hand, if at least two leaves have not been probed, there is no way to decide in which unprobed leaf the target is located.

The Relative-Localization problem slightly differs since, in this variant, all leaves must be probed. Indeed, after having probed all leaves but one, a last probe is necessary to decide whether the target occupies the last (unprobed) leaf or the center of the star.

To conclude this section, let us observe the following properties. They will not be used further in this chapter, however, we believe that they are interesting by themselves and give some hints on the difficulty of designing a strategy for locating a target.

First, let us notice that the metric dimension is not closed under isometric subgraphs. That is, there exists a graph G having an isometric subgraph H such that $MD(H) > MD(G)$. Let H be the star S_4 and let G be obtained from H by adding two adjacent vertices u and v each adjacent to a different leaf of H and a vertex w adjacent to one of the two remaining leaves of H. In this case, $MD(H) = 3$ and $MD(G) = 2$ (by probing u and the only leaf in G that is not w). This kind of result is also true for our parameters.

Observation 5.2.1. *There are graphs G having an isometric subgraph H such that $\lambda_k(H) > \lambda_k(G)$.*

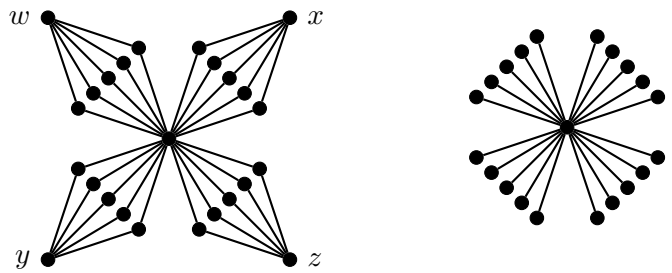


Figure 5.1: A graph G (left) and an isometric subgraph H of G (right).

Proof. Let $k \geq 1$ and $q \geq 2$ and let $H_{k,q}$ be the star $S_{(k+1)q}$ with center c and $(k+1)q$ leaves $v_1, \ldots, v_{(k+1)q}$. Let $G_{k,q}$ be the graph obtained from $H_{k,q}$ by adding q vertices $s_1, \ldots, s_q$ such that s_i is adjacent to $v_{(i-1)(k+1)+1}, \ldots, v_{i(k+1)}$ for every $1 \leq i \leq q$. The graphs $G_{4,4}$ and $H_{4,4}$ are depicted in Fig. 5.1. Clearly, $H_{k,q}$ is an isometric subgraph of $G_{k,q}$, (*i.e.*, distances of $G_{k,q}$ are preserved in $H_{k,q}$).

By the paragraph above, $\lambda_k(H_{k,q}) = \lceil \frac{q(k+1)-1}{k} \rceil$. On the other hand, $\lambda_k(G_{k,q}) \leq \lceil \frac{q}{k} \rceil + 1$ as proved by the following strategy. Probe sequentially every vertex in $s_1, \ldots, s_{q-1}$. If, during the first step, the target is at distance 2 from the probed vertices, then the target is in c. Otherwise, if, at some step $t \leq \lceil \frac{q}{k} \rceil$, the target is at distance 0 from some s_j, the target is at s_j. Finally, if, at step $t \leq \lceil \frac{q}{k} \rceil$, the target is at distance 1 from some s_j, then probe the vertices $v_{(j-1)(k+1)+1}, \ldots, v_{j(k+1)-1}$ to locate the target. $\diamond$

As stated in the introduction, there is a strong connection between the metric dimension and our sequential game. For instance, one k-strategy for locating a target in a graph G consists of considering a minimum resolving set R of G, and probing all vertices of R in $\lceil MD(G)/k \rceil$ steps. In general though, this strategy can be arbitrarily far from being optimal. As an illustration, note that for the graphs $G_{k,q}$ constructed in the proof of Observation 5.2.1, we have $MD(G_{k,q}) = (k+1)q - 1$ (all vertices that are leaves in $H_{k,q}$ must be probed but one), while $\lambda_k(G_{k,q}) \leq \lceil \frac{q}{k} \rceil + 1$.

5.3 Complexity of Localization and Relative-Localization

In this section, we prove that the (Relative) Localization problem is polynomial-time solvable when both k and ℓ are fixed but NP-complete when only one of k and ℓ is fixed. The proof when ℓ is fixed is an almost straightforward reduction from the Metric Dimension problem. In the case when k is fixed, the proof is a much more involved reduction from the 3-Dimensional Matching problem. Our proofs, through several modifications, also apply to the Relative-Localization problem. The proof that the (Relative) Localization problem is in NP is given as a separate claim (Claim 5.3.2) as it is used in all of the NP-completeness proofs.

Theorem 5.3.1. *Let $k \geq 1$ ($k \geq 2$ for the* Relative Localization *problem) and $\ell \geq 1$ be two fixed integers. The (*Relative) Localization *problem is polynomial-time solvable (in time $n^{O(k\ell)}$).*

Proof. Let G be any n-node graph. Let us consider the following tree $\mathcal{T}$ that will be used to represent all possible strategies that probe exactly k vertices per step and last at most ℓ steps in G.

The tree $\mathcal{T}$ is rooted in r and all leaves are at distance 2ℓ from the root. The two types of vertices of $\mathcal{T}$ are labelled by subsets of vertices of $V(G)$. For any vertex $v \in V(\mathcal{T})$ at even distance from r, its label $L(v) \subseteq V(G)$ represents the set of possible locations of the target at this moment. For any vertex $v \in V(\mathcal{T})$ at odd distance from r, its label $L(v) \subseteq V(G)$, of size k, represents the set of vertices that are probed at this moment.

Precisely, $\mathcal{T}$ is defined as follows. Its root r is labelled with $L(r) = V(G)$ (initially, the target may be anywhere). Then, given a vertex $v \in V(\mathcal{T})$ at even distance from r and such that $L(v) = S \subseteq V(G)$, the node v has exactly $\binom{n}{k}$ children labelled by each of the subsets of size k of $V(G)$. Then, for every $Q \in V(G)^k$, let w be the child of v such that $L(w) = Q$. The at most n children of w are defined as follows. Let $(S_1, \cdots, S_q)$ be the partition of S such that, for any $x, y \in S$, the vertices x and y belong to the same S_i if and only if probing the vertices of Q knowing that the target is in S gives the same answer (distance vector) for x and y. Then, w has exactly q children $s_1, \ldots, s_q$ such that $L(s_i) = S_i$ for every $1 \leq i \leq q$. Intuitively, each child of w corresponds to the possible locations of the target in response to the probing of the vertices of Q.

First, note that $|V(\mathcal{T})|$ is polynomial in n when k and ℓ are fixed. Precisely, since $\mathcal{T}$ has at most $(\binom{n}{k}n)^\ell$ leaves (due to the degree of the nodes and the height of $\mathcal{T}$) and all leaves are at distance 2ℓ from r, $|V(\mathcal{T})|$ is upper bounded by $O(2\ell(\binom{n}{k}n)^\ell) = n^{O(k\ell)}$.

Secondly, every strategy (of length ℓ and probing k vertices per turn) is "contained" in $\mathcal{T}$. Indeed, any subtree T' of $\mathcal{T}$ built as follows represents a strategy: start with T' reduced to the root r, then while possible, for any leaf v of T', if v is at an even distance from r, choose a single child of v and add it to T' (this is the probing that the strategy performs in this situation), otherwise, if v is at odd distance from r, add all its children to T'. It is easy to see that, in this way, any strategy, winning (locating the target in at most ℓ turns, wherever it is) or not, can be represented.

By the same reasoning, for every node v at even distance $2(\ell - \ell')$ from r, the subtree of $\mathcal{T}$ rooted in v "contains" all strategies of length ℓ' and probing k vertices per turn, assuming that, initially, the target occupies a vertex in $L(v)$. Let us say that v is *valid* if it contains at least one such winning strategy.

To find out if there is a winning strategy in G, let us proceed by dynamic programming, bottom-up from the leaves of this tree to the root. A leaf v of $\mathcal{T}$ is valid if and only if $L(v)$ is a singleton (indeed, the leaves of $\mathcal{T}$ represent strategies without any probe so the location of the target must be uniquely identified). Then, a vertex v at odd distance from the root is valid if and only if all its children are valid (after a probing, there must be a winning strategy, whatever be the answer). Finally, a vertex v at even distance from the root is valid if and only if at least one of its children is valid. Indeed, the subtree rooted at v contains a winning strategy if, knowing that the target is in $L(v)$, there exists at least one possible probing (one set of k vertices to be probed) that leads toward a winning strategy, whatever be the answer to this probing.

Therefore, there is a winning strategy for G if and only if the root is valid which

can be decided in time $|V(\mathcal{T})| = n^{O(k\ell)}$. □

Claim 5.3.2. *The (*Relative) Localization Problem is in NP.

Proof of the claim. The proof is done for the Localization Problem. The certificate is a k-strategy which can be described by a rooted decision tree T as follows. The nodes of T are labelled by sets of k vertices (the vertices to be probed at a given step) and its edges are labelled by sets of vertices representing the possible locations of the target. Precisely, the root node represents the first k vertices to be probed in G according to the k-strategy. For every node $v \in V(T)$ (but the root), the label $L_e \subseteq V(G)$ of the parent-edge e of v represents the current possible locations of the target and the label $L_v \subseteq V(G)$, $|L_v| \leq k$, is the set of vertices to be probed according to the strategy, given that the target occupies a vertex in L_e. Then, every child w of v corresponds to a possible outcome (after probing the vertices in L_v). That is, L_{vw} is the new set of possible locations after having probed L_v (given that the target was in L_e). Note that, clearly, $L_{vw} \subseteq L_e$. Moreover, we may restrict our attention to *progressive* strategies, *i.e.*, strategies for which, for every non-root vertex v with parent-edge e, and for every child-edge f of v, $L_f \subset L_e$. Indeed, otherwise, the vertices probed in L_v are not relevant and a better choice would be any subset containing at least one vertex of L_e (two vertices of L_e in the case of the Relative Localization Problem, where by definition $k \geq 2$, and this is the only part of the proof that differs between the two problems).

The previous remark shows that we can restrict ourselves to k-strategies represented by rooted trees where all non-leaf nodes have at least two children. Moreover, any such tree representing a winning strategy (a k-strategy that locates the target) has exactly $|V(G)|$ leaves since there is a one-to-one correspondence between a path from the root to a leaf of T with the location of the target in G. A trivial induction on $|V(T)|$ allows to show that any rooted tree with n leaves and where all non-leaf nodes have at least two children, has at most $2n$ nodes. Thus, any winning k-strategy may be encoded polynomially and the Localization Problem is in NP. ◇

5.3.1 When the Number k of Probed Vertices per Step is Fixed

For a fixed integer $k \geq 1$, the k-Probe Localization problem takes a graph G and an integer $\ell \geq 1$ as inputs and asks whether $\lambda_k(G) \leq \ell$. Analogously, for any fixed integer $k \geq 2$, the k-Probe Relative-Localization problem takes a graph G and an integer $\ell \geq 1$ as inputs and asks whether $\lambda_k^{rel}(G) \leq \ell$.

Theorem 5.3.3. *For every $k \geq 1$, the k-*Probe Localization *problem is NP-complete in the class of graphs with a universal vertex.*

Proof. The problem is in NP by Claim 5.3.2. Let us prove it is NP-hard by a reduction from the 3-Dimensional Matching (3DM) problem which is a well known NP-hard problem. The 3DM problem takes a set $\mathcal{X} = I_1 \cup I_2 \cup I_3$ of $3n$ elements ($|I_1| = |I_2| = |I_3| = n$) and a set $\mathcal{S}$ of triples $(x, y, z) \in I_1 \times I_2 \times I_3$ as inputs and asks whether there are n triples of $\mathcal{S}$ that are pairwise disjoint.

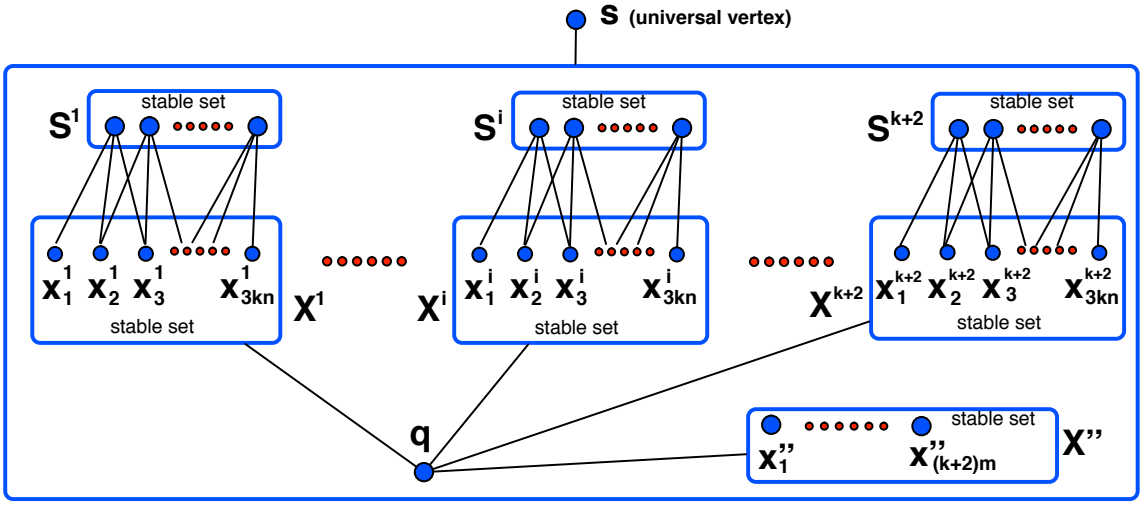


Figure 5.2: Example of a graph G constructed from an instance of 3DM in the proof of Theorem 5.3.3. A thin line between one vertex (blue circle) and one rectangle represents all edges between this vertex and every vertex in the rectangle. The instance of 3DM is encoded by the edges between the vertices in S^i (representing the sets) and the vertices in X^i, for every $1 \leq i \leq k+2$.

Let $k \geq 1$ be a fixed integer and let $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ be an instance of 3DM. First, we may assume that $|\mathcal{X}| = 3kn$ since, if not, it is sufficient to take k disjoint copies of $(\mathcal{X}, \mathcal{S})$. Moreover, we may assume that $m = |\mathcal{S}|$ is such that $2m - 1 \equiv 0 \mod k$ (for instance by adding dummy triples if needed). Let $\mathcal{X} = \{x_1, \ldots, x_{3kn}\}$ and $\mathcal{S} = \{S_1, \ldots, S_m\}$.

From $(\mathcal{X}, \mathcal{S})$, we construct, in polynomial time, a graph $G = (V, E)$ with the vertex-set $V = X \cup X'' \cup S \cup \{s\} \cup \{q\}$ such that (see Fig. 5.2):

- $X = X^1 \cup \cdots \cup X^{k+2}$ with $X^i = \{x^i_1, \ldots, x^i_{3kn}\}$ for every $i \leq k+2$. Each of the vertices x^i_j, for $i \in [\![1, k+2]\!]$, represents the element x_j, for $j \leq 3kn$;
- $X'' = \{x''_1, \ldots, x''_{(k+2)m}\}$;
- $S = S^1 \cup \cdots \cup S^{k+2}$ with $S^i = \{s^i_j, 1 \leq j \leq m\}$ for every $i \in [\![1, k+2]\!]$. Each of the vertices s^i_j, for $i \in [\![1, k+2]\!]$, represents the element S_j, for $j \leq m$.

The edges of G are as follows:

- there is an edge between s and every vertex of $V \setminus \{s\}$;
- there is an edge between q and every vertex of $X \cup X''$;
- for every $j \in [\![1, 3kn]\!]$ and every $g \in [\![1, m]\!]$ such that $x_j \in S_g$, there is an edge between x^i_j and s^i_g for every $i \in [\![1, k+2]\!]$.

Let $p = \frac{m(k+2)-1}{k} \in \mathbb{N}$. We prove the theorem by showing that $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ admits a 3DM if and only if $\lambda_k(G) \leq (k+2)n + p + 1$.

Claim 5.3.4. *If $\mathcal{I}$ admits a 3DM, then $\lambda_k(G) \leq (k+2)n + p + 1$.*

Proof of the claim. Let $Y \subseteq \mathcal{S}$ be a 3DM of $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ (of size $|Y| = kn$). Up to renumbering the sets and the elements, let us assume that $Y = \{S_1, S_2, \ldots, S_{kn}\}$ and assume that $S_i = \{x_{3(i-1)+1}, x_{3(i-1)+2}, x_{3(i-1)+3}\}$ for every $i \in [\![1, kn]\!]$. Note that, because Y is a 3DM of size kn, $\bigcup_{1 \leq i \leq kn} S_i = \mathcal{X}$, (*i.e.*, all elements are covered).

We describe a k-strategy Φ to locate the target in G in at most $(k+2)n+p+1$ steps. The first step of Φ consists of probing only the vertex q. Three cases may occur. Either $H_1 = \{q\}$ (recall that H_s, here and further, denotes the set of vertices that remain possible locations for the target after the s^{th} step) in which case the target is located. Or the target is at distance 2 from q, *i.e.*, $H_1 = S$, in which case Φ sequentially probes every vertex of S but one until the target is located, which takes at most p extra steps. Or the target is at distance 1 from q and $H_1 = X \cup X'' \cup \{s\}$.

Hence, we may assume that $H_1 = X \cup X'' \cup \{s\}$. In this case, Φ proceeds by Phases of at most n steps each. There will be at most $k+2$ such Phases. Intuitively, during Phase $i \leq k+2$, the strategy Φ probes vertices in S^i in such a way that either the target is located at one of the vertices of X^i, or, at the end of the Phase, the target is known not to be in X^i.

Let us assume by induction on $1 \leq i \leq k+2$ and $1 \leq j \leq n$ that, before the j^{th} step of Phase i, if the target has not been located yet, then the set of possible locations for the target is

$$H_{1+(i-1)n+j-1} = \sigma \cup X'' \cup \left\{x^i_{3k(j-1)+1}, \ldots, x^i_{3kn}\right\} \cup \left(\bigcup_{i<y\leq k+2} X^y\right),$$

where $\sigma = \{s\}$ if $i = j = 1$ (or possibly, in the case $k = 1$, if $i = 1$ and $j = 2$), and $\sigma = \emptyset$ otherwise.

This holds for $i = j = 1$. Then, the strategy Φ consists of probing the vertices in $\mathcal{P}_{i,j} = \{s^i_{k(j-1)+1}, \ldots, s^i_{kj}\}$. There are three cases to consider. Before going into the details of the cases, recall that the sets $S_{k(j-1)+1}, \ldots, S_{kj}$ belong to the 3DM Y and so are pairwise disjoint. Hence, by construction of G, for every $a, b \in \mathcal{P}_{i,j}$, we have $(N_G(a) \cap X^i) \cap (N_G(b) \cap X^i) = \emptyset$.

- Either all vertices of $\mathcal{P}_{i,j}$ are at distance 1 from the target. In this case, the target is located at s (this case may only happen for $i = j = 1$ or, possibly, $i = 1$ and $j = 2$ in the case $k = 1$).

- Or exactly one vertex, say $s^i_{k(j-1)+x}$ for $1 \leq x \leq k$, of $\mathcal{P}_{i,j}$ is at distance 1 from the target. Let $y = k(j-1) + x$. In this case, the target must occupy one of $x^i_{3(y-1)+1}, x^i_{3(y-1)+2}, x^i_{3(y-1)+3}$ (the vertices corresponding to the elements that are contained in S_y). The strategy Φ probes two of these vertices, until the target is located in at most two extra steps. Therefore, in this case, the target is located in at most $1 + (i-1)n + j + 2 \leq (k+2)n + p + 1$ steps (since $i \leq k+2$ and $j \leq n$).

- The last case is when all the vertices of $\mathcal{P}_{i,j}$ are at distance 2 from the target. In particular the target cannot occupy a vertex in $U = \{s\} \cup \{x^i_{3k(j-1)+1}, \dots, x^i_{3kj}\}$. And so, if $j < n$, then

$$H_{1+(i-1)n+j} = H_{1+(i-1)n+j-1} \setminus U = X'' \cup \left\{x^i_{3kj+1}, \dots, x^i_{3kn}\right\} \cup \left(\bigcup_{i<y\leq k+2} X^y \right),$$

  hence the induction hypothesis holds for $j+1$. Finally, if $j = n$, then

$$H_{1+in} = H_{1+(i-1)n+n-1} \setminus U = X'' \cup \left(\bigcup_{i+1<y\leq k+2} X^y \right)$$

  and the induction hypothesis holds for $i+1$ and $j = 1$. In this case, Phase $i+1$ starts if $i+1 \leq k+2$.

After the n^{th} step of Phase $k+2$, we get that $H_{1+(k+2)n} = X''$. The strategy Φ ends by sequentially probing every vertex of X'' but one. So, the target can be located in at most p extra steps. Therefore, $\lambda_k(G) \leq (k+2)n + p + 1$. ◇

Claim 5.3.5. *If every 3DM of $\mathcal{I}$ has size strictly less than kn, then $\lambda_k(G) > (k+2)n + p + 1$.*

Proof of the claim. Let us assume that every 3DM of $\mathcal{I}$ has size strictly less than kn. We show that every k-strategy needs at least $(k+2)n + p + 2$ steps to guarantee the localization of the target in G. To avoid technicalities, let us assume that $H_0 = X \cup X''$, *i.e.*, the target is known *a priori* to occupy a vertex in $X \cup X''$. We show that even with this extra assumption (that is not favourable for the target), every k-strategy needs at least $(k+2)n + p + 2$ steps to guarantee the localization of the target.

Let Φ be any k-strategy. First, let us note that, since $H_0 = X \cup X''$ and both q and s are universal for $X \cup X''$, then probing q or s does not bring further information. Therefore, we may assume that Φ never probes q nor s. Let us now describe the information retrieved upon probing vertices in X, X'' or S.

(a) Let $u \in X''$. Note that $dist_G(u, z) = 2$ for every $z \in X \cup X'' \setminus \{u\}$. Therefore, probing u only determines if the target is on u or not, and gives no further information. In other words, probing u only allows to remove u from the set of possible locations.

(b) Let $u \in X^i$ for any $i \leq k+2$. Note that $dist_G(u, z) = 2$ for every $z \in X \cup X'' \setminus \{u\}$. Therefore, similarly, probing u only allows to remove u from the set of possible locations.

(c) Let $u \in S^i$ for any $i \leq k+2$. Let $\{x, y, z\} = N_G(u) \cap X^i$, *i.e.*, x, y, z are the vertices corresponding to the elements contained in the set that corresponds to u. Note that $dist_G(u, z) = 2$ for every $z \in X \cup X'' \setminus \{x, y, z\}$. Therefore, probing u removes at most three vertices, namely x, y, z, from the set of possible locations.

(d) More generally, let $Z \subseteq S^i$ with $|Z| < kn$. Probing all vertices of Z allows to remove $N_G(Z) \cap X^i$, *i.e.*, at most $3|Z|$ vertices, from the set of possible locations.

(e) Finally, let $Z \subseteq S^i$ with $|Z| = kn$. Because $\mathcal{I}$ has no 3DM of size kn, there must be at least two vertices of Z whose neighbourhoods intersect in X^i. That is, $|N_G(Z) \cap X^i| \leq 3kn-1$. Probing all vertices of Z allows to remove at most $3kn-1$ vertices from the set of possible locations.

Let $P \subseteq X \cup X'' \cup S$ be the set of all vertices that have been probed during the $(k+2)n+p+1$ first steps of Φ. We show that, at this point, the set of possible locations for the target still contains at least two vertices and so an extra step is required.

For every $0 \leq j \leq kn$, let α_j be the number of sets S^i that contain exactly $kn - j$ vertices of P. Formally, $\alpha_j = |\{i \mid 1 \leq i \leq k+2, |S^i \cap P| = kn - j\}|$. For every $kn < j \leq m$, let α_j be the number of sets S^i whose exactly j vertices have been probed, *i.e.*, $\alpha_j = |\{i \mid 1 \leq i \leq k+2, |S^i \cap P| = j\}|$. By definition, since $|S^i| = m$ for every $i \leq k+2$:

$$\sum_{0 \leq j \leq m} \alpha_j = k + 2. \tag{5.1}$$

Let $y = |X \cap P|$ be the total number of vertices probed in X and let $x'' = |X'' \cap P|$ be the total number of vertices probed in X''. By definition of y, x'', and the α's, the total number ρ of vertices that have been probed after $(k+2)n + p + 1$ steps satisfies:

$$\rho = y + x'' + \sum_{kn < j \leq m} j\alpha_j + \sum_{0 \leq j \leq kn} (kn - j)\alpha_j. \tag{5.2}$$

Moreover, since at most k vertices can be probed each step:

$$\rho \leq k[(k+2)n + p + 1] \tag{5.3}$$

Note that, by Item (a) above, if $x'' \leq (k+2)m - 2$, then at least two vertices have not been probed and, therefore, are still potential locations for the target (as noticed above, probing a vertex of X'' is the only way to remove it from the set of possible locations). In such a case, another step would be needed to ensure the localization. Therefore, we may assume that $x'' \in \{(k+2)m - 1; (k+2)m\}$.

