ALGORITMOS EXATOS E COMPLEXIDADE COMPUTACIONAL PARA O PROBLEMA DO CONJUNTO GEODÉTICO FORTE

JOÃO HENRIQUE GONÇALVES DE SOUSA

ALGORITMOS EXATOS E COMPLEXIDADE COMPUTACIONAL PARA O PROBLEMA DO CONJUNTO GEODÉTICO FORTE

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Vinícius Fernandes dos Santos
Coorientador: Sebastián Alberto Urrutia

Belo Horizonte

Dezembro de 2018

JOÃO HENRIQUE GONÇALVES DE SOUSA

EXACT ALGORITHMS AND COMPUTATIONAL COMPLEXITY FOR THE STRONG GEODETIC SET PROBLEM

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Vinícius Fernandes dos Santos
Co-Advisor: Sebastián Alberto Urrutia

Belo Horizonte

December 2018

Ficha catalográfica elaborada pela Biblioteca do ICEx - UFMG

Sousa, João Henrique Gonçalves de

S725e Exact algorithms and computational complexity for the strong geodetic set problem / João Henrique Gonçalves de Sousa — Belo Horizonte, 2018.
xx, 48 p.: il.; 29 cm.

Dissertação (mestrado) - Universidade Federal de Minas Gerais – Departamento de Ciência da Computação.

Orientador: Vinícius Fernandes dos Santos
Coorientador: Sebastián Alberto Urrutia

1. Computação – Teses. 2. Conjunto geodético forte. 3. Teoria dos grafos. 4. Algoritmos exatos. 5. Problemas NP-completo. 6. Caminhos mínimos. I. Orientador. II. Coorientador. III. Título.

CDU 519.6*52(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Exact algorithms and computational complexity for the strong geodetic set problem

JOÃO HENRIQUE GONÇALVES DE SOUSA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Vinicius Fernandes dos Santos

PROF. VINÍCIUS FERNANDES DOS SANTOS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. SEBASTIÁN ALBERTO URRUTIA - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. GABRIEL DE MORAIS COUTINHO
Departamento de Ciência da Computação - UFMG

PROF. MITRE COSTA DOURADO
Instituto de Matemática. - UFRJ

Carlos Vinícius Gomes Costa Lima

DR. CARLOS VINÍCIUS GOMES COSTA LIMA
DCC- Bolsista PNPD/CAPES - UFMG

Belo Horizonte, 18 de dezembro de 2018.

Dedico este trabalho aos meus pais e irmão, que me deram forças para enfrentar cada desafio que tive neste período. Todo sentimento de apreensão ou insegurança se tornaram pequenos diante deste apoio incondicional, me sinto privilegiado e extremamente agradecido por isso.

Acknowledgments

Agradeço aos meus orientadores, que acompanharam de perto e empregaram esforços para a realização de cada resultado presente neste trabalho. Sou grato especialmente pela paciência e perseverança do meu orientador Vinícius dos Santos, que não deixou que eu desistisse do principal resultado apresentado neste trabalho, mesmo após inúmeras tentativas frustradas. Agradeço também ao Carlos Vinícius G. C. Lima que contribuiu imensamente ao trabalho com produtivas discussões e revisões.

Agradeço aos colegas de laboratório (LAPO) e professores do DCC, que se mostraram sempre solícitos contribuindo com diversas discussões enriquecedoras. Cada pessoa envolvida e cada incentivo foram essenciais para a realização deste trabalho.

Agradeço ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro para a realização deste projeto.

Resumo

Investigamos o problema do conjunto geodético forte (CGF), um problema NP-completo de convexidade em grafos cujo objetivo é encontrar um conjunto geodético forte mínimo. Um conjunto geodético forte é um conjunto de vértices S no qual é possível atribuir um único caminho mínimo para cada par de vértices em S de forma que todos os vértices do grafo pertençam a pelo menos um de tais caminhos mínimos. Obtivemos resultados referentes à complexidade de tempo. Nossa abordagem se baseia na análise do problema para diferentes classes de grafos, possibilitando a construção de uma hierarquia que especifica a complexidade do problema para grafos bipartidos, grafos co-bipartidos, grafos cordais, grafos blocos, grafos *threshold*, grafos cactos e árvores. Além disso, estudamos o problema sob o ponto de vista de complexidade parametrizada, onde concluímos que o mesmo é tratável por parâmetro fixo quando os parâmetros são o diâmetro do grafo e o parâmetro natural em conjunto. Também foi provado que o PCGF parametrizado pelo diâmetro não está em XP, fortalecendo o resultado anterior. Definimos o problema do reconhecimento de conjuntos geodéticos fortes (RCGF), que consiste em decidir se um dado conjunto de vértices S é um conjunto geodético forte. Obtivemos uma prova de NP-completude do RCGF utilizando-se de uma redução a partir de uma variante do problema de satisfatibilidade. Finalmente, valendo-se da forte relação entre os dois problemas citados, determinamos a complexidade computacional do RCGF para grafos bipartidos, grafos blocos, grafos com diâmetro 2, grafos *split*, grafos cactos e árvores.

Palavras-chave: conjunto geodético forte, NP-completude, caminhos mínimos, classes de grafos, complexidade parametrizada, algoritmos exatos.

Abstract

We studied the strong geodetic set problem (SGS), an NP-complete convexity problem in graphs that asks for a minimum strong geodetic set. A strong geodetic set is a vertex set so that it is possible to cover all vertices of the graph by assigning a unique shortest path for each vertex pair in the set. We achieved results concerning the computational complexity of this problem. Our approach is based mainly on analyzing the problem for different graph classes. It is possible to construct a comprehensive graph class hierarchy specifying the complexity of the problem for bipartite graphs, co-bipartite graphs, chordal graphs, block graphs, threshold graphs, cacti graphs and trees. In addition, we state that the SGS is fixed parameter tractable when the parameters are the graph's diameter and the natural parameter together. Besides that, we prove that the SGS parameterized by the diameter is not in XP, strengthening the previous result. Moreover, we define the strong geodetic set recognition problem (SGSR), which can be seen as a subproblem of the SGS. We obtained a proof of the NP-completeness of this problem by presenting a polynomial reduction from a variant of the satisfiability problem. Finally, given the strong relation between the two cited problems, we could also derive the complexity status of the SGSR for bipartite graphs, block graphs, graphs with diameter 2, split graphs, cacti graphs and trees.

Keywords: strong geodetic set, NP-completeness, shortest paths, graph classes, parameterized complexity, exact algorithms.

List of Figures

Contents

Chapter 1

Introduction

With the increasing use of social networks, vehicular networks and complex networks in general, a great demand for solving graph problems arises. One important class of graph problems consists in determining efficient ways to cover elements of a graph. For instance, the vertex cover problem asks for the minimum cardinality of a set of vertices that can cover all edges of the graph, Grandoni [2006]. This problem models several real world applications, for example, how to allocate the minimum number of cameras to supervise halls of a building.

We are focused on a different kind of covering problem. A paper by Harary et al. [1993] introduced a new paradigm of covering problems involving shortest paths. It consists in finding a smallest set S of vertices of a graph G that has the following property: every vertex of G lies on a shortest path between 2 vertices in S. The geodetic number denotes the cardinality of S. Although it works essentially with shortest paths, it is computationally complex, i.e. NP-Hard, Dourado et al. [2010]. The authors also determine the geodetic number of complete graphs, star graphs, trees, cycles and complete bipartite graphs. It is worth to mention that Dourado et al. [2010] proved the NP-completeness of the geodetic number problem for chordal graphs and for chordal bipartite graphs. Moreover, the authors determined the geodetic number for split graphs.

The GEODETIC NUMBER problem lies in the category of graph convexity problems, which, in general, asks whether it is possible to cover the vertices of a graph respecting some properties. Some of these problems are: HULL SET Everett and Seidman [1985], ISOMETRIC PATH Pan and Chang [2006], STRONG EDGE GEODETIC SET Manuel et al. [2016] and STEINER SET Hernando et al. [2005]. The literature disposes of several important results on these topics, for instance: HULL SET , STRONG EDGE GEODETIC SET and STEINER SET have been proved to be NP-complete. The hardness of these problems motivates several works on the subject.

At Hernando et al. [2005] the authors consider some convexity problems and bring an extensive overview about them. The authors also studied the relation between Steiner, geodetic and hull numbers of a graph, showing that every Steiner set in a connected graph must be monophonic and that every Steiner set in a connected interval graph must be a geodetic set as well. In this dissertation we study another convexity problem, the STRONG GEODETIC SET problem, Manuel et al. [2018].

1.1 Motivation

In Harary et al. [1993], the GEODETIC SET problem was defined and, later, the following practical motivation for the problem was given in Manuel et al. [2018]: a social network has a set of communicating users, in which the vertices represent the users and the edges indicate the possibility of direct communication between two users. Moreover, any communication between users should occur by passing messages through a shortest path. Moreover, there are monitors on the network, so that every user must be in some shortest path between two monitors, which are positioned on vertices of the network. The question is: what is the smallest number of monitors we should allocate at the network so that all users are monitored by at least a pair of monitors.

The STRONG GEODETIC SET problem (SGS) was defined in Manuel et al. [2018]. The problem is motivated similarly, but now each pair of monitors has the ability to supervise users present in a unique shortest path between them (there can be multiple shortest paths between two vertices). Finally, the question is to find the minimum number of monitors needed to cover the network. The STRONG GEODETIC NUMBER of a graph G, $sgn(G)$, refers to the size of a minimum strong geodetic set of G. A formal definition of the problem is given at Section 2.2.

In Manuel et al. [2018], the authors showed that the STRONG GEODETIC SET problem is NP-hard by a reduction from the DOMINATING SET problem. Although the definition of the STRONG GEODETIC SET problem has similarities with the definition of the GEODETIC SET problem, they have relevant different properties. At this dissertation we will use classical GEODETIC SET problem when referring to the latter problem to avoid disarray.

We observed the lack of parameterized algorithms for the SGS in the literature. These algorithms are interesting because they allow the resolution of NP-complete problems with a reduced time complexity when a parameter describing relevant instances of the problem is bounded. Moreover, the SGS is a natural graph problem that has important questions yet to be solved.

1.2 Related works

In Manuel et al. [2018], the authors first defined the STRONG GEODETIC SET problem and proved its NP-completeness by a reduction from the DOMINATING SET problem. This result inspired the use of the DOMINATING SET problem on our reductions. Moreover, the paper presents some important properties of the problem, a comparison between the classical geodetic problem and the strong version and determines minimum strong geodetic sets for Apollonian networks.

After that, the same research group studied the problem for grid-like architectures, Klavžar and Manuel [2018]. The authors established lower bounds and exact results to the strong geodetic number for thin cylinders and thin grids. It is also important to note that the strong geodetic number for general grids was not determined (using a closed formula), which is a fact that shows how hard is the problem.

In Iršič and Klavžar [2018], the authors focused on the SGS for more general Cartesian products of graphs. They achieved results concerning upper bounds on the strong geodetic number of Cartesian products of two general graphs and determined the strong geodetic number for Cartesian products of more simple graphs.

The problem for complete bipartite graphs was studied in Iršič [2018]. The authors analyzed the possibility of determining a closed formula for the strong geodetic number of complete bipartite graphs. A closed formula for $sgn(G)$ of balanced complete bipartite graphs was derived, but finding a closed formula to solve the SGN for general complete bipartite graphs is still an open problem. Moreover, the authors studied the asymptotic behavior of the SGN for complete bipartite graphs using an integer programming formulation and presented a quadratic-time algorithm to solve the problem. Finally, they proved the NP-completeness of the SGS restricted to (general) bipartite graphs and conjectured that the SGS for complete multipartite graphs is NP-complete.

Gledel et al. [2018] introduced the concept of strong geodetic cores, which is a subset X of a strong geodetic set that has the following property: it is possible to cover all vertices of the graph using only shortest paths that have at least one endpoint on X. For instance, a minimum strong geodetic set S of a tree T is its set of leaves, however, any leaf l is a geodetic core of S, because utilizing shortest paths between l and the remaining leaves all vertices of T are covered. Afterwards, some implications of this concept were given, including a better upper bound for the $sgn(G)$ of Cartesian products of graphs.

Finally, it is possible to see that several works have been done aiming to determine bounds and exact values of $sgn(G)$ even for very simple graph classes: complete bipartite graphs, complete multipartite graphs, thin cylinders, thin grids, Apollonian networks and general Cartesian product graphs. Nevertheless, the literature has few results concerning the computational complexity of the SGS for important graph classes, such as chordal graphs, split graphs, cographs, co-bipartite graphs, interval graphs and threshold graphs. In addition, there are important algorithmic related topics that deserve studies, such as: polynomial-time algorithms for the SGS restricted to important graph classes, fixed parameter tractable algorithms, approximation algorithms, heuristics and even exact exponential-time algorithms to solve the SGS.

