

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

UNIVERSITÉ MOULOU D MAMMERI, TIZI-OUZOU

FACULTÉ DES SCIENCES

DÉPARTEMENT DE MATHÉMATIQUES



**THÈSE DE DOCTORAT ES-SCIENCES**

**Filière : Mathématiques**

**Spécialité : Recherche Opérationnelle**

Présentée par :

Djamel TALEM

Sujet :

**CALCUL D'INVARIANTS DANS LES GRAPHE ET ORDRES**

Devant le jury d'examen composé de :

AIDENE Mohamed	Professeur	UMMTO	Président
SADI Bachir	Professeur	UMMTO	Directeur de thèse
AIDER Meziane	Professeur	USTHB	Examineur
BOUCHEMAKH Isma	Professeur	USTHB	Examinatrice
OUKACHA Brahim	Professeur	UMMTO	Examineur
SLIMANI Hachem	Professeur	Univ-Bejaia	Examineur

**Année universitaire 2022-2023**

## Remerciements

---

Tout d'abord, je remercie **Allah**, le tout puissant de m'avoir donné une bonne santé, assez du courage, de patience et de volonté pour achever cette thèse.

Je remercie mon encadrant, Professeur SADI Bachir, pour son encadrement et accompagnement durant tout ce rude et long travail. Je lui adresse mon expression de gratitude eut égard à son aide et apprentissage dont j'ai bénéficié de sa part, mais aussi pour sa patience, son ouverture d'esprit, sa confiance qu'il a su me témoigner et la disponibilité dont il a tant fait preuve.

J'adresse mes plus sincères remerciements au professeur AIDEN Mohamed qui me fait l'honneur de présider le jury de ma soutenance et aux professeurs : AIDER Meziane, BOUCHEMAKH Isma, OUKACHA Brahim et SLIMANI Hachem qui me font l'honneur de faire partie de ce jury.

Je remercie également toutes les personnes avec qui j'ai eu l'occasion de collaborer depuis le début de ma carrière; je citerai en particulier HADJ ALI Abdelkader avec qui j'ai beaucoup travaillé et eu des échanges de connaissances fructueuses dans pratiquement tous les domaines, en particulier dans le domaine des mathématiques. Sans oublier tous les membres du laboratoire LAROMAD sans exception pour tous ces bons moments passés cette dernière année, ainsi que pour tout le respect qu'ils ont éprouvé pour moi.

J'adresse un grand merci à ma famille : ma mère, ma femme, mes enfants mon frère, mes sœurs et tous mes amis et proches pour leur soutien depuis tant d'années...

## Résumé

---

Dans cette thèse, nous nous sommes intéressés au calcul des invariants suivants : décomposition minimum d'un graphe biparti en bicliques, chaîne déviation d'un ensemble partiellement ordonné (ou chaîne déviation d'un poset) et  $P$ -couplage. Les invariants de graphes (resp. des ordres) sont des paramètres qui caractérisent les graphes (resp. les ordres) et dont la valeur est la même pour tous les graphes (resp. tous les ordres) qui sont isomorphes. Deux graphes ou deux ordres sont isomorphes lorsqu'il existe une bijection entre leurs ensembles de sommets conservant les relations d'adjacences et de comparabilités respectivement. On peut citer, à titre d'exemples, l'invariant chromatique dans les graphes et qui est donné par le nombre minimum de couleurs nécessaires sans attribuer la même couleur aux sommets voisins ; L'invariant de stabilité qui est donné par le nombre maximum de sommets deux-à-deux non voisins ; l'invariant de la dimension dans les posets qui est donné par le nombre minimum d'extensions linéaires dont l'intersection coïncide avec le poset en question...

Les problèmes de calcul d'invariants occupent une place très importante en pratique. La résolution des problèmes modélisés par les graphes ou posets dans des différents domaines se ramènent souvent aux problèmes de calcul d'invariants. Malheureusement, en termes de la complexité algorithmique, même si la formulation de ces problèmes est facile, la plupart d'entre eux sont NP-complet, c'est-à-dire les situations où nous disposons d'algorithmes « efficaces » pour les résoudre sont limitées, c'est le cas pour les problèmes du stable maximum, de la coloration minimum, de la dimension d'un ensemble ordonné, de la couverture minimum des sommets et d'un ensemble dominant minimum, etc.

Les invariants abordés dans ce document sont :

1. Énumération de bicliques maximums dans les graphes bipartis ;
2. Calcul d'une clique maximum pour les sommets et une clique maximum pour les arêtes dans un graphe biparti ;
3. Couverture des arêtes et couverture des sommets par des bicliques dans un graphe biparti ;
4. Chaîne déviation d'un poset ;
5. Couplage généralisé ou  $P$ -couplage.

Nos contributions sont résumées comme suit :

Le problème de la décomposition d'un graphe biparti en bicliques est aussi connu pour être difficile en général. En utilisant le parcours en largeur lexicographique (LexBFS), nous proposons d'abord un algorithme linéaire pour la reconnaissance de la classe des graphes bipartis distance héréditaire, nous proposons aussi des algorithmes

linéaires pour calculer une biclique maximum, une partition minimum des arêtes en biclique et une couverture minimum des sommets en bicliques pour la même classe de graphes.

Le problème de la chaîne déviation d'un poset, introduit par Kong et Ribemboin en 1984, est défini comme suit : pour un poset  $P$ ,  $(D^i(P))_{i \geq 0}$  est une suite de posets définies à partir de  $P$  d'une manière récursive, où  $P = D^0(P)$  et pour  $0 < i$ ,  $D^i(P)$  est l'ordre strict sur les antichaînes maximales du poset  $D^{i-1}(P)$ . Les mêmes auteurs ont montré que cette séquence ainsi définie converge vers un ordre total après un nombre fini d'itérations. La notation  $cdev(P)$  désigne le plus petit entier naturel pour lequel  $D^{cdev(P)}(P)$  est une chaîne. Pour un poset  $P$  qui n'est, ni antichaîne ni somme linéaire d'autres posets, nous montrons que la valeur du paramètre  $cdev(P)$  est la distance dans le graphe  $Inc(P)$ , le graphe d'incomparabilité de  $P$ , entre deux sommets additionnels  $0_P$  et  $1_P$ , où  $0_P$  adjacent à tous les éléments minimaux de  $P$  et  $1_P$  adjacent à tous les éléments maximaux de  $P$ . Nous donnons aussi sa valeur pour les deux cas particuliers cités ci-dessus. Enfin, nous montrons que "cdev" est un invariant de comparabilité, c'est-à-dire les posets ayant un même graphe de comparabilité ont le même "cdev".

Quant au problème du  $P$ -couplage, il consiste en la généralisation de la notion du couplage comme suit : étant donné un poset  $P = (E, \leq_P)$ , un graphe biparti  $G = (X, Y, E')$  tels que  $|E| = |E'|$  et une bijection  $f : E \rightarrow E'$ . Un  $P$ -couplage dans  $G$  est un ensemble d'arêtes  $M$  vérifiant  $f(A) \cap M$  est un couplage pour toute antichaîne de  $P$ . Le problème est de trouver un  $P$ -couplage avec un maximum d'arêtes. La complexité de ce problème est inconnue à l'heure actuelle. Nous donnons une condition nécessaire et suffisante pour qu'un ensemble d'arêtes soit un  $P$ -couplage, et nous montrons comment calculer efficacement un  $P$ -couplage maximum pour quelques classes de posets et de graphes très particulières.

**Mots Clés :** Ensemble partiellement ordonné, Décomposition d'un graphe biparti en bicliques, Biclique maximale, Chaîne maximale, Antichaine maximale.

## Abstract

---

In this thesis, we focus on the calculation of the following invariants : decomposition of a bipartite graph into bicliques, invariant computation in a poset and  $P$ -matching. Graph (resp. poset) invariants are parameters that characterize graphs (resp. posets) and whose value is the same for all graphs (resp. posets) that are isomorphic. The formulation of invariant calculation problems is often easy, but their resolution, in general, is far from the same. Situations where we have “efficient” algorithms to solve these problems are rare, in most cases such an algorithm does not exist (or at least the scientific community agrees that it does not exist), this is the case for the problems of maximum independent set, minimum coloring problem, dimension of a poset, etc. Our contributions are summarized as follows :

The problem of decomposition of a bipartite graph into bicliques is also known to be difficult in general. Using the lexicographic width search algorithm, we first propose a linear algorithm for the recognition of the class of distance hereditary bipartite graphs ; we also propose linear algorithms to calculate a maximum biclique, a minimum biclique edge partition and a minimum biclique cover for the same graph class.

Invariant computation in a poset is introduced by Kong and Reimboin : let  $P$  be a poset. We define an order on  $D(P)$ , the set of maximal antichain of  $P$ , as follows : for  $A, B \in D(P)$ ,  $A <_{D(P)} B$  if and only if  $\forall a \in A, \exists b \in B$  such that  $a <_P b$ . In the same way, we define an order on the set of maximal antichains of  $D(P)$ , and so on. At the end, we will have built a sequence of orders  $P, D(P), D^2(P), \dots, D^i(P), \dots$  where  $D^i(P) = D(D^{i-1}(P))$ . They proved that for a finite poset  $P$ , the sequence so constructed converges to a chain, i.e. there exists a natural integer  $i$  such that  $D^i(P)$  is a total order. For a finite poset  $P$  which is neither an antichain nor a linear sum, we show that the value of the smallest natural integer noted  $cdev(P)$  is given by the distance in the incomparability graph of  $P$  between two additional, artificial vertices  $0_P$  and  $1_P$ , where  $0_P$  is adjacent to all minimal elements of  $P$  and  $1_P$  is adjacent to all maximal elements of  $P$ . We also show how to handle the latter two particular cases. Finally, we show that “ $cdev$ ” is a comparability invariant.

The  $P$ -matching consists in generalization the concept of matching as follows : given a poset  $P = (E, \leq_P)$ , a bipartite graph  $G = (X, Y, E')$  such that  $|E| = |E'|$  and a bijection  $f : E \rightarrow E'$ . A  $P$ -matching in  $G$  is a set of  $M$  edges verifying  $f(A) \cap M$  is a matching for any antichain of  $P$ . The problem is to find a maximum  $P$ -matching. The complexity of this problem is unknown. In this thesis, we give a necessary and sufficient condition for a set of edges to be a  $P$ -matching, and show how to calculate a maximum  $P$ -matching for some very specific posets and graph classes.

**Keywords :** Partially ordered set, Chain, Antichain, decomposition of bipartite graph into bicliques, maximal biclique

# Table des figures

1.1	Deux graphes isomorphes . . . . .	7
1.2	Représentation d'un poset par un tableau . . . . .	9
1.3	Graphes de comparabilité et d'incomparabilité d'un poset . . . . .	10
1.4	Diagramme de Hasse d'un poset . . . . .	11
1.5	Un poset $P$ et son dual $P^d$ . . . . .	12
1.6	Graphe ordonné par LexBFS. . . . .	14
1.7	Graphe biparti $G$ ordonné par LexBFS. . . . .	17
1.8	La restriction d'un LexBFS . . . . .	17
1.9	Domino . . . . .	24
1.10	Graphe biparti de permutation. . . . .	25
1.11	Graphe $X$ -convexe . . . . .	26
1.12	A hierarchy of bipartite graph classes. . . . .	26
2.1	Graphe $G(M)$ d'une matrice symétrique $M$ . . . . .	31
2.2	Arête bisimpliciale . . . . .	32
2.3	Graphe sans schéma d'élimination parfait . . . . .	34
2.4	$GBEF$ qui n'est pas triangulé . . . . .	35
2.5	Sommet bisimplicial . . . . .	37
2.6	Suppression des arêtes d'une biclique maximale . . . . .	40
2.7	Graphe $G$ et son carré . . . . .	41
2.8	Cas $N^+(x) \neq N^+(x')$ . . . . .	42
2.9	(a) $v' < x$ ; (b) $x < v'$ . . . . .	43
2.10	Deux cas : $v' \in \mu^+[x', v_n]$ ou $v' \notin \mu^+[x', v_n]$ . . . . .	44
2.11	Domino et sommet bisimplicial . . . . .	45
2.12	Séquence de graphes calculés selon la définition 2.9 . . . . .	46
2.13	Tester si l'ordre sur les sommets est un LexBFS . . . . .	50
3.1	Biclique sommet maximum et biclique arête maximum. . . . .	53
3.2	Graphe biparti, ordonné par LexBFS, et ayant 3 bicliques maximales . . . . .	54
3.3	Énumération des Bicliques maximales. . . . .	59
3.4	Graphe vérifiant $bc(G) = 2 < bp(G) = 3$ . . . . .	62
3.5	La séquence de graphes $H_1, H_2, H_3$ obtenus selon la définition 3.3 . . . . .	64
3.6	Indépendance des arêtes est une propriété héréditaire . . . . .	65
3.7	Partition des arêtes en bicliques . . . . .	68

3.8	Partition des arêtes en bicliques (suite)	68
3.9	Partition des arêtes en bicliques (suite)	69
3.10	Graphe biparti avec $bp = 4$	69
4.1	Les antichaines $A$ et $B$ sont disjointes	75
4.2	Séquence de posets développée à partir de $P$ .	76
4.3	Somme linéaire $P_1 \oplus P_2$ des posets $P_1$ et $P_2$ .	76
4.4	Graphes associés au poset $P$	77
5.1	Éléments d'une instance d'un problème du $P$ -couplage	83
5.2	Réduction d'un problème du $P$ -couplage au problème de la clique	84
5.3	Trou de longueur 7	85
5.4	Le graphe $H$ est sans $C_{2k+1}, k \geq 3$	85
5.5	Le graphe $H$ est sans $C_{2k+1}, k \geq 3$	86
5.6	Le graphe $H$ est sans $C_{2k+1}, k \geq 3$	86
5.7	Le graphe $H$ est sans $C_{2k+1}, k \geq 3$	87

# Table des matières

<b>Table des matières</b>	<b>iv</b>
<b>Introduction</b>	<b>1</b>
<b>1 Préliminaires</b>	<b>4</b>
1.1 Définitions et notations . . . . .	4
1.1.1 Graphe et digraphe . . . . .	4
1.1.2 Ensemble ordonné . . . . .	8
1.1.3 Parcours de graphes . . . . .	12
1.1.4 Graphe biparti ordonné . . . . .	14
1.2 Problème, algorithme, complexité . . . . .	18
1.2.1 Problèmes de décision et d'optimisation . . . . .	18
1.2.2 Complexité d'un algorithme . . . . .	19
1.2.3 Classes de problèmes $\mathcal{P}$ et $\mathcal{NP}$ . . . . .	20
1.2.4 La classe $\mathcal{NP}$ -complet . . . . .	21
1.3 Quelques classes de graphes . . . . .	23
1.3.1 Graphe triangulé . . . . .	23
1.3.2 Graphe distance héréditaire . . . . .	24
1.3.3 Graphe de permutation . . . . .	25
1.3.4 Graphe convexe . . . . .	25
<b>2 Reconnaissance des graphes bipartis distance héréditaire</b>	<b>28</b>
2.1 Schéma d'élimination parfait d'une matrice . . . . .	28
2.2 Arête bisimpliciale et schéma d'élimination parfait . . . . .	32
2.3 Sommet bisimplicial . . . . .	37
2.4 Reconnaissance d'un graphe biparti distance héréditaire . . . . .	40
<b>3 Problèmes de biclique dans un Graphe Biparti Distance Héréditaire</b>	<b>52</b>
3.1 Biclique maximum . . . . .	52
3.2 Couverture des arêtes par des bicliques . . . . .	61
3.3 couverture des sommets par des bicliques . . . . .	70



<b>4</b>	<b>Chaîne-déviation d'un poset</b>	<b>74</b>
4.1	Développement de l'ordre $D(P)$	74
4.2	Caractérisation de l'invariant $cdev(P)$	78
<b>5</b>	<b><math>P</math>-couplage</b>	<b>82</b>
5.1	Définition et caractérisation d'un $P$ -couplage	82
5.2	Études de cas particuliers	87
5.2.1	Le poset $P$ est une antichaîne	87
5.2.2	Le poset $P$ est faible	87
5.2.3	Le graphe $G$ est une étoile	88

# Introduction

La théorie des graphes et la théorie des ordres sont deux branches des mathématiques discrètes très étroitement liées. Elles sont intensivement étudiées ces dernières années vu leurs champs d'applications dans des domaines très variés, comme l'informatique, l'intelligence artificiel, l'économie, la biologie...

Une relation d'ordre est une relation binaire, sur un ensemble d'objets non vide, réflexive, antisymétrique et transitive. La relation d'ordre intervient naturellement dans l'étude des problèmes de parenté, les problèmes de classifications ou d'ordonnement, etc. Les fondements de la théorie des ordres remonte à l'année 1847, elle revient au célèbre mathématicien Britannique **George Boole** [6], le fondateur de l'algèbre de Boole. La théorie commence à se développer petit à petit de 1854 à 1915 par l'implication des mathématiciens **Morgan, Pierce, Peano, Schroder, Cantor, Dedekind, Russel, Huntington et Hausdorff**. La révolution technologique dans les domaines de l'informatique, la recherche opérationnelle, les sciences sociales ..., a fait en sorte que le développement de cette théorie s'est accéléré à partir des années 1960 comme on le sens dans [1, 6, 83, 85, 86, 87].

Quant à la théorie des graphes, qui ne manque pas d'intérêts que la théorie des ordres et qui est plus vieille que celle-ci, son fondement revient au mathématicien **Léonard Euler** en 1735 quand il a modélisé le problème des ponts de **Koenigsberg** par un graphe [33]. Un graphe est un ensemble de sommets (qui peut représenter un ensemble d'objets, des individus, de pages web, etc.) qui sont reliés par des courbes continues appelées des arêtes ou arc selon que les courbes sont orientées ou pas. Le formalisme de graphe constitué par des outils très puissants qui s'adaptent à l'analyse topologique et structurelles de plusieurs types de problèmes auxquels ne ferons face dans notre vie quotidienne. Nous utilisons les graphes pour représenter les interactions de différentes natures entre plusieurs entités d'un système, un lien social ou un lien d'amitié dans un réseau social constitué de personnes, un câble dans un réseau de routeurs ou en informatique, une réaction chimique dans un réseau biologique de protéines, etc.

Les deux théories partagent pratiquement presque les mêmes domaines d'applications, la différence réside peut être dans la formulation des problèmes et la nature des contraintes à considérer pour l'un ou l'autre. Considérons, par exemple, le problème

de la recherche opérationnelle (voir [14]), connu sous le nom de problème du sac à dos : une collectivité dispose d'un budget pour des investissements pris dans un ensemble d'investissements possibles  $\mathcal{X} = \{1, 2, \dots, i, \dots, n\}$ . Chaque investissement  $i$  présente un coût  $c_i$  et une utilité potentielle  $p_i$ . La collectivité recherche un ensemble  $\mathcal{A}$  d'investissements dont le coût ( $\sum c_i, i \in \mathcal{A}$ ) ne dépasse pas la capacité totale  $C$  d'investissement et qui maximise l'utilité ( $\sum p_i, i \in \mathcal{A}$ ). Si, on est dans une situation où certains investissements ne peuvent être faits que si d'autres l'ont déjà été (par exemple, aides à l'installation d'entreprises après l'aménagement d'une zone industrielle), on a un ordre de précedence  $<$  sur les  $n$  investissements. L'ensemble  $\mathcal{A}$  doit alors vérifier la contrainte :  $i \in \mathcal{A}, j \in \mathcal{X}$  et  $j < i$  impliquent  $j \in \mathcal{A}$ . Un tel ensemble  $\mathcal{A}$  est ce que nous appellerons une partie commençante de l'ensemble ordonné  $(\mathcal{X}, <)$  (cf. la définition 1.42 [14]). Le problème d'optimisation devient alors le problème du sac à dos ordonné.

Les différents chapitres qui composent cette thèse sont répartis comme suit :

### **Chapitre 1 : Préliminaires**

Il contient les définitions de base de la théorie des graphes et de la théorie des ordres utilisées dans la suite de cette thèse. Nous avons complété par quelques notions sur la complexité algorithmique nécessaires pour la compréhension et l'évaluation de l'efficacité des différents algorithmes présentés dans les chapitres suivants.

### **Chapitre 2 : Reconnaissance des graphes bipartis distance héréditaire.**

Il est connu qu'un sommet simplicial est un sommet appartenant à une unique clique maximale, un graphe est triangulé si et seulement s'il admet un ordre d'élimination simplicial et un graphe est triangulé si et seulement si tout parcours en largeur lexicographique (lexBFS) sur les sommets est un ordre d'élimination simplicial. Les notions d'arête bisimpliciale et sommet bisimplicial dans un graphe biparti sont similaires à celle d'un sommet simplicial, c'est-à-dire un sommet bisimplicial (resp. une arête bisimpliciale) appartient à une unique biclique maximale. Les graphes bipartis triangulés sont caractérisés par l'existence d'un ordre d'élimination bisimplicial sur les arêtes. Dans ce chapitre, nous montrons que la classe de graphes bipartis distance héréditaire possède des propriétés presque identiques à celles des graphes triangulés : après avoir donné de nouvelles caractérisations d'un sommet bisimplicial, nous montrons qu'un graphe biparti est distance héréditaire si et seulement si tout ordre des sommets défini par un parcours en largeur lexicographique (LexBFS) est un ordre d'élimination bisimplicial, et ceci nous a conduit à un algorithme de reconnaissance de cette classe de graphes en un temps linéaire.

### **Chapitre 3 : Problèmes de bicliques dans un graphe biparti distance héréditaire**

Dans ce chapitre, en appliquant l'algorithme LexBFS à un graphe biparti distance héréditaire, nous proposons quelques algorithmes de complexité linéaires pour calculer une biclique maximum pour les arêtes, une biclique maximum pour les sommets, toutes les bicliques maximales, une partition minimum des arêtes en bicliques et une couverture minimum des sommets en bicliques.

#### Chapitre 4 : Chaîne déviation d'un ordre

Étant donné un ordre  $P$  quelconque, sur l'ensemble des antichaînes maximales de  $P$ , on définit un nouvel ordre strict noté  $D(P)$  comme suit :  $A, B$  deux antichaînes maximales de  $P$ ,  $A <_{D(P)} B$  si et seulement si,  $\forall a \in A, \exists b \in B$  tel que  $a <_P b$ . De la même façon, on définit  $D^2(P)$  sur les antichaînes maximales de  $D(P)$ , ainsi de suite. A la fin, on aura construit une suite d'ordres  $P, D(P), D^2(P), \dots, D^i(P), \dots$ , où  $D^i(P) = D(D(\dots D(P))\dots)$ ,  $i$  fois. Il est connu qu'il existe un plus petit entier naturel dépendant de l'ordre  $P$ , noté  $cdev(P)$  tel que l'ordre  $D^{cdev(P)}(P)$  est un ordre total. Dans ce chapitre, en utilisant le graphe d'incomparabilité  $Inc(P)$  de  $P$ , nous montrons que la valeur du paramètre  $cdev(P)$  est donnée par une distance entre deux points particuliers additionnels dans le graphe  $Inc(P)$ .

#### Chapitre 5 : Problème du $P$ -couplage

Étant donné un poset  $P = (E, \leq_p)$ , un graphe biparti  $G = (X, Y, E')$  tels que  $|E| = |E'|$  et une bijection  $f : E \rightarrow E'$ . Un  $P$ -couplage dans  $G$  est un ensemble d'arêtes  $M$  vérifiant  $f(A) \cap M$  est un couplage toute antichaîne  $A$  de  $P$ . Le problème consiste à trouver un  $P$ -couplage avec un maximum possible d'arêtes. Dans ce chapitre, nous donnons une condition nécessaire et suffisante pour qu'un ensemble d'arêtes soit un  $P$ -couplage ; nous montrons aussi comment calculer efficacement un  $P$ -couplage maximum quand  $P$  et  $G$  sont particuliers.

# Chapitre 1

## Préliminaires

Au cours de ce premier chapitre, nous rappelons toutes les définitions et notions fondamentales relatives à la théorie des graphes, les ensembles ordonnés et la complexité algorithmique qui seront utiles dans les chapitres suivants. Enfin, nous rappelons les définitions et propriétés de quelques classes de graphes, en particulier les classes de graphes bipartis. Nous avons jugé utile d’accompagner, le long de cette thèse, chaque terme nouvellement introduit par son équivalent en anglais, mise en évidence entre les parenthèses. Pour toute information complémentaire, le lecteur peut consulter les références [7, 14, 24, 42, 91, 93].

### 1.1 Définitions et notations

#### 1.1.1 Graphe et digraphe

Un **graphe non orienté**  $G$  ou un graphe tout court (graph) est défini par un couple d’ensembles  $G = (V_G, E_G)$ , où  $V_G$  est un ensemble non vide de **sommets** (vertices) de  $G$  et  $E_G$  est une relation binaire sur  $V_G$  dont les éléments sont les **arêtes** (edges) de  $G$ . S’il n’y a pas d’ambiguïté, on écrit tout simplement  $G = (V, E)$ . Une arête  $e$  est définie par deux sommets  $u$  et  $v$  qui sont ses extrémités (endpoints) et on la note  $e = \{u, v\}$  ou  $e = uv$ . Notons que  $uv$  et  $vu$  représentent la même arête ; si  $u = v$ , l’arête  $e = uu$  est une **boucle** (loop) ; si deux sommets sont reliés par plusieurs arêtes,  $G$  est un **multigraphe** (multigraph). Un graphe  $G$  est **simple** (simple graph) lorsque il ne contient ni boucle ni arêtes multiples. Autrement dit, s’il est sans boucle et entre deux sommets il y a, au plus, une arête.

Un graphe  $G$  est caractérisé par deux paramètres fondamentaux : le nombre de ses sommets  $n$  et le nombre de ses arêtes  $m$  et qui sont appelés respectivement l’**ordre** et la **taille** de  $G$ .

Deux sommets  $u, v$  sont **adjacents** (adjacent vertices) ou **voisins** (neighbors), s’il existe une arête  $e = uv$ . Dans ce cas, on dit que l’arête  $e$  est **incidente** (incident)

aux sommets  $u$  et  $v$ . Deux arêtes sont adjacentes lorsqu'elles ne sont pas disjointes. La notation  $u \sim_G v$  (resp.  $u \not\sim_G v$ ) signifie que les sommets  $u, v$  sont adjacents (resp. non adjacents) dans  $G$ .

Pour un sommet  $u$  et un ensemble non vides  $W$ , on écrit  $u \sim_G W$  pour dire que le sommet  $u$  est adjacent à tous les sommets de  $W$ . L'ensemble des sommets adjacents à  $v$ , excepté lui-même, est appelé **voisinage ouvert** (open neighborhood) de  $v$ , il est noté  $N(v) = \{u \in V, uv \in E\}$ ; le voisinage **fermé** (closed neighborhood) d'un sommet  $v$ , noté  $N[v]$  est l'ensemble  $N(v) \cup \{v\}$ .

Un graphe est **fini** (finite graph) si son ensemble de sommets est fini. Un graphe **vide** (empty graph) est un graphe  $G = (V, E)$  tel que  $E = \emptyset$ , c'est-à-dire un graphe sans arêtes. Un graphe  $G$  est **biparti** (bipartite graph) si l'ensemble de ses sommets peut être partitionné en deux sous-ensembles  $V_1$  et  $V_2$  de sorte que toute arête ait une extrémité dans  $V_1$  et une autre extrémité dans  $V_2$ ; on le note  $G = (V_1, V_2, E)$  ou  $G = (V_1 \cup V_2, E)$ .

Le **degré** (degree) d'un sommet  $v$  dans un graphe simple  $G$  est la quantité  $d_G(v) = |N_G(v)|$ . Un sommet de degré nul est dit **isolé** (isolated vertex) et un sommet de degré un est un sommet **pendant** (pendant vertex). Le plus petit degré d'un graphe  $G$  est  $\delta(G) = \min\{d(v), v \in V\}$  et son plus grand degré est  $\Delta(G) = \max\{d(v), v \in V\}$ , ils sont respectivement égaux au degré d'un sommet ayant le moins et le plus de voisins. Un graphe est **régulier** (regular graph) si tous ses sommets ont le même degré. Si ce degré vaut  $k$ , le graphe est dit  $k$ -régulier.

Deux sommets  $u$  et  $v$  sont des **vrais jumeaux** (true twins) s'ils ont le même voisinage et sont adjacents, c'est-à-dire qu'ils ont le même voisinage fermé  $N[u] = N[v]$ , ils sont des **faux jumeaux** (false twins) s'ils ont le même voisinage mais ne sont pas adjacents, soit  $N(u) = N(v)$ .

Un **graphe orienté** ou **digraphe** (directed graph ou digraph) est un couple  $\vec{G} = (V_{\vec{G}}, U_{\vec{G}})$  formé d'un ensemble de sommets  $V_{\vec{G}}$  et d'un ensemble de couples ordonnés  $U_{\vec{G}} \subseteq V_{\vec{G}} \times V_{\vec{G}}$  dont les éléments sont les **arcs** de  $\vec{G}$ . Tout arc  $a$  est défini par un couple ordonné de deux sommets  $(u, v)$  tel que  $a = (u, v)$ ;  $u$  est l'**extrémité initiale** (start vertex) de l'arc  $a$ , notée  $I(a) = u$  et  $v$  son **extrémité terminale** (end vertex), notée  $T(a) = v$ . Graphiquement, l'arc  $(u, v)$  est représenté par une courbe reliant les sommets  $u$  et  $v$ , orientée de  $u$  vers  $v$ . A tout digraphe  $\vec{G}$ , nous pouvons associer un graphe  $G$  avec le même ensemble de sommets en remplaçant chaque arc par une arête avec les mêmes extrémités. Ce graphe est le graphe **sous-jacent** de  $\vec{G}$ , noté  $G(\vec{G})$ . A l'inverse, tout graphe  $G$  peut être vu comme un digraphe, en remplaçant chaque arête par deux arcs d'orientations opposées avec les mêmes extrémités; ce digraphe est le digraphe associé à  $G$ , noté  $\vec{G}(G)$ .

Tout concept valide pour les graphes non orientés, excepté ceux pour lesquels l'orientation joue un rôle essentiel, s'applique automatiquement aux digraphes. Par exemple, le degré d'un sommet  $u$  dans un digraphe  $\vec{G}$  est simplement le degré de  $u$  dans le graphe sous-jacent de  $\vec{G}$ ; un digraphe  $\vec{G}$  est simple si son graphe sous-jacent  $G$  est simple. etc.

Un graphe **complet** (complet graph) est un graphe dans lequel deux sommets quelconques sont adjacents. Une **clique** (clique) d'ordre  $n$  est un sous-graphe complet d'ordre  $n$ ; on la note  $K_n$ . Un graphe biparti complet (complet bipartite graph)  $G = (V_1, V_2, E)$  est noté  $K_{p,q}$ , où  $|V_1| = p$ ,  $|V_2| = q$  et dans lequel tout sommet de  $V_1$  est adjacent à tous les sommets de  $V_2$ . Une **biclique** (biclique) est un sous-graphe biparti complet.

Le **complémentaire** (complement) d'un graphe  $G = (V, E)$  est le graphe  $G^c = (V, E^c)$ , où  $E^c$  est l'ensemble des arêtes qui manquent à  $G$  pour être complet. Ainsi, deux sommets sont adjacents dans  $G^c$  si et seulement s'il ne le sont pas dans  $G$ .

