UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

AN EXPERIMENTAL GUIDED APPROACH TO THE METRIC DIMENSION ON DIFFERENT GRAPH FAMILIES

ALEX HERRERO BRAVO

Thesis supervisor: ANTONI LOZANO BOIXADORS (Department of Computer Science)

Degree: Bachelor's Degree in Informatics Engineering (Computing)

Bachelor's thesis

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

26/01/2024

Abstract

This undergraduate thesis explores the determination of the metric dimension of various graphs for which certain researchers have provided bounds or have not thoroughly investigated. The metric dimension of a graph is a well-known concept in graph theory, representing the smallest number of vertices required to uniquely identify all other vertices in the graph using distances.

This study aims to extend the current knowledge and potentially discover tighter bounds for the metric dimension. To achieve this, we will make use of solvers, which are powerful computational tools for solving NP-Complete problems. Through a comprehensive analysis of different graphs and their associated metrics, this research contributes to the understanding of the metric dimension and provides valuable insights for future investigations in graph theory.

Abstract

Este Trabajo Final de Grado explora la determinación de la dimensión métrica de varios grafos para los que ciertos investigadores han proporcionado cotas o no han investigado a fondo. La dimensión métrica de un grafo es un concepto bien conocido en teoría de grafos, representando el número mínimo de vértices necesarios para identificar de manera única todos los demás vértices en el grafo usando distancias.

Este estudio tiene como objetivo ampliar el conocimiento actual y potencialmente descubrir cotas más estrictas para la dimensión métrica. Para lograr esto, utilizaremos *solvers*, que son herramientas computacionales poderosas para resolver problemas NP-Completos. A través de un análisis exhaustivo de diferentes grafos y sus métricas asociadas, esta investigación contribuye a la comprensión de la dimensión métrica y proporciona valiosos conocimientos para futuras investigaciones en teoría de grafos.

Abstract

Aquesta treball final de grau explora la determinació de la dimensió mètrica de diversos grafs pels quals alguns investigadors han proporcionat fites o no han investigat a fons. La dimensió mètrica d'un graf és un concepte ben conegut en la teoria de grafs, que representa el nombre més petit de vèrtexos necessaris per identificar de manera única tots els altres vèrtexos en el graf utilitzant distàncies.

Aquest estudi té com a objectiu ampliar els coneixements actuals i potencialment descobrir noves fites més ajustades per a la dimensió mètrica. Per aconseguir-ho, farem ús de *solvers*, que són eines computacionals potents per resoldre problemes NP-complets. Mitjançant un anàlisi exhaustiu de diferents grafs i les seves mètriques associades, aquesta recerca contribueix a la comprensió de la dimensió mètrica i proporciona idees valuoses per a futures investigacions en teoria de grafs.

Contents

List of Figures

List of Tables

1 Introduction

P vs NP is one of the seven millennium prize problems[1], awarded with 1 million dollars, which asks a simple question: If a problem is easy to check the correctness of a solution (these problems are called NP problems), is it also easy to find a solution efficiently? These problems are called P problems. In this context, we assume that *easy* is synonymous with the existence of an *efficient* (more formally, polynomial time) algorithm.

For instance, sorting numbers is considered an easy problem because we have efficient algorithms for solving it. Conversely, given a set of cities and a distance d, it's NP-hard to decide whether there is a route of length at most d that traverses all cities exactly once.

Behind this way of classifying problems, there are special problems called NP-hard problems (because they are, at least, as hard as any NP problem) that have a curious property: any NP problem can essentially be reduced to an NP-hard problem efficiently (more formally, polynomial time). Now, it's easy to see that if we can solve one of these NP-hard problems, we can also solve any NP problem and the answer to the previous question, P vs NP, would be *yes* (because you could transform one problem into an NP-hard problem, solve this one, and transform the solution back to your initial problem).

Many real applications will arise for solving P vs NP question and will improve our efficiency solving any kind of real problems (finding optimal values in any situation, scheduling, optimal routes for delivering packages...), so it's crucial to have a well known knowledge on the state-of-the-art NP hard knowledge problems.

My objective on this Bachelor's thesis will be to provide experimental results (and theoretical results if I have the opportunity) in one particular NP-complete problem (NP-complete problem is an NP-hard problem that is also an NP problem) called *The Metric Dimension*, trying to contribute and encourage future research on this problem for solving the P vs NP question.

Given a graph $G = (V, E)$, we characterize the *Metric Dimension* of G as the cardinality of the smallest set $W \subseteq V$ such that for every pair $x, y \in V$, there exists one vertex $w \in W$ such that $d(w, x) \neq d(w, y)$ where $d(w, x)$ is the distance from w to x. In this case, we say that w resolves the pair x, y. The problem was first studied by Harary and Melter[2] and Slater[3] in 1976 and 1975, respectively, and is also known as the *Harary's problem*. The Metric Dimension problem is part of the famous Garey and Johnson's[4] book on computational complexity.

1.1 Context

This Bachelor's thesis was developed at the Facultat d'Informàtica de Barcelona, part of the Universitat Politècnica de Catalunya (UPC, also known as *BarcelonaTech*). The project was developed by Alex Herrero under the supervision of Antoni Lozano, a PhD in Computer Science specialised Complexity Theory and Graph Theory. This project is situated within the field of Graph Theory, specifically focusing on a well-known problem referred to as the Metric Dimension, which is known to be NP-complete, trying to contribute to the research of this problem.

1.2 Terms and Concepts

Before introducing the problem, background, and motivation, I must first provide some theoretical concepts to enhance understanding of the thesis.

- **Graph** $G = (V, E)$ is a set of nodes V with some relations called edges E. For instance **Figure** 1 could be an example of a graph with $V = \{x_1, x_2, x_3, x_4\}$ and $E = \{x_1x_2, x_1x_4, x_2x_3, x_3x_4\}$
- **Distance between two nodes** is the minimum number of edges that must be traversed to move from one node to another. If it is not possible to travel from one node to another, we say that the distance is infinite or an impossible number (for example -1 or n if we have n vertices). The distance between $u, v \in V$ is denoted as $d(u, v)$.
- **Metric Representation** in a graph $G = (V, E)$ of $v \in V$ with respect to a set $W \subset V$ where $W = \{w_1, w_2, \ldots, w_n\}$ is an ordered tuple defined as $r(v|W) = (d(w_1, v), d(w_2, v), \ldots, d(w_n, v))$
- **Resolving Set** is a subset W of vertices such that any Metric Representation is unique in the graph. The smallest cardinality set among the resolving sets is also called the *basis*.
- **Metric Dimension** refers to the challenge of finding the resolving set with the smallest cardinality. When talking about a particular graph, we are referring to the cardinality of the *basis*, commonly denoted as $\beta(G)$. The decisional version of the Metric Dimension is an NP-complete problem.
- **Graph Family** referrs to a collection of graphs that shares a common property.
- **Integer Linear Programming** (ILP) is an optimization problem where the goal is to maximize/minimize a linear function with some restrictions, one of them being that variables could take only integer values. This problem is a well-known NP-complete problem.
- **Integer Linear Programming Solver** is specialized software designed to solve ILP (Integer Linear Programming) instances, typically known for its high speed and efficiency.

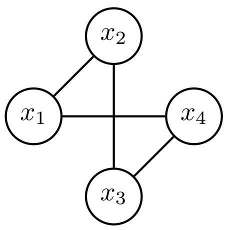


Figure 1: An example of a Graph, self elaborated

Let's give an example for the Metric Dimension: Suppose I want x_1 to be the only vertex in my resolving set. At a distance of 1 we have x_2 and x_4, so these vertices cannot be distinguished by their metric representation. Choosing any single vertex on this graph will have the same problem.

Now, suppose I choose x_1 and x_4 to comprise my resolving set and let $W = \{x_1, x_4\}$. In this case, the Metric Representation of each vertex is unique (see **Table 1**) and W is a resolving set since any metric representation in the graph is unique. For instance Metric Representation $(1, 2)$ would only referr to vertex x_2. Notice that W is the smallest resolving set we could find, so we say that W is a *basis* and the Metric Dimension of this graph is $\beta(G) = 2$, the number of vertices in W.

x_i	$d(x_1, x_i)$	$d(x_4, x_i)$	$r(x_i \vert W)$
x_1	0	1	$(0, 1)$
x_2	1	2	$(1, 2)$
x_3	2	1	$(2, 1)$
x_4	1	0	$(1, 0)$

Table 1: Distance and Metric Representation of each vertex, self-elaborated

1.3 Problem formulation

NP-complete problems play an essential role when talking about the famous P vs NP problem, so it's important to have well-known knowledge on this complexity class.

Metric Dimension is one NP-complete problem described before and studied by different UPC-FIB professors such as Lozano[5] (the director of this thesis) and Serna[6] among others. My goal for this project is to contribute experimental results to this problem and, if possible, theoretical results.

In particular, this thesis systematically collects and provides empirical results (and theoretical results if it's possible), for metric dimensions for various graph families contributing to future research work. The specific objectives of this research project include:

- **Data Collection**: Gather metric dimension data for different graph families.
- **Conjecture Testing**: Investigate conjectures related to metric dimensions in graph theory and use the empirical dataset to either support or refute these conjectures.
- **Improving Bounds**: Try to improve existing theoretical bounds for metric dimensions based on the empirical data and, if it's possible, mathematical proofs. Aim to develop more accurate and applicable bounds that better represent graph structures.
- **Study New Graphs Families**: Promote the investigation of non-studied graph families with our experimental results.
- **Stimulation of Further Research**: Promote the use of the dataset as a foundation for future studies in graph theory. Encourage researchers to build upon this dataset to address complex problems in the field. This is the main motivation for this project

By addressing these objectives, the thesis aims to bridge the gap between theoretical knowledge and empirical data in the field of metric dimensions in graph theory, ultimately advancing the understanding of graph properties and improving existing bounds (and also creating new ones).

1.4 Stakeholders

- **Director:** As I mentioned earlier, Antoni Lozano has conducted work on the Metric Dimension pertaining to a particular graph family called *tournaments*. Tournaments would be part of the thesis and it is possible that an experimental approach may contribute to future theoretical work.
- **Researchers:** This project contributes to the P vs NP problem by providing experimental data on different graph families. I hope that this project will prove useful to many researchers in the future.

- **General Public:** Although not direct stakeholders, the general public may benefit from the research findings in terms of potential applications or advancing their understanding of the metric dimension. For instance, the Metric Dimension could be used in Biology, Detecting Network Motifs [7], among others...

2 Justification

Many articles related to the Metric Dimension provided (without highlighting if it's optimal) lower/upper bounds for specific graph families without conducting any experimental results. Also authors provide conjectures and, again, without an experimental approach for supporting his words. In this context, this research project aims to bridge the gap by providing empirical results for the metric dimensions for different families of graphs. For instance T.Vetrik[8] provides the following table:

$\dim(C_n(1,2,3,4))$	Lower Bound	Upper Bound
$n \equiv 0 \pmod 8$	5	6
$n \equiv 1 \pmod 8$	5	6
$n \equiv 2 \pmod 8$	4	5
$n \equiv 3 \pmod 8$	4	5
$n \equiv 4 \pmod 8$	4	4
$n \equiv 5 \pmod 8$	4	5
$n \equiv 6 \pmod 8$	5	5
$n \equiv 7 \pmod 8$	5	6

Table 2: Lower and Upper Bounds for the Circulant Graphs by T.Vertik

And he conjectures that the Metric Dimension is exactly the Upper Bound. But he didn't provide any experimental results and/or theoretical arguments to support his conjecture. Also, it doesn't exist any *library* with experimental datasets for different graph families.

The need for experimental data is because:

- Theoretical bounds and conjectures are valuable starting points in graph theory research. By providing experimental data on metric dimensions, researchers can refine and improve their theories to make them more accurate and applicable.
- Conjectures in graph theory often lack empirical validation. Our research project seeks to provide data that can either support or refute these conjectures, thereby advancing the field's understanding of metric dimensions and related graph properties.
- By making metric dimension data available to the research community, we aim to stimulate further studies and investigations into this important field. This dataset can serve as a valuable resource for researchers looking to explore new avenues in graph theory.

Initially, Antoni Lozano and I contemplated the possibility of embarking on a fully theoretical thesis, which was our initial inclination. However, the decision to opt for an experimental thesis, rather than a purely theoretical one, stemmed from our recognition that, within a four-month timeframe, there was a possibility that we might not be able to deliver any significant results.

On the other hand, an experimental approach allows us to generate empirical data, which, at the very least, can serve as a valuable resource for fellow researchers in the field. Even in the absence of groundbreaking theoretical discoveries, the worst-case scenario is that we provide a dataset that future researchers can leverage and incorporate into their investigations. This strategic choice aligns with our objective to maximize the utility of our research efforts within the specified time constraints. Experimental approaches could always include theoretical results, resulting in a *middle point* between our initial idea and a thesis where we can contribute results for sure.

Another possibility could be to find an exact algorithm for the Metric Dimension. But, as I mentioned earlier, this is one of the most recognized hardest problem in mathematics and computer science and many researchers believe that $P \neq NP$ so no algorithm would exist[9].

3 Scope

3.1 Objective and Requirements

3.1.1 Main Objectives

As I mentioned in **section 1.3** this thesis will consider two main objectives:

1. **Provide experimental results** on different graph families for different questions such as having a dataset to stimulate future research, support or reject conjectures from different researchers and try to conjecture new (or more accurate) upper and lower bounds for the metric dimension on different families.

2. **Provide theoretical results** analysing experimental data. This probably is the most challenging part and knowing the time restrictions for this thesis I will not get more deeply if me or my director could not see anything interesting in data. If new theoretical knowledge arise from this part, we will provide more effort on this part since this could contribute significantly more than experimental approaches.

3.1.2 Requirements

To provide experimental results, I must first acquire the experimental data. The experimental data comprises the following subsections:

1. **Creating an ILP Formulation**: Given that the Metric Dimension problem is NP-complete, it's possible that no algorithm exists for solving it efficiently. In such cases, alternative methods must be considered. *Integer Linear Programming* (ILP) offers several powerful solvers equipped with extensive optimization capabilities that, in practice, solve ILP instances very quickly. As previously mentioned, since this problem falls within the category of optimization NP-complete problems, it is feasible to transform the Metric Dimension problem into an ILP problem and tackle it using dedicated solvers. This also entails verifying the correctness of my formulation and providing a mathematical proof.

2. **Creating a Workplace**: Once the formulation is created, I need to establish a workspace where instances of the Metric Dimension problem can be solved, and data can be generated. This involves becoming familiar with how to utilize the solver effectively.

3. **Graph Instances**: Finally, I need to identify a robust source from which I can obtain millions of instances of concrete graphs, as these will be the input data for the workplace.

3.2 Obstacles and Risks

Here below are some potential risks and obstacles that we may encounter during the development of this bachelor thesis:

- **Unexpected or Underwhelming Results:** There is a possibility that the data may not exhibit any discernible patterns, and analysing the graphs themselves may not yield meaningful results. This could lead to the potential waste of time spent on analysis without discovering relevant information, more than providing my results.
- **Problems with Data Source:** Since the data originates from other authors, I must rely on the assumption that the author provided correct data. If one experiment fails, I have to repeat it with another source.
- **Skill level:** As I'm not an experienced researcher, I may overlook important aspects when analysing the graphs and making decisions.
- **Miscellaneous Other Problems:** Family or personal issues could always arise.

4 Methodology and Rigor

Several established software development methodologies are currently in use across the software industry. *Agile* methodology is renowned for its incremental and efficient approach, ensuring the maintenance of a functional version of the project. *Sprints* are integral to *Agile*, representing predefined cycles with fixed durations. In my case, as Antoni Lozano requested the implementation of the ILP formulation with the solver by July, my *sprints* will not be focused on the development direction. Instead, I will create *sprints* for research, experimental work, and theoretical results. The duration of these *sprints* may vary, especially in the theoretical phase, where unexpected discoveries could occur. However, I will establish time limits and/or average durations for each *sprint* to maintain progress and structure.

To maintain rigor and organize my thoughts effectively, I will leverage Obsidian[10], a note-taking and knowledge management tool. Obsidian will serve as a valuable resource for documenting my progress, insights, challenges, and decisions throughout the project. It will enable me to maintain a detailed diary of my work, ensuring a clear record of my thought processes and the project's evolution.

For version control and collaboration, GitHub[11] will be used as a repository. GitHub's robust version control features will facilitate efficient management of code changes and saving my experimental results. The combination of Obsidian and GitHub will provide the necessary structure and organization to maintain rigor and transparency in the project, even as coding remains just one facet of the broader research effort.

In addition to these tools and methods, I will benefit from the guidance of my director, who not only oversees the project but also possesses extensive research experience. To ensure that the project stays aligned with research objectives and remains on track, I will hold weekly meetings with my director. These meetings will serve as opportunities to discuss progress, seek advice, address challenges, and refine the project's direction. Having a seasoned researcher as my director will provide valuable insights and expertise, enhancing the quality and rigor of the project.

5 Description of Tasks

This project started in the last week of June and will continue until the second week of July, during these weeks we specify the main objective of this thesis and our first ILP codification. Then, we will resume in the first week of September, with the aim to conclude the project by the last week of December with a total of (expected) **726.5 hours** (more information **Table 4**) with an average of 30/40 hours per week.

5.1 Management

This section encompasses all documentation related to the management of the thesis. All these tasks require *Overleaf* [12], an online LaTeX editor and a PC. They also require the preceding tasks be completed (see **Figure 2**)

- **M1 - Context and Scope:** Defining the context and scope of the project is expected to take approximately 20 hours.
- **M2 - Project Planning:** Establishing the project plan is estimated to require around 10 hours.
- **M3 - Budget and Sustainability:** Defining the budget and sustainability aspects is estimated to take about 15 hours.
- **M4 - Final Document:** Recollecting all these three documents and put it together in a single file (also fixing mistakes) will require about 20 hours.

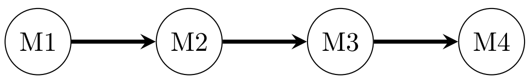


Figure 2: Task Dependencies for Managment, self elaborated

5.2 Development

5.2.1 Workplace

This section outlines the tasks necessary to establish an efficient workplace. I undertook these tasks at the request of my director during the last July week I met him.

- **W1 - Creating a Correct ILP Formulation:** Developing a precise *Integer Linear Programming* (ILP) formulation, along with providing a mathematical proof, to check its correctness, for the *Metric Dimension* problem, required approximately 2 hours. No specific requirements were needed for this task as I had acquired the necessary knowledge during my degree.
- **W2 - Becoming Familiar with CPLEX API:** CPLEX[13] is our chosen ILP solver due to its robustness and efficiency, I dedicated time to become acquainted with the solver and the C++ API provided by IBM. This task, including reviewing documentation[14], is estimated to take 10 hours. I may revisit it in the future as well.
- **W3 - Installing CPLEX:** The process of verifying my student status to obtain the CPLEX software for free and installing the correct version took approximately 10 hours. This task required the use of my laptop (where I'm going to execute all experiments).

- **W4 - Coding the Workplace:** Finally, I began coding the workplace. Given my familiarity with the programming language to be used, this task required only 5 hours. It is important to note that W4 is dependent on W1, W2, and W3 since I needed to incorporate the ILP formulation into my code, install CPLEX and become familiar with its API. Additionally, I relied on my designated PC for this project, as well as a suitable Code Editor.

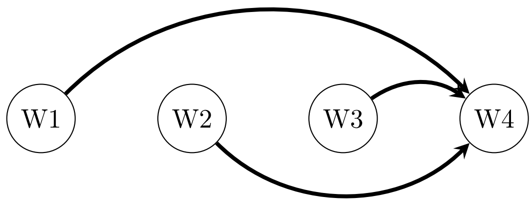


Figure 3: Task Dependencies for Workplace, self elaborated

5.2.2 Research and Analyze

Tasks related to the main objective of the project will be provided here. The total time spent on this task is counted per week, since these tasks are going to be repeated. In Gantt Diagram (which can be consulted in **Figure 7**) I will put it over time. Also, they require the preceding tasks to be done, except RA2 which will require W4 for obvious reasons.

- **RA1 - Choosing a Graph Family:** I need to update my knowledge on different graph families and select an interesting class based on current research papers. On average, this task is estimated to take around 7 hours.

- **RA2 - Collecting Data:** After choosing a graph family, I have to collect graph instances for getting his Metric Dimension. Data will be provided from [15] so, on average, it may take approximately 2 hours for solving all the instances for a particular family. This will require a PC with my coded workplace.

- **RA3 - Identifying Patterns:** Using the collected and solved data, I will attempt to identify common patterns. The duration of this task is uncertain, since it depends on the complexity of the data and patterns discovered. On average, it could take about 17 hours.

- **RA4 - Mathematical Proof:** This task is contingent on the findings from RA3; if there are interesting patterns, I will proceed to develop mathematical proofs. If not, this task may not be necessary. Assuming interesting patterns are found, I estimate dedicating approximately 40 hours to this task.

