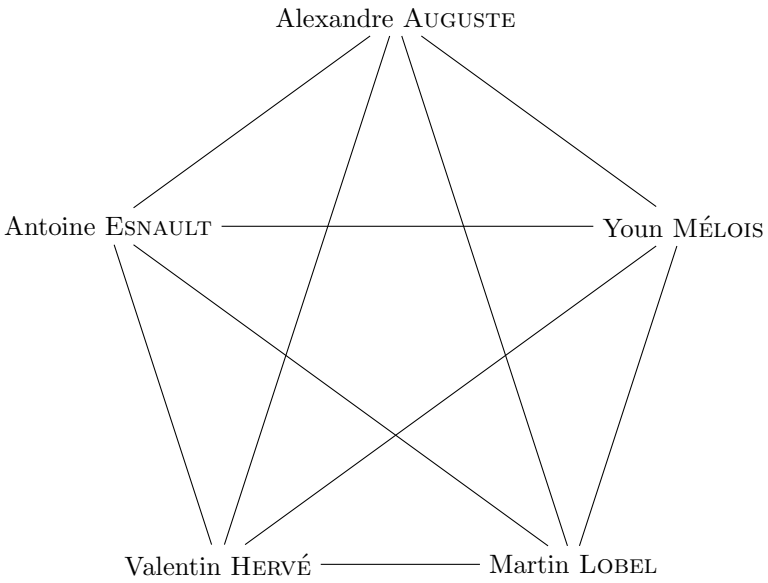
Final Project

The Maximum Edge Weight Clique Problem



Class group :
CIR3 - Team 1

Teacher :
Leandro Montero

January 6, 2023

Contents

1 Introduction

1.1 Presentation of the subject

Given an undirected simple connected graph $G = (V, E)$, a clique is a complete maximal induced subgraph of G. The Maximum Clique Problem consists in finding a clique of maximum size. This is a classic graph theory problem that has many-real life applications in various fields such as social networks, chemistry, bioinformatics. In addition, this problem is NP-Hard and it has been studied as a combinatorial optimization problem, being very important in operations research and theoretical computer science.

Let $G = (V, E, w)$ be an undirected simple weigheted connected graph such taht $|V| = n, |E| = m$ and $w : E \to \mathbb{R}_{[1,100]}$ is the weight function. That is, G is a graph with bounded weights on its edges. The Maximum Edge Weight Clique Problem (MEWC) consists in finding a clique that maximises the sum of the weights of its edges. Clearly MEWC is also a NP-Hard problem since the Maximum Clique Problem is the special case in which all weights are equal.

1.2 Configuration

- Ajouter les configurations que l'on a utilise pour les tests du style la config

On this project, we decide to use **the C++ language** to developp and implement our algorithms. Nevertheless, a debate took place, especially on the choice of the language. We hesitated between Python and C++. The first one was for us easier to handle and was a tool in which we had more confidence in our ability to use it efficiently. The latter was nevertheless preferred because of its speed of execution, which was an important criterion for the study. We did have some problems with memory allocation, which made us regret this choice at times.

To share the code between us, we used **Github**[1]. A tool that was difficult for some to get used to quickly, especially on the configuration of the project at home, but which brought us a significant gain in efficiency once we had understood how to use it. To share information and communicate between us, we used **Discord**.

1.3 Example of real-life situations

As we said, the MEWC has many real-life applications in various fields such as social networks, chemistry, bioinformatics. We will take a practical example which could include and involve ISEN students in our future. We will reuse the presented case to illustrate the different algorithms that we will implement later in the report.

An example of real-life situations that can be modelled as MEWC is the team formation process during a project, or in the search for a particular social group.

We can imagine that the Student Office of ISEN Nantes is looking to reinforce the video games club of its school. Indeed, the latter has no succession for the following year and is thus led to die if no member presents himself. The future members of the office will have to be in contact with each other during a whole year and it is thus important to find people with common interests so that no tension is formed during their studies. The office has access to the Steam profiles of the students within ISEN (Steam is a video game digital distribution service that gives information about the games played by each one) as well as a record made by the gaming club of the different games played by their members. The fact of playing games in common could bring some people closer, and this makes it a good criterion to create a group that could take over the club because it would share common interests. Otherwise, it would allow to see which games and which group could be present at different events they could organize. Here, forming a team from a group of individuals can be considered as a maximum edge weight clique problem because the goal is to select a subset of individuals such that the members share a common interest that could.

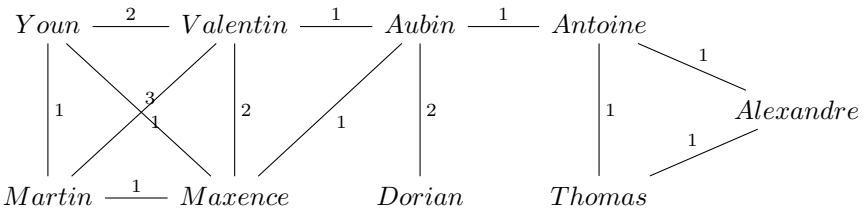
[1] `https://github.com/sehnryr/Final-Graph-Project-ISEN-CIR3`

To model this problem as a maximum edge weight clique problem, you can represent each **individual** as a **vertex** in a graph and add an **edge** between two vertices if the corresponding individuals **share atleast one game on Steam**. The weight of the edge could represent the number of game that they have in common.

For example, on a small scale we could imagine :

Students	**Game played**
Youn	Minecraft, Civilisations, Lost ARK, Among US
Martin	Minecraft
Valentin	Genshin, Minecraft, Civilisations
Maxence	Genshin, Minecraft, Lost ARK, Among US
Aubin	CSGO, Genshin, Overwatch, Stardew Valley
Dorian	CSGO, Paladins, Overwatch
Antoine	League of Legends, Stardew Valley
Thomas	League of Legends, The Last of US
Alexandre	League of Legends, Dofus

Which would give us this graph :



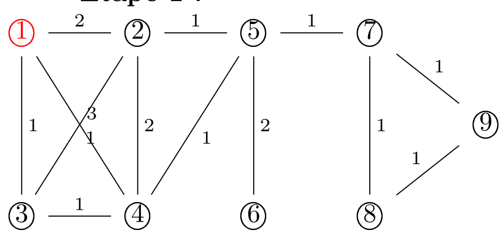
The goal of the maximum edge weight clique problem in this context would be to find a complete subset of individuals such that the sum of the weights of the edges between the individuals is maximized. In this example, the maximum weight clique would be the clique consisting of nodes Youn, Valentin, Martin, Maxence with a total weight of $10(2+1+1+1+3+2)$.

2 Exact Algorithm

2.1 Presentation

2.2 How it works

Etape 1 :



J'analyse ici le graphique, par exemple ici le graph prendra 1 en entree bla

2.3 Pseudo code

2.4 Complexity

2.5 Instance

2.6 Experiments

2.7 Analysis

3 Constructive Algorithm

3.1 Presentation

3.2 How it works

3.3 Pseudo code

3.4 Complexity

3.5 Instance

3.6 Experiments

3.7 Analysis

4 Local Search Algorithm

4.1 Presentation

4.2 How it works

4.3 Pseudo code

4.4 Complexity

4.5 Instance

4.6 Experiments

4.7 Analysis

5 Grasp Algorithm

5.1 Presentation

5.2 How it works

5.3 Pseudo code

5.4 Complexity

5.5 Instance

5.6 Experiments

5.7 Analysis

6 Conclusion

References