

**MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE
LA RECHERCHE SCIENTIFIQUE**

UNIVERSITÉ DE M'HAMED BOUGARA BOUMERDES



FACULTÉ DES SCIENCES
DÉPARTEMENT MATHÉMATIQUES (RO)

MÉMOIRE DE MASTER

Recherche Opérationnelle Modélisation et Aide à la Décision

**Recherche du Plus Grand Sous-Graphe Commun
Entre Deux Graphes**

Présenté par :
Bouhadj Mourad
Graine Mounir

Devant le jury d'examen composé de :

M ^{me} Fatma Messaoudi;	Maître Assistant classe "A" ;	UMBB ;	Président.
M ^{me} Meriem Akli;	Maître Assistant classe "A" ;	UMBB ;	Rapporteur.
M ^{me} Massika Ikhlef;	Maître Assistant classe "B" ;	UMBB ;	Examineur.

Soutenue le : 02/07/2017

Table des matières

Remerciements	4
Dédicaces	5
Introduction générale	6
1 Concepts de Base de la Théorie des Graphes	8
1.1 Introduction	8
1.2 Définitions et Notations	8
1.2.1 Graphe	8
1.2.2 Sommets Adjacents	9
1.2.3 Boucle	10
1.2.4 Degré et Ordre	10
1.3 Graphes Orientés	11
1.4 Graphe Étiqueté et Graphe Pondéré	11
1.4.1 Graphe étiqueté	11
1.4.2 Graphes Pondéré	12
1.5 Chaines	12
1.6 Chemin	13
1.7 Cycle	14
1.8 Circuit	16
1.9 Quelques classes de graphes	16
1.9.1 Graphe Simple	16
1.9.2 Graphe Complet	17

1.9.3	Graphe K-régulier	17
1.9.4	Graphe Biparti	18
1.9.5	Graphe Planaire	18
1.9.6	Graphe Parfait	19
1.9.7	Graphe de l'hyper-cube	19
1.9.8	Arborescence	20
1.10	Graphe et Sous-Graphe	20
1.10.1	Sous-Graphe	21
1.10.2	Graphe Partiel	21
1.10.3	Sous-Graphe Partiel	22
1.10.4	Clique	22
1.10.5	Graphe Complémentaire	23
1.11	Connexité de Graphes	24
1.11.1	Graphe Connexe	24
1.11.2	Graphe Fortement Connexe	24
1.11.3	Composantes Connexes	24
1.11.4	Isthme	25
1.12	Conclusion	26
2	Problèmes Célèbres en Théorie des Graphes	27
2.1	Introduction	27
2.2	Représentation matricielles des graphes	27
2.2.1	Matrices d'Adjacences	27
2.2.2	Caractéristiques de la Matrice d'Adjacence	28
2.2.3	Opérations sur les Graphes	29
2.3	Problèmes Célèbres en Théorie des Graphes	32
2.3.1	Stable	32
2.3.2	Couplage	32
2.3.3	Transversal	33
2.3.4	Recouvrement	33
2.3.5	Problème de Coloration	33
2.4	Problème du Plus Grand Sous-Graphe Commun	34
2.4.1	Définition	34

2.4.2	Domaine d'Utilisation	35
2.4.3	Algorithme Proposé pour la Recherche du PGSGC	36
2.5	Conclusion	37
3	Implémentation de l'Algorithme	38
3.1	Introduction	38
3.2	Langage utilisé	38
3.2.1	Choix du Langage	38
3.2.2	Généralités sur MATLAB	39
3.3	Présentation de l'application :	40
3.4	Exemples d'applications	40
3.4.1	Exemple de Deux Chaînes	40
3.4.2	Exemple de Deux Cycles	42
3.4.3	Exemple de Deux graphes complets	43
3.4.4	Exemples de Deux graphes réguliers	44
3.4.5	Exemple de Deux arbres	47
3.4.6	Exemple de Deux Graphes Quelconques	48
3.4.7	Conclusion	52
	Conclusion générale	53
	Bibliographie	54

Remerciements

*Nous remercions, en premier, **DIEU "Allah"** qui nous a donné la force et la volonté de mener ce travail à bout.*

En suite, nous remercions Madame Meriem Akli qui nous a proposé le thème de cette étude et qui nous a suivi sérieusement dans la préparation de ce mémoire.

Nous remercions particulièrement Madame Fatma Messaoudi pour avoir accepté de nous faire honneur de présider notre jury.

Nous remercions également, Madame Massika Ikhlef d'avoir accepté d'examiner notre mémoire

Pour finir, nous souhaitons adresser nos sincères remerciements à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce mémoire et qui nous ont accompagnée durant ces cinq années de d'étude et de travail.

Dédicaces

Je dédie ce travail
À mes chers **PARENTS**, pour tous leurs sacrifices, leur amour, leur tendresse, leur
soutien et leurs prières tout au long de mes études,
À mon chère frère, **ZAKARIA**, pour son soutien et ses encouragements,
À toute ma famille,
À tous mes amis,
À tous ceux qui me sont chers et dont je n'ai pas cité de nom.

Mourad

Je dédie ce travail
À ma chère **MÈRE**,
À mon **PÈRE**,
À tous mes proches,
À tous mes amis,

Mounir

Introduction générale

La théorie des graphes est un domaine fondé par Claude Berge au début des années soixante, sous cette appellation, bien que les graphes soient largement utilisés en chimie, en biologie, en psychologie, en l'économie Il se trouve que les graphes soient un outil irremplaçable, très efficace pour la modélisation des systèmes et problèmes réels.

L'appellation de théorie des graphes est apparue suite au fameux problème des sept ponts reliant les quatre régions de la ville de Königsberg. Le problème consiste à répondre à la question suivante : est-il possible de visiter les quatre régions de la ville de Königsberg sans passer deux fois par le même pont ? La réponse a été présentée pour la première fois par Euler en modélisant le problème sous forme d'un graphe où les régions sont présentées des sommets et les ponts par des arêtes.

La théorie des graphes ouvre un grand champ de modélisation conduisant à des solutions efficaces pour de nombreux problèmes, en les ramenant à des configurations, qui se dessinent simplement à l'aide de points (sommets) et de liaisons (arêtes) entre ces points. La modélisation de problèmes sous forme de graphes est un outil important, et elle est largement utilisée dans la recherche mathématique, dans l'électricité, la programmation, la gestion des réseaux, l'administration, la sociologie, l'économie, les ventes, la communication

Ce travail fait l'objet de l'étude d'un problème de recherche du plus grand sous graphe commun entre deux graphes. Le problème de recherche du plus grand sous graphe commun entre deux graphes est un problème difficile, il est de complexité NP-Difficile. Ces dernières décennies, de nombreux chercheurs se sont intéressés à ce problème. L'intérêt de recherche

du plus grand sous graphe commun entre deux graphes est de définir ou de mesurer la similarité entre les objets représentés par ces graphes.

Le mémoire est réparti sur trois chapitre. Dans un premier temps nous allons donner les définitions préliminaire et générales sur la théorie des graphes. En suite, le deuxième chapitre sera axé sur les définitions nécessaires à notre étude et la présentation de quelques problèmes célèbres en théorie des graphes. parmi lesquels le problème de recherche du plus grand sous graphe commun. Ce chapitre sera achevé par la proposition d'un algorithme e recherche du plus grand sous graphe commun. Le troisième chapitre sera consacré à la description de l'environnement d'implémentation et aux résultats numériques obtenus sur les différents types de graphes. Le mémoire sera achevé par une conclusion générale.

1

Concepts de Base de la Théorie des Graphes

1.1 Introduction

Avant de lancer toute étude, il est nécessaire de rappeler les définitions élémentaires relatives au domaine de l'étude. Ce chapitre fera l'objet du rappel des concepts base de la théorie des graphes. Nous commençons par donner la définition d'un graphe et de ces propriétés, ensuite nous allons donner quelques classes de graphes puis aborder la connexité des graphes. Nous terminerons le chapitre par une conclusion.

1.2 Définitions et Notations

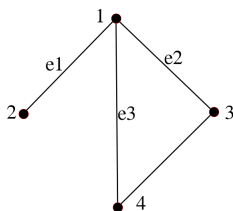
1.2.1 Graphe

Un graphe G est un couple d'ensembles $(V(G), E(G))$, tel que $V(G)$ est l'ensemble non vide de sommets et de cardinalité $|V(G)| = n$ et $E(G)$ est l'ensemble non vide d'arêtes et de cardinalité $|E(G)| = m$. Et on note $G = (V(G), E(G))$ ou tout simplement $G = (V, E)$.

Un graphe peut être représenté géométriquement par des points pour les sommets et des lignes pour les arêtes reliant les différents sommets.

Exemple

Soit le graphe $G = (V, E)$ défini par $V = \{1, 2, 3, 4\}$ et $E = \{e_1, e_2, e_3, e_4\}$ avec $e_1 = (1, 2)$, $e_2 = (1, 3)$, $e_3 = (1, 4)$ et $e_4 = (3, 4)$.