- **RA5 - Weekly Meetings:** Weekly meetings with my Director will be done for aiming our current data and intuition. The meetings have an estimation of 1'5 hours. Meetings doesn't need a previous task completed since Antoni can help me choosing graph family, identifying patterns or with a mathematical proof.

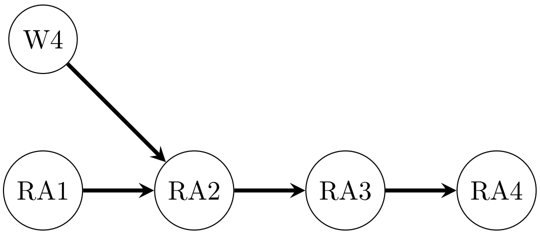


Figure 4: Task Dependencies for Research and Analyse, self elaborated

5.2.3 Documentation

While collecting data and performing mathematical proofs, it is crucial to document each step in the memory. These tasks will be carried out using the *Overleaf* online editor.

- **D1 - Documenting Data:** Following RA2, the next step is to record our data and findings in the memory. As I have prepared *scripts* to facilitate this process, it is estimated to take approximately 4 hours.
- **D2 - Documenting Proof:** After completing RA4, I will need to document the mathematical knowledge and formalism in the memory. On average, this task is expected to take around 15 hours.

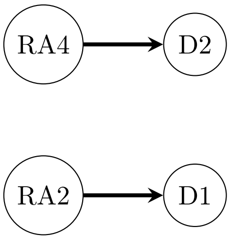


Figure 5: Task Dependencies for Documentation, self elaborated

5.2.4 Miscellaneous

Every project is subject to unpredictable issues. For instance, we could stumble upon an important topic, and my director and I might decide to allocate more effort to it, even if it's beyond my current expertise and requires additional time. Allocating a #6 Graph Family could be complicated because if any task is delayed, it could jeopardize the timely completion of the thesis. I have set aside miscellaneous time to account for unexpected problems or the possibility that my director and I want to approach the final stages of the thesis differently. If one family takes less time than expected, we may also consider adding a #6 family. If not, this time will be saved for reading again the memory and preparing the oral defense.

6 Human and Material Resources

To ensure the success of this thesis, I will assume various roles during different phases of the project. As I said before, I will need only a computer as a material resource, the place where I'll be executing experiments. Let's categorize these primary roles and provide justifications for each:

- **Researcher:** Tasks W1, RA1, RA2, RA3, RA4, D1, and D2 are related to generating new theoretical mathematical knowledge. Therefore, the role responsible for contributing this novel theoretical knowledge is unequivocally a researcher. During this phase, I will require access to my computer, which will serve as a platform for accessing important articles, online books to update my knowledge, and, of course, for documenting this thesis.

- **Developer:** Tasks W2, W3, and W4 will leverage all of my programming skills and incorporate new technologies, such as the CPLEX API. This is the part of the project where I'm going to develop a new tool to execute experiments, and this job is for a developer. Documentation for CPLEX is available on the IBM website[13] and, for this, I will also need a computer.

- **Project Manager:** Prior to commencing any task, it is imperative to establish a well-structured schedule to ensure the thesis's successful completion within the allocated timeframe. Effective scheduling plays a pivotal role in project management. This role will cover all tasks related to the project management of the thesis such as M1, M2, M3 and M4.

Apart from myself, I expect to require the assistance of only two individuals: my GEP tutor (**GEP-T**), Paola Lorenza, and my director (**D**), Antoni Lozano. Paola can assist me with all the content related to project management, while Antoni can provide valuable support with the formalism and experience that a researcher needs. In **Table 4**, I specify tasks where I could need their help.

Tasks	Corresponding Rol	Material Resource
W1, RA1, RA2, RA3, RA4, D1, D2	Researcher	Computer
W2, W3, W4	Developer	Computer
M1, M2, M3, M4	Project Manager	-

Table 3: Human Resources tasks related (summary), self elaborated

7 Estimations and Gantt

This section provides an overview of all the tasks, requirements, durations, and the Gantt chart. The tasks related to **Research and Analyse** are expected to be executed five times, once for each graph family. If my director and I decide to allocate more time to a specific graph family or choose an alternative approach, I will document it in the report and provide an explanation.

Task Description	**Acronym**	**Duration**	**Tasks to be Completed Before**	**Requirements (if needed)**	**Human Resource**
Context and Scope	M1	20 hours	-	*Overleaf*, PC	GEP-T
Project Planning	M2	10 hours	M1	*Overleaf*, PC	GEP-T
Budget and Sustainability	M3	15 hours	M2	*Overleaf*, PC	GEP-T
Final Document	M4	20 hours	M3	*Overleaf*, PC	GEP-T
Creating a Correct ILP Formulation	W1	2 hours	-	-	-
Becoming Familiar with CPLEX API	W2	10 hours	-	CPLEX Documentation[14]	-
Installing CPLEX	W3	10 hours	-	PC I'm gonna use for this project	-
Coding the Workplace	W4	5 hours	W1,W2,W3	Code Editor, PC	-
Choosing a Graph Family	RA1	35 hours	-	Research sources	D
Collecting Data	RA2	10 hours	W4, RA1	PC with workplace installed	-
Identifying Patterns	RA3	85 hours	RA2	Data Access	D
Mathematical Proof	RA4	200 hours	RA3	-	D
Weekly Meetings	RA5	25.5 hours	-	-	D
Documenting Data	D1	44 hours	RA2	*Overleaf*, PC	-
Documenting Proof	D2	165 hours	RA4	*Overleaf*, PC	D
Miscellaneous	MISC	70 hours	-	-	Unexpected
Total: 726.5 hours					

Table 4: Summary of Project Tasks, self elaborated

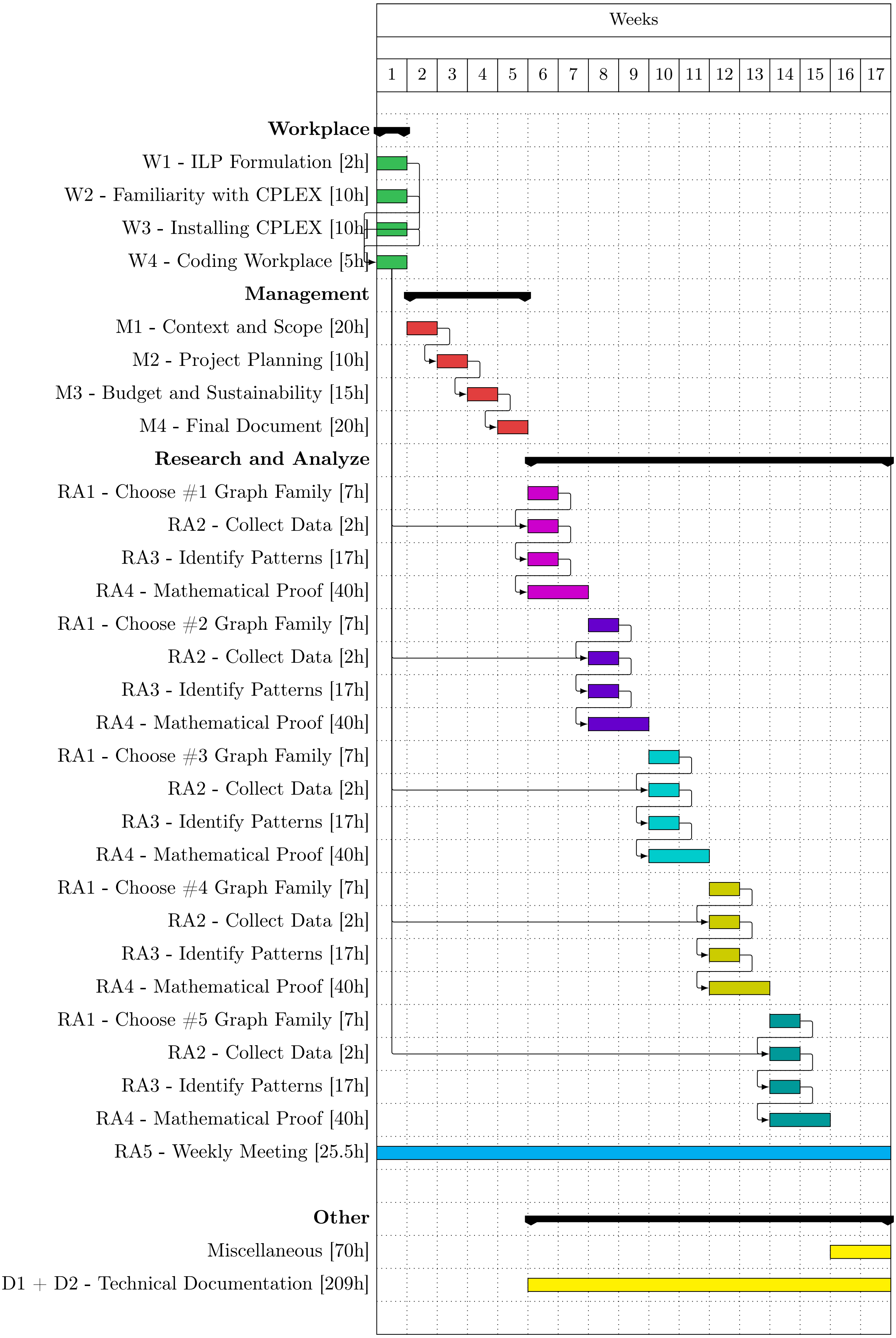


Figure 6: Self elaborated Gantt Diagram for this project, created using the ‘pgfgantt‘ package in LaTeX

8 Risk Management

As mentioned in the previous section on obstacles, the main challenges I anticipate are related to my skill level and the reliability of the data and knowledge sources. There is also a possibility of encountering technical issues with my online editor, the risk of losing my work, or facing problems with my laptop. However, all of these potential problems can be mitigated by implementing a regular backup strategy for the project, which involves creating backups every week. In this section, I will primarily focus on addressing the two main obstacles.

8.1 Skill Level Obstacle

As I am not a specialized researcher, there is a risk that I may overlook important aspects or lack specific knowledge that is beyond my current level of expertise. Fortunately, I am fortunate to have Antoni Lozano as my director, which provides a valuable resource to overcome skill-related obstacles. I plan to meet with him on a weekly basis, and this regular interaction will help me address any skill-related challenges that may arise.

If we fail to identify any pattern there's no possibility to aim a mathematical proof, so I will reschedule subsequent tasks to start one week earlier, and if necessary, conduct Research and Analyse tasks for a new Graph Family taking one week from Miscellaneous time. This could delay the project one week long (if I use all the Miscellaneous task time). Also, we can see a pattern in data, but we are not able to provide a mathematical proof; In that case I will report our *failed* proof in the thesis to advise future researcher to not follow this step.

8.2 Data Source Obstacle

The data used in this research is provided by [15], which is sourced from Brendan McKay[16]. I have confidence in Professor McKay's extensive knowledge in combinatorial data. However, it is essential to acknowledge that human errors can occasionally occur in any database. In case I detect any discrepancies or errors in the data, I will promptly contact Professor McKay through his online email, as provided by the *Australian National University* [16]. This can halt the project until Professor McKay's replies to me. As an alternative, I would create my personal database for the family I will study. This implies creating a new task **RA1.5** called *creating instances* with a duration of 15 hours (since maybe I have to create millions of graphs instances with my PC) and **Research and Analyse** tasks would be as follows:

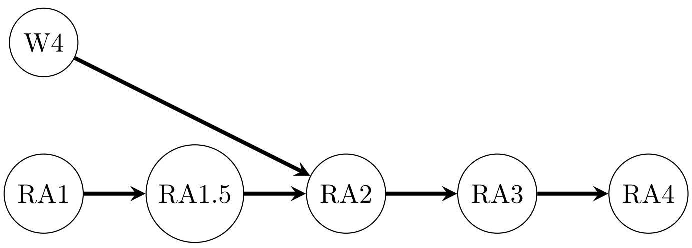


Figure 7: Non-Expected Task Dependencies for Research and Analyse, self elaborated

9 Budget

In the following sections, I will detail various costs associated with the development of the project. Considering it is a theoretical project, no production or commercialization costs will be taken into account. Therefore, I will only consider costs related to human resources, design, planning, implementation, and materials. Notice that calculating exact costs will be difficult, because of that I will make certain assumptions and approximations during the following subsections.

9.1 Hardware

The only hardware I will require is my laptop (an Asus Zenbook UM425UAZ[17] with a price of 878€), which is where I will conduct, code all my experiments and find research information. The amortization and useful years will be provided in the following **Table 5**, calculating the amortization. Considering I work an average of 40 hours per week:

$$Amortization = CostComputer \cdot \frac{1\text{useful life}}{4years} \cdot \frac{1year}{52weeks} \cdot \frac{1week}{40hours} \cdot DedicatedHours$$

In our case:

$$Amortization = 878€ \cdot \frac{1\text{useful life}}{4years} \cdot \frac{1year}{52weeks} \cdot \frac{1week}{40hours} \cdot 726.5 = 76.66€$$

Device	Useful Years	Price	Amortization
Asus Zenbook UM425UAZ	4 years	878€	76.66€

Table 5: Device Cost and Amortization, self elaborated

9.2 Software

Obviously, I will require various software tools to accomplish this thesis. In the preceding sections, I discussed CPLEX, Obsidian, GitHub, and LaTeX, but naturally, I will be running this software on an operating system (specifically, Ubuntu OS). Additionally, I'll need a suitable code editor (in my case, neovim, which I have been using since my second year of the degree), and a compiler for, later, executing my C++ code (I've chosen g++ for its excellent performance throughout my degree).

The only non-open-source (by definition an *OpenSource* tool is free) software I will use are the ILP solver CPLEX, GitHub and Obsidian. Fortunately, CPLEX offers a 1-year free license for academic purposes, GitHub provides free repository hosting and Obsidian is a free software. Therefore, no additional costs will be considered.

Product	Cost	Amortization
LaTeX	0€	0€
Ubuntu OS	0€	0€
neovim	0€	0€
g++	0€	0€
GitHub	0€	0€
Obsidian	0€	0€
CPLEX	0€	0€
Total	**0€**	**0€**

Table 6: Product Cost and Amortization, self elaborated

9.3 Human Resources

The project will be developed by a single person assuming various roles. Consequently, I will calculate the corresponding salary for each role. To determine the hourly salaries in Euros (€) for these roles, I will refer to the annual salary data from *GlassDoor*[18]. With the assumption of an average workday of 8 hours and 5 working days per week, I will calculate the hourly rates for each role based on their respective annual salaries. After calculating the total salary I will multiply it by 1.35 to pay the Social Security.

$$OneHourSalary = Salary \cdot \frac{1year}{12months} \cdot \frac{1month}{20\text{days working}} \cdot \frac{1day}{8hours}$$

Corresponding Rol	Salary	Expected Hours	Total	Total + Social Security
Researcher	22.31€/hour[19]	566.5 hours	12638,615€	17062.12€
Developer	18.64€/hour[20]	25 hours	466€	629.10€
Project Manager	22.89€/hour[21]	65 hours	1487.85€	2008.60€
Total	-	**656.5 hours**	14592,465€	**19699,82€**

Table 7: Human Resources Cost without Social Secutiry, self elaborated

Antoni Lozano and Paola Lorenza will provide assistance with the research and project management aspects of the thesis respective. However, since I do not have precise information regarding the amount of time Antoni and Paola will allocate to this thesis, I will assume their contributions are voluntary. They are affiliated with the university, and UPC is responsible for compensating their efforts, even if they choose not to provide direct support to my thesis. Notice that other projects with other conditions have to consider their respective support with economical compensation.

At the end of the thesis I will add the Miscellaneous time because for now it's still unknown the use of it.

9.4 Indirect Costs

- **Electricity:** The official Asus page for my laptop[22] claims a battery life of up to 16 hours and specifies a battery capacity of 67Wh. Assuming I will need 80W for recharging my laptop (I'll make an assumption that my laptop needs 67W, and I lose 15% recharging because *Joule Effect*, so more or less I will need 80W for recharging my laptop), we can calculate the cost as follows: First I have to calculate how many hours I will spend recharging my laptop during the thesis

$$726.5hours \cdot \frac{1recharge}{16hours} = 45.45 \approx 46 \text{ recharges}$$

With this information and assuming I can charge my laptop in 1.5h (because[22] offers fast-charge 49mins for 60% of charge and I'll consider the rest of the charge can be done in 40mins) I'll calculate the total of hours and power I'll be using for recharging the laptop

$$46 \text{ recharges} \cdot \frac{1.5 \text{ hours}}{1 \text{ recharge}} = 69hours$$

$$46 \text{ recharge} \cdot \frac{80\text{W}}{1 \text{ recharge}} \cdot \frac{1\text{kW}}{1000\text{W}} = 3.68kW$$

Concluding that

$$69h \cdot 3.68kW = 253.92kWh$$

Electricity price[23] at October 1, 2023 has a price of (mean) 0.10€/kWh. Concluding that the total price of the electric use is:

$$253.92kWh \cdot \frac{0.10€}{1kWh} = 25.40€$$

This is an approximation, supposing I would only need my laptop for the thesis. But this is not true, as I would need it for other projects related to the university. Since these projects are separate from the thesis, I will assume responsibility for all non-project-related electricity costs.

- **Internet:** I pay 27€ monthly for the internet. The project is expected to long five months, making a total of 108€. However, I will not use only internet for this project, so I have to allocate a proportional part of the project. Considering 726.5 hours for this project (calculated in 4), I can calculate the proportional part spent on the internet as:

$$CostInternet = \frac{27€}{1month} \cdot \frac{1month}{30days} \cdot \frac{1day}{24hours} \cdot 726.5 \text{ hours}$$

Making a total of **27.24€**.

If Internet and/or Electricity fails I could always go to *Campus Nord* and/or different libraries with free *Wi-Fi* and electricity provided by *Generalitat de Catalunya* so no extra cost I will assume from this part. No other costs related to workspace are assumed because this project will be developed at UPC and at my house. Fortunately, I live in Barcelona in a house without any rent or mortgage, and I always walk to *Campus Nord*. Therefore, I won't incur any additional expenses related to the workspace.

Resource	**Price**
Electricity	25.40€
Internet	27.24€
Total	52.64€

Table 8: Indirect Costs, self elaborated

9.5 Contingency

Part of the budget must be reserved to ensure that the project can proceed securely, especially from an economic perspective, in case any of the unfortunate events mentioned above occur. I will allocate 15% as the standard percentage to deal with future events.

Source	**Price (€)**	**Percentage**	**Cost (€)**
Hardware	76.66	15%	11.50€
Human Resource	19699.82	15%	2954.97€
Indirect Costs	52.64	15%	7.90€
Total	-	-	**2974.37€**

Table 9: Contingency, self elaborated

9.6 Unexpected costs

An unexpected event could always arise so I'll be saving part of the budget to extra hours for the different roles I will take during the thesis. The most exposed role to do extra hours is the researcher one (maybe for my skill level, maybe because mathematics are an abstract science and I'll have to take different approach to solve one problem, maybe because I need to find more complex knowledge to solve one problem...). Developer role is the safest one to suffer extra hours because doing a *backup* (in our case: uploading the project to GitHub) of my current process would be sufficient to reduce the probability of more work. Project Manager would also have the same probability of an extra working if the researcher thinks he needs more time. In summarize:

Position	**Extra Hours**	**Probability**	**Cost**	**Cost with Social Security**
Researcher	40 hours	40%	892.40€	1204.74€
Developer	5 hours	5%	93.20€	125.82€
Project Manager	10 hours	40%	228.90€	309.02€
Total	**55 hours**	-	-	**1639.58€**

Table 10: Unexpected Human Resources Cost, self elaborated

I considered that, maybe, I get stacked in an important mathematical proof and me and my director decided to dedicate more effort to it and if the researcher role doesn't fit the schedule the project manager would have to re-schedule all the different tasks, that's the reason why they have the same probability.

Also, I have to consider what if my computer fails. I will immediately contact with a repair service. Lucky for me, I bought this computer less than a year ago, so I will expect a 5% of chances to repair it.

Resource	**Unexpected Event**	**Price**	**Probability**
Asus Zenbook UM425UAZ	Repair Service	60€	5%

Table 11: Unexpected Events, self elaborated

I will include an additional 60€ in the budget, **making a total of 1699.58€** for the unexpected costs, as a preventive measure to address this situation. It is highly likely that I won't need to use this amount, and if that is the case, I can always reallocate it to the primary funding source.

9.7 Final Budget

In summary, we will consolidate all the budget elements we've reviewed up to this point. We'll then combine these elements to determine the project's total budget. This will give us a comprehensive understanding of the project's overall expenses and how the budget is distributed among its various aspects.