Let us assume that $x'' = (k+2)m$ (below, we point out the few differences in the case $x'' = (k+2)m - 1$). In that case, all vertices in X'' are removed from the set of possible locations of the target that must be in X. Let $0 < j \leq kn$ and let $i \leq k+2$ such that $kn - j$ vertices have been probed in S^i. By Item (d) above, probing the vertices in S^i removes at most $3(kn - j)$ vertices of X^i (and no other vertices) from the set of possible locations of the target. In other words, it leaves at least $3j$ vertices of X^i as possible locations. Let $i \leq k+2$ such that kn vertices have been probed in S^i. By Item (e) above, probing the vertices in S^i removes at most $3kn - 1$ vertices of X^i (and no other vertices) from the possible locations of the target. In other words, one vertex of X^i is still a possible location.

Summing over all $j \in [\![0, kn]\!]$, the vertices probed in S leave at least $\alpha_0 + \sum_{1 \leq j \leq kn} 3j\alpha_j$ vertices of X as possible locations for the target. To ensure the localization of the target without more steps, only one vertex of X must remain as a possible location (in the case when $x'' = (k+2)m - 1$, *i.e.*, one vertex of X'' is still a possible location, then no vertex of X must remain possible). Since, by Item (b) above, only the y vertices probed in X may remove further vertices from the set of possible locations, it follows that:

$$y + 1 \geq \alpha_0 + \sum_{1 \leq j \leq kn} 3j\alpha_j. \tag{5.4}$$

In the case where $x'' = (k+2)m - 1$, this is $y \geq \alpha_0 + \sum_{1 \leq j \leq kn} 3j\alpha_j$.

We are now ready to show that the above inequalities lead to a contradiction, proving that an extra step is required. For this purpose, let us consider again the total number ρ of vertices that have been probed during the first $(k+2)n + p + 1$ steps.

$$\begin{aligned}
\rho &= y + x'' + \sum_{kn < j \leq m} j\alpha_j + \sum_{0 \leq j \leq kn} (kn - j)\alpha_j && \text{(by Equation (5.2))} \\
&= y + x'' + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + kn \sum_{0 \leq j \leq m} \alpha_j - \sum_{0 \leq j \leq kn} j\alpha_j \\
&= y + x'' + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + kn(k+2) - \sum_{0 \leq j \leq kn} j\alpha_j && \text{(by Equation (5.1))} \\
&= y + (k+2)m + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + kn(k+2) - \sum_{0 \leq j \leq kn} j\alpha_j && \text{(if } x'' = (k+2)m\text{)} \\
&\geq \alpha_0 + \sum_{1 \leq j \leq kn} 3j\alpha_j - 1 + (k+2)m + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + kn(k+2) - \sum_{0 \leq j \leq kn} j\alpha_j \\
& && \text{(by Inequality (5.4)) (if } x'' = (k+2)m\text{)} \\
&= \alpha_0 + \sum_{1 \leq j \leq kn} 3j\alpha_j + pk + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + kn(k+2) - \sum_{0 \leq j \leq kn} j\alpha_j && \text{(by definition of } p\text{)} \\
&= k[n(k+2) + p + 1] + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + \alpha_0 + \sum_{1 \leq j \leq kn} 2j\alpha_j - k \\
&= k[n(k+2) + p + 1] + 2(k+2) - 2\sum_{0 \leq j \leq m} \alpha_j + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j + \alpha_0 + \sum_{1 \leq j \leq kn} 2j\alpha_j - k \\
& && \text{(by Equation (5.1))} \\
&= k[n(k+2) + p + 1] + 4 + \sum_{kn+1 \leq j \leq m} (j - kn)\alpha_j - 2\sum_{kn+1 \leq j \leq m} \alpha_j - \alpha_0 + \sum_{1 \leq j \leq kn} 2(j-1)\alpha_j + k \\
&= k[n(k+2) + p + 1] + 4 + \sum_{kn+2 \leq j \leq m} (j - kn - 1)\alpha_j - \sum_{kn+1 \leq j \leq m} \alpha_j - \alpha_0 + \sum_{1 \leq j \leq kn} 2(j-1)\alpha_j + k \\
&\geq k[n(k+2) + p + 1] + 4 + k - \alpha_0 - \sum_{kn+1 \leq j \leq m} \alpha_j \\
\rho &\geq k[n(k+2) + p + 1] + 2 && \text{(by Equation (5.1))}
\end{aligned}$$

This contradicts Inequality (5.3) and concludes the proof of the claim. ◇ □

Via slight modifications, the previous proof can also be applied to prove the hardness of the k-Probe Relative-Localization problem.

Theorem 5.3.6. *For every $k \geq 2$, the k-Probe Relative-Localization problem is NP-complete in the class of graphs with a universal vertex.*

Proof. The proof of Theorem 5.3.3 applies, except that the strategy designed in Claim 5.3.4 has to start by probing both s and q (instead of only q). In this variant (with relative distances), the localization may require one more step (than with exact distances) in case the target is in $S \cup \{s\}$. The claim still holds since this case (the target in $S \cup \{s\}$) is not the worst case. □

5.3.2 When the Number ℓ of Steps is Fixed

For a fixed integer $\ell \geq 1$, the ℓ-Step Localization problem takes a graph G and an integer $k \geq 1$ as inputs and asks whether $\kappa_\ell(G) \leq k$. In the case where the target must be located through relative distances, the analogous problem ℓ-Step Relative-Localization is defined in the obvious way (but $k \geq 2$ in that case).

Theorem 5.3.7. *For every $\ell \geq 1$, the ℓ-Step Localization problem is NP-complete in the class of graphs with a universal vertex.*

Proof. For $\ell = 1$, the result follows from the fact that computing $\kappa_1(G)$ is exactly the same as computing the metric dimension $MD(G)$ of G, and that the problem of computing the metric dimension is NP-complete in general [50]. So, from now on, let us assume that $\ell \geq 2$.

The problem is in NP by Claim 5.3.2. To prove the NP-hardness let us reduce the Metric Dimension problem (given a graph G and an integer $k \geq 1$, is $MD(G) \leq k$?) restricted to the class of graphs that contain a universal vertex, which is known to be NP-hard [64]. Let G be a graph that contains a universal vertex and k be an integer. We construct, in polynomial time, a graph G' such that $MD(G) \leq k$ if and only if a target hidden in G' can be located in at most ℓ steps by probing at most k vertices per step, *i.e.*, $\kappa_\ell(G') \leq k$.

The construction of G' is as follows. Start from $k(\ell-1)+1$ disjoint copies of G: $G_1, \ldots, G_{k(\ell-1)+1}$. Let v be a universal vertex of G, and for $1 \leq i \leq k(\ell-1)+1$, let v_i denote the copy of v in G_i. Finally, add a universal vertex u to the graph. This results in G'. Clearly, the construction is achieved in polynomial time.

We start by pointing out the following easy claim.

Claim 5.3.8. *For any $1 \leq a \leq k(\ell-1)+1$, if the target is known to occupy a vertex of G_a, then probing a vertex $w \in V(G' \setminus G_a)$ does not remove any vertex in G_a from the set of possible locations.*

Proof of the claim. The vertex u is universal to G_a and, therefore, all vertices of G_a are the same distance from u and every shortest path from w to a vertex of G_a includes u. Thus, any two vertices of G_a cannot be distinguished via their distance to w. ◇

We now prove that $MD(G) \leq k$ if and only if $\kappa_\ell(G') \leq k$.

- First let us assume that $MD(G) \leq k$; we show that $\kappa_\ell(G') \leq k$. Consider the k-strategy where, during step s (for $1 \leq s \leq \ell - 1$), we probe the vertices in $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$.
  - If the target is at one of these vertices, say v_i, then it is located immediately at some step.
  - If the target is at distance 1 from one probed vertex v_i, then it occupies a vertex in the corresponding G_i (unless $k = 1$, in which case the target could also occupy u). Note that, because G has diameter 2, then each of its copies G_i is an isometric subgraph of G'. Hence, any resolving set of size k of G (which exists since $MD(G) \leq k$) is also a resolving set for the vertices of G_i in G'. Probing such a resolving set in G_i during the next step then allows to locate the target. In the case $k = 1$, G_i has at most 3 vertices as otherwise, $MD(G) > 1$ since v_i is a universal vertex in G_i. Then, there are at most two other vertices in G_i that have not been probed (and are not adjacent if there are two, again since otherwise, $MD(G) > 1$), and thus, the target can be located in the next step by probing one of these vertices to distinguish it from u and the other.
  - If the target is at distance 1 from all the $k \geq 2$ v_i vertices, then it is located at u.
  - If at step $\ell - 1$ the target is at distance 2 from the probed vertices, then it is located in $G_{k(\ell-1)+1}$ and can be located at step ℓ since we have assumed that $MD(G) \leq k$ and each G_i is isometric in G'.
- Now we prove the other direction, that is, we show that $MD(G) > k$ implies that $\kappa_\ell(G') > k$. Since there are $k(\ell-1)+1$ copies of G_i and only $k(\ell-1)$ vertices can be probed during the first $\ell-1$ steps, then, on the last step, regardless of the employed strategy, there will always exist a copy, say G_a for some $1 \leq a \leq k(\ell - 1) + 1$, for which no vertices in G_a have been probed. If the target is hidden in G_a, then, by Claim 5.3.8, all the vertices of G_a are still potential locations for the target. The last step is then not sufficient to locate a target hidden in G_a since probing a vertex $w \in V(G' \setminus G_a)$ is useless by Claim 5.3.8, G_a is an isometric subgraph of G', and $MD(G_a) > k$. Hence, $\kappa_\ell(G') > k$. □

A proof establishing the hardness of ℓ-Step Relative-Localization can analogously be obtained by a reduction of the Centroidal Dimension problem.

Theorem 5.3.9. *For every $\ell \geq 1$, the ℓ-Step Relative-Localization problem is NP-complete in the class of graphs with a universal vertex.*

Proof. For $\ell = 1$, the result follows from the fact that $\kappa_1^{rel}(G)$ is exactly the centroidal dimension $CD(G)$ of G, and that computing the centroidal dimension is an NP-complete problem [62]. So let $\ell \geq 2$ be fixed.

The problem is in NP by Claim 5.3.2. To prove its NP-hardness, let us reduce the Centroidal Dimension problem restricted to the class of graphs that contain a universal vertex, which is known to be NP-hard [62]. Let G be a graph that contains

a universal vertex, and $k \geq 2$. We construct, in polynomial time, a graph G' with a universal vertex such that $CD(G) \leq k$ if and only if a target hidden in G' can be located in at most ℓ steps, by probing at most k vertices per step, *i.e.*, $\kappa_\ell^{rel}(G') \leq k$.

The construction of G' is as follows. Start from $k(\ell-1)+1$ disjoint copies of G: $G_1, \ldots, G_{k(\ell-1)+1}$. Let v be a universal vertex of G, and for $1 \leq i \leq k(\ell-1)+1$, let v_i denote the copy of v in G_i. Then, add all the edges so that $v_{k(\ell-1)+1}$ becomes a universal vertex in the whole resulting graph, which is G'.

Claim 5.3.10. *Let $1 \leq a \leq k(\ell-1)+1$, and assume the target is known to occupy, in G', any vertex of G_a. If $CD(G) > k$, then we cannot locate the target in one step by probing k vertices of G'.*

Proof of the claim. Assume k vertices are probed in G_a. Since $CD(G_a) > k$ and G_a is an isometric subgraph of G', there exist at least two vertices $y_1, y_2 \in G_a$ that cannot be distinguished based on the information received. That is y_1 and y_2 have the same relative-distance vector. If any number of the k vertices probed in G_a had instead been replaced by vertices in $G' \setminus G_a$, then the relative-distance vectors of y_1 and y_2 may change but they would still be identical to one another since $v_{k(\ell-1)+1}$ is a universal vertex (and thus, distance 1 from both y_1 and y_2) and a cut vertex which separates all the G_i's. $\diamond$

We are now ready to prove that $CD(G) \leq k$ if and only if $\kappa_\ell^{rel}(G') \leq k$.

- First let us assume that $CD(G) \leq k$. We show that $\kappa_\ell^{rel}(G') \leq k$. Consider the k-strategy where, at step s for $1 \leq s \leq \ell-1$, we probe the vertices in $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$. Then:
  - If the target is closer to one of the vertices in $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$ probed at step s, say $v_{(s-1)k+x}$ for some integer $1 \leq x \leq k$, then the target is at a vertex in $G_{(s-1)k+x}$. Indeed, all the G_is are separated by a cut vertex $v_{k(\ell-1)+1}$ and since $v_{k(\ell-1)+1}$ is universal, it is equidistant from all the vertices of $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$. Note that each G_i is an isometric subgraph of G'. Hence, any centroidal basis of size k of G (which exists since $CD(G) \leq k$) is also a centroidal basis for the vertices of G_i in G'. Probing such a centroidal basis in G_i allows to locate the target during the next step $s+1 \leq \ell$.
  - If the target is equidistant from each of the vertices in $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$ probed at step s, then the target may not be at the vertices in $\{v_{(s-1)k+1}, \ldots, v_{sk}\}$ nor at the vertices of $G_{(s-1)k+1}, \ldots, G_{sk}$. Therefore, if $s < \ell-1$, then $H_s = \{v_{sk+1}, \ldots, v_{(s+1)k+1}\} \cup \bigcup_{0 \leq i \leq k} V(G_{sk+1+i})$. Hence, after $s = \ell-1$ steps, $H_s = V(G_{k(\ell-1)+1})$. Then, since each G_i is an isometric subgraph of G' and $CD(G) \leq k$, probing a centroidal basis in $G_{k(\ell-1)+1}$ allows to locate the target during the next step $s+1 = \ell$.
- Now we prove the other direction, that is, we show that $CD(G) > k$ implies that $\kappa_\ell^{rel}(G') > k$. Whatever be the probing strategy, if, on the last step, there exists a copy, say G_a for some $1 \leq a \leq k(\ell-1)+1$, for which no vertices in G_a have been probed, then there is no way to know at which vertex of G_a the target is located.

Indeed, all G_i's are separated by a cut vertex, so probing a vertex in some G_i provides no information on any other G_j, $j \neq i$. Since there are $k(\ell-1)+1$ copies of G_i and only $k(\ell-1)$ vertices may be probed in the first $\ell-1$ steps, then, on the last step, regardless of the strategy, there will always exist a copy, say G_a for some $1 \leq a \leq k(\ell-1)+1$, for which no vertices in G_a have been probed. According to Claim 5.3.10, the last step is not sufficient to locate the target in G_a. Hence, $\kappa_\ell^{rel}(G') > k$. □

5.4 The Localization Problem in Trees

This section is devoted to the study of the Localization problem in the class of trees. Recall that when $\ell = 1$, the problem is equivalent to the one of determining the metric dimension, which can easily be solved in polynomial time in trees [77, 117]. We first show that when k and ℓ are part of the input, deciding whether $\lambda_k(T) \leq \ell$ for a given tree T is NP-complete. Our reduction actually shows that the difficulty of the problem comes from the choice of the nodes to be probed during the first step. Surprisingly, we show that the first step is actually the only source of hardness. More precisely, our main result is that if the first step is given (intuitively, either given by an oracle or imposed by an adversary), then an optimal strategy (according to this first pre-defined step) can be computed in polynomial time. As a consequence, we design a (+1)-approximation algorithm for the Localization problem in trees and prove that, in contrast with general graphs (Theorem 5.3.3), the k-Probe Localization problem is polynomial-time solvable in the class of trees.

5.4.1 NP-hardness of the First Step

Before proceeding to the proof of the main result of this section, we first need to give an exact formula for λ_k for a particular class of trees. More precisely, let $k \geq 1$ be fixed, and $1 < r \in \mathbb{N}$ be such that $r - 1 \equiv 0 \mod k$. For $1 < n \in \mathbb{N}$, we denote by S_n^r the tree obtained from r copies of S_n (the star with n leaves) by adding one new node c adjacent to the center of each of the r stars.

Lemma 5.4.1. *For every k, r, n as above,*

$$\lambda_k(S_n^r) = \frac{r-1}{k} + \left\lceil \frac{n-1}{k} \right\rceil.$$

Furthermore, $MD(S_n^r) = r(n-1)$.

Proof. For every $1 \leq i \leq r$ and $1 \leq j \leq n$, let c^i denote the center of the i^{th} copy of S_n, denoted by S^i, and let c_j^i denote the j^{th} leaf of the i^{th} copy of S_n. First, we prove that $\lambda_k(S_n^r) \leq \frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil$. Consider the k-strategy Φ where, at each step $1 \leq s \leq \frac{r-1}{k}$, the nodes $c_1^{(s-1)k+1}, \ldots, c_1^{sk}$ are probed. If at step s, one of the probed nodes, say $c_1^{(s-1)k+x}$ for some $1 \leq x \leq k$, is:

- distance 0 from the target, then the target is located at $c_1^{(s-1)k+x}$;

- distance 1 from the target, then the target is located at $c^{(s-1)k+x}$;
- distance 2 from the target and $k = 1$, then the target is located at c or $c_y^{(s-1)k+x}$ for some $2 \leq y \leq n$. The target is then located in a total of at most $s + \lceil \frac{n-1}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by $c_y^{(s-1)k+x}$ and its neighbours which happens to be a star S_n that is also an isometric subgraph of S_n^r;
- distance 2 from the target and $k > 1$, then the target is located at c if it is also distance 2 from the other probed nodes. Otherwise, it is at $c_y^{(s-1)k+x}$ for some $2 \leq y \leq n$. The target is then located in a total of at most $s + \lceil \frac{n-2}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by $c_y^{(s-1)k+x}$ and all its neighbours except for c, which happens to be a star S_{n-1} that is also an isometric subgraph of S_n^r.

If at step $s < \frac{r-1}{k}$ all of the probed nodes are at distance 3 from the target, then the target is located at one of the nodes $c^{sk+1}, \ldots, c^{(s+1)k}$. If at step $s < \frac{r-1}{k}$ all of the probed nodes are at distance 4 from the target, then the target is located at one of the nodes $c_j^{sk+1}, \ldots, c^{(s+1)k_j}$.

If at step $\frac{r-1}{k}$ all of the probed nodes are at distance 3 from the target, then the target is located at c^r. If at step $\frac{r-1}{k}$ all of the probed nodes are at distance 4 from the target, then the target is located at one of the nodes c_j^r. The target is then located in a total of at most $\frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by c^r and all its neighbours except for c which happens to be a star S_n that is also an isometric subgraph of S_n^r.

We now prove that $\lambda_k(S_n^r) > \frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil - 1$. We may assume that the target is on a leaf as this is not a favourable case for it. Consider a k-strategy. Since there are r copies of S_n in S_n^r and at most $k\frac{r-1}{k}$ nodes can be probed during the first $\frac{r-1}{k}$ steps, then, after step $\frac{r-1}{k}$, there will always exist a copy S^a for some $1 \leq a \leq r$ of S_n for which no nodes in S^a have been probed. Assuming the target is in S^a, note that the nodes probed in $S_n^r \setminus S^a$ during the previous steps did not provide any information on its location. Since S^a is a star with n leaves, we require at least $\lceil \frac{n-1}{k} \rceil$ additional steps to locate the target.

The last part of the statement, *i.e.*, $MD(S_n^r) = r(n-1)$, was proved, *e.g.*, in [77, 117]. □

We are now ready to prove that the Localization problem remains NP-complete when restricted to trees.

Theorem 5.4.2. *The* Localization *problem is NP-complete in the class of trees.*

Proof. The problem is in NP by Claim 5.3.2. We now prove its NP-hardness by a reduction from the Hitting Set problem. The inputs are an integer $k \geq 1$, a ground set $B = \{b_1, \ldots, b_n\}$, and a set $\mathcal{S} = \{S_1, \ldots, S_m\}$ of subsets of B, *i.e.*, $S_i \subseteq B$ for every $i \leq m$. The Hitting Set problem aims at deciding if there exists a set $H \subseteq B$ such that $|H| \leq k$ and $H \cap S_i \neq \emptyset$ for every $i \leq m$.

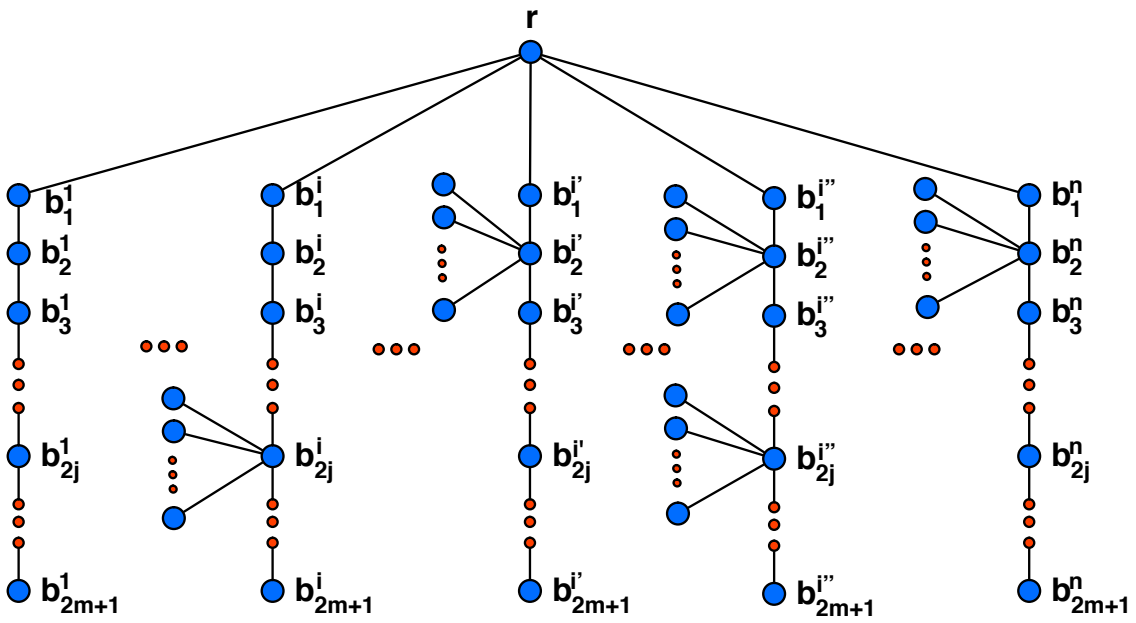


Figure 5.3: Example of a tree T constructed from an instance of Hitting Set in the proof of Theorem 5.4.2. In this example, the elements $b_{i'}, b_{i''}$, and b_n belong to the set S_1 (but not the elements b_1 and b_i) as figured by the three stars at level 2. The elements b_i and $b_{i''}$ belong to S_j (stars at level $2j$) but not the elements $b_1, b_{i'}$, and b_n.

Adding one new element to the ground set and adding this element to one single subset clearly does not change the solution. Therefore, by adding some dummy elements (each one belonging to a single subset), we may assume that all subsets are of the same size σ and that $\sigma - 1 \equiv 0 \mod k$.

Let γ be any integer such that $\gamma - 1 \equiv 0 \mod k$ and $\gamma > n - k - 1$. The instance T of the Localization problem is built as follows (see Fig. 5.3 for an illustration). Start with n node-disjoint paths $B_1, \ldots, B_n$ (called *branches*) of length $2m$, where $B_i = (b_1^i, \ldots, b_{2m+1}^i)$ for each $i \leq n$. Then add one new *root* node r adjacent to b_1^i for all $i \leq n$. For every $1 \leq j \leq m$ and for every $1 \leq i \leq n$ such that $b_i \in S_j$, add γ new nodes adjacent to b_{2j}^i. The subgraph induced by b_{2j}^i and by the γ leaves adjacent to it is referred to as the *star* representing the element i in the set S_j (or representing the set S_j in the branch i). The construction of T is clearly achieved in polynomial time.

Intuitively, it will always be favourable for the target to be located in a leaf of some star because γ is "huge". During the first step of any strategy, the *level* (roughly, the distance to the root) of the target can be identified. Each even level $2j$ corresponds to a set S_j. If, during the first step, one star corresponding to each even level can be eliminated from the possible locations (which corresponds to hit every subset), then the strategy finishes one step earlier than if all subsets cannot be hit (as, in such a situation, all stars would have to be checked).