1.3 Objectives

The main objectives of the present work are to investigate the time complexity of the SGS and the STRONG GEODETIC SET RECOGNITION problem (SGSR). The SGSR problem asks whether a given vertex set is a strong geodetic set, a more detailed definition will be given on Subsection 2.3.1. Moreover, we analyze the problems for different graph classes, aiming to understand the factors that make the problems polynomial-time solvable or NP-hard, assuming $P \neq NP$. An objective list follows:

1. To define the SGSR, study its properties, compare it to the SGS and find its time complexity.

2. To find the computational complexity of the SGS and SGSR for relevant graph classes.

3. To describe efficient polynomial-time algorithms for the problems restricted to graph classes that admit such algorithms.

4. To study the problems on a parameterized complexity point of view and analyze the influence of important parameters of the problem, such as graph's diameter and natural parameter (size of the strong geodetic set).

1.4 Organization of the dissertation

Chapter 2 contains general definitions on graphs, formal definitions of the SGS and SGSR and applications of the problems. Moreover, some important properties of the

SGSR are presented. The remainder of the chapter is dedicated to present concepts of parameterized complexity and graph classes.

Chapter 3 presents polynomial-time algorithms to solve the SGS and the SGSR restricted to certain graph classes. Each section contemplates a graph class and contains properties of the graph classes and the description of polynomial algorithms.

Chapter 4 contains NP-completeness proofs: the first section concerning the SGS for co-bipartite graphs, the second concerning the SGS for chordal graphs and the third concerning the SGSR for general graphs and bipartite graphs.

Chapter 5 contains the description of an exact exponential algorithm for the SGSR at general graphs. The last part of the chapter is dedicated to parameterized complexity results including an FPT algorithm.

Finally, Chapter 6 involves the discussion of the results and their implications. We present two graph classes hierarchy diagrams indicating the complexity status results we achieved for each graph class. Moreover, we propose several future works concerning the SGS and the SGSR.

Chapter 2

Definitions and theoretical references

2.1 General definitions on graph theory

In order to study the STRONG GEODETIC SET problem and other related problems we need to introduce some definitions, see Bondy et al. [1976] for basic concepts on graph theory. We say that a graph is *connected* if there exists a path connecting every vertex pair of G. In this work except when explicitly stated otherwise, we will only consider connected graphs, as a strong geodetic set of minimum cardinality of a disconnected graph is the union of a minimum strong geodetic set for each connected component of the graph. Thus, let $G = (V, E)$ be a simple, undirected and unweighted graph.

Define $D(u, v)$, the *distance* between the vertices u and v, as the number of edges on a shortest path between u and v. We will use u, v-shortest path to refer to any shortest path between u and v. If there is no path connecting u and v, then $D(u, v) = \infty$.

The *diameter* of a graph G is the greatest distance among all distances between pairs of vertices in G.

We say that $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$.

Let $U \subseteq V$. $G[U]$, the *subgraph of* G *induced by* $U \subseteq V$, is a subgraph of G with vertex set U that contains all the edges of G whose both endpoints belong to U.

Let $x \in V$, define $N(x)$ as the *open neighborhood* of x, that is, $N(x)$ is the set of vertices whose distance from x is exactly one.

Let $x \in V$, define $N[x]$ as the *closed neighborhood* of x, $N[x] = N(x) \cup \{x\}$.

We say that v is a *universal vertex* if $N[v] = V$.

A *clique* is a vertex set in which all vertex pairs are adjacent.

A *simplicial vertex* is one whose open neighborhood is a clique.

A *connected component* is a maximal connected induced subgraph of G.

A vertex v is a *cut-vertex* of G if $G - v$ has more connected components than G.

A *biconnected subgraph* is one that has no cut-vertices. A *biconnected component* is a maximal biconnected subgraph of G.

Let $u, v \in V$, $T(u, v)$, the *interval* between u and v, is the set of vertices that belong to some shortest path between u and v, including u and v themselves.

Let $S \subseteq V$, we define $T(S)$ as the union of intervals between each pair of vertices in S,

$$T(S) = \bigcup_{u,v \in S} T(u, v)$$

.

2.2 Strong geodetic set problem

Consider a graph $G = (V, E)$ and let $u, v \in V$, with $u \neq v$. We denote $P((u, v))$ as the set containing all shortest paths between u and v.

Let $S \subseteq V$ and let $U_S = \{(u, v)_1, (u, v)_2, \ldots, (u, v)_j\}$ be the set of distinct vertex pairs in S, observe that $j = \binom{|S|}{2}$ and $u \neq v$ for every vertex pair. We say that $I(S)$ is a shortest path assignment of S if:

$$I(S) = \left\{p_1, p_2, \ldots, p_{|U_S|} \mid p_i \in P((u, v)_i) \;\; \forall i \in \{1, 2, \ldots, |U_S|\}\right\}, \tag{2.1}$$

that is, $I(S)$ is a shortest path assignment for S if it contains, for each pair of distinct vertices (u, v) of S, a unique shortest path between u and v.

Now, we will use $V(p)$ to denote the set of vertices in a path p. A given vertex set S is a strong geodetic set of G if there exists a shortest path assignment $I(S)$ such that:

$$\bigcup_{p \in I(S)} V(p) = V. \tag{2.2}$$

Finally, the problem is to find a minimum cardinality strong geodetic set. The corresponding decision problem is: Given a graph $G = (V, E)$ and a positive integer k, is there a strong geodetic set $S \subseteq V$ with $|S| \leq k$?

2.3 A natural related problem

Studying the STRONG GEODETIC SET problem, we realized the need of verifying efficiently whether a given set $S \subseteq V$ is a strong geodetic set, returning a suitable shortest path assignment $I(S)$. This would allow us to implement a more efficient algorithm solving the STRONG GEODETIC SET problem.

At a first approach to solve the problem exactly, we developed an exponential algorithm that tests all possible shortest path choices between each pair of vertices of a given set S to decide whether any of these choices would cover all vertices (we say that a vertex of the graph is covered if any chosen shortest path contains it). Given that it is possible to compute the distance between all vertex pairs of the graph in time $\mathcal{O}\left((|E| + |V|) \times |V|\right)$, then $\mathcal{O}\left(\tau^{\binom{|S|}{2}} + (|E| + |V|) \times |V|\right)$ time is needed to decide whether S is a strong geodetic set, with τ being the largest number of different shortest paths between two vertices in S. It is worth mentioning that this number can be exponential on the size of the graph.

Given this first approach and the difficulty of obtaining a polynomial-time algorithm, we suspected that this decision problem would be NP-hard. This, at first, was unexpected, because this problem is more restrict than the STRONG GEODETIC SET problem. This analysis has encouraged us to define and study this problem.

2.3.1 Strong geodetic set recognition problem

The STRONG GEODETIC SET RECOGNITION problem (SGSR) is a decision problem which receives as input a graph $G = (V, E)$ and a vertex set $S \subseteq V$. The goal is to answer the following question: Is there a shortest path assignment $I(S)$, as defined in Equation 2.1, such that $\bigcup_{p \in I(S)} V(p) = V$?

Intuitively, we want to decide if, by assigning a shortest path for each pair of vertices in S, it is possible to cover all vertices of the graph using the chosen paths.

Figure 2.1 shows a graph G_1 and a strong geodetic set $S_1 = \{0, 1, 2\}$. The following assignment of shortest paths covers all vertices of the graph:

$$p_1 = \{0, 6, 1\}$$

$$p_2 = \{0, 5, 3, 2\}$$

$$p_3 = \{1, 4, 2\}$$

Figure 2.2 displays a graph G_2 and a strong geodetic set candidate $S_2 = \{0, 1, 2\}$. However, it is possible to conclude that S_2 cannot be a strong geodetic set, since 5 vertices must be covered beyond the vertices in S, and one can only use two paths of size 2 and a path of size 3, which can cover, at most, 4 different vertices.

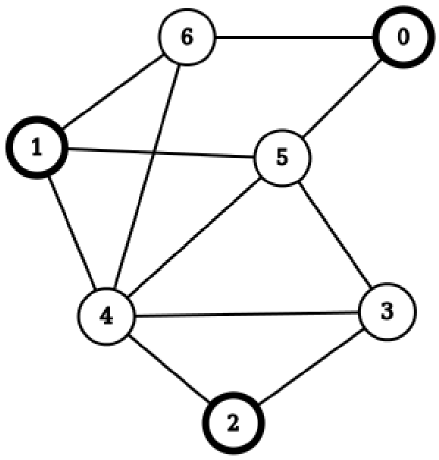


Figure 2.1: Graph G_1. The vertex set $\{0, 1, 2\}$ is a strong geodetic set.

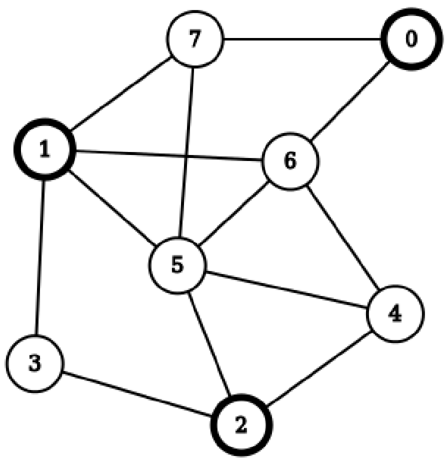


Figure 2.2: Graph G_2. The vertex set $\{0, 1, 2\}$ is not a strong geodetic set.

2.3.2 An application of the SGSR

A practical application of the problem is the following: in a certain city there is a set of police stations equipped with vehicles. In the city there are also several points that must be patrolled by some car throughout the day.

A graph is used to model the police stations, points and distances between them. The monitoring operation has a restriction: whenever a patrol leaves a police station it must travel along a shortest route to another police station in order to reduce costs. In addition, it must be assigned a fixed route between each pair of police stations, giving a permanent patrol schedule for the vehicles.

Given the locations of police stations and points, the problem is to decide whether it is possible to select a route for each pair of police stations (shortest path) so that all points are monitored by some route.

2.3.3 Properties of the SGSR

In this subsection we describe some essential properties of the problem.

Remark 1. Let S be a strong geodetic set of a graph G. It holds that all simplicial vertices of G belong to S.

Proof. Let S be a strong geodetic set of G. Assume, to the contrary, that x is a simplicial vertex and $x \notin S$. Since S is a strong geodetic set, some shortest path between a pair of vertices $y_1, y_k \in S$ must contain x.

Let $p = (y_1, y_2, \ldots, y_i, x, y_{i+1}, \ldots, y_k)$ be a shortest path containing x. Since $y_i, y_{i+1} \in N(x)$ and x is simplicial, then $p' = (y_1, y_2, \ldots, y_i, y_{i+1}, y_k)$ is a y_1, y_k-path shorter than p, contradicting the initial assumption. □

Remark 2. Let $G = (V, E)$ be a graph and consider $S \subseteq V$. If S is a strong geodetic set of G, then S is a (classical) geodetic set of G.

Remark 3. Let $G = (V, E)$ be a graph and $S \subseteq V$, if

$$|V| - |S| > \sum_{u,v \in S} \left(D(u, v) - 1 \right),$$

then S cannot be a strong geodetic set.

Proof. The greatest number of vertices among $V \setminus S$ that can be covered by shortest paths between vertex pairs in S is the sum of the distances between each pair of vertices minus one, since, a path of size k covers at most $k - 1$ vertices outside S. So, if there are more vertices to cover than the maximum number of vertices that can possibly be covered, then S cannot be a strong geodetic set. □

Theorem 2.3.1. *The* STRONG GEODETIC SET RECOGNITION *problem is polynomially reducible to the* STRONG GEODETIC SET *problem.*

Proof. Let P be an instance of the SGSR on the graph $G = (V, E)$ and $S \subseteq V$. We create an instance P' of the SGS on the graph $G' = (V', E')$ and the integer k, where $V' = V \cup \{v' \mid v \in S\}$, $E' = E \cup \{vv' \mid v \in S\}$ and $k = |S|$.

Let P be a **yes** instance, then S is a strong geodetic set for some choice of shortest paths $I(S)$. We state that G' has a strong geodetic set $S' = \{v' \mid v \in S\}$ with respect to $I(S')$. The set $I(S')$ is such that for each shortest path $p(u, v) \in I(S)$ it contains the path $p(u', v') = p(u, v) \cup \{u', v'\}$. Therefore, it holds that S' is a strong geodetic set of size k in G'.

Now, let P' be a **yes** instance, since the k added vertices $\{v' \mid v \in S\}$ to G are simplicial, then these vertices compose the SGS of P'. Consequently, P is a **yes** instance as well, since it is possible to make a shortest path choice $I(S)$ such that for each path in $I(S')$ assign the same path taking out the two endpoints of the path. Note that it is possible to construct an instance P' given P in time $\mathcal{O}(|V|)$. □

This result reinforce the intuition that the SGSR is computationally easier than the SGS. Observe that the presented reduction reveals a straightforward manner to solve the SGSR using a solution to the SGS. At certain cases this result will permit gathering polynomial algorithms to the SGSR based on polynomial algorithms to the SGS.

2.4 Fixed parameter tractability

Fixed-parameter tractable problems (FPT) are problems that can be solved by polynomial-time algorithms when a parameter of the problem is treated as a constant, Downey and Fellows [2012]. A parameter is any metric associated with the problem's instance, for example: diameter, treewidth and max-degree are parameters of graph problems.