Source	**Cost**
Hardware	76.66€
Software	0€
Human Resources	19699.82€
Indirect Costs	52.64€
Contingency	2974.37€
Unexpected Costs	1699.58€
Total	24503.07€

Table 12: Final Budget, self elaborated

9.8 Control Management

After finalizing our budget, it becomes imperative to maintain effective project management. To gauge our project's progress, we rely on two pivotal indicators that highlight the deviation between our estimations and the actual results.

$$\text{Cost Deviation (CD)} = C_e - C_r$$

$$\text{Efficiency Deviation (ED)} = T_e - T_r$$

Here, the subindex **r** represents *real*, and **e** represents *expected* values.

Deviation can be calculated for each completed task, and the results are crucial for subsequent tasks. By calculating the difference between the real and expected cost or time (the *deviation*), we gain insights into how each task deviates from the initial planning. The magnitude of the *deviation* indicates the degree of difference between the actual and expected values. If the absolute value of the *deviation* is substantial, it signifies the need for project decisions, such as re-scheduling. The sign of the number informs us whether we required less cost/time or more.

10 Sustainability Report

In recent years, governments have recognized the critical importance of integrating sustainability into their present and future projects. For instance, according to [24], 79% of consumers are changing their preferences based on environmental impact, and approximately 70% of companies are either already using or transitioning to techniques that prioritize the sustainability of the planet. Our project must also acknowledge this significant aspect for the collective well-being of the global population. Despite my limited prior consideration of sustainability and my relatively modest knowledge in this area, I am committed to doing my utmost to address this important facet within this section.

10.1 Sustainability Dimension

10.1.1 Economic Dimension

The total economic cost related to this project amounts to **24503.07€**, as shown in **Table 12**.

Among all these costs, the most substantial expense is attributed to Human Resources. This is an inherent aspect of the project, given its nature of dealing with abstract concepts that have broader applications. Hardware and Indirect Costs are also essential components, which are common in most Computer Science theses. I believe that no unnecessary costs have been added to the budget, since the specific hardware is required to conduct experiments, and I have utilized entirely free software.

If these topics become state-of-the-art research areas, we may witness improvements in Graph Algorithms, leading to increased efficiency in computation times. This, in turn, could reduce the hardware costs associated with running these algorithms. The goal is to optimize computing processes, ultimately saving time, resources and therefore money.

10.1.2 Environmental Dimension

As demonstrated in **indirect costs** section, the laptop (and the whole project) consumes a total of 253.92kWh. According to a recent report by the Spanish Government [25], the carbon emission rate is 0.302 $\mathrm{kgCO_2/kWh}$, that gives us a producing of $76.68kgCO_2$. I take eco-friendly transportation by walking to *Campus Nord*, producing zero carbon emissions by myself. I think this is the best I can do for my ecological footprint related to this project, as my only CO_2 emission is the unique laptop I have in acquisition.

Our project could use experimental results from other sources and try to provide the theoretical part (reducing also our carbon footprint). The problem is, these sources don't exist, so we first have to create them. After doing the research, our theoretical/experimental results provided by this thesis and future research efforts could contribute to the development of faster algorithms for various graph-related calculations. These faster algorithms have the potential to save time and reduce electricity consumption, ultimately decreasing the carbon footprint.

10.1.3 Social Dimension

P vs NP (and computational complexity in general) has been my favourite topic ever since I first encountered it in my *Data Structures and Algorithms* class. NP-Complete problems hold a special fascination for me, perhaps because they form the core of decisional problems or because I've been eager to determine whether they are indeed the most challenging problems in Computer Science. Working with an NP-Complete problem presents a thrilling challenge and motivates me to acquire new theoretical knowledge.

In relation to the social field, it's evident that all the theoretical and practical knowledge generated by this thesis will have a positive impact, whether through future research endeavours or practical applications. Future research can stimulate further results and applications, considering that many real-world problems can be abstracted as graph problems. It's worth noting that all the information provided in this thesis is novel, and we believe it can inspire future research. Metric Dimension, being an NP-Complete problem, offers the potential for theoretical breakthroughs that could contribute to solving the P vs

NP question and developing faster algorithms in Graph Theory, ultimately improving people's lives. I sincerely hope that my results will indeed inspire future research in these areas.

10.2 Sustainability Matrix

Summarizing all this section, I will provide the Sustainability Matrix.

	PPP	**Useful Life**	**Risks**
Economics	24503.07€	No benefits (research work)	No significant results
Environment	76.68Kg CO_2	Future effective applications	None
Social	8/10	7/10	None

Table 13: Sustainability Matrix, self elaborated

11 Updates

11.1 Updates respect to all the initial planning

Updates on the main focus of the thesis

The thesis began as expected, with a focus on experiments. However, as we observed values from the experiments, we identified an opportunity for theoretical exploration if we make less experiments than expected. Consequently, we shifted our emphasis towards theoretical work, which we considered more valuable from our perspective compared to a purely experimental approach (because theoretical work is more conclusive than experimental work, which, in this case, could be conjectures). Towards the end of the thesis we will still conduct experimental work, but fewer than expected due to the time constraints.

This decision aligns with the primary objective of the thesis, which is to contribute to the Metric Dimension problem.

Update on Tasks and Gantt

Firstly, I must mention that the initial planning was non-realistic, as I cannot predict how much time I will be focused on mathematical proofs or identifying patterns. Consequently, the thesis took less time than expected, as the main part of the thesis, theoretical work, took less time than anticipated.

Additionally, we decided to allocate more time to theoretical work than experimental work. As a result, I had to shift the experiments to the last month of December at the end of the thesis. A summary of the task management can be found in **Figure 8**. Additionally, a new task, E1, emerged with the corresponding requirements W5 and W6. Here is a brief overview of some changes in tasks.

- **E1 - ILP vs WMaxSAT**: Professor Carbonell[26] suggested comparing the performance of ILP solvers against WMaxSAT solvers, which have shown improvement in their performance in recent years. Consequently, I decided to conduct this experiment, and tasks W5 and W6 were created for this purpose (which were also suggestions from him). For this experiment, I spent a total of 20 hours.

- **W5 - Install Gurobi**: Similar to the CPLEX task, but this time it was easier. I encountered numerous issues during the installation of CPLEX, which were not present with Gurobi. This task only took 2 hours.

- **W6 - Becoming familiar with Gurobi and WMaxSAT Solvers**: I had to transform my ILP formulation into a Weighted Max-SAT formulation but this process was straightforward, and these tasks did not pose any problems. Additionally, in this task, I had to create a program to generate ILP and WMaxSAT instances on different files. Overall, the task took a total of 10 hours.

- **Miscellaneous**: I anticipated the possibility of new tasks arising and allocated time in the *Miscellaneous* category. The time spent on W5, W6, and E1 amounted to a total of 32 hours, which will be subtracted from the Miscellaneous time. The remaining 38 hours from the Miscellaneous category will be saved for reading the thesis and preparing for the defence.

- **RA5 - Weekly Meetings**: Antoni faced issues during the last two weeks of November, and we couldn't have our scheduled meetings. Additionally, there were two weeks where we decided not to meet because neither Antoni nor I had anything significant to discuss. So we spent less time on this task.

- **RA1 - Choosing Graph Family**: We wanted tournaments to be part of our thesis, so the first week after GEP deliveries, I focused on tournaments, removing the part of choosing a graph family. Apart from that, because we allocated more time to theoretical results, this task took less time than expected.
- **RA3 - Identifying Patterns / RA4 - Mathematical Proof**: Although we spent most of the time in the thesis doing these tasks, they took less time than expected.

Task Description	**Acronym**	**Duration**	**Tasks to be Completed Before**	**Requirements (if needed)**	**Human Resource**
Context and Scope	M1	20 hours	-	*Overleaf*, PC	GEP-T
Project Planning	M2	10 hours	M1	*Overleaf*, PC	GEP-T
Budget and Sustainability	M3	15 hours	M2	*Overleaf*, PC	GEP-T
Final Document	M4	20 hours	M3	*Overleaf*, PC	GEP-T
Creating a Correct ILP Formulation	W1	2 hours	-	-	-
Becoming Familiar with CPLEX API	W2	10 hours	-	CPLEX Documentation[14]	-
Installing CPLEX	W3	10 hours	-	PC I'm gonna use for this project	-
Coding the Workplace	W4	5 hours	W1,W2,W3	Code Editor, PC	-
Install Gurobi	W5	2 hours	-	PC I'm gonna use for this project	-
Becoming familiar with Gurobi and WMaxSAT solvers	W6	10 hours	-	-	-
ILP vs WMaxSAT	E1	20 hours	W5, W6	PC I'm gonna use for this project	-
Choosing a Graph Family	RA1	14 hours	-	Research sources	D
Collecting Data	RA2	4 hours	W4, RA1	PC with workplace installed	-
Identifying Patterns	RA3	58 hours	RA2	Data Access	D
Mathematical Proof	RA4	120 hours	RA3	-	D
Weekly Meetings	RA5	19.5 hours	-	-	D
Documenting Data	D1	44 hours	RA2	*Overleaf*, PC	-
Documenting Proof	D2	165 hours	RA4	*Overleaf*, PC	D
Miscellaneous	MISC	38 hours	-	-	-
Total: 586.5 hours					

Table 14: Updated summary of Project Tasks, self elaborated

With the following updates, I also have to take into account the Cost Deviation and Efficiency Deviation. The part of the efficiency deviation will be included in the final budget, and the efficiency deviation will be addressed in **Table 15**

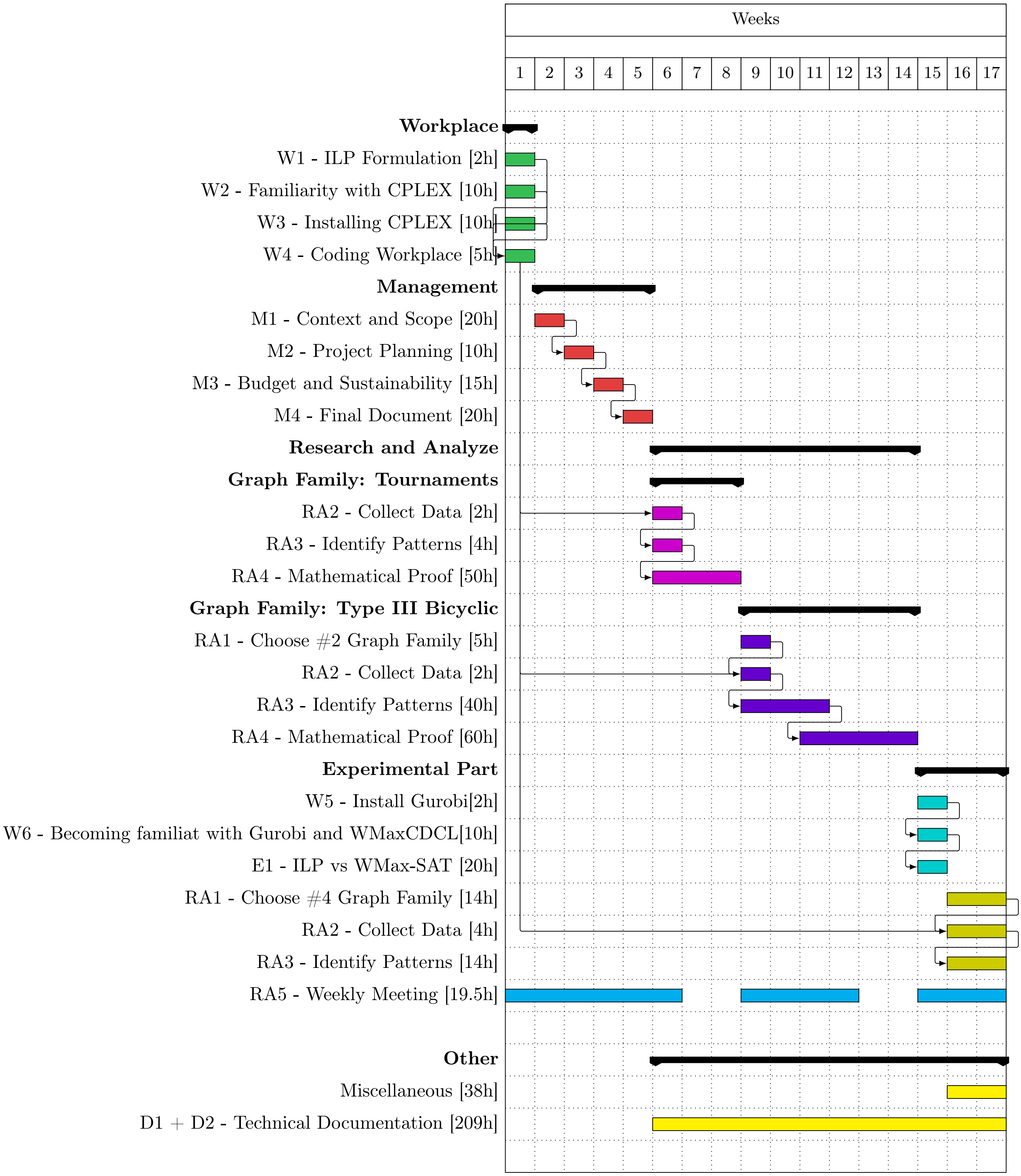


Figure 8: Updated Gantt Diagram for this project, created using the ‘pgfgantt‘ package in LaTeX

Task Description	Acronym	Expected Duration	Real Duration	Deviation
Context and Scope	M1	20 hours	20 hours	0 hours
Project Planning	M2	10 hours	10 hours	0 hours
Budget and Sustainability	M3	15 hours	15 hours	0 hours
Final Document	M4	20 hours	20 hours	0 hours
Creating a Correct ILP Formulation	W1	2 hours	2 hours	0 hours
Becoming Familiar with CPLEX API	W2	10 hours	10 hours	0 hours
Installing CPLEX	W3	10 hours	10 hours	0 hours
Coding the Workplace	W4	5 hours	5 hours	0 hours
Install Gurobi	W5	0 hours	2 hours	-2 hours
Becoming familiar with Gurobi and WMaxSAT solvers	W6	0 hours	10 hours	-10 hours
ILP vs WMaxSAT	E1	0 hours	20 hours	-20 hours
Choosing a Graph Family	RA1	35 hours	14 hours	21 hours
Collecting Data	RA2	10 hours	4 hours	6 hours
Identifying Patterns	RA3	85 hours	58 hours	27 hours
Mathematical Proof	RA4	200 hours	120 hours	80 hours
Weekly Meetings	RA5	25.5 hours	19.5 hours	6 hours
Documenting Data	D1	44 hours	44 hours	0 hours
Documenting Proof	D2	165 hours	165 hours	0 hours
Miscellaneous	MISC	70 hours	38 hours	32 hours

Table 15: Time Deviation for tasks

Updates on Software

During the development of the thesis, I utilized new software tools (Gurobi and *Open Source* WMaxSAT solvers), and I switched my operating system to Debian. Gurobi isn't free software, but they offer a free license for academic purposes, such as this thesis. On the other hand, all the WMaxSAT solvers where *Open-Source*, and Debian is a Linux distribution, which is also free. No additional costs from this part will be added to the budget. I switched to Debian because I had problems with my previous operating system and the reasons related to Gurobi and other solvers are explained in **Chapter 14**

Product	Cost	Amortization
LaTeX	0€	0€
Ubuntu OS	0€	0€
neovim	0€	0€
g++	0€	0€
GitHub	0€	0€
Obsidian	0€	0€
Gurobi	0€	0€
WMaxSAT Solvers	0€	0€
Debian	0€	0€
CPLEX	0€	0€
Total	**0€**	**0€**

Table 16: New Product Cost and Amortization, self elaborated

Update on Human Resources and Contingency

This project finally lasts less than expected so we have to pay few money to the corresponding rols

Corresponding Rol	Salary	Expected Hours	Total	Total + Social Security
Researcher	22.31€/hour[19]	423 hours	9437.13€	12740.13€
Developer	18.64€/hour[20]	41 hours	764.24€	1031.72€
Project Manager	22.89€/hour[21]	65 hours	1487.85€	2008.60€
Total	-	**529 hours**	**11689.22€**	**15780.45€**

Table 17: New Human Resources Cost without Social Secutiry, self elaborated

Source	Price (€)	Percentage	Cost (€)
Hardware	76.66	15%	11.50€
Human Resource	15780.45	15%	2367.07€
Indirect Costs	52.64	15%	7.90€
Total	-	-	**2386.47€**

Table 18: New Contingency, self elaborated

The miscellaneous time is reserved for preparing the defence of the thesis. Therefore, it will not be counted towards the budget as this task occurs after the conclusion of the thesis, and I will consider it as free work by myself.

Updates on Hardware

As I spent less time in the thesis I have to recalculate the amortization of the laptop.

$$Amortization = CostComputer \cdot \frac{1\text{useful life}}{4years} \cdot \frac{1year}{52weeks} \cdot \frac{1week}{40hours} \cdot DedicatedHours$$

In our case:

$$Amortization = 878€ \cdot \frac{1\text{useful life}}{4years} \cdot \frac{1year}{52weeks} \cdot \frac{1week}{40hours} \cdot 586.5 = 61.89€$$

Device	Useful Years	Price	Amortization
Asus Zenbook UM425UAZ	4 years	878€	61.89€

Table 19: New Device Cost and Amortization, self elaborated

Updates on Indirect Costs

As I spent finally 586.5 hours:

- Electricity:

$$586.5hours \cdot \frac{1recharge}{16hours} = 36.625 \approx 37 \text{ recharges}$$

With this information and assuming I can charge my laptop in 1.5h (because[22] offers fast-charge 49mins for 60% of charge and I'll consider the rest of the charge can be done in 40mins) I'll calculate the total of hours and power I'll be using for recharging the laptop

$$36 \text{ recharges} \cdot \frac{1.5 \text{ hours}}{1 \text{ recharge}} = 54 hours$$

$$36 \text{ recharge} \cdot \frac{80\text{W}}{1 \text{ recharge}} \cdot \frac{1\text{kW}}{1000\text{W}} = 2.88 kW$$

Concluding that

$$54h \cdot 2.88kW = 155.52kWh$$

Electricity price[23] at October 1, 2023 has a price of (mean) 0.10€/kWh. Concluding that the total price of the electric use is:

$$155.52kWh \cdot \frac{0.10€}{1kWh} = 15.55€$$

- **Internet:** I pay 27€ monthly for the internet. The project is expected to long five months, making a total of 108€. However, I will not use only internet for this project, so I have to allocate a proportional part of the project. Considering 726.5 hours for this project (calculated in 4), I can calculate the proportional part spent on the internet as:

$$CostInternet = \frac{27€}{1month} \cdot \frac{1month}{30days} \cdot \frac{1day}{24hours} \cdot 586.5 \text{ hours}$$

Making a total of **21.99€**

Resource	Price
Electricity	15.55€
Internet	21.99€
Total	37.54€

Table 20: New Indirect Costs, self elaborated

New Final Budget

Finally, I have to mention that no unexpected costs arose, so I decided to return this money to the budget. With the final summary, I will provide the expected cost, the real cost, and the deviation.

Source	Expected Cost	Real Cost	Cost Deviation
Hardware	76.66€	61.89€	14.77€
Software	0€	0€	0€
Human Resources	19699.82€	15780.45€	3919.37€
Indirect Costs	52.64€	37.54€	15.10€
Contingency	2974.37€	2386.47€	587.70€
Unexpected Costs	1699.58€	0€	1699.58€
Total	24503.07€	18266.35€	6236.72€

Table 21: New Final Budget, self elaborated

Updates on Sustainability

Update on Matrix

Finally I have to update the sustainability Matrix, as the project was shorter than expected.

	PPP	Useful Life	Risks
Economics	18266.35€	No benefits (research work)	No significant results
Environment	46.97Kg CO_2	Future effective applications	None
Social	8/10	7/10	None

Table 22: New Sustainability Matrix, self elaborated

11.1.1 New sustainability questions related to the project

As the project is reaching its final stages, one must assess whether it aligns with contemporary sustainability standards, aiming to minimize its environmental impact and maximize the sustainability of the project in general. One approach to evaluating sustainability involves addressing specific questions recommended during the GEP course. Here, I have selected some interesting questions that I could ask myself, taken from the GEP course.

Have you quantified the environmental impact of undertaking the project? What measures have you taken to reduce the impact? Have you quantified this reduction?
In fact, as the majority of the project involved theoretical work, I didn't use the computer as much as anticipated. Instead, I utilized paper and pen/pencil for doing proofs. However, I didn't keep track of the time spent without the PC, so I will count all the time spent on this thesis as time spent at the computer.

If you carried out the project again, could you use fewer resources?
Indeed, I needed the computer for the experimental part. However, for the aspects of the project that could be completed without the computer, such as proofs, I relied on traditional tools like pen and paper. Given the nature of the experimental work, it's likely that I couldn't have used fewer resources.