Graphe G

1.2.2 Sommets Adjacents

Deux sommets u et v de V sont dits adjacents, s'il existe une arête e dans E tel que ses extrémités sont u et v . Et on note $e = (u, v) \in E$ ou bien uv . Autrement dit, deux sommets u et v , ($u, v \in V$), sont dits adjacents si et seulement s'ils sont reliés par une arête, tel qu'on puisse passer de u à v ou inversement de v à u , directement sans passer par d'autres sommets. On note cette arête de passage par (uv) ou (vu) selon qu'on passe de u à v ou de v à u .

Exemple

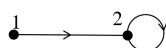
Dans le graphe G , précédant on voit que les sommets adjacents au :

- sommet 1 sont : 2, 4 et 3.
- sommet 2 sont : 1.
- sommet 3 sont : 1 et 4.
- sommet 4 sont : 1 et 3.

1.2.3 Boucle

Une boucle est une arête d'un graphe ayant ses extrémités confondues.

Exemple



Boucle

1.2.4 Degré et Ordre

Le nombre de sommets du graphe G est appelé Ordre de G . Le degré d'un sommet u , noté $d_G(u)$, est le nombre d'arêtes incidentes à u .

On défini aussi :

- le degré minimum de G que l'on note $\delta(G) = \min\{d_G(u); u \in V\}$.
- le degré maximum de G que l'on note $\Delta(G) = \max\{d_G(u); u \in V\}$.

Exemple

Dans le graphe précédant :

- l'ordre du graphe G est égale à 4.
- $\delta(G) = \min\{d_G(u); u \in V\} = 1$.
- $\Delta(G) = \max\{d_G(u); u \in V\} = 3$.

Les sommets d'un graphe ont quelque propriétés, parmi lesquelles nous citons le théorème et le corollaire suivants :

Théorème

La somme des degrés des sommets d'un graphe est égale à 2 fois son nombre d'arêtes.

$$\sum_{u \in V} d_G(u) = 2m$$

Corollaire

Le nombre de sommets de degré impair d'un graphe est pair.

Remarque

Lorsque $d_G(u) = 0$, on dit que le sommet u est isolé et quand $d_G(u) = 1$, il est dit pendent.

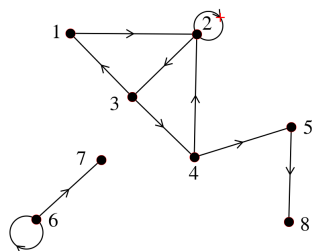
1.3 Graphes Orientés

Un graphe orienté est un graphe dont les arêtes sont orientées. Cela signifie que les extrémités d'une arête ont un sens précis. On parle alors d'arc et non d'arête. On note l'arc par (uv) ou u est son extrémité initiale et v son extrémité finale.

Exemple

Soit $G = (V, E)$ un graphe avec : $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$ et

$E = \{(1, 2), (2, 2), (2, 3), (3, 1), (3, 4), (4, 2), (4, 5), (5, 8), (6, 6), (6, 7)\}$



Graphe Orienté

1.4 Graphe Étiqueté et Graphe Pondéré

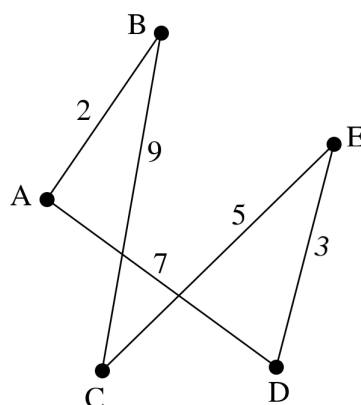
1.4.1 Graphe étiqueté

Un graphe étiqueté est défini par un 6-tuple $G = (V, E, LV, \alpha, LE, \beta)$ avec :

- (V, E) est un graphe.
- LV est un ensemble d'étiquettes de sommets.
- $\alpha : V \rightarrow LV$ est une application attribuant une étiquette à chaque sommet du graphe.
- LE est un ensemble d'étiquettes d'arcs.
- $\beta : E \rightarrow LE$ est une application attribuant une étiquette à chaque arc du graphe.

1.4.2 Graphes Pondéré

On appelle un graphe pondéré, un graphe, orienté ou non, dont les arêtes (ou les arcs) possèdent un poids. C'est à dire que les arêtes possèdent un nombre qui identifie le coût de passage d'un sommet à un autre. Cette pondération est utilisée par exemple dans la représentation d'un réseau routier pour indiquer le temps ou la distance joignant deux villes. Il existe un type de graphe pondéré un peu spécial, on l'appelle graphe probabiliste. Dans ce type de graphe, la somme des pondérations des arêtes sortantes est égale à un. C'est à dire que si on a deux arêtes sortantes d'un sommet, la somme de leur pondération est égale à un. Attention toutefois, les pondérations sont dans ce type de graphe comprises entre 0 et 1. On ne pourra donc pas avoir une arête sortante de poids 2 et l'autre -1 .



Graphes Pondéré

1.5 Chaînes

On appelle chaîne, une suite de sommets adjacents, c'est à dire une suite de sommets reliés par des arêtes. La longueur de la chaîne est le nombre d'arêtes la constituant. C'est à dire une chaîne de longueur q est une séquence de $q+1$ sommets de G noté $p = (u_1, u_2, \dots, u_{q+1})$ telle que chaque sommet u_i , $2 \leq i \leq q$ de la séquence est adjacent aux sommets u_{i-1} et u_{i+1} .

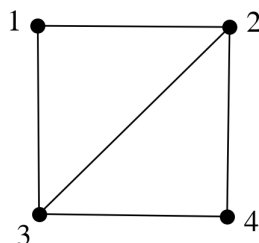
- Une chaîne qui ne passe pas deux fois le même sommet est dite élémentaire. Une chaîne passant une et une seule fois par chacun des sommets de G est dite une chaîne Hamiltonienne. Un graphe n'admettant que des chaînes Hamiltoniennes est dit Graphe smi-Hamiltonien.

- Une chaîne qui ne passe pas deux fois la même arête est dite simple. Une chaîne passant une et une seule fois par chacune des arêtes de G est dite une chaîne Eulerienne. Un graphe n'admettant que des chaînes Eulerien est dit Graphe semi-Eulerien.

Exemple

Dans le graphe suivant, on a :

- $C_1 = (1, 2, 3, 4)$ est une chaîne de longueur 3.
- $C_2 = (1, 3, 4, 2)$ est une chaîne de longueur 3.
- $C_3 = (1, 3, 4)$ est une chaîne de longueur 2.



Chaîne

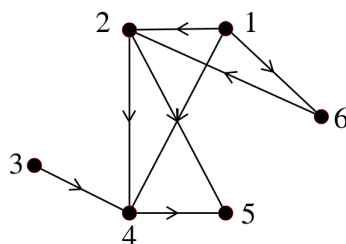
1.6 Chemin

On appelle chemin une suite de sommets reliés par des arcs. Alors un chemin est une chaîne orientée. Comme pour les chaînes, on appelle chemin simple tout chemin qui ne passe pas deux fois par le même sommet.

Exemple

Dans le graphe suivant, on a :

- $C_1 = (1, 6, 2, 5)$ est une chemin de longueur 3.
- $C_2 = (1, 2, 4, 5)$ est une chemin de longueur 3.
- $C_3 = (1, 6, 2)$ est une chemin de longueur 2.

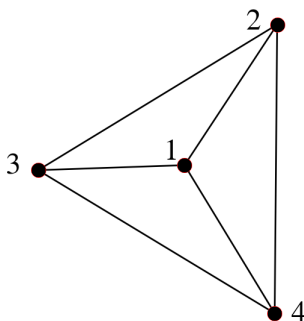


Chemin

1.7 Cycle

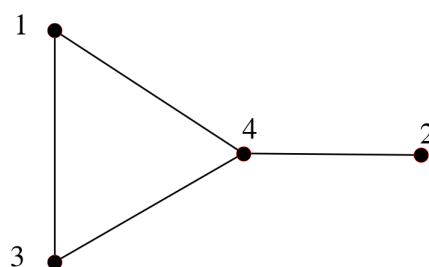
On appelle un cycle une chaîne fermée (le sommet initial est aussi le sommet final). Il est dit élémentaire si la chaîne sur laquelle il est basé est simple, hormis le sommet de départ et d'arrivée). Nous distinguons deux types de cycles :

- **Cycle Eulerien** : Un cycle Eulerien d'un graphe G est un cycle passant une et une seule fois par chacune des arêtes de G . Un graphe admettant un cycle Eulerien est dit Graphe Eulerien.



Cycle Eulerien

- **Cycle Hamiltonien** : Un cycle Hamiltonien d'un graphe G est un cycle passant une et une seule fois par chacun des sommets de G . Un graphe est dit Hamiltonien s'il admet un cycle Hamiltonien.



Cycle Hamiltonien

Remarques

- Plus simplement, on peut dire qu'un graphe est Eulerien (ou semi-Eulerien) s'il est possible de dessiner le graphe sans lever le crayon et sans passer deux fois sur la même arête.
- Contrairement aux graphes Euleriens, il n'existe pas de caractérisation simple des graphes (semi-)Hamiltoniens. Ci-dessous quelques propriétés et conditions suffisantes :
 - Un graphe ayant un sommet de degré 1, ne peut-être Hamiltonien.
 - Si un sommet dans un graphe est de degré 2, alors les deux arêtes incidentes à ce sommet doivent faire partie du cycle Hamiltonien.
 - les graphes complets K_n sont Hamiltoniens.