Formally, an algorithm is FPT under a parameter bounded by k if its time complexity can be expressed as: $\mathcal{O}\left(f(k) \cdot n^c\right)$, with n being the size of the instance, c a constant and $f(k)$ a computable function on k. Fixed-parameter tractable algorithms are generally used to solve NP-Hard problems, and $f(k)$ is, in this case, an exponential function on k.

Algorithms of this type are interesting because the exponential part of the time complexity depends only on the parameter, and not on the size of the instance. In addition, if we have an FPT algorithm on a parameter k and we know that k is small on instances of interest, we will have an efficient algorithm.

It is also important to define the complexity class XP, Downey and Fellows [2012]. A problem parameterized by a k-bounded size parameter is in XP if its time complexity can be bounded by: $\mathcal{O}\left(f(k) \cdot n^{g(k)}\right)$, with n being the size of the instance and $f(k)$ and $g(k)$ being computable functions on k. Observe that XP problems are solvable in polynomial time when k is constant, just like FPT problems. However, XP problems are considered harder, as their exponential part of the complexity depends both on k and the size of the instance. Note that every FPT problem is also an XP problem.

We will investigate the possibility of designing FPT algorithms to the strong geodetic set problem. Alternatively, if it is not possible to implement such algorithms, we will try to provide evidences that the problem does not admit FPT algorithms.

2.5 Graph classes

A graph class is a set of graphs respecting a certain property. For example, the bipartite graph class is a set containing all bipartite graphs. Some important graph classes are: bipartite graphs, chordal graphs, interval graphs, trees and split graphs.

The study of graph classes is essential to graph theory and complexity theory, since in many situations there is interest in analyzing a problem when a certain graph property holds.

Graph classes have a crucial role in this study as long as we seek to investigate the time complexity of some graph problems. In order to do so we will try to set out whether solving a problem is polynomial or NP-hard for some graph class. Hopefully, this work will provide some guidance on recognizing hard instances of the problems and instances that can be solved efficiently (in polynomial time). The graph classes we study will be presented formally throughout the text.

An important observation is the following: if a problem is NP-hard for a given graph class C, then it will be NP-hard for all super-classes of C. Similarly, if a problem is polynomial-time solvable for a given graph class C, then it is polynomial-time solvable for all sub-classes of C.

Chapter 3

Graph classes admitting polynomial algorithms

This chapter contains polynomial algorithms to solve the SGS and SGSR for some important graph classes. We focus on the existence and description of the algorithms, as the main purpose of this work is recognizing hard and easy instances of the problems by analyzing its graph theoretical properties. Presenting polynomial-time algorithms for restricted instances of the SGSR is justified by the NP-completeness of the problem. This result is proved on Chapter 4, which contemplates NP-completeness results.

It is important to observe that some polynomial algorithms for the SGSR come directly from the existence of polynomial algorithms for the SGS at certain graph classes. These results hold because of the strong relation between the problems, which is elucidated in Theorem 2.3.1.

3.1 Block graphs

A block graph is one in which all biconnected components are complete subgraphs. Now, we introduce the definition of a cut-tree, which is an important structure to understand block graphs and the next result.

Definition 3.1.1 (Cut-tree)**.** A cut-tree $T = (V', E')$ of a graph G is a tree in which each vertex represents a biconnected component or a cut-vertex of G. There is an edge $e \in E'$ for each pair of a cut-vertex a and a biconnected component C of G, such that $a \in C$. Figure 3.1 illustrates a graph and its cut-tree.

Theorem 3.1.1. *Let $G = (V, E)$ be a block graph. The set S of simplicial vertices of G is the minimum strong geodetic set of the graph.*

Proof. By Remark 1 it holds that S must be contained in any strong geodetic set of G, so if we prove that S is a strong geodetic set, it has minimum cardinality. The vertices of the graph can be partitioned into two sets: simplicial vertices S and cut-vertices A. Let $T = (V', E')$ be a cut-tree of G and $v \in A$. Consider C_1 and C_2 as two connected components of $T[V' \setminus \{v\}]$. Let f_1 be a leaf of $T[C_1]$ and f_2 a leaf of $T[C_2]$. Note that both f_1 and f_2 represent biconnected components of G, which are complete graphs. As a result, each connected component denoted by f_1 and f_2 has at least one simplicial vertex: s_1 and s_2, respectively. Finally, the s_1, s_2-shortest path contains v, whereas it is a cut-vertex. Thereafter, S is a minimum strong geodetic set of G. □

Corollary 3.1.2. *There is a linear-time algorithm that solves the* STRONG GEODETIC SET *problem for block graphs.*

Proof. The algorithm consists in running a depth first search to find the set A of cut-vertices of the graph. Then, return $V \setminus A$ as solution. □

Corollary 3.1.3. *There is a linear-time algorithm that solves the* STRONG GEODETIC SET RECOGNITION *problem for block graphs.*

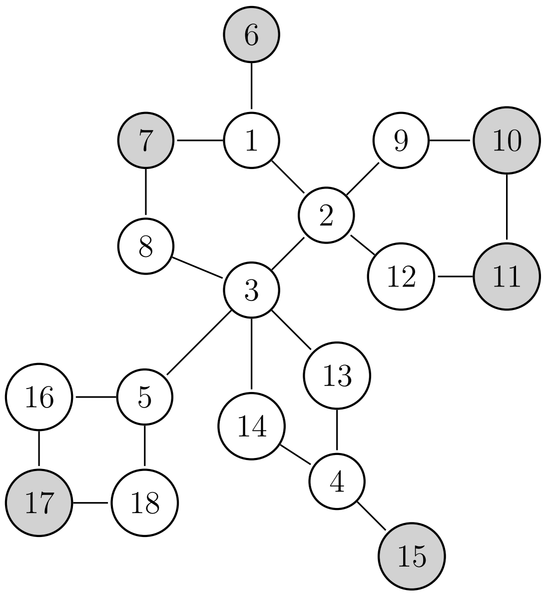
Proof. Given any set $X \subseteq V$, if $S \subseteq X$ then X is a strong geodetic set, otherwise X is not a strong geodetic set. □

3.2 Cacti graphs

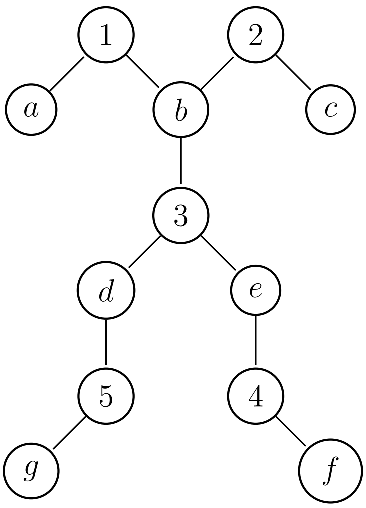
A cactus graph is one in which every edge belongs to at most one cycle. An alternate definition is that all biconnected components are cycles or edges.

3.2.1 A polynomial-time algorithm

We will first illustrate a pre-processing procedure. The procedure receives a cactus graph and its cut-tree. We will consider that the received cut-tree has at least two nodes, as otherwise the algorithm simple consists in solving the SGS for a cycle or an edge.



(a) A cactus graph G. The vertices marked in gray constitute a minimum strong geodetic set of G that would result from the execution of the algorithm presented at Subsection 3.2.1.



(b) A cut-tree T of G. Vertices labelled with letters represent biconnected components of G and those labelled with numbers represent cut-vertices of G. Biconnected components: $a = \{6, 1\}$, $b = \{1, 2, 3, 7, 8\}$, $c = \{2, 9, 10, 11, 12\}$, $d = \{3, 5\}$, $e = \{3, 4, 13, 14\}$, $f = \{4, 15\}$, $g = \{5, 16, 17, 18\}$.

Figure 3.1: A cactus graph and its cut-tree.

1. **Input:** A cactus graph $G = (V, E)$ and its cut-tree $T = (V', E')$.
2. Initialize S as an empty set.
3. For each leaf f in T do:

   - If f corresponds to an edge uv of G (a biconnected component that is an edge), then add its simplicial vertex to S.

   - If f corresponds to an even cycle C of length l whose cut-vertex is a, add a vertex $v \in C$ to S such that $D(a, v) = \dfrac{l}{2}$.

   - If f corresponds to an odd cycle C of length l whose cut-vertex is a, add two vertices $u, v \in C$ to S, such that $D(a, u) = D(a, v) = \left\lfloor \dfrac{l}{2} \right\rfloor$.
4. Finish pre-processing.

Having finished pre-processing, we now define how to process each biconnected component (block) associated to internal vertices of T. Let t be an internal vertex of T, if t represents an odd cycle C of length l do: Define A as the set of cut-vertices of G present in C. Consider $x_1, x_k \in A$ with $x_1 \neq x_k$ and $P = (x_1, x_2, \ldots, x_{k-1}, x_k)$ as the longest path between x_1 and x_k in C. Let $j = \left\lfloor \dfrac{1+k}{2} \right\rfloor$ and $v = x_j$. If $\bigcup_{p,q \in A} p(p, q) \neq V(C)$ add v to S, otherwise, proceed to the next block. Here, $p(p, q)$ denotes the unique shortest path between p and q in C.

If t represents an even cycle C of length l in G do: Define A as the set of cut-vertices of G contained in C. If there are $a_1, a_2 \in A$ such that $D(a_1, a_2) = \frac{l}{2}$ proceed to the next block, otherwise, if $\bigcup_{p,q \in A} p(p, q) \neq V(C)$ add a vertex to S the same way as described for odd cycles at the previous paragraph.

After processing all blocks, if $|S| \geq 3$, then S is a minimum strong geodetic set of G and the algorithm finishes. Otherwise, verify whether G contains any block that is an even cycle, if so, add an arbitrary vertex of G to S and finish. Otherwise, return S and finish.

We can use Figure 3.1 to illustrate the execution of the algorithm. At first, for biconnected components situated at leaves of T (a, c, f, g), the algorithm adds vertices to S as explained at the pre-processing procedure. Now, for block b, the algorithm verifies that the shortest paths between its cut vertices do not suffices to cover all vertices in b, then it adds the vertex 7 to S. For block d, no vertex is added to S, as d represents an edge of G. For block e, no vertex is added as well, because e represents an even cycle that has a pair of cut-vertices whose distance is equal half the size of the cycle. Finally, all blocks have been processed and S is returned.

Theorem 3.2.1. *The algorithm presented above is correct.*

Proof. For now consider that the algorithm receives as input a cactus graph $G = (V, E)$ whose cut-tree $T = (V', E')$ contains at least 3 leaves. We will first show that the returned set S is a strong geodetic set. From the description of the algorithm we know that we will have at least one vertex in S for each leaf of T. Let F be the set of leaves of T and $f_1 \in F$ a leaf that represents an edge $e = ux$ in G whose simplicial vertex is u. And let f_2 be another leaf of T, with $v \in S \cap V(F_2)$, finally note that any path between u and v contains x, covering all vertices of e.

Now let f_1 be a leaf of T that represents an even cycle C of length l. By the algorithm, we add to S a vertex v whose distance to the cycle's cut-vertex a is $\frac{l}{2}$, thus, we have two distinct paths between v and a with length $\frac{l}{2}$: c_1 and c_2. Let f_2 and f_3 be two other leaves of T, that exist by hypothesis. Any shortest path that goes from v to the cited leaves contains a, so we set the shortest path between f_1 and f_2 to pass through c_1 and the shortest path between f_1 and f_3 to pass through c_2, covering all vertices of C.

Now let f_1 be a leaf of T representing an odd cycle C of length l. By the algorithm, we add two vertices to S: v_1 and v_2, such that their distances to the cycle's cut-vertex a are the same: $\left\lfloor \frac{l}{2} \right\rfloor$. Observe that: $p(v_1, a) \cup p(v_2, a) = V(C)$, thus, by choosing any shortest path from v_1 to another vertex $v_3 \in S \cap f_3$, where f_3 is another leaf of T, and from v_2 to the same leaf f_3 all vertices of C will be covered.

Let $t \in T$ be an internal vertex of T that represents an edge $e = uv$ of G. Let C_1 and C_2 be connected components of $T - \{t\}$. In addition, consider f_1 to be a leaf of C_1 and f_2 a leaf of C_2, now note that any path between $x \in S \cap V(f_1)$ and $y \in S \cap V(f_2)$ contains u and v. Therefore, all vertices of e will be covered.

Let $t \in T$ be an internal vertex of T which represents a cycle C of size l at G. Let A denote the set of cut-vertices of C, the algorithm verifies whether $\bigcup_{p,q \in A} p(p, q) = V(C)$, we claim that if that holds, then all vertices of C are covered by shortest paths between vertices in S. In fact, let v be any vertex of C, assuming $\bigcup_{p,q \in A} p(p, q) = V(C)$, there are vertices $a_1, a_2 \in A$ such that $v \in p(a_1, a_2)$. Now, let C_1 and C_2 be the two connected components of $G - \{a_1\}$ such that C_1 is the one that has no vertex of C. Analogously, let C_3 and C_4 be the two connected components of $G - \{a_2\}$ such that C_3 is the one that has no vertex in C. Let $x \in C_1 \cap S$ and $y \in C_3 \cap S$, these vertices exist because the algorithm guarantees that every leaf of T has a vertex in S, observe that any shortest path between x and y contains v. Nevertheless, if $\bigcup_{p,q \in A} p(p, q) \neq V(C)$ and there are no vertices $i, j \in A$ such that $D(i, j) = \frac{l}{2}$, then the algorithm adds a vertex $v \in V(C)$ to S so that there exists vertices $a_1, a_2 \in A$ such that $D(a_1, v) - D(a_2, v) \leq 1$. Thus, it holds that $p(a_1, a_2) \cup p(a_1, v) \cup p(a_2, v) = V(C)$, having all vertices of C covered. Finally, if there are vertices $i, j \in A$ such that $D(i, j) = \frac{l}{2}$, then it is possible to cover all vertices of C, since T has at least 3 leaves.