Is the expected cost similar to the final cost? Have you justified any differences (lessons learnt)?
My initial cost estimate was based on the anticipated time for various tasks. However, the project progressed more efficiently than expected, resulting in a final cost that is 6236.72€ lower than originally estimated.

Has undertaking this project led to meaningful reflections at the personal, professional or ethical level among the people involved?
Antoni Lozano, Mercè Mora, and Enric Rodríguez-Carbonell assisted me in different chapters of this thesis and I hold the belief that a general problem can be approached from various perspectives, each offering unique insights and approaches. Additionally, I attempted to compute the Metric Dimension for a Hypercube of dimension 18 (see **Chapter 15**), and Daniel Jiménez from the Architecture department attempted to assist me (because at this moment I was also taking a course on Graphic Cards and Accelerators and I thought that maybe it would be a good idea to perform this calculation using techniques learned in this course). Unfortunately, we were unable to

complete the task. This experience highlights how different fields of knowledge can contribute to the same problem— a reflection that I find valuable.

What resources do you estimate will be used during the useful life of the project? What will be the environmental impact of these resources?
All the theoretical knowledge we generated during the thesis will remain freely accessible to the mathematician community. This knowledge can be utilized for future improvements in efficiency, contributing to the reduction of environmental impact.

Who will benefit from the use of the project? Could any group be adversely affected by the project? To what extent?
This project has generated valuable theoretical knowledge, which is freely accessible and contributes positively to the scientific community. It does not have a direct negative impact on any community; instead, it provides valuable resources for further research and exploration.

11.2 Laws and Regulations

As this project can be considered a research endeavor conducted at *Facultat de Informàtica de Barcelona*, I must adhere to the Spanish laws and regulations related to researchers. In fact, the last update on BOE-A-2011-9617 was on 11/01/2023, and it specifies the following requirements for individuals involved in research at public universities in Spain, such as *Universitat Politècnica de Catalunya*. However, I cannot answer all the requirements as these laws are not specifically tailored to theses like mine. Therefore, I will only justify only when I can provide answers.

Avoid plagiarism and unauthorized appropriation of authorship of scientific or technological works from others.
During the thesis, I will acknowledge and mention the individuals who assisted me in their respective chapters and all the articles where I find the information. Although I didn't explicitly mention it earlier, Antoni Lozano helped me in ensuring the correctness of the proofs.

Inform the entities for which you provide services of all findings, discoveries, and results susceptible to legal protection and collaborate in the protection and transfer processes of your research results
Every week, I provided updates to my supervisor on my research progress, and activities, and he could inquire about my current project status at any time.

Disseminate the results of your research as indicated in the law, allowing for communication and transfer to other research, social, or technological contexts, and if applicable, for commercialization and valorization. Ensure that your results generate social value
This project is related to the P vs NP problem, one of the seven Millennium Prize Problems[1]. All the knowledge we can generate related to this problem would be valuable to society.

Ensure that your work is relevant to society
As I said in the previous question, this work is related to the P vs NP problem.

Align your research with the strategic objectives of the entities for which you provide services and obtain or collaborate in obtaining the necessary permissions and authorizations before starting your work

Universities also have researchers who are professors, and their work involves seeking new knowledge. In this case, the thesis is focused on generating new knowledge, aligning with one of the university's primary objectives. Additionally, the project had to be approved by the university, requiring permission from the head of the Computer Science Department.

Inform the entities for which you provide services or that finance or supervise your activity of possible delays and redefinitions in the research projects for which you are responsible, as well as the completion of projects or the need to abandon or suspend projects earlier than planned

I had to complete a project management course before starting the thesis, and we discussed a plan during that course that is added in one chapter of this thesis.

Use the name of the entities for which you provide services in your scientific activity in accordance with the internal regulations of those entities and the agreements, pacts, and conventions they subscribe to

As I said at the beginning of the thesis, this project was developed at *Facultat de Informàtica de Barcelona* in *Universitat Politècnica de Catalunya*

Theoretical Work

12 Graph Family: Tournaments

We say that $G = (V, E)$ is a tournament if G is a *Directed Graph* where there is exactly one arc between any pair of vertices. You can think a Tournament as an oriented complete graph. For instance, the following figure is an example of a Tournament with 5 vertices.

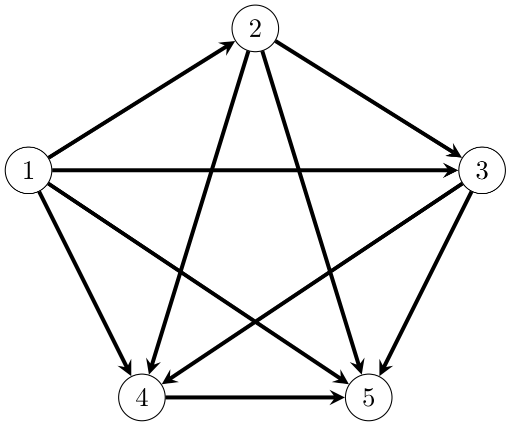


Figure 9: An example of a Tournament, self elaborated

n	$\beta(G) = 1$	$\beta(G) = 2$	$\beta(G) = 3$	$\beta(G) = 4$	$\beta(G) = 5$	$\beta(G) = 6$	$\beta(G) = 7$	$\beta(G) = 8$
2	1	0	0	0	0	0	0	0
3	2	0	0	0	0	0	0	0
4	2	2	0	0	0	0	0	0
5	2	10	0	0	0	0	0	0
6	2	49	5	0	0	0	0	0
7	2	348	106	0	0	0	0	0
8	2	2581	4286	11	0	0	0	0
9	2	16809	174188	537	0	0	0	0

Table 23: Metric Dimension values for tournaments of different order, self-elaborated

As I mentioned before, Antoni Lozano[5] has conducted work on the Metric Dimension of a specific graph family known as tournaments. In particular, Lozano studied a specific type of tournament called strong tournaments, where he proved that the optimal upper bound is exactly $\lfloor \frac{n}{2} \rfloor$. We thought that maybe an experimental approach to Tournaments could contribute to Lozano's work since, at today's day, no one except himself and Chartrand, Raines and Zhang[27] (where they proved that no constant positive integer k exists for bounding the metric dimension on Tournaments) has studied this topic related to this family.

So, the thesis begins with Tournaments, and I computed the Metric Dimension of all Tournaments up to order 9. The results can be found in **Table 23** and in the Official repository I created for this thesis[28]. This table revealed the two main results I'm going to present in this chapter.

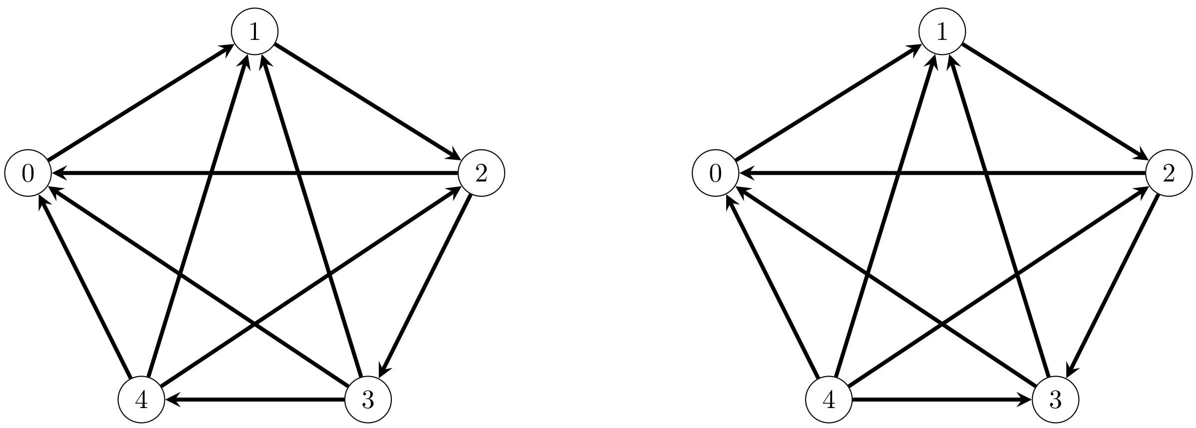
12.1 First Result: Characterisation on the Lower Bound of Tournaments

One curious result that appears in **Table 23** (where you can see in each cell how many tournaments with order n have $\beta(G) = k$) is related to the lower bound of the Metric Dimension: If $n = 2$, there's only one graph, and obviously, this would have $\beta(G) = 1$. Otherwise, only two non-isomorphic graphs fall into this category. This arises the idea of proving only two non-isomorphic tournaments have $\beta(G) = 1$ and, in fact, this is what we are going to prove. The idea behind the following proof is as follows: It is a well-known result in the Metric Dimension problem that if G is a simple graph, then $\beta(G) = 1 \iff G \cong P_n$. Since tournaments are complete directed graphs, we only have to orient the edges such that we have a path that traverses *almost* all vertices, and the starting vertex is the unique element in the resolving set.

12.1.1 Graphs with $\beta(G) = 1$

Let's define two graphs, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, with the following characteristics:

$$\begin{aligned} V &= \{0, 1, \ldots, n-1\} \\ E_1 &= \{(i, j) : j - i = 1 \text{ or } i - j \geq 2\} \\ E_2 &= E_1 - \{(n-2, n-1)\} \cup \{(n-1, n-2)\} \end{aligned}$$



(a) G_1 with $R = \{0\}$ as the Resolving Set

(b) G_2 with $R = \{0\}$ as the Resolving Set

Figure 10: An example of G_1 and G_2 with $n = 5$, self-elaborated

Both of these are tournament graphs, with $j - i = 1$ referred to as **the first rule**, and $i - j \geq 2$ as **the second rule**, you can see and example of each graph in **Figure 10**.

When examining both G_1 and G_2, they share a common path from vertex 0 to $n-2$, following the first rule. While tracing this path, at each step, you will encounter exactly one new, previously unvisited vertex, in line with the second rule. This prompts the question whether vertex 0 forms a resolving set, which is precisely what we aim to prove.

Now, focusing on G_1, we can follow a path from 0 to $n-1$. In this case, we will prove that $d(0, k) = k$ for every $k \in V$.

Lemma 12.1. *$d(0, k) = k$ for every $k \in V$ in G_1.*

Proof. Let's prove that $d(0,k) \leq k$ and $d(0,k) \geq k$.

- Clearly, $d(0,k) \leq k$ since there exists a path $0,1,\ldots,k-1,k$ by the first rule.
- Suppose we have a shortest path from 0 to k of the form $v_0, v_1, \ldots, v_t$ where $v_0 = 0$ and $v_t = k$. By the definition of G_1, we have:

  $$v_{i+1} \leq v_i + 1 \text{ for } 0 \leq i \leq t$$

  and therefore:

  $$v_t \leq v_{t-1} + 1 \leq v_{t-2} + 2 \leq \ldots \leq v_0 + t = t$$

  Since $v_t = k$, we conclude that $d(0,k) \geq k$.

□

By **Lemma 12.1**, we can conclude that vertex 0 forms a resolving set in G_1 since every vertex is distinguished by their distance in respect to 0.

Corollary 12.2. $\beta(G_1) = 1$.

The only difference between G_1 and G_2 is the edge that connects $n-1$ and $n-2$. The induced graph $G_1[\{0,\ldots,n-2\}]$ and $G_2[\{0,\ldots,n-2\}]$ are the same and we can apply **Lemma 12.1**. Let's prove that vertex 0 also forms a resolving set in G_2.

Proposition 1. *$\{0\}$ forms a Resolving Set in G_2.*

Proof. As we only *flip* the edge from vertex $n-1$ to $n-2$, it's easy to see that $G_1[\{0,\ldots,n-2\}] = G_2[\{0,\ldots,n-2\}]$. By **Lemma 12.1**, we can deduce that $d(0,k) = k$ for this induced graph.

Considering the last vertex, labelled as $n-1$, all the edges related to it have the form $(n-1,i)$ with $i \in V - \{n-1\}$. Since vertex $n-1$ cannot appear in a path starting at vertex 0 (because it's not reachable) we conclude that $\forall k \in \{0,\ldots,n-2\}, \quad d(0,k) = k$. Concluding that:

$$d(0,k) = \begin{cases} k & \text{if } k \in V - \{n-1\} \\ \infty & \text{if } k = n-1 \end{cases}$$

Concluding that $\{0\}$ also forms a Resolving Set since vertex $n-1$ could not change any distance related to other vertex. □

By **Proposition 1**, we conclude that $\beta(G_2) = 1$.

Corollary 12.3. $\beta(G_2) = 1$

$\beta(G_1)$ and $\beta(G_2)$ have Metric Dimension equal to one. If $n \geq 3$, $G_1 \ncong G_2$ because in G_2, the vertex $n-1$ is connected to all the other vertices, whereas in G_1, there is no vertex with this property. It's important to note that when $n=1$ or $n=2$, there is only one possible tournament, and in these cases, G_1 and G_2 become the same graph.

Corollary 12.4. *If $n \geq 3$ then $G_1 \ncong G_2$*

This implies that G_1 and G_2 are different tournaments with the same Metric Dimension.

12.1.2 Characterization of Tournaments with $\beta(G) = 1$

After introducing G_1 and G_2, consider a different tournament $G_x = (V_x, E_x)$, where $V_x = \{x_0, \dots, x_{n-1}\}$, with $R = \{x_0\}$ as a resolving set. Our goal is to prove that $G_x \cong G_1 \lor G_x \cong G_2$, providing a characterization of tournaments with a metric dimension equal to one. That's what we're going to do.

First off, let's prove that G_x has the same path as G_1 and G_2.

Lemma 12.5. *Let $G_x = (V_x, E_x)$ be a tournament with $R = \{x_0\}$ as the resolving set. Then there exists a path among at least $n-1$ vertices, with x_0 as the first vertex of the path.*

Proof. Let $W \subseteq V_x$ be the subset of vertices for which the distance between x_0 and $u \in W$ is finite. We first prove that $|W| = n$ or $|W| = n-1$

- By definition, for every $u \in V_x$, we have $0 \le d(x_0, u) \le n-1$ or $d(x_0, u) = \infty$. Because $\beta(G_x) = 1$, at most one vertex could exist with $d(x_0, u) = \infty$. Therefore, this would be the unique vertex not belonging to W. Consequently, we can conclude that $|W| = n-1$ or $|W| = n$, depending on whether, for some $u \in V_x$, $d(x_0, u) = \infty$ or not.

  With this help now let's prove for all $w \in W$, $\quad 0 \le d(x_0, w) \le |W| - 1$.

- By definition, for every $u \in V_x$, we have $0 \le d(x_0, u) \le n-1$. If $|W| = n$, then $W = V_x$, and according to the distance definition, $0 \le d(x_0, u) \le n-1 = |W|-1$. Otherwise, if $|W| = n-1$, it means that there exists one vertex $u \in V_x$ such that $d(x_0, u) = \infty$. This vertex cannot be reached starting at vertex x_0 meaning that there cannot exist a vertex $w \in W$ with $d(x_0, w) = n-1$ since this distance can be reachable only travelling through all vertices starting at x_0. This conclusion establishes that $0 \le d(x_0, u) \le n-2 = |W| - 1$.

Now, let's prove that the vertices in W form a path starting at x_0. Let x_i be the vertex with the distance $d(x_0, x_i) = i$ (we know this is unique because $\beta(G_x) = 1$). Consider x_i, x_{i+1} for $0 \le i \le n-3$. Knowing that $d(x_0, x_i) = i$ and $d(x_0, x_{i+1}) = i+1$, we need to prove that $x_i x_{i+1} \in E_x$. If $n = 2$, we are done. Otherwise, consider the shortest path $S = \{x_0, c_1, c_2, \dots, c_i, x_{i+1}\}$. We know that $d(x_0, x_{i+1}) = i+1$, and $d(x_0, x_{i+1}) = d(x_0, c_i) + 1$. Obviously, $d(x_0, c_i) = i$, and the only vertex with that property is x_i. Therefore, $x_i x_{i+1} \in E_x$, and W forms a path starting at node x_0. ☐

Theorem 12.6. *Let $G_x = (V_x, E_x)$ be a tournament. Then*

$$\beta(G_x) = 1 \iff G_x \cong G_1 \quad \lor \quad G_x \cong G_2$$

Proof. Let's prove both sides of the implication.

($\Leftarrow$:) By **Corollary 12.2** and **Corollary 12.3**, we can conclude it.

($\Rightarrow$:) Consider $V_x = \{x_0, \dots, x_{n-1}\}$ and let $R = \{x_0\}$ be the resolving set.

By **Lemma 12.5**, G_x possesses a path of at least $n-1$ vertices, starting at vertex x_0. Let W represent the subset of vertices constituting this path, and we will employ the notation $x_i \in W$ to denote the vertex for which $d(x_0, x_i) = i$. Our objective is to prove that all these vertices adhere to the **first** and **second** rules described in **Section 12.1.1**.

All x_i must adhere to the **second** rule. If not, let's assume that x_i does not comply with the **second** rule, meaning that there exists a vertex x_j such that $x_j \neq x_{i+1}$, $j > i+1$, and $x_i x_j \in E_x$. Then $d(x_0, x_{i+1}) = d(x_0, x_i) + 1$ and $d(x_0, x_j) = d(x_0, x_i) + 1$, concluding that $d(x_0, x_j) = d(x_0, x_{i+1})$. However, this cannot be possible because $R = \{x_0\}$ is a Resolving Set. Consequently, it is established that all vertices must adhere to this rule.

Now let's consider W.

- If $|W| = n$, you have a Hamiltonian path starting at node x_0 and, obviously, all vertices follow the **first rule**, and $G_x \cong G_1$.
- Otherwise, if $|W| = n-1$, implies the existence of one vertex $x_f \in V_x$ with $d(x_0, x_f) = \infty$. In this case, all the edges related to x_f must be of the form $x_f x_j \in E_x$. If there were an edge $x_k x_f \in E_x$, then $d(x_0, x_f) = d(x_0, x_k) + 1$, and $d(x_0, x_f) \neq \infty$. The first $n-1$ vertices from the path W follow the **first** rule due to the existence of the path. The two last vertices, x_{n-2} and x_f, are connected by the edge $x_f x_{n-2}$, and x_f is connected to all other vertices. This leads to the conclusion that $G_x \cong G_2$.

In both cases we conclude that $G_x \cong G_1 \vee G_x \cong G_2$. □

12.2 Second Result: Updating the Upper Bound

Another interesting result appearing in **Table 23** was the Upper Bound on Tournaments. This bound exactly follows what Lozano's[5] outlined in his article. In fact, I asked him if his proof applied to all tournaments without restriction, and he clarified that it was only for strongly connected tournaments. This raises the idea of examining his article to determine what needs to be adjusted to obtain an Upper Bound for all tournaments without restriction. In this case, the adjustment was straightforward, as we only had to change one proposition.

12.2.1 The Anchor Idea and extending a proposition

Let T be a tournament, and we say that S is an anchor if and only if $\forall u, v \in V(T), \exists w \in S : uw, wv \in E(T) \vee vw, wu \in E(T)$. Lozano proved that for every tournament T exists an anchor S such that $|S| \leq \lfloor n/2 \rfloor$, and he used this anchor as a resolving set to prove that $\beta(T) \leq \lfloor n/2 \rfloor$. The *issue* with his proof is that he didn't consider infinite distance as another valid distance. This is essentially (in his article this is referred in **Proposition 7**) what we are going to address and rectify.

Proposition 2. *Every anchor in a tournament is a resolving set.*

Proof. Let S be an anchor in a tournament T. We need to prove that $\forall u, v \in V(T)$ with $u \neq v, \exists w \in S : d(w, u) \neq d(w, v)$. If $u \in S$, then $d(u, u) = 0$ and $d(u, v) \neq 0$ because $u \neq v$, concluding that u resolves the pair u, v. Otherwise, consider $u, v \in V(T) - S$. By the definition of anchors, $\forall u, v \in V(T) - S, \exists w \in S : uw, wv \in E(T) \vee vw, wu \in E(T)$. Consider the case $uw, wv \in E(T)$, then $d(w, v) = 1$ and $d(w, u)$ would be different from one since, if not, $wu \in E(T)$ and both arcs will exist... But this cannot happen by the definition of tournaments. Concluding that $d(w, v) = 1$ and $d(w, u) \neq 1$, and w would distinguish these two vertices since $d(w, u) \neq d(w, v)$. Symmetric reasoning could be applied to the case when $vw, wu \in E$, concluding that $d(w, u) = 1$ and $d(w, v) \neq 1$, implying $d(w, u) \neq d(w, v)$ and w resolves u, v. □

13 Graph Family: Bicyclic graphs

A simple connected graph $G = (V, E)$ with $|V(G)| = n$ is said to be bicyclic if $|E(G)| = n + 1$. A well-known result in graph theory is that if T is a tree, then $|E(T)| = n - 1$, so a bicyclic graph can be seen as a tree with two additional edges. Khan et al. [29] studied the Metric Dimension on bicyclic graphs without vertices of degree one, categorizing them into three types:

- Bicyclic graph of Type I: Two disjoint cycles C_n and C_m where those cycles share only one common vertex.
- Bicyclic graph of Type II: Two disjoint cycles C_n and C_m joined by a P_r with $r \geq 1$, connecting any vertex of C_n to any vertex of C_m.
- Bicyclic graph of Type III: Three disjoint paths P_r, P_s, P_t and two vertices u, v that connect the beginning and the end of the paths.