More formally, we show below that $\lambda_k(T) \leq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$ if and only if there is a hitting set H of size at most k for $(B, \mathcal{S})$. Let us first show that if there is a hitting set H of size at most k for $(B, \mathcal{S})$, then $\lambda_k(T) \leq \ell$ for any $\ell \geq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$. W.l.o.g. (up to renumbering the elements), let us assume that $H = \{b_1, \ldots, b_k\}$ and let us present the corresponding k-strategy. During the first step, the nodes $b^1_{2m+1}, \ldots, b^k_{2m+1}$ are probed. We consider the following cases.

- First, if the target is at distance exactly $2m+1$ from one of (actually from all) the probed nodes, then it is located at r.
- Then, let us assume that the target is at distance strictly less than $2m+1$ from one of the probed nodes, w.l.o.g., that the target occupies a node in the branch B_1 (including the leaves of the stars in this branch). If the target is at odd distance from b^1_{2m+1}, then the target is located since there is a unique node at distance $2h+1$ from b^1_{2m+1} for each $0 \leq h \leq m$. Otherwise, the target is at distance $d = 2(m-h)$ from b^1_{2m+1} for some $0 \leq h < m$ (if $h = m$, then the target is trivially located). If $b_1 \notin S_{h+1}$, then b^1_{2m+2-d} has degree 2 and b^1_{2m+1-d} is the unique node at distance d from b^1_{2m+1} and the target is located. Otherwise, the target may occupy b^1_{2m+1-d} or any leaf adjacent to b^1_{2m+2-d}. By Observation 5.4.4, this can be checked in $\lceil \frac{\gamma}{k} \rceil$ steps by sequentially checking each of these nodes but one. Overall, in this case, the target is located in at most $1 + \lceil \frac{\gamma}{k} \rceil$ steps (including the first one).
- Hence, we may assume that the target is at distance at least $2m+2$ from each of $b^1_{2m+1}, \ldots, b^k_{2m+1}$. Note that, in this case, the target is the same distance from every probed node. Said differently, the information brought by the first step is that the target is at some distance $d \geq 1$ from the root r and not in branches $B_1, \ldots, B_k$.
  - If d is even, then the target can be at $b^{k+1}_d, \ldots, b^n_d$. Indeed, for every $i \leq n$, and any even distance d', there is a unique node at distance d' from r in the branch B_i. By Observation 5.4.4, the target can be located in $\lceil \frac{n-k-1}{k} \rceil$ steps by sequentially checking each of these nodes but one. Overall, it took $1 + \lceil \frac{n-k-1}{k} \rceil$ steps to locate the target.
  - Otherwise, $d = 2j+1$ for some $j \leq m$. Recall that H is a hitting set. In particular, $|S_j \setminus H| < |S_j| = \sigma$. In the worst case, $|S_j \setminus H| = \sigma - 1$ and, w.l.o.g. (up to renumbering), $S_j \setminus H = \{b_{k+1}, \ldots, b_{k+\sigma-1}\}$. In this case, the target can be located at $b^{k+1}_d, \ldots, b^n_d$ or at any leaf adjacent to one of the nodes $b^{k+1}_{2j}, \ldots, b^{k+\sigma-1}_{2j}$, (*i.e.*, the leaves of the stars corresponding to the set S_j in the branches that have not been hit). Then, the strategy continues by sequentially probing the nodes $b^{k+1}_d, \ldots, b^{n-1}_d$. Note that we start by the branches containing the stars that remain to be checked. There are two cases to be considered.
    - Either after checking $b^{k+1}_d, \ldots, b^{k+\sigma-1}_d$ in $\frac{\sigma-1}{k}$ steps (recall that $\sigma - 1 \equiv 0 \mod k$), the target is located to be in some star (this is the case if it is at distance 2 from one probed node). Then, it remains to identify which

leaf of the star is the location of the target. This can be done in $\frac{\gamma-1}{k}$ steps by sequentially checking each of these leaves but one (Observation 5.4.4). Overall, in this case, the target has been located in $1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$ steps.

* Or the target does not occupy a leaf of a star and is located after a total of $1 + \lceil \frac{n-k-1}{k} \rceil$ steps (including the first step).

To conclude, if the minimum size of a hitting set is at most k, then $\lambda_k(T) \leq \ell$ for any $\ell \geq 1 + \max\{\lceil \frac{\gamma}{k} \rceil, \lceil \frac{n-k-1}{k} \rceil, \frac{\sigma-1}{k} + \frac{\gamma-1}{k}\} = 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$ (the last equality holds since $\gamma > n-k-1$ and, since $\sigma - 1 \equiv 0 \mod k$ and $\sigma > 1$, we have $\frac{\sigma-1}{k} \geq 1$).

We now show that if there are no hitting sets of size at most k for $(B, \mathcal{S})$, then $\lambda_k(T) > \ell$ for any $\ell \leq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$. Consider any k-strategy. After the first step, at most k branches have some node that has been probed. These at most k branches correspond to at most k elements of the ground set B and, since all hitting sets of $(B, \mathcal{S})$ have size at least $k+1$, there must be a set that does not contain any of these k elements. W.l.o.g., let $S_1 = \{b_1, \ldots, b_\sigma\}$ be this set. After the first step, let us assume that the target is located at distance 3 from the root (it is possible to decide this *a posteriori* since we are considering a worst case). Then, the target may be located at any leaf of some star corresponding to S_1. More precisely, the target may be at any node in $\{b_3^1, \ldots, b_3^\sigma\}$ or at any leaf adjacent to one of the nodes in $\{b_2^1, \ldots, b_2^\sigma\}$. Actually, the target may also be at other nodes (the third node of other branches), but we can ignore these choices. Even with this additional assumption, we show that the strategy will last for too long.

Indeed, after the first step, the instance becomes equivalent to an instance that consists of a rooted tree whose root has degree σ and each child of the root is adjacent to $\gamma + 1$ leaves, and the target is known to occupy a leaf. By a direct adaptation of Lemma 5.4.1, locating the target takes another $\frac{\sigma-1}{k} + \lceil \frac{\gamma}{k} \rceil$ steps. Overall, locating the target thus requires at least $1 + \frac{\sigma-1}{k} + \lceil \frac{\gamma}{k} \rceil$ steps. Since $\gamma - 1 \equiv 0 \mod k$, then $\lceil \frac{\gamma}{k} \rceil > \frac{\gamma-1}{k}$ and $\lambda_k(T) > 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$. □

5.4.2 A Polynomial-time Algorithm for the Next Steps

The proof of Theorem 5.4.2 shows that, in our reduction, choosing the nodes to be probed during the first step to ensure an optimal strategy is equivalent to finding a minimum hitting set. We show here that this first step is actually the only source of hardness for solving Localization in trees.

The key argument is the following easy remark. Let us consider a tree T where an immobile target is hidden and assume that a single node $r \in V(T)$ is probed. After this single probe, the distance $d \in \mathbb{N}$ between the target and r is revealed. Therefore, from the second step, the instance becomes equivalent to a tree T' (a subtree of T) rooted in r, whose leaves (all of them) are the same distance d from r, and where the target is known to occupy some leaf of T'. We first present an algorithm that computes in polynomial time (independent of k and ℓ) an optimal strategy to locate the target in such instances.

Let $\mathcal{T}$ be the set of rooted trees with all leaves the same distance from the root. Given a rooted tree $(T, r) \in \mathcal{T}$ (in what follows, we omit r when it is clear from the

context), let $\lambda_k^L(T)$ be the minimum integer h such that there exists a k-strategy Φ for locating a target in at most h steps knowing *a priori* that the target occupies some leaf of T. The next claim is one of the key arguments that makes the problem easier in this context. For any node v in a rooted tree (T, r), we denote by T_v the subtree rooted at v.

Claim 5.4.3. *Let $(T, r) \in \mathcal{T}$ be a tree rooted in r and v be a child of r. If the target is known to occupy a leaf of T, then probing any node in T_v allows to learn if the target occupies a leaf of T_v or a leaf of $T \setminus T_v$.*

Proof of the claim. Let d be the distance between r and the leaves of T. Let w be any node of T_v and let d' be the distance between w and r. The claim follows from the fact that the target occupies a leaf of T_v if and only if its distance to w is strictly less than $d + d'$. ◇

Let $T \in \mathcal{T}$ be a tree rooted in r and v be a child of r, and let us assume that the secret location of the target is some leaf of T_v. Note that $(T_v, v) \in \mathcal{T}$. Let us assume that T_v is not a path and let s be the first step of an optimal strategy Φ in T that probes some node of T_v (such a step s must exist since otherwise the target would never be located in T_v). By Claim 5.4.3, it is sufficient to probe a single node of T_v to learn whether the target occupies a leaf of T_v. Then, applying an optimal strategy ϕ_v in T_v will locate the target in a total of $s+\lambda_k^L(T_v)-1$ steps if the first step of ϕ_v only requires probing a single vertex of T_v and $s + \lambda_k^L(T_v)$ steps otherwise. So, it may be possible to do better. Indeed, probing several nodes of T_v during the s^{th} step of Φ may serve not only to locate the target in T_v but also to "play" the first step of Φ_v. Doing so, the strategy will take only $s + \lambda_k^L(T_v) - 1$ steps. Let $v_1, \ldots, v_{d^*}$ denote the children of r. So, elaborating, an optimal strategy will consist of doing a tradeoff between probing one single node in several of the T_{v_i}'s (and locating "quickly" in which subtree T_{v_i} the target is hidden, since several of them are considered simultaneously) and probing more nodes in some of the T_{v_i}'s in order to get a head start for the strategy in the case the T_{v_i} hosting the target is identified.

For any tree T, let $\pi(T)$ be the minimum integer $q \leq k$ such that there exists a k-strategy for locating a target in T in at most $\lambda_k^L(T)$ steps, knowing *a priori* that the target occupies some leaf of T, and such that q nodes are probed during the first step.

To illustrate this need of a tradeoff and the importance of π, let us consider the example depicted in Fig. 5.4. The root r of T has eight children $v_1, \ldots, v_8$ with the pairs $(\lambda_k^L(T_{v_i}), \pi(T_{v_i}))$ being $(4, 2)$, $(4, 1)$, $(3, 3)$, $(3, 3)$, $(2, 2)$, $(2, 2)$, $(1, 1)$, and $(0, 0)$, respectively. Let $k = 4$. Here, the target can be located in at most four steps, through the following strategy.

- Step 1. The probed nodes are those labeled ① in Fig. 5.4, that is, two nodes of T_{v_1}, one node of T_{v_2}, and one node of T_{v_3}. If the target occupies some leaf of T_{v_1} or T_{v_2}, then there is a strategy for locating the target in at most $\lambda_k^L(T_{v_1}) - 1 = \lambda_k^L(T_{v_2}) - 1 = 3$ extra steps because $\pi(T_{v_1})$ ($\pi(T_{v_2})$, resp.) nodes of T_{v_1} (T_{v_2}, resp.) have been probed. If the target occupies some leaf of T_{v_3}, then there is a strategy for locating the target in at most $\lambda_k^L(T_{v_3}) = 3$ extra steps (that is a total of four steps). Thus, assume that the target occupies a leaf of some subtree T_{v_i}, $4 \leq i \leq 8$.

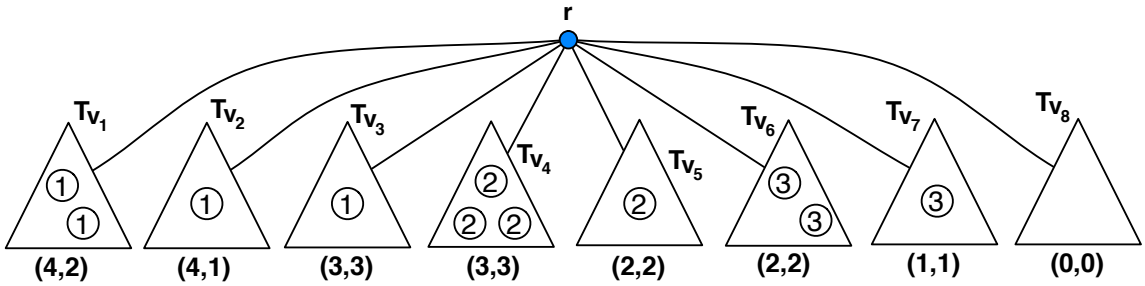


Figure 5.4: A tree $(T, r) \in \mathcal{T}$ rooted at r. The eight children of r are $v_1, \ldots, v_8$. The pair $(\lambda_k^L(T_{v_i}), \pi(T_{v_i}))$ for each T_{v_i} is written below the corresponding subtree.In the figure, one (two, three, resp.) ⓧ in a subtree corresponds to one (two, three, resp.) node (nodes) of this subtree being probed during step x.

- Step 2. The probed nodes are those labeled ② in Fig. 5.4, that is, three nodes of T_{v_4} and one node of T_{v_5}. If the target occupies some leaf of T_{v_4} or T_{v_5}, then using similar arguments to those above, we can show there is a strategy for locating the target in at most two extra steps (that is a total of four steps). Thus, assume that the target occupies a leaf of T_{v_6}, T_{v_7} or T_{v_8}.
- Step 3. The probed nodes are those labeled ③ in Fig. 5.4, that is, two nodes of T_{v_6} and one node of T_{v_7}. Again, if the target occupies some leaf of T_{v_6} or T_{v_7}, then, with at most one extra step, the target is located. Otherwise, the target is on T_{v_8} and there is no need for an extra step.

Let $(T, r) \in \mathcal{T}$ be a tree rooted in r and let $v_1, \ldots, v_{d^*}$ be the children of r. From previous arguments, the computation of an optimal strategy for T consists of determining, for each subtree T_{v_i} $(1 \leq i \leq d^*)$, the first step for which a node of T_{v_i} will be probed (if the target has not been located in a different subtree at a previous step). If one node is probed during this step, then $\lambda_k^L(T_{v_i})$ extra steps are needed if the target occupies some leaf of T_{v_i} (unless $\pi(T_{v_i}) = 1$ in which case $\lambda_k^L(T_{v_i}) - 1$ extra steps are needed). Furthermore, if we want to locate the target in at most $\lambda_k^L(T_{v_i}) - 1$ extra steps (if the target occupies some leaf of T_{v_i}), then $\pi(T_{v_i})$ nodes of T_{v_i} must be probed during this step. Algorithms 1 and 2 compute such an optimal strategy for a tree in $\mathcal{T}$ in polynomial time. We first describe their behaviour, before focusing on proving their correctness.

Description of Algorithm 1 The main algorithm $\mathcal{A}_1(k, (T, r))$ takes an integer $k \geq 1$ and a rooted tree $(T, r) \in \mathcal{T}$ as inputs and computes $(\lambda_k^L(T), \pi(T))$ and a corresponding k-strategy. It proceeds bottom-up by dynamic programming from the leaves to the root. Precisely, let $v_1, \ldots, v_{d^*}$ be the children of r. For any $1 \leq i \leq j \leq d^*$, let $T[i] = T_{v_i}$ be the subtree rooted at v_i, and let $T[i, j] = \{r\} \cup T_{v_i} \cup \ldots \cup T_{v_j}$ $(T[i, j] = \emptyset$ if $i > j)$. To lighten the notations, let us set $\lambda_i = \lambda_k^L(T[i])$ and $\pi_i = \pi(T[i])$ for every $1 \leq i \leq d^*$. Assume that, $(\Lambda, \Pi) = (\lambda_i, \pi_i)_{1 \leq i \leq d^*}$ have been computed recursively and sorted in non-increasing lexicographical order. Then, $\mathcal{A}_2(k, (T, r), (\Lambda, \Pi))$, described in Algorithm 2,

Algorithm 1 $\mathcal{A}_1(k,(T,r))$

Require: An integer k and a tree $(T,r) \in \mathcal{T}$ rooted in r with children $v_1,\dots,v_{d^*}$
Ensure: $(\lambda_k^L(T),\pi(T))$
1: **if** (T,r) is a rooted path **then**
2: **return** $(0,0)$
3: **for** $i=1$ to d^* **do**
4: Let $(\lambda_i,\pi_i)=\mathcal{A}_1(k,(T[i],v_i))$ *// recursive calls to Algorithm 1*
5: Sort the $(\lambda_i,\pi_i)_{1\le i\le d^*}$ in non-increasing lexicographical order
6: **return** $\mathcal{A}_2(k,(T,r),(\lambda_i,\pi_i)_{1\le i\le d^*})$ *// call to Algorithm 2*

Algorithm 2 $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$

Require: An integer k and a tree $(T,r) \in \mathcal{T}$ rooted in r with $v_1,\dots,v_{d^*}$ such that $(\Lambda,\Pi) = (\lambda_i,\pi_i)_{1\le i\le d^*}$ is sorted in non-increasing lexicographical order
Ensure: $(\lambda_k^L(T),\pi(T))$
1: $l \leftarrow 1$, $p \leftarrow k$, $d \leftarrow d^*$
2: **if** $T[d^*]$ is a rooted path **then**
3: $d \leftarrow z$ where $0 \le z < d^*$ is the smallest integer such that $T[z+1]$ is a rooted path
4: $l \leftarrow 1 + \left\lceil \frac{d^*-d-1}{k} \right\rceil$ *// $l \leftarrow 1+\lambda_k^L(T[d+1,d^*])$ (Lem. 5.4.5)*
5: $p \leftarrow k + k\left(\left\lceil \frac{d^*-d-1}{k}\right\rceil - \left\lceil \frac{d^*-d-1}{d^*-d}\right\rceil\right) - (d^*-d-1)$ *// $p \leftarrow k-\pi(T[d+1,d^*])$ (Lem. 5.4.5)*
6: **for** $i=d$ down to 1 **do**
7: **if** $p=0$ or $l < \lambda_i+1$ **then**
8: $p \leftarrow k$, $l \leftarrow \max(l+1,\lambda_i+1)$
9: $\alpha \leftarrow \pi_i - (\pi_i-1)\lceil (l-(\lambda_i+1))/l \rceil$ *// $\alpha=\pi_i$ if, in Line 7, $l<\lambda_i+1$, and $\alpha=1$ otw.*
10: **if** $\alpha \le p$ **then**
11: $p \leftarrow p-\alpha$
12: **else**
13: $p \leftarrow k-1$, $l \leftarrow l+1$ *// $l=1+\lambda_k^L(T[i,d^*])$; $p=k-\pi(T[i,d^*])$ (Lem. 5.4.9)*
14: **return** $(l-1,k-p)$

takes the integer $k \ge 1$, the rooted tree $(T,r) \in \mathcal{T}$, and the sorted tuple (Λ,Π) as inputs and computes $(\lambda_k^L(T),\pi(T))$ and a corresponding strategy.

Description of Algorithm 2 We now informally describe $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$. The first part, from Lines 2 to 5, deals with the subtrees $T_{v_{d+1}},\dots,T_{v_{d^*}}$ that are rooted paths (T_{v_i}'s being paths rooted at one of their two ends, while the second end is a leaf). In other words, these Lines deal with the T_{v_i}'s such that $(\lambda_i,\pi_i)=(0,0)$. Indeed, this case is somehow pathologic, and needs to be treated separately.

The second part (from Line 6) of Algorithm 2 goes as follows. Informally, $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$ recursively builds, for $i=d$ down to 1, an optimal k-strategy Φ for $T[i,d^*]$ from an optimal k-strategy Φ' of $T[i+1,d^*]$ and from an optimal k-strategy Φ'' of $T[i]$ (the latter one being given as input through (λ_i,π_i)). In other words, $(\lambda_k^L(T[i,d^*]),\pi(T[i,d^*]))$ is computed from $(\lambda_k^L(T[i+1,d^*]),\pi(T[i+1,d^*]))$, and

(λ_i, π_i). For every $1 \leq i \leq d+1$, let l_i (resp., p_i) denote the value of l (resp. of p) just before the $(d+2-i)^{\text{th}}$ iteration of the **for** loop (so, l_1 and p_1 are the final values of l and p).Intuitively, let us assume that an optimal strategy for $T[i+1, d^*]$ has been computed, that it takes at most $l_{i+1} - 1$ steps, and that it requires a minimum of $k - p_{i+1} = \pi(T[i+1, d^*])$ nodes to be probed during its first step.Roughly, there are five cases to consider.

- If $\pi_i \leq p_{i+1}$ and $\lambda_i = l_{i+1} - 1$, then the strategy Φ follows Φ' but, in addition, probes π_i nodes of $T[i]$ during its first step. If the target is in $T[i]$, then Φ follows Φ'' (and takes a total of at most λ_i steps), otherwise, it proceeds as Φ' (and takes a total of at most $l_{i+1} - 1$ steps). We get $l_i = l_{i+1}$ and $p_i = p_{i+1} - \pi_i$.

- Else, if $\pi_i > p_{i+1} > 0$ and $\lambda_i = l_{i+1} - 1$, then the first step of Φ probes a unique node in $T[i]$. If the target is in $T[i]$, then Φ follows Φ'' (and takes a total of at most $\lambda_i + 1$ steps). Otherwise, it proceeds as Φ' (and takes a total of at most l_{i+1} steps). We get $l_i = l_{i+1} + 1$ and $p_i = k - 1$.

- Else, if $p_{i+1} = 0$ and $\lambda_i \leq l_{i+1} - 1$, then the first step of Φ probes a unique node in $T[i]$. If the target is in $T[i]$, then Φ follows Φ'' (and takes a total of at most $\lambda_i + 1$ steps). Otherwise, it proceeds as Φ' (and takes a total of at most l_{i+1} steps). We get $l_i = l_{i+1} + 1$ and $p_i = k - 1$.

- Else, if $\lambda_i < l_{i+1} - 1$ and $p_{i+1} > 0$, then the strategy Φ follows Φ' but, in addition, probes one node of $T[i]$ during its first step. If the target is in $T[i]$, then Φ follows Φ'' (and takes a total of at most $\lambda_i + 1$ steps), otherwise, it proceeds as Φ' (and takes a total of at most $l_{i+1} - 1$ steps). We get $l_i = l_{i+1}$ and $p_i = p_{i+1} - 1$.

- Finally, if $(\lambda_i > l_{i+1} - 1)$, then the strategy Φ probes π_i nodes in $T[i]$ during the first step. If the target is in $T[i]$, then Φ follows Φ'' (and takes a total of at most λ_i steps), otherwise, it proceeds as Φ' (and takes a total of at most l_{i+1} steps). We get $l_i = \lambda_i + 1$ and $p_i = k - \pi_i$.

Correctness and complexity of Algorithms 1 and 2 We start by proving the correctness of the two main parts of Algorithm 2, *i.e.*, that of the peculiar case (Lines 2 to 5) and of the general case (from Line 6).

First, we consider the first part of Algorithm 2. Lemma 5.4.5 below proves that Lines 2 to 5 compute $(\lambda_k^L(T[v_{d+1}, d^*]), \pi(T[v_{d+1}, d^*]))$. Let us recall the following observation that is easy to see.

Observation 5.4.4. *For every star S_n with n leaves, $\lambda_k(S_n) = \lceil \frac{n-1}{k} \rceil$.*

We define $\mathcal{S} \subset \mathcal{T}$ as the set of subdivided rooted stars S, (*i.e.*, trees with at most one node of degree at least 3) with all leaves the same distance from the root, where the root of S is the (unique) node with degree at least 3 or one of the two ends of S is a path.

Lemma 5.4.5. *For every subdivided rooted star $S \in \mathcal{S}$ with d leaves, $\lambda_k^L(S) = \lceil \frac{d-1}{k} \rceil$ and $\pi(S) = -k(\lceil \frac{d-1}{k} \rceil - \lceil \frac{d-1}{d} \rceil) + (d-1)$.*

Proof. The strategy consists of sequentially probing each leaf of S but one. Either the target will be probed at some step, or it must be in the unique leaf that has not been probed. During the first step, $\pi(S)$ leaves are probed, and exactly k leaves are probed during every other step. Such a strategy lasts for $\lceil \frac{d-1}{k} \rceil$ steps.

For any strategy using less than $\lceil \frac{d-1}{k} \rceil$ steps, the nodes of at most $k(\lceil \frac{d-1}{k} \rceil - 1) \leq d-2$ branches have been probed. Hence, there are at least two branches of S for which no nodes have been probed and so it is not possible to decide which one of these branches is occupied by the target.

Similarly, it can be checked that, for any strategy using at most $\lceil \frac{d-1}{k} \rceil$ steps and probing less than $\pi(S)$ nodes during the first step, there are at least two branches of S for which no nodes have been probed. To be convinced of that point, notice that $\pi(S) = -k(\lceil \frac{d-1}{k} \rceil - \lceil \frac{d-1}{d} \rceil) + (d-1)$ is equivalent to:

- $\pi(S) = 0$ if $d - 1 = 0$;
- $\pi(S) = k$ if $d - 1 > 0$ and $(d-1) \mod k = 0$;
- $\pi(S) = (d-1) \mod k$ otherwise.