Les propriétés des graphes Hamiltoniens sont très intéressantes, nous en citons :

Théorème

Soit G un graphe simple d'ordre $n > 3$. Si pour toute paire (u, v) de sommets non adjacents, on a $d_G(u) + d_G(v) > n$, alors G est un graphe Hamiltonien.

Corollaires

Soit G un graphe simple d'ordre $n > 3$. Si pour tout sommet u de G , on a $d_G(u) > n/2$, alors G est Hamiltonien.

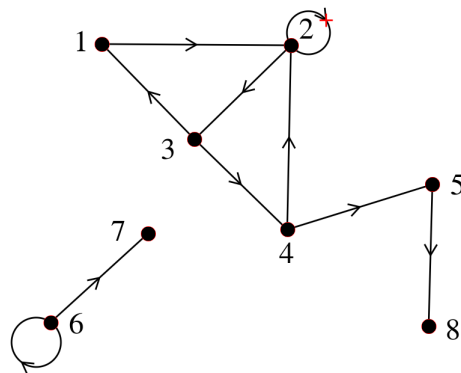
1.8 Circuit

Un circuit est un chemin fermé simple. On entend par chemin fermé, un chemin dont le sommet initial est aussi le sommet final.

Exemple

Dans le graphe suivant, on a :

- $C_1 = (1, 2, 3)$ est une circuit de longueur 3.
- $C_2 = (2, 3, 4)$ est une circuit de longueur 3.



Remarque

Les cycles concernent les graphe non orientés et les circuits concernent les graphe orientés

1.9 Quelques classes de graphes

1.9.1 Graphe Simple

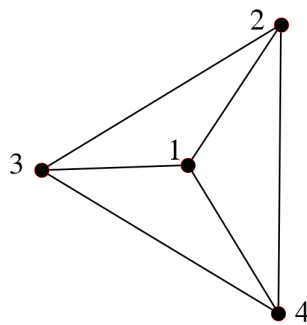
Un graphe G est dit simple, s'il est sans boucles et toute paire de sommets est reliée par au plus une arête(sans arêtes doubles).

1.9.2 Graphe Complet

Un graphe $G = (V, E)$ avec $|V| = n$, est dit complet si tous les sommets sont deux à deux adjacents. Un graphe complet à n sommets est dit aussi n -régulier. Ainsi, chaque sommet de ce graphe est de degré $n - 1$, $(\forall u \in V, d_G(u) = n - 1)$.

Exemple

Le graphe d'ordre 4 ci-dessous illustre un graphe complet dit aussi 3-régulier.



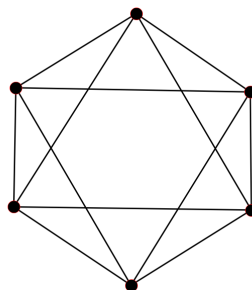
Graphe Complet

1.9.3 Graphe K-régulier

Un graphe G est dit régulier de degré K , si tout ses sommets sont de degré k , $(\forall u \in V, d_G(u) = k)$.

Exemple

Le graphe ci-dessous est d'ordre 6 et il est 4-régulier.



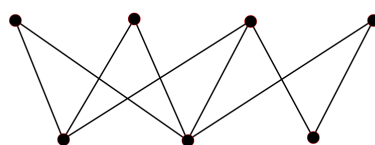
K-régulier

1.9.4 Graphe Biparti

Un graphe $G = (V, E)$ est dit biparti si l'ensemble de ses sommets V peut être partitionner en deux sous-ensembles V_1 et V_2 , de sorte que les sommets de même sous ensemble ne sont pas adjacents.

Exemple

Le graphe G ci-dessous est biparti.



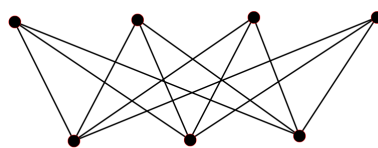
Graphe Biparti

Remarque

Un graphe biparti complet est un graphe biparti tel que tout sommet de V_1 est adjacent à tout les sommets de V_2 si $|V_1| = p$ et $|V_2| = q$. On le note $K_{p,q}$.

Exemple

Le graphe biparti complet $K_{3,4}$

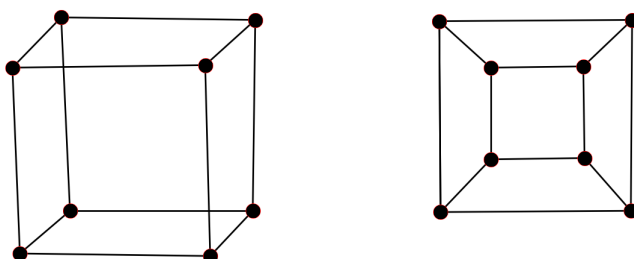


Biparti Complet

1.9.5 Graphe Planaire

Un graphe planaire est un graphe qui a la particularité de pouvoir se représenter sur un plan sans qu'aucune arête (ou arc pour un graphe orienté) n'en croise une autre. Autrement dit, ces graphes sont précisément ceux que l'on peut plonger dans le plan. Si l'on considère

le cube comme un graphe, une représentation de perspective classique ne sera pas planaire comme le montre le graphe sur la gauche. Le graphe sur la droite montre le même graphe sous un autre angle, mais cette fois-ci, il est bien planaire.



Graphe Planaire

1.9.6 Graphe Parfait

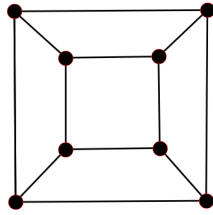
Dans le cadre de la théorie des graphes, Claude Berge a introduit, pour la première fois en 1960, la notion de graphe parfait. Un graphe parfait est un graphe pour lequel le nombre chromatique de chaque sous-graphe induit et la taille de la plus grande clique dudit sous-graphe induit sont égaux.

1.9.7 Graphe de l'hyper-cube

On appelle hyper-cube de dimension n , dont l'ensemble de ses sommets est l'ensemble des n -uplets $\{0, 1\}^n$, et tel que deux sommets sont adjacents si et seulement s'ils diffèrent exactement d'une seule composante. L'hypercube de dimension n , noté Q_n a 2^n sommets et $n2^{(n-1)}$ arêtes. Il est biparti et régulier.

Exemple

Le graphe Q_3



Graphe hyper-cube

1.9.8 Arborescence

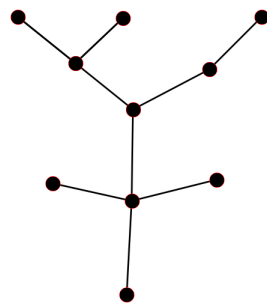
Arbre

Un graphe simple non orienté $G = (V, E)$ est un arbre s'il est connexe et sans cycle. Dans ce cas on le note $A = (V, E)$

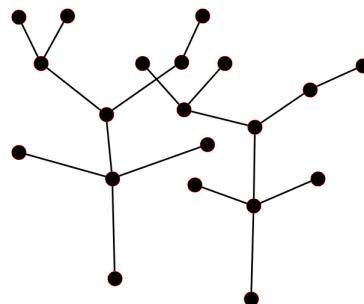
forêt

Une forêt est un graphe simple non orienté dont chaque composante connexe est un arbre.

Exemple



Un arbre



Une forêt

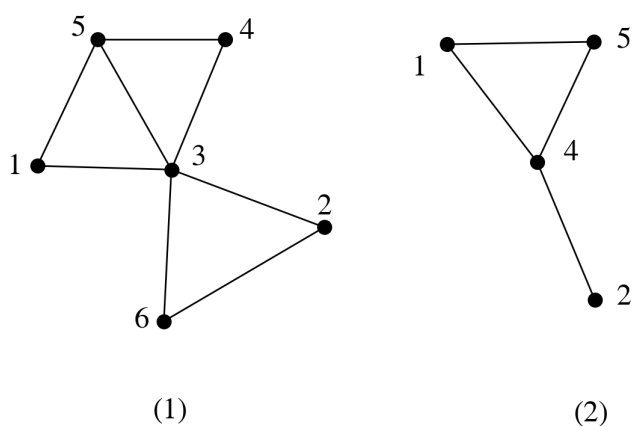
1.10 Graphe et Sous-Graphe

Un sous-graphe ou un graphe partiel est un graphe plongé dans le graphe en question. soit $G = (V, E)$

1.10.1 Sous-Graphe

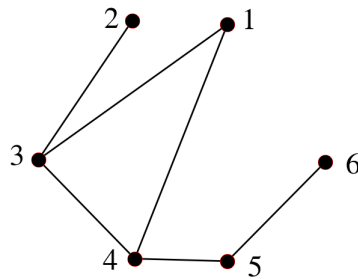
Le graphe $G' = (V', E')$ est le sous-graphe de $G = (V, E)$ induit par V' , si V' et $E' = \{(u, v) \in E, \forall u \in V', \forall v \in V'\}$. Autrement dit, un sous-graphe de $G = (V, E)$ est un graphe défini par un sous-ensemble V' de sommets et toutes les arêtes de G ayant les deux extrémités dans V' . On observe dans un sous-graphe une diminution du nombre de sommets et du nombre d'arêtes.