Now, it remains to argue that the returned set S is minimum. Observe that odd cycles situated at leafs of T must have at least 2 of its vertices in S and even cycles situated at leafs of T must have at least 1 of its vertices in S. Edges located at leaves of T must have its simplicial vertex added to S. Now, observe that for internal vertices of T we add to S the minimum amount of vertices needed, that is, for edges we add none, for cycles that can be covered by shortest paths between its cut-vertices we add none, and for cycles that cannot be covered that way we add a vertex to S, which is the minimum required. □

Corollary 3.2.2. *There is a polynomial-time algorithm that solves the* STRONG GEODETIC SET RECOGNITION *problem for cacti graphs.*

Proof. In order to verify that a given vertex set X of a cactus graph $G = (V, E)$ is a strong geodetic set, we utilize the reduction presented in Theorem 2.3.1. If the reduction is applied to G, then a cactus graph $G' = (V', E')$ arises, this occurs because the reduction only adds one-degree vertices to the graph. Thus, it is possible to solve the SGSR for cacti graphs by solving the SGS at a related cactus. Finally, since it is possible to solve the SGS for cacti graphs in polynomial time, then the SGSR for cacti graphs is also computable in polynomial time. □

3.3 Split graphs and threshold graphs

Split graphs are graphs whose vertex set can be partitioned into a clique and an independent set. Threshold graphs are graphs that can be constructed from the one-vertex graph, including itself, by using the following operations:

1. Add an isolated vertex to the graph.

2. Add a universal vertex to the graph.

Observe also that every threshold graph is a split graph. Moreover, threshold graphs have a characterization by forbidden induced subgraphs: a threshold graph is one that has no $2K_2$, P_4 or C_4 as induced subgraphs, Mahadev and Peled [1995].

3.3.1 The classical geodetic set problem

Recall that on the classical version of the problem every shortest path between a pair of vertices on a geodetic set is used to cover the graph's vertices. Here, $g(G)$ denotes the size of a minimum (classical) geodetic set of G.

Theorem 3.3.1 (Dourado et al. [2010]). *Let G be a connected split graph. Let $V_1 \cup V_2$ be a partition of $V(G)$ such that V_1 is a maximal independent set and V_2 is a clique. Let S denote the set of simplicial vertices of G. Let U denote the set of vertices $u \in V_2 \setminus S$ which have exactly one neighbour in V_1, say u', $V_2 \cap S \subseteq N_G(u')$ and $d_G(u', w) = 2$ for all $w \in V_1 \setminus \{u'\}$.*

1. *If $U = \emptyset$, then $g(G) = |S|$.*
2. *If $U \neq \emptyset$ and there is a vertex $v \in V_2 \setminus S$ such that*

$$(N_G(v) \cap V_1) \cap \left(\bigcup_{u \in U \setminus \{v\}} (N_G(u) \cap V_1) \right) = \emptyset,$$

   then $g(G) = |S| + 1$.

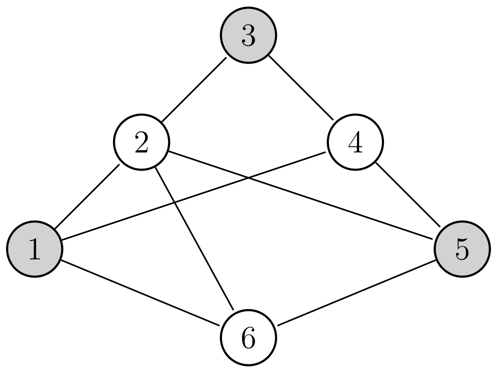
3. *If $U \neq \emptyset$ and there is no vertex $v \in V_2 \setminus S$ as specified in item 2, then $g(G) = |S| + 2$.*

Lemma 3.3.2. *Let $G = (V, E)$ be a connected threshold graph and let $V_1 \cup V_2$ be a partition of V such that V_1 is a maximal independent set and V_2 is a clique. The set U, as defined on the previous theorem, is empty on G.*

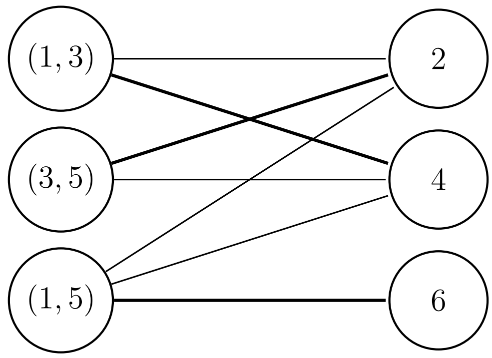
Proof. Assume, to the contrary, that G has a vertex $u \in U$ and let $u' \in N(u) \cap V_1$. Since u is not simplicial, there is at least a vertex $x \in V_2$ that is not adjacent to u'. By the maximality of V_1, x must have at least a neighbor $y \in V_1$, observe that $y \neq u'$. Now, note that the induced subgraph $G[\{u', u, x, y\}]$ is a P_4 (a path with 4 vertices). This happens because the edges $u'u$, ux and xy belong to E and the edges $u'y$, $u'x$ and uy do not. This is a contradiction because threshold graphs do not have P_4 as induced subgraphs. □

Corollary 3.3.3. *Let S be the set of simplicial vertices of a connected threshold graph $G = (V, E)$. S is the minimum classical* GEODETIC SET *of G.*

Proof. Theorem 3.3.1 proves that the geodetic number of connected split graphs is $|S|$ whenever U is empty. Then, considering Lemma 3.3.2, S is a minimum classical GEODETIC SET of G. □



(a) A graph G having diameter 2. We will verify whether $\{1, 3, 5\}$ is a strong geodetic set of G.



(b) The auxiliary bipartite graph H arising from G. The thicker edges represents a size 3 matching, which shows that $\{1, 3, 5\}$ is as strong geodetic set of the graph G.

Figure 3.2: Figures illustrating the algorithm presented at Theorem 3.3.4.

3.3.2 Polynomial-time algorithms for the SGSR

Here we present polynomial-time algorithms to the SGSR restricted to graphs of diameter 2 and to split graphs. The algorithms will be useful as an intermediate step to solving the SGS for threshold graphs.

Theorem 3.3.4. *Let $G = (V, E)$ be a connected graph of diameter 2 and consider $S \subseteq V$. There exists an $\mathcal{O}(|S|^2 \cdot |V \setminus S|)$-time algorithm that decides whether S is a strong geodetic set of G (*STRONG GEODETIC SET RECOGNITION *problem).*

Proof. At first, we construct an auxiliary bipartite graph $H = (A, B, E')$, with $A = \{v_{i,j} \mid i, j \in S \wedge i \neq j\}$ and $B = V \setminus S$, note that we use A and B as vertex set partitions (independent sets) of H. In addition, there is an edge between $v_{i,j} \in A$ and $y \in B$ if and only if (i, y, j) is an i,j-shortest path at G. This construction is illustrated at Figure 3.2.

Now, we calculate a maximum matching M of H. This can be done in time $\mathcal{O}(|E'|)$, Alom et al. [2010]. Observe that $|A| \leq |S|^2$ and $|B| = |V \setminus S|$, then it is possible to compute such a matching in time $\mathcal{O}(|S|^2 \cdot |V \setminus S|)$. Finally, if $|M| = |B|$ output YES, otherwise, output NO.

In order to prove the correctness of the algorithm we prove that the maximum matching M of H has size $|B|$ if and only if S is a strong geodetic set of G. Assume that $|M| = |B|$, then for each vertex b in B there is an edge $v_{i,j}b \in M$ and we use the (i, b, j) shortest path to cover b. Moreover, since M is a matching, for each pair of vertices $i, j \in S$ it will be assigned a unique i,j-shortest path in $I(S)$. Finally, if there are still shortest paths to be assigned in $I(S)$, any choice of shortest paths will guarantee a valid strong geodetic set S.

For the converse, assume that S is a strong geodetic set of G, then there is a shortest path choice $I(S)$ that covers all vertices in $V \setminus S$. Let $u \in V \setminus S$ and let M be an empty set. It holds that at least one p, q-shortest path in $I(S)$ covers u, we add the edge $v_{p,q}u$ to M, observe that $v_{p,q}u \in E'$, by the definition of H. Repeat this process for every $u \in V \setminus S$. It results that M is a maximum matching of H, with $|M| = |B|$. In fact, note that M has exactly one edge incident to each vertex in B and at most one edge in M is incident to a vertex in A, given that there is a unique shortest path in $I(S)$ for each vertex pair of S. □

Theorem 3.3.5. *Let $G = (V, E)$ be a connected split graph and consider $S \subseteq V$. There exists an $\mathcal{O}(|S|^2 \cdot |V \setminus S|)$-time algorithm that decides whether S is a strong geodetic set of G (STRONG GEODETIC SET RECOGNITION problem).*

Proof. We propose a construction that follows the same approach of Theorem 3.3.4. Create an auxiliary bipartite graph $H = (A, B, E')$, with $B = V \setminus S$, now it remains to define A and E': for each pair (i, j) of vertices, with $i, j \in S$ and $i \neq j$ do:

- If $D(i, j) \neq 3$, add a vertex $v_{i,j}$ to A. In addition, add the edges $v_{i,j}k$ for all $k \in B$ such that (i, k, j) is a shortest path in G.
- If $D(i, j) = 3$, add the vertices $v_{i,j}$ and $\overline{v_{i,j}}$ to A. Then, add the edges $v_{i,j}k$ for all $k \in N(i) \cap B$, and add the edges $\overline{v_{i,j}}k'$ for all $k' \in N(j) \cap B$.

Now, we compute a maximum matching M of H in time $\mathcal{O}(|S|^2 \cdot |V \setminus S|)$, the time complexity is derived similarly as in Theorem 3.3.4. Finally, if $|M| = |B|$ output YES, otherwise, output NO.

In order to prove the correctness of the algorithm we prove that the maximum matching M of H has size $|B|$ if and only if S is a strong geodetic set of G. Assume that $|M| = |B|$, then, for each vertex $b \in B$ there is an edge $ab \in M$, with $a \in A$. If $a = v_{i,j}$, with $D(i, j) = 2$, then assign the (i, b, j) shortest path to $I(S)$. On the other hand, if $a = v_{i,j}$ (without loss of generality), and $D(i, j) = 3$, we set b to be on the i, j-shortest path, and the other vertex present on the i, j-shortest path will be the vertex in B that is an endpoint of the edge matching $\overline{v_{i,j}}$. Finally, since M is a matching, for each pair of vertices $i, j \in S$ it will be assigned a unique i,j-shortest path in $I(S)$. Therefore, $I(S)$ defines a strong geodetic set S.

For the converse, assume that S is a strong geodetic set for G defined by $I(S)$. Let M be an empty set. Then, for each i,j-shortest path (i, k, j) of size 2 in $I(S)$, with $k \in B$, add the edge $v_{i,j}k$ in M. And for each i,j-shortest path (i, k, l, j) of size 3 in $I(S)$, add the edges $v_{i,j}k$, if $k \in B$, and $\overline{v_{i,j}}l$, if $l \in B$. Now, remove edges of M until there is exactly one edge in M incident to each vertex in B. Finally, observe that M is a maximum matching of H, with $|M| = |B|$. □

3.3.3 Strong geodetic set problem for threshold graphs

Definition 3.3.1 (Nested neighborhood ordering)**.** Let $G = (V, E)$ be a graph and $X \subseteq V$ with $|X| = m$. A nested neighborhood ordering of X is a sequence $\alpha = (c_1, c_2, \ldots, c_m)$ of the elements of X such that:

$$N(c_i) \subseteq N(c_{i+1}), \forall i \in \{1, 2, \ldots, m-1\}$$

Definition 3.3.2 (Reverse nested neighborhood ordering)**.** Let $G = (V, E)$ be a graph and $X \subseteq V$ with $|X| = m$. A reverse nested neighborhood ordering of X is a sequence $\beta = (c_1, c_2, \ldots, c_m)$ of the elements of X such that:

$$N(c_i) \supseteq N(c_{i+1}), \forall i \in \{1, 2, \ldots, m-1\}$$

On the next definition $\mathcal{P}(V)$ refers to the power set of V, that is, $\mathcal{P}(V)$ is a set containing all subsets of V.