This concludes the proof. □

We now focus on proving the correctness of the second part of Algorithm 2, which is mainly done in Lemma 5.4.9 below. We first introduce three easy observations, whose proofs are omitted.

Observation 5.4.6. *Let $(T, r) \in \mathcal{T}$ be a rooted tree. Then, $\lambda_k^L(T) = 0$ if and only if T is a rooted path, and $\pi(T) = 0$ if and only if T is a rooted path.*

Although the following observation (closedness of λ_k under subtree) does not hold in general graphs (see Observation 5.2.1), it can easily be seen that this holds in the case of trees.

Observation 5.4.7. *For any tree T and any subtree T' of T, $\lambda_k^L(T') \leq \lambda_k(T') \leq \lambda_k(T)$ and $\lambda_k^L(T') \leq \lambda_k^L(T)$.*

The next observation is obvious (indeed, to prove it, just note that the first step probing a single vertex can simply be ignored) but will be quite useful in what follows.

Observation 5.4.8. *For any tree T, there exists a k-strategy for locating the target in at most $\lambda_k(T) + 1$ steps (resp., in at most $\lambda_k^L(T) + 1$ steps if the target is known to occupy a leaf) and that probes a single arbitrary node during its first step.*

We are now ready to prove the next result, which essentially proves that the second part of Algorithm 2 is correct. That is, we prove, provided the T_{v_i}'s are sorted in non-increasing lexicographical order (over (λ_i, π_i)), that the strategy Φ described earlier is optimal for $T[i, d^*]$, that is, it computes $(\lambda_k^L(T[i, d^*]), \pi(T[i, d^*]))$.

Lemma 5.4.9. *For every $1 \leq i \leq d+1$, we have $\lambda_k^L(T[i, d^*]) = l_i - 1$ and $\pi(T[i, d^*]) = k - p_i$.*

Proof. The proof is by induction on $d+1-i \leq d+1$. For $i = d+1$, there are two cases to be considered.

- If $d = d^*$ (*i.e.*, the condition on Line 2 is not satisfied), then, before the first iteration, $l_{d+1} = 1, p_{d+1} = k$ and $T[d+1, d^*] = \emptyset$, and so $\lambda_k^L(\emptyset) = l_{d+1} - 1 = 0$ and $\pi(\emptyset) = k - p_{d+1} = 0$. So the induction hypothesis is satisfied for $i = d+1$.

- Otherwise, $d < d^*$ and $T_{v_{d+1}}, \ldots, T_{v_{d^*}}$ are rooted paths. That is, $T[d+1, d^*] \in \mathcal{S}$. Then, the induction hypothesis for $i = d+1$ is satisfied by Lemma 5.4.5 and Lines 2 to 5 of Algorithm 2.

Let us assume that the induction hypothesis holds for $1 < i+1 \leq d+1$. That is, at the end of the $(d-i)^{\text{th}}$ iteration of the **for** loop, $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$. We will prove that it is also true after the next iteration of the **for** loop, *i.e.*, $\lambda_k^L(T[i, d^*]) = l_i - 1$ and $\pi(T[i, d^*]) = k - p_i$.

It is very important to note that Lines 2 and 3 imply that $\lambda_i > 0$ and $\pi_i > 0$, for every $1 \leq i \leq d$. We consider five cases depending on the values of $p_{i+1}, \pi_i, \lambda_i$, and l_{i+1}.

- Case $0 < \pi_i \leq p_{i+1}$, $l_{i+1} = \lambda_i + 1$.

  By the induction hypothesis, $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$. Because the value of l at the beginning of this iteration of the **for** loop is $l_{i+1} = \lambda_i + 1$, then $\alpha = \pi_i$. Then, since $\pi_i \leq p_{i+1}$, we get that p becomes $p - \alpha = p_{i+1} - \pi_i$ and l is not modified. Hence, $l_i = l_{i+1}$ and $p_i = p_{i+1} - \pi_i$.

  We now prove that $\lambda_k^L(T[i, d^*]) = l_{i+1} - 1$ and $\pi(T[i, d^*]) = k - p_{i+1} + \pi_i$. By Observation 5.4.7, we have $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1$. To prove that $\lambda_k^L(T[i, d^*]) \leq \lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1$, it is sufficient to describe a strategy Φ for $\lambda_k^L(T[i, d^*])$ with a total of at most $l_{i+1} - 1$ steps.

  Let Φ' be an optimal strategy for $T[i+1, d^*]$ probing at most $\pi(T[i+1, d^*])$ nodes during the first step. Also, let Φ'' be an optimal strategy for $T[i]$ probing at most π_i nodes during the first step. The first step of Φ consists of probing π_i nodes of $T[i]$ (as Φ'') and $\pi(T[i+1, d^*]) = k - p_{i+1}$ nodes of $T[i+1, d^*]$ (as Φ'). By assumption, $\pi_i \leq p_{i+1}$, and, by the induction hypothesis, $\pi(T[i+1, d^*]) = k - p_{i+1}$, so $\pi_i + \pi(T[i+1, d^*]) \leq k$ and at most k nodes are probed. By Claim 5.4.3, this first step allows to decide if the target is in $T[i]$ or not (in the latter case, it is in $T[i+1, d^*]$). If the target is in $T[i]$, then Φ continues by following the strategy Φ'' in $T[i]$, which will locate the target in at most $\lambda_i - 1 = l_{i+1} - 2$ extra steps. Otherwise (the target is in $T[i+1, d^*]$), Φ continues by following the optimal strategy Φ' for $T[i+1, d^*]$ which will locate the target in at most $\lambda_k^L(T[i+1, d^*]) - 1 = l_{i+1} - 2$ extra steps. In all cases, Φ locates the target in at most $l_{i+1} - 1$ steps.

  We now prove that $\pi(T[i, d^*]) = k - p_{i+1} + \pi_i$. For purpose of contradiction, let us assume that there is a strategy for locating the target in $T[i, d^*]$ in at most $\lambda_i = l_{i+1} - 1$ steps and probing strictly less than $k - p_{i+1} + \pi_i$ nodes during the first step. By definition, at least π_i nodes of $T[i]$ must be probed during the first step to locate the target in at most $\lambda_i = l_{i+1} - 1$ steps. Thus, it means that strictly less than $k - p_{i+1}$ nodes of $T[i+1, d^*]$ can be probed during the first step.

This contradicts that the strategy performs in at most $\lambda_i = l_{i+1} - 1$ steps since $\pi(T[i+1, d^*]) = k - p_{i+1}$.

- Case $\pi_i > p_{i+1} > 0$, $l_{i+1} = \lambda_i + 1$.

  In this case, it can be checked that $\alpha = \pi_i$ and that the **else** instruction (Line 12) is executed and so $l_i = l_{i+1} + 1$ and $p_i = k - 1$. We will prove that $\lambda_k^L(T[i, d^*]) = l_{i+1}$ and $\pi(T[i, d^*]) = 1$.

  By the induction hypothesis, $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$.

  We prove that $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i+1, d^*]) + 1 = l_{i+1}$. For purpose of contradiction, let us assume that $\lambda_k^L(T[i, d^*]) < l_{i+1}$ and let Φ' be a strategy for locating the target in $T[i, d^*]$ in at most $l_{i+1} - 1$ steps. Since $l_{i+1} - 1 = \lambda_i$, then at least π_i nodes of $T[i]$ must be probed during the first step. Since $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$, at least $k - p_{i+1}$ nodes of $T[i+1, d^*]$ must be probed during the first step. This means that at least $\pi_i + k - p_{i+1} > k$ nodes must be probed during the first step, a contradiction.

  We now prove that $\lambda_k^L(T[i, d^*]) = l_{i+1}$. It is sufficient to design a strategy Φ for locating the target in $T[i, d^*]$ in at most l_{i+1} steps. By Observation 5.4.8, there is a strategy Φ' for $T[i]$ for locating the target in at most $\lambda_i + 1$ steps that probes a single node during the first step. Also, let Φ'' be an optimal strategy for $T[i+1, d^*]$. The first step of Φ consists of probing one node of $T[i]$. If the target is in $T[i]$, the strategy continues with Φ' (in at most $\lambda_i = l_{i+1} - 1$ steps), otherwise, the strategy continues with Φ'' (in at most $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1$ steps). From this, we deduce that $\pi(T[i, d^*]) \leq 1$ and, by definition of π, we get that $\pi(T[i, d^*]) = 1$.

- Case $p_{i+1} = 0$, $l_{i+1} \geq \lambda_i + 1$.

  In this case, because of the **if** instruction (Line 7), the value of p is set to k and $l_i = l_{i+1} + 1$. Then, $\alpha = 1$ and so (**if** instruction of Line 10) $p_i = k - 1$. We will prove that $\lambda_k^L(T[i, d^*]) = l_{i+1}$ and $\pi(T[i, d^*]) = 1$. By the induction hypothesis, $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1 \geq \lambda_i$ and $\pi(T[i+1, d^*]) = k$.

  We first prove that $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i+1, d^*]) + 1 = l_{i+1}$. For purpose of contradiction, let us assume that $\lambda_k^L(T[i, d^*]) < l_{i+1}$ and let Φ be a k-strategy for $T[i, d^*]$ locating the target in at most $l_{i+1} - 1$ steps. First, if a node of $T[i]$ is probed during the first step of Φ, it means that at most $k - 1 < k - p_{i+1} = k$ nodes of $T[i+1, d^*]$ are probed during the first step of Φ, contradicting that $k - p_{i+1} = \pi(T[i+1, d^*])$ is the minimum number of nodes of $T[i+1, d^*]$ that must be probed during the first step of an optimal k-strategy for $T[i+1, d^*]$.

  Hence, neither Φ nor any k-strategy locating the target in $T[i, d^*]$ in at most $l_{i+1} - 1$ steps can probe some node of $T[i]$ during its first step. Below, we will build such a strategy Φ' (that probes some nodes of $T[i]$ during its first step) from Φ, which leads to a contradiction.

Since Φ does not probe any node of $T[i]$ during its first step, then $l_{i+1} - 1 > \lambda_i$ (otherwise, a target hidden in $T[i]$ will not be located in at most $l_{i+1} - 1$ steps, by definition of $\lambda_i > 0$). Let $x > 1$ be the first step of Φ that probes a node of $T[i]$ if the target is in $T[i]$ (such a step exists since $T[i]$ is not a rooted path by definition of d, *i.e.*, since $\lambda_i > 0$). Then, $l_{i+1} - x \geq \lambda_i$ since, otherwise, a target hidden in $T[i]$ could not be located by Φ in at most $l_{i+1} - 1$ steps. If $l_{i+1} - x = \lambda_i$, then the x^{th} step of Φ must probe π_i nodes of $T[i]$. Otherwise, if $l_{i+1} - x > \lambda_i$, we may assume that the x^{th} step of Φ probes a single node of $T[i]$ (by Observation 5.4.8).

Let $i + 1 \leq j \leq d^*$ be such that the first step of Φ probes some nodes of $T[j]$. Because the subtrees have been sorted, $\lambda_j \leq \lambda_i < l_{i+1} - 1$ and we may assume that the first step of Φ probes one node in $T[j]$ (by Observation 5.4.8). Let us define the k-strategy Φ' as follows. The strategy Φ' follows Φ but, during its first step, it probes one node of $T[i]$ instead of probing some nodes of $T[j]$. If the target is located in $T[i]$, then Φ' applies an optimal strategy in $T[i]$ and locates the target in at most $\lambda_i < l_{i+1} - 1$ extra steps. Otherwise, Φ' continues to mimic the strategy Φ until its x^{th} step. If the target has been located in some subtree before the x^{th} step, then Φ' continues to act as Φ. Otherwise, the x^{th} step of Φ' mimics the x^{th} step of Φ but, instead of probing one node of $T[i]$ (resp. π_i nodes of $T[i]$ if $l_{i+1} - x = \lambda_i$), Strategy Φ' probes one node of $T[j]$ (resp. $\pi_j \leq \pi_i$ nodes of $T[j]$ if $l_{i+1} - x = \lambda_i$). Then, Φ' proceeds as Φ.

It is easy to show that Φ' is a k-strategy for $T[i, d^*]$ locating the target in at most $l_{i+1} - 1$ steps, and probing some node of $T[i]$ during its first step, a contradiction.

We now prove that $\lambda_k^L(T[i, d^*]) = l_{i+1}$ and that $\pi(T[i, d^*]) = 1$. It is sufficient to design a strategy Φ for $T[i, d^*]$ locating the target in at most l_{i+1} steps. By Observation 5.4.8, there is a strategy Φ' for $T[i]$ for locating the target in at most $\lambda_i + 1$ steps and probes a single node during the first step. Also, let Φ'' be an optimal strategy for $T[i + 1, d^*]$. The first step of Φ consists of probing one node of $T[i]$. If the target is in $T[i]$, then the strategy continues with Φ' (in at most $\lambda_i \leq l_{i+1} - 1$ extra steps), otherwise, the strategy continues with Φ'' (in at most $\lambda_k^L(T[i + 1, d^*]) = l_{i+1} - 1$ extra steps). From this, we deduce that $\pi(T[i, d^*]) \leq 1$ and, by definition of π, we get that $\pi(T[i, d^*]) = 1$.

- Case $p_{i+1} > 0$, $l_{i+1} > \lambda_i + 1$.

  In this case, the condition of the **if** instruction (Line 7) is not satisfied, $\alpha = 1$, and so the condition of the **if** instruction (Line 10) is satisfied. Hence, $l_i = l_{i+1}$ and $p_i = p_{i+1} - 1$. We will prove that $\lambda_k^L(T[i, d^*]) = l_{i+1} - 1$ and $\pi(T[i, d^*]) = k - p_{i+1} + 1$. By the induction hypothesis, we have $\lambda_k^L(T[i + 1, d^*]) = l_{i+1} - 1 > \lambda_i$ and $\pi(T[i + 1, d^*]) = k - p_{i+1}$. By Observation 5.4.7, $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i + 1, d^*]) = l_{i+1} - 1$ also.

  To prove that $\lambda_k^L(T[i, d^*]) \leq \lambda_k^L(T[i + 1, d^*]) = l_{i+1} - 1$, it is sufficient to describe a strategy Φ for $\lambda_k^L(T[i, d^*])$ with a total of at most $l_{i+1} - 1$ steps. By Observation 5.4.8, there is a strategy Φ' for $T[i]$ for locating the target in at most $\lambda_i + 1$

steps that probes a single node during the first step. Let Φ'' be an optimal strategy for $T[i+1, d^*]$ probing at most $\pi(T[i+1, d^*]) = k - p_{i+1} < k$ nodes during the first step. The first step of Φ consists of probing one node in $T[i]$ (as Φ') and $\pi(T[i+1, d^*]) = k - p_{i+1}$ nodes of $T[i+1, d^*]$ (as Φ''). By assumption, $0 < p_{i+1}$, so $1 + \pi(T[i+1, d^*]) \leq k$ and at most k nodes are probed. By Claim 5.4.3, this first step allows to decide if the target is in $T[i]$ or not (in which case it is in $T[i+1, d^*]$). If the target is in $T[i]$, then Φ continues by following the strategy Φ' in $T[i]$ which will locate the target in at most $\lambda_i < l_{i+1} - 1$ extra steps. Otherwise (the target is in $T[i+1, d^*]$), Φ continues by following the optimal strategy Φ'' for $T[i+1, d^*]$ which will locate the target in at most $\lambda_k^L(T[i+1, d^*]) - 1 = l_{i+1} - 2$ extra steps. In all cases, Φ locates the target in at most $l_{i+1} - 1$ steps.

Let us prove that $\pi(T[i, d^*]) = k - p_{i+1} + 1$. For purpose of contradiction, let us assume that there is a strategy Φ for locating the target in $T[i, d^*]$ in at most $l_{i+1} - 1$ steps that probes strictly less than $k - p_{i+1} + 1$ nodes during the first step. We will show that we can construct a strategy Φ' in $T[i+1, d^*]$ for locating the target in at most $\ell_{i+1} - 1$ steps and probes at most $k - p_{i+1} - 1$ nodes during the first step, a contradiction. If the first step of Φ probes at least one node of $T[i]$, then it probes at most $k - p_{i+1} - 1$ nodes of $T[i+1, d^*]$ contradicting the fact that $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$. Hence, we may assume that the first step of Φ probes $k - p_{i+1}$ nodes of $T[i+1, d^*]$ and no nodes in $T[i]$.

Let $t > 1$ be the minimum integer such that at least one node of $T[i]$ is probed during the t^{th} step of Φ. After step t, at most $l_{i+1} - t - 1$ steps remain and so $l_{i+1} - t - 1 \geq \lambda_i - 1$. Let $j \in [\![i+1, d^*]\!]$ be such that at least one node of $T[j]$ is probed during the first step of Φ. Note that $j > i$ and, because the subtrees are ordered in non-increasing lexicographical order, either $\lambda_j < \lambda_i$ or ($\lambda_j = \lambda_i$ and $\pi_j \leq \pi_i$).

Let us consider the following strategy Φ' for $T[i+1, d^*]$. The first $t-1$ steps of the strategy Φ' follow the ones of Φ but do not probe any node of $T[j]$. That is, for every $j' \in [\![i+1, d^*]\!] \setminus \{j\}$ and for every $t' < t$, the step t' of Φ' probes the same nodes of $T[j']$ as the step t' of Φ. In particular, the first step of Φ' probes at most $k - p_{i+1} - 1$ nodes. If the target has been located in a subtree different from $T[j]$ during the first $t-1$ steps, then Φ' continues as Φ (but without probing the nodes of $T[i]$ since Φ' is a strategy for $T[i+1, d^*]$). Otherwise, the t^{th} step of Φ' proceeds as follows. For every $j' \in [\![i+1, d^*]\!] \setminus \{j\}$, the step t of Φ' probes the same nodes of $T[j']$ as the step t of Φ. Again, the strategy Φ' does not probe any node of $T[i]$. Note that, during its step t, the strategy Φ probes at least one node of $T[i]$, and it probes at least π_i nodes of $T[i]$ if $l_{i+1} - t = \lambda_i$. Therefore, there are two cases to be considered.

- If $l_{i+1} - t > \lambda_j$, then Φ' probes one node of $T[j]$ during step t. If the target is located in $T[j]$ then the next steps of Φ' follow an optimal strategy in $T[j]$ and will locate the target in at most λ_j extra steps. Otherwise, the next steps of Φ' follow the ones of Φ.

– If $l_{i+1} - t = \lambda_j$, then it implies that $l_{i+1} - t = \lambda_i$ (since $\ell_{i+1} - t \geq \lambda_i \geq \lambda_j$) and that the step t of Φ was probing π_i nodes in $T[i]$. The strategy Φ' replaces these π_i probes by probing $\pi_j \leq \pi_i$ nodes of $T[j]$. If the target is located in $T[j]$ then the next steps of Φ' follow an optimal strategy in $T[j]$ and will locate the target in at most $\lambda_j - 1$ extra steps. Otherwise, the next steps of Φ' follow the ones of Φ.

Overall, Φ' is a strategy for locating the target in $T[i+1, d^*]$ in at most $l_{i+1} - 1$ steps that probes at most $k - p_{i+1} - 1$ nodes during the first step. This contradicts the fact that $\pi(T[i+1, d^*]) = k - p_{i+1}$. Hence, $\pi(T[i, d^*]) = k - p_{i+1} + 1$.

- Case $l_{i+1} < \lambda_i + 1$.

  In this case, because of the **if** instruction (Line 7), the value of p is set to k and $l_i = \lambda_i + 1$. Then, $\alpha = \pi_i$ and so (**if** instruction on Line 10) $p_i = k - \pi_i$. We will prove that $\lambda_k^L(T[i, d^*]) = \lambda_i$ and $\pi(T[i, d^*]) = \pi_i$. By the induction hypothesis, $\lambda_k^L(T[i+1, d^*]) = l_{i+1} - 1 < \lambda_i$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$. Also, by Observation 5.4.7, $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i]) = \lambda_i$.

  To prove that $\lambda_k^L(T[i, d^*]) \leq \lambda_i$, it is sufficient to describe a strategy Φ for $\lambda_k^L(T[i, d^*])$ with a total of at most λ_i steps. Let Φ' be an optimal strategy for $T[i]$ probing at most π_i nodes during the first step. Let Φ'' be an optimal strategy for $T[i+1, d^*]$ for locating the target in at most $l_{i+1} - 1 < \lambda_i$ steps and probing at most $\pi(T[i+1, d^*]) = k - p_{i+1}$ nodes during the first step. The first step of Φ probes π_i nodes of $T[i]$ (as Φ'). By Claim 5.4.3, this first step allows to decide if the target is in $T[i]$ or not. If it is in $T[i]$ then Φ follows Φ'. Otherwise, Φ executes Φ'' in $T[i+1, d^*]$.

  To conclude, let us prove that $\pi(T[i, d^*]) = \pi_i$. The previous strategy Φ shows that $\pi(T[i, d^*]) \leq \pi_i$. Since $\lambda_k^L(T[i, d^*]) = \lambda_i$, any strategy for $T[i, d^*]$ must probe at least π_i nodes of $T[i]$ during the first step by definition of π_i. This concludes the proof. □

With Lemma 5.4.9 in hand, we are now ready to prove that Algorithms 1 and 2 are correct. We also prove that their running time is polynomial. More precisely, we prove in Theorem 5.4.11 below that $\mathcal{A}_1(k, (T, r))$ computes $(\lambda_k^L(T), \pi(T))$ and a corresponding k-strategy in time $O(n \log n)$, where n is the number of nodes. To do that, Theorem 5.4.10 proves the correctness and the linear (in the number of children of r) time complexity of $\mathcal{A}_2(k, (T, r), (\Lambda, \Pi))$.

Theorem 5.4.10. *Let $k \geq 1$ and $(T, r) \in \mathcal{T}$ be a tree rooted in r with children $v_1, \ldots, v_{d^*}$, such that the tuples $(\Lambda, \Pi) = (\lambda_i, \pi_i)_{1 \leq i \leq d^*}$ are sorted in non-increasing lexicographical ordering. Then, $\mathcal{A}_2(k, (T, r), (\Lambda, \Pi))$ returns $(\lambda_k^L(T), \pi(T))$ and a corresponding strategy. Furthermore, the time complexity of $\mathcal{A}_2$ is $O(d^*)$ (independent of k).*

Proof. The time complexity is obvious and the correctness follows from Lemma 5.4.9 for $i = 1$. The fact that the strategy is also returned is not explicitly described in Algorithm 2 but directly follows from the proof of Lemma 5.4.9. □

Theorem 5.4.11. *Let $k \geq 1$ and let $(T, r) \in \mathcal{T}$ be an n-node rooted tree. Then, $\mathcal{A}_1(k, (T, r))$ returns $(\lambda_k^L(T), \pi(T))$ and a corresponding strategy. Furthermore, the time complexity of $\mathcal{A}_1$ is $O(n \log n)$ (independent of k).*

Proof. The correctness is simply proved by induction and by Theorem 5.4.10. For the time complexity, at every recursive call on a subtree T_v rooted at v (with d_v children), the additional number of operations is $O(d_v \log d_v)$ (sorting) plus $O(d_v)$ (Algorithm $\mathcal{A}_2$, by Theorem 5.4.10). Since, in any n-node tree T, $\sum_{v \in V(T)} d_v = 2(n-1)$, this gives a total complexity of $O(\sum_{v \in V(T)} d_v \log d_v) = O(n \log n)$. Again, the strategy is not explicit in our presentation but can be easily computed. □

5.4.3 A Polynomial-time $(+1)$-approximation

From Algorithm $\mathcal{A}_1(k, (T, r))$, it is easy to get an efficient approximation algorithm for solving Localization in trees when k and ℓ are part of the input, and a polynomial time algorithm when k is fixed.

Theorem 5.4.12. *There exists an algorithm with running time $O(n \log n)$ that, given any integer $k \geq 1$ and n-node tree T, computes a k-strategy for locating a target in T in at most $\lambda_k(T) + 1$ steps.*

Proof. The strategy proceeds as follows. The first step probes any arbitrary node r of T. Let d be the distance between r and the target, $L \subseteq V(T)$ be the set of nodes at distance exactly d from r, and T^d be the subtree induced by r and every node on a path between r and the nodes in L. Note that $(T^d, r) \in \mathcal{T}$ and that the target is occupying a leaf of T^d. Hence, the target can be located by applying $\mathcal{A}_1(k, (T^d, r))$. By Theorem 5.4.11, this will locate the target in at most $1 + \max_d \lambda_k^L(T^d) \leq 1 + \lambda_k(T)$ steps. □

Theorem 5.4.13. *There exists an algorithm with running time $O(n^{k+2} \log n)$ that, given any integer $k \geq 1$ and n-node tree T, computes an optimal k-strategy for locating a target in T in at most $\lambda_k(T)$ steps.*

Proof. The proof is similar to the one of Theorem 5.4.12, but instead of probing a single node during the first step, we enumerate all the $O(n^k)$ possibilities for the first step, and, for each of them, we then apply Algorithm $\mathcal{A}_1$. For one of these instances, the target will be located within $\lambda_k(T)$ steps. □

To conclude this section, it is important to mention that both Theorems 5.4.12 and 5.4.13 also hold in the case of edge-weighted trees. Indeed, distances are only used in Claim 5.4.3 which clearly holds for edge-weighted trees.