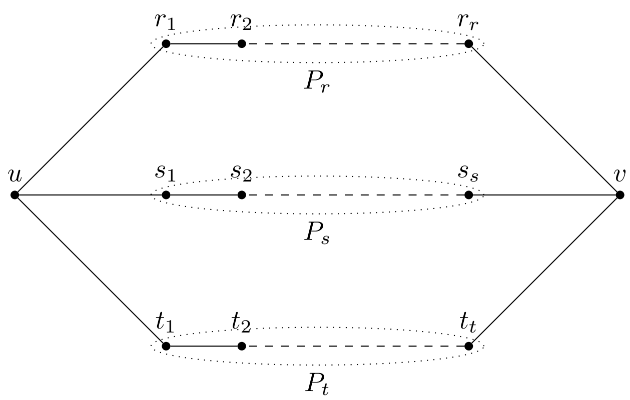


Figure 11: Type III Bicyclic graph, self-elaborated

In [29], the authors studied the Metric Dimension of Type I and II bicyclic graphs, providing a proof where the lower and upper bounds of these graphs are 2 and 3. They presented an open problem, urging researchers to explore the Metric Dimension on Type III bicyclic graphs to establish a proof of a constant Metric Dimension. My initial impression of this open problem was that I could find a proof, or at least offer experimental insights to future researchers, given my initial thoughts that they share a symmetry between paths.

I initially explored various techniques to identify potential subcases for proving this graph. I began by following a Hamiltonian Path, breaking these paths into two disjoint sets. I also attempted to alter path lengths to observe changes in the metric dimension (In this case, I observed that if paths have the same length or if two paths have the same length, and the third path is greater by two units in length, then the metric dimension is 3.), also I tried to create a Spanning Tree formed by the execution of the BFS algorithm and see if I can prove that these trees where different... However, no significant effects emerged (except I found cases where $\beta(G) = 3$). An experimental result on this graph yielded a Metric Dimension equal to 2 or 3, mirroring the findings for the other types of bicyclic graphs.

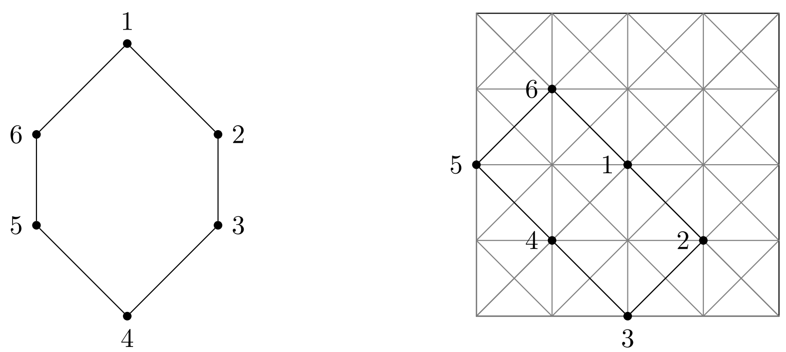
At this point in the thesis, Antoni Lozano was occupied, and I didn't want to disturb him. I recalled that Mercè Mora had studied (and is currently studying) the Metric Dimension many years ago[30], [31] and is currently teaching Graph Theory and Linear Algebra at *Facultat de Informàtica de Barcelona* in 2023. She had also collaborated with Antoni Lozano on the Antimagic labeling problem[32] in graph theory. Consequently, I decided to reach out to her, hoping she could assist me in deciphering the various cases for this graph.

She provided me with the idea of how to break down the cases with the corresponding subcases. With the subcases separated, I conducted random experiments to validate Mora's insights, and she was proven correct.

13.1 The idea behind Mora's thoughts

Mercè Mora gave me the following idea: Let G be a graph with $\beta(G) = 2$, then G can be *embedded* in a *strong product* of paths $P_n \boxtimes P_n$ of order n. Mora et al.[30] described the idea of embedding on the strong product of paths as follows: The strong product of two paths $P_n \boxtimes P_n$ has the Cartesian product $[0, n-1] \times [0, n-1]$ as the set of vertices, and two different vertices (i, j) and (i', j') are adjacent if and only if $|i - i'| \leq 1$ and $|j - j'| \leq 1$. Now we consider the representation of the graph $P_n \boxtimes P_n$ on the Cartesian plane, identifying the vertex (i, j) with the point on the plane (i, j). The idea of the embedding is considering the metric representation of a vertex v with respect to the resolving set $R = \{a, b\}$ as the coordinates on the plane. They recalled that it is straightforward to see that G is isomorphic to a subgraph of $P_n \boxtimes P_n$ by identifying the vertex $v \in V(G)$ to the vertex $(x, y) \in V(P_n \boxtimes P_n)$ such that $(x, y) = r(v|R) = (d(a, v), d(b, v))$. If two vertices x, y are adjacent in G, then $|d(x, u) - d(y, u)| \leq 1$ and $|d(x, v) - d(y, v)| \leq 1$ for some $u, v \in V(G)$ (recall that if two vertices x_1, x_2 are adyacent in G and $d(x_0, x_1) = d$ for some $x_0 \in V$ then $d(x_0, x_2) \in \{d-1, d, d+1\}$), hence $r(x|R)r(y|R)$ are adjacent in $P_n \boxtimes P_n$. The subgraph in $P_n \boxtimes P_n$ isomorphic to G is G^* with $V(G^*) = \{r(x|R) : x \in V(G)\}$ and $E(G^*) = \{r(x|R)r(y|R) : xy \in E(G)\}$.

Even though we can *embed* a graph in the strong product of graphs if $\beta(G) = 2$, Mercè recalled that it is a good idea to try to guess what the embedding would be like. With this information, we can deduce the resolving sets (that will be on the X axis and Y axis of the plane) and prove it. For instance, Figure 12 is an example of embedding C_6 in a $P_5 \boxtimes P_5$ with $R = \{5, 3\}$, and the Metric Representation of each vertex is $r(1|R) = (2, 2)$, $r(2|R) = (3, 1)$, $r(3|R) = (2, 0)$, $r(4|R) = (1, 1)$, $r(5|R) = (0, 2)$, and $r(6|R) = (1, 3)$.



(a) C_6 with $R = \{5, 3\}$ as the Resolving Set

(b) An embedding of C_6 in $P_5 \boxtimes P_5$

Figure 12: An example of embedding a graph, self-elaborated

13.2 Some cases of Type III bicyclic graphs

Although Mercè gave me the different subcases to prove, I couldn't prove them all. Due to the arrival of December and the last steps of the thesis, I decided not to spend more time on these proofs, as I wanted to focus on other aspects of this thesis.

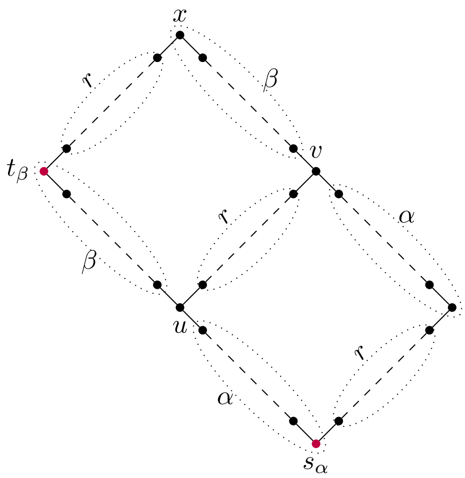
13.2.1 Subcases proved

Here we are going to prove that $R = \{a, b\}$ is a Resolving Set for different subcases. The idea of this proof is to consider two vertices $u, v \in V$, where $u \neq v$, and the condition $d(a, u) = d(a, v)$. Then, consider the first vertex where both paths $a - u$ and $a - v$ have been split. With this vertex, you can now *guess* the locations of the different vertices and observe that $d(b, u) \neq d(b, v)$.

Proposition 3. *Let G be a bicyclic graph of type III. Then $\beta(G) = 2$ for the following cases.*

Proof. Let P_r, P_s, P_t be the three paths of the graphs with length r, s, t. Let's proof by cases:

1. $1 \leq r < s \leq t$ and r, s, t same parity



$$\begin{cases} s = r + 2\alpha, \alpha \geq 1 \\ t = r + 2\beta, \beta \geq 1 \end{cases}.$$

Figure 13: First subcase of type III bicyclic graphs, self-elaborated

We aim to prove that $R = \{s_\alpha, t_\beta\}$ forms a Resolving Set. Let $i, j \in V$ be distinct vertices such that $d(s_\alpha, i) = d(s_\alpha, j)$, let's prove that $d(t_\beta, i) \neq d(t_\beta, j)$. Let C be the union of the vertices that forms paths $s_\alpha - u$, $u - t_\beta$, $t_\beta - x$ and C' be the vertices from the path $u - v$.

Clearly, the paths $s_\alpha - i$ and $s_\alpha - j$ could be *split* at s_α because $\deg(s_\alpha) = 2$. Otherwise, they have to be *split* at one vertex of w with $\deg(w) \geq 3$. The unique vertices with $\deg(w) \geq 3$ are u and v, and the paths could not be *split* at v because a shorter path would appear to one of those vertices without passing by v.

Now consider both scenarios, where paths have been split at s_α or v.

- s_α branch off the paths $s_\alpha - i$, $s_\alpha - j$: Suppose $j \notin C$ and $j \notin C'$, then $d(t_\beta, j) = r + \beta + 1 + \alpha + r + 1 - d(s_\alpha, j) = 2r + 2 + \alpha + \beta - d(s_\alpha, j)$ and if $i \in C$ then $d(t_\beta, i) = |d(t_\beta, s_\alpha) - d(s_\alpha, i)|$ or otherwise if $i \in C'$ then $d(t_\beta, i) = d(t_\beta, u) + d(u, i)$. Let's suppose $d(t_\beta, i) = d(t_\beta, j)$.

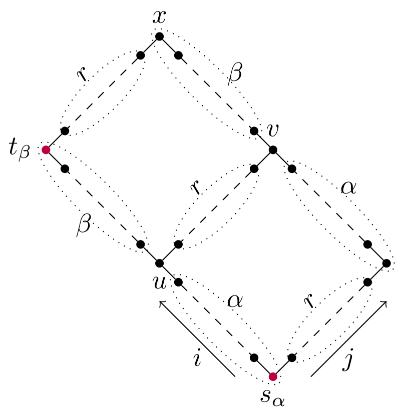


Figure 14: Paths $s_\alpha - i$ and $s_\alpha - j$, split at s_α

– If $d(t_\beta, i) = d(t_\beta, s_\alpha) - d(s_\alpha, i)$ (i is on path $s_\alpha - t_\beta$)

$$\begin{aligned} d(t_\beta, i) &= d(t_\beta, j) \\ \xrightarrow{\text{substitute}} d(t_\beta, s_\alpha) - d(s_\alpha, i) &= 2r + 2 + \alpha + \beta - d(s_\alpha, j) \\ \xrightarrow{d(t_\beta, s_\alpha) = \alpha + \beta} \alpha + \beta - d(s_\alpha, i) &= 2r + 2 + \alpha + \beta - d(s_\alpha, j) \\ \xrightarrow{d(s_\alpha, i) = d(s_\alpha, j) \text{ and simplify}} 2r + 2 &= 0 \end{aligned}$$

This is a contradiction because $2r + 2 > 0$.

– If $d(t_\beta, i) = d(s_\alpha, i) - d(t_\beta, s_\alpha)$ (i on path $t_\beta - x$)

$$\begin{aligned} d(t_\beta, i) &= d(t_\beta, j) \\ \xrightarrow{\text{substitute}} d(s_\alpha, i) - d(t_\beta, s_\alpha) &= 2r + 2 + \alpha + \beta - d(s_\alpha, j) \\ \xrightarrow{\text{reorder}} d(s_\alpha, i) + d(s_\alpha, j) &= 2r + 2 + \alpha + \beta + d(t_\beta, s_\alpha) \\ \xrightarrow{d = d(s_\alpha, i) = d(s_\alpha, j)} 2d &= 2r + 2 + 2\alpha + 2\beta \\ \xrightarrow{\div 2} d = d(s_\alpha, i) = d(s_\alpha, j) &= \alpha + \beta + r + 1. \end{aligned}$$

This would be possible if and only if $i = j = x$. But we suppose $i \neq j$, contradiction.

– If $d(t_\beta, i) = d(t_\beta, u) + d(u, i)$ (i on path $u - v$)

$$\begin{aligned} d(t_\beta, i) &= d(t_\beta, j) \\ \xrightarrow{\text{substitute}} d(t_\beta, u) + d(u, i) &= 2r + 2 + \alpha + \beta - d(s_\alpha, j) \\ \xrightarrow{d(t_\beta, u) = \beta} \beta + d(u, i) &= 2r + 2 + \alpha + \beta - d(s_\alpha, j) \\ \xrightarrow{\text{reorder and simplify}} d(u, i) + d(s_\alpha, j) &= 2r + 2 + \alpha \end{aligned}$$

We know $d(u,i) \leq r+1$ because i is on path $u-v$, so:

$$\begin{aligned} d(u,i) &= 2r+2+\alpha-d(s_\alpha,j) \\ \xrightarrow{d(u,i)\leq r+1} 2r+2+\alpha-d(s_\alpha,j) &\leq r+1 \\ \xrightarrow{\text{reorder}} d(s_\alpha,j) &\geq \alpha+r+1 \\ \xrightarrow{d(s_\alpha,i)=d(s_\alpha,j)} d(s_\alpha,i) &\geq \alpha+r+1 \end{aligned}$$

In this case i,j would *reconnect* their paths at v following the same path and meaning that $i=j$ but we suppose $i \neq j$, contradiction.

- Vertex u branches off the paths $s_\alpha - i$ and $s_\alpha - j$. Suppose the path $u-j$ passes inside the path $u-v$. If j is inside $u-v$, the shortest path from t_β is obtained without passing through vertex x (see **Figure 15**). Otherwise, if j is outside $u-v$, you have to pass through this vertex. In this case, we are going to consider these two subcases:

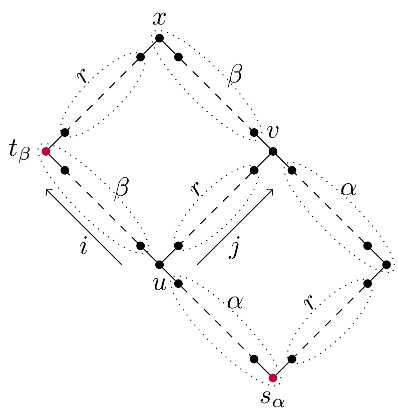


Figure 15: Paths $s_\alpha - i$ and $s_\alpha - j$, split at u

  – If path $t_\beta - j$ doesn't pass through x: $d(t_\beta,j) = d(t_\beta,u)+d(u,j)$ and $d(t_\beta,i) = |d(t_\beta,u) - d(u,i)|$. Let's suppose $d(t_\beta,i) = d(t_\beta,j)$
    * $d(t_\beta,i) = d(t_\beta,u) - d(u,i)$ (i on path $t_\beta - u$):

$$\begin{aligned} d(t_\beta,i) &= d(t_\beta,j) \\ \xrightarrow{\text{substitute}} d(t_\beta,u)-d(u,i) &= d(t_\beta,u)+d(u,j) \\ \xrightarrow{\text{simplify}} -d(u,i) &= d(u,j) \end{aligned}$$

    This is only possible if and only if $u=i=j$ but $i \neq j$, contradiction.
    * $d(t_\beta,i) = d(u,i) - d(t_\beta,u)$ (i on path $t_\beta - x$):

$$\begin{aligned} d(t_\beta,i) &= d(t_\beta,j) \\ \xrightarrow{\text{substitute}} d(u,i)-d(t_\beta,u) &= d(t_\beta,u)+d(u,j) \\ \xrightarrow{\text{simplify}} -d(t_\beta,u) &= d(t_\beta,u) \end{aligned}$$

    This is not possible because $d(t_\beta,u) = \beta \geq 1$, contradiction.

– If path $t_\beta - j$ pass through x: $d(t_\beta, j) = d(t_\beta, x) + d(x, u) - d(u, j)$ and $d(t_\beta, i) = |d(t_\beta, u) - d(u, i)|$. Let's suppose $d(t_\beta, i) = d(t_\beta, j)$:

  * $d(t_\beta, i) = d(t_\beta, u) - d(u, i)$ (i on path $t_\beta - u$):

$$
\begin{aligned}
&d(t_\beta, i) = d(t_\beta, j) \\
\xrightarrow{\text{substitute}}\ &d(t_\beta, u) - d(u, i) = d(t_\beta, x) + d(x, u) - d(u, j) \\
\xrightarrow{d(u,i)=d(u,j)\text{ and simplify}}\ &d(t_\beta, u) = d(t_\beta, x) + d(x, u) \\
\xrightarrow[d(x,u)=\beta+r+1]{d(t_\beta,u)=\beta \quad d(t_\beta,x)=r+1}\ &\beta = r + 1 + \beta + r + 1 \\
\xrightarrow{\text{simplify}}\ &0 = 2r + 2
\end{aligned}
$$

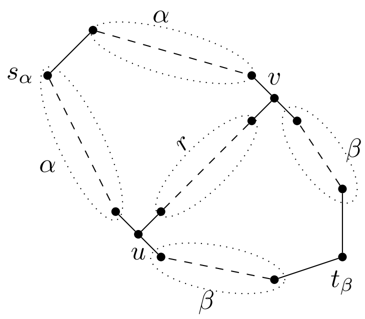
  Contradiction, because $2r + 2 \geq 1$.

  * $d(t_\beta, i) = d(u, i) - d(t_\beta, u)$ (i on path $t_\beta - x$):

$$
\begin{aligned}
&d(t_\beta, i) = d(t_\beta, j) \\
\xrightarrow{\text{substitute}}\ &d(u, i) - d(t_\beta, u) = d(t_\beta, x) + d(x, u) - d(u, j) \\
\xrightarrow{\text{reorder}}\ &d(u, i) + d(u, j) = d(t_\beta, x) + d(x, u) + d(t_\beta, u) \\
\xrightarrow[d(x,u)=\beta+r+1 \quad d(t_\beta,u)=\beta]{d(u,i)=d(u,j) \quad d(t_\beta,x)=r+1}\ &2d(u, i) = r + 1 + \beta + r + 1 + \beta = 2r + 2\beta + 2 \\
\xrightarrow{\div 2}\ &d(u, i) = d(u, j) = r + \beta + 1.
\end{aligned}
$$

  But this is possible if and only if $i = j = x$. But $i \neq j$, contradiction.

2. $1 \leq r < s \leq t$ s, t different parity.



$$\begin{cases} s = 2\alpha, \alpha \geq 1 \\ t = 2\beta + 1, \beta \geq 1 \end{cases} .$$

Figure 16: Second case of bicyclic type III graph, self-elaborated

Without lose of generality suppose t is odd and s is even. Let t_β be the vertex at position $\beta + 1$ on path P_t and s_α be the vertex at position α on path P_s. Let's proof $R = \{t_\beta, s_\alpha\}$ forms a Resolving Set. Let $i, j \in V$ such that $i \neq j$ and $d(t_\beta, i) = d(t_\beta, j)$. Let's proof $d(s_\alpha, i) \neq d(s_\alpha, j)$. Let $s_\alpha - i$, $s_\alpha - j$ be the paths from s_α to i, j and let w be the vertex where both paths have been split. Obviously $w = t_\beta$ or $w = u$ or $w = v$. Let's consider the following cases:

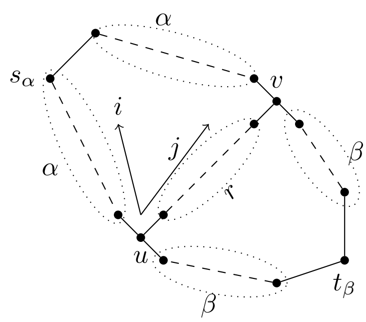


Figure 17: Path $t_\beta - i$ and $t_\beta - j$ split at u, self-elaborated

- $w = u$ or $w = v$: Let's consider $w = u$ and a symmetric reasoning would be valid for $w = v$. Suppose the path $u - j$ passes inside the path $u - v$. Then $d(s_\alpha, i) = |d(s_\alpha, u) - d(u, i)|$ and $d(s_\alpha, j)$ could take three possible value:

$$\begin{cases} d(s_\alpha, j) = d(s_\alpha, u) + d(u, j) \rightarrow \ u \text{ "nearly" } j \text{ and } j \text{ in path } u - v \\ d(s_\alpha, j) = d(s_\alpha, v) + d(v, j) \rightarrow \ v \text{ "nearly" } j \text{ and } j \text{ in path } u - v \\ d(s_\alpha, j) = d(s_\alpha, v) - d(v, j) \rightarrow \ v \text{ in path } s_\alpha - v \end{cases}$$

Now let's compare the three cases

- $d(s_\alpha, j) = d(s_\alpha, u) + d(u, j)$ and $d(s_\alpha, i) = d(s_\alpha, u) - d(u, i)$ (i on path $u - s_\alpha$):

$$\begin{aligned} d(s_\alpha, i) &= d(s_\alpha, j) \\ \xrightarrow{\text{substitute}} d(s_\alpha, u) - d(u, i) &= d(s_\alpha, u) + d(u, j) \\ \xrightarrow{\text{simplify}} -d(u, i) &= d(u, j) \end{aligned}$$

This would be possible if and only if $i = j = u$. But we suppose $i \neq j$, contradiction.