Exemple

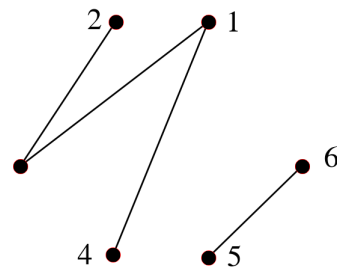


1.10.2 Graphe Partiel

Le graphe $G' = (V', E')$ est un graphe partiel de $G = (V, E)$, si $V' = V$ et $E' \subset E$. Dans ce cas on observe dans un graphe partiel une diminution du nombre d'arêtes et une conservation du nombre de sommets.

Exemple

Graphe G



Graphe partiel de G

1.10.3 Sous-Graphe Partiel

En combinant ces deux définitions, on peut définir un sous-graphe partiel. Il s'agit donc d'un graphe partiel d'un sous graphe. C'est concrètement un graphe au quel on a enlevé des sommets et leurs arêtes incidentes. Un sous-graphe partiel de $G = (V, E)$ est un graphe partiel d'un sous-graphe.

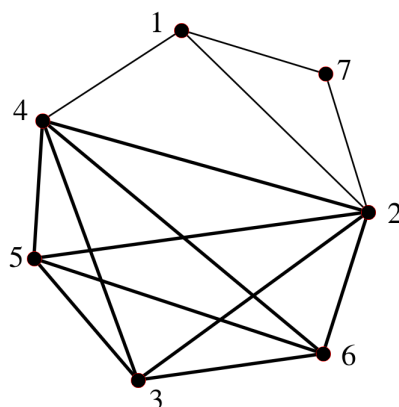
1.10.4 Clique

On appelle Clique un sous-graphe complet d'un graphe $G = (V, E)$.

Exemple

Dans le graphe suivant, on a :

- $C=(4,5,3,6)$ est une clique de ce graphe.
- $C=(4,5,3,6,2)$ est une clique max de ce graphe.

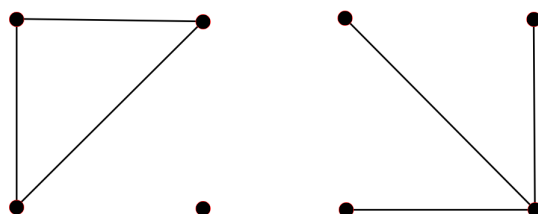


Clique

1.10.5 Graphe Complémentaire

Le graphe complémentaire de $G = (V(G), E(G))$ est le graphe dont l'ensemble des sommets est $V(G)$ et deux sommets distincts sont adjacents s'ils ne le sont pas dans G .

Exemple



Graphe Complémentaire

Remarque

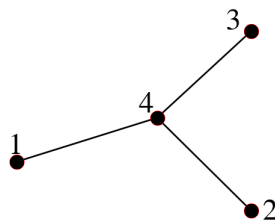
L'union d'un graphe et de son Complémentaire donne un graphe complet.

1.11 Connexité de Graphes

1.11.1 Graphe Connexe

Un graphe est dit connexe s'il existe pour chaque paire de sommets une chaîne les reliant. Autrement dit, on peut toujours passer d'un sommet à un autre soit directement soit en passant par un ou plusieurs autres sommets.

Exemple



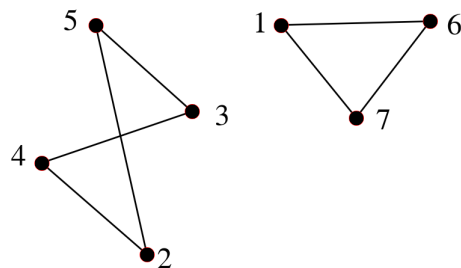
Graphe Connexe

1.11.2 Graphe Fortement Connexe

On définit un graphe fortement connexe comme étant un graphe orienté dont toutes les paires de sommets peuvent être reliées par un chemin.

1.11.3 Composantes Connexes

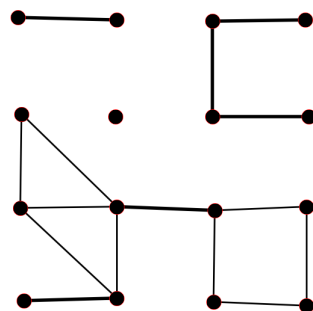
On appelle composantes connexes, les sous-graphe connexes. Il peut y avoir plusieurs composantes connexes dans un seul graphe. Cependant, la composante connexe est le plus grand sous-graphe que l'on puisse obtenir. Par conséquent, la composante connexe d'un graphe connexe est ce graphe lui même.

Exemple

Composantes Connexes

1.11.4 Isthme

Un isthme, ou un pont, est, en théorie des graphes, une aête d'un graphe dont l'élimination induit un graphe avec plus de composantes connexes que le graphe initial. De façon équivalente, une aête est un isthme si et seulement si elle n'est pas contenue dans un cycle.

Exemple

Isthme

1.12 Conclusion

Il est clair que ce chapitre n'est pas exhaustif concernant les définitions. On s'est limité aux généralités essentielles de la théorie des graphes. Dans le chapitre suivant nous allons reprendre quelques définitions en relation avec notre travail telle que la complexité algorithmique des problèmes de la théorie des graphes la représentation matricielle des graphes et bien d'autres¹.

1. Les définitions du chapitre 1 ont été tirées de Claude Berge et Bondy Murty

2

Problèmes Célèbres en Théorie des Graphes

2.1 Introduction

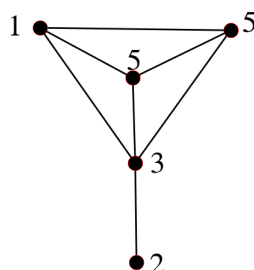
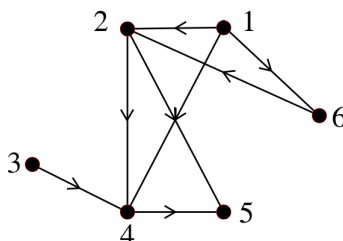
Ce chapitre fait l'objet, en premier, d'un complément de définitions relative aux problèmes célèbres en théorie des graphes, en suite on présente quelques problèmes célèbres en théorie des graphes et on apportera une attention particulière au problème de morphisme de graphe dans un but de développer une méthode basée sur le plus grand sous-graphe commun pour estimer le degré de similarité entre deux graphes, est un domaine largement exploité cette dernière décennie.

2.2 Représentation matricielles des graphes

2.2.1 Matrices d'Adjacences

On peut représenter un graphe simple par une matrice d'adjacences. Soit $G = (V, E)$ un graphe avec $|V| = n$ et $|E| = m$. On distingue deux représentations matricielles :

- **Matrice d'adjacence sommets-sommets** : Elle est notée M . Elle est de taille $(n \times n)$. La valeur $M(i, j)$ varie entre 1 et 0, selon que le sommet i soit adjacent au sommet j ou pas sur le graphe. Cette représentation est plus fréquente que la suivante.
- **Matrice d'adjacence sommets-arêtes** : Elle est noté M aussi. Elle est de taille $(n \times m)$. La valeur $M(i, j)$ varie entre 1 et 0, selon que le sommet i soit une des extrémités de l'arête j ou pas sur le graphe.

Exemple1**Exemple2****2.2.2 Caractéristiques de la Matrice d'Adjacence**

Cette matrice a plusieurs caractéristiques :

- C'est une matrice carrée (nombre de lignes=nombre de colonnes=nombre de sommets du graphe), ceci pour la matrice d'adjacence sommets-sommets. Pour la matrice d'adjacence sommets-arêtes, il y a autant de lignes que sommets et autant de colonnes que d'arêtes.

- Il n'y a que des 0 sur la diagonale. Un 1 sur la diagonale signifie l'existence d'une boucle dans le graphe, ceci pour la matrice d'adjacence sommets-sommets.
- La matrice d'adjacence sommets-sommets est symétrique ($M(i, j) = M(j, i)$) par rapport à la diagonale pour un graphe non orienté.
- Dans la matrice d'adjacence sommets-arêtes, $\sum_{j=1, m} M(i, j) = d_G(i)$.
- Une fois que l'on fixe l'ordre des sommets, il existe une matrice d'adjacences unique pour chaque graphe.

2.2.3 Opérations sur les Graphes

Dans ce qui suit lorsque nous parlons de matrice d'adjacence nous sous entendons la matrice d'adjacence sommets-sommets.

La manipulation des matrices représentatrices de graphes nécessite la maîtrise des opérations suivantes : union, intersection, produit et produit cartésien.