Un graphe **partiel** ou sous-graphe **couvrant** (spanning subgraph) d'un graphe  $G = (V, E)$  est un sous-graphe  $G' = (V, E')$  tel que  $E' \subset E$ , c'est-à-dire  $G'$  est obtenu à partir de  $G$  par suppressions d'arêtes sans toucher aux sommets. Si  $A$  est l'ensemble des arêtes supprimées, on écrit  $G' = G \setminus A = (V, E - A)$  pour désigner le graphe partiel qui en résulte. Par exemple, tout graphe simple est un sous-graphe couvrant d'un graphe complet. Le sous-graphe **induit** (induced subgraph) par un sous-ensemble de sommets  $S$  est le graphe  $G[S] = (S, E_S)$ , où  $E_S = \{uv \text{ tel que } u, v \in S\}$ , c'est-à-dire l'ensemble des arêtes dont les deux extrémités sont dans  $S$ . Le sous-graphe induit  $G[S]$  est le sous-graphe obtenu par la suppression consécutive des sommets de  $V_G - S$ .

Une **chaîne** (chain) dans un graphe  $G = (V, E)$  est une séquence de sommets  $\mu = (v_1, v_2, \dots, v_q)$  telle que  $v_i$  et  $v_{i+1}$  sont adjacents, pour  $i = 1, \dots, q - 1$ . Si  $v_1$  et  $v_q$  sont adjacents,  $\mu$  est un **cycle** (cycle). La notation  $\mu[u, v]$  désigne la chaîne joignant les sommets  $u$  et  $v$ . La **longueur** (length) de la chaîne  $\mu$ , noté  $l(\mu)$  est donnée par le nombre de ses arêtes. Une chaîne (resp. cycle) qui ne rencontre pas deux fois le même sommet est dite **élémentaire** (elementary); une chaîne (resp. cycle) qui n'utilise pas deux fois la même arête (resp. arc) est dite **simple** (simple). Une **corde** (chord) dans une chaîne (resp. cycle)  $\mu$  est une arête ( n'appartenant pas à  $\mu$ ) qui connecte deux sommets non consécutifs dans  $\mu$ ; un **trou** (hole) est un cycle d'au moins quatre sommets et sans corde qu'on note par  $C_n, n > 3$ . Un **antitrou** (antihole) est le complémentaire d'un trou.

Un **chemin** (path) dans un graphe orienté  $(\vec{G}, U)$  est une chaîne  $\mu = (v_1, v_2, \dots, v_q)$  telle que  $(v_i, v_{i+1})$  est un arc, pour  $i = 1, \dots, q - 1$ . Si  $(v_q, v_1)$  est un arc,  $\mu$  est un **circuit** (circuit). Dans un graphe non orienté, les notions de chaîne (resp. cycle) et de chemin (resp. circuit) sont confondus.

Un ensemble  $S$  de sommets (ou d'arêtes) est **maximal** (resp. **maximum**) pour une certaine propriété  $\Pi$  si  $S$  possède la propriété  $\Pi$ , et il n'existe aucun ensemble  $S_0$  possédant la même propriété tel que  $S \subset S_0$  (resp.  $|S| < |S_0|$ ). De même, l'ensemble  $S$  est **minimal** (resp. **minimum**) pour  $\Pi$  si  $S$  possède la propriété  $\Pi$ , et il n'existe aucun ensemble  $S_0$  possédant la même propriété tel que  $S_0 \subset S$  (resp.  $|S_0| < |S|$ ).

Un graphe est **connexe** (connected graph) si, pour tout couple de sommets  $u$  et  $v$ , il y a une chaîne  $\mu[u, v]$ . Si non, il est non connexe (not connected graph). Dans un graphe non connexe, une **composante connexe** (connected component) est un sous-graphe connexe maximal .

Un sommet (resp. un ensemble) d'**articulation** (cutvertex (resp. cutset)) est un sommet (resp. ensemble de sommets) tel que sa suppression augmente le nombre de composantes connexes. Parfois, on dit **séparateur** (separator) au lieu d'ensemble d'articulation. Pour deux graphes connexes  $G_1$  et  $G_2$ , la notation  $G_1 + G_2$  désigne le graphe non connexe dont les composantes connexes sont  $G_1$  et  $G_2$ . En particulier,  $mK_2$  désigne le graphe possédant  $m$  composantes connexes formées d'un  $K_2$ .

Deux graphes  $G = (V_G, E_G)$  et  $H = (V_H, E_H)$  sont **isomorphes** (isomorphic graphs) s'il existe une bijection  $f : V_G \rightarrow V_H$  telle que :

$$\forall u, v \in V_G, \quad uv \in E_G \Leftrightarrow f(u)f(v) \in E_H$$

Autrement dit, deux sommets  $u$  et  $v$  sont adjacents dans  $G$  si et seulement si leurs images  $f(u)$  et  $f(v)$  sont adjacents dans  $H$ . On note alors  $G \simeq H$ . Ainsi,

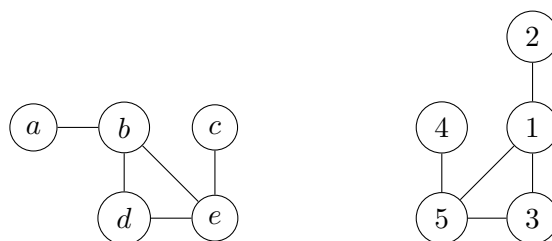


FIGURE 1.1 – Deux graphes isomorphes

deux graphes sont isomorphes s'ils sont identiques ou ils ont la même structure, c'est-à-dire ils diffèrent seulement par les noms des sommets et des arêtes. Les deux graphes de la figure 2.1 sont isomorphes. En effet, l'application  $f$  telle que  $f(a) = 2, f(b) = 1, f(c) = 4, f(d) = 3, f(e) = 5$  est un isomorphisme qui conserve les



relations d'adjacence.

La **distance** (distance)  $d_G(u, v)$  entre les sommets  $u, v$  dans un graphe  $G$  est la longueur d'une plus courte chaîne joignant  $u$  à  $v$ . Si les deux sommets ne sont pas dans une même composante connexe, alors  $d_G(u, v) = +\infty$ . Pour un graphe  $G = (V, E)$  et un entier positif  $k$ , la  **$k$ -ième puissance** (the  $k$ th power) de  $G$  est le graphe  $G^k$  avec le même ensemble de sommets  $V$  et deux sommets  $u, v$  sont adjacents dans  $G^k$  si et seulement si  $d_G(u, v) \leq k$ . Pour un entier positif  $k$ , l'ensemble des sommets situés à une distance inférieur à  $k$  par rapport à un sommet  $v$  est  $D_G(v, k) = \{u \in V, d_G(u, v) \leq k\}$ . L'**excentricité** (eccentricity) d'un sommet  $v$  de  $G$  est  $exc(v) = \max_{w \in V} d_G(v, w)$ ; le **diamètre** (diameter) d'un graphe  $G$  est  $diam(G) = \max_{v \in V} exc(v)$ ; le **rayon** de  $G$  est  $rayon(G) = \min_{v \in V} exc(v)$ .

Un sommet  $v$  est **simplicial** (simplicial vertex) si l'ensemble de ses voisins  $N(v)$  induit une clique, dans ce cas  $N[v]$  induit une clique maximale. Autrement dit, un sommet  $v$  est simplicial s'il appartient à une unique clique maximale.

Un **stable** (stable set ou independent set) dans  $G$  est un ensemble de sommets deux à deux non adjacents, c'est-à-dire un sous-graphe sans arêtes. Rappelons qu'une clique dans un graphe est un stable dans son complémentaire et inversement.

Plusieurs classes de graphes sont définies par des configurations interdites, c'est-à-dire un graphe appartenant à cette classe ne doit pas contenir une certaine configuration comme sous-graphe induit. Plus précisément, si  $\mathcal{F}$  est un ensemble de configurations, un graphe  $G$  est dit  $\mathcal{F}$ -libre ou sans  $\mathcal{F}$  ( $\mathcal{F}$ -free) s'il ne contient aucun élément de  $\mathcal{F}$  comme sous-graphe induit. En particulier, si  $\mathcal{F}$  est réduite à un seul élément  $H$ , nous écrirons sans  $H$  ou  $H$ -libre plutôt que sans  $\{H\}$ . Par exemple, un arbre est un graphe connexe sans cycle ou cycle-free, Un graphe  $2k_2$ -libre est un graphe sans sous-graphe induit défini par deux arêtes disjointes.

On dit qu'une classe de graphes  $\mathcal{F}$  est **héréditaire** (hereditary class) si tout sous-graphe induit d'un graphe de  $\mathcal{F}$  appartient lui-même à la classe  $\mathcal{F}$  (c'est-à-dire  $\mathcal{F}$  est close par suppression de sommet). Par exemple, les graphes bipartis, les graphes sans cycle et les  $2k_2$ -libres sont des classes de graphes héréditaires.

### 1.1.2 Ensemble ordonné

Un **ensemble ordonné** (ordered set) ou un **ordre** (order) est un couple  $P = (X, \leq_P)$ , où  $X$  est un ensemble non vide et  $P$  est une relation d'ordre, c'est-à-dire une relation binaire qui est

1. Réflexive :  $\forall x \in X, x \leq x$ ;
2. Antisymétrique :  $\forall x, y \in X, x \leq y$  et  $y \leq x \Rightarrow x = y$ ;
3. Transitive :  $\forall x, y, z \in X, x \leq y$  et  $y \leq z \Rightarrow x \leq z$ .

L'ordre  $P$  est fini lorsque l'ensemble  $X$  est fini. Deux éléments  $x, y \in X$  sont **comparables** (comparable) dans  $P$  et on écrit  $x \sim_P y$  si,  $x \leq y$  (on lit  $x$  inférieur ou égal à  $y$ ) ou  $y \leq x$ . Sinon, ils sont **incomparables** (incomparable) et on écrit  $x \parallel_P y$ .

Un **ordre total** (totally ordered) ou une **chaîne** (chain) est un ordre dans lequel tout élément est comparable à tous les autres éléments. Un **ordre partiel** (partially ordered) est un ordre dans lequel il existe au moins deux éléments incomparables. Un **ordre strict** (strict order) est un couple  $P = (X, <_P)$  où " $<_P$ " est une relation binaire irreflexive et transitive. Dans toute la suite, nous utilisons l'abréviation "**poset**" pour désigner un ensemble partiellement ordonné.

Un couple  $(x, y) \in X \times X$  est une **couverture** (cover) et on écrit  $x \prec_P y$  si  $x <_P y$  et il n'existe aucun élément  $z \in X$  vérifiant  $x <_P z <_P y$ . On dit alors que  $y$  **couvre**  $x$  ou que  $x$  est **couvert** par  $y$ .

On peut représenter un poset  $P = (X, \leq_P)$ , avec  $|X| = n$ , par un tableau carré  $n \times n$  en associant à chaque élément une ligne et une colonne; on met dans une case  $c_{ij}$  la croix  $\times$  si et seulement si  $i \leq_P j$  (voir l'exemple ci-dessous et la figure 1.2)

**Exemple 1.1.** Considérons l'ensemble suivant  $X = \{2, 3, 4, 6, 8, 9, 12, 18\}$ . On définit l'ordre  $P$  sur  $X$  comme suit :

$$x \leq_P y \Leftrightarrow x \text{ divise } y$$

$\leq_P$	2	3	4	6	8	9	12	18
2	x		x	x	x		x	x
3		x		x		x	x	x
4			x		x		x	
6				x			x	x
8					x			
9						x		x
12							x	
18								x

FIGURE 1.2 – Représentation d'un poset par un tableau

La restriction de la relation  $\leq_P$  à un sous-ensemble  $Y$  de  $X$ , noté  $\leq_Y$ , est appelée **sous-ordre** (sub-order) de  $P$  ou ordre **induit** (induced) par  $Y$ . Une **chaîne** (chain) de  $P$  est un sous-ensemble d'éléments de  $X$  deux à deux comparables; sa longueur est le nombre de ses éléments moins un;  $d(P)$  est le nombre d'éléments dans la plus longue chaîne de  $P$ . Une **antichaîne** (antichain) est un sous-ensemble d'éléments de  $X$  deux à

deux incomparables.

Pour  $x, y \in X$ , la relation  $x <_P y$  signifie aussi que  $y$  est un **successeur** (successor) de  $x$ , donc  $x$  est un **prédécesseur** (predecessor) de  $y$ ;  $\mathbf{Succ}(x) = \{y \in X, x <_P y\}$  est l'ensemble de tous les successeurs de  $x$ ;  $\mathbf{Pred}(x) = \{y \in X, y <_P x\}$  est l'ensemble de tous les prédécesseurs de  $x$ . Un élément  $x$  est **maximal** (maximal element) dans  $P$  si  $\mathbf{Succ}(x) = \emptyset$ ; un élément  $x$  est **minimal** (minimal element) dans  $P$  si  $\mathbf{Pred}(x) = \emptyset$ . On note  $\mathbf{Max}(P)$  (resp.  $\mathbf{Min}(P)$ ) l'ensemble des maximaux (resp. des minimaux) de  $P$ .

On peut associer, à un poset  $P = (X, \leq)$ , un graphe de différentes manières : le graphe de **comparabilité** (comparability graph) qui est le graphe non orienté  $\mathit{Comp}(P) = (X, \mathit{Comp}_P)$  dont les arêtes sont les paires  $xy$  d'éléments comparables dans  $P$ . Plus précisément, pour  $x, y \in X$

$$xy \in \mathit{comp}_P \Leftrightarrow x \sim_P y$$

Le graphe  $\mathit{Inc}(P) = (X, \mathit{Inc}_P)$  d'incomparabilité de  $P$  est défini de manière semblable, ses arêtes étant les paires  $xy$  d'éléments incomparables dans  $P$ . On a alors

$$xy \in \mathit{Inc}_P \Leftrightarrow x \parallel_P y$$

Notons que pour  $x, y \in X$ , on a

$$xy \in \mathit{Comp}_P \Leftrightarrow xy \notin \mathit{Inc}_P \tag{1.1}$$

Ceci montre que les deux graphes sont complémentaires.

La figure 1.3 représente les graphes de comparabilité et d'incomparabilité de l'ensemble ordonné de l'exemple 2.1.

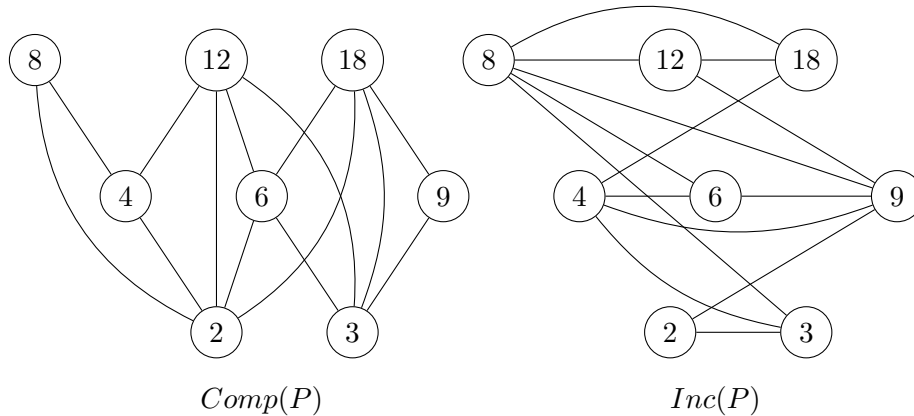


FIGURE 1.3 – Graphes de comparabilité et d'incomparabilité d'un poset

L'une des représentations les plus économiques d'un poset  $P = (X, \leq_P)$  est le **diagramme de Hasse** (Hasse diagram) : une représentation dans laquelle les éléments  $x$

sont représentés par des points  $p(x)$  du plan, de telle sorte que les deux règles suivantes soient respectées :

1.  $x <_P y$  si et seulement si l'horizontale passant par  $p(x)$  est au-dessous de celle passant par  $p(y)$ ,
2. Les points  $p(x)$  et  $p(y)$  sont joints par un segment si et seulement si le couple  $(x, y)$  est une couverture.

Pour tout ordre  $P = (X, \leq_P)$ , on définit l'application  $rang : X \rightarrow \mathbb{N}$  par

$$rang(x) = \begin{cases} 0 & \text{si } x \in \text{Min}(P), \\ 1 + \max\{rang(y), y \in \text{Pred}(x)\} & \text{sinon} \end{cases} \quad (1.2)$$

Le  $i$ -ième **niveau** (level) de  $P$  est le sous-ensemble  $N_{i-1} = \{x \in X / rang(x) = i - 1\}$ . Notons que les éléments d'un même niveau sont deux à deux incomparables. De plus,  $\forall x \in N_i, \exists x_0, x_1, \dots, x_i = x$  tels que  $x_0 \prec_p x_1 \prec_p \dots \prec_p x_i = x$ .

La figure 1.4 représente le diagramme Hasse de l'ensemble ordonné de l'exemple 2.1.

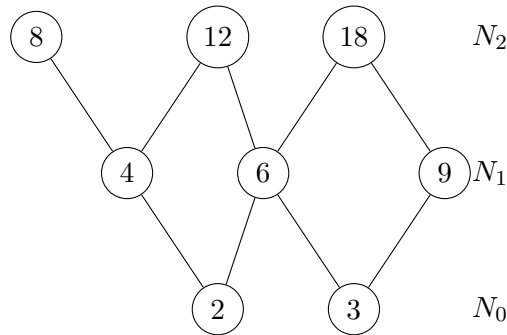


FIGURE 1.4 – Diagramme de Hasse d'un poset

Deux ensembles ordonnés  $P = (X, \leq_P)$  et  $Q = (Y, \leq_Q)$  sont dits **isomorphes** (isomorphic) ou **de même type** (the same type) s'il existe une bijection  $f$  de  $X$  sur  $Y$  vérifiant :

$$x \leq_P y \Leftrightarrow f(x) \leq_Q f(y)$$

La bijection  $f$  est appelée un **isomorphisme** d'ordre (isomorphism) entre  $P$  et  $Q$  et on écrit  $P \equiv Q$ . Dans le cas où  $P = Q$ , on dit que  $f$  est un **automorphisme** (automorphism) de  $P$ .

Deux ensembles ordonnés  $P = (X, \leq_P)$  et  $Q = (Y, \leq_Q)$  sont dits **anti-isomorphes** (dually isomorphic) ou **duaux** (dual) s'il existe une bijection  $f$  de  $X$  sur  $Y$  vérifiant, pour tous  $x, y \in X$  :

$$x \leq_P y \Leftrightarrow f(x) \geq_Q f(y)$$

La bijection  $f$  est appelée un **anti-isomorphisme** d'ordre entre  $P$  et  $Q$  et on écrit  $P \equiv_d Q$ .

Un cas particulièrement intéressant d'anti-isomorphisme est obtenu en considérant l'ensemble ordonné  $P^d = (X, \leq_d)$ , le dual d'un ensemble ordonné  $P = (X, \leq_P)$  et défini par :

$$x \leq_d y \Leftrightarrow y \leq_P x$$

Étant donné un poset  $P = (X, \leq_P)$ , un sous-ensemble  $S$  d'éléments de  $X$  est dit

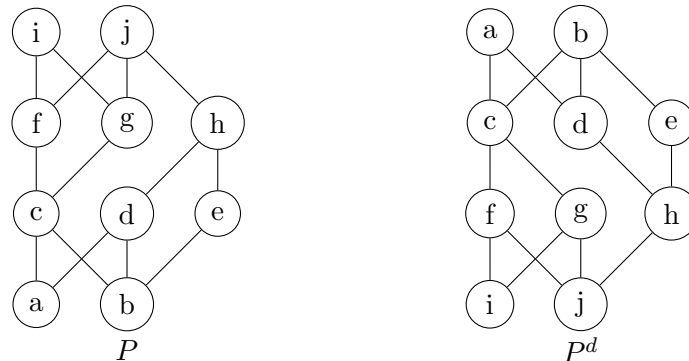


FIGURE 1.5 – Un poset  $P$  et son dual  $P^d$

**autonome** (autonomous) si pour tout élément  $x$  dans  $X - S$ , soit  $x$  est comparable à tous les éléments de  $S$ , soit  $x$  est incomparable à tous les éléments de  $S$ . Dans le poset  $P$  représenté par la figure 1.5, on peut vérifier que l'ensemble  $S = \{c, f, g\}$  est autonome, mais l'ensemble  $S' = \{d, e, h\}$  ne l'est pas, car il existe  $a \in X - S'$  qui est comparable avec  $d$  et incomparable avec  $e$ .

### 1.1.3 Parcours de graphes

Les parcours de graphes sont des algorithmes qui, à partir d'un sommet source (ou origine)  $s$  arbitraire, permettent de visiter et marquer tous les sommets un par un. Au cours du parcours, un sommet est marqué seulement s'il est voisin d'un sommet déjà marqué. À la fin de la procédure, les sommets accessibles seront ordonnés ou numérotés selon leur ordre de visite. Autrement dit, l'ordre de visite est un ordre total sur les sommets accessibles. Bien sûr, les algorithmes de parcours sont utilisés dans les graphes orientés et non orientés et aussi dans les posets. En théorie des graphes par exemple, on utilise souvent les parcours pour la reconnaissance des classes de graphes : la reconnaissance des graphes bipartis, des graphes sans circuit... Ils sont aussi utilisés pour calculer une solution optimale dans de nombreux problèmes d'optimisation combinatoire, problème de l'arbre de poids minimum (l'algorithme de **Prim**), problème de cheminement ( l'algorithme de Dijkstra), problème de flot (l'algorithme de **Belleman** et **Ford**) ... Parfois, un simple algorithme glouton appliqué aux sommets énumérés par un algorithme de parcours peut être derrière la reconnaissance d'une classe de graphe

particulière et la résolution de nombreux problèmes sur cette même classe comme c'est le cas avec les graphes triangulés, d'intervalles et de comparabilité ....[42, 49, 88].

En principe, il existe deux types de parcours fondamentaux, le parcours en largeur (BFS, pour Breadth First Search) et le parcours en profondeur (DFS, pour Depth First Search). Soient un graphe non valué  $G = (V, E)$  d'ordre  $n$  et un sommet origine  $s$ .

#### Parcours en profondeur

Le parcours DFS utilise la stratégie suivante : en arrivant à un sommet  $u$  il traite l'un de ses voisins qui n'est pas encore visité, soit le sommet  $v$ . Avant de traiter le voisin suivant de  $u$ , il traite d'abord tous les descendants du sommet  $v$ . Le même traitement sera appliqué à tous les sommets accessibles depuis la source. Pour cela, on utilise la structure de données **pile** (stack) dont le principe est "dernier arrivé, premier sorti" (LIFO, pour Last In First Out).

#### Parcours en largeur

Le parcours BFS visite et énumère systématiquement tous les sommets accessibles depuis  $s$  et calcule les distances entre  $s$  et ces sommets comme suit :

pour  $k \geq 1$ , le parcours BFS visite et énumère tous les sommets situés à une distance  $k$  de  $s$  avant de passer aux sommets dont la distance est  $k + 1$ . Dans ce parcours, on utilise la structure de données **file** (queue) dont le principe est "premier arrivé, premier sorti" (FIFO, pour First In First Out).

- Remarque 1.1.**
1. S'il existe toujours des sommets inaccessibles depuis le sommet origine  $s$ , on prend un sommet non visité comme une nouvelle origine, à partir duquel on démarre à nouveau le parcours, jusqu'à ce que tous les sommets soient visités.
  2. Dans les parcours BFS et DFS, les arêtes empruntées pour découvrir de nouveaux sommets non marqués, voisins des sommets déjà marqués, constituent une arborescence de racine  $s$  contenant tous les sommets accessibles depuis  $s$ . Plus précisément, elle constitue une arborescence des plus courtes distances de  $s$  aux sommets accessibles.

Pour plus de détails sur ces deux parcours de graphes, nous proposons de consulter la référence [17].

Dans la suite de cette thèse, nous nous intéressons à un type de parcours plus particulier, le **parcours en largeur lexicographique** (ou LexBFS, pour Lexicographic Breadth First Search). LexBFS a été introduit en 1976 par ROSE et al [88] pour la reconnaissance d'une classe de graphes particulière, les graphes triangulés. Au cours des premières années de son introduction, LexBFS n'a pas été trop popularisé, mais à partir des années 1990, ses applications n'ont pas cessé de se multiplier (voir [20, 26, 27]).

**Définition 1.1.** Soient  $L_1 = (a_1, a_2, \dots, a_k)$ ,  $L_2 = (b_1, b_2, \dots, b_t)$  deux vecteurs dont les

coordonnées sont des entiers naturels. On dit que  $L_1$  est inférieur lexicographiquement à  $L_2$  et on écrit  $L_1 <_{lex} L_2$  si et seulement si

1. Il existe  $j \leq \min(k, t)$  tel que  $a_i = b_i$  pour  $i < j$  et  $a_j < b_j$ , ou
2.  $k < t$  et  $a_i = b_i$  pour  $i = 1, \dots, k$

Le vecteur obtenu par la concaténation d'un entier  $a$  à la fin d'un vecteur  $L = (a_1, a_2, \dots, a_k)$  est  $L + a = (a_1, a_2, \dots, a_k, a)$ .

Dans le parcours LexBFS, on énumère les sommets de  $G$  de  $n$  à 1 à partir d'un sommet origine  $s$  comme suit : on associe à chaque sommet  $v$  une étiquette  $L(v)$  qui est un vecteur. Au départ,  $L(s) = n$  et pour  $v \neq s$ ,  $L(v) = \emptyset$ . Pour  $i$  allant de  $n$  à 1, on assigne l'entier  $i$  au sommet  $v$  ayant une étiquette maximale lexicographiquement, et pour tout sommet  $w$ , voisin de  $v$ , non encore numéroté, on concatène l'entier  $i$  à la fin de l'étiquette  $L(w)$ . Sur la figure 1.6, nous montrons comment dérouler le parcours LexBFS sur un graphe.

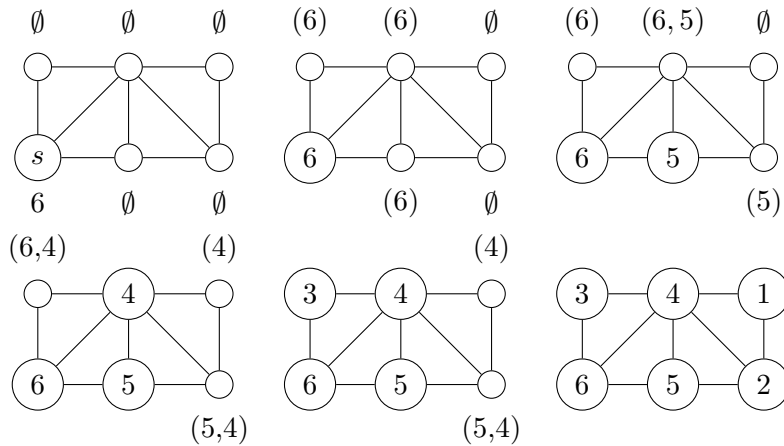


FIGURE 1.6 – Graphe ordonné par LexBFS.

### 1.1.4 Graphe biparti ordonné

Parfois, pour chercher une structure optimale dans un graphe, on est amené à traiter les sommets ou les arêtes selon une certaine numérotation ou un certain ordre. Soit un graphe  $G = (V, E)$ , où  $|V| = n$ . Un ordre (ou une numérotation) sur les sommets de  $G$  est une bijection  $\sigma : \{1, 2, \dots, n\} \rightarrow V$  qui, à tout entier  $i, 1 \leq i \leq n$ , fait associer un sommet  $\sigma(i) = v_i$ , et tel que pour  $i, j \in \{1, 2, \dots, n\}$

$$v_i <_{\sigma} v_j \Leftrightarrow i < j$$

L'ordre des sommets obtenu par le parcours LexBFS est appelé l'ordre **LexBFS** (LexBFS-ordering) ; pour le calculer, on utilise l'algorithme suivant :

**Algorithme 1.1.** (LexBFS, parcours en largeur lexicographique).

**Données :** un graphe  $G = (V, E)$  et un sommet source  $s$ .

**Résultat :** un ordre total  $\sigma$  de  $V$ .

1.  $L(s) = n$ ; pour tout sommet  $v \neq s, L(v) = \emptyset$ .
2. Pour  $i$  de  $n$  à 1 faire :
3. Choisir un sommet  $v$  d'étiquette lexicographique maximale.
4.  $\sigma(i) = v$
5. Pour chaque sommet non numéroté  $w$  de  $N(v)$  faire :
6.  $L(w) = L(w) + i$  // Concaténation de  $i$  à la fin de l'étiquette de  $w$ .

**Remarque 1.2.** De l'algorithme 1.1, on déduit facilement que pour tout sommet  $v$ , son étiquette,  $L(v) = (l_1, l_2, \dots, l_k)$ , vérifie  $l_1 < l_2 < \dots < l_k$ .

Dans la suite de cette thèse, sauf indication contraire,  $G = (V, E)$ , avec  $V = X \cup Y$ , est un graphe biparti connexe et  $\sigma = (v_1, v_2, \dots, v_n)$  est un ordre LexBFS sur les sommets.

Si  $v_i, v_j$  sont deux sommets voisins, avec  $v_i <_\sigma v_j$ , on écrit  $v_i \rightarrow v_j$ . La notation  $\mu^+ = (x_1, x_2, \dots, x_k)$  désigne une chaîne telle que  $x_i \rightarrow x_{i+1}$ , pour  $i = 1, \dots, k-1$ . D'une manière duale,  $\mu^- = (x_1, x_2, \dots, x_k)$  est une chaîne avec  $x_{i+1} \rightarrow x_i$ , pour  $i = 1, \dots, k-1$ . Pour un entier  $k > 0$ , l'ensemble des voisins supérieurs d'ordre  $k$  d'un sommet  $v$  par rapport à  $\sigma$  est  $N_k^+(v) = \{w \in V / \exists \mu^+[v, w], l(\mu^+) = k\}$ ;  $N_k^-(v) = \{w \in V / \exists \mu^-[w, v], l(\mu^-) = k\}$  est l'ensemble des voisins inférieurs d'ordre  $k$  de  $v$ . En particulier,  $N^+(v) = \{w \in V, v \rightarrow w\}$  et  $N^-(v) = \{w \in V, w \rightarrow v\}$  sont respectivement l'ensemble des voisins supérieurs et inférieurs de  $v$ . On note  $N_k^+[v] = N_k^+(v) \cup \{v\}$  et  $N_k^-[v] = N_k^-(v) \cup \{v\}$ .

Par définition d'un ordre LexBFS, il est facile de voir que l'étiquette associée à chaque sommet  $v_i, i < n$ , est constituée par les indices qui énumèrent les voisins supérieurs de  $v_i$  comme le montre le lemme suivant

**Lemme 1.1.** [88] Dans un graphe  $G = (V, E)$  muni d'un ordre lexBFS  $\sigma = (v_1, v_2, \dots, v_n)$ , pour tout  $i = 1, \dots, n-1$ ,

$$L(v_i) = \{j / v_i \rightarrow v_j\}.$$

D'une manière équivalente, nous avons :

$$j \in L(v_i) \Leftrightarrow v_j \in N^+(v_i)$$

**Observation 1.** Soient  $u, v$  deux sommets dans un graphe  $G$  ordonné par LexBFS  $\sigma$ . Alors

$$u <_\sigma v \Rightarrow \forall u' \in N^+(u), \exists v' \in N^+(v) : u' \leq_\sigma v'$$



*Démonstration.* Par définition de LexBFS,  $u <_{\sigma} v$  implique  $L(u) \leq_{lex} L(v)$  qui implique  $\forall i \in L(u), \exists j \in L(v)$  tels que  $i \leq j$ . On en déduit, d'après le lemme 1.1, que  $\forall u' \in N^+(u), \exists v' \in N^+(v) : u' \leq_{\sigma} v'$ .  $\square$

**Remarque 1.3.** Il est connu que dans un graphe ordonné par lexBFS (voir [88]) :

1. une arborescence des plus courtes distances par rapport à la source  $s = v_n$  est calculée ;
2. l'ensemble des sommets est partitionné en niveaux  $N_0, N_1, \dots, N_k$  tels que  $N_0 = \{v_n\}$ , et

$$\forall i = 1, \dots, k, N_i = \{v, d(v, v_n) = i\}$$

3. une arête relie deux sommets qui sont, soit dans un même niveau soit dans deux niveaux consécutifs ;
4. si  $N_i, N_{i+1}$  sont deux niveaux consécutifs, les sommets de  $N_i$  sont énumérés avant l'énumération des sommets de  $N_{i+1}$ .