5.5 Further work

In this chapter, we have studied the computational complexity of the Localization problem. We have established the importance of its two main parameters, namely the number k of vertices that can be probed each step, and the number ℓ of steps within

which the target must be located. Namely, fixing any of these two parameters makes the problem NP-complete. This remains true for the Relative-Localization problem as well.

We have then focused on the case of trees. For that case, we have proved that the Localization problem remains NP-complete. However, the only source of hardness is due to the first probing step as, as soon as the second step begins, an optimal way to play can be computed in polynomial time. As a consequence, the problem, though hard, can be approximated efficiently.

Our results in trees leave the open question of whether determining $\lambda_k(T)$ is FPT (in k) in the class of trees T. Also, we do not know the complexity of determining whether $\kappa_\ell(T) \leq k$ for a tree T. An interesting line of research could be to study all those problems in other graph classes, such as interval or planar graphs.

The problem of determining $\lambda_k^{rel}(T)$ for a tree T seems to be much more intricate even for simple topologies. A first step towards a better understanding of it would be to fully understand the centroidal dimension of paths (*i.e.*, to determine $\kappa_1^{rel}(P)$ for every path P), which has been initiated in [62]. A more challenging direction would then be to consider the case of all trees.

Chapter 6

Oriented Metric Dimension

6.1 Introduction

A natural way of generalizing graph theoretical problems is to consider their directed counterparts. In this chapter, the metric dimension of digraphs is considered. The metric dimension of digraphs was first considered by Chartrand, Rains, and Zhang in [43], before receiving further consideration in several works (see [57, 58, 98, 105, 107]). It is worthwhile recalling that, in digraphs, distances have behaviours that differ from those in undirected graphs. Notably, an important point that should be addressed is that, in the context of general digraphs D, we might have $dist(u, v) \neq dist(v, u)$ for any two vertices u, v, where $dist(u, v)$ here refers to the length of a shortest directed path from u to v. A digraph D is *strongly-connected* (or *strong* for short) if, for every $u, v \in V(D)$, there is a directed path from u to v, and conversely one from v to u. Hence, if D is not strong, then there are vertices $u, v \in V(D)$ such that no directed paths from u to v exist. In such a case, we set $dist(u, v) = +\infty$.

These peculiar aspects of distances in digraphs must be taken into account when defining directed notions of resolving sets and metric dimension. Throughout this chapter, the notions of resolving sets and metric dimension in digraphs are with respect to the following definitions. Let R be a subset of vertices of a digraph D. Two vertices u, v of D are said to be *distinguished*, denoted by $u \nsim_R v$, if there exists $w \in R$ such that $dist(w, u) \neq dist(w, v)$. Otherwise, u and v are *undistinguished* by R, which is denoted by $u \sim_R v$. In particular, if $dist(w, u)$ is finite and $dist(w, v)$ is not for some $w \in R$, then $u \nsim_R v$. A set $R \subseteq V(D)$ is called *resolving* if all pairs of vertices of D are distinguished by R. The metric dimension of D, denoted by $MD(D)$, is then the smallest size of a resolving set. Note that $MD(D)$ is defined for every digraph D; in particular, we have $MD(D) < |V(D)|$ since $R = V(D) \setminus \{v\}$ is a resolving set for any $v \in V(D)$ (as having any vertex in a resolving set makes it distinguished from all other vertices).

Our definitions of directed resolving sets and metric dimension actually differ from those originally introduced by Chartrand, Rains, and Zhang. On the one hand, in their definition of resolving sets, they consider the distances from each of the vertices not in R to the vertices in R in order to distinguish the vertices of D. In our definition, the

distances from each of the vertices in R to the vertices not in R are considered. Note that both definitions are equivalent on that point, as, given a digraph D, if we reverse the direction of all arcs, resulting in a digraph $\widetilde{D}$, then any shortest path from u to v in D becomes a shortest path from v to u in $\widetilde{D}$.

On the other hand, their definition of resolving sets requires that the distances from each pair of distinct vertices to the vertices in R which distinguish them be defined, while our definition (with distances from vertices in R to the other vertices) allows for undefined distances ($+\infty$) to be used as well. Contrary to our definition, this implies that their definition of metric dimension is not defined for all digraphs. As far as we know, the characterization of digraphs that admit a metric dimension (following their definition) is still an open problem [43].

Although our definitions and those of Chartrand, Rains, and Zhang are different, it is worthwhile mentioning that most of our investigations in this chapter also apply to their context, as we mainly focus on strong digraphs, in which case our definitions and theirs are equivalent (up to reversing all arcs).

To date, the investigations on the metric dimension of digraphs have thus been with respect to the definitions originally introduced by Chartrand, Rains, and Zhang. As a first step, they notably gave in [43], a characterization of digraphs with metric dimension 1. Complexity aspects were considered in [107], where it was proved that determining the metric dimension of a strong digraph is NP-complete. Bounds on the metric dimension of various digraph families were later exhibited (Cayley digraphs [57], line digraphs [58], tournaments [98], digraphs with cyclic covering [105], De Bruijn and Kautz digraphs [107], etc.).

6.1.1 From Undirected Graphs to Oriented Graphs

To avoid any confusion, let us recall that an *orientation* D of an undirected graph G is obtained when every edge uv of G is oriented either from u to v (resulting in the arc (u, v)) or conversely (resulting in the arc (v, u)). An *oriented graph* D is a directed graph that is an orientation of a simple graph. Note that when G is simple, D cannot have two vertices u, v such that (u, v) and (v, u) are arcs. Such symmetric arcs are allowed in digraphs, which is the main difference between oriented graphs and digraphs. Throughout this chapter, when simply referring to a *graph*, we mean an undirected graph.

In [44], Chartrand, Rains, and Zhang considered the following way of linking resolving sets of undirected graphs and digraphs. They considered, for a given graph G, the worst orientations of G for the metric dimension, *i.e.*, orientations of G with maximum metric dimension. Looking at our definition of resolving sets and metric dimension, this is a legitimate question as it has to be pointed out that, for a graph, the metric dimension might or might not be preserved when orienting its edges. An interesting example (reported, *e.g.*, in [43, 98]) is the case of a graph G with a Hamiltonian path: while $MD(G)$ can be arbitrarily large in general (consider, *e.g.*, any complete graph), there is an orientation D of G verifying $MD(D) = 1$ (just orient all edges of a Hamiltonian path from the first vertex towards the last vertex, and all remaining edges in the

opposite direction). Conversely, there exist orientations D of G for which $MD(D)$ can be much larger than $MD(G)$. As an example, let us consider any path P with $2n+1$ vertices $v_0, ..., v_{2n}$. Clearly, $MD(P) = 1$; however, the orientation D of P obtained by making every vertex v_{2k+1} $(k = 0, ..., n-1)$ become a source, (*i.e.*, orienting its incident edges away) verifies $MD(D) = n$. As shown in this chapter, this phenomenon occurs for strong orientations as well.

In [44], the authors proved that, for every positive integer k, there exist infinitely many graphs for which the metric dimension of any of its strongly-connected orientations is exactly k. They have also proved that there is no constant k such that the metric dimension of any tournament is at most k.

6.1.2 Our Results

All results in this chapter are from [c-6], which is joint work with J. Bensmail and N. Nisse. Motivated by the observations above, we investigate, throughout this chapter, the parameter $WOMD$ defined as follows. For any connected graph G, let $WOMD(G)$ denote the maximum value of $MD(D)$ over all strong orientations D of G. Let us extend this definition to graph families as follows. For any family $\mathcal{G}$ of 2-edge-connected graphs[†], let $WOMD(\mathcal{G}) = \max_{G \in \mathcal{G}} \frac{WOMD(G)}{|V(G)|}$. Section 6.2 first introduces tools and results that will be used in the next sections. In Section 6.3, bounds on $WOMD(\mathcal{G}_\Delta)$ are proved, where $\mathcal{G}_\Delta$ refers to the family of 2-edge-connected graphs with maximum degree Δ. In particular, we prove that we asymptotically have $\frac{2}{5} \leq WOMD(\mathcal{G}_3) \leq \frac{1}{2}$. In Section 6.4, we then consider the families of grids and tori. For the family $\mathcal{T}$ of tori, we prove that we asymptotically have $WEOMD(\mathcal{T}) = \frac{1}{2}$, where the parameter $WEOMD(\mathcal{T})$ is defined similarly to $WOMD(\mathcal{T})$ except that only strong Eulerian orientations of tori, (*i.e.*, all vertices have in-degree and out-degree 2) are considered. For the family $\mathcal{G}$ of grids, we then prove that asymptotically $\frac{1}{2} \leq WOMD(\mathcal{G}) \leq \frac{2}{3}$. Remaining open questions and problems are gathered in Section 6.5.

6.2 Tools and Preliminary Results

We start off by pointing out the following property of resolving sets in digraphs having vertices with the same in-neighbourhood. This result will be one of our main tools for building digraphs with large metric dimension.

Lemma 6.2.1. *Let D be a digraph and $S \subseteq V(D)$ be a subset of $|S| \geq 2$ vertices such that, for every $u, v \in S$, we have $N^-(u) = N^-(v)$. Then, any resolving set of D contains at least $|S| - 1$ vertices of S.*

Proof. If two vertices $u, v \in S$ do not belong to a resolving set R, then $dist(w, u) = dist(w, v)$ for every $w \in R$, contradicting that R is a resolving set. □

[†]The edge-connectivity requirement, here and further, is to guarantee the good definition of $WOMD(G)$ for every $G \in \mathcal{G}_\Delta$, as it is a well-known fact that a graph has strong orientations if and only if it is 2-edge-connected (see [111]).

We now introduce a technique that will be used in the next sections for exhibiting upper bounds on the metric dimension of strong digraphs with maximum out-degree at least 2. The technique is based on a connection between the resolving sets of a such digraph and the vertex covers of a particular graph associated to it. A *vertex cover* of a graph G is a subset $S \subseteq V(G)$ of vertices such that, for every edge uv of G, at least one of u and v belongs to S. To any digraph D we associate an *auxiliary (undirected) graph* D_{aux} constructed as follows:

- the vertices of D_{aux} are those of D;
- for every two distinct vertices u, v of D such that $N_D^-(u) \cap N_D^-(v) \neq \emptyset$, let us add the edge uv to D_{aux}.

In other words, D_{aux} is the simple undirected graph depicting the pairs of distinct vertices of D sharing an in-neighbour. By construction, note that, in D_{aux}, every two distinct vertices are joined by at most one edge.

It turns out that, for strong digraphs D with maximum out-degree at least 2, a vertex cover of D_{aux} is resolving in D.

Lemma 6.2.2. *Let D be a strong digraph with $\Delta^+(D) \geq 2$. Then, any vertex cover of D_{aux} is a resolving set of D.*

Proof. Towards a contradiction, assume the claim is false, *i.e.*, there exists a set $S \subseteq V(D)$ which is a vertex cover of D_{aux} but not a resolving set of D. Since $\Delta^+(D) \geq 2$, there are edges in D_{aux} and thus $S \neq \emptyset$. Let v_1, v_2 be two vertices that cannot be distinguished through their distances from S; in other words, for every $w \in S$ (note that $w \neq v_1, v_2$), we have $\text{dist}_D(w, v_1) = \text{dist}_D(w, v_2)$, and that distance is finite since D is strong. Now consider such a vertex $w \in S$ at minimum distance from v_1 and v_2. In D, any shortest path P_1 from w to v_1 has the same length as any shortest path P_2 from w to v_2.

Because $v_1 \neq v_2$ and P_1, P_2 are shortest paths, note that all vertices of P_1 and P_2 cannot be the same; let thus x_1 denote the first vertex of P_1 that does not belong to P_2, and, similarly, let thus x_2 denote the first vertex of P_2 that does not belong to P_1. In other words, the first vertices of P_1 and P_2 coincide up to some vertex x, but the next vertices x_1 (in P_1) and x_2 (in P_2) are different. So, D_{aux} contains the edge x_1x_2, and at least one of x_1, x_2 belongs to S. Furthermore, x_1 and x_2 are closer to v_1, v_2 than w is; this is a contradiction to the original choice of w. □

Lemma 6.2.2 shows that a resolving set of any strong digraph (with maximum out-degree at least 2) can be obtained by considering every vertex and choosing at least all of its out-neighbours but one. The proof suggests that this is because this is a way to distinguish all shortest paths from a vertex to other ones.

Corollary 6.2.3. *For every strong digraph D with $\Delta^+(D) \geq 2$, the metric dimension $MD(D)$ of D is at most the size of a minimum vertex cover of D_{aux}.*

Unfortunately, determining the minimum size of a vertex cover of a given graph is an NP-complete problem in general [69]. However, in the context of Corollary 6.2.3, we are

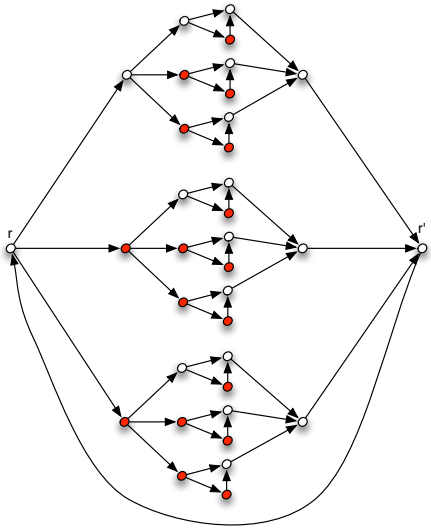


Figure 6.1: The oriented graph $D_{3,3,2}$. The set of red vertices is an example of an optimal resolving set.

mostly interested in having reasonable upper bounds on the size of a minimum vertex cover of D_{aux}. Such upper bounds can be exhibited when D has particular additional properties, as will be shown in the next sections.

6.3 Strong Oriented Graphs with Bounded Maximum Degree

By the *maximum degree* $\Delta(D)$ of a given oriented graph D, we mean the maximum degree of its underlying undirected graph, (*i.e.*, the maximum value of $d^-(v) + d^+(v)$ over the vertices v of D). In this section, we investigate the maximum value that $MD(D)$ can take among all strong orientations D of a graph with given maximum degree. Since a strong oriented graph D with $\Delta(D) = 2$ is a directed cycle, in which case $MD(D)$ is trivially 1, we focus on cases where $\Delta(D) \geq 3$.

All our lower bounds in this section are obtained through the following constructions. For any $k \in \mathbb{N}$ and $\Delta \geq 2$, we denote by $T_{\Delta,k}$ the rooted Δ-ary complete tree with depth k. More precisely, $T_{\Delta,k}$ is a rooted tree such that every non-leaf vertex has Δ children and all leaves are at distance k from the root. Note that $|V(T_{\Delta,k})| = \frac{\Delta^{k+1}-1}{\Delta-1}$ and $T_{\Delta,k}$ has Δ^k leaves and maximum degree $\Delta + 1$. For any $k \in \mathbb{N}$ and $\Delta, i \geq 2$, let $D_{\Delta,k,i}$ be the oriented graph defined as follows (see Figure 6.1 for an illustration). Start with T being a copy of $T_{\Delta,k-1}$ with all edges oriented from the root to the leaves. Let $v_1^{k-1}, \cdots, v_{\Delta^{k-1}}^{k-1}$ be the leaves of T and let r be its root. For every $1 \leq j \leq \Delta^{k-1}$, add i out-neighbours $u_1^j, \cdots, u_i^j$ to v_j^{k-1}. Then, for $1 \leq j \leq \Delta^{k-1}$ and $1 \leq \ell < i$, add the arc (u_ℓ^j, u_i^j). Then, add a copy T' of $T_{\Delta,k-2}$ where all edges are oriented from the leaves to the root. Let $v'_1, \cdots, v'_{\Delta^{k-2}}$ be the leaves of T' and let r' be its root. For every

$1 \leq j \leq \Delta^{k-2}$ and for every $1 \leq \ell \leq \Delta$, add the arc $(u_i^{\Delta(j-1)+\ell}, v'_j)$. Finally, add the arc (r', r); note that this ensures that $D_{\Delta,k,i}$ is strong.

Theorem 6.3.1. *For every $k \in \mathbb{N}$ and $\Delta, i \geq 2$, $D_{\Delta,k,i}$ is a strong oriented graph with maximum degree $\Delta + 1$,*

$$|V(D_{\Delta,k,i})| = \frac{\Delta^k - 1}{\Delta - 1} + i\Delta^{k-1} + \frac{\Delta^{k-1} - 1}{\Delta - 1}$$

and

$$MD(D_{\Delta,k,i}) \geq \Delta^{k-1} - 1 + \Delta^{k-1} \max\{1, i-2\}.$$

Proof. We only need to prove the last statement. For every $1 \leq \ell \leq \Delta^{k-1}$, let $v_1^\ell, \cdots, v_{\Delta^\ell}^\ell$ denote the vertices of $D_{\Delta,k,i}$ at distance ℓ from $r = v_1^0$. Note that, for every $0 \leq \ell \leq k-2$ and $1 \leq j \leq \Delta^\ell$, the vertices $v_{\Delta(j-1)+1}^{\ell+1}, \cdots, v_{\Delta(j-1)+\Delta}^{\ell+1}$ have the same in-neighbourhood $\{v_j^\ell\}$. By Lemma 6.2.1, every resolving set of $D_{\Delta,k,i}$ thus has to include at least $\Delta - 1$ of these vertices. For every $1 \leq j \leq \Delta^{k-1}$, the vertices $v_{i(j-1)+1}^k, \cdots, v_{i(j-1)+i-1}^k$ have the same in-neighbourhood $\{v_j^{k-1}\}$. Again by Lemma 6.2.1, every resolving set of $D_{\Delta,k,i}$ must thus include at least $i-2$ of these vertices. Moreover, it can be checked that, when $i = 2$, every resolving set of $D_{\Delta,k,i}$ must include at least one of $v_{2(j-1)+1}^k, v_{2(j-1)+2}^k$. Figure 6.1 shows an example of a resolving set of $D_{3,3,2}$.

Hence, any resolving set R of $D_{\Delta,k,i}$ verifies

$$|R| \geq \left(\sum_{\ell=0}^{k-2} \Delta^\ell (\Delta - 1) \right) + \Delta^{k-1} \max\{1, i-2\}$$

which can be manipulated into the claimed lower bound. □

In the rest of this section, we exhibit upper bounds on $MD(D)$ for oriented graphs D with bounded maximum degree, some of which are close to lower bounds that can be established using Theorem 6.3.1.

6.3.1 Strong Subcubic Oriented Graphs

We begin with strong subcubic, (*i.e.*, with maximum degree 3) oriented graphs D. The upper bound is obtained from Corollary 6.2.3.

Lemma 6.3.2. *For every strong subcubic n-node oriented graph D, we have* $\mathrm{MD}(D) \leq \frac{n}{2}$.

Proof. In D, there are only three types of vertices, namely:

- vertices v with $d^-(v) = d^+(v) = 1$;
- vertices v with $d^-(v) = 1$ and $d^+(v) = 2$;
- vertices v with $d^-(v) = 2$ and $d^+(v) = 1$.

Note that from the point of view of the arcs out-going from the vertices, only the vertices v verifying $d^+(v) = 2$ create edges in D_{aux}. More precisely, every such vertex v yields at most one edge in D_{aux} ("at most" because two such vertices can have the same two out-neighbours, in which case only one edge is created). Since D verifies $\sum_{v \in V(D)} d_D^-(v) = \sum_{v \in V(D)} d_D^+(v)$, clearly its number of vertices v with $d^+(v) = 2$ is at most $\frac{1}{2}n$. This yields that D_{aux} is a graph with order n and at most $\frac{1}{2}n$ edges. Thus, D_{aux} admits a vertex cover S with size at most $\frac{1}{2}n$: one such set can be obtained, *e.g.*, by considering each of its edges in turn, and arbitrarily adding one of its ends to S. The result now follows from Lemma 6.2.2. □

From Theorem 6.3.1 and Lemma 6.3.2, we thus get:

Corollary 6.3.3. *Let $\mathcal{G}_3$ be the family of 2-edge-connected graphs with maximum degree 3. For any $\epsilon > 0$, we have*

$$\frac{2}{5} - \epsilon \leq WOMD(\mathcal{G}_3) \leq \frac{1}{2}.$$

Proof. Let $G \in \mathcal{G}_3$ and D be any strong orientation of G (it exists because G is 2-edge-connected). The upper bound follows from Lemma 6.3.2 (since, because $\sum_{v \in V(D)} d^-(v) = \sum_{v \in V(D)} d^+(v)$ and D is strong, we have $\Delta^+(D) \geq 2$). The lower bound follows from Theorem 6.3.1 by considering the oriented graph $D_{2,k,2}$. Indeed, any resolving set of $D_{2,k2}$ has at least $2 * 2^{k-1} - 1$ vertices and $n = |V(D_{2,k,2})| = 5 * 2^{k-1} - 2$. Hence, $\lim_{k \to \infty} MD(D_{2,k,2}) \geq \frac{2n}{5}$. □

6.3.2 Strong Oriented Graphs with Maximum Degree at least 4

In the next result, we exhibit a general upper bound on $MD(D)$ for every strong digraph D with given maximum in-degree and maximum out-degree (at least 2). Recall that a *proper vertex-colouring* of an undirected graph is a partition of the vertices into stable sets.

Theorem 6.3.4. *For every strong n-node digraph D with maximum in-degree Δ^- and maximum out-degree $\Delta^+ \geq 2$, we have*

$$\mathrm{MD}(D) \leq \frac{\Delta^-(\Delta^+ - 1)}{\Delta^-(\Delta^+ - 1) + 1} n.$$

Proof. The maximum degree of a vertex v of D_{aux} is $\Delta^-(\Delta^+ - 1)$: this is because v has at most Δ^- in-neighbours in D, each of which, if it has an out-neighbour different from v, might yield a new edge incident to v in D_{aux}. So each of these at most Δ^- in-neighbours of v in D might create, in D_{aux}, up to $\Delta^+ - 1$ edges incident to v. Hence, the maximum degree of D_{aux} is $\Delta^-(\Delta^+ - 1)$. From greedy colouring arguments, it thus follows that D_{aux} admits a proper vertex-colouring using at most $\Delta^-(\Delta^+ - 1) + 1$ colours.

The claim now follows from Lemma 6.2.2 by just noting that, for any graph with a given proper vertex-colouring, a vertex cover can be obtained by taking all colour

classes but one. In particular, since a proper k-vertex-colouring of an n-node graph always has a colour class with size at least $\frac{1}{k}n$, we deduce the claim by considering, as a vertex cover of D_{aux}, all colour classes but a biggest one of any proper $(\Delta^-(\Delta^+-1)+1)$-vertex-colouring. □

Theorems 6.3.1 and 6.3.4 yield the following:

Corollary 6.3.5. *Let $\mathcal{G}_4$ be the family of 2-edge-connected graphs with maximum degree 4. For any $\epsilon > 0$, we have*

$$\frac{1}{2} - \epsilon \leq WOMD(\mathcal{G}_4) \leq \frac{6}{7}.$$

Proof. Let $G \in \mathcal{G}_4$ and let D be a strong orientation of G (it exists because G is 2-edge-connected). The upper bound follows from Theorem 6.3.4, since a strong oriented graph with maximum degree 4 has maximum in-degree and maximum out-degree at most 3 (and at least 2, since $\sum_{v \in V(D)} d^-(v) = \sum_{v \in V(D)} d^+(v)$). Therefore, the largest upper bound given by Theorem 6.3.4 is when $\Delta^+(D) = \Delta^-(D) = 3$ which leads to the upper bound of 6/7. The lower bound follows from Theorem 6.3.1 by considering the oriented graph $D_{3,k,2}$. Indeed, for all $k \in \mathbb{N}$, $MD(D_{3,k,2}) \geq 2 * 3^{k-1} - 1$ and $|V(D_{\Delta,k,i})| = 4 * 3^{k-1} - 1$. □

More generally, *i.e.*, for larger values of the maximum degree, the construction in Theorem 6.3.1 is asymptotically optimal:

Corollary 6.3.6. *Let $\mathcal{G}_{\Delta+1}$ be the family of 2-edge-connected graphs with maximum degree $\Delta + 1$. Then,*

$$\lim_{\Delta \to \infty} WOMD(\mathcal{G}_{\Delta+1}) = 1.$$

Proof. By definition, $WOMD(\mathcal{G}_{\Delta+1}) \leq 1$ for every Δ. To prove the claim, it is sufficient to show that $\lim_{\Delta \to \infty} WOMD(D_{\Delta,k,\Delta}) = 1$. By Theorem 6.3.1, for $\Delta \geq 3$,

$$MD(D_{\Delta,k,\Delta}) \geq (\Delta - 1)\Delta^{k-1} - 1.$$

Moreover, $|V(D_{\Delta,k,\Delta})|(\Delta - 1) = \Delta^{k+1} + \Delta^{k-1} - 2$. Hence,

$$\frac{MD(D_{\Delta,k,\Delta})}{|V(D_{\Delta,k,\Delta})|} \geq \frac{(\Delta-1)^2\Delta^{k-1} - (\Delta-1)}{\Delta^{k+1} + \Delta^{k-1} - 2} = \frac{1 - \frac{1}{\Delta}\left(2 - \frac{1}{\Delta} + \frac{1}{\Delta^{k-1}} - \frac{1}{\Delta^k}\right)}{1 + \frac{1}{\Delta^2}\left(1 - \frac{2}{\Delta^{k-1}}\right)} \underset{\Delta \to \infty}{\to} 1.$$

□

6.4 Strong Orientations of Grids and Tori

By a *grid* $G_{n \times m}$, we refer to the Cartesian product $P_n \Box P_m$ of two paths P_n, P_m. A *torus* $T_{n \times m}$ is the Cartesian product $C_n \Box C_m$ of two cycles C_n, C_m. In the undirected context, it is easy to see that $MD(G_{n \times m}) = 2$ while $MD(T_{n \times m}) = 3$ (see, *e.g.*, [102]); however, things get a bit more tricky in the directed context.