Definition 3.3.3 (Vertex covering)**.** Let $G = (V, E)$ be a graph and $S \subseteq V$. Define the function U: $\mathcal{P}(V) \Rightarrow \mathcal{P}(V)$. We say that $U(S)$ is a vertex covering of S in G if there is a shortest path choice $I(S)$ such that:

$$\bigcup_{p \in I(S)} p = U(S)$$

.

Now, we look into solving the STRONG GEODETIC SET problem for threshold graphs in polynomial time. In order to prove the correctness of the proposed algorithm we first prove some lemmas. Let $G = (V, E)$ be a threshold graph. The vertices in V are partitioned into a clique C and an independent set I. It is known that threshold graphs have the property that its vertices belonging to the independent set or to the clique can be nested neighborhood ordered, Mahadev and Peled [1995]. Let $\alpha = (c_1, c_2, \dots, c_m)$ be a nested neighborhood ordering of C and $\beta = (i_1, i_2, \dots, i_n)$ be a reverse nested neighborhood ordering of I.

Lemma 3.3.6. *Let a and b be vertices in C, with $N[a] \neq N[b]$, such that a precedes b in the nested neighborhood ordering α of C. Consider $S \subseteq V$ and let $I(S \cup \{b\})$ be any selection of shortest paths, there is $I(S \cup \{a\})$ with:*

$$V(I(S \cup \{b\})) \subseteq V(I(S \cup \{a\}))$$

.

Proof. We will construct the set $I(S \cup \{a\})$. We add to $I(S \cup \{a\})$ the same i,j-shortest paths in $I(S \cup \{b\})$, with $i \neq j \neq b$. Now, observe that the vertices in $V(I(S \cup \{b\})) \setminus V(I(S \cup \{a\}))$ must be covered by shortest paths between b and vertices in S. Since $N[a] \neq N[b]$ and a precedes b in α, we have that b has at least a neighbor x that is not neighbor of a. We add to $I(S \cup \{a\})$ the (a, b, x) shortest path. Then, for each (b, u, y) shortest path in $I(S \cup \{b\})$ we add (a, u, y) to $I(S \cup \{a\})$. Finally, we will have that $V(I(S \cup \{b\})) \subseteq V(I(S \cup \{a\}))$.

□

Lemma 3.3.7. *Let $G = (V, E)$ be a threshold graph and let $Q = \{c_1, c_2, ..., c_{|Q|}\}$, such that $\alpha = (c_1, c_2, ..., c_{|Q|})$ is a nested neighborhood ordering of Q. If there is a strong geodetic set of size k in G, then $O = I \cup Q$ is a strong geodetic set, with $|I| + |Q| = k$.*

Proof. Let $O' = I \cup Q'$ be a k-sized strong geodetic set of G. If $O' \neq O$, then there is a vertex $u \in Q'$ so that $u \notin Q$, and because $|O| = |O'| = k$, there exists a vertex $v \in Q$ so that $v \notin Q'$. Note that the vertices in Q are α-ordered, then v appears before u in the α-ordering. If $N[u] \neq N[v]$, Lemma 3.3.6 assures that if we replace u by v in O', then the obtained set is a strong geodetic set. If $N[u] = N[v]$, then shortest paths arising from u or v can cover the same vertices, except u and v. However, O' is a strong geodetic set with $v \notin O'$, hence, there is an a, b-shortest path that covers v, with $a, b \in O'$. So by replacing u by v, we now set the a, b-geodesic to cover u, this will be possible because u and v are twin vertices. Consequently, the same vertices will be covered after the swap of u by v. Finally, note that we can repeat this process of exchanging a vertex in O' by a vertex in O until we transform O' into O, assuring that O is a strong geodetic set of size k. $\square$

Theorem 3.3.8. *Let $G = (V, E)$ be a threshold graph. There is an algorithm that solves the* STRONG GEODETIC SET *problem for G in time $\mathcal{O}(|V|^3)$.*

Proof. It follows a description of the algorithm. Consider that G is partitioned into a clique C and an independent set I. Let α be a nested neighborhood ordering of C. Observe that it is possible to recognize a threshold graph and output its α-ordering in linear time on the size of the graph, Heggernes and Kratsch [2007].

1. Let U be an empty set. Add all simplicial vertices of G to U.

2. Run the algorithm explained in Lemma 3.3.4 to decide if U is a strong geodetic set. If so, return U and finish.

3. Let $v \in C$ be the first vertex in the nested neighborhood ordering of C that is not in U. Add v to U. Go to the second step.

The correctness of the algorithm follows from Lemma 3.3.7. Note that the algorithm constructs a minimum strong geodetic set greedily. $\square$

Chapter 4

NP-Completeness results

4.1 Strong geodetic set problem

This section is dedicated to NP-completeness proofs obtained for the STRONG GEODETIC SET problem. During the study of the SGS we noticed the hardness of the problem even for very restricted graph classes. Here we present proofs based on reductions from the dominating set, a problem that has been commonly used to prove hardness results for either the classical GEODETIC SET problem and the STRONG GEODETIC SET problem.

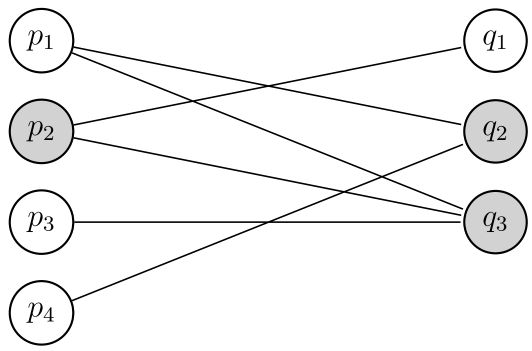
4.1.1 Co-bipartite graphs

A co-bipartite graph is the complement of a bipartite graph. Alternatively, a graph is said to be co-bipartite if its vertex set can be partitioned into two cliques. Note that the maximum diameter of a connected co-bipartite graph is 3.

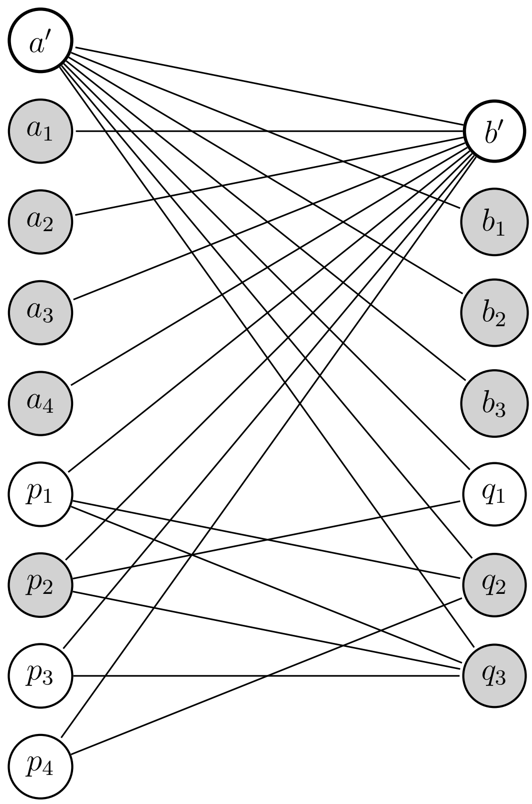
Now we introduce the DOMINATING SET problem, an NP-complete problem. Let $G = (V, E)$ be a graph, we say that $D \subseteq V$ is a dominating set of G if for every vertex $v \in V$ it holds that: $v \in D$ or v is adjacent to a vertex in D. The decision version of the problem is: Is there a dominating set D of G such that $|D| \leq k$?

Theorem 4.1.1. *The* STRONG GEODETIC SET *problem restricted to co-bipartite graphs is NP-Complete.*

Proof. Let $G = (V, E)$ be a co-bipartite graph. The present problem is in NP, because given a set $S \subseteq V$ and a set of shortest paths choices $I(S)$, it is easy to verify whether the chosen paths cover all vertices in V and if $|S| \leq k$.



(a) A connected bipartite graph G. The vertices marked in gray constitute a dominating set of G.



(b) A graph H arising from G as indicated on Theorem 4.1.1. Consider that both the vertices on the left side and on the right side induce cliques, the edges of the cliques were omitted for clarity sake. The vertices marked in gray compose the strong geodetic set S.

Figure 4.1: Figures illustrating the polynomial reduction presented on Theorem 4.1.1

Now, in order to prove that SGS for co-bipartite graphs is NP-Hard, a polynomial reduction inspired by Ekim and Erey [2014] is presented. That is, we reduce the dominating set problem for connected bipartite graphs to the SGS for co-bipartite graphs.

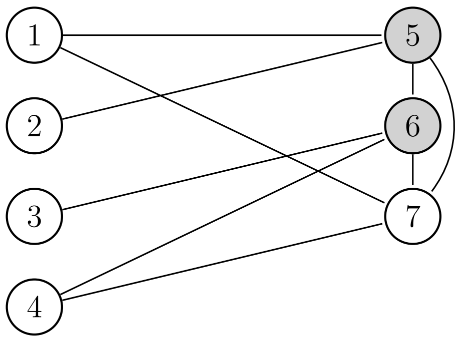
Figure 4.1 contains an example of the reduction presented here. Let $G = (V, E)$ be a connected bipartite graph with parts A and B, having sizes greater than or equal to 2, with $A = \{p_1, p_2, \ldots, p_{|A|}\}$ and $B = \{q_1, q_2, \ldots, q_{|B|}\}$. Then, construct the graph $H = (V', E')$, with: $V' = V \cup \overline{A} \cup \overline{B} \cup \{a', b'\}$, such that $\overline{A} = \{a_1, a_2, \ldots, a_{|A|}\}$ and $\overline{B} = \{b_1, b_2, \ldots, b_{|B|}\}$, note that $|A| = |\overline{A}|$ and $|B| = |\overline{B}|$. The edge set E' contains all edges in E and the necessary additions such that a' and b' are universal vertices, $A \cup \overline{A} \cup \{a'\}$ is a clique and $B \cup \overline{B} \cup \{b'\}$ is a clique as well. Observe that H is a co-bipartite graph.

Let D be a dominating set of G, with $|D| = k$, we will show that H has a strong geodetic set $S = D \cup \overline{A} \cup \overline{B}$. We will construct a suitable $I(S)$ that covers all vertices. The shortest path (b_1, a', a_1) is assigned to cover a'. The shortest path (b_2, b', a_1) is assigned to cover b'. For any vertex $p_i \in A \setminus S$, it holds that p_i has at least a neighbor $u \in B \cap S$, then the shortest path (u, p_i, a_i) is assigned to cover p_i. Finally, for any vertex $q_i \in B \setminus E$, it holds that q_i has at least a neighbor $v \in A \cap S$, so we assign the (v, q_i, b_i) shortest path to cover q_i. Concluding, S is a strong geodetic set of H, with $|S| = |D| + |V|$.

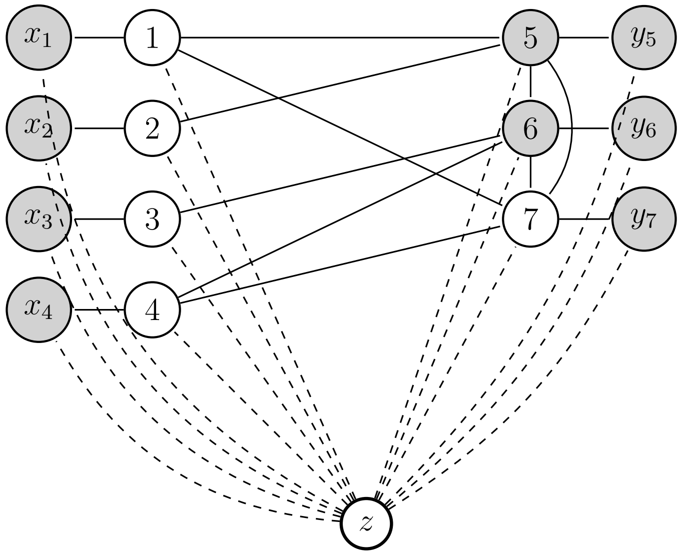
Now, it remains to prove that if S is a strong geodetic set of H, with $|S| \leq k + |\overline{A} \cup \overline{B}|$, then G has a dominating set D with $|D| \leq k$. Note that if S is a SGS, then $\overline{A} \cup \overline{B} \subseteq S$, as $\overline{A}$ and $\overline{B}$ contain only simplicial vertices. Now, we show that $S \cap V$ is a dominating set of G. Since S is a strong geodetic set of H, for each vertex $x \in A \setminus S$ there is a shortest path in $I(S)$ that contains x. Note that $D(u, v) \leq 2$ for all $u, v \in V'$, so there exists a shortest path (a, x, b) in H such that $a, b \in S$. Recall that one of the vertices at the shortest path must be in B, and denote it b. This holds because x has no neighbors in $\overline{B}$ and b' cannot be in a shortest path, because b' is a universal vertex. Concluding, every vertex $x \in A \setminus S$ has a neighbor in B that belongs to S, similarly, the same holds for any vertex $x' \in B \setminus S$. Thereafter, $S \cap V$ is a dominating set of G, with $|S \cap V| \leq k$, since $|S \cap (V' \setminus V)| \geq |V|$. □

4.1.2 Chordal graphs

A graph is said to be chordal if it has no induced cycles of length 4 or more. In this section we present a reduction from the dominating set problem for split graphs to the strong geodetic set problem for chordal graphs. Note that solving the



(a) The figure displays a split graph G with independent set $I = \{1, 2, 3, 4\}$ and clique $C = \{5, 6, 7\}$. The set of vertices marked in gray consists in a dominating set of G.