**Lemme 1.2.** *Soit  $G$  un graphe ordonné par LexBFS. Alors,  $G$  est biparti si et seulement si les niveaux de  $G$  sont des stables.*

*Démonstration.* Rappelons qu'un graphe  $G$  est biparti si et seulement s'il ne contient pas de cycle induit de longueur impaire.

Soient  $u, v$  deux sommets dans un graphe biparti appartenant à un même niveau  $N_i$ . D'après la remarque 1.3,  $d(v, v_n) = d(u, v_n) = i$ . Si  $u$  et  $v$  sont adjacents, alors les trois sommets  $v_n, u$  et  $v$  appartiennent à un même cycle de longueur  $d(v, v_n) + d(u, v_n) + 1 = 2i + 1$ , c'est-à-dire un cycle de longueur impaire. Mais, ceci est en contradiction avec le fait que  $G$  est un graphe biparti. Par conséquent, la condition est nécessaire.

Réciproquement, puisque les sommets d'un même niveau ne sont pas adjacents, alors deux sommets adjacents appartiennent à deux niveaux consécutifs (remarque 1.3). Ainsi, les ensembles  $X$  et  $Y$  qui sont respectivement l'union des niveaux pairs et des niveaux impairs partitionnent l'ensemble  $V$  en deux stables. Par conséquent, le graphe  $G$  est un biparti.  $\square$

La figure 1.7 montre un graphe biparti  $G$  ordonné par l'ordre LexBFS  $\sigma = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9)$ . L'ensemble des voisins supérieurs du sommet  $v_1$  est  $N^+(v_1) = \{v_3, v_4\}$ . Les ensembles des voisins supérieurs d'ordre 2 et 3 de  $v_1$  sont respectivement  $N_2^+(v_1) = \{v_6, v_7, v_8\}$  et  $N_3^+(v_1) = \{v_9\}$ . Les niveaux de  $G$  sont  $N_0, N_1, N_2, N_3$ .

**Lemme 1.3.** *Soit  $G$  un graphe biparti ordonné par LexBFS  $\sigma$ , et soient  $u, v$  deux sommets tels que  $u <_{\sigma} v$ . Alors*

$$N^+(u) \cap N^+(v) = \emptyset \Rightarrow \exists v' \in N^+(v), \forall u' \in N^+(u), u' <_{\sigma} v'$$

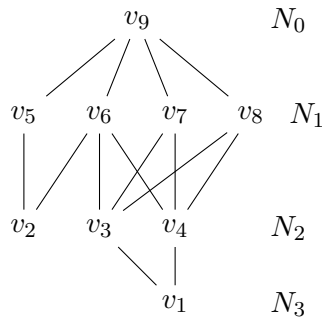


FIGURE 1.7 – Graphe biparti  $G$  ordonné par LexBFS.

*Démonstration.* Par définition de LexBFS,  $u <_{\sigma} v$  implique  $L(u) \leq_{lex} L(v)$ , et puisque  $N^+(u) \cap N^+(v) = \emptyset$ , ceci implique, d'après le lemme 1.1,  $L(u) <_{lex} L(v)$  et  $L(u) \cap L(v) = \emptyset$ . On en déduit, d'après la remarque 1.2, que  $l_1(v)$ , la première coordonnée du vecteur  $L(v)$ , est strictement supérieure à toutes les coordonnées de  $L(u)$ . Il en résulte, d'après le lemme 1.1, qu'il existe un sommet  $v' = \max N^+(v)$ ,  $\forall u' \in N^+(u), u' <_{\sigma} v'$ .  $\square$

**Définition 1.2.** Soit  $G$  un graphe ordonné par LexBFS  $\sigma = (v_1, v_2, \dots, v_n)$ . On dit que la restriction de  $\sigma$  sur un sous-ensemble de sommets  $A$  est un LexBFS, si  $\forall x, y \in A$

$$x <_{\sigma} y \Rightarrow \forall u \in N^+(x) \cap A, \exists v \in N^+(y) \cap A, u \leq_{\sigma} v$$

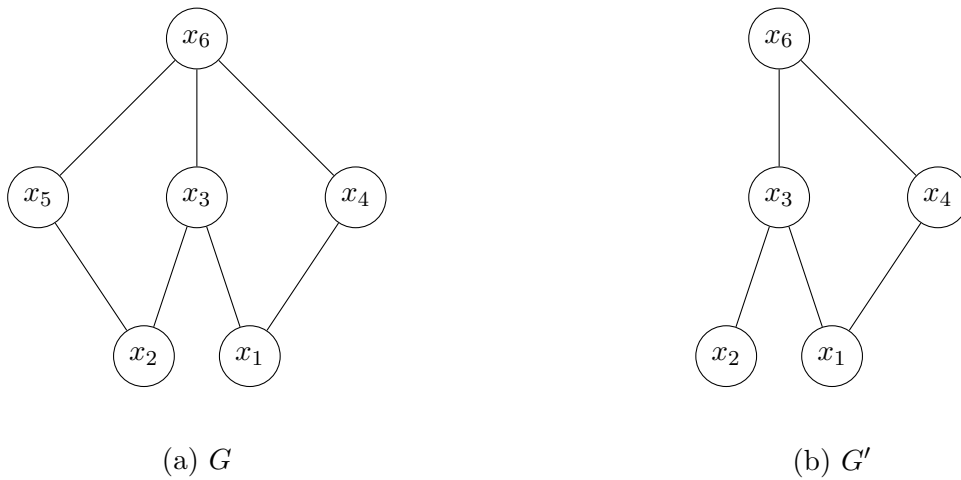


FIGURE 1.8 – La restriction d'un LexBFS

**Remarque 1.4.** Notons qu'en général, la restriction d'un ordre LexBFS sur un sous-ensemble de sommets n'est pas un ordre LexBFS. En effet, dans la figure 1.8, on peut

vérifier que  $\sigma = (x_1, x_2, x_3, x_4, x_5, x_6)$  est un LexBFS sur  $G$ . Cependant, la restriction de  $\sigma$  sur l'ensemble des sommets  $A = (x_1, x_2, x_3, x_4, x_6)$  dont le sous-graphe induit est le graphe  $G'$ , ne l'est pas. Par exemple, on a  $x_1 <_{\sigma} x_2$  mais il n'existe aucun élément dans  $N^+(x_2) \cap A = \{x_3\}$  qui soit supérieur à  $x_4 \in N^+(x_1) \cap A = \{x_3, x_4\}$ .

## 1.2 Problème, algorithme, complexité

La théorie de la complexité est une branche des mathématiques et de l'informatique qui s'intéresse à l'étude de la difficulté intrinsèque des **problèmes algorithmiques** (algorithmic problems), en particulier les problèmes d'optimisation combinatoire dont la définition est donnée ci-dessous, et qui vise à classer ces problèmes en deux classes principales : la classe des problèmes solubles par un algorithme efficace et celle des problèmes qui ne le sont pas. L'efficacité d'un algorithme est mesurée par la quantité de ressources à utiliser pour résoudre un problème. On distingue deux types de ressources ou de complexités : la complexité temporelle, qui est le temps que met un algorithme pour résoudre un problème et la complexité spatiale, qui est l'espace mémoire qu'il utilise au cours de son exécution.

### 1.2.1 Problèmes de décision et d'optimisation

Le problème de la clique, par exemple, est l'un des problèmes qu'on résout par un algorithme et qui peut être formulé de deux manières : on donne un graphe  $G$  et on cherche une clique avec un nombre maximum de sommets ou bien on donne un graphe  $G$  et un entier positif  $k$  et on se demande si ce graphe possède une clique avec au moins  $k$  sommets. Dans les deux cas, pour résoudre ce problème, il faut donner un graphe auquel nous appliquons un algorithme. Ici, la donnée d'un graphe ou d'un graphe avec un entier positif constitue une instance du problème de la clique. D'une manière générale, un problème algorithmique est une question générique qui s'applique à un ensemble d'éléments en entrée appelés ses **instances** (problem instances) ou ses **données** (problem data).

On distingue ainsi deux types de problèmes : ceux qui consistent à répondre par oui ou par non à une question posée, appelés **problèmes de décision** (decision problem) ; ceux qui consistent à maximiser ou minimiser une certaine application sur un ensemble fini à valeurs le plus souvent entières ou réelles, appelés **problèmes d'optimisation** (optimisation problem).

Les versions  $\prec$  optimisation  $\succ$  et  $\prec$  décision  $\succ$  du problème de la clique sont respectivement données comme suit :

---

*Instance* : un graphe  $G$

*Objectif* : Déterminer une clique maximum de  $G$ .

---

*Instance* : Un graphe  $G$  et un entier positif  $k$

*Question* : Existe-il dans  $G$  une clique avec au moins  $k$  sommets ?

---

### 1.2.2 Complexité d'un algorithme

Comme nous l'avons signalé précédemment, la théorie de la complexité s'intéresse aux problèmes solubles par des algorithmes. D'une manière informelle, un **algorithme** (algorithm) résolvant un problème  $p$  est un ensemble fini d'opérations de calcul élémentaires (opérations arithmétiques, comparaisons, affectations ...) qui prend en entrée toute donnée  $x$  du problème  $p$  et produit en sortie un résultat  $A(x)$ . Dans cette thèse, les instances sont des graphes, donc une opération élémentaire peut être aussi un test d'adjacence ou d'incidence, le parcours d'un arc, la suppression d'un sommet ou d'une arête, etc.

Le nombre d'opérations élémentaires effectuées par un algorithme sur une instance  $x$  dépend d'un paramètre appelé la **taille** de  $x$ . Plus précisément, la taille d'une instance  $x$ , notée  $tail(x)$  ou  $|x|$ , désigne généralement le nombre de cases mémoires nécessaires à sa représentation en machine.

Par exemple, l'entier  $x = 19$  est une instance du problème de primalité de taille 5. En effet, le codage en binaire de 19 est 10011, donc il faut 5 cases mémoire pour la représentation de 19 par un ordinateur. Quand une instance est un graphe, sa taille est une fonction des paramètres  $n, m$  qui sont respectivement le nombre de sommets et d'arêtes de ce dernier.

L'exécution d'un algorithme  $A$  sur une instance  $x$  de taille  $|x| = n$  produit deux fonctions, la fonction de complexité temporelle  $t_A(n)$  et la fonction de complexité spatiale  $s_A(n)$  dont les définitions sont : la **complexité temporelle** (temporal complexity) d'un algorithme est une fonction qui mesure le nombre d'instructions élémentaires effectuées au cours de son exécution ; et la **complexité spatiale** (spacial complexity) d'un algorithme est une fonction qui mesure le nombre de cases mémoires occupées par les données manipulées par l'algorithme au cours de son exécution.

Dans la suite, nous ne nous intéresserons qu'à la complexité temporelle des algorithmes étudiés, mais il faut noter que celle-ci est toujours supérieure à la complexité en espace, car l'utilisation de chaque unité élémentaire de mémoire nécessite au moins l'exécution d'une instruction.

Il est clair que le temps d'exécution d'un algorithme est proportionnel à la taille des instances du problème considéré. De plus, ce temps d'exécution peut varier sur des instances de même taille. En effet, pour chercher, par exemple, un élément dans un tableau de taille  $n$ , on peut le trouver à la première position, donc au premier test, comme on peut le trouver à la dernière position du tableau et donc il faut exécuter  $n$  opérations élémentaires. Ainsi, pour éviter les mauvaises surprises dans le sens où

on ne peut avoir un temps d'exécution plus pire que celui que l'on a prévu, on définit généralement une complexité en considérant la pire instance possible parmi toutes les instances de taille  $n$ , c'est-à-dire celle qui nécessite le maximum possible d'opérations.

La théorie de la complexité ne s'intéresse pas à l'expression exacte des fonctions de complexités des algorithmes, mais plutôt à la nature de celles-ci : elle cherche une frontière entre la fonction de complexité acceptable et la fonction de complexité inacceptable, et elle fait l'hypothèse que cette frontière se situe à la limite entre fonction polynomiale et non polynomiale. Par exemple, un algorithme pour lequel  $t(n) = 6n^4$  est efficace, mais un algorithme pour lequel  $t(n) = e^n$  (ou  $t(n) = 2^n$ ) ne l'est pas. La pratique confirme cette hypothèse, car pour une instance de taille  $n = 100$ , il faut des années pour un ordinateur, capable d'effectuer 1 million d'opérations par seconde, pour exécuter  $2^{100}$  opérations.

Habituellement, la complexité en temps est exprimée à l'aide de la notation "O" (on lit "grand O"). Soient deux fonctions  $f; g : \mathbb{N} \rightarrow \mathbb{R}_+$ . On dit que  $f = O(g)$  lorsqu'il existe un entier  $n_0$  et une constante réelle  $c$  tel que pour tout  $n \geq n_0$ ,  $f(n) \leq cg(n)$ . Autrement dit, à partir d'un certain entier  $n_0$ , la fonction  $f$  est, à un facteur constant près, majorée par la fonction  $g$  (ou  $f$  ne peut pas croître plus que  $g$ ). Par exemple, si  $f(n) = 9n^4 + 2n^3 + 5n + 1$  alors,  $f = O(n^4)$ . En effet,  $\forall n \geq 1$

$$9n^4 + 2n^3 + 5n + 1 \leq 9n^4 + 2n^4 + 5n^4 + n^4 = 17n^4$$

donc il suffit de poser  $n_0 = 1$  et  $c = 17$ . D'une manière générale, si  $f$  est un polynôme de degré  $k$ , alors  $f = O(n^k)$ .

**Définition 1.3.** Un algorithme  $A$  est **polynomial** (polynomial algorithm) si sa complexité est majorée par un polynôme en la taille des données, c'est-à-dire il existe un entier  $k$  tel que  $t_A = O(n^k)$ . Dans le cas contraire, il est dit **exponentiel** (exponential algorithm).

### 1.2.3 Classes de problèmes $\mathcal{P}$ et $\mathcal{NP}$

Pour définir les classes de problèmes, on utilise la notion de problème de décision, mais il est facile de se convaincre qu'à tout problème d'optimisation correspond un problème de décision ayant une même difficulté. La difficulté d'un problème de décision est déterminée par la complexité du meilleur algorithme qui le résout : si cet algorithme est polynomial, le problème est facile.

**Définition 1.4.** Un problème de décision  $p$  est dans la classe  $\mathcal{P}$  s'il existe un algorithme qui le résout avec une complexité temporelle  $O(n^k)$ , où  $n$  représente la taille de ses instances et  $k$  est un réel positif qui dépend de l'algorithme considéré.

**Remarque 1.5.** Bien que pour montrer qu'un problème  $p$  est dans la classe  $\mathcal{P}$ , il suffit de trouver un algorithme qui peut le résoudre en  $O(n^k)$ , mais trouver un algorithme avec une constante  $k$  la plus petite possible reste un véritable défi.

**Définition 1.5.** Un problème de décision  $p$  est dans la classe  $\mathcal{NP}$  si toute solution proposée pour  $p$  peut être vérifiée et validée en temps polynomial (ce que l'on appelle un certificat du oui).

Considérons la version « décision » du problème de la clique : étant donné un graphe  $G$  et un entier positif  $k$ , existe-t-il une clique avec au moins  $k$  sommets ? Si l'on dispose d'un ensemble  $C$  de sommets, on peut vérifier en temps polynomial (précisément en  $O(|C|^2)$ ) que  $|C| \geq k$  et que  $C$  est une clique, donc ce problème est clairement dans  $\mathcal{NP}$ .

Naturellement, si on dispose d'un algorithme polynomial pour résoudre un problème donné, ce même algorithme peut être utilisé pour vérifier si une solution fournie est bien une solution réalisable du même problème. Par conséquent

$$\mathcal{P} \subseteq \mathcal{NP}.$$

La question de savoir si  $\mathcal{P} = \mathcal{NP}$  est toujours un problème ouvert, le plus important, de la théorie de la complexité. Cela revient à savoir si le fait de chercher une solution est aussi simple que de vérifier une solution. La conjecture admise est

$$\mathcal{P} \neq \mathcal{NP}$$

.

#### 1.2.4 La classe $\mathcal{NP}$ -complet

La communauté scientifique conjecture que  $\mathcal{P} \neq \mathcal{NP}$ . Cette hypothèse implique l'existence d'un ensemble de problèmes de décision qui sont insolubles par des algorithmes polynomiaux. Cette classe de problèmes contient sans doute les problèmes les plus difficiles de la classe  $\mathcal{NP}$ . Cependant, pour affirmer que certains problèmes sont les plus difficiles, il faut pouvoir les comparer. La notion de réduction polynomiale est à la base de cette comparaison.

**Définition 1.6.** Soient  $p, q$  deux problèmes de décision et  $I_p$  ( resp.  $I_q$ ) leurs ensembles d'instances respectivement. On dit que  $p$  se transforme (ou se réduit) polynômialement en  $q$  s'il existe une application  $f : I_p \rightarrow I_q$ , calculable en un temps polynomial et transformant toute instance  $x$  de  $p$  en une instance  $f(x)$  de  $q$  admettant la même réponse que  $x$ . Plus précisément,  $x$  est une instance positive si et seulement si  $f(x)$  l'est. On écrit alors  $p \leq q$ , et qui signifie que le problème  $p$  n'est pas plus difficile que le problème  $q$ .

Considérons le problème du couplage (matching problem) dans un graphe. Un **couplage** est un ensemble d'arêtes  $M$  qui sont deux à deux disjointes. La version "décision" de ce problème est

---

Instance : un graphe  $G = (V(G), E(G))$  et un entier positif  $k$ .

Question : Existe-t-il un couplage  $M$  de cardinalité  $|M| \geq k$  ?

---

Ce problème se réduit en un temps polynomial au problème de la clique comme suit : soit  $H = (V(H), E(H))$  le graphe dont les sommets sont les arêtes de  $G$ , et deux sommets de  $H$  sont adjacents si et seulement si les arêtes qui leurs correspondent dans  $G$  sont disjointes. Il est facile de se convaincre que le graphe  $H$  peut être calculé en un temps polynomial et que le graphe  $G$  admet un couplage de taille  $k$  si et seulement si le graphe  $H$  admet une clique de taille  $k$ .

**Remarque 1.6.** Sachant que toute clique dans un graphe  $G$  est un stable dans son complémentaire  $G^c$ , et tout stable dans  $G$  est une clique dans  $G^c$ . Ceci montre que le problème de la clique peut se réduire en un temps polynomial au problème du stable, et vice-versa. Ainsi, ces deux problèmes ont une même difficulté. En général, deux problèmes  $p, q$  sont dits équivalents (en termes de la difficulté) si  $p \leq q$  et  $q \leq p$ . On écrit alors

$$p \sim q$$

La notion de réduction polynomiale a donné naissance à une autre hypothèse, celle d'existence d'une classe de problèmes, notée  $\mathcal{NP}$ -complet, contenant tous les problèmes de décision qui sont équivalents et qui sont aussi difficiles que tous les problèmes de la classe  $\mathcal{NP}$ .

**Définition 1.7.** Un problème  $q$  est dit  $\mathcal{NP}$ -complet si :

1.  $q \in \mathcal{NP}$
2.  $\forall p \in \mathcal{NP}, p \leq q$ .

L'existence de la classe de problèmes  $\mathcal{NP}$ -complet a été confirmée par COOK et LEVIN, en montrant qu'il existe au moins un problème  $\mathcal{NP}$ -complet, appelé problème  $\mathcal{SAT}$ , et qui est défini comme suit : soient  $X = \{x_1, \dots, x_k\}$  un ensemble de variables booléennes et  $\bar{X} = \{\bar{x}_i : x_i \in X\}$ , où  $\bar{x}$  est la négation de la variable  $x$ . Une assignation pour  $X$  est une application  $T : X \cup \bar{X} \rightarrow \{\text{vrai}, \text{faux}\}$  vérifiant :

$$T(x) = \text{vrai} \text{ si et seulement si } T(\bar{x}) = \text{faux}.$$

Les éléments de  $X \cup \bar{X}$  sont appelés les littéraux sur  $X$ . Une clause sur  $X$  est une disjonction de littéraux, par exemple,  $C = x_1 \vee \bar{x}_2 \vee x_4$  est une clause constituée par trois littéraux.

Une formule booléenne est sous forme normale conjonctive si elle s'écrit comme la conjonction de clauses. Le problème  $\mathcal{SAT}$  consiste à décider si une formule booléenne sous forme normale conjonctive est satisfaite, c'est-à-dire s'il existe une assignation  $T$  telle que toutes les clauses sont vraies. Par exemple pour  $X = \{x_1, x_2, x_3\}$ , la formule  $Z = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_2 \vee \bar{x}_3) \wedge (x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_3)$  est satisfaite, car l'assignation définie par  $T(x_1) = \text{vrai}$ ,  $T(x_2) = \text{faux}$  et  $T(x_3) = \text{vrai}$  satisfait toutes les clauses.

La version "décision du problème  $SAT$ ) s'annonce comme suit :

Instance : Une famille de  $k$  clauses formées sur l'ensemble  $X$  de  $n$  variables.

Question : Existe-t-il une assignation  $T$  qui satisfait la formule

$$Z = C_1 \wedge C_2 \wedge \dots \wedge C_k?$$

En utilisant la notion de réduction, pour montrer qu'un problème de décision  $p$  est  $\mathcal{NP}$ -complet, il suffit de trouver un problème  $q$ , connu pour être  $\mathcal{NP}$ -complet, et montrer ensuite que  $q \leq p$ . C'est ainsi qu'à partir de  $SAT$ , on montre progressivement la  $\mathcal{NP}$ -complétude d'autres problèmes qui, à leurs tour, peuvent être utilisés pour démontrer la  $\mathcal{NP}$ -complétude de nouveaux problèmes. Par exemple, à partir de  $SAT$ , on a montré que  $3 - SAT$  est  $NP$ -complet (Karp (1972)[60]). Le Problème  $3 - SAT$  s'énonce comme  $SAT$  avec la condition que toutes les clauses sont de cardinalité 3).

### 1.3 Quelques classes de graphes

La plupart des problèmes rencontrés en optimisation combinatoire ne disposent pas d'algorithme efficace pour les résoudre. Ainsi, dès qu'un problème est connu pour être difficile, les chercheurs se trouvent en face de deux alternatives : renoncer à la solution exacte et pencher vers la recherche des algorithmes d'approximations efficaces ou limiter la recherche d'une solution exacte sur une classe de graphes particulière dans laquelle des algorithmes efficaces sont espérés. Cependant, un autre problème voit le jour quant à la seconde alternative, le problème de reconnaissance de graphes. En effet, une classe de graphes n'est intéressante que si on dispose d'un algorithme efficace pour sa reconnaissance. Nous présentons ci-dessous brièvement quelques classes de graphes bipartis que nous allons rencontrer par la suite, avec leurs propriétés et les liens entre elles. Pour plus de détail sur les classes de graphes, le lecteur peut consulter la référence [94].

#### 1.3.1 Graphe triangulé

Un graphe **triangulé ou cordal** (chordal graph) est un graphe sans trou. De manière équivalente, un graphe triangulé ne contient pas de cycle induit de longueur 4 ou plus. Sans doute, la classe des graphes triangulés est l'une des classes les plus importantes en théorie des graphes et en algorithmique, elle a été introduite en 1958 par Hajnal et al [50]. La reconnaissance de cette classe peut se faire en un temps linéaire, soit en utilisant l'algorithme LexBFS [88] ou en utilisant l'algorithme (maximum cardinality search MCS) [97]. Plusieurs problèmes qui sont  $NP$ -complet, comme les problèmes du stable, de la clique, de coloration ..., deviennent polynomiaux quand ils sont considérés dans cette classe [43].

**Remarque 1.7.** Rappelons qu'un graphe est biparti si et seulement s'il n'admet pas de cycle impair, donc tous les trous d'un biparti sont pairs, et il est facile de voir qu'une



corde dans un tel cycle est une arête reliant deux sommets non consécutifs et n'ayant pas une même parité.

En 1978 et à partir de la remarque ci-dessus, Golumbic et Goss [43] ont généralisé d'une manière naturelle la définition d'un graphe triangulé au graphe triangulé biparti : un graphe **biparti triangulé** (chordal bipartite graph) est un graphe biparti dans lequel tout cycle de longueur supérieure ou égale à six contient une corde ; on utilise aussi la notation **(6,1)-cordal** pour désigner un graphe biparti triangulé [93]. Cette classe de graphes peut être reconnue, en utilisant la matrice d'adjacence biparti, en  $O(\min(m \log n, n^2))$  [55, 67, 80, 93], où  $n$  et  $m$  sont respectivement le nombre de sommets et d'arêtes.

Il est clair qu'un sous-graphe induit d'un graphe sans trou est sans trou, donc tout graphe induit d'un graphe triangulé (resp. biparti triangulé) est triangulé (resp. biparti triangulé). Autrement dit, la classe des graphes triangulés est une classe héréditaire. Les graphes complets et les arbres sont des exemples simples de graphes triangulés. Les graphes bipartis complets, un domino (voir la figure 1.9) sont des graphes bipartis triangulés.

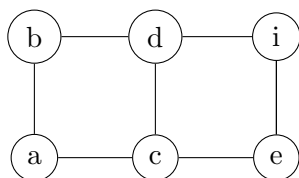


FIGURE 1.9 – Domino

### 1.3.2 Graphe distance héréditaire

Un graphe **distance héréditaire** (Distance-Hereditary graph) est un graphe dans lequel les distances entre sommets dans tout sous-graphe induit connexe sont les mêmes que celles du graphe tout entier ; autrement dit, tout sous-graphe induit hérite les distances du graphe entier. Les graphes distance héréditaire ont été nommés et étudiés pour la première fois par Howorka en 1977 [56]. Un graphe distance héréditaire peut être reconnu en un temps linéaire [5, 22].

Un graphe **biparti distance héréditaire** (Bipartite Distance-Hereditary graph) est un graphe qui est en même temps distance héréditaire et biparti. Plusieurs caractérisations pour cette classe de graphes existent, parmi celles ci, un graphe biparti  $G$  est distance héréditaire si et seulement si tout cycle de longueur supérieure ou égal à 6 possède au moins deux cordes ; on dit aussi graphe (6,2)-cordal ((6,2)-chordal graph) [73, 93]. Il vient immédiatement qu'un graphe (6,2)-cordal ne peut pas contenir un domino, donc les graphes (6,2)-cordal constituent une sous-classe de graphes domino-libre.

Dans la suite de cette thèse, nous utilisons l'abréviation *GBDH* pour désigner un graphe biparti distance héréditaire.

### 1.3.3 Grappe de permutation

Un graphe  $G = (V, E)$  est de **permutation** (permutation graph) s'il existe deux permutations  $p$  et  $q$  sur l'ensemble des sommets  $V$  tels que deux sommets  $x$  et  $y$  sont adjacents si et seulement si  $x$  précède  $y$  dans l'une des permutations et  $y$  précède  $x$  dans l'autre [94]. Un graphe biparti de permutation (bipartite permutation graph) est un graphe de permutation et biparti en même temps. Un graphe de permutation est un graphe d'intersection : considérons deux droites parallèles. Sur la première, on représente les sommets de  $G$  selon l'ordre de  $p$  et sur la deuxième, on représente les mêmes sommets selon l'ordre de  $q$ . Pour chaque sommet  $x$ , on relie  $p(x)$  et  $q(x)$  par un segment de droite. Deux sommets  $x$  et  $y$  sont adjacents si et seulement si les deux segments qui leurs correspondent se coupent. Ainsi, le graphe  $G$  n'est alors rien d'autre que le graphe d'intersection de ces segments, voir la fig. 1.10 qui montre un graphe biparti de permutation.

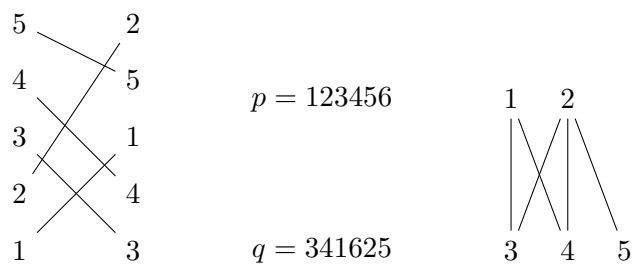


FIGURE 1.10 – Grappe biparti de permutation.

Un graphe biparti de permutation peut être reconnu en un temps linéaire [94]. Plusieurs problèmes  $\mathcal{NP}$ -complet, comme les problèmes du stable, de l'ensemble dominant, du cycle hamiltonien..., deviennent linéaires dans cette classe de graphes [45].

### 1.3.4 Grappe convexe

Soit  $G = (X, Y, E)$  un graphe biparti. On dit qu'un ordre  $<$  sur  $X$  possède la **propriété d'adjacence** (adjacency property) si les voisins de tout sommet  $y \in Y$  sont consécutifs dans  $X$ , c'est-à-dire pour tout  $y \in Y$ ,  $N(y)$  est un intervalle pour l'ordre  $<$  dans  $X$ . Un graphe biparti  $G = (X, Y, E)$  est **convexe** en  $X$  ou  $X$ -convexe (convex bipartite graph on  $X$ ) si les sommets de  $X$  peuvent être ordonnés par un ordre ayant une propriété d'adjacence. On définit d'une manière similaire un graphe biparti  $Y$ -convexe. Un graphe biparti  $G = (X, Y, E)$  est **biconvexe** (biconvex graph) s'il est  $X$ -convexe et  $Y$ -convexe.

Dans la figure 1.11, nous avons le graphe biparti  $G = (X, Y, E)$ , avec  $X = \{u, v, w, x\}$  et  $Y = \{a, b, c, d\}$ . On voit bien que l'ordre  $\sigma = (v, u, w, x)$  possède la propriété d'adjacence sur  $X$ , donc le graphe  $G$  est un graphe  $X$ -convexe. Ce même graphe n'est pas  $Y$ -convexe, et donc il n'est pas biconvexe. Cependant, si nous rajoutons l'arête  $e = va$  à ce graphe, il devient biconvexe.

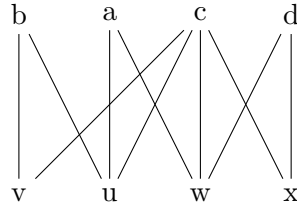


FIGURE 1.11 – Graphe  $X$ -convexe

Cette classe de graphes a été introduite en 1967 par Glover [40] quand il a modélisé un problème de fabrications de tiges sous forme d'un problème de couplage dans un graphe biparti. Des algorithmes linéaires pour reconnaître un graphe  $X$ -convexe existent [9, 49]. Plusieurs problèmes peuvent être résolus efficacement dans cette classe de graphe : en plus du problème de couplage maximum [66, 95], nous avons le problème du stable maximum [92], le problème de l'ensemble dominant [21]....