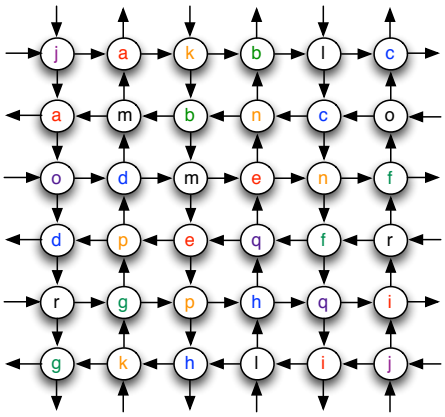


Figure 6.2: An orientation $\vec{T}^*$ of the $6 * 6$ torus $T_{6,6}$ verifying $MD(\vec{T}^*) = |V(T_{6,6})|/2$. Every two vertices marked with a same letter have the same in-neighbourhood; thus, every resolving set must contain at least one of them.

Grids and tori have maximum degree 4; thus, bounds on the maximum metric dimension of a strong oriented grid or torus can be derived from our results in Section 6.3.2. In this section, we improve these bounds through dedicated proofs and arguments. We first consider strong Eulerian oriented tori (all vertices have in-degree and out-degree 2), for which we exhibit the maximum value of the metric dimension. We then consider strong oriented grids, for which we provide improved bounds.

6.4.1 Strong Eulerian Orientations of Tori

Let $0 < n \leq m$ be two integers, and let $T_{n\times m}$ be the torus on nm vertices. That is, $V(T_{n\times m}) = \{(i,j) \mid 0 \leq i < n, 0 \leq j < m\}$, and $(i,j),(k,\ell) \in E(T_{n\times m})$ if and only if $|i-k| \in \{1, n-1\}$ and $j = \ell$, or $|j-\ell| \in \{1, m-1\}$ and $i = k$. By convention, the vertex $(0,0)$ is regarded as the topmost, leftmost vertex of the torus. That is, $\{(0,j) \in V(T_{n\times m}) \mid 0 \leq j < m\}$ is the topmost (or first) row, and $\{(i,0) \in V(T_{n\times m}) \mid 0 \leq i < n\}$ is the leftmost (or first) column.

As a main result in this section, we determine the maximum metric dimension of a strong Eulerian oriented torus. More precisely, we study the following slight modifications of the parameter $WOMD$. For a connected graph G, we denote by $WEOMD(G)$ the maximum value of $MD(D)$ over all strong Eulerian orientations D of G. For a family $\mathcal{G}$ of 2-edge-connected graphs, we set $WEOMD(\mathcal{G}) = \max_{G\in\mathcal{G}} \frac{WEOMD(G)}{|V(G)|}$.

Theorem 6.4.1. *For the family $\mathcal{T}$ of tori, we have $WEOMD(\mathcal{T}) = \frac{1}{2}$.*

We first show that there exist strong Eulerian oriented tori D with $MD(D) \geq \frac{nm}{2}$.

Lemma 6.4.2. *For every $n_0, m_0 \in \mathbb{N}$, there is $n \geq n_0, m \geq m_0$, and a strong Eulerian orientation $\vec{T}^*$ of the torus $T_{n\times m}$ such that $MD(\vec{T}^*) \geq \frac{nm}{2}$.*

Proof. Let n (resp., m) be the smallest even integer greater or equal to n_0 (resp., m_0). We orient $T_{n\times m}$ in the following way, resulting in $\vec{T}^*$ (see Figure 6.2 for an illustration). The edges of the even rows of $T_{n\times m}$ are oriented from left to right, *i.e.*, $((2i,j)(2i,j+1 \mod m))$ is an arc for every $0 \leq j < m$ and $0 \leq i < n/2$. The edges of the odd rows are oriented from right to left, *i.e.*, $((2i+1,j)(2i+1,j-1 \mod m))$ is an arc for every $0 \leq j < m$ and $0 \leq i < n/2$. The edges of the even columns are oriented from top to bottom, *i.e.*, $((i,2j)(i+1 \mod n, 2j))$ is an arc for every $0 \leq j < m/2$ and $0 \leq i < n$. The edges of the odd columns are oriented from bottom to top, *i.e.*, $((i,2j+1)(i-1 \mod n, 2j+1))$ is an arc for every $0 \leq j < m/2$ and $0 \leq i < n$.

For every $0 \leq i < n/2$ and $0 \leq j < m/2$, vertices $(2i,2j+1)$ and $(2i+1,2j)$ have the same in-neighbourhood. Moreover, $(2i,2j)$ and $(2i-1 \mod n, 2j-1 \mod m)$ have the same in-neighbourhood. By Lemma 6.2.1, any resolving set of $\vec{T}^*$ must contain at least one vertex of each of these $\frac{nm}{2}$ pairs of vertices. Hence, $MD(\vec{T}^*) \geq \frac{nm}{2}$. □

We now prove the upper bound.

Lemma 6.4.3. *For every strong Eulerian oriented torus $\vec{T}_{n\times m}$ with n rows and m columns,*

$$MD(\vec{T}_{n\times m}) \leq \frac{n'm'}{2} + n'' + m'',$$

where, for $x \in \{n,m\}$, (x',x'') equals $(x,0)$ if x is even and $(x-1,x)$ otherwise. In particular, if both n and m are even, then $MD(\vec{T}_{n\times m}) \leq \frac{nm}{2}$.

Proof. Let us first consider the case when n and m are even. The proof is constructive and provides a resolving set of size at most $\frac{nm}{2}$. The algorithm starts with the set $R = \{(i,j) \in V(\vec{T}_{n\times m}) \mid i+j \text{ even}\}$ (note that it is a minimum vertex cover and a stable set of size $\frac{nm}{2}$) and iteratively performs local modifications (swaps one vertex in R with one of its neighbours not in R) without changing the size of R until R becomes a resolving set R^*.

Let us assume that $R = \{(i,j) \in V(\vec{T}_{n\times m}) \mid i+j \text{ even}\}$ is not a resolving set (otherwise, we are done). This means that at least two vertices are not distinguishable by their distances to the vertices in R. Let u and v be two such vertices. Recall that we denote this relationship by $u \sim_R v$.

Necessarily, if $u \sim_R v$ then $u,v \notin R$ (since any vertex $w \in R$ is the only one at distance 0 from itself, it can be distinguished from every other vertex). Moreover, since R is a vertex cover and $d^-(u) = d^-(v) = 2$, each of u and v must have two in-neighbours in R. Since u and v are not distinguishable, they must have the same in-neighbours, denoted by $n_u, n_v \in R$. That is, since each vertex has exactly two in-neighbours and two out-neighbours (by Eulerianity), $N^+(n_u) = N^+(n_v) = \{u,v\}$ and $N^-(u) = N^-(v) = \{n_u, n_v\}$. In what follows, by convention, let us assume that u and n_u are in the same row, say $r \in \{0,...,n-1\}$, and v and n_v are in row $r+1 \mod n$ (note that the row numbers increase from the top of the torus to the bottom). There are two cases (depending on whether u is on the "left" or on the "right" of the "square" (u,v,n_u,n_v)), as depicted in Figure 6.3.

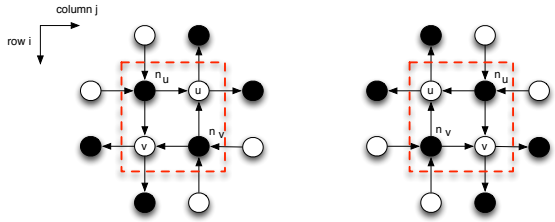


Figure 6.3: The two cases of "bad squares". Black vertices are the ones in the initial set R.

If $u \sim_R v$, then this implies that u (and similarly v) can be distinguished from any vertex different from v (resp. u). Moreover, if there are four vertices u, v, x, y such that $u \sim_R v$ and $x \sim_R y$, then $\{u, v, n_u, n_v\} \cap \{x, y, n_x, n_y\} = \emptyset$. Each such (u, v, n_u, n_v), where $u \sim_R v$, is called a *bad square*. Formally, this discussion implies:

Claim 6.4.4. *For every $u, v \in V(\vec{T}_{n\times m})$, if $u \sim_R v$, then u and v belong to the same bad square $\{u, v, n_u, n_v\}$. Moreover, all bad squares are vertex-disjoint.*

Let $\{Q_i = (u^i, v^i, n_u^i, n_v^i) \mid 1 \leq i \leq p\}$ be the set of all (vertex-disjoint) bad squares such that $u^i \sim_R v^i$ for every $i \leq p$, where p is the number of pairs of undistinguishable vertices. Let $Q = \bigcup_{i\leq p} Q_i$.

The algorithm that computes R^* from R is very simple. Start with $R^* = R$. For every $i \leq p$, remove n_v^i from R^* and add u^i to R^*. For every $1 \leq i \leq p$, let R_i be the set obtained after swapping n_v^j and u^j for every $j \leq i$ (and $R = R_0$ and $R^* = R_p$). Note that, since all bad squares are disjoint, $|R^*| = |R_i| = |R|$, for every $i \leq p$.

Remark 6.4.5. *Any vertex in R that either does not belong to a bad square or that belongs to the upper row of a bad square is also in R^*.*

The remainder of this proof aims at proving that the obtained set R^* is a resolving set, containing clearly half of the vertices of $\vec{T}_{n\times m}$.

Claim 6.4.6. *For any $x, y \in V(\vec{T}_{n\times m}) \setminus Q$ that are distinguishable by R, x and y are distinguishable by R^*.*

Proof of the claim. Let $x, y \in V(\vec{T}_{n\times m}) \setminus Q$ be distinguished by $R = R_0$, and let us prove by induction on $i \leq p$ that x and y are distinguishable by R_i. Let $i \geq 1$; by the induction hypothesis, x and y are distinguishable in R_{i-1}, so there is a vertex $q \in R_{i-1}$ such that $dist(q, x) \neq dist(q, y)$. If $q \in R_i$, then x and y are distinguished. Otherwise, $q = n_v^i$. Note that, for every vertex $w \notin Q_i$, $dist(n_v^i, w) = dist(n_u^i, w)$. Hence, $dist(n_u^i, x) \neq dist(n_u^i, y)$ and x and y can be distinguished by R_i. ◇

Claim 6.4.7. *For every $i \leq p$, every vertex in Q_i can be distinguished from any other vertex by R^*.*

Proof of the claim. Indeed, $n_u^i, u^i \in R^*$, and v^i is the only vertex not in R^* at distance 1 from $n_u^i \in R^*$. It remains to prove that n_v^i can be distinguished from any other vertex. Let us consider the case when n_v^i is the bottom-right vertex of Q_i (the case when n_v^i

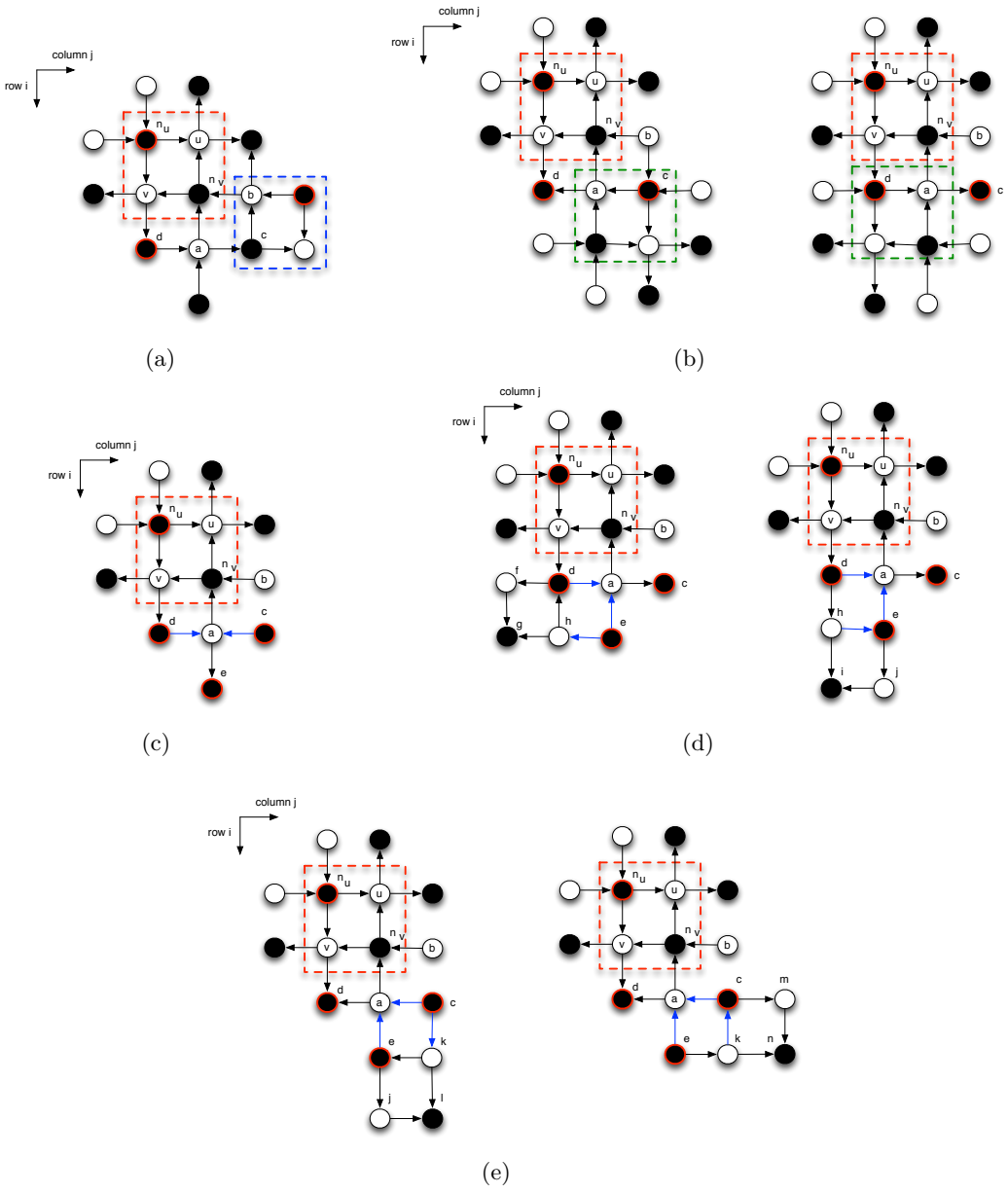


Figure 6.4: Different cases considered in the proof of Lemma 6.4.3. Black vertices are the ones of R. A black vertex circled in red is in $R \cap R^*$. Dotted squares are bad squares. Blue arcs are the ones whose orientation has been fixed depending on the cases, while the orientations of black arcs are forced.

is the bottom-left vertex of Q_i is symmetric). Let a and b be the two in-neighbours of n_v^i. Note that $a, b \notin R$. Let c be the vertex ($\neq n_v^i$) adjacent to a and b. Let d be the out-neighbour of v which is adjacent to a (via either (a, d) or (d, a)). Since the bad squares are disjoint, d cannot be in the lower row of a bad square and, so, by the above remark, $d \in R \cap R^*$; see Figure 6.4. There are several cases to be considered.

- **Case 1:** $c \notin R^*$.

  This is the case where b and c are in a same bad square (depicted in blue in Figure 6.4a). Therefore, $b \in R^*$. Moreover, this bad square and the fact that the in-degree and out-degree of every vertex is 2 force the orientation of the arcs to be in such a way that n_v^i is the only vertex at distance 1 from b and at distance 2 from d. Hence, n_v^i is distinguishable from any other vertex.

- **Case 2:** $c \in R^*$.

  - **Case 2.1:** $a \in R^*$.
    So a must be in a bad square. There are two cases depicted by the green dotted squares in Figures 6.4b. In both cases, since c and d are in R^*, n_v^i is the only vertex not in R^* that is at distance 1 from a. Hence, n_v^i is distinguishable from any other vertex.
  - **Case 2.2:** $a \notin R^*$
    Therefore, the vertex $e \notin \{c, d, n_v^i\}$ is adjacent to a and belongs to R^*. Let h be the vertex different from a that is adjacent to d and e.
    We now consider the possible values of $N^-(a)$.
    * **Case 2.2.1:** $N^-(a) = \{c, d\}$ (see Figure 6.4c).
      Since $e \in R^*$, n_v^i is the only vertex not in R^* that is at distance 2 from c and d.
    * **Case 2.2.2:** $N^-(a) = \{d, e\}$ and (e, h) is an arc (see Figure 6.4d, left).
      Since $\{a, e, h, d\}$ is not a bad square (since $a \notin R^*$), there is an arc from h to d. Note that n_v^i is at distance 2 from d and e. The only other vertex not in R^* that may be at distance 2 from d and e is the vertex g (on the left of h). In that case, the vertex $f \neq h$ that is adjacent to d and g must be such that there is an arc from f to g. Since either f or g belongs to R^*, g and n_v^i can be distinguished.
    * **Case 2.2.3:** $N^-(a) = \{d, e\}$ and (h, e) is an arc (see Figure 6.4d, right).
      Since $\{a, e, h, d\}$ is not a bad square (since $a \notin R^*$), there is an arc from d to h. Note that n_v^i is at distance 2 from d and e. The only other vertex not in R^* that may be at distance 2 from d and e is the vertex i (below h). In that case, there is an arc from h to i. Since either i or h belongs to R^*, i and n_v^i can be distinguished.
    * **Case 2.2.4:** $N^-(a) = \{c, e\}$ (see Figures 6.4e).
      Let $k \neq a$ be the vertex adjacent to c and e. The two cases, depending on whether there is the arc (c, k) or (k, c), are similar to the previous Cases 2.2.2 and 2.2.3.

This concludes the proof of the claim. ◇

Hence, in the case n, m even, R^* is a resolving set of size $\frac{nm}{2}$. In the cases when n (resp., m) is odd, we first add all the vertices of the first row (resp., of the first column) to the resolving set. The remaining vertices induce a grid with even sides on which we proceed as above. □

6.4.2 Strong Oriented Grids

In this section, we consider the maximum metric dimension of a strong oriented grid. For every such grid, we deal with its vertices using the same terminology introduced in Section 6.4.1 for tori, (*i.e.*, the vertices of the topmost row have first coordinate 0, and the vertices of the leftmost column have second coordinate 0). Our main result to be proved in this section is the following.

Theorem 6.4.8. *Let $\mathcal{G}$ be the family of grids. For any $\epsilon > 0$, we have*

$$\frac{1}{2} - \epsilon \leq WOMD(\mathcal{G}) \leq \frac{2}{3} + \epsilon.$$

We start off by exhibiting strong orientations of grids for which the metric dimension is about half of the vertices.

Lemma 6.4.9. *For every $n_0, m_0 \in \mathbb{N}$, there is $n \geq n_0$, $m \geq m_0$ and a strong orientation $\vec{G}^*$ of the grid $G_{n\times m}$ such that $MD(\vec{G}^*) \geq \frac{nm}{2} - \frac{n+m}{2}$.*

Proof. Let n (resp., m) be the smallest even integer greater or equal to n_0 (resp., m_0). We orient $G_{n\times m}$ as follows, resulting in $\vec{G}^*$. All edges of the even rows are oriented from right to left, while all edges of the odd rows are oriented from left to right. All edges of the even columns are oriented from top to bottom, while all edges of the odd columns are oriented from bottom to top. Note that $\vec{G}^*$ is indeed strong under the assumption that n and m are even (in particular, no corner vertex is a source or sink).

For every even $0 \leq i < n$ and odd $1 \leq j < m-1$, the vertices (i, j) and $(i+1, j+1)$ have the same in-neighbourhood. Similarly, for every odd $1 \leq i < n-1$ and odd $1 \leq j < m$, the vertices (i, j) and $(i+1, j-1)$ have the same in-neighbourhood. For each of these pairs of vertices, Lemma 6.2.1 implies that at least one of the two vertices must belong to any resolving set of $\vec{G}^*$. The only vertices that do not appear in these pairs are those of the form $(0, 2k)$, $(2k+1, 0)$, $(2k, m-1)$, and $(n-1, 2k+1)$ for $k \in \mathbb{N}$ and the vertices $(n-1, 0)$ and $(n-1, m-1)$. There are $n+m$ such vertices. The bound then follows. □

We now prove that every strong oriented grid has a resolving set including $\frac{2}{3}$ of the vertices.

Theorem 6.4.10. *For every strong oriented grid $\vec{G}_{n,m}$ with n rows and m columns, if $m \equiv 0 \mod 3$ or $n \equiv 0 \mod 3$, then $MD(\vec{G}_{n,m}) \leq \frac{2nm}{3}$, and $MD(\vec{G}_{n,m}) \leq \lfloor \frac{2nm}{3} \rfloor + 2m$ otherwise.*

Proof. Let us first consider the case when $m \mod 3 = 0$ (the case $n \mod 3 = 0$ is similar up to rotation). The algorithm starts with the set $R = \{V(G_{n\times m}) \setminus (i, 3j-1) | 0 \leq i \leq n-1, 1 \leq j \leq m/3\}$, (*i.e.*, R contains the first 2 out of every 3 columns from left to right in the grid) and iteratively performs local modifications (swaps one vertex in R with one of its neighbours not in R) without changing the size of R until R becomes a resolving set R^*. Note that $|R| = \frac{2nm}{3}$.

Assume R is not a resolving set (otherwise, we are done). This means that at least two vertices are not distinguishable by their distances from the vertices in R. Let u and v be two such vertices. Clearly, $u, v \notin R$ as otherwise they are distinguishable since one of them is the only vertex at distance 0 from itself.