– $d(s_\alpha, j) = d(s_\alpha, u) + d(u, j)$ and $d(s_\alpha, i) = d(u, i) - d(s_\alpha, u)$ (i on path $v - s_\alpha$):

$$\begin{aligned} & d(s_\alpha, i) = d(s_\alpha, j) \\ \xrightarrow{\text{substitute}}\ & d(u, i) - d(s_\alpha, u) = d(s_\alpha, u) + d(u, j) \\ \xrightarrow{\text{reorder}}\ & 2d(s_\alpha, u) = d(u, i) - d(u, j) \\ \xrightarrow{d(u,i)=d(u,j)}\ & d(s_\alpha, u) = 0 \end{aligned}$$

Contradiction because $d(s_\alpha, u) = \alpha \geq 1$

– $d(s_\alpha, j) = d(s_\alpha, v) + d(v, j)$ and $d(s_\alpha, i) = d(s_\alpha, u) - d(u, i)$:

$$\begin{aligned} & d(s_\alpha, i) = d(s_\alpha, j) \\ \xrightarrow{\text{substitute}}\ & d(s_\alpha, u) - d(u, i) = d(s_\alpha, v) + d(v, j) \\ \xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha}\ & \alpha - d(u, i) = \alpha + 1 + d(v, j) \\ \xrightarrow{\text{simplify and reorder}}\ & -d(u, i) - d(v, j) = 1 \\ \xrightarrow{\times -1}\ & d(u, i) + d(v, j) = -1 \end{aligned}$$

Contradiction.

– $d(s_\alpha, j) = d(s_\alpha, v) + d(v, j)$ and $d(s_\alpha, i) = d(u, i) - d(s_\alpha, u)$:

$$\begin{aligned} & d(s_\alpha, i) = d(s_\alpha, j) \\ \xrightarrow{\text{substitute}}\ & d(u, i) - d(s_\alpha, u) = d(s_\alpha, v) + d(v, j) \\ \xrightarrow[d(s_\alpha,u)=\alpha]{d(s_\alpha,v)=\alpha+1}\ & d(u, i) - \alpha = \alpha + 1 + d(v, j) \\ \xrightarrow{\text{reorder}}\ & d(u, i) = 2\alpha + 1 + d(v, j) \end{aligned}$$

Suppose the split exists. Then, starting from vertex i and traveling a distance of $2\alpha + 1$ would lead to vertex v. Subsequently, traveling a distance of $d(v, j)$ would reach vertex j. However, this leads to a contradiction because $i = j$, which contradicts the assumption that $i \neq j$.

– $d(s_\alpha, j) = d(s_\alpha, v) - d(v, j)$ and $d(s_\alpha, i) = d(s_\alpha, u) - d(u, i)$:

$$\begin{aligned} & d(s_\alpha, i) = d(s_\alpha, j) \\ \xrightarrow{\text{substitute}}\ & d(s_\alpha, u) - d(u, i) = d(s_\alpha, v) - d(v, j) \\ \xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha}\ & \alpha - d(u, i) = \alpha + 1 - d(v, j) \\ \xrightarrow{\text{reorder and simplify}}\ & d(u, i) = d(v, j) - 1 \\ \xrightarrow{d(v,j)=d(u,j)-d(u,v)}\ & d(u, i) = d(u, j) - d(u, v) - 1 \\ \xrightarrow{d(u,j)=d(v,j) \text{ and simplify}}\ & d(u, v) + 1 = 0 \end{aligned}$$

Contradiction because $d(u, v) + 1 \geq 1$

– $d(s_\alpha, j) = d(s_\alpha, v) - d(v, j)$ and $d(s_\alpha, i) = d(u, i) - d(s_\alpha, u)$:

$$\begin{aligned}
d(s_\alpha, i) &= d(s_\alpha, j) \\
\xrightarrow{\text{substitute}} d(u, i) - d(s_\alpha, u) &= d(s_\alpha, v) - d(v, j) \\
\xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha} d(u, i) - \alpha &= \alpha + 1 - d(v, j) \\
\xrightarrow{\text{reorder}} d(v, j) &= 2\alpha + 1 - d(u, i) \\
\xrightarrow{\text{substitute}} d(u, j) - d(u, v) &= 2\alpha + 1 - d(u, i) \\
\xrightarrow{\text{substitute}} d(u, i) + d(u, j) &= 2\alpha + 1 + d(u, v)
\end{aligned}$$

If we consider the cycle formed by P_r and P_s, then $d(u, i) + d(u, j)$ is equal to the length of the cycle. Half of this length would reach the same vertex in this case, concluding that $i = j$, contradiction.

- $w = t_\beta$: Then each distance could take only two possible values:

$$\begin{cases} d(s_\alpha, i) = d(s_\alpha, u) + d(u, i) \ (1) \\ d(s_\alpha, i) = d(s_\alpha, u) - d(u, i) \ (2) \end{cases} \quad \begin{cases} d(s_\alpha, j) = d(s_\alpha, v) + d(v, j) \ (3) \\ d(s_\alpha, j) = d(s_\alpha, v) - d(v, j) \ (4) \end{cases}$$

– $(1) = (3)$

$$\begin{aligned}
d(s_\alpha, i) &= d(s_\alpha, j) \\
\xrightarrow{\text{substitute}} d(s_\alpha, u) + d(u, i) &= d(s_\alpha, v) + d(v, j) \\
\xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha} \alpha + d(u, i) &= \alpha + 1 + d(v, j) \\
\xrightarrow{\text{simplify}} d(u, i) - d(v, j) &= 1
\end{aligned}$$

– $(1) = (4)$

$$\begin{aligned}
d(s_\alpha, i) &= d(s_\alpha, j) \\
\xrightarrow{\text{substitute}} d(s_\alpha, u) + d(u, i) &= d(s_\alpha, v) - d(v, j) \\
\xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha} \alpha + d(u, i) &= \alpha + 1 - d(v, j) \\
\xrightarrow{\text{simplify}} d(u, i) + d(v, j) &= 1
\end{aligned}$$

– $(2) = (3)$

$$\begin{aligned}
d(s_\alpha, i) &= d(s_\alpha, j) \\
\xrightarrow{\text{substitute}} d(s_\alpha, u) - d(u, i) &= d(s_\alpha, v) + d(v, j) \\
\xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha} \alpha - d(u, i) &= \alpha + 1 + d(v, j) \\
\xrightarrow{\text{simplify}} d(u, i) + d(v, j) &= -1
\end{aligned}$$

– (2) = (4)

$$\begin{aligned}
& d(s_\alpha, i) = d(s_\alpha, j) \\
\xrightarrow{\text{substitute}} \; & d(s_\alpha, u) - d(u, i) = d(s_\alpha, v) - d(v, j) \\
\xrightarrow[d(s_\alpha,v)=\alpha+1]{d(s_\alpha,u)=\alpha} \; & \alpha - d(u, i) = \alpha + 1 - d(v, j) \\
\xrightarrow{\text{simplify}} \; & d(u, i) + d(v, j) = 1
\end{aligned}$$

We know that if $d(t_\beta, i) = d(t_\beta, u) + d(u, i)$ then $d(t_\beta, j) = d(t_\beta, j) + d(v, j)$ or if $d(t_\beta, i) = d(t_\beta, u) - d(u, i)$ then $d(t_\beta, j) = d(t_\beta, j) - d(v, j)$. In both cases we conclude that $d(u, i) = d(v, j)$. Then:

* (1) = (3) is false because $d(u, i) - d(v, j) = 0$
* (1) = (4) and (2) = (4) is false because $d(u, i) + d(v, j) = 2d = 1$ is a contradiction because even $\neq$ odd.
* (2) = (3) is false because $d(u, i) + d(v, j) \geq 0$.

3. $r = 0,\ s, t \geq 1$

   Three subcases:

   (a) s, t even: Refer to the first case with $r = 0$ ($1 \leq r < s \leq t$ with same parity).

   (b) s, t odd/even: Refer to the second case with $r = 0$ ($1 \leq r < s \leq t$ with different parity).

   (c) s, t odd: Unproven case.

4. $s = r, t = r + k,\ k \geq 1$ and $k \neq 2$

   Consider the following subcases:

   (a) If r, t has different parity: Refer to second case.

   (b) r, t same parity and $t \neq r + 2$, (meaning that $t = r + 2 + 2\alpha,\ \alpha \geq 1$)

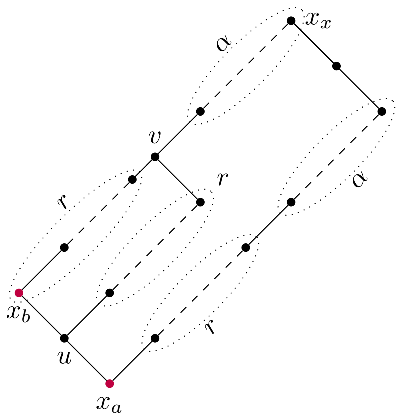


Figure 18: Fourth case of Bicyclic Type III graph, self-elaborated

Let's proof $R = \{x_a, x_b\}$ forms a Resolving Set. Let $i, j \in V$ such that $d(x_a, i) = d(x_a, j)$ and $i \neq j$. Let's proof $d(x_b, i) \neq d(x_b, j)$. Let w be the vertex where both path has been split. Obviously $w = u$ or $w = x_a$. Consider the following subcases:

- $w = u$. Then i, j are on the cycle formed by vertices $u - x_b - v - u$. Suppose path $u - j$ passes through vertex x_b and $u - i$ not. Then $d(x_b, i) = d(u, i) - 1$ and $d(x_b, j)$ could take three possible values.

$$\begin{cases} d(x_b, j) = d(x_b, u) + d(u, j) \ (1) \\ d(x_b, j) = d(x_b, v) + d(v, j) \ (2) \end{cases}$$

Let's suppose $d(x_b, i) = d(x_b, j)$:

  - $d(x_b, i) = (1)$

$$\begin{aligned} d(x_b, i) &= d(x_b, j) \\ \xrightarrow{\text{substitute}} d(u, i) - 1 &= d(x_b, u) + d(u, j) \\ \xrightarrow{d(u,i)=d(u,j) \text{ and simplify}} d(x_b, u) &= -1 \end{aligned}$$

Contradiction because by definition $d(x_b, u) \geq 0$

  - $d(x_b, i) = (2)$

$$\begin{aligned} d(x_b, i) &= d(x_b, j) \\ \xrightarrow{\text{substitute}} d(u, i) - 1 &= d(x_b, v) + d(v, j) \\ \xrightarrow{d(v,j)=d(v,u)-d(u,j)} d(u, i) - 1 &= d(x_b, v) + d(v, u) - d(u, j) \\ \xrightarrow{d=d(u,i)=d(u,j) \text{ and reorder}} 2d &= d(x_b, v) + d(v, u) + 1 \\ \xrightarrow[d(v,u)=r+1]{d(x_b,v)=r} 2d &= r + r + 1 + 1 \\ \xrightarrow{\div 2} d(u, i) = d(u, j) &= r + 1 \end{aligned}$$

This would be true if $i = j = v$ but we suppose $i \neq j$, contradiction.

- $w = x_a$: Then each distance could take only two possible values:

$$\begin{cases} d(x_b, j) = d(x_b, x_a) + d(x_a, j) \ (1) \\ d(x_b, j) = 2r + 2 + 2\alpha - d(x_a, j) \ (2) \end{cases} \qquad \begin{cases} d(x_b, i) = d(x_a, i) - d(x_b, x_a) \ (3) \\ d(x_b, i) = d(x_b, u) + d(u, i) \ (4) \end{cases}$$

Let's suppose $d(x_b, j) = d(x_b, i)$. Now consider the following cases:

  - $(1) = (3)$:

$$\begin{aligned} d(x_b, j) &= d(x_b, i) \\ \xrightarrow{\text{substitute}} d(x_b, x_a) + d(x_a, j) &= d(x_a, i) - d(x_b, x_a) \\ \xrightarrow{d(x_a,i)=d(x_a,j) \text{ and simplify}} d(x_b, x_a) &= -d(x_b, x_a) \\ \xrightarrow{d(x_b,x_a)=2} 2 &= -2 \end{aligned}$$

Contradiction.

– (1) = (4)

$$\begin{aligned}
&d(x_b, j) = d(x_b, i)\\
\xrightarrow{\text{substitute}}\ &d(x_b, x_a) + d(x_a, j) = d(x_b, u) + d(u, i)\\
\xrightarrow[d(x_b,u)=1]{d(x_b,x_a)=2}\ &2 + d(x_a, j) = 1 + d(u, i)\\
\xrightarrow{\text{reorder}}\ &d(x_a, j) = d(u, i) - 1
\end{aligned}$$

But in this case i is on path s and $d(x_a, i) = d(x_a, u) + d(u, i) = 1 + d(u, i)$ and we assume $d(x_a, i) = d(x_a, j)$ arriving a contradiction because $d(u, i) + 1 \neq d(u, i) - 1$.

– (2) = (3)

$$\begin{aligned}
&d(x_b, i) = d(x_b, j)\\
\xrightarrow{\text{substitute}}\ &2r + 2\alpha + 2 - d(x_a, j) = d(x_a, i) - d(x_b, x_a)\\
\xrightarrow{d(x_b,x_a)=2}\ &2r + 2\alpha + 2 - d(x_a, j) = d(x_a, i) - 2\\
\xrightarrow{\text{reorder}}\ &2r + 2\alpha + 4 = d(x_a, i) + d(x_a, j)\\
\xrightarrow{d=d(x_a,i)=d(x_a,j)}\ &2 \cdot (r + \alpha + 2) = 2d\\
\xrightarrow{\div 2}\ &d(x_a, i) = d(x_a, j) = r + \alpha + 2
\end{aligned}$$

This would be true if and only if $i = j = x_x$ but we suppose $i \neq j$, contradiction.

– (2) = (4)

$$\begin{aligned}
&d(x_b, j) = d(x_b, i)\\
\xrightarrow{\text{substitute}}\ &2r + 2\alpha + 2 - d(x_a, j) = d(x_b, u) + d(u, i)\\
\xrightarrow{d(x_b,u)=1}\ &2r + 2\alpha + 2 - d(x_a, j) = 1 + d(u, i)\\
\xrightarrow{\text{reorder}}\ &2r + 2\alpha + 1 - d(x_a, j) = d(u, i)
\end{aligned}$$

We know $d(u, i) \leq r + 1$ because i is on path S. Then:

$$\begin{aligned}
\xrightarrow{\text{apply inequality}}\ &d(u, i) = 2r + 2\alpha + 1 - d(x_a, j) \leq r + 1\\
\xrightarrow{\text{reorder}}\ &d(x_a, j) \geq r + 2\alpha
\end{aligned}$$

But we know $d(x_a, i) = d(x_a, u) + d(u, i) = 1 + d(u, i) \leq r + 2$ deducing that $r + 2 \leq d(x_a, i) \leq r + 2\alpha$ and concluding that $\alpha = 1$. In this case $i = v$ and $d(x_b, i) = d(x_a, i) - d(x_a, x_b)$ and we can refer to (2) $\neq$ (3).

□

13.2.2 Conjectures to the remaining cases

The remaining cases are:

- $r = s = t$ and $r \geq 1$
- $s = r, t = r + 2$

- $1 \leq r < s \leq t$: s and t have the same parity, while r has a different one.

For the first two cases, Mercè believes that these instances have $\beta(G) = 3$ (which aligns with the initial experimental findings of $\beta(G) = 3$). However, for the last case, she suggests that $\beta(G) = 2$. To validate these hypotheses, I conducted additional experiments. The results confirmed her beliefs. Encouraged by these findings, I propose the following conjecture:

Conjecture 1. *Let G be a type III bicyclic graph, then $\beta(G) = 2 \vee \beta(G) = 3$.*

It would be awesome if someone could take more time than I can on these subcases to complete the three different families provided by Khan et al.

If you are seeking a starting point to conclude the final subcases, I conducted experiments for the first 99 instances of the first two subcases. Although I also ran experiments for the last subcase, the results were not significant, except for consistently observing a Metric Dimension of two. Resolving Sets can be found on the Official repository[28]. Here, I will illustrate the pattern observed in the data for the Resolving Sets in these subcases:

Length of paths	Resolving Set
$r = s = t$ and $r \geq 2$	$R = \{2r, 5 + 3 \cdot (r-2), 6 + 3 \cdot (r-2)\}$
$r = r, s = r, t = r + 2$ and $r \geq 4$	$R = \{2r, 3r, 3r + 1\}$

Table 24: Conjectures of the Resolving Sets for the remaining subcases, self-elaborated

The exact Resolving Sets can be found on the Official Repository. Also, we labeled the vertices with the following numbers: Vertex u is 0, vertex $n - 1$ is v, vertices from P_r go from 1 to r, vertices from P_s go from $r + 1$ to $r + s$, and vertices from P_t go from $r + s + 1$ to $r + s + t$.

Experimental Work

14 ILP vs Weighted Max-SAT

14.1 The Integer Linear Programming formulation

During the thesis I used ILP solvers and for that I needed a robust ILP formulation, so it's crucial for me to provide a correct one.

In the context of Metric Dimension, two essential aspects demand our attention: the identification of the **smallest resolving set** and the assurance that no two different vertices have the same Metric Representation. This leads to the concept that **every pair of nodes is resolved by, at least, one node** (because there will be, at least, one differing distance in the metric representation). It's crucial to note that in this context, we define a vertex u as resolving a pair of vertices v and w if and only if $d(u, v) \neq d(u, w)$, where $d(u, v)$ represents the distance between vertices u and v (from u to v).

Notably, each vertex falls into one of two categories: it either belongs to the resolving set or does not. This binary distinction suggests a potential binary constraint. Utilizing this information, we can now present the following Integer Linear Programming (ILP) formulation. Let $G = (V, E)$ be a graph with $V = \{x_1, x_2, \ldots, x_n\}$. Then:

Objective Function:

$$\text{Minimize:} \quad \sum_{i=1}^{n} x_i$$

Subject to:

$$\text{Constraint 1:} \quad \sum_{v \in CDT(i,j)} v \geq 1 \text{ for every } 1 \leq i < j \leq n$$

$$\text{Constraint 2:} \quad x_i \in \{0, 1\} \quad \text{for every } 1 \leq i \leq n$$

Where:

$$CDT(i, j) = \{x_k \in V : \; d(x_k, x_i) \neq d(x_k, x_j)\}$$

$$x_i = 1 \iff x_i \in \text{Resolving Set}$$

With the binary restriction, we differentiate whether a vertex belongs to the resolving set or not. Minimizing the sum represents the *basis* of this graph. Now, we need to verify if this basis constitutes a resolving set or not.

Let $W = \{w_1, w_2, \ldots, w_n\}$ represent the resolving set of graph G, and let $x_i, x_j \in V, x_i \neq x_j$. We consider the values $r(x_i|W)$ and $r(x_j|W)$.

If x_i or x_j belongs to the resolving set, we are done since the problem itself asks about different metric representations of vertices not belonging to this set.

Now consider $x_i, x_j \in V - W$. When examining the constraint $\sum_{v \in CDT(i,j)} v \geq 1$, we are ensuring that there exists at least one vertex $w_k \in W$ such that $d(w_k, x_i) \neq d(w_k, x_j)$. This condition implies that the resolving values $r(x_i|W)$ and $r(x_j|W)$ must be different since, at least, one vertex distance on the metric representation will be different. Doing this for every pair of nodes concludes our proof.