(b) The figure depicts a chordal graph H that arises from G (Figure 4.2a). The edges incident to z are dashed for clarity sake. Note that z is a universal vertex. The set of vertices marked in gray consists in a strong geodetic set.

Figure 4.2: Figures illustrating the polynomial reduction presented on Theorem 4.1.2

DOMINATING SET problem for split graphs is NP-complete, Bertossi [1984]. Hence, we show that computing a minimum strong geodetic set for chordal graphs is NP-complete.

Theorem 4.1.2. *The* STRONG GEODETIC SET *problem for chordal graphs is NP-complete.*

Proof. Let $G = (V, E)$ be a connected split graph with its vertex set partitioned into C and I, with C a clique and I an independent set. And let the graph H be obtained from G as follows: for each vertex $u \in I$ add the vertex x_u to H, and, for each vertex $v \in C$ add the vertex y_v to H, and finally, add a universal vertex z to H. Besides that, for each vertex $u \in I$ add an edge between u and x_u, and, for each vertex $v \in C$ add an edge between v and y_v. Observe that the diameter of H is 2.

Figure 4.2 displays an example of the construction of H.

Assume that G has a dominating set D, with $|D| \leq k$. Let $X = \{x_u \mid u \in I\}$ and $Y = \{y_u \mid u \in C\}$. We show that $S = D \cup X \cup Y$ is a strong geodetic set in H. First, note that any y, y'-shortest path contains z, with $y, y' \in Y$ and $y \neq y'$, then we include the shortest path (y, z, y') in $I(S)$. Now, let u be a vertex in $I \setminus S$, which implies that $u \notin D$, and then u has a neighbor $v \in C \cap D$, that is, $v \in S$. We include the x_u, v-shortest path that contains u in $I(S)$. Let p be a vertex in $C \setminus S$, analogously, p has a neighbor $q \in I \cap S$. We include the y_u, q-shortest path that contains p in $I(S)$. Thus, S is a strong geodetic set in H, with $|S| \leq k + |V|$.

For the converse, assume that H has a strong geodetic set S with $|S| \leq k + |V|$. First, observe that the vertices in $X \cup Y$ are simplicials, so $X \cup Y \subseteq S$. Now, note that if some strong geodetic set S of H contains z, $S \setminus \{z\}$ is a strong geodetic set too. This happens because any y, y'-shortest path contains z, with $y, y' \in Y$ and $y \neq y'$. Hence, we will assume that $z \notin S$.

We now prove that $D = S \cap V$ is a dominating set of G. Let $u \in V$ be a vertex not in D. So there exists an m, n-shortest path in H that contains u. As the diameter of H is 2, this path must be in the form (m,u,n). Suppose, for contradiction, that neither m or n are in V. So there are three cases for m and n: m and n are in X, $m \in X$ and $n \in Y$ and m and n are in Y. In all cases there is only one possible shortest path: (m, z, n). Therefore, m or n must be in $V \cap S$, so u has a neighbor in D. Concluding, D is a dominating set of G, and $|D| \leq k$, because $|(X \cup Y) \cap S| = |V|$.

Finally, observe that it is possible to construct H in linear time on the size of G. Therefore, we have presented a polynomial reduction. Now, it remains to prove that H is chordal. Let α be a cycle of H having size at least 4. If α contains z it is easy to see that the cycle contains a chord, because z is universal. If α contains a vertex $w \in X \cup Y$ it must contain z as well, because w is a 2-degree vertex adjacent to z. So, it remains to consider cycles that have only vertices in V, but we know that G is a split graph and, hence, a chordal graph. Therefore, H is a chordal graph.

□

4.2 Strong geodetic set recognition problem

In order to prove the NP-completeness of the STRONG GEODETIC SET RECOGNITION problem we first introduce a variant of the satisfiability problem: $3-SAT_3$, an NP-complete problem, Schaefer [1978]. An instance of $3-SAT_3$ is defined by a set $U = \{x_1, x_2, \ldots, x_n\}$ of variables and a set $C = \{c_1, c_2, \ldots, c_m\}$ of clauses. Each clause is a

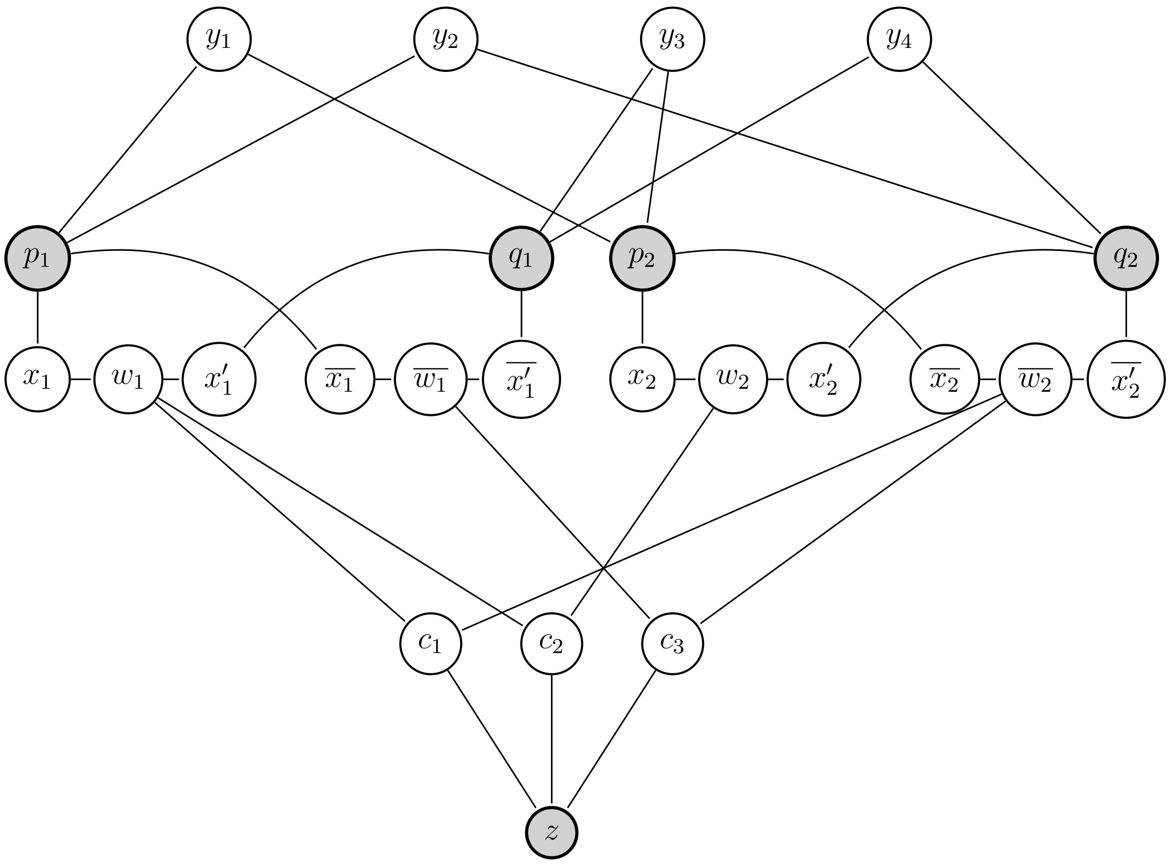


Figure 4.3: Figure illustrating the instance of the SGSR that arises from an instance of the $3-SAT_3$: $U = \{x_1, x_2\}$, $C = \{c_1, c_2, c_3\}$, with $c_1 = (x_1, \overline{x_2})$, $c_2 = (x_1, x_2)$ and $c_3 = (\overline{x_1}, \overline{x_2})$. The vertices marked in gray belongs to S.

disjunction of 2 or 3 literals (a variable or a negated variable). In addition, any variable appears in 2 or 3 clauses. The problem is to decide if there is a truth assignment that satisfies all clauses in C.

Theorem 4.2.1. *The* STRONG GEODETIC SET RECOGNITION *problem is NP-complete.*

Proof. Let $G = (V, E)$ and $S \subseteq V$ denote an instance of the strong geodetic recognition problem. The problem is in NP, because given a set $I(S)$ of shortest paths used to pass through all vertices in V, one can verify in polynomial time that all vertices of G are covered by the paths and that each pair u, v of vertices in S has exactly one valid u, v-shortest path that belongs to $I(S)$.

Now, it remains to prove that the problem is NP-Hard. We will present a polynomial reduction from $3-SAT_3$ to the STRONG GEODETIC SET RECOGNITION problem. Let $U = \{x_1, x_2, \dots, x_n\}$ be the set of variables and $C = \{c_1, c_2, \dots, c_m\}$ be the set of clauses of a $3-SAT_3$ instance. We will assume that any variable appears 2 or 3 times on the set of clauses, also, assume that every variable appears at least once on its positive form and once on its negative form. We can assume that because if a variable only appears either on a positive or negative form, we can construct an equivalent instance removing this variable and the clauses it appears by setting such literals as true or false, respectively. Note that each literal can satisfy at most 2 clauses.

Now, construct an instance of the strong geodetic set recognition problem in a graph $G = (V, E)$ defined as follows (Figure 4.3 shows an example of the construction): for each variable $x_i \in U$ add a gadget containing 8 vertices (variable gadget): x_i, x_i', $\overline{x_i}$, $\overline{x_i'}$, w_i, $\overline{w_i}$, p_i and q_i. Then, add the edges $x_i w_i$, $w_i x_i'$, $\overline{x_i w_i}$, $\overline{w_i}\overline{x_i'}$, $q_i\overline{x_i'}$, $q_i x_i'$, $p_i x_i$ and $p_i\overline{x_i}$.

Now, for each clause $c_i \in C$ add a vertex c_i. Moreover, add a vertex z adjacent to all vertices $c_i \in C$. It remains to add the edges that represent the relation between variables and clauses. Let $c_i \in C$ be a clause. For each positive literal $x_i \in c_i$ add the edge $c_i w_i$, and, for each negative literal $\overline{x_i} \in c_i$ add the edge $c_i\overline{w_i}$. Repeat this procedure for all clauses in C.

The last part of the construction is: for every pair of vertices (p_i, p_j), (p_i, q_j) and, (q_i, q_j) with $i \neq j$ add a new vertex y, an edge between the first vertex of the pair and y and an edge between y and the second vertex of the pair. Thus, creating a path of size 2 between each pair of vertices as described. We define $[n] = \{1, 2, \ldots, n\}$, now let $P = \{p_i \mid i \in [n]\}$, $Q = \{q_i \mid i \in [n]\}$, $W = \{w_i, \overline{w_i} \mid i \in [n]\}$ and $S = P \cup Q \cup \{z\}$. Finally, for the constructed instance, it consists in deciding whether S is a strong geodetic set in G. Observe that the size of the graph G is limited by a polynomial function on the size of the $3 - SAT_3$ instance.

Now we prove that if the instance of $3 - SAT_3$ given by the set of variables U and clauses C is satisfiable, then S is a strong geodetic set in G. Let $T = \{t_1, t_2, \ldots, t_n\}$ be a truth assignment of U which satisfies all clauses in C. At first, note that the length of a shortest path between a vertex in $P \cup Q$ and z is 4. So, if x_i is set to true, then assign the (p_i, x_i, w_i, c, z) shortest path between p_i and z and the (q_i, x_i', w_i, c', z) shortest path between q_i and z, with c and c' denoting the clauses that the literal x_i satisfies, observe that any literal satisfies one or two clauses, thus, if two clauses are satisfied $c \neq c'$, else, $c = c'$. Now, assign the shortest path $(p_i, \overline{x_i}, \overline{w_i}, \overline{x_i'}, q_i)$ between p_i and q_i, note that $D(p_i, q_i) = 4$.

If x_i is set to false in T, then the paths will be chosen on an analogous way. We will choose the paths $(p_i, \overline{x_i}, \overline{w_i}, c, z)$, $(q_i, \overline{x_i'}, \overline{w_i}, c', z)$, $(p_i, x_i, w_i, x_i', q_i)$. By this time, all vertices in variable gadgets and all clause vertices are covered. This holds because the vertices w_i (representing a positive literal) and $\overline{w_i}$(representing a negative literal) are adjacent to all clauses (clause vertices) that each one satisfies, and it is possible to cover these clauses with (p_i, z) and (q_i, z)-shortest paths. It remains to define the paths between vertices in S that are in different variable gadgets. We assign to $I(S)$ the unique 2-length shortest path between these vertices. Finally, note that all vertices are covered, hence, S is a strong geodetic set of G.

Now, assume that S is a strong geodetic set of G, with $I(S)$ being its assignment of shortest paths. Consider the variable $x_i \in U$ and observe that the (p_i, z) and (q_i, z) shortest paths have two options:

- The p_i, z-shortest path passes through x_i and the q_i, z-shortest path passes through x_i'.