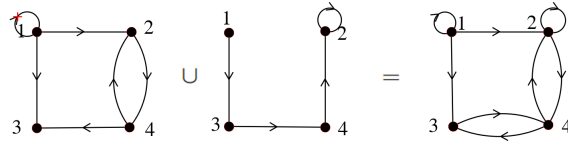
Soient les deux graphes $G_1 = (V, E_1)$ et $G_2 = (V, E_2)$, dont les matrices d'adjacence sont suivantes :

$$M_1 = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \quad M_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Union

L'union de deux graphes $G_1 = (V, E_1)$ et $G_2 = (V, E_2)$ est notée $G_{12} = (V, E_1) \cup (V, E_2) = (V, E_{12})$. Sur le plan matriciel ca donne :

$M_1 \cup M_2$, qui est définie par $(M_1 \cup M_2)[i, j] = M_1[i, j] \vee M_2[i, j]$.

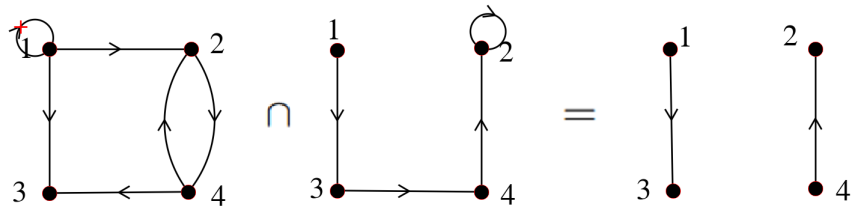
Exemple

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cup \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Intersection

L'union de deux graphes $G_1 = (V, E_1)$ et $G_2 = (V, E_2)$ est notée $G_1 \cap G_2 = (V, E_1) \cap (V, E_2) = (V, E_1 \cap E_2)$. Sur le plan matriciel ca donne :

$M_1 \cap M_2$, qui est définie par $(M_{12})[i, j] = M_1[i, j] \wedge M_2[i, j]$.

Exemple

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix} \cap \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Produit

Le produit de deux graphes $G_1 = (V, E_1)$ et $G_2 = (V, E_2)$ est notée

$$G_{12} = (V, E_1) \circ (V, E_2) = (V, E_1 \circ E_2)$$

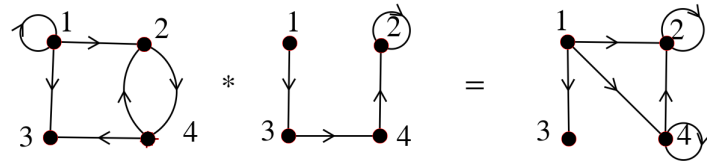
Sur le plan matriciel ca donne : M_{12} qui est défini par $M_1 \circ M_2$)[i, j], avec :

$$(M_1 \circ M_2)[i, j] = (\forall k : M_1[i, k] \wedge M_2[k, j])$$

Le produit des matrices est obtenu en faisant une opération similaire au produit matriciel en algèbre linéaire, en remplaçant la somme (+) par $\wedge$ et $\cdot$ par $\vee$. Rappelons que le produit matriciel en algèbre linéaire est défini par :

$$(M_1 M_2)[i, j] = (\sum_k M_1[i, k] \cdot M_2[k, j])$$

Exemple



On voit bien sur cet exemple que l'entrée (1, 2) est issue de

$$(1 \vee 0) \vee (1 \vee 1) \vee (1 \vee 0) \vee (0 \vee 1) \text{ qui est égale à } 1.$$

Produit Cartésien

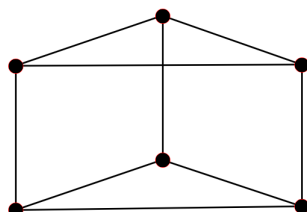
Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes. Le graphe notée $G_1 \boxtimes G_2$, appelée produit cartésien des deux graphes G_1 et G_2 , et tel que $V(G_1 \times G_2) = V_1(G_1) \times V_2(G_2)$ et (u, u') est adjacent à (v, v') si et seulement si :

$$\begin{cases} u=v \text{ et } u'v' \in E_2(G_2) \\ \text{ou bien} \\ u'v' \text{ et } uv \in E_1(G_1) \end{cases}$$

Exemple

Le graphe ci-dessous est le produit cartésien de P_2 et de K_3 , tel que :

P_2 est une chaîne de deux sommets et K_3 est un graphe complet de trois sommets



Après ce brèf rappel sur les graphes vus sous forme de matrices, nous allons entamer deuxième partie des problèmes célèbres en théorie des graphes.

2.3 Problèmes Célèbres en Théorie des Graphes

2.3.1 Stable

Considérons G , un graphe non orienté quelconque. Un ensemble stable de G est un ensemble de sommets non adjacents, qui forme donc un sous-graphe G' sans arêtes. Un ensemble stable est donc un sous-ensemble de sommets du graphe G tel que tout couple de sommets de ce sous-ensemble ne soit pas adjacents (voisins). Pour chaque couple de sommet de ce sous-ensemble, il ne doit donc pas exister d'arêtes les reliant. C'est donc l'opposé d'une clique qui, par définition est un sous-graphe tel que tout couple de sommets de ce sous-graphe soient reliés par une arête..

2.3.2 Couplage

Soit un graphe $G = (V, E)$. Un couplage M est un ensemble d'arêtes deux à deux non adjacentes : $\forall ((u_1, u_2), (v_1, v_2)) \in M, (u_1, u_2) \neq (v_1, v_2) \implies (u_1, u_2) \cap (v_1, v_2) = \{\}$ Un couplage maximal est un couplage M ayant la propriété que si une arête e est ajoutée, alors $M \cup e$ n'est pas un couplage. Un couplage maximum est un couplage contenant le plus grand nombre possible arête. Autrement dit un couplage dans un graphe G est un ensemble arête $E' \subset E$ tel que les arête de E' sont deux à deux non adjacentes. Un couplage est dit parfait si tout sommet de G est une extrémité d'une arête de E' .

2.3.3 Transversal

Un transversal d'un graphe est un sous-ensemble de sommets T tel que toute arête du graphe est incidente à au moins un sommet de T . Le complémentaire d'un transversal est un stable S .

2.3.4 Recouvrement

Un recouvrement d'un graphe est un sous-ensemble d'arête R tel que tout sommet du graphe est incidente à l'une des arêtes de R . Un couplage dans un graphe est un arête-recouvrement c'est un couplage parfait.

2.3.5 Problème de Coloration

La notion de coloration dans un graphe $G = (V, E)$ est une affectation de l'ensemble des couleurs $1, \dots, k$ à chaque sommet de G de telle sorte que, pour toute arête (uv) , $(uv) \in E$ de G , les sommets u et v ont chacun couleur différente. Un graphe est dit K -colorable s'il admet une K couleur.

Nombre Chromatique

On appelle nombre chromatique d'un graphe G noté $\chi(G)$ le plus petit nombre de couleurs nécessaire pour colorier les sommets du graphe G , de sorte que deux sommets adjacents ne portent pas la même couleur.

Polynôme Chromatique

Soit $G = (V, E)$ un graphe et $k \in \mathbb{N}^*$, on note $P(G, k)$ le nombre de coloration valide du graphe G en k couleurs, autrement dit les coefficients du polynôme chromatique représentent nombre de façon de colorer un graphe avec une liste de k -couleurs.

Indice Chromatique

Soit $G = (V, E)$ un graphe, on appelle indice chromatique $q(G)$ du graphe G le plus petit entier q qui a la propriété suivante : Il est possible avec q couleurs de colorier les arêtes du

graphe G , de sorte que deux arêtes adjacentes ne soient pas de la même couleur.

2.4 Problème du Plus Grand Sous-Graphe Commun

2.4.1 Définition

Soient deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$. $G_c = (V_c, E_c)$ est le plus grand sous-graphe commun des deux graphes G_1 et G_2 tel que :

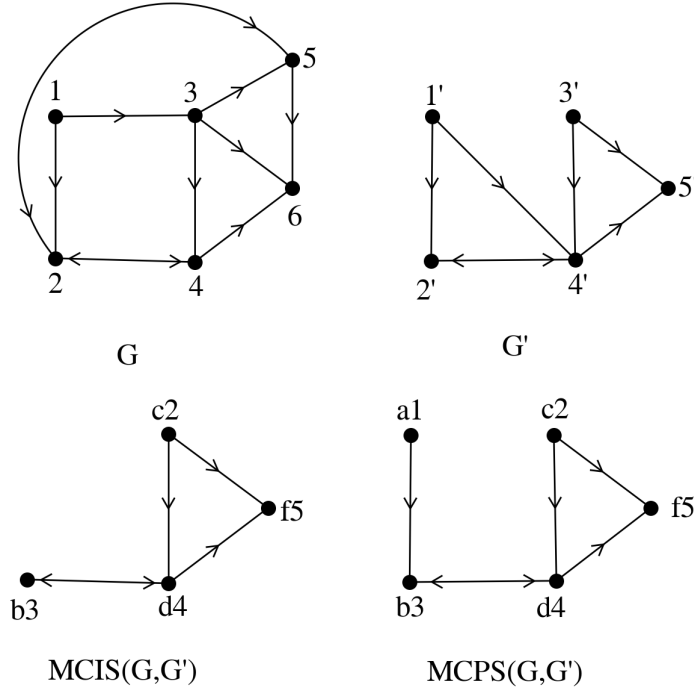
- $V_c \subseteq V_1 \cap V_2$.
- $E_c \subseteq E_1 \cap E_2$.