Claim 6.4.11. *For every $u, v \in V(G_{n\times m})$, if $u \sim_R v$, then u and v belong to the same column in $\vec{G}_{n,m}$.*

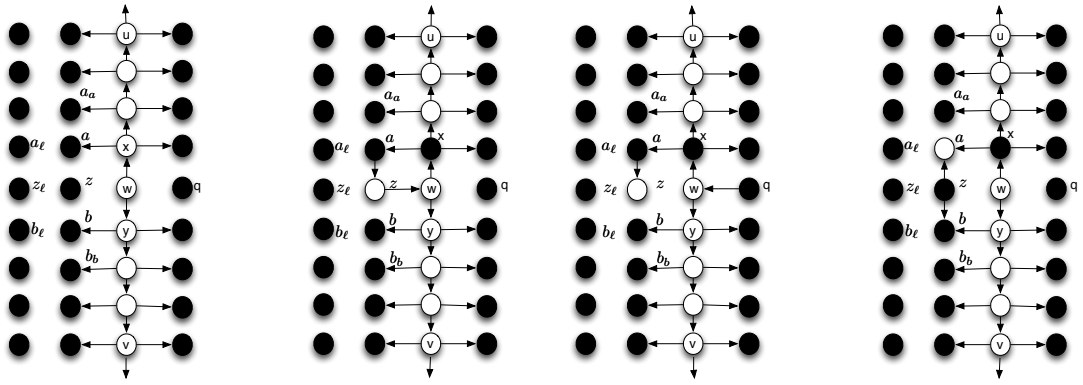
Proof of the claim. For purpose of contradiction, let us assume that $u \sim_R v$ with $u \in C_1$ and $v \in C_2$ where C_1 and C_2 are two distinct columns of $V(\vec{G}_{n,m})$ which contain no vertices in R. W.l.o.g., C_1 is to the left of C_2. Let C_1^r be the column just to the right of C_1 and let C_2^ℓ be the column just to the left of C_2. Note that all the vertices of C_1^r and C_2^ℓ are in R and that C_1^r and C_2^ℓ are distinct. Since only strong orientations are considered and C_1^r separates u from every vertex in the columns to the right of C_1^r, there exists a vertex a in C_1^r such that, for every vertex x in a column to the right of C_1^r (in particular, for every vertex in C_2^ℓ), $dist(a, u) < dist(x, u)$. Similarly, there exists a vertex b in C_2^ℓ such that, for every vertex x in a column to the left of C_2^ℓ (in particular, for every vertex in C_1^r), $dist(b, v) < dist(x, v)$. Therefore, $dist(b, v) < dist(a, v)$ and $dist(a, u) < dist(b, u)$ and, it is not possible to have both $dist(b, v) = dist(b, u)$ and $dist(a, u) = dist(a, v)$ simultaneously. Since $a, b \in R$, u and v are distinguished, a contradiction. ◇

Claim 6.4.12. *For every $u, v \in V(\vec{G}_{n,m})$ such that $u \sim_R v$, in a column C (containing no vertices in R), there is a unique vertex $w \in C$ at the same distance from u and v such that, for any $z \in R$, every shortest path from z to u (v resp.) passes through w.*

Proof of the claim. W.l.o.g., let us assume that u is in a row above v. Since u and v are not distinguishable, $dist(x, u) = dist(x, v)$ for any vertex $x \in R$. Let $z \in R$ be a vertex of R that minimizes its distance to u (and so to v). Let P_u (resp., P_v) be a shortest path from z to u (resp., v). All vertices of P_u (resp., of P_v) are not in R (by the minimality of the distance between z and u) and so are in C. The only possibility then, is that both P_u and P_v start with a common arc (z, w) (with $w \in C$ uniquely defined) and then P_u goes up to u, while P_v goes down to v.

Now, let x be any vertex in R and let Q be any shortest path from x to u. For purpose of contradiction, let us assume that Q does not pass through w. Let y be the last vertex of Q in R (possibly $y = x$). Therefore, the path Q' from y to u has all its vertices (but y) in C. In particular, if y is above u (or in the same row), Q' enters C and goes down to u, and if u is above y, Q' enters C and goes up to u. In all cases, if Q (and so Q') does not pass through w, then y must be closer to u than to v, contradicting that u and v are not distinguished. The same proof holds for any path from x to v. ◇

The vertex $w \notin R$ defined in the previous claim is called the *last common vertex* (LCV) of the two undistinguished vertices u and v. Let Q be the set of all vertices $w \in V(\vec{G}_{n,m}) \setminus R$ such that w is an LCV for two vertices $u, v \in V(\vec{G}_{n,m})$ such that $u \sim_R v$. Note that one of these vertices w may be an LCV for multiple pairs of vertices that are not distinguishable; but in these cases, the local modifications the algorithm makes are sufficient to distinguish all the vertices in all the pairs with the same LCV.



(a) (b) (c)

Figure 6.5: Configurations with two undistinguished vertices u and v. In (a) (resp., in $(b), (c)$), black vertices are those in R (resp., in R^*), and white ones are in $V(\vec{G}_{n,m}) \setminus R$ (resp., in $V(\vec{G}_{n,m}) \setminus R^*$). The vertex w is the LCV of u and v. Figures (b) depict the cases when $z \notin R^*$. Figure (c) is the case when $a \notin R^*$.

The algorithm computes R^* from R as follows. Start with $R^* = R$. For every $w \in Q$, the algorithm proceeds as follows. Let $w \in Q$ and let u and v be two undistinguished vertices such that w is their LCV (u and v exist by definition of $w \in Q$). W.l.o.g., let us assume that u is above v. Let z^w be the neighbour to the left of w, x^w be the neighbour above w, and y^w be the neighbour below w (it may be that $x^w = u$, in which case $y^w = v$) in the grid underlying $\vec{G}_{n,m}$. Also, let a^w and b^w be the neighbours above and below z^w resp. in the underlying grid. Note that any column with no vertices in R has two columns on its left, so it is the case for the column of w and so, a^w, z^w, and b^w exist. Then, the algorithm proceeds to do the following swap between a vertex in R (either z^w or a^w) and a vertex not in R (the vertex x^w):

- If (a^w, z^w) or (b^w, z^w) is an arc, then remove z^w from R^* and add x^w to R^*.
- Else, remove a^w from R^* and add x^w to R^*.

The remainder of this proof aims at proving that the obtained set R^* is a resolving set. For this purpose, we need further notations. Let $w \in Q$ be the LCV of two undistinguished vertices u and v, and let x^w, y^w, z^w, a^w, b^w be defined relative to w as above. In addition, let q^w be the neighbour to the right of w (if it exists, *i.e.*, if w is not in the rightmost column) in the underlying grid. Also, let a^w_ℓ, b^w_ℓ, and z^w_ℓ be the neighbours to the left of a^w, b^w, and z^w resp. (note that any column with no vertices in R has two columns on its left, so it is the case for the column of w and so, a^w_ℓ, b^w_ℓ, and z^w_ℓ exist) and let a^w_a and b^w_b be the neighbours above and below a^w and b^w resp. (if they exist, that is, they do not surpass the dimensions of the grid) in the underlying grid. Finally, let $H_w = \{w, z^w, a^w, b^w, a^w_\ell, b^w_\ell, z^w_\ell, a^w_a, b^w_b, q^w\} \cup \{u, v \mid u \sim_R v, w \ LCV \text{ of } u \text{ and } v\}$. All superscripts ^{w} will be omitted if there is no ambiguity.

Claim 6.4.13. *For any $w, w' \in Q$, we have $(H_w \setminus \{q^w\}) \cap (H_{w'} \setminus \{q^{w'}\}) = \emptyset$. In particular, the modifications done by the algorithm (relative to each $w \in Q$) are independent of each other.*

Proof of the claim. Let $u \sim_R v$ with w as their *LCV* and such that $dist(w,u) = dist(w,v)$ is maximum. Let C be the column of w, u, and v. As mentioned in the proof of Claim 6.4.12, there must be a directed (shortest and included in C) path from w to u and a directed (shortest and included in C) path from w to v. Moreover, Claim 6.4.12 implies that all the vertices of these paths (but w) have out-degree 3 (since all shortest paths from R to u and v go through w). In particular, u and v have out-degree 3 (unless they are in the first or last row). It is then easy to see that, if $(H_w \setminus \{q^w\}) \cap (H_{w'} \setminus \{q^{w'}\}) \neq \emptyset$ this would contradict the orientations of these arcs (see Figure 6.5). ◇

Claim 6.4.14. *For any $w \in Q$, any $s \in H_w$, and any $t \in V(\vec{G}_{n,m})$, we have $s \nsim_{R^*} t$.*

Proof of the claim. For any $w \in Q$, let $H_w = \{w, z, a, b, a_\ell, b_\ell, z_\ell, a_a, b_b, q\} \cup \{u, v \mid u \sim_R v, w \text{ } LCV \text{ of } u \text{ and } v\}$ (the superscript w's are omitted here as there is no ambiguity). Note that $x, q, a_\ell, b_\ell, z_\ell \in R^*$ due to the algorithm and so $x, q, a_\ell, b_\ell, z_\ell \in R^*$ are distinguishable from all other vertices.

Then, let P_{xu} be the directed (shortest) path from x to u (with no vertices in R) which exists by the proof of Claim 6.4.12. Let S_{xu} be the set of out-neighbours in R^* of all the vertices in P_{xu}. Every vertex r in P_{xu} is distinguishable from every other vertex by its distance to x. Indeed, if $dist(x,r) = 1$, then r can be distinguished from a since either $a \in R^*$ or a is the single vertex both at distance 1 from x and z. Otherwise, for any vertex $t \neq r$ at distance $dist(x,r)$ from x, any path from x to t crosses a vertex in $S_{xu} \subseteq R^*$ and so $r \nsim_{R^*} t$.

Now, it remains to show that every vertex in $H_w \setminus (\{x, q, a_\ell, b_\ell, z_\ell\} \cup V(P_{xu}))$ can be distinguished from all other vertices. There are two cases to be considered depending on whether z or a is not in R^*.

Case $z \notin R^*$. Then, by definition of the algorithm, (a,z) or (b,z) is an arc.

- If (a,z) is an arc, then z is distinguishable from all other vertices as it is the only vertex at distance 1 from $a \in R^*$ that is not in R^* since (x,a) is an arc (see proof of Claim 6.4.13), and $a_a, a_\ell \in R^*$ (if a_a exists) by Claim 6.4.13.
- Else, if (b,z) is an arc, then z is distinguishable from all other vertices as it is the only vertex at distance 1 from $b \in R^*$ that is not in R^* since (y,b) is an arc (see proof of Claim 6.4.13), and $b_b, b_\ell \in R^*$ (if b_b exists) by Claim 6.4.13.

Therefore, if $z \notin R^*$, it is distinguishable.

Now, we will show that all vertices on the directed (shortest) path from w to v are also distinguishable from every other vertex. Let P_{wv} be the set of vertices of the directed (shortest) path from w to v (w, v included) and let S_{wv} be the set of out-neighbours in R^* of the vertices in P_{wv}. Note that $x \in S_{wv}$. Either q exists and (w,q) or (q,w) is an arc or q does not exist and thus, (z,w) is an arc since

$\vec{G}_{n,m}$ is strong. Note that $a_a \in R^*$ ($b_b \in R^*$ resp.) if a_a exists (b_b exists resp.) by Claim 6.4.13.

- Let us first assume that (w, q) is an arc or q does not exist. Therefore, (z, w) is an arc since $\vec{G}_{n,m}$ is strong.
  - Let us first assume that (a, z) is an arc. Let $T = \{a_\ell, a_a, z_\ell, b\} \cup S_{wv} \subseteq R^*$ (or let $T = \{a_\ell, z_\ell, b\} \cup S_{wv}$ if a_a does not exist). Note that for any $t \in T$, $t \in R^*$. For any vertex $r \in P_{wv}$ and $t \in T$, we have $dist(a, r) \leq dist(t, r)$ (since by Claim 6.4.12, all shortest paths from t to r pass through w). Moreover, for any vertex $h \in V(\vec{G}_{n,m}) \setminus (P_{wv} \cup \{z\})$, there exists $t \in T$ such that $dist(a, h) > dist(t, h)$ since any shortest path from a to h passes through a vertex $t \in T$. Thus, all vertices $r \in P_{wv}$ are distinguishable from every vertex in $V(\vec{G}_{n,m}) \setminus P_{wv}$ (since it has already been shown that z is distinguishable from all other vertices). Finally, r is distinguished from every other vertex of P_{wv} by their distances from a. Hence, every vertex $r \in P_{wv}$ can be distinguished by R^* from all other vertices.
  - Let us assume that (b, z) is an arc. Let $T = \{b_\ell, b_b, z_\ell, a\} \cup S_{wv} \subseteq R^*$ (or let $T = \{b_\ell, z_\ell, a\} \cup S_{wv}$ if b_b does not exist). Note that for any $t \in T$, $t \in R^*$. For any vertex $r \in P_{wv}$ and $t \in T$, we have $dist(b, r) \leq dist(t, r)$ (since by Claim 6.4.12, all shortest paths from t to r pass through w). Moreover, for any vertex $h \in V(\vec{G}_{n,m}) \setminus (P_{wv} \cup \{z\})$, there exists $t \in T$ such that $dist(b, h) > dist(t, h)$ since any shortest path from b to h passes through a vertex $t \in T$. Thus, all vertices $r \in P_{wv}$ are distinguishable from every vertex in $V(\vec{G}_{n,m}) \setminus P_{wv}$. Finally, r is distinguished from every other vertex of P_{wv} by their distances from b. Hence, every vertex $r \in P_{wv}$ can be distinguished by R^* from all other vertices.
- Second, let us assume that (q, w) is an arc. Let $N = N^+(q) \setminus \{w\}$. Let q_r be the neighbour to the right of q in $\vec{G}_{n,m}$. Note that for all $p \in (N \setminus \{q_r\})$, $p \in R^*$ due to the algorithm and thus, only q_r may not be in R^*, which is the case if either $q_r = a^{w'}$ or $q_r = z^{w'}$ for another LCV $w' \in Q$.
  - Let us assume that $N \subseteq R^*$. Let $T = N \cup S_{wv} \cup \{a, z_\ell, b\}$. Note that $T \subseteq R^*$. As above, all vertices on the directed (shortest) path P_{wv} from w to v (w, v included) are distinguishable from any other vertex by their distances from q and from the vertices of T.
  - Let us assume that $q_r = a^{w'}$ for another LCV w' and such that $q_r \notin R^*$. Let $T = (N \setminus \{q_r)\}) \cup \{a_a^{w'}, z^{w'}, a, z_\ell, b\} \cup S_{wv}$. Note that $(x^{w'}, a^{w'})$ is an arc (by the proof of Claim 6.4.13 applied to w') and that $a_a^{w'}, z^{w'}, a, z_\ell, b \in R^*$. Then, as above, all vertices on the directed (shortest) path P_{wv} from w to v (w, v included) are distinguishable from any other vertex by their distances from q and from the vertices of T.
  - Let us assume that $q_r = z^{w'}$ for another LCV w' and such that $q_r \notin R^*$. There are two subcases: either $(w', z^{w'})$ is an arc or $(z^{w'}, w')$ is an arc.
    - Let us assume that $(w', z^{w'})$ is an arc. Let $T = (N \setminus \{q_r)\}) \cup \{a^{w'}, b^{w'}, a, z_\ell, b\} \cup S_{wv}$. Note that $a^{w'}, b^{w'}, a, z_\ell, b \in R^*$. Then, as

above, all vertices on the directed (shortest) path P_{wv} from w to v (w, v included) are distinguishable from any other vertex by their distances from q and from the vertices of T.

* Let us assume that $(z^{w'}, w')$ is an arc. By the algorithm, since $z^{w'} \notin R^*$, either $(a^{w'}, z^{w'})$ or $(b^{w'}, z^{w'})$ is an arc. W.l.o.g., let $(a^{w'}, z^{w'})$ be an arc. Let $T = (N \setminus \{q_r)\}) \cup \{a^{w'}, b^{w'}, a, z_\ell, b\} \cup S_{wv} \cup S_{w'v'}$ where $S_{w'v'}$ is defined analogously to S_{wv} for w' and $v^{w'}$. Then, as above, all vertices on the directed (shortest) path P_{wv} from w to v (w, v included) are distinguishable from any other vertex not in $P_{w'v'}$ (defined respectively to w' and v') by their distances from q and from the vertices of T. Note here that $z^{w'}$ is distinguished from all other vertices as it is the only vertex at distance 1 from both q and $a^{w'}$ that is not in R^*.
  Finally, all the vertices of the directed (shortest) path P_{wv} from w to v (w, v included) are distinguishable from all the vertices of the directed (shortest) path $P_{w'v'}$ by their distances from q and $a^{w'}$. Indeed, for any vertex $r \in P_{wv}$, $dist(q, r) < dist(a^{w'}, r)$ and for any vertex $r' \in P_{w'v'}$, $dist(q, r') \geq dist(a^{w'}, r')$.

Therefore, for any $w \in Q$, any $s \in H_w$, and any $t \in V(\vec{G}_{n,m})$ such that $z \notin R^*$, we have $s \nsim_{R^*} t$.

Case $a \notin R^*$**.** In this case, (z, a) and (z, b) are arcs. Then, a is distinguishable from all other vertices as it is the only vertex not in R^* that is at distance 1 from both $z, x \in R^*$.

The proof that all vertices on the directed (shortest) path from w to v are also distinguishable from every other vertex is analogous to the one above when $z \notin R^*$ with z taking on the role that a had in the other case for distinguishing these vertices from the rest, and so is omitted.

Therefore, for any $w \in Q$ such that $a \notin R^*$, any $s \in H_w$, and any $t \in V(\vec{G}_{n,m})$, we have $s \nsim_{R^*} t$.

$\diamond$

Claim 6.4.15. *For all vertices $s, t \in V(G_{n\times m})$ such that $s \nsim_R t$, we have $s \nsim_{R^*} t$.*

Proof of the claim. Let $s \in V(\vec{G}_{n,m}) \setminus \bigcup_{w\in Q} H_w$, let us show that s can be distinguished from every vertex $t \in V(\vec{G}_{n,m}) \setminus \bigcup_{w\in Q} H_w$ (note that, if s and/or $t \in \bigcup_{w\in Q} H_w$, the result follows from Claim 6.4.14). Note that $s \nsim_R t$ and so, there is $k \in R$ such that $dist(k, s) \neq dist(k, t)$. If $k \in R^*$, it is still the case and we are done. Otherwise, there are two cases to be considered.

- Let us first assume that $k = a^w$ for some $w \in Q$ such that $a^w = a \notin R^*$. In that case, s can still be distinguished from t by one of a_a or a_ℓ. Indeed, all shortest paths from a to any other vertex in $V(\vec{G}_{n,m})$ pass through a_a and/or a_ℓ that are in R^* (recall that, if $a \notin R^*$, it implies that there is the arc (z, a)). Therefore, if $dist(a, s) \neq dist(a, t)$ then $dist(a_a, s) \neq dist(a_a, t)$ and/or $dist(a_\ell, s) \neq dist(a_\ell, t)$.

- Second, let us assume that $k = z^w$ for some $w \in Q$ such that $z^w = z \notin R^*$.

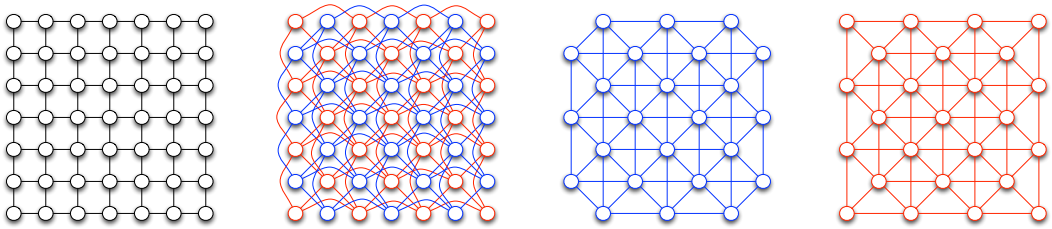
  If all z's out-neighbours are in R^*, then as above, s and t can still be distinguished by one of z's neighbours. So, let us assume that (z, w) is an arc.

  There are four remaining cases to be considered.

  - First, let us assume that there is a vertex $h \in R^*$ that is both on a shortest path from z to s and on a shortest path from z to t. This case is trivial as h distinguishes s and t since z distinguished s and t.
  - Second, let us assume that there are two vertices $h, p \in R^*$ where h is on a shortest path from z to s and p is on a shortest path from z to t where h (p resp.) is not on a shortest path from z to t (z to s resp.) as otherwise, we are in the first case. For purpose of contradiction, assume that neither h nor p can distinguish s and t. Then, $dist(h, s) = dist(h, t)$ and $dist(p, s) = dist(p, t)$. W.l.o.g., let us assume $dist(z, s) < dist(z, t)$. Then $dist(z, s) = dist(z, h) + dist(h, s) = dist(z, h) + dist(h, t) \geq dist(z, t)$, a contradiction. Therefore, h or p can distinguish s and t.
  - Then, let us consider the case when there exist a shortest path from z to s and a shortest path from z to t, both containing no vertices in R^*. In this case, both s and t must be in the same column C as w. Moreover, x cannot be on the path between z and s (resp., t) since then, it would be the first case. Therefore, both s and t are below w and one of s and t must be below the other, w.l.o.g., say t is below s, and there must exist a directed (shortest) path from w to s and from w to t that is entirely contained in C. In this case, as in Claim 6.4.14, either a or b (depending on which of the arcs (a, z) or (b, z) exists) can distinguish s and t.
  - Finally, let us assume that there is a vertex $h \in R^*$ on every shortest path from z to s and no shortest path from z to t containing a vertex in R^* (or *vice versa*). Then, t must be in the same column as w (and below w since the shortest path from z to t does not cross $x \in R^*$) and the directed shortest path from w to t is entirely contained in C. Let us assume that there is an arc (a, z) (the case when there is an arc (b, z) is similar and at least one of these cases must occur since $z \notin R^*$). Let us emphasize that no shortest path from a to t goes through a vertex in R^* (by the previous cases and since $dist(a, t) = dist(z, t) + 1$), therefore, the only shortest path from a to t goes through z and w and goes down along C until t. If there is a shortest path from a to s that passes through z, then a distinguishes s and t since z did. Otherwise, any shortest path from a to s must go through a_a or a_ℓ. If $dist(a, s) = dist(a, t)$ (otherwise, a distinguishes s and t), then $\min\{dist(a_a, s), (a_\ell, s)\} = dist(a, s) - 1$. Since clearly $\min\{dist(a_a, t), (a_\ell, t)\} > dist(a, t)$, then at least one of a_a and a_ℓ can distinguish s and t.

$\diamond$

This concludes the proof that R^* is a resolving set.



(a) $G_{9,9}$ (b) A^* (c) C_1 (d) C_2

Figure 6.6: The grid $G_{9,9}$ and the associated graph A^*.

Finally, in the case when m is not divisible by 3, we first add all the vertices of the last $x \in \{1, 2\}$ columns if $m \mod 3 = x$ to our resolving set, and then the remaining vertices induce a grid with a number of columns that is divisible by 3 on which we proceed as above. □

6.5 Further Work

In this chapter, we have investigated, for a few families of graphs, the worst strong orientations in terms of metric dimension. In particular settings, such as when considering strong Eulerian orientations of tori, we managed to identify the worst possible orientations (Theorem 6.4.1). For other families (graphs with bounded maximum degree and grids), we have exhibited both lower and upper bounds on $WOMD$ that are more or less distant apart. As further work on this topic, it would be interesting to lower the gap between our lower and upper bounds, or consider strong orientations of other graph families.

In particular, two appealing directions could be to improve Corollary 6.3.3 and Theorem 6.4.8. For graphs with maximum degree 3, we do wonder whether there are strong orientations for which the metric dimension is more than $\frac{2}{5}$ of the vertices. It is also legitimate to ask whether our upper bound ($\frac{1}{2}$ of the vertices), which was obtained from the simple technique described in Corollary 6.2.3, can be lowered further.

In Theorem 6.4.10, we proved that any strong orientation of a grid asymptotically has metric dimension at most $\frac{2}{3}$ of the vertices. Towards improving this upper bound, one could consider applying Corollary 6.2.3, for instance as follows. For a given oriented grid D, let A^* be the graph obtained as follows (where we deal with the vertices of D using the same terminology as in Section 6.4):

- $V(A^*) = V(D)$.
- We add, in A^*, an edge between two vertices (i, j) and (i', j') if they are joined by a path of length exactly 2 in the grid underlying D. That is, the edge is added whenever (i', j') is of the form $(i-1, j-1)$, $(i-2, j)$, $(i-1, j+1)$, $(i, j+2)$, $(i+1, j+1)$, $(i+2, j)$, $(i+1, j-1)$, or $(i, j-2)$.

Note that A^* has two connected components C_1, C_2 being basically obtained by glueing K_4's along edges. See Figure 6.6 for an illustration.

It can be noticed that for any oriented grid D, its auxiliary graph D_{aux} is a subgraph of A^*. From Corollary 6.2.3, any upper bound on the size of a minimum vertex cover of A^* is thus also an upper bound on $MD(D)$ (assuming D is strong, in which case it necessarily verifies $\Delta^+(D) \geq 2$). Unfortunately, we have observed that any minimum vertex cover of A^* covers $\frac{3}{4}$ of the vertices, which is not better than our upper bound in Theorem 6.4.10.

There is still hope, however, to improve our upper bound using the vertex cover method. Indeed, under the assumption that D is a strong oriented graph, actually D_{aux} can be far from having all the edges that A^* has. For instance, it can easily be proved that, in D_{aux}, it is not possible that a vertex (i, j) is adjacent to all four vertices $(i-2, j)$, $(i, j+2)$, $(i+2, j)$, $(i, j-2)$ (if they exist). Using a computer, we were actually able to check on small grids that, for all strong orientations D, the minimum vertex cover of D_{aux} has size at most $\frac{1}{2}$ of the vertices. This leads us to raising the following two questions related to our upper bound in Theorem 6.4.10:

Question 6.5.1. *For any strong orientation D of a grid $G_{n\times m}$, do the minimum vertex covers of D_{aux} have size at most $\frac{nm}{2}$?*

Question 6.5.2. *For any strong orientation D of a grid $G_{n\times m}$, do we have $MD(D) \leq \frac{nm}{2}$?*

Note that if the upper bound in Question 6.5.2 held, then it would be quite close to the lower bound we have established in Lemma 6.4.9.