La figure 1.12 montre quelques classes de graphes bipartis ordonnées par inclusion qu'on peut trouver dans [3, 28]

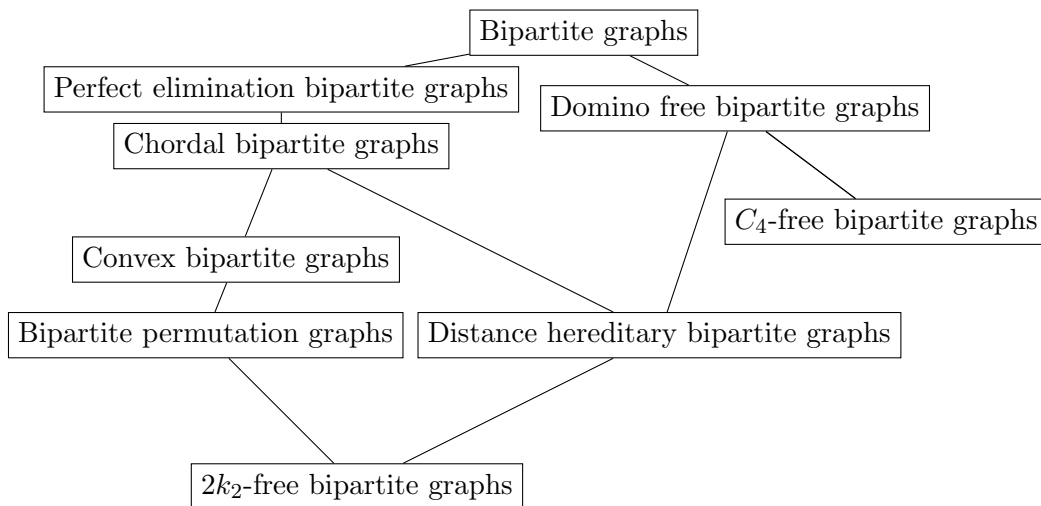


FIGURE 1.12 – A hierarchy of bipartite graph classes.

## **Conclusion du chapitre**

Ainsi, nous disposons d'une liste de définitions et notations de base qui n'est pas exhaustive mais qui est largement suffisante pour aborder la suite de ce document.

## Chapitre 2

# Reconnaissance des graphes bipartis distance héréditaire

La plus part des problèmes d'optimisation combinatoires sont NP-complet, en particulier les problèmes de calcul d'invariants dans les graphes. Par ailleurs, à tout problème difficile correspond au moins une classe de graphes particulière à l'intérieur de laquelle ce même problème devient facile. De plus, les graphes modélisant des problèmes pratiques dans diverses domaines sont souvent des graphes particuliers. Par conséquent, disposer d'un algorithme efficace pour la reconnaissance d'une classe de graphes peut être d'une importance majeure dans ce contexte. Dans ce chapitre, nous rappelons le lien existant entre les graphes bipartis et la résolution des systèmes linéaires carrés. Par la suite, nous rappelons les notions d'arête bisimpliciale et sommet bisimplicial en introduisant de nouvelles caractérisations quant aux sommets bisimpliciaux. Enfin, nous utilisons le parcours en largeur lexicographique (ou parcours LexBFS) pour établir un algorithme linéaire de reconnaissance d'un graphe biparti distance héréditaire.

### 2.1 Schéma d'élimination parfait d'une matrice

Un système linéaire d'ordre  $n$  est défini par la donnée de  $n$  équations linéaires à  $n$  inconnues  $x_1, x_2, \dots, x_n$ . Il est défini comme suit

$$\begin{cases} m_{11}x_1 + m_{12}x_2 + \dots + m_{1n}x_n = b_1 \\ m_{21}x_1 + m_{22}x_2 + \dots + m_{2n}x_n = b_2 \\ \vdots \\ m_{n1}x_1 + m_{n2}x_2 + \dots + m_{nn}x_n = b_n \end{cases}$$

En notation matricielle, ce système s'écrit

$$Mx = b \tag{2.1}$$

où  $M = (m_{ij})$  est une matrice carrée d'ordre  $n$ ,  $b \in \mathbb{R}^n$  est un vecteur connu,  $x \in \mathbb{R}^n$  est un vecteur inconnu. Les coefficients  $m_{ij}$  sont appelés les entrées de la matrice  $M$ .

Il est connu qu'un système linéaire carré admet une unique solution si et seulement si la matrice  $M$  est inversible. Il est aussi connu que si on ajoute à une ligne une combinaison linéaire des autres lignes, on trouve la même solution. Deux systèmes carrés, de même ordre et admettant une même solution sont dits équivalents.

La méthode d'élimination de Gauss (appelé aussi Gauss-Jordan) est l'une des méthodes qu'on utilise pour résoudre les systèmes linéaires; en se basant sur les propriétés citées ci-dessus, elle transforme le système (2.1), au bout de  $n$  itérations, en un système diagonal équivalent

$$I_n x = b'$$

où  $I_n$  est la matrice identité. Autrement dit, le système diagonal est obtenu après une suite finie de systèmes équivalents

$$M^{(k)} x = b^{(k)}, 1 \leq k \leq n$$

où  $M^{(1)} = M$ ,  $b^{(1)} = b$ , et pour  $k = 1, \dots, n$ ,  $M^{(k)}$ ,  $b^{(k)}$  sont respectivement la matrice et le vecteur correspondants à l'itération  $k$ .

Pour calculer la matrice  $M^{(k+1)}$ ,  $1 \leq k < n$ , par la méthode de Gauss-Jordan, on procède comme suit : si l'entrée  $m_{kk}^{(k)} \neq 0$ , on la choisit comme pivot, on divise ensuite les membres de l'équation  $k$  par le pivot  $m_{kk}^{(k)}$  et par des opérations élémentaires sur les lignes, on transforme toutes les autres entrées sur la colonne  $j$  en zéros. Plus précisément, les entrées de la matrice  $M^{(k+1)} = (m_{ij}^{(k+1)})$  s'obtiennent en effectuant les opérations suivantes

$$m_{ij}^{(k+1)} = \begin{cases} \frac{m_{kj}^{(k)}}{m_{kk}^{(k)}} & \text{si } i = k \\ m_{ij}^{(k)} - \frac{m_{ik}^{(k)} \times m_{kj}^{(k)}}{m_{kk}^{(k)}} & \text{si } i \neq k \end{cases} \quad (2.2)$$

où on a supposé que le pivot  $m_{kk}^{(k)} \neq 0$ . Le passage de l'itération  $k$  à  $k+1$  a pour conséquence la transformation de la colonne  $k$  en vecteur unité qui sera conservé jusqu'à la fin des itérations. Ainsi, la matrice  $M^{(k+1)}$  se présente sous la forme

$$M^{(k+1)} = \begin{pmatrix} 1 & 0 & \dots & 0 & m_{1(k+1)}^{(k+1)} & \dots & m_{1n}^{(k+1)} \\ 0 & 1 & \dots & 0 & m_{2(k+1)}^{(k+1)} & \dots & m_{2n}^{(k+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & m_{k(k+1)}^{(k+1)} & \vdots & \vdots \\ 0 & 0 & \dots & 0 & m_{(k+1)(k+1)}^{(k+1)} & \dots & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & m_{n(k+1)}^{(k+1)} & \dots & m_{nn}^{(k+1)} \end{pmatrix}$$

**Remarque 2.1.** Comme mentionné ci-dessus, la méthode de Gauss-Jordan n'est correctement applicable que si les pivots  $a_{kk}^{(k)}$  sont non nuls, pour  $k = 1, \dots, n - 1$ . De plus, le fait que les termes diagonaux de la matrice  $M$  soient non nuls n'empêche pas de tomber sur un pivot nul au cours de la réduction. Si une telle situation se produit, on fait appel à la technique de changement de pivot qui consiste à échanger des lignes (ou des colonnes) du système de manière à ce qu'aucun pivot ne soit nul.

Un autre problème peut survenir lors du déroulement de la méthode de Gauss appelé remplissage de zéros.

**Définition 2.1.** Le **remplissage de zéros** (fill-in) se produit quand une entrée nulle à l'itération  $k$  est transformée en une entrée non nulle à l'itération  $k + 1$ , c'est-à-dire  $m_{ij}^{(k)} = 0$  et  $m_{ij}^{(k+1)} \neq 0$ .

**Exemple 2.1.** L'application de la réduction de Gauss à la matrice  $M = M^{(1)}$ , en pivotant par  $m_{11}^{(1)} = 2$ , produit les fill-in suivants :  $m_{25}^{(1)} = 0$ ,  $m_{25}^{(2)} = -4 \neq 0$  et  $m_{52}^{(1)} = 0$ ,  $m_{52}^{(2)} = -4 \neq 0$ .

$$M^{(1)} = \begin{pmatrix} 2 & 2 & 0 & 4 & 4 \\ 2 & 3 & 0 & 5 & 0 \\ 0 & 0 & 4 & 2 & 3 \\ 4 & 5 & 2 & 7 & 9 \\ 4 & 0 & 3 & 9 & 10 \end{pmatrix}; M^{(2)} = \begin{pmatrix} 1 & 1 & 0 & 2 & 2 \\ 0 & 1 & 0 & 1 & -4 \\ 0 & 0 & 4 & 2 & 3 \\ 0 & 1 & 2 & 3 & 1 \\ 0 & -4 & 3 & 1 & 2 \end{pmatrix}$$

Le problème de fill-in survient souvent lorsque la matrice  $M$  est creuse (i.e plusieurs de ses entrées sont nulles). L'exemple 2.1 montre que le choix arbitraire d'un pivot non nul n'est pas suffisant pour éviter un fill-in. Sachant que la représentation des matrices, d'une manière efficace, par la machine, se fait en listant leurs entrées non nulles, la production de fill-in au cours de la réduction augmente la complexité spatiale de la méthode. Par conséquent, il est naturel de se poser la question suivante : Peut-on réarranger la matrice  $M$  pour éviter de tomber sur un fill-in au cours de l'application de la méthode de Gauss ? La réponse sera affirmative si la matrice  $M$  vérifie la définition suivante :

**Définition 2.2.** Une matrice carrée  $M$  admet un **schéma d'élimination parfait** (perfect elimination scheme) s'il existe une séquence de pivots pour lesquels l'application de la méthode de Gauss sur  $M$  conserve les entrées nulles jusqu'à la fin de la procédure.

Il reste maintenant à savoir comment caractériser une matrice ayant un tel schéma. Une condition nécessaire et suffisante pour l'existence d'un schéma d'élimination parfait pour une matrice symétrique et sans zéro sur la diagonale est donnée par Rose en 1970 [89].

**Définition 2.3.** Soit  $M = (m_{ij})$  une matrice symétrique d'ordre  $n$ . On définit le graphe  $G(M)$  comme suit :  $v_1, v_2, \dots, v_n$  sont les sommets de  $G(M)$  ; une paire  $v_i v_j$  est une arête si et seulement si l'entrée  $m_{ij} \neq 0$  et  $i \neq j$ .

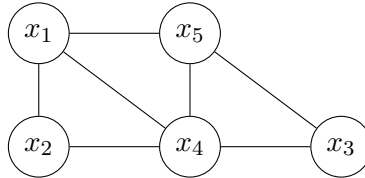


FIGURE 2.1 – Graphe  $G(M)$  d'une matrice symétrique  $M$ .

On peut vérifier que la figure 2.1 présente le graphe  $G(M)$  de la matrice symétrique de l'exemple 2.1.

La présence d'un fill-in à l'itération  $k$ , où  $m_{kk}^{(k)}$  est le pivot, implique l'existence de deux indices  $i, j$  tels que  $m_{ij}^{(k)} = 0$  et  $m_{ij}^{(k+1)} \neq 0$ . On en déduit, d'après l'équation (2.2), que  $m_{ik}^{(k)} \neq 0$ ,  $m_{kj}^{(k)} \neq 0$ . Ceci implique, si  $M$  est une matrice symétrique et d'après la définition du graphe  $G(M)$ , que  $v_i \in N(v_k)$ ,  $v_j \in N(v_k)$  et  $v_i, v_j$  ne sont pas adjacents, c'est-à-dire le sommet  $v_k$  n'est pas un sommet simplicial. Nous voyons cela clairement avec la matrice  $M$  de l'exemple 2.1 : en pivotant par  $m_{11} = 2$ , nous avons produit deux fill-in, car le sommet correspondant à la première ligne dans le graphe  $G(M)$  est  $x_1$ , et que celui-ci n'est pas simplicial. Le lecteur peut vérifier facilement que pivoter par  $m_{22}$  ou  $m_{33}$ , il n'y aura pas de fill-in, car  $x_2$  et  $x_3$  sont des sommets simpliciaux dans  $G(M)$ .

Le théorème ci-dessous montre qu'une matrice symétrique  $M$ , avec  $m_{kk} \neq 0$ ,  $k = 1, \dots, n$  admet un schéma d'élimination parfait si et seulement si le graphe  $G(M)$  admet un ordre d'élimination simplicial (O.E.S). Autrement dit, si et seulement si le graphe  $G(M)$  est un graphe triangulé (voir par exemple [88], où les graphes triangulés sont caractérisés par l'existence d'un O.E.S).

**Théorème 2.1.** Soit  $M$  une matrice symétrique sans entrées diagonales nulles. Les conditions suivantes sont équivalentes

1.  $M$  possède un schéma d'élimination parfait.
2.  $G(M)$  est un graphe triangulé.

Ainsi, pour montrer qu'une matrice symétrique  $M$  ayant toutes ses entrées diagonales non nulles, admet un schéma d'élimination parfait, il suffit de montrer que le graphe  $G(M)$  est triangulé. En 1976, Rose et al [88] ont montré qu'un graphe  $G$  est triangulé si est seulement si l'ordre sur ses sommets calculé par LexBFS est un ordre d'élimination simplicial. Par conséquent, si  $\sigma = (v_1, v_2, \dots, v_n)$  est un O.E.S sur les sommets de  $G(M)$ , réarranger les lignes de la matrice  $M$  selon l'ordre  $\sigma$  ne produit aucun fill-in.



## 2.2 Arête bisimpliciale et schéma d'élimination parfait

Le problème d'existence d'un schéma d'élimination parfait pour une matrice symétrique se ramène au problème de reconnaissance d'un graphe triangulé, ce que l'on peut faire efficacement grâce à l'algorithme LexBFS. Qu'en est-il de l'existence d'un tel schéma pour une matrice non symétrique? Gulombic et Goss [43] ont montré que dans ce cas, ce problème se ramène au problème de reconnaissance d'un graphe biparti d'élimination parfait, une classe de graphes bipartis qu'ils ont introduite en 1978 en utilisant la notion d'arête bisimpliciale.

**Définition 2.4.** Une arête  $xy$  dans un graphe  $G = (V, E)$  est **bisimpliciale** (bisimplicial edge) si le sous-graphe induit par  $N(x) \cup N(y)$  est une biclique (un sous-graphe biparti complet). Autrement dit,  $xy$  est bisimpliciale si

$$\forall a \in N(x), \forall b \in N(y), ab \in E.$$

Si  $xy$  est une arête bisimpliciale, on note  $B_{xy} = (N(x), N(y))$  la biclique induite par  $N(x) \cup N(y)$ .

Dans le graphe  $G$  de la figure 2.2, l'arête  $x_1y_2$  est bisimpliciale, car le sous-graphe  $G[N(x_1) \cup N(y_2)]$  de la même figure, induit par  $N(x_1) \cup N(y_2)$ , est une biclique. De même,  $x_4y_1$  est une arête bisimpliciale.

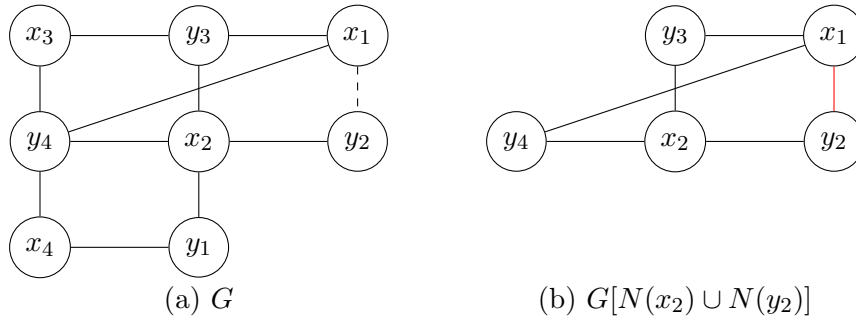


FIGURE 2.2 – Arête bisimpliciale

**Remarque 2.2.** Notons que, par définition d'un sous-graphe induit, si  $N_G(x) \cup N_G(y)$  induit une biclique dans un graphe  $G$ , pour deux sommets  $x$  et  $y$ , alors  $N_{G'}(x) \cup N_{G'}(y)$  induit une biclique dans tout sous-graphe induit  $G'$  contenant les sommets  $x$  et  $y$ . Par conséquent, si  $xy$  est une arête bisimpliciale dans un graphe  $G$ , alors elle est bisimpliciale dans tout sous-graphe induit contenant les sommets  $x$  et  $y$ .

Soit à résoudre, par la méthode de Gauss-Jordan, le système linéaire d'ordre  $n$  suivant

$$Mx = b \tag{2.3}$$

où  $M$  est une matrice inversible non symétrique.

**Définition 2.5.** On définit le graphe biparti  $B(M) = (X, Y, E)$  de la matrice  $M$ , où les ensembles  $X = \{x_1, x_2, \dots, x_n\}$ ,  $Y = \{y_1, y_2, \dots, y_n\}$  représentent respectivement les lignes et les colonnes de  $M$ ; et une paire de sommets  $x_i y_j$  est une arête si et seulement si l'entrée  $m_{ij} \neq 0$ . Notons que le nombre d'arêtes dans  $B(M)$  est égal au nombre d'entrées non nulles dans la matrice  $M$ .

On peut vérifier facilement que le graphe  $G$  de la figure 2.2 est le graphe  $B(M)$  de la matrice  $M$  suivante

$$M = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 5 & 4 & 1 & 1 \\ 0 & 0 & 2 & 4 \\ 2 & 0 & 0 & 6 \end{pmatrix}$$

Notons que, pour avoir un système équivalent au système (2.3), on peut pivoter par une entrée qui n'appartient pas forcément à la diagonale de la matrice  $M$ . En effet, pour calculer la matrice  $M^{(k+1)}$ , en pivotant par l'entrée  $m_{i_0 j_0}^{(k)}$ , l'équation (2.2) devienne

$$m_{ij}^{(k+1)} = \begin{cases} \frac{m_{ij}^{(k)}}{m_{i_0 j_0}^{(k)}} & \text{si } i = i_0 \\ m_{ij}^{(k)} - \frac{m_{i j_0}^{(k)} \times m_{i_0 j}^{(k)}}{m_{i_0 j_0}^{(k)}} & \text{si } i \neq i_0 \end{cases} \quad (2.4)$$

Appliquons la méthode de Gauss-Jordan à la matrice ci-dessus, en pivotant par  $m_{12}$ , puis par  $m_{13}$ .

1. Cas 1, le pivot est  $m_{12} = 2$

$$M^{(1)} = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 5 & 4 & 1 & 1 \\ 0 & 0 & 2 & 4 \\ 2 & 0 & 0 & 6 \end{pmatrix} \Rightarrow M^{(2)} = \begin{pmatrix} 0 & 1 & 1/2 & 3/2 \\ 5 & 0 & -1 & -5 \\ 0 & 0 & 2 & 4 \\ 2 & 0 & 0 & 6 \end{pmatrix}$$

2. Cas 2, le pivot est  $m_{13} = 1$

$$M^{(1)} = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 5 & 4 & 1 & 1 \\ 0 & 0 & 2 & 4 \\ 2 & 0 & 0 & 6 \end{pmatrix} \Rightarrow M^{(2)} = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 5 & 2 & 0 & -2 \\ 0 & -4 & 0 & -2 \\ 2 & 0 & 0 & 6 \end{pmatrix}$$

On voit bien qu'il n'y a aucun fill-in dans le cas 1, car dans le graphe  $B(M)$ , l'arête  $x_1 y_2$ , qui correspond au pivot  $m_{12}$ , est une arête bisimpliciale. Par contre, dans le second cas, l'arête  $x_1 y_3$ , qui correspond au pivot  $m_{13}$ , n'est pas bisimpliciale. En effet,  $y_2 \in N(x_1)$  et  $x_3 \in N(y_3)$  mais  $x_3 y_2$  n'est pas une arête. Nous allons voir ci-dessous qu'une entrée  $m_{ij}^{(k)}$ , prise comme pivot, ne produit pas de fill-in si et seulement si l'arête  $x_i y_j$  est bisimpliciale dans  $B(M)$ .

**Définition 2.6.** Dans un graphe  $G = (V, E)$ , une séquence d'arêtes  $[e_1, e_2, \dots, e_k]$ , deux à deux non adjacentes, constitue un schéma d'élimination parfait si elle vérifie les conditions suivantes :

1. l'arête  $e_i$  est bisimpliciale dans le sous-graphe induit par la suppression des extrémités des arêtes  $e_1, e_2, \dots, e_{i-1}$ , pour  $i = 1, \dots, k$ .
2. l'ensemble  $E - \{e_1, e_2, \dots, e_k\}$  est vide.

Autrement dit, si le graphe obtenu après la suppression de ces arêtes avec leurs extrémités est vide.

**Définition 2.7.** Un graphe biparti  $G = (X, Y, E)$  est un **graphe d'élimination parfait** (perfect elimination bipartite graph) s'il admet un schéma d'élimination parfait pour les arêtes. Dans la suite, nous utilisons l'abréviation GBEP pour désigner un graphe biparti d'élimination parfait.

On peut vérifier que la séquence d'arêtes  $[x_1y_2, x_3y_3, x_2y_4, x_4y_1]$  est un schéma d'élimination parfait pour le graphe  $G$  dans la figure 2.2. Par contre, le graphe de la figure 2.3 n'est pas un graphe d'élimination parfait. En effet, le sous-graphe obtenu après la suppression de l'arête bisimpliciale  $x_4y_1$  est un  $C_6$ , et celui-ci n'a aucune arête bisimpliciale à supprimer.

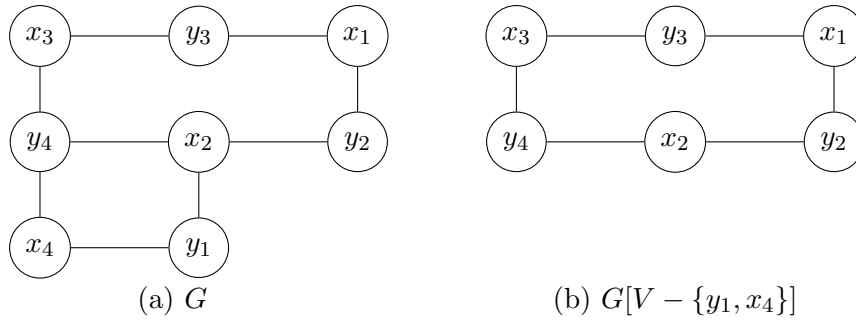


FIGURE 2.3 – Graphe sans schéma d'élimination parfait

Nous venons de voir qu'à toute arête bisimpliciale dans  $G$  correspond un pivot acceptable dans la matrice  $B(M)$ , et vice versa. Le théorème suivant donne une condition nécessaire et suffisante pour l'existence d'un schéma d'élimination parfait pour une matrice non symétrique.

**Théorème 2.2.** [43] Soit  $M$  une matrice non symétrique. Les conditions suivantes sont équivalentes

1.  $M$  possède un schéma d'élimination parfait.
2.  $B(M)$  est un GBEP.

Le graphe de la figure 2.3 montre que le sous-graphe obtenu par la suppression d'une arête bisimpliciale peut ne pas contenir d'arête bisimpliciale. Cependant, le théorème suivant montre que la notion d'arête bisimpliciale est une propriété héréditaire dans un GBEP.

**Théorème 2.3.** [43] *Dans un GBEP, le sous-graphe induit par la suppression des extrémités d'une arête bisimpliciale est un GBEP.*

Ainsi, pour construire un schéma d'élimination parfait dans un GBEP, il n'y a aucune condition sur le choix d'une arête bisimpliciale à supprimer au cours des itérations. On en déduit, qu'un GBEP peut être reconnu par un algorithme glouton de la manière suivante : on cherche une arête bisimpliciale dans  $G$ , on supprime celle-ci ainsi que ses extrémités. On reprend le même travail sur le sous-graphe induit qui en résulte, car ce dernier est un GBEP, et ainsi de suite. Ce processus converge vers un graphe vide si et seulement si  $G$  est un graphe d'élimination parfait. Dans le pire des cas, pour un graphe  $G = (X, Y, E)$ , avec  $|X| = |Y| = n$ , un tel algorithme décide si le graphe  $G$  est un GBEP en  $O(n^5)$  [41]. Dans [41], l'auteur a proposé un algorithme de reconnaissance d'un GBEP avec une complexité de  $O(n^3)$ . En 2013, un nouvel algorithme a été proposé par M. Bomhoff et B. Manthey [8] qui peut reconnaître si un graphe donné est un graphe d'élimination parfait avec une complexité  $O(nm)$ .

La classe des graphes bipartis triangulés est strictement incluse dans celle des graphes d'élimination parfait. En effet, le théorème ci-dessous montre que tout graphe biparti triangulé possède un ordre d'élimination parfait, et donc est un GBEP. Le graphe biparti de la figure 2.4 n'est pas triangulé, car il contient un  $C_6$  sans corde qui est :  $C_6 = (x_1, y_2, x_2, y_3, x_3, y_4, x_1)$ . Ce même graphe est un GBEP, car la séquence  $[x_1y_1, x_2y_2, x_3y_3, x_4y_4]$  est un schéma d'élimination parfait.

**Théorème 2.4.** [43] *Tout graphe biparti triangulé est un graphe d'élimination parfait.*

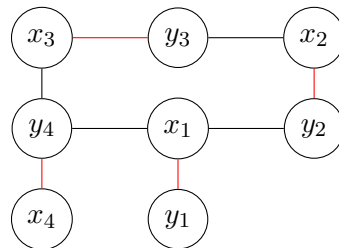


FIGURE 2.4 – GBEP qui n'est pas triangulé

Dans la suite de cette section, nous rappelons quelques propriétés d'une arête bisimpliciale qui seront utiles ultérieurement. Commençons par l'observation suivante qui montre que la distance entre deux sommets appartenant à une même biclique est majorée par deux.

**Observation 2.** [28] Si  $B = (X_B, Y_B)$  est une biclique dans un graphe  $G$ , alors

$$d_G(u, v) \leq 2, \forall u, v \in X_B \cup Y_B$$

De cette observation, on tire l'observation suivante

**Observation 3.** Si  $d_G(u, v) \geq 3$ , alors les sommets  $u$  et  $v$  ne peuvent pas appartenir simultanément à une même biclique.

**Lemme 2.5.** Dans un graphe  $G$ , si  $B_1 = (X_1, Y_1)$  et  $B_2 = (X_2, Y_2)$  sont deux bicliques maximales telles que  $X_1 = X_2$  ou  $Y_1 = Y_2$ , alors

$$B_1 = B_2$$

*Démonstration.* Sans perte de généralité, supposons que  $X_1 = X_2$  mais  $Y_1 \neq Y_2$ , et soit  $y \in Y_1/Y_2$ . On a  $B_1 = (X_1, Y_1)$ , donc  $y \sim X_1$ ; et donc  $y \sim X_2$ . Il en résulte que  $(X_2, Y_2 \cup \{y\})$  est une biclique. Or, ceci contredit le fait que la biclique  $B_2 = (X_2, Y_2)$  est maximale.  $\square$

**Remarque 2.3.** Notons que si  $B = (X, Y)$  est une biclique, alors  $\forall x \in X, \forall y \in Y$ ,

$$X \subseteq N(y) \text{ et } Y \subseteq N(x)$$

Ainsi, si  $xy$  est une arête bisimpliciale, la biclique  $B_{xy} = (N(x), N(y))$  est maximale.

**corollaire 2.6.** Si  $xy$  et  $xz$  sont deux arêtes bisimpliciales, alors

$$B_{xy} = B_{xz}.$$

*Démonstration.* Par définition,  $B_{xy} = (N(x), N(y))$  et  $B_{xz} = (N(x), N(z))$ . D'après la remarque 2.3, ces bicliques sont maximales, et d'après le lemme 2.5, elles sont égales.  $\square$

La proposition ci dessous donne une caractérisation d'une arête bisimpliciale.

**Proposition 2.7.** Soit  $xy$  une arête dans un graphe biparti  $G$ . Les deux conditions suivantes sont équivalentes

1.  $xy$  est une arête bisimpliciale
2.  $xy$  appartient à une unique biclique maximale.

*Démonstration.* Soit  $xy$  une arête bisimpliciale. Par définition,  $B_{xy} = (N(x), N(y))$  est une biclique maximale. D'après la remarque 2.3,  $B_{xy}$  contient toutes les bicliques qui contiennent l'arête  $xy$  et donc elle est unique.

Réciproquement, soit  $B = (X_B, Y_B)$  l'unique biclique maximale contenant l'arête  $xy$ , avec  $x \in X_B, y \in Y_B$ . Notons que  $B$  doit contenir toutes les bicliques qui contiennent l'arête  $xy$ . Supposons que l'arête  $xy$  n'est pas bisimpliciale. Par définition, il existe  $x' \in N(y), y' \in N(x)$  tels que  $x'y' \notin E$ . D'après l'observation 3, les sommets  $x'$  et  $y'$  n'appartiennent pas à une même biclique, donc au moins l'un de ces deux sommets n'appartient pas à  $B$ . Sans perte de généralité, on suppose que  $x' \notin B$ . Par conséquent,  $xy \in B' = (\{y\}, \{x, x'\})$  et  $B$  ne contient pas la biclique  $B'$ , contradiction.  $\square$

Soient  $xy$  une arête bisimpliciale et  $B_{xy} = (N(y), N(x))$  l'unique biclique maximale contenant cette arête. Il est clair que, pour tout  $a \in N(y)$  et pour tout  $b \in N(x)$ ,  $a \sim N(x)$  et  $b \sim N(y)$ . Il en résulte que  $N(x) \subseteq N(a)$  et  $N(y) \subseteq N(b)$ . D'où le lemme suivant :

**Lemme 2.8.** [8] Dans un graphe biparti  $G = (X, Y, E)$ , si  $xy$  est une arête bisimpliciale, alors  $d(x) = \min\{d(a), a \in N(y)\}$  et  $d(y) = \min\{d(b), b \in N(x)\}$ .

### 2.3 Sommet bisimplicial

Comme nous l'avons vu, une arête bisimpliciale  $xy$  appartient à une unique biclique maximale  $B_{xy}$ . Cependant ses extrémités  $x$  et  $y$  peuvent appartenir à plusieurs bicliques maximales. En effet, dans la figure 2.5, on voit bien que  $xd$  est une arête bisimpliciale appartenant à l'unique biclique maximale  $B_{xd} = (\{x, v\}, \{d, c\})$  tandis que ses extrémités  $x$  et  $d$  appartiennent à plusieurs bicliques maximales qui sont  $B_{xd}, (\{x, v, u, w\}, \{c\}), (\{v\}, \{d, c, b\})$ . Il est naturel de se demander quand est-ce qu'un sommet appartient à une unique biclique maximale? Existe-t-il une classe de graphes bipartis qui en admette de tels sommets et qui soit stable par la suppression de ceux-ci?

**Définition 2.8.** Dans un graphe biparti  $G$ , un sommet  $x$  est dit **bisimplicial** (bisimplicial vertex) si l'arête  $xy$  est bisimpliciale pour tout sommet  $y$  dans  $N(x)$ . Autrement dit, si toutes les arêtes incidentes à  $x$  sont des arêtes bisimpliciales.

Dans la figure 2.5, on peut vérifier que toutes les arêtes incidentes au sommet  $a$  sont bisimpliciales, donc  $a$  est un sommet bisimplicial.