- The p_i, z-shortest path passes through $\overline{x_i}$ and the q_i, z-shortest path passes through $\overline{x_i'}$.

This affirmation holds because, otherwise, one of the vertices in $\{x_i, x_i', \overline{x_i}, \overline{x_i'}\}$ would not be covered, as the p_i, q_i-shortest path can cover either x_i and x_i' or $\overline{x_i}$ and $\overline{x_i'}$.

Therefore, the variable gadget forces a choice between either a positive or a negative literal. It is important to note that only shortest paths between z and a vertex in $P \cup Q$ are able to cover clause vertices. Now, consider the following truth assignment for U. For each $x_i \in U$, if $I(S)$ assigns the (p_i, z) and (q_i, z) shortest paths to pass through x_i and x_i' set x_i to true, and if $I(S)$ assigns the (p_i, z) and (q_i, z) shortest paths to pass through $\overline{x_i}$ and $\overline{x_i'}$ set x_i to false. This truth assignment satisfies all clauses in C, since S is a strong geodetic set of G, which must cover all clause vertices. Hence, the $3 - SAT_3$ instance is satisfiable and the proof is concluded.

□

Corollary 4.2.2. *The* STRONG GEODETIC SET RECOGNITION *problem is NP-complete even when restricted to bipartite graphs with diameter bounded by 6.*

Proof. Consider the graph $G = (V, E)$ constructed on Theorem 4.2.1. Let $X = \{x_i, x_i', \overline{x_i}, \overline{x_i'}\}$ and let Y be a set containing all y vertices of G. Now, let $A = P \cup Q \cup W \cup \{z\}$ and $B = C \cup X \cup Y$. Note that the vertices in G can be partitioned into A and B, which are both independent sets, hence, G is bipartite. Also, observe that the largest distance in the graph occurs between a vertex y in Y and a clause vertex that is not satisfied by either variable gadgets adjacent to y, this distance is 6. This confirms the corollary. □

Corollary 4.2.3. *The* STRONG GEODETIC SET *problem is NP-complete even when restricted to bipartite graphs.*

Proof. Recall Theorem 2.3.1. The theorem assures that we can reduce any instance of the STRONG GEODETIC SET RECOGNITION problem restricted to bipartite graphs to an instance of the STRONG GEODETIC SET problem restricted to bipartite graphs. Note that if a graph is bipartite, then the resulting graph after the reduction will still be bipartite, because only one-degree vertices are added. This result was first published in Iršič [2018], but we choose to state it here because it is a straightforward corollary of Theorem 4.2.1. □

Theorem 4.2.4. *The* STRONG GEODETIC SET RECOGNITION *problem restricted to bipartite graphs with max-degree bounded by 4 is NP-complete.*

Proof. In this proof we will use an adaptation of the reduction presented on Theorem 4.2.1. We will reduce an instance Π of $3-SAT_3$ to an instance Π' of SGSR. Let $U = \{x_1, x_2, ..., x_n\}$ be the set of variables and $C = \{c_1, c_2, ..., c_m\}$ be the set of clauses of the $3-SAT_3$ instance Π, we will consider the same assumptions made at Theorem 4.2.1 about the instance. We also assume that $|U|$ and $|C|$ are exact powers of 2, dummy variables and clauses can be added to a generic instance in order to satisfy this constraint.

Now, let $G = (V, E)$ be the graph associated with an instance of SGSR obtained after the reduction explained at Theorem 4.2.1. We will present some adaptations on G in order to construct a graph G' associated with the instance Π'. Let $G' = G[P \cup Q \cup X \cup W \cup C]$, now do the following modifications to G': add a vertex z and connect z to all clause vertices using a binary tree T_z. Illustrating, suppose that there are 8 clause vertices, then z is connected to two added auxiliary vertices a_1 and a_2. Afterwards, add the auxiliary vertices a_3, a_4, a_5 and a_6, with a_1 adjacent to a_3 and a_4 and a_2 adjacent to a_5 and a_6. Finally, the vertices a_3, a_4, a_5 and a_6 connects directly to the clause vertices and the introduced binary tree is complete. Observe that the introduction of this gadget makes z a 2-degree vertex and all introduced auxiliary vertices have degree 3.

Recall that $P = \{p_1, p_2, \dots, p_n\}$ and $Q = \{q_1, q_2, \dots, q_n\}$. Now, add the following gadgets to G': Add a vertex $\overline{p}$ and connect $\overline{p}$ to all vertices in P using a binary tree T_P, as explained previously. Analogously, Add a vertex $\overline{q}$ and connect $\overline{q}$ to all vertices in Q using an additional binary tree T_Q. Finally, add a vertex y and the edges $\overline{p}y$ and $\overline{q}y$ (connecting the trees T_P and T_Q), the resulting binary tree is called T_y. Concluding the construction, let $\alpha = \log_2 n$ and $\beta = \alpha - 1$. If $\beta > 1$, then, for every edge e among the edges $p_i x_i$, $p_i \overline{x_i}$, $q_i \overline{x_i'}$ and $q_i x_i'$ for every $i \in \{1, 2, \dots, n\}$, replace e by a path P_e having β edges. The construction of G' is complete, now it remains to prove that the $3-SAT_3$ instance Π is equivalent to recognizing whether the set $S = P \cup Q \cup z$ is a strong geodetic set of G' (instance Π'). Observe that G' is a bipartite graph with max-degree equals 4 (a clause vertex associated with a 3-sized clause has exactly 4 neighbours).

Assume that the $3-SAT_3$ instance Π is satisfiable, then there exists a truth assignment $T = \{t_1, t_2, \ldots, t_n\}$ of U that satisfies all clauses in C. Now, we construct a shortest path assignment $I(S)$ proving that S is a strong geodetic set of G' (instance Π'). For every variable x_i that is set to true in the truth assignment T do the following:

- let $C_{p_i,z}$ be a shortest path between p_i and z such that $C_{p_i,z} = (p_i, P_{p_i,x_i}, w_i, c, P_{c,z})$, here P_{p_i,x_i} denotes the path that replaces the edge p_ix_i in the construction and $P_{c,z}$ denotes the unique shortest-path between a clause vertex $c \in N(w_i)$ and z, observe that this shortest path lies in a binary tree added to the construction. Add $C_{p_i,z}$ to $I(S)$.

- Analogously, let $C_{q_i,z}$ be a shortest path between q_i and z such that $C_{q_i,z} = (q_i, P_{q_i,x'_i}, w_i, c', P_{c',z})$, here, if w_i is adjacent to 2 different clause vertices, then $c' \in N(w_i)$ and $c \neq c'$, otherwise, $c = c'$. Add $C_{q_i,z}$ to $I(S)$.

- Finally, let C_{p_i,q_i} be a shortest path between p_i and q_i such that $C_{p_i,q_i} = (p_i, P_{p_i,\overline{x_i}}, \overline{w_i}, \overline{x'_i}, P_{\overline{x'_i},q_i})$. Add C_{p_i,q_i} to $I(S)$.

Variables that are set to false will be treated analogously, refer to Theorem 4.2.1. Now, observe that all vertices in variable gadgets are covered. Moreover, given that the $3-SAT_3$ instance Π is satisfiable all clause vertices are covered as well, because the shortest path assignment explained covers (satisfies) the same clause vertices (clauses) as the truth assignment T. This also implies that all auxiliary vertices in the binary tree T_z are covered. Now it remains to determine the shortest paths between vertices in $S \setminus \{z\}$ lying in different variable gadgets. Every such paths will traverse the binary tree T_y, covering all auxiliary vertices in it. Finally, all vertices of G' are covered and S is a strong geodetic set of G'.

For the converse, assume that S is a strong geodetic set of G', hence, there exists a shortest path assignment $I(S)$ that covers all vertices of G'. First, note that for every variable $x_i \in U$, both shortest paths between p_i, z and between q_i, z must traverse either w_i or $\overline{w_i}$, the argument explained at Theorem 4.2.1 is also valid here. Now, observe that the p_i, q_i-shortest path has size equals $2 \times \alpha$, hence, there is two shortest paths options, one traverse the binary tree T_y and another traverse the variable gadget. In $I(S)$ all such shortest paths must choose the second option. Moreover, observe that shortest paths between vertices in $S \setminus \{z\}$ from different variable gadgets will always traverse the binary tree T_y, avoiding that these paths cover clause vertices.

Concluding, since every variable gadget forces a choice between a positive or a negated literal, the existence of a shortest path assignment covering all clause vertices indicates the existence of a truth assignment at the instance Π that satisfies all clauses, thus, the proof is concluded. $\square$

Chapter 5

An exact algorithm and parameterized complexity results

We present an exact exponential algorithm for the SGSR for general graphs. The algorithm backtracks on the options of covering each vertex on a more clever manner than simply backtracking on every option of shortest path between each vertex pair. Then, we present some parameterized complexity results based on the combinatorial structure of the problems.

5.1 An exact algorithm to solve the strong geodetic set recognition problem

A pseudo-code is described at Algorithm 1. Initially, we construct a data structure indicating every shortest path that can be assigned to cover each vertex, this can be done using a breadth first search algorithm. Then, for each vertex $x \notin S$ the algorithm tries every possible i,j-shortest path to cover it, with $i, j \in S$. During the recursion every combination of ways to cover the vertices will be tested, granting the correctness of the algorithm. Observe that when an i,j-shortest path is set to cover some vertex x the i,j-shortest path is no longer available, instead, the (i,x) and (x,j) shortest paths become available and the *covering* data structure is updated accordingly.

A naive approach to solve the problem would be to test recursively every combination of shortest paths choices between each pair of vertices in S. Note that the number of different shortest paths between two vertices of a graph can be exponential on the number of vertices at the graph. Let ϕ denote the greatest number of different shortest paths between a pair of vertices on a graph. Thereafter the complexity of this

---

Algorithm 1: Strong geodetic set recognition

Input: A graph $G = (V, E)$ and $S \subseteq V$.
Output: Indicates if S is a strong geodetic set of G. If so returns a paths choice $I(S)$ that covers all vertices in V.

Step 1. (Initialization)
Compute a breadth first search for all vertices in V, storing the distance between each pair of vertices;

Step 2. (Pre-processing)
For each pair of distinct vertices i, j in S compute the i,j-interval. Then construct a data structure named *covering* that indicates, for each vertex $x \in V$, which shortest paths can have x as an interval vertex;
Define a set *paths* that contains all pairs of vertices in S;
Define a set $notCovered$ with vertices not covered by shortest paths yet, initialize it with $V \setminus S$;

Step 3. (Recursive function)

```
Function(covering, paths, notCovered)
if notCovered is empty then
|   return 1;
end
if paths is empty then
|   return 0;
end
v ← notCovered[0]; //an arbitrary vertex on the notCovered set
if covering[v] is empty then
|   return 0;
end
foreach i,j-shortest path that can pass though v, as indicated at covering do
|   Restore the parameters covering, paths and notCovered as received at the
|     function;
|   Remove v from notCovered;
|   remove (i, j) from paths;
|   add (i, v) and (v, j) to paths;
|   update covering observing the modification of paths;
|   call Function with the updated parameters;
|   if Function returns 1 then
|   |   return 1;
|   end
end
return 0;
```

Step 4. (Output)
If the function returns 0, then S is not a strong geodetic set;
Otherwise, it is possible to construct the shortest paths choice $I(S)$ by retrieving which shortest paths were assigned to cover each vertex;

---

naive algorithm would be $\mathcal{O}\left(\phi^{\binom{|S|}{2}}\right)$ disregarding polynomial factors.

A simple analysis of Algorithm 1 tells us that its time complexity on the worst case is $\mathcal{O}\left(\binom{|S|}{2}^{|V|-|S|}\right)$ ignoring polynomial factors. In fact, observe that for each vertex in $V \setminus S$ there are at most $\binom{|S|}{2}$ shortest paths that can be used to cover it, thus, the algorithm does a backtrack considering all possibilities. Observe that even though the number of available shortest paths grows after a shortest path split, the number of intervals containing any vertex v does not increases, as at most one of the two resulting shortest paths after a split can have v in its interval. The achieved time complexity is smaller than the complexity of the baseline. Observe that this algorithm can be used as a procedure for an algorithm solving the SGS. For instance, a straightforward algorithm would be: generate all possible subsets of V on a crescent manner, them verify whether each set is a strong geodetic set with Algorithm 1. The first set with a positive response is a minimum strong geodetic set.

5.2 Parameterized complexity results

We decided to investigate the STRONG GEODETIC SET problem for graphs with bounded diameter because we can assure that any shortest path will cover at most $D+1$ vertices of the graph, with D being the graph's diameter. We will see that this property will lead to the development of an FPT-algorithm. At first, we show a result that frustrates our first expectations.

Theorem 5.2.1. *The* STRONG GEODETIC SET *and the* STRONG GEODETIC SET RECOGNITION *problems parameterized by the diameter are not in XP, assuming $P \neq NP$.*

Proof. First, we will use the fact that the STRONG GEODETIC SET problem for co-bipartite graphs is NP-hard, as proved in Theorem 4.1.1. Recall that connected co-bipartite graphs have diameter bounded by 3. Assume, for contradiction, that the STRONG GEODETIC SET problem parameterized by the diameter is in XP, with an algorithm that solves the problem in $\mathcal{O}\left(f(D) \cdot n^{f(D)}\right)$ time, where n denotes the size of the instance. Consequently, it would be possible to solve the problem for co-bipartite graphs in polynomial time, implying that $P = NP$. Finally, assuming that $P \neq NP$, we conclude that the STRONG GEODETIC SET problem parameterized by the diameter is not in XP.