Part IV

Conclusion and Further Work

Chapter 7

Perspectives

In this thesis, we have mainly studied 2-player pursuit-evasion games in graphs. Specifically, we have studied the eternal domination game and its generalization, the spy game. We have also investigated the metric dimension of oriented graphs and a sequential version of the metric dimension of a graph, called the Localization problem. We studied the complexity of some of these problems, showing the spy game to be NP-hard, and the Localization and Relative-Localization problems to be NP-complete even when the number of turns or number of probings per turn are fixed but not both. Otherwise, our approach has been to study these problems in particular graph classes.

Our most notable results were for trees in the spy game and the Localization problem. In particular, for the spy game, we proved that the fractional guard number is equal to the "integral" guard number in trees. This allowed us to use Linear Programming to determine the guard number and a corresponding strategy in trees in polynomial time and without this method we were not able to solve the problem in trees. As mentioned before, this is the first exact algorithm, as far as we know, that uses a fractional relaxation in combinatorial games to solve the "integral" version of the game. With this being said, we believe using Linear Programming and fractional relaxations could be fruitful in terms of solving other combinatorial games.

Our second most notable result was a polynomial-time $(+1)$-approximation algorithm for the Localization problem in trees. Precisely, we showed that the problem is NP-complete in trees, but that the "difficulty" of the problem originates from the first turn of probing vertices. That is, in trees, given any arbitrary first turn of probing vertices as input, we came up with an exact polynomial-time algorithm that solves the problem from the second turn on. This result is of particular interest as, as far as we know, $(+1)$-approximation algorithms are rare.

In terms of further work, aside from the open problems and directions mentioned in the conclusions of Chapters 3, 4, 5, and 6 of this thesis, the following problems are of particular interest and some are stated explicitly below. The exact complexity of the eternal domination game and the spy game are yet to be determined with both only known to be NP-hard. It would be interesting if both of these games are EXPTIME-complete like cops and robbers and it seems likely that they are at least PSPACE-hard like many games on graphs. It is also of interest to completely resolve the case of Cartesian grids for both the eternal domination game and the spy game as this class of

graphs is of particular interest in general but especially for the first game as was shown in the state of the art. Lastly, as should be mentioned in any thesis related to cops and robbers, the famed Meyniel's conjecture still remains wide open. That is, for an n-node connected graph G and an $\epsilon > 0$, no one has even been able to manage to prove that $c(G) = O(n^{1-\epsilon})$.

Question 7.0.1 (**Complexity of the eternal domination game).** *Given a graph G and an integer $k > 0$ as inputs, what is the computational complexity of determining whether $\gamma^{\infty}_{all}(G) \leq k$? It is known to be NP-hard, so is it NP-complete? PSPACE-complete? EXPTIME-complete?*

Question 7.0.2 (**Complexity of the spy game).** *Given a graph G and an integer $k > 0$ as inputs and two fixed integers $s > 1$ and $d \geq 0$, what is the computational complexity of determining whether $gn_{s,d}(G) \leq k$? It is known to be NP-hard, so is it NP-complete? PSPACE-complete? EXPTIME-complete?*

Question 7.0.3 (**Eternal domimation in Cartesian grids).** *For an $n \times m$ Cartesian grid $G_{n\times m}$, is it true that $\gamma^{\infty}_{all}(G_{n\times m}) = \gamma(G_{n\times m}) + O(1)$?*

Conjecture 7.0.4 (**Meyniel's Conjecture).** *For any n-node connected graph G, $c(G) = O(\sqrt{n})$.*

Apart from the results detailed in this manuscript, I have studied the outcomes and complexity of a scoring colouring game on graphs called the orthogonal colouring game [s-13, j-1]. I have studied two different variants of the game of cops and robbers, one called hyperopic cops and robbers [j-2], and the other called wall cops and robbers [c-7]. Using a different technique than the one presented in Chapter 4, for the eternal domination game in strong grids, we obtained a better result for "smaller" grids (but a weaker asymptotic result) in [s-15]. Lastly, I have studied an edge-weighting problem closely related to the 1-2-3 Conjecture [83] in [s-14].

In the next 5 to 10 years, I see my research continuing in combinatorial games on graphs but also expanding to other problems in graph theory. In particular, algorithmic complexity results interest me. Thus, questions like the complexity of the eternal domination game and the spy game are appealing. Also, the famed Meyniel's conjecture is another problem I would like to take some more time to tackle. Although I have not really done any work on parameterized complexity, I think this will be a future direction for my research in the coming years as well.

Bibliography

My Bibliography

— International journals —

[j-1] S. D. Andres, M. Huggan, F. Mc Inerney, and R. J. Nowakowski. The orthogonal colouring game. *To appear in Theoretical Computer Science*, 2019.

[j-2] A. Bonato, N. Clarke, D. Cox, S. Finbow, F. Mc Inerney, and M. E. Messinger. Hyperopic cops and robbers. *To appear in Theoretical Computer Science*, 2019.

[j-3] N. Cohen, N. A. Martins, F. Mc Inerney, N. Nisse, S. Pérennes, and R. Sampaio. Spy-game on graphs: Complexity and simple topologies. *Theoretical Computer Science*, 725:1–15, 2018.

[j-4] N. Cohen, F. Mc Inerney, N. Nisse, and S. Pérennes. Study of a combinatorial game in graphs through linear programming. *To appear in Algorithmica*, 2019.

— International conferences —

[c-5] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and Stéphane Pérennes. Sequential metric dimension. In *Proceedings of the 16th Workshop on Approximation and Online Algorithms, WAOA 2018*, 2018. RR, `https://hal.inria.fr/hal-01717629`.

[c-6] J. Bensmail, F. Mc Inerney, and N. Nisse. Metric dimension: From graphs to digraphs. In *Proceedings of the 10th Latin & American Algorithms, Graphs and Optimization Symposium, LAGOS 2019*, 2019. RR, `https://hal.inria.fr/hal-01938290`.

[c-7] A. Bonato and F. Mc Inerney. The game of wall cops and robbers. In *Proceedings of the 2nd International Conference on Computational Intelligence, ICC3 2015*, volume 412 of *Advances in Intelligent Systems and Computing*. Springer, 2015.

[c-8] N. Cohen, F. Mc Inerney, N. Nisse, and S. Pérennes. Study of a combinatorial game in graphs through linear programming. In *Proceedings of the 28th International Symposium on Algorithms and Computation (ISAAC 2017)*, pages 22:1–22:13, 2017. RR, `https://hal.archives-ouvertes.fr/hal-01462890`.

[c-9] F. Mc Inerney, N. Nisse, and S. Pérennes. Eternal domination in grids. In *Proceedings of the 11th International Conference on Algorithms and Complexity (CIAC 2019)*, pages 311–322, 2019. RR, `https://hal.inria.fr/hal-01790322v2`.

— National conferences —

[c-10] J. Bensmail, D. Mazauric, F. Mc Inerney, N. Nisse, and S. Pérennes. Localiser une cible dans un graphe. In *In 20es Rencontres Francophones sur les aspects Algorithmiques des Télécommunications AlgoTel 2018*, 2018.

[c-11] J. Bensmail, F. Mc Inerney, and N. Nisse. Dimension métrique des graphes orientés. In *In 21es Rencontres Francophones sur les aspects Algorithmiques des Télécommunications AlgoTel 2019*, 2019.

[c-12] N. Cohen, N. Martins, F. Mc Inerney, N. Nisse, S. Pérennes, and R. Sampaio. Enquêter dans les graphes. In *In 19es Rencontres Francophones sur les aspects Algorithmiques des Télécommunications AlgoTel 2017*, 2017.

— Submitted papers —

[s-13] S. D. Andres, F. Dross, M. Huggan, F. Mc Inerney, and R. J. Nowakowski. On the complexity of orthogonal colouring games and the np-completeness of recognising graphs admitting a strictly matched involution. *Submitted to Discrete Applied Mathematics*. RR, `https://hal.inria.fr/hal-02053265`.

[s-14] J. Bensmail, F. Mc Inerney, and K. Szabo Lyngsie. On $\{a, b\}$-edge-weightings of bipartite graphs with odd a, b. *Submitted to DMTCS*. RR, `https://hal.inria.fr/hal-01988399`.

[s-15] A. Gagnon, A. Hassler, J. Huang, A. Krim-Yee, F. Mc Inerney, A. M. Zacarías, B. Seamone, and V. Virgile. A method for eternally dominating strong grids. *Submitted to DMTCS*. RR, `https://hal.inria.fr/hal-02004770`.

References

---

[16] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8:1–12, 1984.

[17] N. Alon, D. Moshkovitz, and S. Safra. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms*, 2(2):153–177, 2006.

[18] J. Arquilla and H. Fredricksen. "graphing" an optimal grand strategy. *Military Operations Research*, 1(3):3–17, 1995.

[19] G. Bagan, A. Joffard, and H. Kheddouci. Eternal dominating sets on digraphs and orientations of graphs. *CoRR*, abs/1805.09623, 2018.

[20] R. F. Bailey and P.J. Cameron. Base size, metric dimension and other invariants of groups and graphs. *Bulletin of the London Mathematical Society*, 43:209–242, 2011.

[21] S. Bard, C. Duffy, M. Edwards, G. Macgillivray, and F. Yang. Eternal domination in split graphs. *J. Comb. Math. Comb. Comput.*, 101:121–130, 2017.

[22] I. Beaton, S. Finbow, and J.A. MacDonald. Eternal domination numbers of $4 \times n$ grid graphs. *J. Comb. Math. Comb. Comput.*, 85:33–48, 2013.

[23] R. Belmonte, F. V. Fomin, P. A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM J. Discrete Mathematics*, 31(2):1217–1243, 2017.

[24] Y. Ben-Haim, S. Gravier, A. Lobstein, and J. Moncel. Adaptive identification in graphs. *J. Comb. Theory, Ser. A*, 115(7):1114–1126, 2008.

[25] A. Berarducci and B. Intrigila. On the cop number of a graph. *Advances in Applied Mathematics*, 14:389–403, 1993.

[26] H. L. Bodlaender. On the complexity of some coloring games. *Internat. J. Found. Comput. Sci.*, 2:133–147, 1991.

[27] H. L. Bodlaender and T. Kloks. Efficient and constructive algorithms for the pathwidth and treewidth of graphs. *Journal of Algorithms*, 21:358–402, 1996.

[28] A. Bonato and W. B. Kinnersley. Bounds on the localization number. *CoRR*, abs/1806.05286, 2018.

[29] A. Bonato and R. Nowakowski. *The game of Cops and Robbers on Graphs*. American Math. Soc., 2011.

[30] J. A. Bondy and U. S. R. Murty. *Graph theory*, volume 244 of *Graduate Texts in Mathematics*. Springer, 2008.

[31] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokół, and M. Sleszynska-Nowak. Centroidal localization game. *To appear in Electronic Journal of Combinatorics*, 2017.

[32] B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokół, and M. Sleszynska-Nowak. Localization game on geometric and planar graphs. *To appear in Discrete Applied Mathematics*, 2017.

[33] C. L. Bouton. Nim, a game with a complete mathematical theory. *Annals of Mathematics*, 3(14):35–39, 1901.

[34] A. Braga, C. Souza, and O. Lee. The eternal dominating set problem for proper interval graphs. *Information Processing Letters*, 115, 2015.

[35] A. Brandt, J. Diemunsch, C. Erbes, J. LeGrand, and C. Moffatt. A robber locating strategy for trees. *Discrete Applied Mathematics*, 232:99 – 106, 2017.

[36] B. Brešar, P. Dorbec, W. Goddard, and B. L. Hartnell. Vizing's conjecture: A survey and recent results. *Journal of Graph Theory*, 69(1):46–76, 2012.

[37] B. Brešar, P. Dorbec, S. Klavžar, G. Košmrlj, and G. Renault. Complexity of the game domination problem. *Theoretical Computer Science*, 648(4):1–7, 2016.

[38] B. Brešar, S. Klavžar, and D. F. Rall. Domination game and an imagination strategy. *SIAM J. Discrete Mathematics*, 24:979–991, 2010.

[39] B. Brešar, S. Klavžar, and D. F. Rall. Domination game played on trees and spanning subgraphs. *Discrete Mathematics*, 313:915–923, 2013.

[40] A. Burger, E. J. Cockayne, W. R. Gründlingh, C. M. Mynhardt, J. H. van Vuuren, and W. Winterbach. Infinite order domination in graphs. *J. Comb. Math. Comb. Comput.*, 50:179–194, 2004.

[41] J. M. Carraher, I. Choi, M. Delcourt, L. H. Erickson, and D. B. West. Locating a robber on a graph via distance queries. *Theoretical Computer Science*, 463:54–61, 2012.

[42] G. Chartrand, L. Eroh, M. Johnson, and O. Oellermann. Resolvability in graphs and the metric dimension of a graph. *Discrete Applied Mathematics*, 105(1-3):99–113, 2000.

[43] G. Chartrand, M. Rains, and P. Zhang. The directed distance dimension of oriented graphs. *Mathematica Bohemica*, 125:155–168, 2000.

[44] G. Chartrand, M. Rains, and P. Zhang. On the dimension of oriented graphs. *Utilitas Mathematica*, 60:139–151, 2001.

[45] R. Chérifi, S. Gravier, X. Lagraula, C. Payan, and I. Zighem. Domination number of the cross product of paths. *Discrete Applied Mathematics*, 94(1-3):101–139, 1999.

[46] E. Chiniforooshan. A better bound for the cop number of general graphs. *Journal of Graph Theory*, 58(1):45–48, 2008.

[47] W. E. Clark and S. Suen. An inequality related to vizing's conjecture. *Electronic Journal of Combinatorics*, 7(1), 2000.

[48] N. E. Clarke. *Constrained Cops and Robber*. Ph.d. thesis, Dalhousie University, Canada, 2002.

[49] A. Z. Delaney and M. E. Messinger. Closing the gap: Eternal domination on $3 \times n$ grids. *Contributions to Discrete Mathematics*, 12(1):47–61, 2017.

[50] J. Díaz, O. Pottonen, M. J. Serna, and E. J. van Leeuwen. Complexity of metric dimension on planar graphs. *J. Comput. Syst. Sci.*, 83(1):132–158, 2017.

[51] R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.

[52] P. Dorbec, G. Košmrlj, and G. Renault. The domination game played on unions of graphs. *Discrete Mathematics*, 338:71–79, 2015.

[53] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness. ii. on completeness for w[1]. *Theoretical Computer Science*, 141:109–131, 1995.

[54] R. G. Downey and M. R. Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

[55] P. A. Dreyer. *Applications and variations of domination in graphs*. Ph.d. thesis, Rutgers University, U.S.A., 2000.

[56] H. E. Dudeney. *The Canterbury puzzles, puzzle 73*. Dover, 1908.

[57] M. Fehr, S. Gosselin, and O. R. Oellermann. The metric dimension of cayley digraphs. *Discrete Mathematics*, 306:31–41, 2006.

[58] M. Feng, M. Xu, and K. Wang. On the metric dimension of line graphs. *Discrete Applied Mathematics*, 161:802–805, 2013.

[59] S. Finbow, M. E. Messinger, and M. F. van Bommel. Eternal domination in $3 \times n$ grids. *Australas. J. Combin.*, 61:156–174, 2015.

[60] F. V. Fomin, F. Giroire, A. Jean-Marie, D. Mazauric, and N. Nisse. To satisfy impatient web surfers is hard. *Theoretical Computer Science*, 526:1–17, 2014.

[61] F. V. Fomin, P. A. Golovach, J. Kratochvíl, N. Nisse, and K. Suchan. Pursuing a fast robber on a graph. *Theoretical Computer Science*, 411(7-9):1167–1181, 2010.

[62] F. Foucaud, R. Klasing, and P. J. Slater. Centroidal bases in graphs. *Networks*, 64(2):96–108, 2014.

[63] F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. i. bounds. *Theoretical Computer Science*, 668:43–58, 2017.

[64] F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. algorithms and complexity. *Algorithmica*, 78(3):914–944, 2017.

[65] Peter Frankl. Cops and robbers in graphs with large girth and cayley graphs. *Discrete Applied Mathematics*, 17:301–305, 1987.

[66] M. Gardner. Mathematical games - the fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223:120–123, 1970.

[67] M. Gardner. Mathematical games: Cram, crosscram and quadraphage: new games having elusive winning strategies. *Scientific American*, 230(2):106–108, 1974.

[68] M. R. Garey and D. S. Johnson. *Computers and Intractability - A guide to NP-completeness*. W.H. Freeman and Company, 1979.

[69] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1990.

[70] T. Gavenciak, P. Gordinowicz, V. Jelínek, P. Klavík, and J. Kratochvíl. Cops and robbers on intersection graphs. *European Journal of Combinatorics*, 72:45–69, 2018.

[71] F. Giroire, N. Nisse, S. Pérennes, and R. P. Soares. Fractional combinatorial games. Technical report, INRIA, 2013. RR8371, `http://hal.inria.fr/hal-00865345`.

[72] W. Goddard, S. M. Hedetniemi, and S. T. Hedetniemi. Eternal security in graphs. *J. Comb. Math. Comb. Comput.*, 52:160–180, 2005.

[73] W. Goddard and M. A. Henning. Domination in planar graphs with small diameter. *Journal of Graph Theory*, 40(1):1–25, 2002.

[74] J. L. Goldwasser, W. F. Klostermeyer, and C. M. Mynhardt. Eternal protection in grid graphs. *Utilitas Mathematica*, 91:47–64, 2013.

[75] D. Gonçalves, A. Pinlou, M. Rao, and S. Thomassé. The domination number of grids. *SIAM J. Discrete Mathematics*, 25(3):1443–1453, 2011.

[76] S. Gravier and A. Khelladi. On the domination number of cross products of graphs. *Discrete Mathematics*, 145(1-3):273–277, 1995.

[77] F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combinatoria*, 2:191–195, 1976.

[78] S. Hartung and A. Nichterlein. On the parameterized and approximation hardness of metric dimension. In *Proceedings of the 28th Conference on Computational Complexity, CCC*, pages 266–276. IEEE Computer Society, 2013.

[79] J. Haslegrave, R. A. B. Johnson, and S. Koch. Locating a robber with multiple probes. *Discrete Mathematics*, 341(1):184–193, 2018.

[80] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater. *Fundamentals of Domination in Graphs*. Marcel Dekker, New York, 1998.

[81] A. Isaza, J. Lu, V. Bulitko, and R. Greiner. A cover-based approach to multi-agent moving target pursuit. In *4th Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2008.

[82] Y. Kakuda, T. Kikuno, and N. Yoshida. A linear algorithm for the domination number of a series-parallel graph. *Discrete Applied Mathematics*, 5(3):299–311, 1983.

[83] M. Karoński, T. Łuczak, and A. Thomason. Edge weights and vertex colours. *J. Comb. Theory, Ser. B*, 91:151–157, 2004.

[84] R. M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[85] M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. Information Theory*, 44(2):599–611, 1998.

[86] A. Kehagias and G. Konstantinidis. Selfish cops and active robber: Multi-player pursuit evasion on graphs. *To appear in Theoretical Computer Science*, 2019.

[87] W. B. Kinnersley. Cops and robbers is exptime-complete. *J. Comb. Theory, Ser. B*, 111:201–220, 2015.

[88] W. F. Klostermeyer, M. Lawrence, and G. MacGillivray. Dynamic dominating sets: the eviction model for eternal domination. Manuscript, 2014.

[89] W. F. Klostermeyer and G. MacGillivray. Eternal dominating sets in graphs. *J. Comb. Math. Comb. Comput.*, 68:97–111, 2009.

[90] W. F. Klostermeyer and C. M. Mynhardt. Eternal total domination in graphs. *Ars Combinatoria*, 68:473–492, 2012.

[91] W. F. Klostermeyer and C. M. Mynhardt. Protecting a graph with mobile guards. *Applicable Analysis and Discrete Mathematics*, 10, 2014.

[92] A. Kosowski, B. Li, N. Nisse, and K. Suchan. k-chordal graphs: From cops and robber to compact routing via treewidth. *Algorithmica*, 72(3):758–777, 2015.

[93] R. Kusters. Memoryless determinacy of parity games. *Automata, Logics, and Infinite Games: A Guide to Current Research*, LNCS, 2500:95–106, 2002.

[94] I. Lamprou, R. Martin, and S. Schewe. Perpetually dominating large grids. In *10th Int. Conf. on Algorithms and Complexity (CIAC 2017)*, pages 393–404, 2017.

[95] U. Larsson, R. J. Nowakowski, J. P. Neto, and C. P. Santos. Guaranteed scoring games. *Electronic Journal of Combinatorics*, 23, 2016. Paper 3.27.

[96] U. Larsson, R. J. Nowakowski, and C. P. Santos. Game comparison through play. *Theoretical Computer Science*, 725:52–63, 2018.

[97] U. Larsson, R. J. Nowakowski, and C. P. Santos. Games with guaranteed scores and waiting moves. *Internat. J. Game Theory*, 47:653–671, 2018.

[98] A. Lozano. Symmetry breaking in tournaments. *The Electronic Journal of Combinatorics*, 20(1):Paper #P69, 2013.

[99] L. Lu and X. Peng. On Meyniel's conjecture of the cop number. *Journal of Graph Theory*, 71(2):192–205, 2012.

[100] G. MacGillivray and K. Seyffarth. Domination numbers of planar graphs. *Journal of Graph Theory*, 22(3):213–229, 1996.

[101] M. Mamino. On the computational complexity of a game of cops and robbers. *Theoretical Computer Science*, 477:48–56, 2013.

[102] P. Manuel, B. Rajan, I. Rajasingh, and M. C. Monica. Landmarks in torus networks. *Journal of Discrete Mathematical Sciences and Cryptography*, 9(2):263–271, 2013.

[103] J. F. Mertens. Stochastic games. *Handbook of game theory with economic applications*, 3:1809–1832, 2002.

[104] R. J. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43:235–239, 1983.

[105] S. Pancahayani and R. Simanjuntak. Directed metric dimension of oriented graphs with cyclic covering. *J. Comb. Math. Comb. Comput.*, 94:15–25, 2015.

[106] A. Quilliot. *Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes.* Thèse de doctorat d'état, Université de Paris VI, France, 1983.

[107] B. Rajan, I. Rajasingh, J.A. Cynthia, and P. Manuel. Metric dimension of directed graphs. *International Journal of Computer Mathematics*, 91(7):1397–1406, 2014.

[108] C. S. Revelle. Can you protect the roman empire? *Johns Hopkins Magazine*, 50(2), 1997.

[109] C. S. Revelle and K. E. Rosing. Defendens imperium romanum: A classical problem in military strategy. *American Mathematical Monthly*, 107:585–594, 2000.

[110] M. Rinemberg and F. J. Soulignac. The eternal dominating set problem for interval graphs. *Information Processing Letters*, 146:27–29, 2019.

[111] H. E. Robbins. A theorem on graphs, with an application to a problem in traffic control. *American Mathematical Monthly*, 46:281–283, 1939.

[112] B. S. W. Schröder. The copnumber of a graph is bounded by $\lfloor \frac{3}{2} genus(g) \rfloor + 3$. *Categorical perspectives (Kent, OH, 1998), Trends in Mathematics*, pages 243–263, 2001.

[113] A. Scott and B. Sudakov. A bound for the cops and robbers problem. *SIAM J. Discrete Mathematics*, 25(3):1438–1442, 2011.

[114] S. M. Seager. Locating a robber on a graph. *Discrete Mathematics*, 312(22):3265–3269, 2012.

[115] S. M. Seager. A sequential locating game on graphs. *Ars Combinatoria*, 110:45–54, 2013.

[116] S. M. Seager. Locating a backtracking robber on a tree. *Theoretical Computer Science*, 539:28–37, 2014.

[117] P. J. Slater. Leaves of trees. In *Proceedings of the Sixth Southeastern Conference on Combinatorics, Graph Theory, and Computing (Florida Atlantic Univ., Boca Raton, Fla., 1975)*, pages 549–559. Congressus Numerantium, No. XIV. Utilitas Mathematica, Winnipeg, Man., 1975.

[118] Peter J. Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987.

[119] I. Stewart. Defend the roman empire! *Scientific American*, pages 136–138, 1999.

[120] C. M. van Bommel and M. F. van Bommel. Eternal domination numbers of $5 \times n$ grid graphs. *J. Comb. Math. Comb. Comput.*, 97:83–102, 2016.

[121] J. M. M. van Rooij and H. L. Bodlaender. Exact algorithms for dominating set. *Discrete Applied Mathematics*, 159(17):2147–2164, 2011.

[122] N. Vieille. Stochastic games: Recent results. *Handbook of game theory with economic applications*, 3:1833–1850, 2002.

[123] V. G. Vizing. Some unsolved problems in graph theory. *Uspehi Mat. Nauk*, 23(6):117–134, 1968.