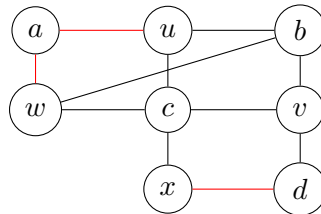


FIGURE 2.5 – Sommet bisimplicial

**Remarque 2.4.** La notion de sommet bisimplicial comme celle d'arête bisimpliciale, est héritée par tout sous-graphe induit le contenant (voir la remarque 2.2).

Les propriétés d'un sommet bisimplicial sont données par la proposition suivante :

**Proposition 2.9.** Soit  $x$  un sommet dans un graphe biparti  $G = (X, Y, E)$ . Alors, les assertions suivantes sont équivalentes.

1. le sommet  $x$  est bisimplicial;

2. le sommet  $x$  appartient à une unique biclique maximale notée  $B_x$  ;
3.  $D_G(x, 2) = \{y \in X \cup Y, d(x, y) \leq 2\} = B_x$  ;
4.  $\forall y, y' \in N(x), N(y) = N(y')$  (les voisins de  $x$  sont des faux jumeaux).

*Démonstration.* Sans perte de généralité, supposons que  $x \in X$

1) $\Rightarrow$ 2)

Rappelons que si  $x$  est un sommet bisimplicial, alors  $xy$  est une arête bisimpliciale pour tout  $y \in N(x)$ , donc les bicliques contenant  $x$  sont de la forme  $B_{xy}$ , avec  $y \in N(x)$ . Il en résulte, d'après le corollaire 2.6, que toutes ces bicliques sont égales. D'où, le sommet  $x$  appartient à une unique biclique maximale.

2) $\Rightarrow$ 3)

Il est clair que  $B_x \subseteq D_G(x, 2)$ .

Montrons que  $D_G(x, 2) \subseteq B_x$ .

Soit  $y \in D_G(x, 2)$ , alors  $d_G(x, y) \leq 2$ , et donc il existe au moins une biclique maximale qui contient les sommets  $x$  et  $y$ . Puisque  $x$  appartient à une unique biclique maximale  $B_x$ , donc  $y \in B_x$ . Par conséquent  $D_G(x, 2) \subseteq B_x$ .

3) $\Rightarrow$  4)

Soient  $y, y' \in N(x)$ . Alors  $y, y' \in B_x$ . Soit  $x' \in N(y)$ , alors  $d_G(x, x') = 2$ , donc  $x' \in B_x = D_G(x, 2)$  et donc  $x' \in N(y')$ . Par un raisonnement identique, on montre que, si  $x' \in N(y')$ , alors  $x' \in N(y)$ . Par conséquent  $N(y) = N(y')$ .

4) $\Rightarrow$  1)

Supposons que  $x$  n'est pas bisimplicial, donc il existe  $y \in N(x)$  tel que  $xy$  n'est pas bisimpliciale. On en déduit qu'il existe  $y' \in N(x)$  et  $x' \in N(y)$  tels que  $x'y' \notin E$ , c'est-à-dire  $x' \in N(y)$  et  $x' \notin N(y')$ . Par conséquent,  $N(y) \neq N(y')$ .  $\square$

**corollaire 2.10.** *Si  $x$  est un sommet bisimplicial, alors pour tout  $y \in N(x)$ ,  $B_x = B_{xy}$ .*

*Démonstration.* D'après la proposition 2.9, assertion 2,  $x$  appartient à une unique biclique maximale  $B_x$ . D'autre part et par définition,  $xy$  est une arête bisimpliciale pour tout  $y \in N(x)$ , et donc  $x \in B_{xy}$  qui est aussi unique d'après la proposition 2.7. Par conséquent  $B_x = B_{xy}$ .  $\square$

Plusieurs sommets bisimpliciaux peuvent appartenir à une même biclique. Mais, deux sommets bisimpliciaux dans un graphe biparti, connexe et non complet ne sont jamais adjacents d'après le lemme suivant.

**Lemme 2.11.** *Soient  $x, y$  deux sommets adjacents dans un graphe  $G$  biparti connexe. Les deux conditions suivantes sont équivalentes*

1.  $x$  et  $y$  sont bisimpliciaux.
2. le graphe  $G$  est complet.

*Démonstration.* Si  $G$  est un graphe biparti complet, alors  $G$  est l'unique biclique maximale contenant tous ses sommets et donc tous les sommets de  $G$  sont bisimpliciaux.

Réciproquement, puisque  $x, y$  sont adjacents et bisimpliciaux, alors  $\forall x' \in N(y), \forall y' \in N(x), N(x) = N(x')$  et  $N(y) = N(y')$  (d'après l'assertion 4 ci-dessus). Ceci implique que les éléments de la biclique  $B_{xy} = (N(x), N(y))$  n'ont aucun voisin à l'extérieur de  $B_{xy}$ . Puisque  $G$  est connexe, il en résulte que  $G = B_{xy}$ .  $\square$

**corollaire 2.12.** *Si  $x$  est un sommet bisimplicial, alors tout sommet  $x'$  tel que  $N(x) = N(x')$  est aussi un sommet bisimplicial. Autrement dit, si  $x$  et  $x'$  sont bisimpliciaux tels que  $d(x, x') = 2$ , alors  $N(x) = N(x')$ .*

*Démonstration.* C'est une conséquence directe de la proposition 2.9, assertion 4.  $\square$

Le lemme suivant montre que le graphe partiel obtenu par la suppression des arêtes d'une biclique  $B_x$ , où  $x$  est un sommet bisimplicial, contient une seule composante non triviale et des sommets isolés.

**Lemme 2.13.** *Soient  $G$  un graphe biparti connexe et non complet,  $x$  un sommet bisimplicial et  $G'$  le graphe partiel obtenu par la suppression des arêtes de  $B_x$ . Alors, pour qu'un sommet  $y$  dans  $G'$  soit isolé il faut et il suffit que  $y = x$  ou  $y \in N(x)$  ou  $N(x) = N(y)$ .*

*Démonstration.* 1. **La condition est nécessaire :**

Par définition, la biclique  $B_x$  contient tous les voisins de  $x$ , elle contient donc toutes les arêtes incidentes à  $x$ , par conséquent, le sommet  $x$  est isolé dans  $G'$ . Pour les mêmes raisons, tout sommet  $y$  vérifiant  $N(x) = N(y)$  est aussi isolé dans  $G'$ . Soit maintenant  $y \in N(x)$ . Par définition,  $xy$  est une arête bisimpliciale et donc la biclique  $B_x = B_{xy} = (N(x), N(y))$  contient tous les voisins de  $y$ , et donc  $y$  est isolé dans  $G'$ .

2. **La condition est suffisante :**

Soit  $y$  un sommet tel que  $N(x) \neq N(y)$ ,  $y \notin N(x)$  et  $x \neq y$ . On a deux cas :

- (a) Le sommet  $y$  appartient à  $B_x$ . Soit  $z$  un sommet adjacent à  $x$ . Notons que dans ce cas  $B_x = (N(x), N(z))$ . Puisque,  $y \notin N(x)$ , alors  $y \in N(z)$ , et puisque  $x \in N(z)$ ,  $x \neq y$  et  $N(x) \neq N(y)$ , alors  $N(x) \subset N(y)$ . Ceci montre que le sommet  $y$  possède au moins un voisin en dehors de  $B_x$ , et donc  $y$  est l'extrémité d'au moins une arête que  $B_x$  ne contient pas. Par conséquent,  $y$  n'est pas isolé dans  $G'$ .
- (b) Le sommet  $y$  n'appartient pas à  $B_x$ , dans ce cas la biclique  $B_x$  ne contient aucune arête incidente à  $y$ , donc  $y$  n'est pas isolé dans  $G'$ .

$\square$

Dans la figure 2.6, nous avons un graphe biparti  $G$ , et le sous-graphe  $G'$  obtenu par la suppression des arêtes de la biclique maximale  $B = (\{x, v\}, \{c, d\})$ . On voit bien que le sous-graphe défini par la composante non triviale n'est pas un sous-graphe induit, il



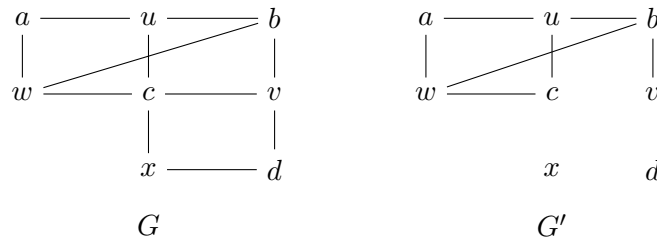


FIGURE 2.6 – Suppression des arêtes d'une biclique maximale

manque l'arête  $cv$ , pourtant  $xd$  est une arête bisimpliciale dans  $B$ . Ainsi, les composantes connexes qui en résultent par la suppression des arêtes d'une biclique maximale ne sont pas des sous-graphes induits en général. Cependant, les choses sont différentes si la biclique maximale  $B$  contient un sommet bisimplicial.

**Lemme 2.14.** *Soient  $G$  un graphe biparti,  $x$  un sommet bisimplicial et  $G'$  le sous-graphe obtenu par suppression des arêtes de la biclique  $B_x$ . Alors l'unique composante connexe non triviale de  $G'$  est un sous-graphe induit.*

*Démonstration.* Soit  $C$  cette composante connexe et soient  $a, b$  deux sommets non adjacents dans  $C$ . Par définition d'un sous-graphe induit, les sommets  $a$  et  $b$  ne doivent pas être adjacents dans  $G$ . Par l'absurde, supposons le contraire, c'est-à-dire  $a \sim_G b$ . Ceci implique que l'arête  $ab \in B_x$ , donc  $a \sim_G x$  ou  $b \sim_G x$ . Sans perte de généralité, on suppose que  $x \sim_G a$ , donc  $ax$  est une arête bisimpliciale car  $x$  est un sommet bisimplicial. Mais, d'après le lemme 2.13, la suppression des arêtes de  $B_x$  implique que  $a$  est un sommet isolé dans le sous-graphe  $G'$ . Par conséquent, on a une contradiction du fait que  $a$  et  $b$  appartiennent à une même composante connexe.  $\square$

Remarquons que dans la figure 2.6, le sommet  $a$  est bisimplicial dans  $G$ , et on peut vérifier que les composantes connexes obtenues par la suppression des arêtes de  $B_a$  sont des sous-graphes induits.

## 2.4 Reconnaissance d'un graphe biparti distance héréditaire

Nous allons voir dans cette section que la classe de graphes Bipartis distance héréditaire possède de bonnes propriétés algorithmiques, des propriétés similaires à celles des graphes triangulés. Dirac [25] a montré qu'un graphe triangulé admet au moins un sommet simplicial, il a affirmé qu'un graphe triangulé qui n'est pas une clique contient au moins deux sommets simpliciaux non adjacents. Fulkerson et Gross [35] ont montré qu'un graphe est triangulé si et seulement si il admet un ordre d'élimination simplicial. Le calcul d'un ordre d'élimination simplicial, et donc la reconnaissance d'un graphe triangulé, peut se faire en  $O(n + m)$  par l'algorithme LexBFS qui est dû à Rose

et al. [88], ou par l'algorithme de **recherche par ordre de cardinalité maximum** (MCS, pour maximum cardinality search) et qui est dû à Tarjan et Yannakakis [97].

Étant donné un graphe  $G = (V(G), E(G))$ , rappelons que le carré d'un graphe  $G$  est le graphe  $G^2$  ayant les mêmes sommets que  $G$ , et deux sommets  $u, v$  sont adjacents dans  $G^2$  (i.e.  $u \sim_{G^2} v$ ) si et seulement si leur distance dans  $G$  est majorée par 2 (c-à-d.  $d_G(u, v) \leq 2$ ).

La figure 2.7 montre un graphe biparti et son carré. Le lemme ci-dessous donne une relation entre un sommet simplicial et un sommet bisimplicial.

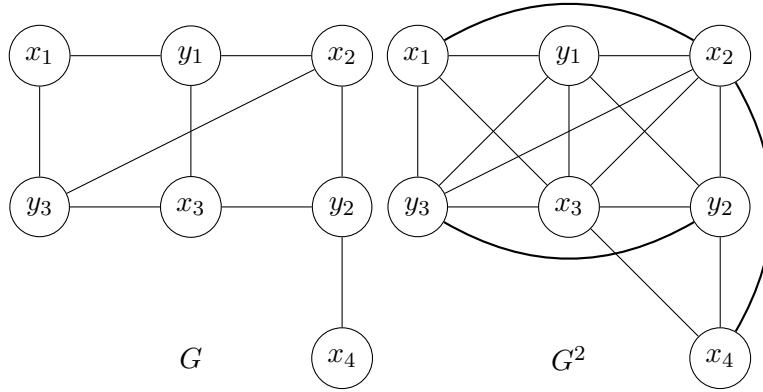


FIGURE 2.7 – Graphe  $G$  et son carré

**Lemme 2.15.** Soient  $G = (V, E)$  un graphe biparti et  $v$  un sommet. Les deux assertions suivantes sont équivalentes

1.  $v$  est bisimplicial dans  $G$
2.  $v$  est simplicial dans  $G^2$ .

*Démonstration.* Supposons que le sommet  $v$  est bisimplicial dans  $G$  et soient  $u, w$  deux voisins de  $v$  dans  $G^2$ . Il s'agit de montrer que  $u \sim_{G^2} w$ . Par définition de  $G^2$ , on a  $d_G(v, u) \leq 2$  et  $d_G(v, w) \leq 2$ ; et donc les sommets  $u$  et  $w$  sont dans une même biclique  $B_v = D_G(v, 2)$ , c'est-à-dire  $d_G(u, w) \leq 2$  (voir observation 2). Ceci implique  $d_{G^2}(u, w) = 1$ , et donc  $v$  est un sommet simplicial dans  $G^2$ .

Réciproquement, Supposons que  $v$  est un sommet simplicial dans  $G^2$ . Il s'agit de montrer que  $D_G(v, 2)$  est une biclique, c'est-à-dire  $d_G(u, w) \leq 2$ , pour tout  $u, w \in D_G(v, 2)$  (voir observation 2). Soit  $u, w \in D_G(v, 2)$ . Alors  $d_{G^2}(v, u) = 1$  et  $d_{G^2}(v, w) = 1$ ; et puisque  $v$  est simplicial dans  $G^2$ , alors les sommets  $u$  et  $w$  sont adjacents dans  $G^2$ . On en déduit  $d_G(u, w) \leq 2$  et donc  $D_G(v, 2)$  est une biclique.  $\square$

Le lemme suivant dû à F. Nicolai [76, 77] est cité dans [93] sous forme d'un corollaire.

**Lemme 2.16.** Soit  $G = (X, Y, E)$  un graphe biparti. Les conditions suivantes sont équivalentes :

1.  $G$  est distance héréditaire ;
2. Il existe dans  $G$  un ordre d'élimination bisimplicial.

**corollaire 2.17.** Un graphe biparti  $G$  est distance héréditaire si et seulement si son carré  $G^2$  est un graphe triangulé.

*Démonstration.* Sachant qu'un graphe est triangulé si et seulement s'il admet un ordre d'élimination simplicial [35], ce corollaire est une conséquence directe des lemmes 2.15 et 2.16.  $\square$

**Lemme 2.18.** Soit  $G$  un graphe biparti distance héréditaire ordonné par *lexBFS*  $\sigma$  et soient  $x, x'$  deux sommets tels que  $N^+(x) \cap N^+(x') \neq \emptyset$ . Alors

$$x <_{\sigma} x' \Rightarrow N^+(x) \subseteq N^+(x')$$

*Démonstration.* Puisque le graphe  $G$  est biparti et  $N^+(x) \cap N^+(x') \neq \emptyset$ , alors les sommets  $x$  et  $x'$  sont dans un même niveau et ne sont pas adjacents.

1. Si  $N^+(x) = N^+(x')$  ou si  $x$  est un sommet pendant, alors rien à démontrer.

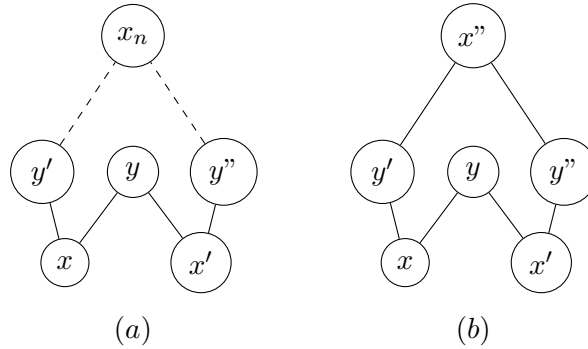


FIGURE 2.8 – Cas  $N^+(x) \neq N^+(x')$

2. Si  $N^+(x) \neq N^+(x')$  et  $x$  n'est pas un sommet pendant.

Par l'absurde, supposons que  $N^+(x)$  n'est pas inclus dans  $N^+(x')$ . Alors il existe  $y' \in N^+(x) \setminus N^+(x')$ . Puisque  $x < x'$ , alors par définition de *LexBFS*, il existe  $y'' \in N^+(x')$  tel que  $y' <_{\sigma} y''$ . Il en résulte que par les points  $x, x', y, y', y''$  et  $x_n$  passe un cycle (voir 2.8 (a)) de longueur supérieure ou égale à 6. Le graphe  $G$  étant distance héréditaire, donc il existe un cycle  $\mu = (x, y, x', y'', x'', y')$  (voir 2.8 (b)); et ce cycle doit avoir au moins deux cordes. Puisque  $y'' \notin N^+(x)$ , les cordes de  $\mu$  sont  $yx''$  et  $x'y'$ . Mais, le fait que  $x'y'$  soit une corde contredit l'hypothèse que  $y' \in N^+(x) \setminus N^+(x')$ .

□

Le théorème ci-dessous montre que l'ordre d'élimination bisimplicial dans un graphe biparti distance héréditaire peut être calculé par l'algorithme LexBFS, et donc avec une complexité linéaire.

**Théorème 2.19.** *Soient  $G$  un graphe biparti et  $\sigma = (v_1, v_2, \dots, v_n)$  un ordre LexBFS sur ses sommets. Les conditions suivantes sont équivalentes*

1.  $G$  est distance héréditaire ;
2.  $\sigma$  est un ordre d'élimination bisimplicial.

*Démonstration.* Soit  $G$  un graphe biparti distance héréditaire ordonné par LexBFS  $\sigma$ . Puisque pour  $i = 1, \dots, n$ , la restriction de LexBFS  $\sigma$  sur l'ensemble  $\{v_i, v_{i+1}, \dots, v_n\}$  est aussi un LexBFS, alors pour montrer que  $\sigma$  est un ordre d'élimination bisimplicial, il suffit de montrer que  $v_1$  est un sommet bisimplicial.

Soit  $x \in N(v_1) = N^+(v_1)$ . Il s'agit de montrer que l'arête  $v_1x$  est bisimpliciale. Soient  $x', v' \in V$  tels que  $v_1x', xv' \in E$ . Montrons que  $x'v' \in E$ . On a deux cas selon que  $v'$  est un voisin inférieur ( $v' < x$ ) ou  $v'$  est un voisin supérieur ( $x < v'$ ) de  $x$ .

1.  $v' < x$  (i.e.  $v' \in N^-(x)$ ), voir la figure 2.9 (a)  
Puisque  $v_1 \rightarrow x$ ,  $v' \rightarrow x$  et  $x' \in N^+(v_1)$ , alors, d'après le lemme 2.18,  $N^+(v_1) \subseteq N^+(v')$ , et donc  $x' \in N^+(v')$ , c'est-à-dire  $x'v' \in E$ . Par conséquent,  $v_1x$  est une arête bisimpliciale.

2.  $x < v'$ , voir la figure 2.9 (b)

Remarquons que, par définition de LexBFS, il existe un chemin  $\mu^+[x', v_n]$  allant

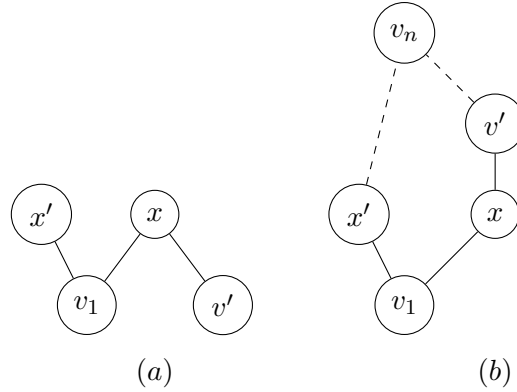


FIGURE 2.9 – (a)  $v' < x$ ; (b)  $x < v'$

de  $x'$  à  $v_n$  et un autre chemin  $\mu^+[v', v_n]$  allant de  $v'$  à  $v_n$ . Là aussi, nous avons deux cas selon que  $v'$  appartient ou n'appartient pas à  $\mu^+[x', v_n]$

- (a) Si  $v' \in \mu^+[x', v_n]$  (voir la figure 2.10.(a)), alors  $x'v' \in E$  et donc  $v_1x$  est une arête bisimpliciale .
- (b) si  $v' \notin \mu^+[x', v_n]$ , alors il existe un sommet  $v'' \neq v'$  tel que les sommets  $x', v_1, x, v', v_n$  appartiennent à un cycle de longueur supérieure à 6 (voir la figure 2.10.(b)). Soit maintenant  $x''$  un sommet tel que  $v'' \rightarrow x''$  (ce sommet existe et il peut coïncider avec  $v_n$ ), voir la figure 2.10.(c). Il en résulte que les sommets  $x'', v'', x', v_1, x, v'$  appartiennent à un même cycle, et puisque  $G$  est un graphe biparti distance héréditaire, alors  $\mu = (v'', x', v_1, x, v', x'')$ , lui-même est un cycle de longueur 6 et ayant au moins deux cordes. Puisque une arête joint deux sommets appartenant à deux niveaux consécutifs et vu la forme du cycle  $\mu$ , ces cordes sont  $x'v'$  et  $xv''$ , et donc l'arête  $v_1x$  est bisimpliciale.

Dans tous les cas,  $v_1x$  est une arête bisimpliciale d'où, le sommets  $v_1$  est bisimplicial.

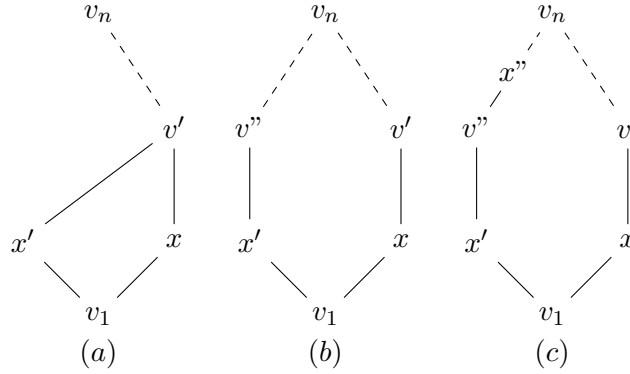


FIGURE 2.10 – Deux cas :  $v' \in \mu^+[x', v_n]$  ou  $v' \notin \mu^+[x', v_n]$

Réciproquement, si  $\sigma$  est un ordre d'élimination bisimplicial dans  $G$ , alors d'après le lemme 2.16, le graphe  $G$  est un graphe biparti distance héréditaire.  $\square$

Le théorème 2.19 montre aussi que l'algorithme LexBFS permet de reconnaître efficacement un graphe biparti distance héréditaire. En effet, étant donné un graphe  $G$  ayant  $n$  sommets et  $m$  arêtes, il suffit d'appliquer LexBFS sur les sommets de  $G$  et vérifier si ce dernier est un ordre d'élimination bisimplicial. L'application de LexBFS peut se faire en  $O(n + m)$  [88]. Par ailleurs, vérifier si un sommet quelconque  $x_i$  est bisimplicial peut se faire en un temps  $d(x_i) \times d(y)$ , où  $y \in N(x)$ , et donc ce temps est majoré par  $\Delta(G)d(x_i)$ . Il en résulte qu'au pire des cas, la complexité de tester si LexBFS est un ordre d'élimination bisimplicial est en  $O(\sum_{i=1}^n \Delta(G)d(x_i)) = O(2\Delta(G)m)$ .

Pour décider si un graphe biparti ordonné par LexBFS est distance héréditaire, l'algorithme décrit ci-dessus nécessite  $n$  étapes, car il doit parcourir et tester tous les sommets de  $G$ . En se basant sur l'assertion 4 de la proposition 2.9, nous présentons une nouvelle caractérisation d'un graphe biparti distance héréditaire qui permettra par

la suite, de réduire ce nombre d'étapes et aboutir à un algorithme de reconnaissance linéaire pour cette classe de graphes.

On peut vérifier facilement qu'un sommet bisimplicial ne peut pas appartenir ni à un domino ni à un trou  $C_n, n \geq 6$ . Le lemme suivant montre qu'il ne peut pas avoir plus d'un seul voisin en commun avec un domino ou un trou.

**Lemme 2.20.** *Soient  $G$  un graphe biparti,  $x$  un sommet bisimplicial et  $\mu$  un domino. Alors  $|N(x) \cap \mu| \leq 1$*

*Démonstration.* Soit  $\mu = (a, b, c, d, e, f)$  un domino. Par l'absurde et sans perte de généralité, on suppose que  $N(x) \cap \mu = \{a, c\}$ . Les deux configurations possibles sont présentées par la figure 2.11. Mais, d'après la proposition 2.9, assertion 4, on aurait  $N(a) = N(c)$  et donc on aurait  $c \sim f$  et  $a \sim d$ . Or, ceci est en contradiction avec le fait que  $\mu$  est un domino.

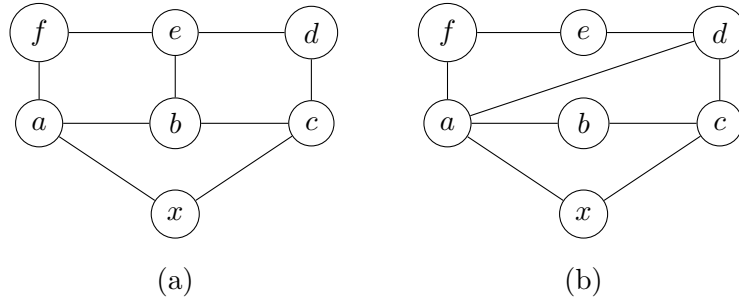


FIGURE 2.11 – Domino et sommet bisimplicial

□

**Remarque 2.5.** Pour la même raison que dans le lemme 2.20, un sommet bisimplicial ne peut pas avoir plus d'un voisin en commun avec un trou  $C_n, n \geq 6$ .

**corollaire 2.21.** *Si  $x$  est un sommet bisimplicial qui n'est pas pendant et  $\mu$  un domino tel que  $\mu \cap N(x) = \{y\}$ , alors pour tout  $z \in N(x) \setminus \{y\}$ ,  $\mu' = \mu \setminus \{y\} \cup \{z\}$  est un domino.*

*Démonstration.* Puisque  $y$  et  $z$  sont des voisins de  $x$ , alors  $N(y) = N(z)$  (assertion 4 de la proposition 2.9) et donc  $\mu' = \mu \setminus \{y\} \cup \{z\}$  est isomorphe à  $\mu$ . □

Le corollaire 2.21 montre que si  $x$  est un sommet bisimplicial dans un graphe  $G$  qui contient un domino (resp. un trou), alors le sous-graphe induit par la suppression du sommet  $x$  et ses voisins, excepté un seul de ses voisins, contient un domino (resp. un trou).

Rappelons qu'un graphe biparti distance héréditaire contient au moins un sommet bisimplicial, et que tout sous-graphe induit connexe d'un graphe distance héréditaire est un graphe distance héréditaire. Par ailleurs, les voisins de tout sommet bisimplicial

$v$  sont des faux jumeaux (assertion 4, proposition 2.9), donc si  $G$  est connexe, alors le graphe obtenu par la suppression de  $v$  et tous ses voisins excepté un seul (autrement dit, le sous-graphe induit par les sommets  $(V - N[v]) \cup \{u\}$ , où  $u$  est l'un des voisins de  $v$ ) est aussi connexe ; en particulier, le sous-graphe obtenu d'un graphe connexe par la suppression d'un sommet pendant est connexe. D'où la définition suivante

**Définition 2.9.** Étant donné un graphe biparti distance héréditaire connexe  $G = (X, Y, E)$ , on définit la séquence de graphes connexes  $H_0, H_1, \dots, H_k$  comme suit

1.  $H_0 = G$  ;
2. pour  $i > 0$  :
  - (a)  $H_i$  est le sous-graphe induit de  $H_{i-1}$  par la suppression d'un sommet bisimplicial  $x$  et tous ses voisins, excepté un seul, si  $x$  n'est pas pendant dans  $H_{i-1}$  ;
  - (b)  $H_i$  est le sous-graphe induit de  $H_{i-1}$  par la suppression d'un sommet bisimplicial  $x$ , si  $x$  est un sommet pendant dans  $H_{i-1}$ .

**Exemple 2.2.** On peut vérifier que l'ordre des sommets sur le graphe biparti  $G$  de la figure 2.12 est un LexBFS. De plus, cet ordre est un ordre d'élimination bisimplicial, donc  $G$  est un graphe biparti distance héréditaire. Le sommet  $x_1$  est bisimplicial et ses

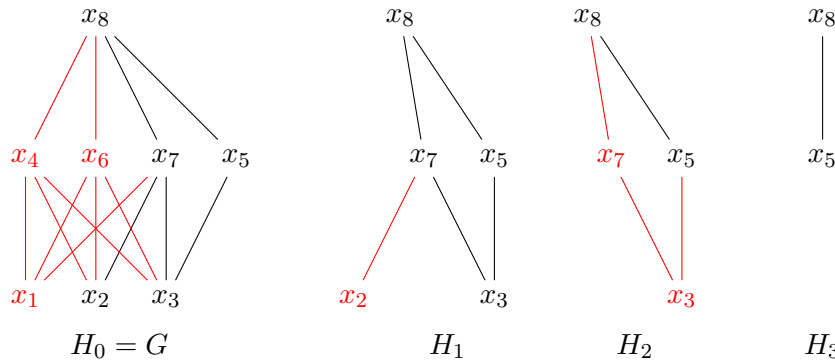


FIGURE 2.12 – Séquence de graphes calculés selon la définition 2.9

voisins sont  $x_4, x_6$  et  $x_7$ . On laisse  $x_7$ , et on supprime  $x_1, x_4$  et  $x_6$  et on aura le sous-graphe  $H_1$ .  $x_2$  est un sommet pendant dans  $H_1$  ; on le supprime et on aura le sous-graphe  $H_2$ . Dans  $H_2$ , le sommet  $x_3$  est bisimplicial avec deux voisins  $x_5$  et  $x_7$ . En supprimant  $x_3$  et  $x_7$ , on trouve le graphe  $H_3$ . En supprimant  $x_5$  dans  $H_3$ , on trouve le graphe  $H_4$  réduit au sommet  $x_8$ , et donc la séquence s'arrête.

**Théorème 2.22.** Pour qu'un graphe  $G$  admette une séquence de graphes  $H_0 = G, H_1, \dots, H_k$  définie selon la définition 2.9 et convergeant vers un graphe  $H_k = (V_k, E_k)$  vide, il faut et il suffit que  $G$  soit un graphe biparti distance héréditaire.

*Démonstration.* Pour les raisons évoquées ci-dessus, la condition est suffisante.