Sous-Graphe Induit

Le graphe $G' = (V', E')$ est un sous-graphe induit du graphe $G = (V, E)$ si $V_{G'} \subseteq V_G$ et $E_{G'} = E_G \cap (V_{G'} \times V_{G'})$. En d'autres termes, G' est obtenu à partir de G en retirant tout sommet de G qui ne se trouve pas dans $V_{G'}$, et en conservant uniquement les arêtes dont les deux extrémités se trouvent dans $V_{G'}$.

Sous-Graphe Partiel

Le graphe $G' = (V', E')$ est un sous-graphe partiel du graphe $G = (V, E)$, obtenu en supprimant des sommets et des arêtes de G . Le graphe G' est un sous-graphe partiel du graphe G si $V' \subseteq V$ et $E' \subseteq E \cap V' \times V'$.

Exemple**2.4.2 Domaine d'Utilisation**

La recherche d'un plus grand sous-graphe commun est un problème NP-difficile particulièrement complexe qui n'en reste cependant pas moins intéressant du fait de ses nombreuses applications en chimie, biologie ou encore en traitement d'images, lorsqu'il est nécessaire de mesurer la similarité d'objets représentés par des graphes. La similarité de deux graphes est généralement définie par rapport à la taille d'un sous-graphe commun [24] : plus ce sous-graphe commun est grand, plus les graphes sont similaires. Un sous-graphe commun est un graphe isomorphe à des sous-graphes de G et G' . La taille d'un sous-graphe est définie différemment selon que l'on considère des sous-graphes induits ou partiels : un plus grand sous-graphe partiel (MCPS) est un sous-graphe partiel qui a le plus grand nombre d'arêtes, tandis qu'un plus grand sous-graphe induit (MCIS) est un sous-graphe induit qui a le plus grand nombre de sommets. Le calcul du degré de similarité de deux graphes a fait l'objet de nombreuses études cette dernière décennie. De multiples mesures ont été développées, nous citons celle donnée par Bunke et Kshear en 1998 [6] ($sim(G_1, G_2) = 1 - \frac{|PGSC(G_1, G_2)|}{\max\{|G_1|, |G_2|\}}$). Dans ce travail nous nous intéressons à

la recherche du plus grand sous-graphe commun entre deux graphes.

2.4.3 Algorithme Proposé pour la Recherche du PGSGC

Ces dernières années de nombreuses approches ont été développées pour la recherche plus grand sous-graphe commun. Samba Ndojh Ndiaye et Christine Solnon[1] ont proposé une approche basée sur la programmation par contraintes. Et avec Mael Minot [2], ils ont proposé une approche basée sur décomposition du graphe de compatibilité. D'autres méthodes basées sur la clique maximum ont été proposées. Nous nous sommes inspirés de ces dernières pour concevoir notre algorithme. Notre algorithme est composé de deux étapes principales :

- la première étape consiste à trouver la plus petite clique des cliques max des deux graphes en question.
- la seconde consiste à ajouter des sommets à cette plus petite clique de sorte à ce que le graphe obtenu admette un sous-graphe isomorphe dans les deux graphes de départ.

La recherche d'une clique max dans un graphe est un problème largement étudié [24, 25, 20, 21]. Dans notre étude, nous avons utilisé la fonction pré-définie dont le pseudo algorithme est donné par :

Algorithme 1 Algorithme de recherche de Clique max

```

1 : function MaxClique ( $G, K, C$ )
2 : if  $C = \emptyset$  then noter la clique maximale  $K$ 
3 : else for each  $x \in C$  do  $MaxClique(G, K \cup \{x\}, C \cap \Gamma(x))$ 
4 : end function

```

Vu la complexité du problème il nous est difficile d'appliquer cet algorithme aux problèmes de taille importante en particulier ceux danses. En effet, dans ce cas il se peut que plusieurs cliques puissent exister dans un seul graphe, alors pour initialiser l'algorithme il faudra essayer tous les cas possibles. Aussi pour des raisons de complexité, avant de lancer l'algorithme il faut s'assurer que les deux graphes en question ne doivent pas être isomorphes.

2.5 Conclusion

L'intérêt de ce chapitre est de présenter quelques problèmes célèbres de la théorie des graphes, essentiellement de présenter le problème de recherche du plus grand sous graphe commun de deux graphes et de proposer un algorithme de recherche. Le chapitre suivant fera l'objet d'implémentation de l'algorithme proposé¹.

1. Les définitions du chapitre 2 a été tiré de Claude Berge et bondy Murty

3

Implémentation de l'Algorithme

3.1 Introduction

Dans ce chapitre nous allons presenter l'environnement de programmation en premier. En suite, une serie de resultats sera affiche. Et nous terminerons le chapitre par une une petite analyse des résultats.

3.2 Langage utilisé

3.2.1 Choix du Langage

Notre choix s'est porté sur Matlab, pour les raisons suivantes :

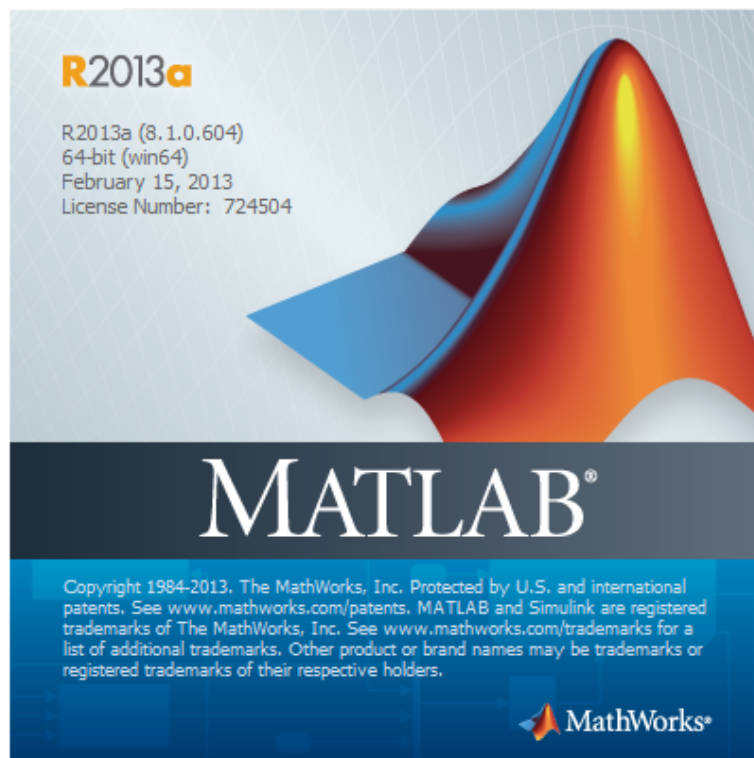
- La maniabilité du langage : constitué d'un ensemble de possibilités faisant en sorte que le programmeur travaille avec aisance, assuré d'une part par la syntaxe du langage et d'autre part, par un aspect visuel clair représentatif à la fois du détail et du global.

- Le bagage du langage : il contient une interface graphique puissante ainsi qu'une grande variété de méthodes scientifiques implémentées.

3.2.2 Généralités sur MATLAB

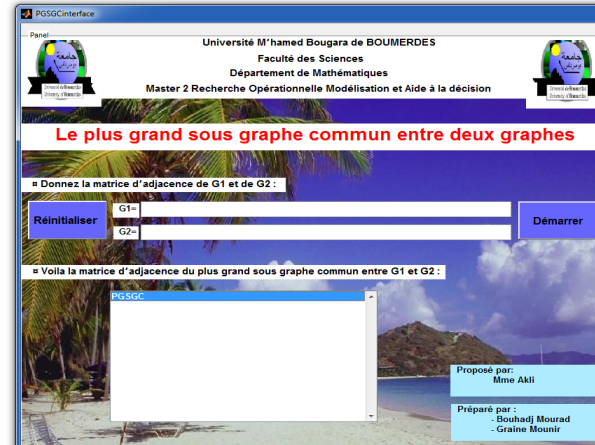
MATLAB est un logiciel parfaitement dédié à la résolution de problèmes d'analyse numérique ou du traitement du signal. Il Permet d'effectu  des calculs matricielles ou de visualiser les r sultats sous forme graphique. La formulation des probl mes s'apparente   la formulation math matique des probl mes   r soudre. L'utilisation de ce logiciel consiste   lancer des lignes de commandes, qui peuvent le plus souvent ressembler   la programmation en C.

Le nom de MATALB vient de MATrix LABoratory, les  l ments de donn es de base manipul s par Matlab  tant des matrices (mais pouvant  videmment se r duire   des vecteurs et des scalaires) qui ne n cessitent ni dimensionnelle ni d claration de type. Contrairement aux langages classiques, les fonctions du Matlab permettent de manipuler directement et de mani re interactive ces donn es matricielles, le rendant aussi particuli rement efficace en calcul num rique, analyse et visualisation de donn es.