Using the same argument we can prove that the STRONG GEODETIC SET RECOGNITION problem parameterized by the diameter is also not in XP. At Theorem 4.2.1, the SGSR was proved to be NP-complete through a reduction that results on a graph with diameter equal to 6. Consequently, an XP algorithm would imply that $P = NP$, which we assume to be false. □

Theorem 5.2.2. *Let $G = (V, E)$ be a graph with diameter D. The problem of deciding whether G has a strong geodetic set S with cardinality k is fixed parameter tractable on the parameters D and k.*

Proof. Let $S \subseteq V$ with $|S| = k$ and let $U = V \setminus S$, note that $|U| = |V| - k$. If S is a strong geodetic set, then every vertex in U must be internal of some shortest path between vertices in S. In addition, observe that there are $\binom{k}{2}$ pairs of vertices of S, and for each of these pairs it will be assigned a shortest path that will cover at most $D - 1$ vertices in U. Therefore, if

$$|V| - k > \binom{k}{2} \times (D - 1),$$

then no set S with cardinality k can be a strong geodetic set. Otherwise:

$$|V| \leq \binom{k}{2} \times (D - 1) + k.$$

Resulting that the size of the graph is bounded by a polynomial function of D and k, which means that we found a polynomial kernel of the problem in polynomial time. Therefore, the STRONG GEODETIC SET problem parameterized by D and k is fixed parameter tractable. □

Observe that the SGSR is FPT on the parameters D and k, as well. But now k indicates the size of the set S given as input. The same argumentation applies.

Chapter 6

Conclusion and further works

6.1 Conclusion

In this work we achieved proofs for the NP-Completeness of the SGS for chordal graphs and for co-bipartite graphs. The structure of the referred proofs are similar, having as a base the dominating set problem. We perceived that SGS has a structure that can be related to the dominating set structure by reductions using graphs of diameter 2, resulting in our NP-completeness proofs. At the same time, this design gave us an interesting result: the fact that SGS parameterized by the diameter is not in XP. Consequently, although graphs with higher diameter may have more shortest paths options between a pair of vertices, the hardness of the problem is maintained at graphs with diameter 2.

As a contrast, we concluded that the SGSR is solvable in polynomial time for graphs with diameter 2. This was the first divergence between the computational complexity of the SGS and the SGSR, which corroborates the intuition that the SGSR is a computationally easier problem than the SGS. Moreover, we find a simple polynomial reduction from the SGSR to the SGS, which highlights the strong relation between the problems and suggests that the SGSR is indeed easier than the SGS and can be seen as a straightforward subproblem.

We found polynomial-time algorithms to solve the SGSR for split graphs and for graphs with diameter 2 using a maximum matching algorithm. A greedy algorithm to solve the SGS for threshold graphs was derived, the algorithm was mainly based on the nested neighborhood ordering present at threshold graphs. In addition, we described polynomial algorithms to solve the SGS and the SGSR for cacti graphs decomposing the problem on biconnected components. Polynomial-time algorithms for block graphs could be described on a similar manner, taking advantage of the decomposable structure

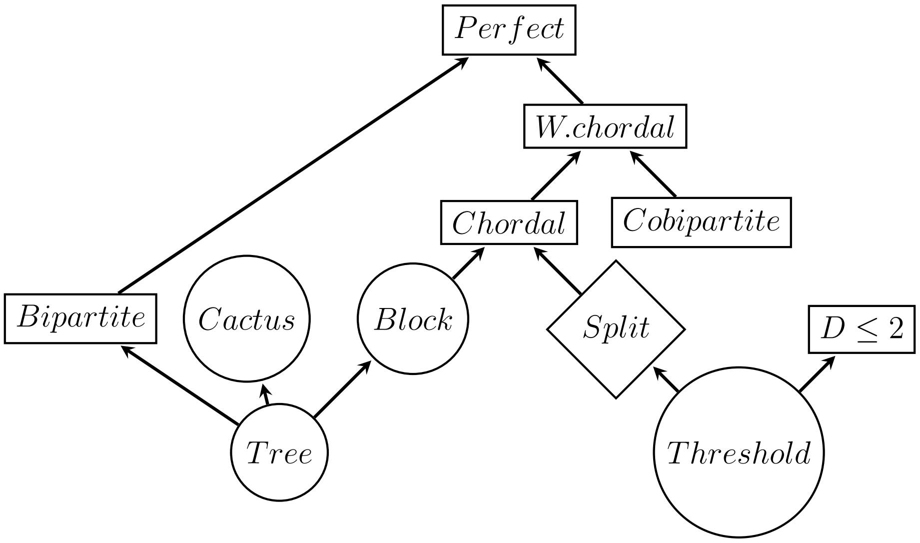


Figure 6.1: Figure containing a graph classes hierarchy indicating the complexity status of the SGS for each graph class. Circular nodes indicate problems solvable in polynomial time, rectangular nodes indicate NP-Complete problems and nodes with diamond shape indicate problems with unknown complexity status. The arrows of the edges point to super-classes, W. chordal refers to the weakly chordal class and $D \leq 2$ reefers to graphs with diameter at most 2.

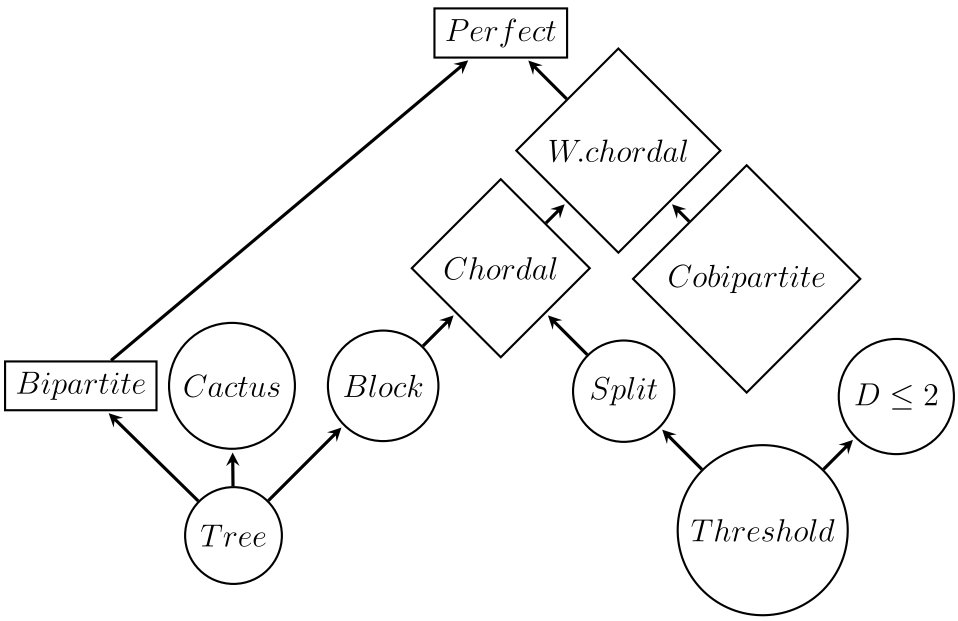


Figure 6.2: Figure containing a graph classes hierarchy indicating the complexity status of the SGSR for each graph class. The figure has the same notations as Figure 6.1.

of the graph. Figures 6.1 and 6.2 display complexity results to the SGS and SGSR, respectively. Besides the complexity statuses of the SGS for perfect graphs, bipartite graphs and trees, all results presented here are our contributions.

When we first try to prove the NP-completeness of the SGSR we attempted several reductions from the dominating set, using similar designs to those used at Theorem 4.1.1 and Theorem 4.1.2. However, the structure of the dominating set problem seemed to be harder to relate with the SGSR, then we reduced from the $3 - SAT_3$, which finally gave us the NP-completeness of the SGSR. The obtained reduction also determinates that the SGSR is NP-hard for bipartite graphs and for graphs with diameter bounded by 6, consequently, the SGSR parameterized by the diameter is not in XP, which is another result that indicates its difficulty. The referred NP-completeness proof is probably the most important result of the work, increasing the understanding of the SGS and also gathering knowledge about a new convexity problem with interesting applications (SGSR).

6.2 Further works

As we are dealing with recently defined problems: SGS was defined by Manuel et al. [2018] and SGSR was defined at the present dissertation, there are several open problems concerning these two topics. Here we present a list of them:

1. Is the SGS restricted to split graphs NP-Complete? This is an interesting question because we have the complexity for chordal graphs and for threshold graphs, but the complexity for split graphs remains open.

2. Does the SGSR parameterized only by the number of vertices on S belongs to FPT or XP? This would tell us whether the problem for a fixed number of vertices is polynomial-time solvable.

3. Does the SGSR parameterized by the treewidth belongs to FPT or XP? It seems to be an interesting question, because graphs with small treewidth tend to have fewer shortest path options between any pair of vertices.

4. Is the SGS parameterized by k W[2]-hard? We believe that is the case because it is a problem that seems to be at least as hard as the dominating set problem parameterized by the natural parameter, a W[2]-hard problem.

5. Are there any good approximation algorithms to solve the SGS and SGSR problems? What are the best possible approximation ratios?

6. We have that the SGSR for graphs with diameter bounded by 2 is polynomially solvable, we also have that the SGSR for graphs with diameter at least 6 is NP-complete, what is the complexity status of the problem for graphs with diameters 3, 4 and 5?

7. The studied problems have complex combinatorial structures and probably can not be solved viably by exact algorithms for practical purposes. Are there any efficient heuristics to solve these problems?

8. Can SGSR be solved by a single exponential-time exact algorithm?

The enumerated problems are natural questions that came up during our research. We believe that answering these questions will increase the understanding of the SGS and SGSR. We hope to answer some of these questions on future works and hope that more researchers get interested on these topics.

Bibliography

Alom, B. M., Das, S., and Islam, M. S. (2010). Finding the maximum matching in a bipartite graph. *DUET Journal*, 1(1):1--4.

Bertossi, A. A. (1984). Dominating sets for split and bipartite graphs. *Information processing letters*, 19(1):37--40.

Bondy, J. A., Murty, U. S. R., et al. (1976). *Graph theory with applications*, volume 290. Citeseer.

Dourado, M. C., Protti, F., Rautenbach, D., and Szwarcfiter, J. L. (2010). Some remarks on the geodetic number of a graph. *Discrete Mathematics*, 310(4):832--837.

Downey, R. G. and Fellows, M. R. (2012). *Parameterized complexity*. Springer Science & Business Media.

Ekim, T. and Erey, A. (2014). Block decomposition approach to compute a minimum geodetic set. *RAIRO-Operations Research*, 48(4):497--507.

Everett, M. G. and Seidman, S. B. (1985). The hull number of a graph. *Discrete Mathematics*, 57(3):217--223.

Gledel, V., Iršič, V., and Klavžar, S. (2018). Strong geodetic cores and cartesian product graphs. *arXiv preprint arXiv:1803.11423*.

Grandoni, F. (2006). A note on the complexity of minimum dominating set. *Journal of Discrete Algorithms*, 4(2):209--214.

Harary, F., Loukakis, E., and Tsouros, C. (1993). The geodetic number of a graph. *Mathematical and Computer Modelling*, 17(11):89--95.

Heggernes, P. and Kratsch, D. (2007). Linear-time certifying recognition algorithms and forbidden induced subgraphs. *Nordic Journal of Computing*, 14(1-2):87--108.

Hernando, C., Jiang, T., Mora, M., Pelayo, I. M., and Seara, C. (2005). On the steiner, geodetic and hull numbers of graphs. *Discrete Mathematics*, 293(1):139--154.

Iršič, V. (2018). Strong geodetic number of complete bipartite graphs and of graphs with specified diameter. *Graphs and Combinatorics*, 34(3):443--456.

Iršič, V. and Klavžar, S. (2018). Strong geodetic problem on cartesian products of graphs. *RAIRO-Operations Research*, 52(1):205–216.

Klavžar, S. and Manuel, P. (2018). Strong geodetic problem in grid-like architectures. *Bulletin of the Malaysian Mathematical Sciences Society*, 41(3):1671--1680.

Mahadev, N. V. and Peled, U. N. (1995). *Threshold graphs and related topics*, volume 56. Elsevier.

Manuel, P., Klavžar, S., Xavier, A., Arokiaraj, A., and Thomas, E. (2016). Strong edge geodetic problem in networks. *Open Mathematics*, 15(1):1225--1235.

Manuel, P., Klavžar, S., Xavier, A., Arokiaraj, A., and Thomas, E. (2018). Strong geodetic problem in networks. *Discussiones Mathematicae Graph Theory*.

Pan, J.-J. and Chang, G. J. (2006). Isometric path numbers of graphs. *Discrete Mathematics*, 306(17):2091--2096.

Schaefer, T. J. (1978). The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216--226. ACM.