La condition est nécessaire, car si le graphe  $G$  n'est pas distance héréditaire, alors il contient un domino ou un trou, et d'après le lemme 2.20 et le corollaire 2.21, le sous-graphe induit par la suppression d'un sommet bisimplicial ou par la suppression d'un sommet bisimplicial et ses voisins excepté un seul conserve ce domino ou ce trou. Puisque  $G$  est fini, alors la séquence définie ci-dessus converge forcément vers un graphe  $H_k = (V_k, E_k)$  non vide et sans aucun sommet bisimplicial.  $\square$

Le Théorème 2.22 propose un nouvel algorithme pour la reconnaissance d'un graphe biparti distance héréditaire. En effet, dans ce cas, on cherche à chaque itération, un sommet bisimplicial qu'on va supprimer avec tous ses voisins sauf un seul. Nous reprenons le même travail sur le sous-graphe qui en résulte jusqu'à ce que l'ensemble des sommets se réduits à l'ensemble vide; et si le graphe de départ n'est pas un biparti distance héréditaire, l'algorithme s'arrête avant que l'ensemble des sommets soit vide. L'avantage ici, c'est que le nombre d'itérations est beaucoup moins important que dans l'algorithme proposé par le théorème 2.19, car le nombre de sommets, dans le passage d'une étape à une autre, est réduit d'au moins deux sommets si le sommet bisimplicial qui correspond n'est pas pendant. Cependant, nous ne pouvons pas affirmer que le théorème 2.22 propose un algorithme plus efficace que celui proposé par le théorème 2.19, car à présent, on ne dispose d'aucun moyen indiquant comment choisir le sommet bisimplicial dans chaque étape.

**Lemme 2.23.** *Soit  $G = (V, E)$  un graphe biparti distance héréditaire connexe et ordonné par LexBFS  $\sigma = (v_1, v_2, \dots, v_n)$ . Alors, la restriction de  $\sigma$  sur le sous-graphe  $G'$  obtenu par la suppression de  $v_1$  ou par la suppression de  $v_1$  et ses voisins, excepté un seul, est aussi un LexBFS.*

*Démonstration.* Soit  $A$  l'ensemble des sommets de  $G'$ . Par définition, il s'agit de montrer que pour  $u, v \in A$ ,

$$u <_{\sigma} v \Rightarrow \forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) \cap A : u' \leq_{\sigma} v' \quad (2.5)$$

1. Si  $v_1$  est pendant, alors  $A = V - \{v_1\}$ . On en déduit que  $\forall u \in A, N^+(u) \cap A = N^+(u)$ , et puisque  $\sigma$  est un LexBFS sur  $V$ , il l'est aussi sur  $A$ .
2. Si  $v_1$  n'est pas pendant, alors  $A = (V - N[v_1]) \cup \{w\}$ , où  $w$  est l'un des voisins de  $v_1$ . Soient  $N_0, N_1, \dots, N_k$  les niveaux de  $G$  définis par  $\sigma$  (voir la figure 1.7). Notons que  $v_1 \in N_k$  et  $N(v_1) = N^+(v_1) \subseteq N_{k-1}$ .

Soient, maintenant,  $u, v \in A$  tels que  $u <_{\sigma} v$ . Selon que le sommet  $v \in N_k$  ou  $v \notin N_k$ , nous avons deux cas

- (a) Si  $v \notin N_k$ , alors  $N^+(v) \cap N_{k-1} = \emptyset$  et donc  $N^+(v) \cap A = N^+(v)$ . Par ailleurs et puisque  $\sigma$  est un LexBFS sur  $V$ , on a  $\forall u' \in N^+(u), \exists v' \in N^+(v)$  tels que  $u' \leq_{\sigma} v'$ , et donc  $\forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) = N^+(v) \cap A : u' \leq_{\sigma} v'$ ; et par conséquent, l'équation (2.5) est vraie.



- (b) Si  $v \in N_k$ , alors même  $u \in N_k$  car  $u < v$ ; et selon que  $N^+(u) \cap N^+(v) = \emptyset$  ou  $N^+(u) \cap N^+(v) \neq \emptyset$ , nous avons deux cas
- i. Si  $N^+(u) \cap N^+(v) \neq \emptyset$ , et puisque  $G$  est un graphe biparti distance héréditaire, alors  $N^+(u) \subseteq N^+(v)$  (voir le lemme 2.18), et ceci implique  $N^+(u) \cap A \subseteq N^+(v) \cap A$ .  $\sigma$  étant un LexBFS sur  $V$ , donc  $\forall u' \in N^+(u), \exists v' \in N^+(v)$  tels que  $u' \leq_\sigma v'$ , et donc  $\forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) \cap A : u' \leq_\sigma v'$ . Par conséquent, l'équation (2.5) est vraie.
  - ii. Si  $N^+(u) \cap N^+(v) = \emptyset$ , alors, d'après le lemme 1.3,

$$\exists v' \in N^+(v), \forall u' \in N^+(u), u' <_\sigma v' \quad (2.6)$$

Notons que  $v' \notin N^+[v_1]$ , sinon on aurait  $u <_\sigma v_1$  et ceci est impossible. Ainsi,  $v' \in A$ , et donc  $v' \in N^+(v) \cap A$ . Ainsi, l'équation (2.6) implique

$$\exists v' \in N^+(v) \cap A, \forall u' \in N^+(u), u' <_\sigma v' \quad (2.7)$$

ce qui implique

$$\exists v' \in N^+(v) \cap A, \forall u' \in N^+(u) \cap A, u' <_\sigma v' \quad (2.8)$$

D'où, l'équation (2.5) est vraie.

Par conséquent, la restriction de  $\sigma$  sur  $A$  est un lexBFS.  $\square$

Le lemme 2.23 et le théorème 2.22 conduisent à un algorithme plus efficace que celui proposé par le théorème 2.19. En effet, en appliquant le théorème 2.22 à un graphe muni d'un ordre LexBFS, le lemme 2.23 affirme que si le graphe de départ est un graphe biparti distance héréditaire, alors le plus petit sommet dans le sous-graphe obtenu par la suppression d'un sommet bisimplicial et ses voisins, excepté un seul, est un sommet bisimplicial.

### Algorithme 2.1. Reconnaissance d'un graphe biparti distance héréditaire

**Entrée** : Un biparti connexe  $G = (V, E)$  avec  $|V| = n$ ,  $|E| = m$  et un ordre lexBFS  $\sigma = (x_1, x_2, \dots, x_n)$ .

**Sortie** :  $G = (V, E)$  est un graphe biparti distance héréditaire si et seulement si  $\sigma$  est un ordre d'élimination bisimplicial.

1. **debut**  $W = V$ ,  $x = x_1 = \min W$ ,  $m(x) \in N^+(x)$  //  $m(x)$  est l'un des voisins supérieurs de  $x$
2. **Tant que**  $m(x) \neq x_n$  **faire** // si  $m(x) = x_n$  alors  $W$  induit la biclique  $B_x$
3. **Si**  $x$  est pendant **faire**
4.  $W = W - \{x\}$ ,  $x = \min_\sigma W$  //  $\min_\sigma = \text{minimum pour } \sigma \text{ dans } W$
5. **Sinon**
6. **Pour tout**  $y \in N^+(x) \setminus m(x)$  **faire**

7. **Si**  $N_W(y) \neq N_W(m(x))$  **alors**  $G$  n'est pas un  $(6, 2)$ -chordal
8. **Sinon**
9.  $W = (W - N^+[x]) \cup \{m(x)\}$ ;  $x = \min_\sigma W$
10. **fin si**
11. **fin pour**
12. **fin si**
13. **fin tant que**
14. Retourner vrai
15. **Fin**

---

**Théorème 2.24.** *L'algorithme 2.1 teste si LexBFS est un ordre d'élimination bisimplicial, et donc si  $G$  est un graphe biparti distance héréditaire, avec une complexité  $O(m + n)$ .*

*Démonstration.* Avant de démontrer la complexité, justifions l'algorithme.

### 1. Justification.

- (a) Si le graphe d'entrée est un biparti distance héréditaire, alors
  - i. Si  $m(x) \neq x_n$  (e.i.  $x_n \notin N^+(x)$ ), le sous-graphe induit par  $W$  à chaque itération est connexe (définition 2.9), la restriction de  $\sigma$  sur  $W$  est un LexBFS (lemme 2.23) et  $x = \min W$  est bisimplicial (théorème 2.19).
  - ii. Si  $x_n \in N^+(x)$ , alors  $W = B_x$  est une biclique, et donc  $W$  est un graphe biparti distance héréditaire, et l'algorithme s'arrête pour renvoyer vrai.
- (b) S'il existe une itération où la ligne 7 est vraie, c'est-à-dire, pour  $x = \min W$ ,  $N_W(y) \neq N_W(m(x))$ , où  $m(x), y$  sont deux voisins de  $x$  dans  $W$  (donc sont des voisins supérieurs dans  $G$ ), alors  $x$  n'est pas bisimplicial dans  $W$  (assertion 4, proposition 2.9). Ceci implique que la restriction de  $\sigma$  sur  $W$  n'est pas un ordre d'élimination bisimplicial, et donc elle n'est pas un LexBFS (théorème 2.19). Par conséquent,  $G$  n'est pas un graphe biparti distance héréditaire (lemme 2.23).

### 2. La complexité.

Le pire des cas correspond à celui où l'entrée est un graphe biparti distance héréditaire, et donc à chaque itération le plus petit sommet  $x = \min W$  est bisimplicial. Dans chaque itération de l'algorithme, on teste si  $x$  est bisimplicial pour le supprimer ensuite avec ses voisins, en ne laissant qu'un seul voisin. On supprime aussi les arêtes qui lui sont incidentes. Le sommet et les arêtes non supprimés dans une étape donnée seront utilisés et supprimés dans une itération ultérieure. Il en résulte que chaque sommet et chaque arête seront examinés au plus 2 fois. Par conséquent, la complexité temporelle de cet algorithme est en  $O(n + m)$ .

□

**Exemple 2.3.** Dans la figure 2.13, nous avons un graphe biparti d'ordre 9. On peut vérifier facilement que l'ordre des sommets est un LexBFS. Appliquons maintenant l'algorithme 2.1 pour voir s'il est biparti distance héréditaire.

On a

$$W = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9\}, x = x_1 \text{ et } N^+(x_1) = \{x_4, x_6, x_7\}, m(x) = x_7$$

(le choix de  $x_7$  est arbitraire, on aurait pu choisir  $x_4$  ou  $x_6$ , car ils sont des jumeaux (proposition 2.9, assertion 4)). Passons à l'itération 1

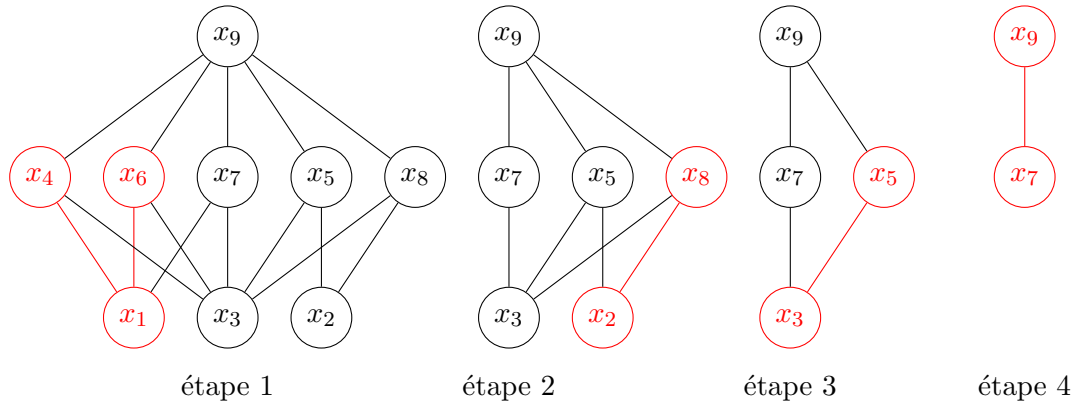


FIGURE 2.13 – Tester si l'ordre sur les sommets est un LexBFS

1.  $m(x) = x_7 \neq x_9$ ; pour  $y \in N^+(x_1) \setminus \{x_7\} = \{x_4, x_6\}$ ,  $N_W(y) = N_W(x_7) = \{x_1, x_3, x_9\}$ . Ceci implique que le sommet  $x = x_1$  est bisimplicial.

Posons

$$W = \{x_2, x_3, x_5, x_7, x_8, x_9\}, x = x_2 \text{ et } N^+(x_2) = N_W(x_2) = \{x_8, x_5\}, m(x) = x_5$$

.

et passons à l'itération 2.

2.  $m(x) = x_5 \neq x_9$ ; pour  $y \in N^+(x_2) \setminus \{x_5\} = \{x_8\}$ ,  $N_W(y) = N_W(x_5) = \{x_2, x_3, x_9\}$ . Ceci implique que le sommet  $x = x_2$  est bisimplicial.

Posons

$$W = \{x_3, x_5, x_7, x_9\}, x = x_3 \text{ et } N^+(x_3) = N_W(x_3) = \{x_5, x_7\}, m(x) = x_7$$

.

et passons à l'itération 3.

3.  $m(x) = x_7 \neq x_9$ ; pour  $y \in N^+(x_3) \setminus \{x_7\} = \{x_5\}$ ,  $N_W(y) = N_W(x_7) = \{x_3, x_9\}$ . Ceci implique que le sommet  $x = x_3$  est bisimplicial.

Posons

$$W = \{x_7, x_9\}, x = x_7 \text{ et } N_W(x_7) = \{x_5, x_7\}, m(x) = x_9$$

- .
- et passons à l'itération 4.
4.  $m(x) = x_9$ ,  $W = B_{x_7}$  est une biclique et l'algorithme s'arrête pour conclure que le graphe d'entrée est un graphe  $(6, 2)$ -chordal.

## Conclusion du chapitre 2

Nous avons rappelé dans ce chapitre la propriété très importante d'un graphe biparti distance héréditaire qui est l'existence d'un ordre d'élimination bisimplicial des sommets. Nous avons montré que cet ordre des sommets peut être calculé par le parcours LexBFS. Enfin, nous avons montré que, dans un graphe biparti distance héréditaire, la restriction d'un LexBFS sur tout sous-graphe obtenu par la suppression d'un sous-ensemble de sommets bien choisi est aussi un LexBFS; et grâce à cette dernière propriété, nous avons établi un algorithme linéaire pour la reconnaissance de cette classe de graphes.

## Chapitre 3

# Problèmes de biclique dans un Graphe Biparti Distance Héréditaire

Ce chapitre est dédié aux différents types de problèmes liés au calcul de bicliques dans la classe de graphes bipartis distance héréditaire. Nous commençons par établir un algorithme qui énumère toutes les bicliques maximales pour en déduire ensuite deux algorithmes : le premier calcule une biclique maximum pour les sommets et le second calcule une biclique maximum pour les arêtes. Dans la seconde partie, nous présentons un algorithme pour calculer une couverture minimum des arêtes en bicliques. Nous terminons par un algorithme calculant une couverture minimum des sommets en bicliques.

### 3.1 Biclique maximum

Une biclique dans un graphe biparti est un sous-graphe induit, noté  $B = (X_B, Y_B)$  tel que pour tout  $x \in X_B$  et pour tout  $y \in Y_B$ ,  $xy \in E$ . La résolution de nombreux problèmes, dans divers domaines comme e-commerce, bio-informatique, services web, optimisation de la fabrication [23], analyse de concept formel [36], analyse d'ADN en biologie [78], conjonction de clustering [70]... etc, se ramènent à l'étude des bicliques dans un graphe biparti. Il est clair qu'une biclique  $B = (X_B, Y_B)$  contient  $|X_B| + |Y_B|$  sommets et  $|X_B| \times |Y_B|$  arêtes. Ainsi, on distingue au moins deux variantes pour le problème de biclique maximum : **le problème de biclique sommet maximum** (the vertex maximum biclique problem) qui cherche une biclique avec un nombre de sommets aussi grand que possible, et **le problème de biclique arête maximum** (the edge maximum biclique problem) qui cherche une biclique avec un nombre maximum d'arêtes .

La figure 3.1 montre une biclique arête maximum dont les arêtes sont en rouge, et une biclique sommet maximum dont les arêtes sont en bleu. Une troisième variante pour ce même problème est **le problème de biclique maximum équilibré** (the maximum

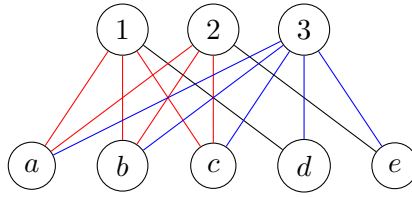


FIGURE 3.1 – Biclique sommet maximum et biclique arête maximum.

balanced biclique) qui consiste à trouver une biclique  $B = (X_B, Y_B)$  maximum vérifiant la contrainte  $|X_B| = |Y_B|$ .

Calculer une biclique maximum pour les sommets dans un graphe biparti est un problème polynomial [69]. Cependant, des algorithmes polynomiaux pour les deux variantes du même problème n'existent pas encore et ne pourront jamais exister, sauf si  $\mathcal{P} = \mathcal{NP}$ . En effet, la  $\mathcal{NP}$ -complétude du problème de la biclique arête-maximum et le problème de biclique équilibrée figurent respectivement dans [82] et [69].

Dans un graphe biparti  $G$ , pour calculer une biclique arête-maximum, il est nécessaire de générer toutes les bicliques maximales de  $G$  (d'ailleurs, c'est pour ça qu'il est dans  $\mathcal{NP}$ -complet). Mais, ce nombre peut être une fonction exponentielle en la taille de la donnée. En effet, *Prismer* a montré qu'un graphe biparti d'ordre  $n$  peut avoir jusqu'à  $2^{n/2}$  bicliques maximales [84]. Par ailleurs, le nombre de ces bicliques peut être majoré par un polynôme quand la donnée est un graphe particulier.

**Proposition 3.1.** [62] *Dans un graphe  $G = (V, E)$  biparti triangulé de taille  $|E| = m$ , le nombre de bicliques maximales est majoré par  $m$ .*

A partir de ce résultat, *Kloks* et *Kratsh* [62] ont montré que la taille d'une biclique arête-maximum dans cette classe peut être calculée en  $O(\min(m \log n, n^2))$ .

*Alexe et al.* ont montré que dans un graphe biparti  $G = (X, Y, E)$   $X$ -convexe, le nombre de bicliques maximales est en  $O(|Y|^2)$  [2]. Tandis que tous les algorithmes de calcul de la biclique maximum pour les arêtes se basent sur l'énumération de toutes les bicliques maximales, *Nussbaum et al.* [78] ont trouvé un algorithme de complexité  $O(n \log^3 n \log \log n)$  pour calculer une biclique arête-maximum dans un graphe convexe, et ceci après avoir trouvé une technique qui leur a permis d'éviter l'énumération exhaustive de toutes les bicliques maximales.

Ci-dessous, nous montrons que le nombre de bicliques maximales dans un graphe biparti distance héréditaire est majoré par le nombre de ses sommets. De plus, nous proposons Trois algorithmes gloutons pour cette classe de graphes : le premier génère toutes les bicliques maximales, le second calcule la taille d'une biclique maximum pour les sommets et le dernier calcule une biclique maximum pour les arêtes.

**Proposition 3.2.** *Dans un graphe biparti distance héréditaire  $G = (V, E)$  d'ordre  $|V| = n$ , le nombre de bicliques maximales est majoré par  $n$ .*

*Démonstration.* Soit  $G$  un graphe biparti distance héréditaire ordonné par lexBFS  $\sigma = (x_1, x_2, \dots, x_n)$ . Soit  $B$  une biclique maximale de  $G$  et soit  $x_i$  le plus petit sommet pour  $\sigma$  dans  $B$ , alors  $B$  est une biclique maximale dans  $G_i$ , le sous-graphe induit par les sommets  $x_i, x_{i+1}, \dots, x_n$ . Puisque  $x_i$  est bisimplicial dans  $G_i$ , alors d'après la proposition 2.9,  $x_i$  appartient à une unique biclique maximale  $B_{x_i}$ , donc  $B = B_{x_i}$ . Par conséquent, si  $b(G)$  désigne le nombre de bicliques maximales, alors  $b(G) \leq n$ .  $\square$

**Remarque 3.1.** Notons que, dans la proposition 3.2, si  $G$  n'est pas vide alors une biclique maximale contient au moins une arête et donc au moins deux sommets. Il en résulte que  $G$  contient au plus  $n - 1$  bicliques maximales.

La proposition 3.2 montre aussi que dans un graphe biparti distance héréditaire ordonné par LexBFS  $\sigma = (x_1, x_2, \dots, x_n)$ , les bicliques maximales sont de la forme  $B_{x_i}$  où  $x_i$  est un sommet bisimplicial dans  $G_i$ , le sous-graphe induit par les sommets  $x_i, x_{i+1}, \dots, x_n$ . Cependant, une biclique  $B_{x_i}$  maximale dans  $G_i$  n'est pas nécessairement maximale dans  $G$ . En effet, dans la figure 3.2, nous avons un graphe biparti distance héréditaire d'ordre 10 et avec seulement 3 bicliques maximales

$$B_{x_0} = (\{x_0, x_1, x_2, x_5, x_6, x_7, x_8\}, \{x_3, x_4\}); B_{x_2} = (\{x_2, x_3, x_4, x_9\}, \{x_5, x_6\})$$

et

$$B_{x_3} = (\{x_3, x_4, x_9\}, \{x_5, x_6, x_7, x_8\})$$

Donc, il est naturel de s'interroger sur la particularité des bicliques  $B_{x_i}$  qui sont aussi maximales dans  $G$ .

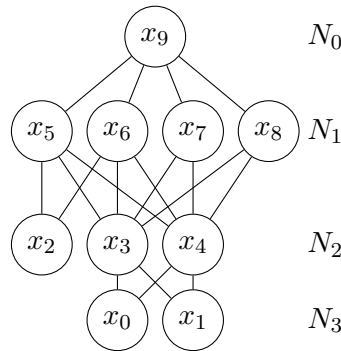


FIGURE 3.2 – Graphe biparti, ordonné par LexBFS, et ayant 3 bicliques maximales

Dans toute la suite de ce chapitre, nous supposons que  $G = (X, Y, E)$  est un graphe biparti distance héréditaire avec  $m, n$  sont respectivement le nombre d'arêtes et le nombre

de sommets ;  $\sigma = (v_1, v_2, \dots, v_n)$  est un ordre LexBFS. Pour tout  $i$ ,  $G_i$  est le sous-graphe induit par les sommets  $v_i, v_{i+1}, \dots, v_n$  et  $B_{v_i}$  est l'unique biclique maximale dans  $G_i$  contenant  $v_i$ .

**Remarque 3.2.** Notons que les voisins supérieurs de tout sommet  $v_i, i < n$ , dans  $G$  sont les mêmes que dans le sous-graphe  $G_i$ . Notons aussi

$$B_{v_i} = (N^+(v_i), N_{G_i}(u)) = (N_{G_i}(u), N^+(v_i))$$

où  $u \in N^+(v_i)$  et  $N_{G_i}(u)$  est l'ensemble des voisins de  $u$  dans le sous-graphe  $G_i$ .

Par exemple dans la figure 3.2,  $x_4$  est un sommet bisimplicial dans le sous-graphe  $G_4 = G[\{x_4, x_5, x_6, x_7, x_8, x_9\}]$  ;  
 $B_{x_4} = (\{x_4, x_9\}, \{x_5, x_6, x_7, x_8\})$  où  $N^+(x_4) = \{x_5, x_6, x_7, x_8\}$  et  $\{x_4, x_9\} = N_{G_4}(x_5) = N_{G_4}(x_6) = N_{G_4}(x_7) = N_{G_4}(x_8)$ .

Rappelons que les sommets de  $G$  sont partitionnés en  $k$  niveaux  $N_0, N_1, \dots, N_{k-1}$  où  $N_0 = \{v_n\}, N_1 = N(x_n) = N^-(v_n)$  ; d'une manière générale, pour  $0 < l < k - 1$ , le niveau  $N_l$  est constitué par les voisins supérieurs des éléments de  $N_{l-1}$ , c'est-à-dire  $N_l = \bigcup_{v \in N_{l-1}} N^+(v)$ .

**Définition 3.1.** Pour tout sommet  $v$ , posons

$$\pi(v) = d_G(v, v_n)$$

et pour toute biclique maximale  $B$ , posons

$$\mathcal{I}(B) = \max_{u, v \in B} |\pi(u) - \pi(v)|$$

En tenant compte de la définition 3.1, on tire la remarque suivante

- Remarque 3.3.**
1. Pour tout  $l = 0, \dots, k - 1$ ,  $N_l = \{v, \pi(v) = l\}$  ;
  2. Si  $v <_\sigma w$ , alors  $d_G(v, v_n) \geq d_G(w, v_n)$  et donc  $\pi(v) \geq \pi(w)$  ;
  3. Pour tout sommet  $v$ ,  $\mathcal{I}(B_v) \leq 2$  (voir l'observation 2).

**Observation 4.** Pour tout sommet  $v$

$$\mathcal{I}(B_v) = 2 \Leftrightarrow v_n \notin N^+(v).$$

*Démonstration.* Puisque  $v$  est le plus petit sommet dans  $B_v$ , alors  $\mathcal{I}(B_v) = 2$  implique qu'il existe  $w \in B_v$  tel que  $\pi(v) - \pi(w) = 2$ . Puisque, par définition  $\pi(w) \geq 0$ , alors  $\pi(v) = d(v, v_n) \geq 2$ , et donc  $v_n \notin N^+(v)$ .

Réciproquement, et par l'absurde, si  $v_n \in N^+(v)$ , on aurait  $\pi(v) = 1$ . Puisque  $v$  est le plus petit sommet pour  $\sigma$  dans  $B_v$ , alors on aurait  $\mathcal{I}(B_v) = 1$  ; ceci est une contradiction.  $\square$



**Remarque 3.4.** Il est facile de voir que d'après l'observation 4

$$\mathcal{I}(B_{v_i}) = 1 \Leftrightarrow v_n \in N^+(v_i)$$

Rappelons qu'une biclique  $B = (X_B, Y_B)$  n'est pas maximale si et seulement s'il existe un sommet  $v \notin B$  tel que  $v \sim_G X_B$  ou  $v \sim Y_B$ , d'où la définition suivante :

**Définition 3.2.** On dit qu'une biclique  $B = (X_B, Y_B)$  est maximale pour  $X_B$  (resp. maximale pour  $Y_B$ ) dans  $G$  s'il n'existe pas de biclique maximale  $B' = (X'_B, Y'_B)$  tel que  $X_B \subset X'_B$  (resp.  $Y_B \subset Y'_B$ ).

Il est clair que si  $B = (X_B, Y_B)$  est maximale dans  $G$ , alors elle est maximale pour  $X_B$  et maximale pour  $Y_B$ , et vice versa. De plus, s'il existe  $y \in Y_B$  (resp.  $x \in X_B$ ) tel que  $B = (N(y), Y_B)$  (resp.  $B = (X_B, N(x))$ ) est une biclique, alors elle est maximale pour  $N(y)$  (resp. pour  $N(x)$ ) dans  $G$ .

**Lemme 3.3.** Pour tout sommet  $v_i$ , si  $v_n \notin N^+(v_i)$ , alors la biclique  $B_{v_i}$  est maximale pour  $N^+(v_i)$  dans  $G$ .

*Démonstration.* Puisque  $v_n \notin N^+(v_i)$ , alors d'après l'observation 4

$$\mathcal{I}(B_{v_i}) = 2$$

donc il existe  $w \in B_{v_i}$  tels que

$$\pi(v_i) - \pi(w) = 2$$

Ceci implique qu'il existe  $l \geq 2$  tels que

$$\pi(v_i) = l \text{ et } \pi(w) = l - 2$$

Rappelons que  $B_{v_i} = (N^+(v_i), N_{G_i}(u)), u \in N^+(v_i)$ . Par l'absurde, supposons que la biclique  $B_{v_i}$  n'est pas maximale pour  $N^+(v_i)$ , donc il existe un sommet  $w' \in G$  tel  $B_{v_i}$  est incluse dans la biclique  $B = (N^+(v_i) \cup \{w'\}, N_{G_i}(u))$ . Ainsi,  $w' \in N^-(v_i)$ , donc  $w' \in N_{l+1}$ , c'est-à-dire  $\pi(w') = l + 1$ . Par conséquent, on aurait

$$\pi(w') - \pi(w) = l + 1 - (l - 2) = 3$$

mais ceci contredit le fait que  $w$  et  $w'$  appartient à une même biclique (voir la remarque 3.3).  $\square$

Par contre, si pour un sommet  $v_i$ ,  $v_n \in N^+(v_i)$ , la biclique  $B_{v_i}$  peut ne pas être maximale pour  $N^+(v_i)$  : dans la figure 3.2, le sommet source est le sommet  $x_9$  et on a, par exemple  $x_9 \in N^+(x_5) = \{x_9\}$ , et  $B_{x_5}$  n'est pas maximale pour  $N^+(x_5)$ , car la biclique  $B_{x_3} = (\{x_3, x_4, x_9\}, \{x_5, x_6, x_7, x_8\})$  contient  $B_{x_5}$  ; plus précisément,

$$N^+(x_5) = \{x_9\} \subset N^+(x_3) = \{x_3, x_4, x_9\}$$

**Remarque 3.5.** Notons que si  $v_i <_\sigma v_j$ , alors

1.  $N^+(v_i) \cap N^+(v_j) = \emptyset$  ou
2.  $N^+(v_i) \cap N^+(v_j) \neq \emptyset$ , et donc  $N^+(v_i) \subsetneq N^+(v_j)$  ou  $N^+(v_i) = N^+(v_j)$  (d'après le lemme 2.18).

**Lemme 3.4.** *Soit  $v_i$  un sommet tel que  $v_n \notin N^+(v_i)$ . Alors les deux conditions suivantes sont équivalentes*

1. La biclique  $B_{v_i} = (N_{G_i}(w), N^+(v_i))$ ,  $w \in N^+(v_i)$  est maximale dans  $G$  ;
2.  $\forall i_0 < i, N^+(v_{i_0}) \neq N^+(v_i)$ .

*Démonstration.* S'il existe  $i_0 < i$  tel que  $N^+(v_{i_0}) = N^+(v_i)$ , alors  $B = (N^+(v_i), N_{G_i}(w) \cup \{v_{i_0}\})$  est une biclique contenant strictement la biclique  $B_{v_i}$ . Ceci est absurde.

Inversement, si  $\forall i_0 < i, N^+(v_{i_0}) \neq N^+(v_i)$ . Compte tenu de la remarque 3.5, ceci implique que  $v_{i_0} \approx N^+(v_i)$ . On en déduit que  $B_{v_i}$  est maximale pour  $N_{G_i}(w), w \in N^+(v_i)$ . Par ailleurs,  $B_{v_i}$  est maximale pour  $N^+(v_i)$ , car  $v_n \notin N^+(v_i)$  (voir le lemme 3.3), par conséquent, elle est maximale pour  $G$ .  $\square$

**Observation 5.** *Soit  $v_i$  un sommet dans le niveau  $N_2$ . Alors*

$$N^+(v_i) \neq N_1 \Rightarrow \forall i_0 < i, N^+(v_{i_0}) \neq N_1$$

*Démonstration.* Soit  $i_0 < i$  tel que  $N^+(v_{i_0}) = N_1$ . Alors

$$v_{i_0} <_{\sigma} v_i, v_{i_0} \in N_2 \text{ et } N^+(v_{i_0}) \cap N^+(v_i) \neq \emptyset.$$

Il en résulte, d'après le lemme 2.18, que  $N^+(v_{i_0}) \subseteq N^+(v_i)$ , et donc  $N^+(v_i) = N_1$ .  $\square$

**Lemme 3.5.** *Soit  $v_i$  un sommet tel que  $v_n \in N^+(v_i)$ . Alors les deux conditions suivantes sont équivalentes*

1.  $B_{v_i}$  maximale dans  $G$  ;
2.  $v_i = \min N_1$  et  $N^+(v_{i-1}) \neq N_1 = N^-(v_n)$

*Démonstration.* Soit  $B_{v_i}$  une biclique maximale. Puisque  $v_n \in N^+(v_i)$ , alors  $v_i \in N_1$ , donc  $B_{v_i}$  est incluse dans la biclique  $(N_0, N_1)$ ; et puisque  $B_{v_i}$  maximale, alors  $B_{v_i} = (N_0, N_1)$ . On en déduit que  $v_i = \min N_1$  et  $\forall i_0 < i, N^+(v_{i_0}) \neq N_1 = N^-(v_n)$ .