Although there exists ILP formulations for this problem I decided to implement entirely one because all the ILP formulation I found always assume that graphs are connected and simple and it will be incorrect to use this ILP formulation for graphs like Tournaments as you may seen in **Chapter 12**

14.2 SAT to ILP and viceversa

One interesting fact I looked examining my Integer Linear Programming formulation was the constraints: If we forget about optimization doing a reduction from ILP to SAT in polynomial time and space is trivial:

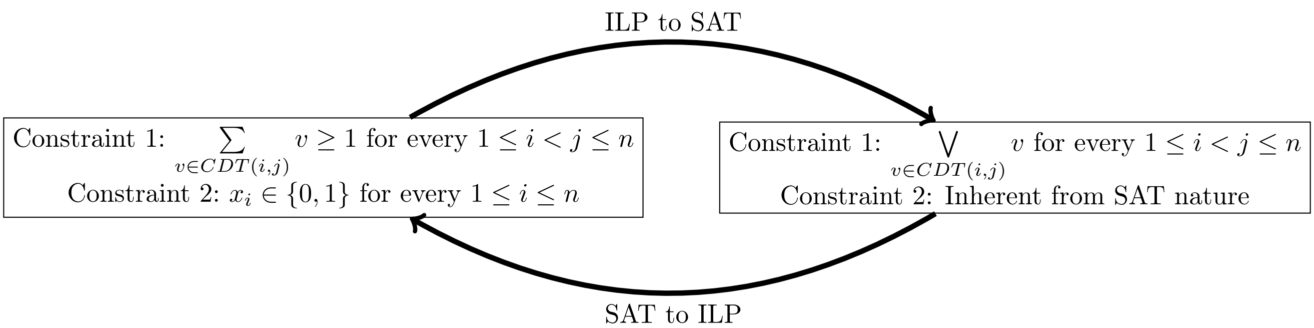


Figure 19: Transforming constraints from ILP to SAT and vice versa, self-elaborated

Finding a model in SAT and ILP (excluding the optimization part) is trivial, as we only need to traverse all the ***Constraint 1***, setting one variable to 1, since no negative literal appear on the formula. Since SAT is a more specific boolean problem than $0 - 1$ ILP (which is a subproblem of Integer Linear Programming), I initially considered that SAT solvers might exhibit better performance than ILP solvers, again, without considering the optimization part.

In my studies of *Logics in Information Technology*, I learned how to solve optimization problems using SAT by appending constraints at the end of the formula to select *atMostK* literals to true until the SAT formulation becomes unsatisfiable. However, this strategy proves inefficient with numerous constraints, as it requires spending time on deleting and adding constraints more than on solving.

Finally, seeking clarification, I contacted Enric Rodriguez-Carbonell[26], who is currently immersed in Boolean Satisfiability and its applications, including SAT solvers and integration with ILP solvers. He confirmed that these tasks appear quite manageable for a SAT solver, and the optimization method taught in *Logics in Information Technology* is not the most efficient. He recommended considering a variant of the SAT problem: The Weighted MAX-SAT problem, which has seen significant development in solver technology, akin to the progress in SAT and ILP.

14.3 Weighted MAX-SAT problem

The SAT problem involves determining the existence of a model, given a formula in conjunctive normal form (CNF), such that the formula is satisfiable. In contrast, the MAX-SAT problem aims to identify, within a CNF formula, the model that maximizes the number of satisfiable clauses. This problem seeks to optimize the satisfaction of clauses rather than solely checking for satisfiability.

An instance of the Weighted MAX-SAT problem consists of a formula $F = (C_1, w_1) \wedge (C_2, w_2) \wedge (C_3, w_3) \wedge \ldots \wedge (C_n, w_n)$, where C_i is a clause formed by the disjunction of literals, and w_i is a non-negative number representing the weight of the clause. The problem aims to minimize the total weight by satisfying clauses. It can be conceptualized as follows: "If you don't satisfy clause C_i, you incur a cost of w_i," and the goal is to minimize this cumulative weight through strategic satisfaction of clauses.

14.4 Weighted Max-SAT formulation

Lucky for us, we can trivially transform the constraints in ILP to clauses in SAT trivially (see **Figure 19**). The question arises when we have to assign values to the weights. Fortunately, we can assign infinite values to the weights to only accept valid solutions to the ILP problem (because every constraint has to be satisfied), and we can penalize when we have to set one variable to true. Here is a little brief:

ILP to WMSAT

Minimize: $\sum_{i=1}^{n} x_i$
Constraint 1: $\sum_{v \in CDT(i,j)} v \geq 1$ for every $1 \leq i < j \leq n$
Constraint 2: $x_i \in \{0, 1\}$ for every $1 \leq i \leq n$

Minimize section: $\bigwedge_{i=1}^{n} (\neg x_i, 1)$
Constraint 1: $(\bigvee_{v \in CDT(i,j)} v, \infty)$ for every $1 \leq i < j \leq n$
Constraint 2: Inherent from SAT nature

WMSAT to ILP

Figure 20: Transforming constraints from ILP to WMax-SAT and vice versa, self-elaborated

Constraint 1: $\sum_{v \in CDT(i,j)} v \geq 1$ for every $1 \leq i < j \leq n$ can be transformed into $(\bigvee_{v \in CDT(i,j)} v, \infty)$ for every $1 \leq i < j \leq n$. This transformation is possible because at least one variable must be true to satisfy the constraint, and the variables can only take boolean values. In SAT, a clause is formed by the disjunction of literals, so the clause will be satisfied if at least one literal is true. Moreover, since no negative numbers are present in the ILP formulation, no negative literals will arise from this constraint. Regarding the infinite weight, we consider only models where all the clauses must be true. Adding an infinite weight ensures that we account for all the constraints from this formulation.

Constraint 2: Since SAT variables can only take boolean values, no further reduction is needed for this constraint; it naturally aligns with the characteristics of SAT problems.

Minimize Function: In this case, we penalize the formulation when it chooses to evaluate a literal as true. To achieve this, we force the formulation to make the literal negative, and if the lit eral is set to true, we incur a cost. In this instance, we use a trivial cost of 1, but any finite number will suffice. It's important to note that all literals are assigned the same weight, as there is no preference for prioritizing one over another.

Here I will show you an example. The following ILP formulation is the Metric Dimension formulation of the Hypercube graph Q_2 (which is the graph formed by the cartesian product $K_2 \times K_2$):

Objective Function:

$$\text{Minimize:} \quad x_0 + x_1 + x_2 + x_3$$

Subject to:

$$\begin{aligned} x_0 + x_1 + x_2 + x_3 &\geq 1 \\ x_0 + x_3 &\geq 1 \\ x_1 + x_2 &\geq 1 \\ x_0, x_1, x_2, x_3 &\in \{0, 1\} \end{aligned}$$

The objective function can be seen it as the following clauses where, if we select one node to the resolving set we pay cost:

$$\text{Minimize: } x_0 + x_1 + x_2 + x_3 \iff (\neg x_0, 1) \wedge (\neg x_1, 1) \wedge (\neg x_2, 1) \wedge (\neg x_3, 1)$$

and we only have to select valid resolving set, so all the constraints must be true. Because of that we put infinite weight on the clauses:

$$\begin{aligned} &x_0 + x_1 + x_2 + x_3 \geq 1 \iff (x_0 \vee x_1 \vee x_2 \vee x_3, \infty) \\ &x_0 + x_3 \geq 1 \iff (x_0 \vee x_3, \infty) \\ &x_1 + x_2 \geq 1 \iff (x_1 \vee x_2, \infty) \end{aligned}$$

And our ILP formulation can be transformed in this way:

$$\begin{array}{|c|} \hline \text{Minimize: } x_0 + x_1 + x_2 + x_3 \\ \text{Subject To:} \\ x_0 + x_3 \geq 1 \\ x_1 + \text{`}x_2 \geq 1 \\ x_0 + x_1 + \text{`}x_2 + x_3 \geq 1 \\ \hline \end{array} \;\; \overset{\text{ILP to WMSAT}}{\underset{\text{WMSAT to ILP}}{\rightleftarrows}} \;\; \begin{array}{|c|} \hline (\neg x_0, 1) \wedge (\neg x_1, 1) \wedge (\neg x_2, 1) \\ \wedge(\neg x_3, 1) \wedge (x_0 \vee x_3, \infty) \wedge (x_1 \vee x_2, \infty) \\ \wedge(x_0 \vee x_1 \vee x_2 \vee x_3, \infty) \\ \hline \end{array}$$

Figure 21: Example of reduction using Metric Dimension ILP formulation for graph Q_2

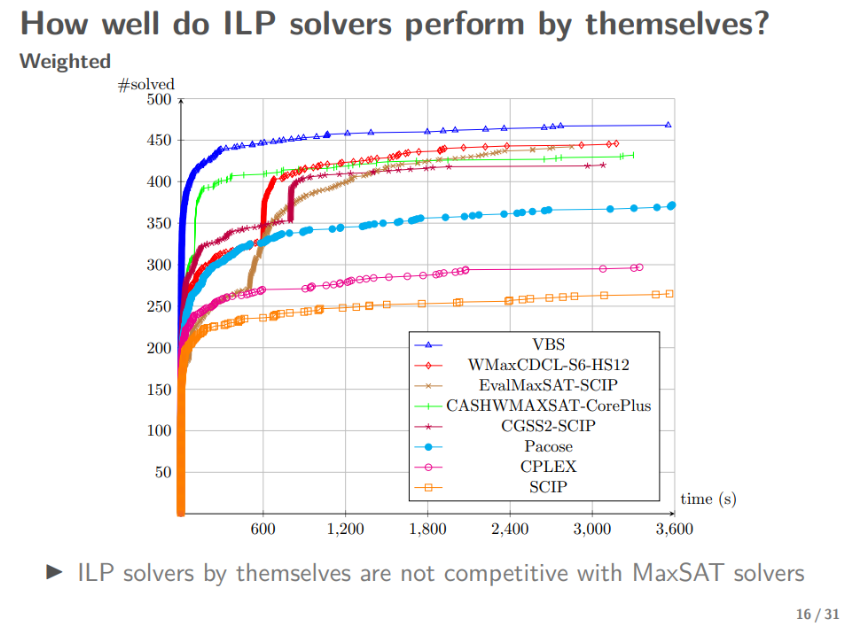


Figure 22: How well do ILP solvers perform by themselves? By Max-SAT competition 2023

14.5 Experiments: ILP Solver vs WMSAT Solver

Professor Carbonell recommended Gurobi as my ILP solver because while IBM's had internal problems with the CPLEX team, Gurobi was more developed, and CPLEX wasn't as optimized. Additionally, Gurobi offers an academic license, making it freely available for academic purposes such as this thesis. For the Weighted Max-SAT Solver, he redirected me to the Max-SAT competition[33] and suggested choosing the winner of this year. All these solvers are *Open-Source*, and fortunately, we have the flexibility to choose the one that best suits our needs.

I started by choosing WMaxCDCL-S6-HS12. This solver first executes SCIP for 600 seconds; if a solution is not found, it then executes MaxHS for 1200 seconds. Finally, if a solution still does not appear, it executes WMaxCDCL for the remaining time. I selected this solver because it seemed to be one of the best overall in the competition (see **Figure 22**), and I downloaded it from the competition. However, I encountered an Execution Error in its binary object, and initially, I thought I hadn't executed it properly. After reviewing the code and attempting to execute different binary objects, I decided to contact the creators of WMaxCDCL. Fortunately, I discovered that one of the creators is from Spain [34]: Jordi Coll from the *Universitat de Girona*. Professor Coll informed me that, indeed, the binary published on the Max-SAT 2023 Web-Page (as of 8/12/2023) has a problem. He also mentioned that they had encountered numerous issues with simple instances like my formulation and the problem was on the SCIP file, so he suggested me to use MaxHS or WMaxCDCL. As I want to take the one who has the better performance I will execute both.

Also, as Professor Coll mentioned that my formulation was *easy* for a Weighted Max-SAT solver, I decided to compare the performance of CASHWMAXSAT-CorePlus. In the competition, it appeared to have very good performance until the final moments, where Coll's Solver and EvalMaxSAT-SCIP surpassed it.

Finally, in **Figure 22**, they emphasized that "ILP solvers by themselves are not competitive with MaxSAT solvers" and provided CPLEX and SCIP as examples. As CPLEX demonstrated better performance than SCIP, and considering that I had installed CPLEX at the beginning of the thesis, it would also be interesting to include CPLEX in the comparison.

14.6 Samples for instances and execution

For this experiment, my goal was to generate graphs of various sizes to compare different amounts of constraints. I specifically chose the different dimensions of Hypercube Graphs because the Hypercube Graph of dimension d, denoted as Q_d, consists of 2^d vertices and $2^{d-1} \cdot d$ edges. By selecting different values of d, we can create samples that grow exponentially. Due to the fact that my formulation has $O(n^2)$ constraints, I will have $O(2^{d^2})$ constraints. Hence providing a sufficient basis for conducting experiments. Here is a little brief of how many **Constraint 1** are provided in every hypercube graph:

Q_d	**#Variables**	**#Constraint 1**
Q_1	2	1
Q_2	4	6
Q_3	8	28
Q_4	16	120
Q_5	32	496
Q_6	64	2016
Q_7	128	8128
Q_8	256	32640
Q_9	512	130816

Table 25: HyperCube number of constraints and variables

For the *Integer Linear Programming* solver I will pass the files in the Linear Programming format[35] while the Weighted Max-SAT solver format will be the new one introduced in 2022 by the Max-SAT competition. The file I used for generate the files can be found in [28].

For the execution I will use the following environment:

- **Laptop**: Asus ZenBook UM425UAZ
- **Operating System**: Debian 12
- **Samples**: Metric Dimension formulation for the Hypercube graph from dimension 1 to 9
- **Messuring Time**: *time* command from Unix, getting *real time*
- **Calls for the solver**: Execute binary object passing the file in .lp/.wcnf format.

14.7 Results

Firstly, I must mention that certain solvers took a considerable amount of time to solve certain instances, with some samples exceeding 30 minutes. Consequently, I decided to implement a *timeout* of five minutes; if the solver couldn't find a solution within this time frame, I would terminate the execution.

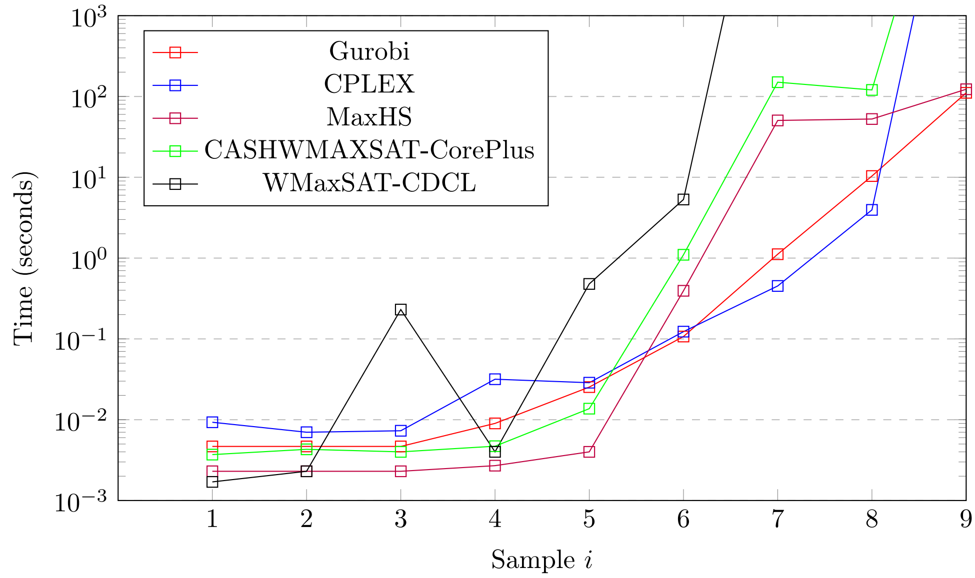


Figure 23: Graphic showing the performance of solvers, self-elaborated

With this information I firstly runned 3 times every solver and I get the mean of the times to compare the performance between them. All times can be seen it in **Appendix A** and in the official repository.

Although we can't set a winner of this comparison it's interesting to take in account interesting facts:

- Firstly, it's interesting to examine **Table 26**: Jordi Coll noted that MaxHS is not a Weighted Max-SAT solver; instead, it solves the *Hitting Set* problem. This problem is defined as follows: given a pair (C, k) with $C = \{S_1, S_2, \ldots, S_m\}$ as a collection of subsets of S and $k \in \mathbb{N}$, we want to determine if there exists a subset $S' \subset S$ such that $|S'| < k$ and $S' \cap S_i \neq \emptyset$ for all $S_i \in C$.
- MaxHS also uses CPLEX.
- No Weighted Max-SAT solver appears at the top (although WMaxCDCL won the first and second samples, they are considered *short* instances, and their significance is not considered).
- As the size increases, ILP solvers gain an advantage over other solvers.
- Gurobi and MaxHS are the only solvers that solved sample 9 in less than five minutes.
- Surprisingly, CPLEX performed better than Gurobi in samples 7 and 8.
- Gurobi didn't win alone in any sample.

Hitting Set and Integer Linear Programming (ILP) stand out as the top performers in the Metric Dimension problem. Interestingly, no Weighted Max-SAT solver emerged as the winner. It's noteworthy to mention that, curiously, the Max-SAT competition asserted that ILP solvers are not competitive by themselves. Additionally, it is important to note that solvers undergo preprocessing techniques, where attempts are made to simplify the instance into a more "easy" format for the solver. However, the impact of these preprocessing techniques on the Metric Dimension problem remains uncertain. As a result, we cannot designate a specific technique as the unequivocal winner. While a more in-depth experiment would be intriguing, it falls beyond the scope of this thesis due to time restrictions and not aligning with the primary focus of the research. In fact, this experiment arises the idea from **Chapter 15**

Q_d	**Winner**
Q_1	WMaxCDCL
Q_2	WMaxCDCL/MaxHS
Q_3	MaxHS
Q_4	MaxHS
Q_5	MaxHS
Q_6	Gurobi/CPLEX
Q_7	CPLEX
Q_8	CPLEX
Q_9	Gurobi/MaxHS

Table 26: Winner of every sample, self-elaborated

15 Graph Family: Hypercube graphs

An interesting graph family studied by different authors [31], [36] is the Hypercube Graph. The Hypercube graph is a special *Hamming* graph. The Hypercube graph of dimension d, denoted as Q_d, is the graph with 2^d vertices represented by all binary strings of length d where two vertices are connected if and only if their binary representations differ on one bit. The problem of computing the Metric Dimension on Hypercube graphs has been extensively studied by various authors and techniques [37]–[39], many of which involve approximations. As of today, known values [40] for the Metric Dimension on Hypercube graphs are

$$\beta(Q_d) = \begin{cases} n & \text{if } n = 1,2,3,4 \\ n-1 & \text{if } n = 5,6,7 \\ n-2 & \text{if } n = 8,9 \\ n-3 & \text{if } n = 10,11 \\ n-4 & \text{if } n = 12,13 \\ n-5 & \text{if } n = 14,15,16 \\ n-6 & \text{if } n = 17 \end{cases}$$

As I'm using *solvers* I have to compute 2^d nodes, calculate all distances, and create the file for Linear Programming. Because of that I need to assess the performance and optimize my code to generate the files quickly. With this goal in mind, I first switched my programming language to C, which is the lowest-level programming language I know, ensuring that I can implement all possible optimizations. I also implemented parallelism using *OpenMP* for calculating the distance with respect to one vertex using the BFS algorithm. Additionally, I employ bit manipulation when possible instead of using normal arithmetic operations. I created the full distance matrix instead of half of the matrix (knowing it's symmetric) because, later, when I started creating the file, I need to look at consecutive positions. Implementing only half of the matrix might cause "jumping" to different rows of the matrix, leading to cache misses. All this code can be found on my GitHub[28].

So, I attempted to compute the Metric Dimension with this initial approach. I started by computing Q_{12} to assess whether it would be feasible to compute the Metric Dimension for Q_{18}. However, as I computed this value, I realized that this approach is entirely unfeasible. The file generated for the Metric Dimension formulation for this graph occupied over 200GB. Moreover, when I passed this file to the solvers (Gurobi and CPLEX), both of them were terminated because Linux detected that I used more than 4GB of RAM during this process. As this graph grows with a scale of 2 on one more dimension and, supposing I doubled the size of the file and the RAM I need to execute the solver, I may need over 12800GB of space and 256GB of RAM. Since I don't have these resources, I'm going to discard this first way to compute this dimension.

But here's an interesting fact I found: During the generation of the constraints, I noticed that many constraints were repeated, and only a few of them are different. For instance, **Table 27** provides information on the number of possible variables, the number of generated constraints, the number of constraints that are different between them, and the ratio between the different constraints with respect to the total.

Q_d	**#Variables**	**#Constraint**	**#Different Constraints**	**Ratio different constraints**
Q_1	2	1	1	100%
Q_2	4	6	3	50%
Q_3	8	28	7	25%
Q_4	16	120	21	17.5%
Q_5	32	496	61	12.30%
Q_6	64	2016	183	9.05%
Q_7	128	8128	547	6.73%
Q_8	256	32640	1641	5.03%
Q_9	512	130816	4921	3.76%

Table 27: HyperCube different number of constraints, self-elaborated

At this moment, I asked myself: How could one contribute to this work? Well, the generation of constraints takes $O(n^3)$ time (because for every pair of nodes, you have to look at all the nodes that *resolve* the pair), and much of this time is going to be spent on computing constraints that are repeated! So, I suggest a research direction related to the Metric Dimension on this graph: Investigate a new way to generate only valuable constraints. This will lower the time spent on generating constraints and the size and memory that one needs to solve different instances of Hypercubes.