3.3 Présentation de l'application :

Dans le but de faire une interface conviviale et simple à utiliser, on a opté pour le GUI (Guide User Interface) de Matlab. Une fenêtre d'accueil s'affiche au début de l'exécution. On aura cette interface :

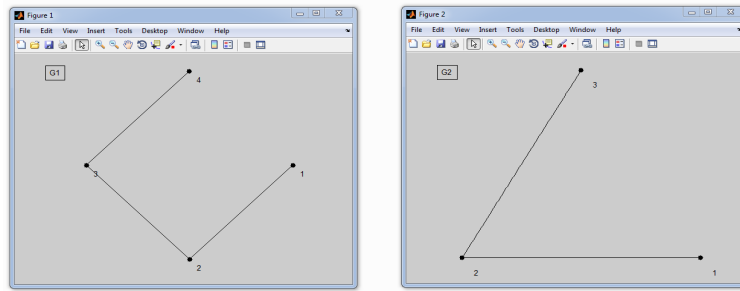


Comme c'est indiqué sur fenêtre d'accueil on clique sur Bouton **"Réinitialiser"** puis on introduit la matrice d'adjacence de G_1 et G_2 dans les cases correspondantes, ensuite on clique sur le bouton **"Démarrer"**. Ce dernier fait l'appel à notre application afin d'avoir le graphe correspondant à G_1 et G_2 , et pour avoir le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence.

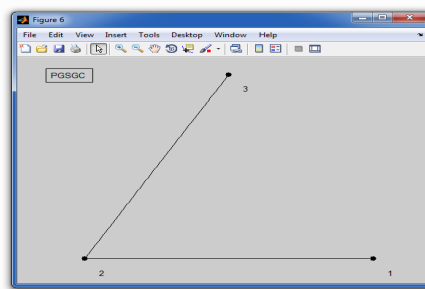
3.4 Exemples d'applications

3.4.1 Exemple de Deux Chaînes

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



Dans ce cas, il est clair que le plus grand sous graphe commun entre deux chaînes est la chaîne de la cardinalité minimale. Le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence sont :

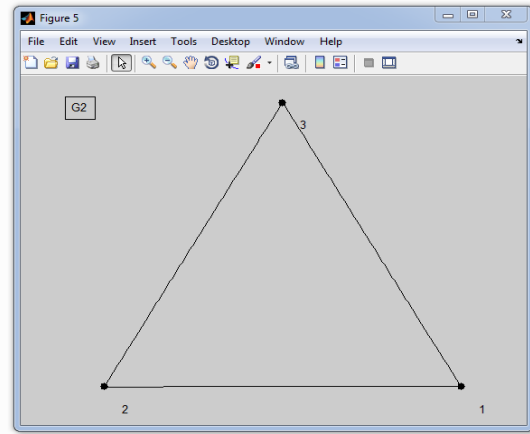
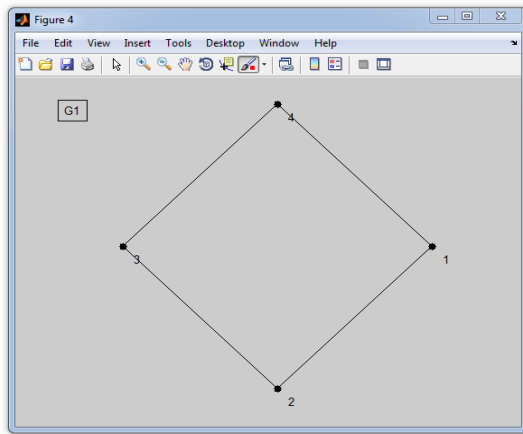


0	1	0
1	0	1
0	1	0

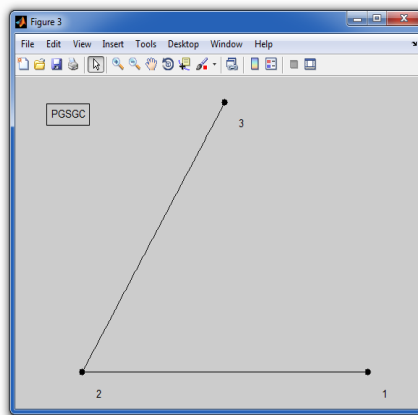
3.4.2 Exemple de Deux Cycles

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

Dans ce cas, on a introduit la matrice d'adjacence de G_1 et G_2 , on remarque facilement



que le plus grand sous graphe commun entre un graphe carré G_1 et un graphe triangle G_2 est une chaîne de trois sommets. Le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence sont :



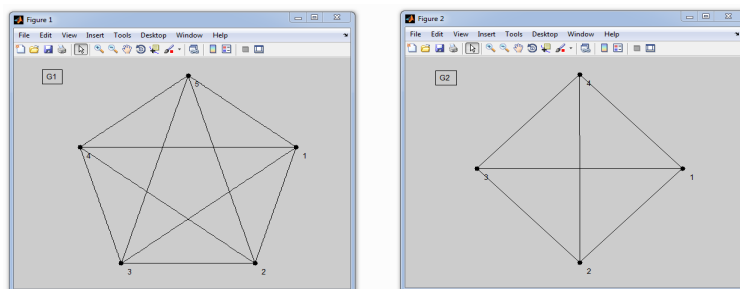
```

0 1 0
1 0 1
0 1 0

```

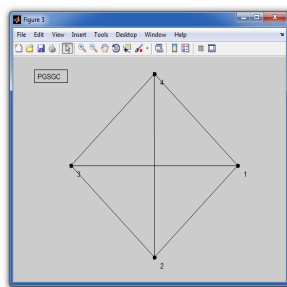
3.4.3 Exemple de Deux graphes complets

$$G_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$



On sait tous que le plus grand sous graphe commun entre le k_5 et k_4 est un k_4 , se qui est montré par l'exemple.

Le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence :



```

0 1 1 1
1 0 1 1
1 1 0 1
1 1 1 0

```

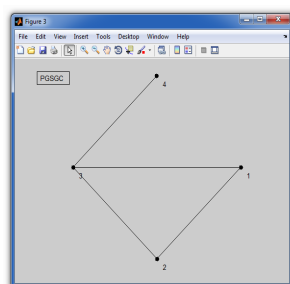
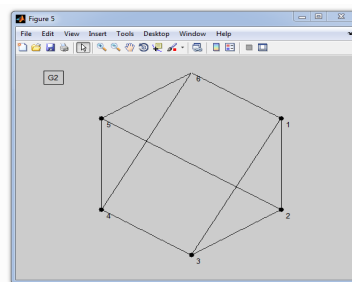
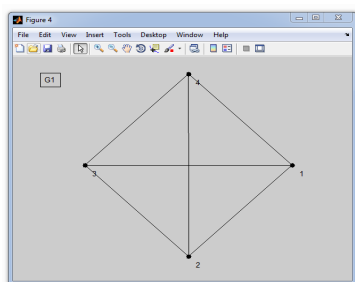
3.4.4 Exemples de Deux graphes réguliers

1. Différent nombre de sommet et même régularité.

$$G_1 = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence :

2. Même nombre du sommet et différent régularité



0	1	1	0
1	0	1	0
1	1	0	1
0	0	1	0

$$G_1 = \begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Le plus grand sous graphe commun entre G_1 et G_2 et sa matrice d'adjacence sont :

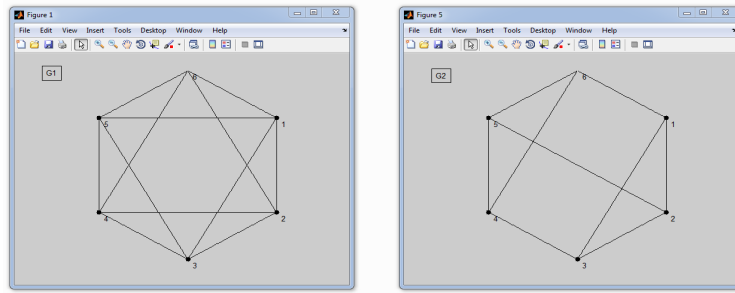
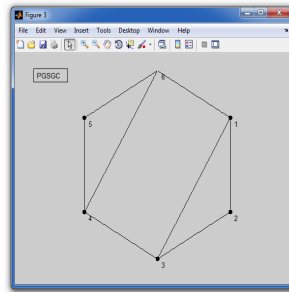


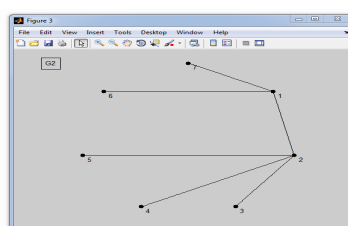
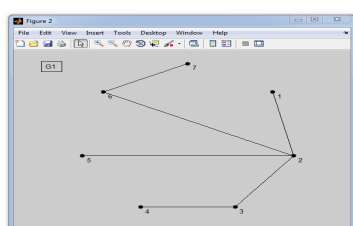
FIGURE 3.1 –