Réciproquement, si  $v_i = \min N_1$ , alors  $B_{v_i} = (N_1, N_0) = (N^-(v_n), \{v_n\}) = (N(v_n), \{v_n\})$ , donc  $B_{v_i}$  est maximale pour  $N_1$ ; et si  $N^+(v_{i-1}) \neq N_1$ , alors, d'après l'observation 5,  $\forall j < i, v_j \approx N_1$ , et donc  $B_{v_i}$  est maximale pour  $N_0$ . Par conséquent,  $B_{v_i}$  est maximale dans  $G$ .  $\square$

Les lemmes 3.4 et 3.5 aboutissent à l'algorithme ci-dessous qui énumère toutes les bicliques maximales d'un graphe biparti distance héréditaire ordonné par LexBFS avec une complexité linéaire.

**Algorithme 3.1.** Énumération de toutes les Bicliques maximales

**Entrée :** Un graphe Biparti distance héréditaire  $G = (V, E)$  ordonné LexBFS  $\sigma = \{v_1, v_2, \dots, v_n\}$ .

**Sortie :**  $\theta$ , le nombre de bicliques maximales dans  $G$ .

**début**

1.  $i = 1 ; \theta = 1$
2. Tant que  $v_n \notin N^+(v_{i+1})$  faire
3.     Si  $N^+(v_{i+1}) = N^+(v_i)$  , alors
4.          $i \leftarrow i + 1$
5.     Sinon
6.          $i \leftarrow i + 1, \theta = \theta + 1$ .
7.     Finsi
8. Fin tant que
9. Si  $v_n \in N^+(v_{i+1})$  faire
10. si  $|N^+(v_i)| \neq |N^-(v_n)|$  alors
11.      $\theta = \theta + 1$
12. fin si
13. retourner  $\theta$

**fin**

---

**Théorème 3.6.** L'algorithme 3.1 énumère toutes les bicliques maximale de  $G$  en  $O(m)$ .

*Démonstration.* 1. **Correction**

Selon le lemme 3.4, dans la ligne 6,  $\theta$  est incrémenté si et seulement si une nouvelle biclique maximale  $B$ , vérifiant  $\mathcal{I}(B) = 2$ , est détectée. Selon le lemme 3.5,  $\theta$  est incrémenté dans la ligne 11 si et seulement si la biclique  $B = (N_1, N_0)$  est maximale.

2. **Complexité**

Il est claire que la complexité de cet algorithme est dominée par celle de la ligne 3, où pour chaque sommet  $v_i$ , on calcule  $N^+(v_i)$  au plus une seule fois. Par conséquent, à la fin de l'algorithme, le nombre d'opérations effectuées est en  $O(\sum_1^n |N^+(v_i)|) = O(m)$ . □

**Exemple 3.1.** Montrons sur la figure 4.2 comment appliquer l'algorithme 3.1 pour énumérer toutes les bicliques maximales de  $G$ . On peut vérifier que les sommets sont ordonnés par LexBFS et que cet ordre est un ordre d'élimination bisimplicial, donc  $G$  est un graphe biparti distance héréditaire.

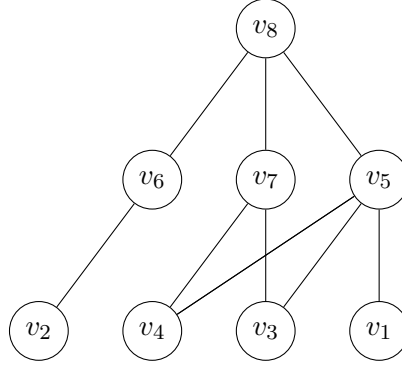


FIGURE 3.3 – Énumération des Bicliques maximales.

1.  $i = 1, \theta = 1, B_{v_1} = (\{v_1, v_3, v_4, v_8\}, \{v_5\})$  ;
2.  $i = 2$   
 $v_8 \notin N^+(v_2), N^+(v_1) = \{v_5\} \neq N^+(v_2) = \{v_6\}$ , // on incrémente  $\theta$  et  $i$   
 $\Rightarrow \theta = 2, B_{v_2} = (\{v_2, v_8\}, \{v_6\})$ ,  $i \leftarrow 3$ .
3.  $i = 3$ ,  
 $v_8 \notin N^+(v_3), N^+(v_2) = \{v_6\} \neq N^+(v_3) = \{v_5, v_7\}$ ,  
 $\Rightarrow \theta = 3, B_{v_3} = (\{v_3, v_4, v_8\}, \{v_5, v_7\})$ ,  $i \leftarrow 4$ .
4.  $i = 4$ ,  
 $v_8 \notin N^+(v_4), N^+(v_3) = \{v_5, v_7\} = N^+(v_4)$ , // puisque il y a égalité, on incrémente  $i$  seulement.  
 $\Rightarrow i \leftarrow 5$ .
5.  $i = 5$ ,  
 $v_8 \in N^+(v_5), |N^+(v_4)| = 2 \neq |N^-(v_8)| = 3, \Rightarrow \theta = 4, B_{v_5} = (\{v_5, v_6, v_7\}, \{v_8\})$ .

Ainsi, ce graphe possède 4 bicliques maximales.

Pour calculer une biclique sommet-maximum avec une complexité linéaire dans un graphe biparti distance héréditaire, il suffit de modifier légèrement l'algorithme 3.1 comme suit :

**Algorithme 3.2.** *Biclique Sommet-Maximum (Maximum Vertex Biclique)*

**Entrée :** Un graphe biparti distance héréditaire connexe  $G = (V, E)$  et  $\text{lexBFS} = \sigma = \{v_1, v_2, \dots, v_n\}$ .

**Sortie :**  $\text{max}$  = la taille d'une biclique sommet-maximum.

**debut**

1.  $i = 1$  ;  $\text{max} = |N^+(v_1)| + |N_{G_i}(m(v_i))|$ ,  $m(v_i) \in N^+(v_i)$  ; // Ici,  $G_i = G$
2. Tant que  $v_n \notin N^+(v_{i+1})$  faire
3. Si  $N^+(v_{i+1}) = N^+(v_i)$ , alors

4.  $i \leftarrow i + 1$
  5. *Sinon*
  6. calculer  $|N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$  et  $i \leftarrow i + 1$ .
  7. *Fin si*
  8. *Si*  $\max < |N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$ , *alors*
  9.  $\max \leftarrow |N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$ .
  10. *Fin si*
  11. *Fin tant que*
  12. *Si*  $v_n \in N^+(v_{i+1})$  *faire*
  13. *Si*  $|N^+(v_i)| \neq |N^-(v_n)|$  *alors*
  14. Calculer  $|N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$ .
  15. *Finsi*
  16. *Si*  $\max < |N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$ , *alors*
  17.  $\max \leftarrow |N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$ .
  18. *Fin si*
  19. *Fin si*
  20. *Retourner*  $\max$
- fin**

---

**Remarque 3.6.** Pour calculer une biclique arête-maximum, il suffit de remplacer dans l'algorithme 3.2 la somme

$$|N^+(v_{i+1})| + |N_{G_{i+1}}(m(v_{i+1}))|$$

par le produit

$$|N^+(v_{i+1})| \times |N_{G_{i+1}}(m(v_{i+1}))|$$

**Exemple 3.2.** Montrons sur la figure 4.2 comment appliquer l'algorithme 3.2 pour calculer le nombre de sommets dans une biclique sommet-maximum de  $G$ .

1.  $i = 1$ ,  $\max = |N^+(v_1)| + |N_G(m(v_1))| = 1 + 4 = 5$ , avec  $m(v_1) = v_5$  ;
2.  $i = 2$   
 $v_8 \notin N^+(v_2)$ ,  $N^+(v_1) = \{v_5\} \neq N^+(v_2) = \{v_6\}$ ,  
 $\Rightarrow \max = \max(5, |N^+(v_2)| + |N_{G_2}(v_6)|) = \max(5, 2) = 5$ , avec  $m(v_2) = v_6$ .
3.  $i = 3$ ,  
 $v_8 \notin N^+(v_3)$ ,  $N^+(v_2) = \{v_6\} \neq N^+(v_3) = \{v_5, v_7\}$ ,  
 $\Rightarrow \max = \max(5, |N^+(v_3)| + |N_{G_3}(v_5)|) = \max(5, 5) = 5$ , avec  $m(v_3) = v_5$ , //  
Ici, on aurait pu choisir  $m(v_3) = v_7$  et ça ne change rien, car  $v_5$  et  $v_6$  ont les mêmes voisins dans  $G_3$ .

4.  $i = 4$ ,  
 $v_8 \notin N^+(v_4)$ ,  $N^+(v_3) = \{v_5, v_7\} = N^+(v_4)$ , // puisque il y a égalité, on ne fait rien.
5.  $i = 5$ ,  
 $v_8 \in N^+(v_5)$ ,  $|N^+(v_4)| = 2 \neq |N^-(v_8)| = 3$ ,  
 $\Rightarrow \max = \max(5, |N^+(v_5)| + |N_{G_5}(v_8)|) = \max(5, 4) = 5$ , avec  $m(v_5) = v_8$ .

Par conséquent, le nombre de sommets dans une biclique sommet-maximum est 5.

### 3.2 Couverture des arêtes par des bicliques

Étant donné un graphe biparti  $G = (X, Y, E)$ , le problème de **couverture des arêtes** de  $G$  par des bicliques (Biclique Edge Cover problem) consiste à trouver une famille de bicliques

$$\mathcal{F} = \{B_1 = (X_1, Y_1, E_1), B_2 = (X_2, Y_2, E_2), \dots, B_k = (X_k, Y_k, E_k)\}$$

de cardinalité minimale telle que

$$E = \cup_{1 \leq i \leq k} E_i$$

c'est-à-dire toute arête de  $G$  appartient à au moins une biclique de  $\mathcal{F}$ ; le problème de **partition des arêtes** de  $G$  en bicliques (Biclique Edge Partition Problem) est un problème de couverture avec la condition supplémentaire que toute arête de  $G$  doit être dans une unique biclique de  $\mathcal{F}$ , c'est-à-dire la famille  $\{E_1, E_2, \dots, E_k\}$  doit être une partition de  $E$ . Les domaines d'application pour les problèmes de couverture et de partition des arêtes en bicliques sont nombreux, on les trouve dans la bioinformatique [74, 75], dans la sécurité informatique [30], dans le carrelage des bases de données [38], dans les automates finis [47] et dans les dessins des graphes [32].

Notons par  $bc(G)$  (resp.  $bp(G)$ ) le plus petit nombre de bicliques qui couvrent (resp. partitionnent) les arêtes de  $G$ . Les problèmes de décision associés au calcul des invariants  $bc(G)$  et  $bp(G)$  peuvent être formulés de cette façon :

---

#### PROBLÈME DE COUVERTURE DES ARÊTES EN BICLIQUES

---

*Instance* : Un graphe biparti  $G$  et un entier positif  $k$ .

*Question* : Est ce que  $bc(G) \leq k$  ?

---

#### PROBLÈME DE PARTITION DES ARÊTES EN BICLIQUES

---

*Instance* : Un graphe biparti  $G$  et un entier positif  $k$ .

*Question* : Est ce que  $bp(G) \leq k$  ?

La  $\mathcal{NP}$ -complétude du paramètre  $bc(G)$  pour un graphe biparti a été démontré par Orlin en 1977 [79], et il a conjecturé que le calcul du paramètre  $bp(G)$  est aussi  $\mathcal{NP}$ -complet. En 1993 Jiang et Ravikumar [58] ont confirmé la  $\mathcal{NP}$ -complétude du paramètre  $bp(G)$ . En 1996, Müller a montré que la restriction du calcul de  $pc(G)$  sur un graphe biparti triangulé reste aussi  $\mathcal{NP}$ -complet [73]. Amilhastre et al. [3] ont montré que pour un graphe biparti  $G$  sans domino, les paramètres  $bc(G)$  et  $bp(G)$  sont égaux et on peut les calculer avec une complexité  $O(mn)$ , où  $m$  et  $n$  sont respectivement le nombre d'arêtes et de sommets dans  $G$ . Un algorithme polynomial a été proposé par Müller dans [73] pour calculer le paramètre  $bp(G)$  pour la classe de graphes bipartis distance héréditaire, une sous-classe des graphes sans domino. Dans cette section, en utilisant le parcours LexBFS, nous proposons un algorithme linéaire pour calculer les paramètres  $bp(G)$  et  $bc(G)$  pour un graphe biparti distance héréditaire.

En générale, pour tout graphe biparti  $G = (X, Y, E)$ , nous avons :

$$bc(G) \leq bp(G) \leq \min\{|X|, |Y|\}$$

En effet, la première inégalité vient de fait que toute partition des arêtes en bicliques est aussi une couverture en bicliques des mêmes arêtes, la seconde inégalité vient de fait que les deux familles  $\mathcal{F}_1 = \{(x_i, N(x_i)), x_i \in X\}$  et  $\mathcal{F}_2 = \{(y_j, N(y_j)), y_j \in Y\}$  sont des partitions des arêtes de  $G$  en bicliques. Notons que ces inégalités peuvent être strictes comme est le cas dans la figure 3.4

$$bc(G) = |\{(ab, xy), (ac, xz)\}| = 2$$

et

$$bp(G) = |\{(ab, xy), (ac, z), (x, c)\}| = 3.$$

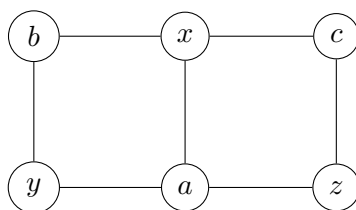


FIGURE 3.4 – Graphe vérifiant  $bc(G) = 2 < bp(G) = 3$ .

Pour la seconde inégalité, il suffit de considérer un graphe biparti complet qui n'est pas une étoile.

Dans cette section, sauf indication du contraire,  $G = (V, E)$  est un graphe biparti distance héréditaire ordonné par LexBFS  $\sigma = (v_1, v_2, \dots, v_n)$ . Rappelons que si  $G$  est connexe non complet, alors le sous-graphe obtenu par la suppression des arêtes de  $B_{v_1}$

est constitué par une unique composante connexe non triviale et qui est un sous-graphe induit (voir les lemmes 2.13, 2.14). Le lemme suivant montre que la restriction de l'ordre lexBFS sur cette composante connexe est un LexBFS.

**Lemme 3.7.** *La restriction de  $\sigma$  sur la composante connexe obtenue par la suppression des arêtes de la biclique  $B_{v_1}$  est un LexBFS.*

*Démonstration.* Soit  $A$  l'ensemble des sommets de cette composante connexe. Notons que, d'après le lemme 2.13,

$$A = V - (N[v_1] \cup \{w, N(w) = N(v_1)\})$$

Il s'agit de montrer que pour  $u, v \in A$ ,

$$u <_{\sigma} v \Rightarrow \forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) \cap A : u' \leq_{\sigma} v' \quad (3.1)$$

Soient  $N_0, N_1, \dots, N_k$  les niveaux de  $G$  définis par  $\sigma$  et soient  $u, v \in A$  tels que  $u <_{\sigma} v$ . Selon que le sommet  $v \in N_k$  ou  $v \notin N_k$ , nous avons deux cas

1. Si  $v \notin N_k$ , alors  $N^+(v) \not\subseteq N_{k-1}$  et donc  $N^+(v) \cap A = N^+(v)$ . Par ailleurs et puisque  $\sigma$  est un LexBFS sur  $V$ , on a

$$\forall u' \in N^+(u), \exists v' \in N^+(v) \text{ tels que } u' \leq_{\sigma} v'$$

et donc

$$\forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) = N^+(v) \cap A : u' \leq_{\sigma} v'$$

Par conséquent, l'équation 3.1 est vraie.

2. Si  $v \in N_k$ , alors même  $u \in N_k$  car  $u < v$ ; et selon que  $N^+(u) \cap N^+(v) = \emptyset$  ou  $N^+(u) \cap N^+(v) \neq \emptyset$ , nous avons deux cas
  - (a) Si  $N^+(u) \cap N^+(v) \neq \emptyset$ , et puisque  $G$  est un graphe biparti distance héréditaire, alors  $N^+(u) \subseteq N^+(v)$  (voir le lemme 2.18), et ceci implique  $N^+(u) \cap A \subseteq N^+(v) \cap A$ .  $\sigma$  étant un LexBFS sur  $V$ , donc  $\forall u' \in N^+(u), \exists v' \in N^+(v)$  tels que  $u' \leq_{\sigma} v'$ , et donc  $\forall u' \in N^+(u) \cap A, \exists v' \in N^+(v) \cap A : u' \leq_{\sigma} v'$ . Par conséquent, l'équation 3.1 est vraie.
  - (b) Si  $N^+(u) \cap N^+(v) = \emptyset$ , alors, d'après le lemme 1.3,

$$\exists v' \in N^+(v), \forall u' \in N^+(u), u' <_{\sigma} v' \quad (3.2)$$

Notons que  $v' \notin N^+[v_1]$ , sinon on aurait  $u <_{\sigma} v_1$  et ceci impossible. Ainsi,  $v' \in A$ , et donc  $v' \in N^+(v) \cap A$ . Ainsi, l'équation 3.2 implique

$$\exists v' \in N^+(v) \cap A, \forall u' \in N^+(u), u' <_{\sigma} v' \quad (3.3)$$

qui implique

$$\exists v' \in N^+(v) \cap A, \forall u' \in N^+(u) \cap A, u' <_{\sigma} v' \quad (3.4)$$

D'où, l'équation 3.1 est vraie.



Par conséquent, la restriction de  $\sigma$  sur  $A$  est un lexBFS.  $\square$

**Définition 3.3.** Soient  $H_1, H_2, \dots, H_k$  une séquence de graphes définis comme suite :

1.  $H_1 = G$ ;
2. Pour  $i = 2, \dots, k$ ,  $H_i$  est la composante connexe non triviale obtenue par la suppression des arêtes de la biclique maximale contenant le plus petit sommet pour  $\sigma$  dans  $H_{i-1}$ .
3.  $H_k$  est une biclique.

**Remarque 3.7.** Notons que dans la définition 3.3, le plus petit sommet pour  $\sigma$  dans  $H_i$  est bisimplicial et  $H_{i+1}$  est la composante connexe du graphe obtenu de  $H_i$  par la suppression des arêtes de la biclique qui contient ce sommet.

**Exemple 3.3.** La figure 3.5 montre la séquence de graphes calculée selon la définition 3.3 à partir du graphe  $G$ , qui est un graphe biparti distance héréditaire ordonné par LexBFS. Ici,  $k = 3$ , et comme il est indiqué dans la définition,  $H_3 = B_{v_8}$  est une biclique.

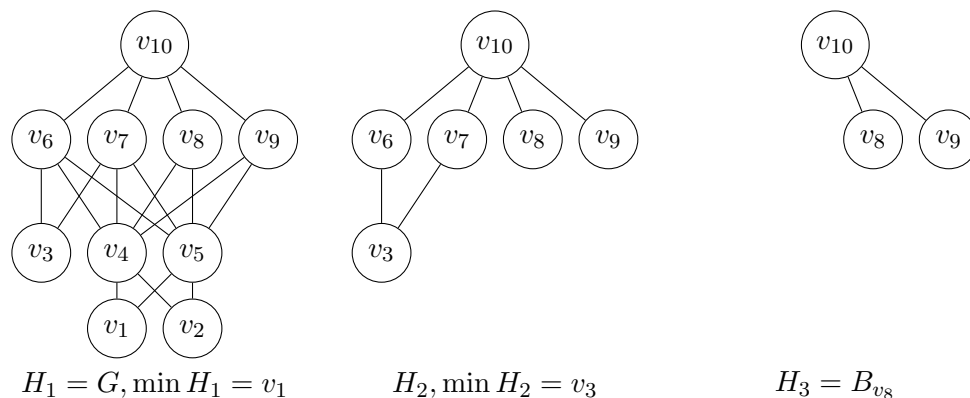


FIGURE 3.5 – La séquence de graphes  $H_1, H_2, H_3$  obtenus selon la définition 3.3

**Définition 3.4.** Une paire d'arêtes  $e_1 = u_1v_1$ ,  $e_2 = u_2v_2$  d'un graphe biparti  $G$  sont dites dépendantes dans  $G$  si  $e_1 \cap e_2 \neq \emptyset$  ou  $G_{\{u_1, u_2, v_1, v_2\}}$  est un cycle de longueur 4 dans  $G$ ; sinon elles sont dites indépendantes

Il est facile de voir que les arêtes d'une même biclique sont deux à deux dépendantes et les arêtes deux à deux indépendantes appartiennent à une même biclique.

**Lemme 3.8.** Soient  $H_1, H_2, \dots, H_k$  une séquence de graphes définie selon la définition 3.3. Pour  $i, 1 < i \leq k$ , soient  $e_1 = xy$ ,  $e_2 = x'y'$  deux arêtes dans  $H_i$ . Alors les deux conditions suivantes sont équivalentes

1.  $e_1$  et  $e_2$  sont indépendantes dans  $G$ ;
2.  $e_1$  et  $e_2$  sont indépendantes dans  $H_i$ .

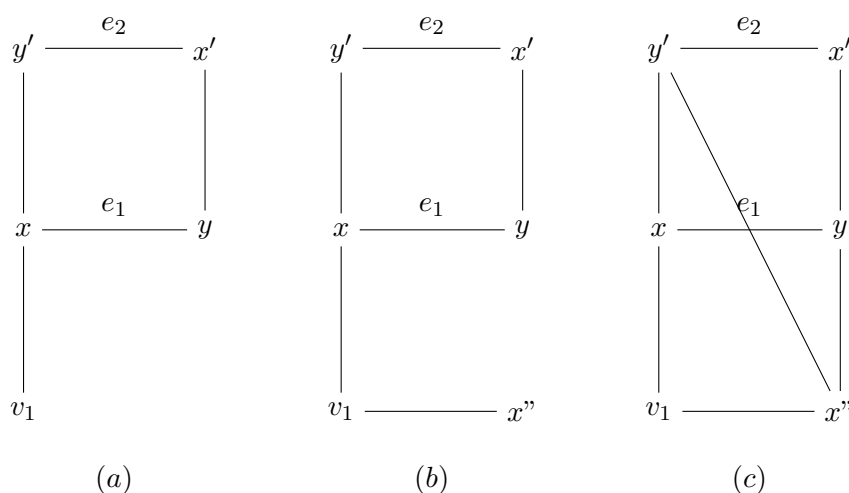


FIGURE 3.6 – Indépendance des arêtes est une propriété héréditaire

*Démonstration.* Puisque  $H_i$  est un sous-graphe induit de  $G$ , pour tout  $i$ , alors deux arêtes indépendantes dans  $G$  sont aussi indépendantes dans  $H_i$ , pour tout  $i = 2, \dots, k$ .

Pour démontrer la réciproque, on utilise la récurrence. Pour  $i = 2$ , soient deux arêtes  $e_1 = xy$  et  $e_2 = x'y'$  indépendantes dans  $H_2$ . Par l'absurde, supposons qu'elles sont dépendantes dans  $G$ . Alors ces deux arêtes appartiennent à un  $C_4$  dans  $G$ , et sans perte de généralité, on suppose que l'arête  $xy'$  appartient à  $G - H_2$ . Plus précisément  $xy' \in B_{v_1}$ . Par définition de la biclique  $B_{v_1}$ , on aurait  $x \in N^+(v_1)$  ou  $y' \in N^+(v_1)$ . Sans perte de généralité, on suppose que  $x \in N^+(v_1)$ , voir la figure 3.8(a). Selon que  $v_1$  est pendant ou non, nous aurons deux cas

1.  $v_1$  est un sommet pendant comme dans la même figure 3.8(a). Ainsi, on aurait  $e_1 \in B_{v_1}$ , et ceci est une contradiction.
2.  $v_1$  n'est pas pendant, et donc il existe  $x'' \in N^+(v_1)$  comme dans la figure (b). Ainsi, on aurait, d'après l'assertion 4 de la proposition 2.9,  $N^+(x) = N^+(x'')$  comme dans la figure 3.8 (c). Ainsi, on aurait  $e_1 \in B_{v_1}$ , et ceci est une contradiction.

Par conséquent, les deux arêtes sont aussi indépendantes dans  $G$ .

Supposons maintenant que pour tout  $i < k$  la propriété est vraie, c'est-à-dire si  $e_1, e_2$  sont indépendantes dans  $H_i$ , alors elles le sont dans  $G$ . Soient  $e_1, e_2$  deux arêtes indépendantes dans  $H_{i+1}$ . Alors elles sont indépendantes dans  $H_i$ , et d'après l'hypothèse de récurrence, elles sont aussi indépendantes dans  $G$ .  $\square$

**corollaire 3.9.** Soient  $H_1, H_2, \dots, H_k$  une séquence de graphes définie selon la définition 3.3, et  $e_1, e_2, \dots, e_k$  des arêtes telles que, pour  $i = 1, \dots, k$ ,  $e_i$  incidente à  $w_i$ , où  $w_i$  est le plus petit sommet dans  $H_i$  (et qui bien sûr est bisimplicial). Alors, les arêtes  $e_1, e_2, \dots, e_k$  sont deux à deux indépendantes dans  $G$ .

*Démonstration.* Soient  $i, j$  tels que  $1 \leq i < j \leq k$ . Rappelons que  $w_i$  est un sommet bisimplicial et  $e_i$  est une arête bisimpliciale dans  $H_i$ , donc  $B_{w_i} = B_{e_i}$  est l'unique biclique maximale dans  $H_i$  contenant  $w_i$  et  $e_i$ . On en déduit, par définition, que  $e_i$  est indépendante avec toutes les arêtes qui n'appartiennent pas à  $B_{w_i}$  dans  $H_i$ . En particulier,  $e_i$  et  $e_j$  sont indépendantes dans  $H_i$ ; et d'après le lemme 3.8,  $e_i$  et  $e_j$  sont indépendantes dans  $G$ .  $\square$

Le théorème suivant montre comment calculer une partition minimale des arêtes en biclique.

**Théorème 3.10.** *Sous l'hypothèse de la définition 3.3*

$$\mathcal{P} = \{B_{w_1}, B_{w_2}, \dots, B_{w_k}\} \quad (3.5)$$

est une partition minimum des arêtes de  $G$  en bicliques, où pour  $i = 1, \dots, k$ ,  $B_{w_i} = (N_{H_i}(w_i), N_{H_i}(u))$ ,  $u \in N_{H_i}(w_i)$  est la biclique maximale dans  $H_i$  contenant le plus petit sommet bisimplicial  $w_i$ .

*Démonstration.* Il s'agit de montrer que

$$k = bp(G)$$

Par construction,  $\mathcal{F}$  est une partition des arête de  $G$  en bicliques, donc

$$k \geq bp(G)$$

D'autre part, pour  $i = 1$  à  $k$ , soit  $e_i$  une arête incidente à  $w_i$  dans  $H_i$ . D'après le corolaire 3.9, les arêtes  $e_1, e_2, \dots, e_k$  sont deux à deux indépendantes, donc il faut au moins  $k$  bicliques pour couvrir ces arêtes, et donc

$$k \leq bp(G)$$

Par conséquent,

$$k = bp(G)$$

$\square$

**Algorithme 3.3.** *PARTITION MINIMUM DES ARÊTES EN BICLIQUES*

**Entrée :** Un graphe biparti distance héréditaire connexe  $G = (V, E)$  et un ordre  $lexBFS$   $\sigma = (v_1, v_2, \dots, v_n)$ .

**Sortie :** La valeur du paramètre  $bp(G)$ .

1. **Début**
2.  $k = 0$ ;  $v = v_1 = \min_{\sigma} W$ ;  $W = V$ ;
3. **Tant que**  $W \neq \emptyset$  **faire**
4. **Si**  $v$  est isolé **alors**

5.  $W = W - \{v\}$  ;
6. **Si non**  $W = W - N^+[v]$  ;  $v = \min_{\sigma} W$  ;  $k = k + 1$  ;
7. **Fin si** ;
8. **Fin tant que**.
9. Afficher  $k$
10. **Fin**

**Théorème 3.11.** *L'algorithme 3.3 calcule le paramètre  $bp(G)$  d'un graphe biparti distance héréditaire avec une complexité temporelle en  $O(n)$ .*

*Démonstration.* 1. **La correction**

Rappelons que si  $v$  est un sommet bisimplicial dans le graphe  $G$ , alors  $B_v = (N(v), N(u)) = (N^+(v), N(u))$  où  $u$  est l'un des voisins de  $v$ , donc la suppression des sommet  $N[v]$  entraîne la suppression de toutes les arêtes de  $B_v$ . Ainsi, à chaque itération de l'algorithme 3.3 et en tenant compte du lemme 2.13, le sous-graphe induit par  $W$  est constitué par une seule composante non triviale et éventuellement par des sommets isolés qui sont les jumeaux du sommet bisimplicial supprimé. Dans tous les cas,  $v = \min W$  est un sommet bisimplicial et la restriction de  $\sigma$  sur la composante non triviale est un LexBFS (lemme 3.7). Par ailleurs le paramètre  $k$  est incrémenté que si  $v = \min W$  n'est pas isolé. Par conséquent, l'algorithme va s'arrêter quand  $W = \emptyset$ , et d'après le théorème 3.10,  $k = bp(G)$ , d'où la correction de l'algorithme.

2. **La complexité**

Remarquons que la complexité est dominée par les lignes 5 et 6 où chaque sommet est supprimé une seule fois, soit dans la ligne 5 s'il est isolé, soit dans la ligne 6 si non. Puisque la suppression d'un sommet peut se faire en  $O(1)$ , alors la suppression de  $n$  sommets peut se faire en  $O(n)$ . □

**Exemple 3.4.** On peut vérifier facilement que le graphe  $G$  de la figure 3.7 est un graphe biparti distance héréditaire et que l'ordre  $\sigma = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12})$  est un LexBFS. Nous montrons ci-dessous les différentes étapes correspondantes aux itérations de l'algorithme 3.3.

1. **étape 0** :  $W_0 = V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}$ ,  $v = v_1, k = 0$ .

2. **étape 1** :

$$W_0 \neq \emptyset, v_1 = \min W_0, N^+[v_1] = \{v_1, v_6\}$$

$$W_1 = V - N^+[v_1] = \{v_2, v_3, v_4, v_5, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\}, k = 1$$

on aura le graphe de la figure 3.7 (b).