If someone wants to get an idea about how the constraints can grow intelligently, we can make an approximation using polynomial interpolation. In this approach, we create a polynomial function that passes through all our points, with the number of different constraints as the points to interpolate. The interpolation can be performed in several ways; for instance, in the *Numerical Computation* course I took at FIB, I learned Lagrange Interpolation, which involves creating the unique polynomial of the lowest degree that interpolates the given set of data. We can try to *estimate* how many different constraints would be in dimension d by evaluating the polynomial.

The points I will use for creating the polynomial are, on the X-axis, the number of variables, and on the Y-axis, the number of different constraints. Using the formula to generate the Lagrange interpolation you get the following polynomial that, at some point, starts decreasing (see **Figure 24**).

This is not what someone would expect to happen to the constraints as the dimension is growing. So, I did another Lagrange interpolation using the dimension as the value on the X-axis. This time, the polynomial seems to be more realistic than the other, as you can see in **Figure 25**

$$\begin{aligned}L(x) = &-\frac{140933}{6519544440437235712}x^8 + \frac{9815663}{44741971402555392}x^7\\ &-\frac{19433359}{264224240566272}x^6 + \frac{5439785}{524254445568}x^5\\ &-\frac{14601851}{22196256768}x^4 + \frac{132933515}{7167541248}x^3\\ &-\frac{25871131}{143990784}x^2 + \frac{272128285}{166676832}x\\ &-\frac{447559271}{265641201}\end{aligned}$$

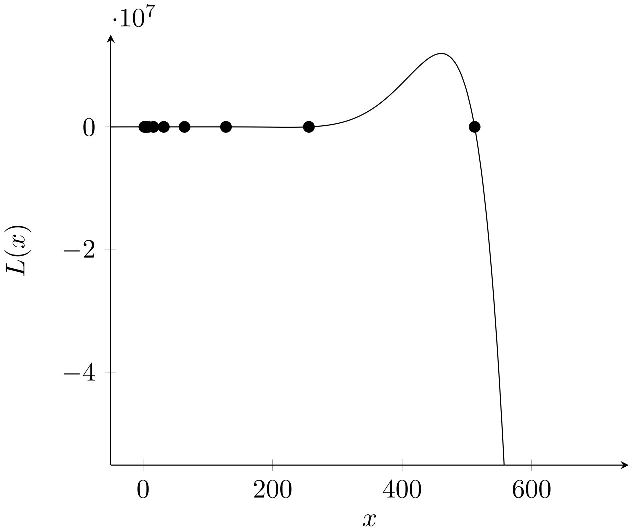


Figure 24: Lagrange Interpolation using $(2^d, \#\text{Different Constraints})$ as points, self-elaborated

$$\begin{aligned}L(x) = &\frac{1}{315}x^8 - \frac{4}{45}x^7 + \frac{16}{15}x^6\\ &-\frac{62}{9}x^5 + \frac{128}{5}x^4 - \frac{2416}{45}x^3\\ &+\frac{18374}{315}x^2 - \frac{73}{3}x + 1\end{aligned}$$

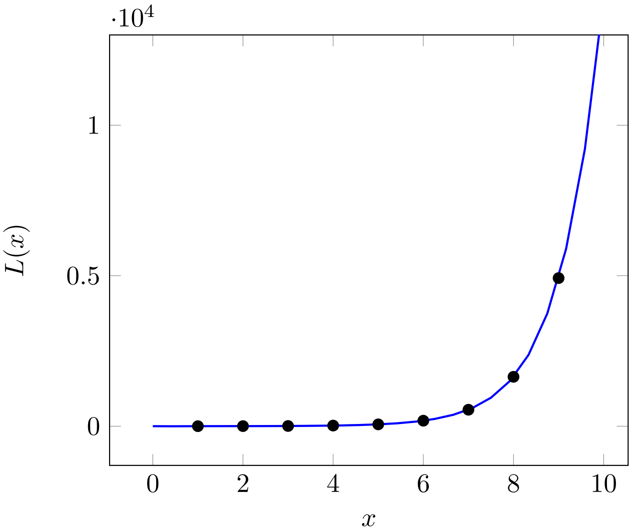


Figure 25: Lagrange Interpolation using the dimension as the X-value, self-elaborated

Now, we can get an idea about what research on selecting the real-valued constraints would be like. **Table 28** reflects the evaluation of the polynomial on the next ten dimensions. As you can see, we have a similar number of repeated constraints in Q_9 and without repeated constraints in Q_{13}. We are computing the Metric Dimension of 2^{13} nodes using the power of the actual 2^9 nodes!.

Chapter 14 uses Hypercubes as samples to execute different solvers. However, I used all the constraints (including repeats) to execute solvers because I don't know if Metric Dimension, by its nature, repeats many constraints or not. A more detailed study on the constraints could contribute to the Metric Dimension problem, although solvers could have pre-processing techniques where they delete repeated constraints or not. Specifically talking about Hypercubes, it would be interesting to see how time grows as the input expands and try to predict how much time would be spent with the different constraints.

Q_d	**Expected #Constraints using interpolation**
Q_{10}	14251
Q_{11}	38655
Q_{12}	97021
Q_{13}	225525
Q_{14}	488671
Q_{15}	994971
Q_{16}	1918417
Q_{17}	3527025
Q_{18}	6219859
Q_{19}	10574071
Q_{20}	17403621

Table 28: Expected number of different constraints

16 Conclusions and Special Thanks

16.1 Conclusions

16.1.1 About Tournaments

I *reformulated* my director's proposition in his article *Symmetry Breaking in Tournaments*[5], establishing an upper bound of $\beta(T) \leq \lfloor n/2 \rfloor$ for all tournaments T without any restriction. Additionally, we provide a characterization of tournaments with Metric Dimension equal to one, outlining the specific form that these graphs must have.

16.1.2 About type III bicyclic graphs

Although we couldn't end the proof for all the different subcases we only left two of the six different subcases.

16.1.3 ILP vs WMaxSAT

We cannot definitively declare one technique superior to another for the Metric Dimension problem, but we dismiss the notion that Weighted Max-SAT (WMaxSAT) solvers are more efficient than Integer Linear Programming (ILP) solvers. Refer to **Chapter 14** for a detailed exploration!

16.1.4 Hypercube graphs

We observed an interesting behavior in the number of constraints in *HyperCube* graphs, where many of the generated constraints are repeated.

16.2 Open Questions

- Someone can complete the three subcases I provided in **Chapter 13** to entirely cover this new graph family in the Metric Dimension, contributing to the work done in [29].
- It would be highly intriguing to conduct a more in-depth experiment in **Chapter 14**, considering different parameters and performing a more detailed analysis of the solver techniques.
- Professor Carbonell mentioned that Gurobi was more developed than CPLEX because IBM's CPLEX team faced internal problems. Surprisingly, CPLEX performed better on some results

than Gurobi. Therefore, although they aren't open-source tools, it would be interesting to see where CPLEX performs better than Gurobi and vice versa.

- Exploring the performance of *Hitting Set* solvers would be interesting.

- **Chapter 15** discusses the number of constraints generated by the usual formulation and how many of them are repeated or not. I recommend reading **Chapter 15** to take a look at the constraints and encourage the reader to estimate how time would be affected. A more in-depth study on generating constraints for the Hypercube graph would also be valuable.

- Obviously, a dataset with the Metric Dimension of different graphs would be valuable to stimulate future research.

- Future studies on the Metric Dimension are also promising.

16.3 Special Thanks

First and foremost, I'd like to express my gratitude to Antoni Lozano for taking on the role of my director and for initiating the concept of this project. His $\Theta(2^{n!})$ wisdom has been invaluable in completing this bachelor's thesis, and I deeply appreciate Antoni Lozano as both a mentor and a researcher.

I also extend my appreciation to Mercè Mora, whose assistance with the idea for proving the Metric Dimension on bicyclic graphs has been invaluable. I am truly grateful for her support, even though she is not directly involved in this thesis. Also, thanks to Enric Rodríguez for suggesting the experiment of **Chapter 14**.

And last but not least, heartfelt thanks to all the friends I've made during my time studying for my Computer Engineering degree and to the professors who have shared their passion for their respective subjects.

References

[1] *P vs np - clay mathematics institute*, `https://www.claymath.org/millennium/p-vs-np/`, Accessed: 18/09/2023.

[2] F. Harary and R. A. Melter, "On the metric dimension of a graph," *Ars combin*, vol. 2, no. 191-195, p. 1, 1976.

[3] P. J. Slater, "Leaves of trees," *Congr. Numer*, vol. 14, no. 549-559, p. 37, 1975.

[4] M. R. Garey, "Computers and intractability: A guide to the theory of np-completeness, freeman," *Fundamental*, 1997.

[5] A. Lozano, "Symmetry breaking in tournaments," *Electronic journal of combinatorics*, vol. 20, no. 1, Paper 69, pp. 1–14, 2013.

[6] J. Díaz, O. Pottonen, M. Serna, and E. J. van Leeuwen, "Complexity of metric dimension on planar graphs," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 132–158, 2017.

[7] R. C. Tillquist, R. M. Frongillo, and M. E. Lladser, "Getting the lay of the land in discrete space: A survey of metric dimension and its applications," *arXiv preprint arXiv:2104.07201*, 2021.

[8] T. Vetrik, "On the metric dimension of circulant graphs with 4 generators," *Contributions to Discrete Mathematics*, vol. 12, no. 2, 2017.

[9] L. Fortnow, "The status of the p versus np problem," *Communications of the ACM*, vol. 52, no. 9, pp. 78–86, 2009.

[10] *Obsidian*, `https://obsidian.md/`, Accessed: 01/10/2023.

[11] *Github*, `https://github.com/`, Accessed: 01/10/2023.

[12] *Overleaf*, `https://www.overleaf.com/`, Accessed: 27/09/2023.

[13] *Ibm ilog cplex*, `https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer`, Accessed: 27/09/2023.

[14] *Ibm ilog cplex optimization studio*, `https://www.ibm.com/docs/en/icos/22.1.1?topic=optimizers-cplex-c-reference-manual`, Accessed: 25/09/2023.

[15] *Combinatorial data*, `https://users.cecs.anu.edu.au/~bdm/data/`, Accessed: 26/09/2023.

[16] *Brendan mckay*, `https://users.cecs.anu.edu.au/~bdm/`, Accessed: 26/09/2023.

[17] *Asus zenbook um425uaz*, `https://www.pccomponentes.com/asus-zenbook-14-um425uaz-ki016w-amd-ryzen-7-5700u-16gb-512gb-ssd-14`, Accessed: 28/09/2023.

[18] *Glassdoor*, `https://www.glassdoor.es/index.htm`, Accessed: 29/09/2023.

[19] *Glassdoor researcher*, `https://www.glassdoor.com/Salaries/spain-research-scientist-salary-SRCH_IL.0,5_IN219_KO6,24.htm`, Accessed: 29/09/2023.

[20] *Glassdoor programmer*, `https://www.glassdoor.com/Salaries/spain-software-engineer-salary-SRCH_IL.0,5_IN219_KO6,23.htm`, Accessed: 29/09/2023.

[21] *Glassdoor project*, `https://www.glassdoor.com/Salaries/spain-project-manager-salary-SRCH_IL.0,5_IN219_KO6,21.htm`, Accessed: 29/09/2023.

[22] *Official page asus zenbook um425uaz*, `https://www.asus.com/laptops/for-home/zenbook/zenbook-14-um425-ua/`, Accessed: 29/09/2023.

[23] *Tarifaluzhora*, `https://tarifaluzhora.es/`, Accessed: 01/10/2023.

[24] "How sustainability is fundamentally changing consumer preferences," Capgemini Research Institute, Tech. Rep., 2021.

[25] G. de España, "Instrucciones para usar la calculadora de huella de carbono para organizaciones," Ministerio para la Transición Ecológica y el Reto Demográfico, Tech. Rep., 2023.

[26] *Enric rodriguez-carbonell webpage*, `https://www.cs.upc.edu/~erodri/`, Accessed: 29/11/2023.

[27] G. Chartrand, M. Raines, and P. Zhang, "On the dimension of oriented graphs," *Utilitas Mathematica*, vol. 60, 2001.

[28] *Github repository to visit information*, `https://github.com/AleexHrB/MetricDimension-TFG`, Accessed: 30/12/2023.

[29] A. Khan, G. Haidar, N. Abbas, M. U. I. Khan, A. U. K. Niazi, and A. U. I. Khan, "Metric dimensions of bicyclic graphs," *Mathematics*, vol. 11, no. 4, p. 869, 2023.

[30] M. Claverol, A. García, G. Hernández, *et al.*, "Metric dimension of maximal outerplanar graphs," *Bulletin of the Malaysian Mathematical Sciences Society*, vol. 44, pp. 2603–2630, 2021.

[31] J. Cáceres, C. Hernando, M. Mora, *et al.*, "On the metric dimension of cartesian products of graphs," *SIAM Journal on Discrete Mathematics*, vol. 21, no. 2, pp. 423–441, 2007.

[32] A. Lozano, M. Mora, C. Seara, and J. Tey, "Caterpillars are antimagic," *Mediterranean Journal of Mathematics*, vol. 18, pp. 1–12, 2021.

[33] *Max-sat competition 2023*, `https://maxsat-evaluations.github.io/2023/index.html`, Accessed: 30/11/2023.

[34] J. Berg, M. Järvisalo, R. Martins, and A. Niskanen, "Maxsat evaluation 2023,"

[35] *Lp file format: Algebraic representation*, `https://www.ibm.com/docs/en/icos/22.1.0?topic=cplex-lp-file-format-algebraic-representation`, Accessed: 08/12/2023.

[36] A. Kelenc, A. T. M. Toshi, R. Skrekovski, and I. G. Yero, "On metric dimensions of hypercubes," *arXiv preprint arXiv:2102.10916*, 2021.

[37] J. Kratica, V. Kovačević-Vujčić, and M. Čangalović, "Computing the metric dimension of graphs by genetic algorithms," *Computational Optimization and Applications*, vol. 44, no. 2, pp. 343–361, 2009.

[38] A. Hertz, "An ip-based swapping algorithm for the metric dimension and minimal doubly resolving set problems in hypercubes," *Optimization Letters*, vol. 14, pp. 355–367, 2020.

[39] D. T. Murdiansyah and Adiwijaya, "Computing the metric dimension of hypercube graphs by particle swarm optimization algorithms," pp. 171–178, 2017.

[40] A. Beardon, "Resolving the hypercube," *Discrete Applied Mathematics*, vol. 161, no. 13, pp. 1882–1887, 2013, ISSN: 0166-218X. DOI: `https://doi.org/10.1016/j.dam.2013.02.012`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0166218X13000644`.

A Times from Chapter14

Q_d	**Gurobi First Execution**	**Gurobi Second Execution**	**Gurobi Third Execution**	**Mean**
Q_1	$0.006s$	$0.004s$	$0.004s$	$0.00467s$
Q_2	$0.005s$	$0.005s$	$0.004s$	$0.00467s$
Q_3	$0.005s$	$0.005s$	$0.004s$	$0.00467s$
Q_4	$0.009s$	$0.009s$	$0.009s$	$0.0090s$
Q_5	$0.026s$	$0.025s$	$0.025s$	$0.02533s$
Q_6	$0.113s$	$0.104s$	$0.104s$	$0.1070s$
Q_7	$1.244s$	$1.123s$	$1.133s$	$1.1167s$
Q_8	$11.431s$	$9.406s$	$10.243s$	$10.3600s$
Q_9	$1m51.586s$	$1m49.659s$	$1m51.522s$	$1m50.9223s$

Table 29: Gurobi Execution, self-elaborated

Q_d	**CPLEX First Execution**	**CPLEX Second Execution**	**CPLEX Third Execution**	**Mean**
Q_1	$0.011s$	$0.009s$	$0.008s$	$0.0093s$
Q_2	$0.007s$	$0.007s$	$0.007s$	$0.0070s$
Q_3	$0.007s$	$0.008s$	$0.007s$	$0.0073s$
Q_4	$0.026s$	$0.020s$	$0.049s$	$0.0317s$
Q_5	$0.030s$	$0.029s$	$0.027s$	$0.0287s$
Q_6	$0.122s$	$0.129s$	$0.118s$	$0.1230s$
Q_7	$0.446s$	$0.450s$	$0.460s$	$0.4520s$
Q_8	$3.868s$	$3.996s$	$4.015s$	$3.9500s$
Q_9	*Timeout*	*Timeout*	*Timeout*	*Timeout*

Table 30: CPLEX Execution, self-elaborated

Q_d	**MaxHS First Execution**	**MaxHS Second Execution**	**MaxHS Third Execution**	**Mean**
Q_1	$0.003s$	$0.002s$	$0.002s$	$0.0023s$
Q_2	$0.003s$	$0.002s$	$0.002s$	$0.0023s$
Q_3	$0.002s$	$0.003s$	$0.002s$	$0.0023s$
Q_4	$0.003s$	$0.003s$	$0.002s$	$0.0027s$
Q_5	$0.004s$	$0.004s$	$0.004s$	$0.0040s$
Q_6	$0.393s$	$0.395s$	$0.394s$	$0.3940s$
Q_7	$50.527s$	$50.509s$	$50.510s$	$50.5153s$
Q_8	$52.773s$	$52.495s$	$52.571s$	$52.5950s$
Q_9	$2m3.287s$	$2m3.065s$	$2m2.117s$	$2m2.823s$

Table 31: MaxHS Execution, self-elaborated

Q_d	CWMSAT-CP First Exec	CWMSAT-CP Second Exec	CWMSAT-CP Third Exec	Mean
Q_1	$0.004s$	$0.003s$	$0.004s$	$0.0037s$
Q_2	$0.005s$	$0.005s$	$0.003s$	$0.0043s$
Q_3	$0.004s$	$0.005s$	$0.003s$	$0.004s$
Q_4	$0.006s$	$0.004s$	$0.004s$	$0.0047s$
Q_5	$0.014s$	$0.014s$	$0.013s$	$0.0137s$
Q_6	$1.148s$	$1.072s$	$1.078s$	$1.0993s$
Q_7	$2m30.107s$	$2m30.502s$	$2m30.129s$	$2m30.246s$
Q_8	$2m30.488s$	$2m30.423s$	$2m30.418s$	$2m30.443$
Q_9	*Timeout*	*Timeout*	*Timeout*	*Timeout*

Table 32: CASHWMAXSAT-CorePlus Execution, self-elaborated

Q_d	WMSAT-CDCL First Exec	WMSAT-CDCL Second Exec	WMSAT-CDCL Third Exec	Mean
Q_1	$0.002s$	$0.002s$	$0.001s$	$0.0017s$
Q_2	$0.002s$	$0.003s$	$0.002s$	$0.0023s$
Q_3	$0.223s$	$0.226s$	$0.233s$	$0.2307s$
Q_4	$0.005s$	$0.005s$	$0.002s$	$0.0040s$
Q_5	$0.469s$	$0.476s$	$0.489s$	$0.4788s$
Q_6	$5.235s$	$5.479s$	$5.221s$	$5.3117s$
Q_7	*Timeout*	*Timeout*	*Timeout*	*Timeout*
Q_8	*Timeout*	*Timeout*	*Timeout*	*Timeout*
Q_9	*Timeout*	*Timeout*	*Timeout*	*Timeout*

Table 33: WMaxSAT-CDCL Execution, self-elaborated

B How to execute experiments from Chapter14

For generating the SAT files:

1. Go to `https://github.com/AleexHrB/MetricDimension-TFG`
2. Go to `HyperCubes/satSamples`
3. Execute `generate_file.sh`

For generating the ILP files:

1. Go to `https://github.com/AleexHrB/MetricDimension-TFG`
2. Go to `HyperCubes/ilpSamples`
3. Execute `generate_file.sh`

For executing SAT times:

1. Go to `https://github.com/AleexHrB/MetricDimension-TFG`
2. Go to `HyperCubes/samples_hcube/satTimes`
3. Download solvers at `https://maxsat-evaluations.github.io/2023/descriptions.html` and put the binary file on the folder.
4. Execute `exec_solver.sh` where `solver` is the name of the solver you want to execute.

For executing ILP times:

1. Go to `https://github.com/AleexHrB/MetricDimension-TFG`
2. Go to `HyperCubes/samples_hcube/ilpTimes`
3. Reclam Gurobi/CPLEX at their corresponding webpage and put the binary files on the folder.
4. Execute `exec_solver.sh` where `solver` is the name of the solver you want to execute.