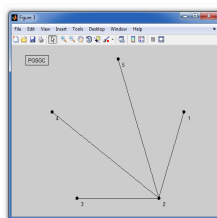
0	1	1	0	0	1
1	0	1	0	0	0
1	1	0	1	0	0
0	0	1	0	1	1
0	0	0	1	0	1
1	0	0	1	1	0

3.4.5 Exemple de Deux arbres

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$



Le plus grand sous graphe commun entre G_1 et G_2 et sa matrice d'adjacence :



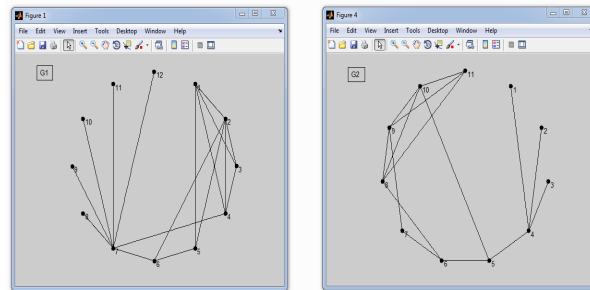
0	1	0	0
1	0	1	1
1	1	0	0
0	1	0	0
0	1	0	0

3.4.6 Exemple de Deux Graphes Quelconques

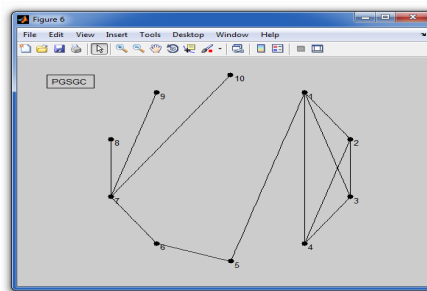
1. 1^{er} cas

$$G_1 = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & & & & & & & & \\ 0 & 0 & & & & & & & & \end{pmatrix}$$

$$G_2 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & & & \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & & & & & & & & & \end{pmatrix}$$



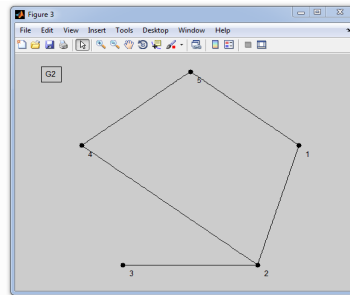
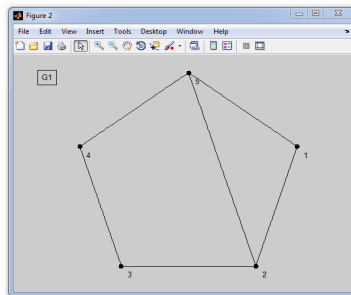
Le plus grand sous graphe commun entre G_1 et G_2 avec sa matrice d'adjacence :



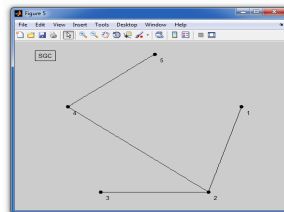
0	1	1	1	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0
1	1	0	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0
0	0	0	0	1	0	1	0	0	0
0	0	0	0	1	0	1	1	1	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0

2. 2^{ème} cas

$$G_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix} \quad G_2 = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$$



Un sous graphe commun avec sa matrice d'adjacence



0	1	0	0	0
1	0	1	1	0
0	1	0	0	0
0	1	0	0	1
0	0	0	1	0

3.4.7 Conclusion

Nous voyons bien que cet algorithme fonctionne parfaitement pour quelques classes de graphes. Avec cette application on a la matrice d'adjacence et le graphe. Néanmoins, il reste impuissant face aux graphes de dimension importante et aux graphes contenant plus d'une seule clique. Il sera intéressant de développer ces derniers cas.

Conclusion générale

La recherche du plus grand sous-graphe commun entre deux graphes est un problème NP-difficile, Dans ce travail nous nous sommes intéressés à la recherche du plus grand sous-graphe commun induit.

Les résultats obtenus sont satisfaisants dans la mesure où l'algorithme proposé permet d'afficher, après insertion des matrices d'adjacences des graphes en question, le plus grand sous-graphe commun induit.

Néanmoins les graphes insérés doivent être des graphes simple de taille relativement petite. Et la chose la plus importante est que les graphes insérés ne doivent pas être isomorphe. Jusqu'à présent nous nous contentons de graphes contenant une seule clique de cardinalité maximale.

Nous jugeons, qu'il soit intéressant compléter cette étude par la résolution des lacunes citées précédemment pour avoir une étude exhaustive.

Bibliographie

- [1] Samba Ndojh Ndiaye et Christine Solnon, *Modélisation et résolution de problèmes de plus grands sous-graphes communs à l'aide de la programmation par contraintes*. Université de Lyon, 2011.
- [2] Mael Minot, Samba Ndojh NDIAYE, Christine SOLNON *Recherche d'un plus grand sous graphe commun par décomposition du graphe de compatibilité*. France Université Lyon 1, 2015.
- [3] Sébastien Sorlin, *Mesurer la similarité de graphes*. Université Claude Bernard Lyon I, 2006 .
- [4] Claude Berge, *Graphes et hypergraphes*. Université Dunod, 1973.
- [5] F.Affif Chaouche, *Problèmes d'existence en structure discrètes*. ,Thèse de doctorat, Université USTHB, 2008.
- [6] H.Bunke and K.Sharer, *A graph distance metric based on the maximal common subgraph*. Université Pattern Recognition Letters, 1998.
- [7] D.Conte and P.Foggia and M.Vento, *Challenging complexity of maximum common subgraph detection algorithms : A performance analysis of three algorithms on a wide database of graphs*. Graph Algorithms and Applications, 2007.

- [8] P.J.Durand, R.Pasari, J.W.Baker, and C.Tsai, *An efficient algorithm for similarity analysis of molecules*, Internet Journal of Chemistry, 1999.
- [9] J.J.McGrego, *Backtrack search algorithms and the maximal common subgraph problem.* , Software Practice and Experience, 1982.
- [10] J.W.Raymond, E.J.Gardiner, and P.Willett, *Calculation of graph similarity using maximum common edge subgraphs*, The Computer Journal, 2002.
- [11] J.W.Raymond and P.Willett, *Maximum common subgraph isomorphism algorithms for the matching of chemical structures*, Journal of computeraided molecular design, 2002.
- [12] C.Solnon, *Alldifferent-based filtering for subgraph isomorphism*, Artificial Intelligence, 2010.
- [13] P. Vismara and B. Valery, *Finding maximum common connected subgraphs using clique detection or constraint satisfaction algorithms*, Communications in Computer and Information Science, 2008.
- [14] E.Balas and C.S.Yu, *Finding a maximum clique in an arbitrary graph*, SIAM Journal on Computing, 1986.
- [15] H.G.Barrow and R.M.Burstall, *Subgraph isomorphism, matching relational structures and maximal cliques*, Information Processing Letters, 1976.
- [16] Donatello Conte, Pasquale Foggia, and Mario Vento, *Challenging complexity of maximum common subgraph detection algorithms : A performance analysis of three algorithms on a wide database of graphs*, Graph Algorithms Appl, 2007.
- [17] Matjaz Depolli, Janez Konc, Kati Rozman, Roman Trobec, and Dusanka Janezic, *Exact parallel maximum clique algorithm for general and protein graphs*, Journal of

chemical information and modeling, 2013.

- [18] j.P.Durand, Rohit Pasari, Johnnie W Baker, and Chun-che Tsai, *An efficient algorithm for similarity analysis of molecules*, Internet Journal of Chemistry, 1999.
- [19] I.Koch, *Enumerating all connected maximal common subgraphs in two graphs*, Theoretical Computer Science, 2001.
- [20] Ciaran McCreesh and Patrick Prosser, *Multithreading a state-of-the-art maximum clique algorithm*, Algorithm, 2013.
- [21] Ciaran McCreesh and Patrick Prosser, *The shape of the search tree for the maximum clique problem*, and the implications for parallel branch and bound, 2014.
- [22] J.James McGregor, *Backtrack search algorithms and the maximal common subgraph problem*, Software : Practice and Experience, 1982.
- [23] Samba Ndojh Ndiaye and Christine Solnon, *Cp models for maximum common subgraph problems*, In Principles and Practice of Constraint Programming?CP 2011, pages 637 ?644. Springer,2011.
- [24] J.W.Raymond, E.J.Gardiner, and P.Willett Rascal, *calculation of graph similarity using maximum common edge subgraphs*, The Computer Journal, 45(6) : 631 ?644, 2002.
- [25] Christine Solnon, *Solving maximum clique problems with ant colony optimization* liris.cnrs.fr/csolnon/antclique.html, 2006.
- [26] Christine Solnon and Serge Fenet, *A study of ACO capabilities for solving the maximum clique problem*, Heuristics, 12(3) : 155 ?180, 2006.

- [27] Philippe Vismara, *Programmation par contraintes pour les problemes de plus grand sous-graphe commun*, Journees Francophones de Programmation par Contraintes, pages 327-335, 2011.

NB : L'ensemble de la documentation a été téléchargée sur www.google.com Les définitions des chapitres 1 et 2 ont été tiré de Claude Berge et bondy Murty