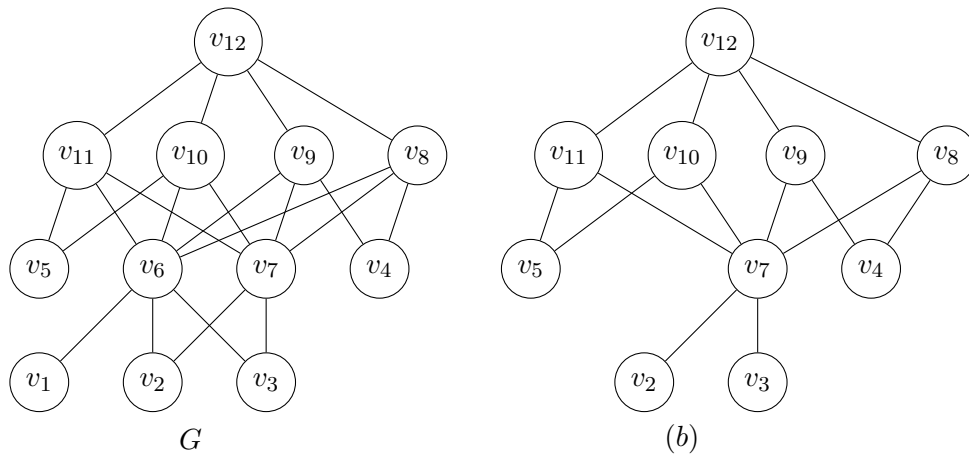


FIGURE 3.7 – Partition des arêtes en bicliques

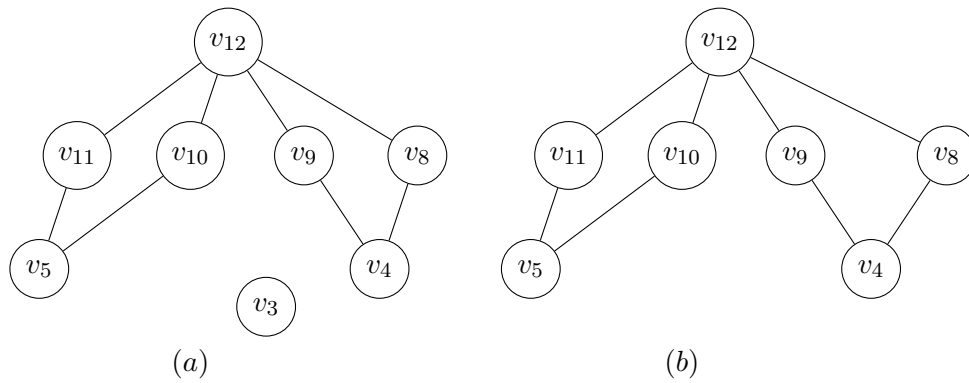


FIGURE 3.8 – Partition des arêtes en bicliques (suite)

3. **étape 2.**

$$W_1 \neq \emptyset, v_2 = \min W_1, N^+[v_2] = \{v_2, v_7\}$$

$$W_2 = W_1 - N^+[v_2] = \{v_3, v_4, v_5, v_8, v_9, v_{10}, v_{11}, v_{12}\}, k = 2$$

on aura le graphe de la figure 3.8 (a).

4. **étape 3.**

$$W_2 \neq \emptyset, v_3 = \min W_2, N^+[v_2] = \{\}$$

$$W_3 = W_2 - N^+[v_3] = \{v_4, v_5, v_8, v_9, v_{10}, v_{11}, v_{12}\}, k = 2$$

Le sommet  $v_3$  est isolé,  $k$  n'est pas incrémenté et on aura le graphe de la figure 3.8 (b).

5. **étape 4 :**

$$W_3 \neq \emptyset, v_4 = \min W_3, N^+[v_4] = \{v_8, v_9\}$$

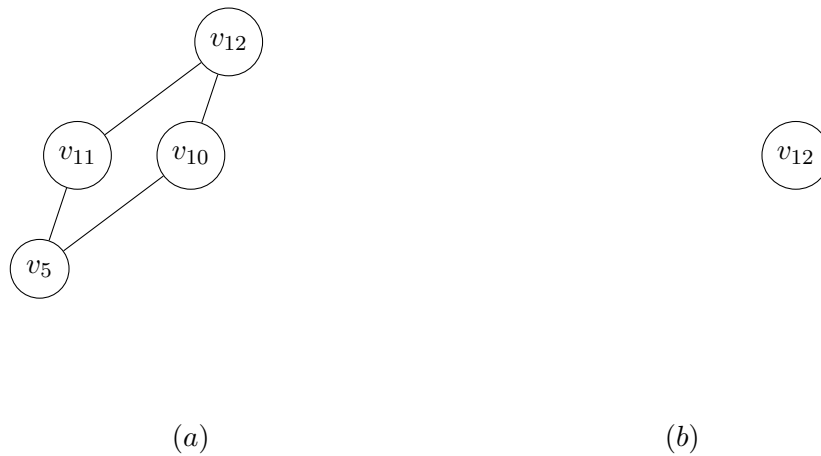


FIGURE 3.9 – Partition des arêtes en bicliques (suite)

$$W_4 = W_3 - N^+[v_4] = \{v_5, v_{10}, v_{11}, v_{12}\}, k = 3$$

on aura le graphe de la figure 3.9 (a).

6. **étape 5 :**

$$W_4 \neq \emptyset, v_5 = \min W_4, N^+[v_5] = \{v_{10}, v_{11}\}$$

$$W_5 = W_4 - N^+[v_5] = \{v_{12}\}, k = 4$$

on aura le graphe de la figure 3.9 (b) réduit à un sommet, donc on s'arrête et  $bp(G) = 4$ .

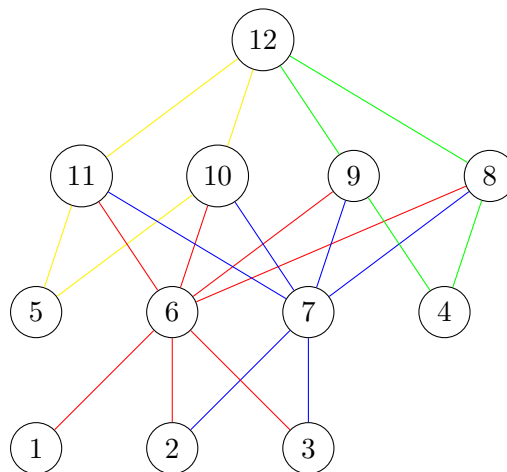


FIGURE 3.10 – Graphe biparti avec  $bp = 4$

La figure 3.10 montre la partition minimum des arêtes de  $G$  en 4 bicliques  $B_{v_1}, B_{v_2}, B_{v_4}$  et  $B_{v_5}$  qui sont représentés respectivement par les couleurs rouge, bleu, verte et jaune.

### 3.3 couverture des sommets par des bicliques

Étant donné un graphe biparti  $G = (X, Y, E)$ , le **problème de couverture des sommets de  $G$  par des bicliques** (Biclique Vertex Cover Problem) consiste à trouver une famille de bicliques  $\mathcal{F} = \{B_1 = (V_1, E_1), B_2 = (V_2, E_2), \dots, B_k = (V_k, E_k)\}$  de cardinalité minimale telle que tout sommet de  $G$  appartient à au moins une biclique de  $\mathcal{F}$ . Le **problème de partition des sommets** (Biclique vertex Partition Problem) est définie de la même manière avec la condition supplémentaire que tout sommet de  $G$  soit dans une unique biclique de  $\mathcal{F}$ , c'est-à-dire la famille des sommets  $\{V_1, V_2, \dots, V_k\}$  doit être une partition de l'ensemble  $V = X \cup Y$ . Notons par  $bvc(G)$  (resp.  $bvp(G)$ ) le plus petit nombre de bicliques qui couvrent (resp. partitionnent) les sommets de  $G$ . Contrairement aux invariants  $bc(G)$  et  $bp(G)$ , Fleischner et al. [34] ont remarqué que les invariants  $bvc$  et  $bvp$  sont égaux pour tout graphe  $G$ . La couverture des sommets d'un graphe par des bicliques intervient dans la sécurité de l'information, l'exploration de données, le commerce électronique, la recherche d'informations et la gestion des réseaux [15, 54].

Puisque les problèmes de couverture et de partition des sommets par des bicliques sont équivalents, dans la suite nous parlons que du problème de couverture. Le problème de décision associé au calcul de l'invariant  $pvc(G)$  est défini comme suit :

---

**PROBLÈME** : COUVERTURE DES SOMMETS PAR DES BICLIQUES

---

**Entrée** : Un graphe biparti  $G = (V, E)$  et un entier positif  $k$ .

**Question** : Peut-on couvrir les sommets de  $G$  par  $k$  bicliques ?

Ce problème est connu pour être  $\mathcal{NP}$ -complet [54]. Duginov [28] a montré que ce problème reste  $\mathcal{NP}$ -complet sur les classes de graphes d'élimination parfait et  $C_4$ -libre, et il a montré que la considération de ce problème sur les classes de graphes bipartis distance héréditaire et bipartis de permutation devient polynomiaux. En appliquant l'algorithme LexBFS sur un graphe biparti distance héréditaire, nous présentons un algorithme linéaire qui calcule le nombre minimum de bicliques nécessaires pour couvrir tous les sommets.

Dans la suite de cette section,  $G = (V, E)$  est un graphe biparti distance héréditaire,  $\sigma = (v_1, v_2, \dots, v_n)$  est un ordre LexBFS sur  $V$  et  $G_i$  est le sous-graphe induit par les sommets  $v_i, v_{i+1}, \dots, v_n$ . Rappelons que  $v_i$  est bisimpliciale dans  $G_i$  et l'unique biclique maximale dans  $G_i$  qui contient  $v_i$  est  $B_{v_i} = (N^+(v_i), N_{G_i}(u))$ , où le sommet  $u$  est l'un des voisins supérieur de  $v_i$ . Puisque les voisins supérieurs de tout sommet dans  $G_i$  sont les mêmes que dans  $G$ , alors il est clair que la restriction de  $\sigma$  sur  $G_i$  est un LexBFS.

**Définition 3.5.** On définit la séquence de sommets  $w_1, w_2, \dots, w_k$  comme suit :  $w_1 = v_1$ ,  $w_2$  est le plus petit sommet n'appartenant pas à  $B_{v_1}$  et  $w_3$  est le plus petit

sommet n'appartenant pas à

$$B_{w_1} \cup B_{w_2}$$

D'une manière générale, pour  $i = 2, \dots, k$ ,  $w_i$  est le plus petit sommet n'appartenant pas à

$$\bigcup_{s=1}^{s=i-1} B_{w_s}$$

**Théorème 3.12.** *Sous l'hypothèse de la définition 3.5, la famille  $\mathcal{F}$  de bicliques définie par*

$$\mathcal{F} = \{B_{w_1}, B_{w_2}, \dots, B_{w_k}\}$$

*est une couverture minimum des sommets.*

*Démonstration.* Il s'agit de montrer que

$$bvc(G) = k$$

Par définition, la famille de bicliques  $\mathcal{F}$  couvre tous les sommets de  $G$ . Ceci implique

$$bvc(G) \leq k$$

Par définition aussi, pour  $i \neq j$ , les sommets  $w_i$  et  $w_j$  ne peuvent pas appartenir à une même biclique de  $G$ . Ceci implique qu'il faut au moins  $k$  bicliques pour couvrir les sommets  $w_1, w_2, \dots, w_k$ , c'est-à-dire

$$bvc(G) \geq k$$

Par conséquent,

$$bvc(G) = k.$$

□

**Algorithme 3.4.** *Couverture des sommets en Bicliques (Vertices Biclique Cover)*

**Entrée :** *Un graphe biparti distance héréditaire connexe  $G = (V, E)$  et un ordre lex BFS  $\sigma = (v_1, v_2, \dots, v_n)$ .*

**Sortie :**  *$bvc(G)$ , le nombre minimum de bicliques qui couvrent tous les sommets de  $G$ .*

1. *Début*
2.  $k \leftarrow 0$ ;  $W \leftarrow \emptyset$ ;  $v = v_1 = \min(V - W) = \min V$
3. **Tant que  $W \neq V$  faire**
4.  $B_v = \{w/v \leq_\sigma w, d(v, w) \leq 2\}$ ;
5.  $W = W \cup B_v$ ;
6.  $v = \min(V - W)$ ;
7.  $k = k + 1$ ;



Le lemme suivant va nous servir dans le calcul de la complexité, il montre que deux bicliques  $B_{w_i}$ ,  $B_{w_j}$  ne peuvent pas avoir des arêtes en commun si les sommets  $w_i$  et  $w_j$  appartiennent à un même niveau.

**Lemme 3.13.** Pour  $i = 1, \dots, k$ , soit  $E_{w_i}$  l'ensemble des arêtes de la biclique  $B_{w_i}$ . Alors

$$\pi(w_i) = \pi(w_j) \Rightarrow E_{w_i} \cap E_{w_j} = \emptyset$$

*Démonstration.* Soient  $i$  et  $j$  deux indices différents. Rappelons que, d'après définition 3.5,  $d(w_i, w_j) > 2$ .

Par l'absurde, supposons qu'il existe une arête  $e$  dans  $E_{w_i} \cap E_{w_j}$ . Alors  $e \cap B_{w_i} \neq \emptyset$  et  $e \cap B_{w_j} \neq \emptyset$ . Il en résulte que l'une des extrémités de  $e$  est un voisin supérieur en commun de  $w_i$  et  $w_j$ . Ainsi, on aurait  $d(w_i, w_j) = 2$ , et ceci est une contradiction.  $\square$

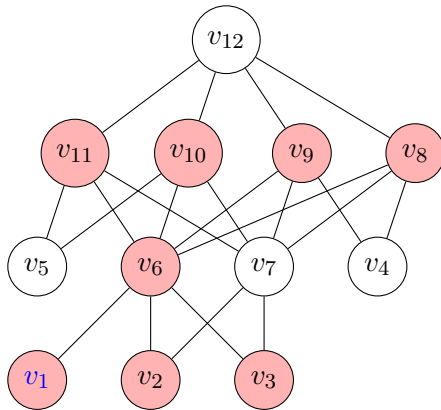
**Remarque 3.8.** Notons que, d'après le lemme 3.13, dans une couverture des sommets par des bicliques calculée selon la définition 3.5, une arête appartient à au plus deux bicliques.

**Théorème 3.14.** L'algorithme ci-dessus calcule une couverture minimum des sommets d'un graphe biparti distance héréditaire avec une complexité  $O(n + m)$ .

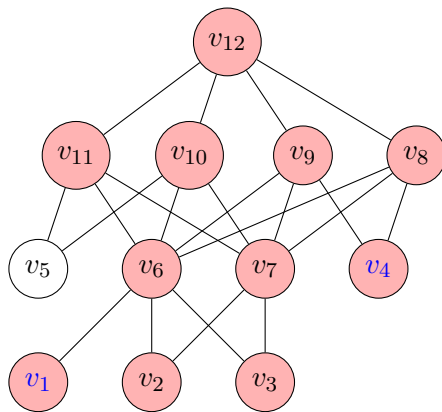
*Démonstration.* D'après le théorème 3.12, l'algorithme 3.4 est correcte.

Quant à la complexité, remarquons qu'elle est dominée par les lignes 4 et 5 où chaque sommet  $w$  est utilisé deux fois, une fois pour le mettre dans l'ensemble  $W$  et une autre fois pour calculer la biclique  $B_w$ ; quand aux arêtes, elles sont utilisées pour calculer les bicliques  $B_w$  et d'après la remarque 3.8, toute arête est utilisée au plus deux fois. Par conséquent, la complexité temporelle est en  $O(m + n)$ .  $\square$

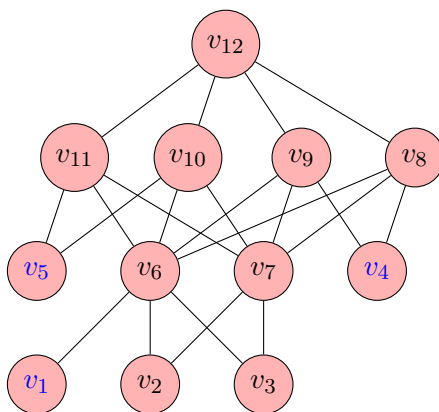
**Exemple 3.5.** Appliquons l'algorithme 3.4 au graphe ci-dessous.



$$\begin{aligned}
 k &= 1 \\
 B_v &= B_{v_1} \\
 W &= \{v_1, v_2, v_3, v_6, v_8, v_9, v_{10}, v_{11}\} \\
 V - W &= \{v_4, v_5, v_7, v_{12}\} \\
 v &= \min(V - W) = v_4
 \end{aligned}$$



$$\begin{aligned}
 k &= 2 \\
 B_v &= B_{v_4} \\
 W &= \{v_1, v_2, v_3, v_4, v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}\} \\
 V - W &= \{v_5\} \\
 v &= \min(V - W) = v_5
 \end{aligned}$$



$$\begin{aligned}
 k &= 3 \\
 B_v &= B_{v_5} \\
 W &= V, \text{ L'algorithme s'arrête} \\
 \theta &= 3
 \end{aligned}$$

### Conclusion du chapitre 3

Dans ce chapitre, nous avons vu que tous les problèmes liés aux calcul de bicliques dans les graphes bipartis sont NP-complet, excepté le problème de la biclique maximum pour les sommets qui est polynômial. En utilisant le parcours LexBFS, nous avons montré que tous ces problèmes peuvent être résolus en temps linéaire quand ils sont considérés sur un graphe biparti distance héréditaire.

## Chapitre 4

# Chaîne-déviations d'un poset

Kong and Ribemboin ont introduit une nouvelle relation d'ordre strict sur l'ensemble des antichaînes maximales pour l'inclusion d'un ensemble partiellement ordonné  $P$ , ce qui permet d'obtenir un nouvel ensemble ordonné  $D(P)$ . En appliquant récursivement cette définition, on obtient alors une suite d'ordres  $P, D(P), D^2(P), \dots$ . Ils ont montré que cette suite converge vers un ordre total après un nombre fini d'itérations. plus précisément, il existe un plus petit entier naturel dépendant de  $P$ , noté  $cdev(P)$  tel que  $D^{cdev(P)}(P)$  est une chaîne. Dans ce chapitre, pour un poset  $P$  qui n'est ni une antichaîne ni une somme linéaire d'autres posets, nous montrons que la valeur du paramètre  $cdev(P)$  est la distance dans le graphe  $Inc(P)$  entre deux sommets additionnels  $0_P$  et  $1_P$ , où  $0_P$  adjacent à tous les éléments minimaux de  $P$  et  $1_P$  adjacent à tous les éléments maximaux de  $P$ . Nous donnons aussi sa valeur pour les deux cas particuliers cités ci-dessus. Enfin, nous montrons que le paramètre "cdev" est un invariant de comparabilité, i.e les posets ayant un même graphe de comparabilité ont le même "cdev". voir [14, 48] pour plus d'information sur les invariants de comparabilité d'un poset fini.

### 4.1 Développement de l'ordre $D(P)$

**Définition 4.1.** Soit  $P$  un poset. Le couple  $(D(P), <_{D(P)})$  est l'ordre strict sur l'ensemble des antichaînes maximales  $D(P)$  de  $P$  défini comme suit : pour  $A, B \in D(P)$ ,  $A <_{D(P)} B$  si et seulement si pour  $a$  dans  $A$ , il existe  $b$  dans  $B$  tels que  $a <_P b$ .

Cette définition est équivalente à la suivante :

**Définition 4.2.** [63]  $A <_{D(P)} B$  si et seulement si pour tout  $b$  dans  $B$ , il existe  $a$  dans  $A$  tel que  $a <_P b$ .

**Lemme 4.1.** Soient  $A$  et  $B$  deux antichaînes incomparables dans  $D(P)$ . Alors

$$\forall b \in B, \exists a \in A \text{ tels que } a \parallel_P b$$

*Démonstration.* Soient  $A, B \in D(P)$  telles que  $A \parallel_{D(P)} B$ . Selon que  $A \cap B \neq \emptyset$  ou  $A \cap B = \emptyset$ , nous avons deux cas

1. Si  $A \cap B \neq \emptyset$ , c'est évident. En effet,  $\forall b \in B, \exists x \in A \cap B$  tels que  $b \parallel_P x$ .
2. Si  $A \cap B = \emptyset$ , par définition de  $D(P)$ , et puisque  $A$  et  $B$  sont incomparables, alors  $\exists a_1, a_2 \in A$  et  $\exists b_1, b_2 \in B$  tels que  $a_1 <_P b_1$  et  $b_2 <_P a_2$ .  
Par l'absurde, supposons que  $\exists b \in B$  tel que  $\forall a \in A, a \sim_P b$ . Puisque  $A$  est une

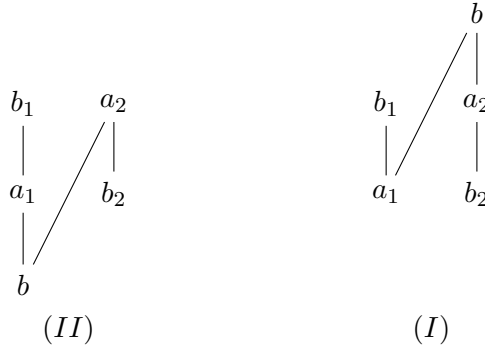


FIGURE 4.1 – Les antichaines  $A$  et  $B$  sont disjointes

antichaîne, on en déduit que, soit  $a <_P b, \forall a \in A$ , soit  $b <_P a, \forall a \in A$ .

- (a) Si  $b <_P a$ , pour tout  $a \in A$ , alors, en particulier, on aurait  $b <_P a_1$  (voir la figure 4.1(I)), et donc  $b <_P b_1$ , mais ceci est une contradiction.
- (b) Si  $a <_P b$ , pour tout  $a \in A$ , alors, en particulier, on aurait  $a_2 <_P b$  (voir la figure 4.1(II)), donc  $b_2 <_P b$ , mais ceci est une contradiction.

Dans les deux cas, nous avons une contradiction avec le fait que  $A$  est maximale. □

**Définition 4.3.** Étant donné un poset  $P$ , on définit une séquence  $(D^n(P))_{n \geq 0}$  de posets comme suit

$$\begin{cases} D^0(P) = P \\ D^{n+1}(P) = D(D^n(P)) \text{ si } n \geq 0. \end{cases}$$

Kong et Ribenboim ont prouvé que pour un poset fini  $P$ , il existe  $n \geq 0$  tel que  $D^n(P)$  est une chaîne.

**Définition 4.4.** On pose "cdev( $P$ )", et on lit chaîne-déviante de  $P$ , le plus petit entier positif tel que  $D^{\text{cdev}(P)}(P)$  est une chaîne.

**Théorème 4.2.** [63] Pour tout poset  $P$ , on a

$$\text{cdev}(P) \leq 2h(P) + 1$$

La figure 4.2 montre comment se forme la séquence  $P, D(P), D^2(P), \dots, D^i(P)$ . Dans cet exemple,  $D^3(P)$  est un ordre total.  $\text{cdev}(P) = 3$  réduit à un seul élément, donc  $\text{cdev}(P) = 3$ .

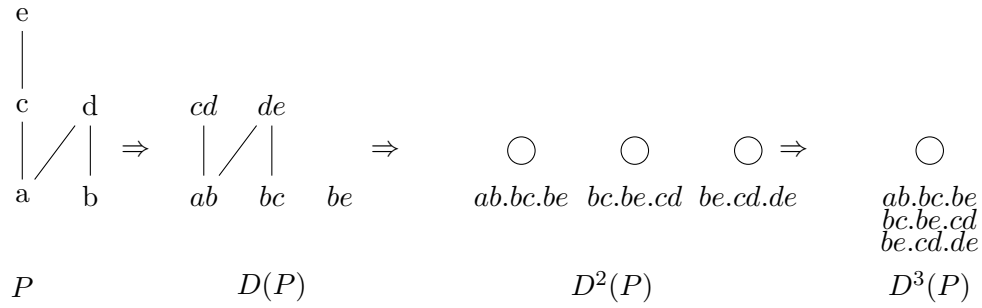


FIGURE 4.2 – Séquence de posets développée à partir de  $P$ .

**Remarque 4.1.** Notons que pour tout poset  $P$ ,  $MinP \in D(P)$ . De plus, par définition du poset  $D(P)$ ,  $\forall A \in D(P)$ ,  $Pred_{D(P)}(A) = \emptyset$  si et seulement si  $A \cap MinP \neq \emptyset$ . Par conséquent

$$MinD(P) = \{A \in D(P), A \cap MinP \neq \emptyset\}$$

D'une manière similaire,

$$MaxD(P) = \{A \in D(P), A \cap MaxP \neq \emptyset\}.$$

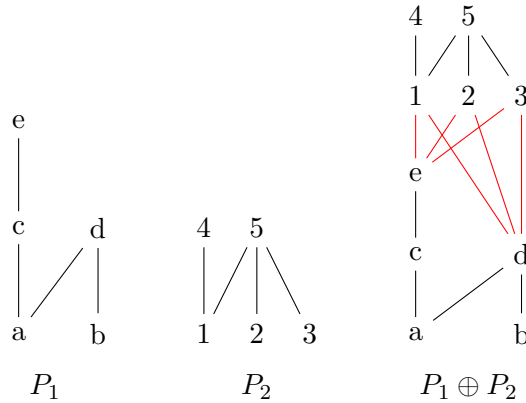


FIGURE 4.3 – Somme linéaire  $P_1 \oplus P_2$  des posets  $P_1$  et  $P_2$ .

Soient  $P_1 = (X_1, \leq_1)$ ,  $P_2 = (X_2, \leq_2)$  deux posets tels que  $X_1 \cap X_2 = \emptyset$ . On appelle **somme ordinale** (ordinal sum) ou **somme linéaire** (linear sum) de  $P_1$  et  $P_2$  l'ordre

$$P = (X, \leq_P) = P_1 \oplus P_2$$

tel que

$$X = X_1 \cup X_2$$

et

$$x \leq_P y \Leftrightarrow \begin{cases} x \in X_1, y \in X_2 & \text{ou,} \\ x, y \in X_i, i = 1, 2 \text{ et } x \leq_i y \end{cases} \quad (4.1)$$

Le diagramme de la somme ordinale  $P = P_1 \oplus P_2$  s'obtient en mettant le diagramme de  $P_2$  au-dessus de celui de  $P_1$  et relier chaque élément maximal de  $P_1$  à chaque élément minimal de  $P_2$ , voir la figure 4.3. Il est à noter que la somme ordinale de deux posets n'est pas commutative. Nous notons par

$$\oplus_{1 \leq i \leq h} P_i$$

la somme linéaire des posets  $(P_1, \leq_1), (P_2, \leq_2), \dots, (P_h, \leq_h)$  [14].

Soient  $P$  un poset et  $Inc(P)$  son graphe d'incomparabilité. Nous définissons le graphe  $G(P)$  en rajoutons au graphe  $Inc(P)$  deux sommets  $0_P$  et  $1_P$  tels que  $0_P$  adjacent à tous les éléments de  $MinP$  et  $1_P$  adjacent à tous les éléments de  $MaxP$  (voir la figure 4.4). Pour  $x, y \in G(P)$ ,  $d_{G(P)}(x, y)$  est la distance entre  $x$  et  $y$  dans le graphe  $G(P)$ .

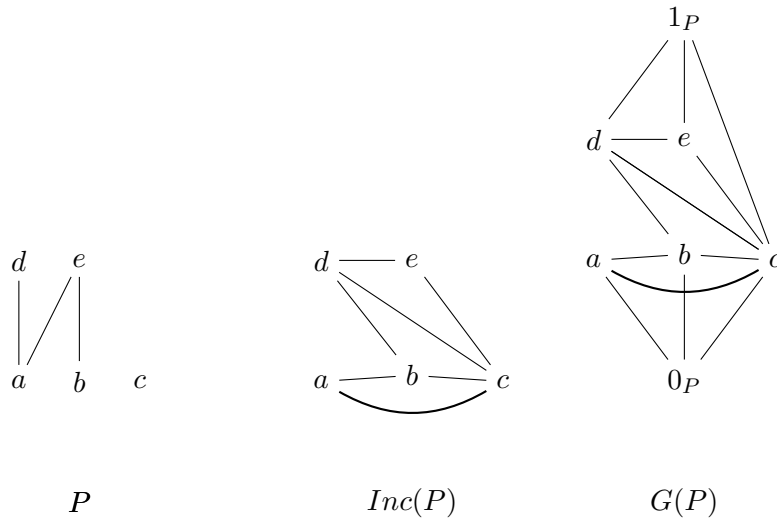


FIGURE 4.4 – Graphes associés au poset  $P$

**Remarque 4.2.** Notons que le graphe  $Inc(P)$  est connexe si et seulement si  $P$  n'est pas une somme linéaire des posets, et donc la distance entre les sommets  $0_P$  et  $1_P$  dans le graphe  $G(P)$  est définie si et seulement si  $P$  n'est pas une somme linéaire des posets. Dans ce cas, et puisque les sommets  $0_P, 1_P$  sont adjacents à chaque sommet de  $MinP$  et  $MaxP$  respectivement, il est facile de voir que si  $\mu$  est le plus court chemin entre  $0_P$  et  $1_P$  dans  $G(P)$ , alors il satisfait

$$|\mu \cap MinP| = |\mu \cap MaxP| = 1$$

## 4.2 Caractérisation de l'invariant $cdev(P)$

L'introduction du graphe  $G(P)$  va nous servir à montrer que  $cdev(P)$  est égal à la distance entre les sommets  $0_P$  et  $1_P$  dans  $G(P)$  si  $P$  n'est pas une antichaîne ni une somme linéaire d'autres posets (Théorème 4.6). Maintenant, il est clair que  $cdev(P) = 0$  si  $P$  est une chaîne et  $cdev(P) = 1$  si  $P$  est une antichaîne avec au moins deux éléments. Si  $P$  est une somme linéaire des posets  $P_1, P_2, \dots, P_h$ , le Théorème 4.3 montre qu'il existe  $i \in \{1, \dots, h\}$  tel que  $cdev(P) = cdev(P_i)$ .

**Théorème 4.3.** *Si un poset  $P$  est une somme linéaire des posets  $(P_1, \leq_1), (P_2, \leq_2), \dots, (P_h, \leq_h)$ , alors :*

$$cdev(P) = \max\{cdev(P_i), 1 \leq i \leq h\}.$$

*Démonstration.* Soient  $P = \oplus_{1 \leq i \leq h} P_i$  et  $\alpha = \max\{cdev(P_i), 1 \leq i \leq h\}$ . Notons que  $A \in D(P)$  si et seulement si  $\exists i \in \{1, 2, \dots, h\}$  tel que  $A \in D(P_i)$ , donc  $D(P) = \oplus_{1 \leq i \leq h} D(P_i)$ . Il en résulte que

$$D^\alpha(P) = \oplus_{1 \leq i \leq h} D^\alpha(P_i)$$

est une chaîne. Par conséquent,

$$cdev(P) = \alpha = \max\{cdev(P_i), 1 \leq i \leq h\}$$

□

**Remarque 4.3.** Par définition du graphe  $G(P)$ , il est facile de remarquer que  $d_{G(P)}(0_P, 1_P) = 2$  si et seulement si  $P$  contient au moins un élément isolé.

**Lemme 4.4.** *Soit  $P$  un poset qui n'est pas une antichaîne. Alors  $d_{G(P)}(0_P, 1_P) = 2$  si et seulement si  $D(P)$  est une antichaîne ayant au moins deux éléments.*

*Démonstration.* Par définition du graphe  $G(P)$  et puisque  $P$  n'est pas une antichaîne,  $d_{G(P)}(0_P, 1_P) = 2$  implique, premièrement  $MaxP \neq MinP$  et donc  $D(P)$  contient au moins deux éléments qui sont  $MinP$  et  $MaxP$ , deuxièmement il existe au moins un élément isolé  $x$  de  $P$  et donc  $x$  appartient à toutes les antichaînes maximales de  $P$ , i.e. les antichînes maximales de  $P$  sont deux à deux incomparables dans  $D(P)$ . D'où,  $D(P)$  est une antichîne avec au moins deux éléments.

Réciproquement, si  $D(P)$  est une antichîne avec au moins deux éléments alors, premièrement  $P$  n'est pas une antichaîne, autrement  $D(P)$  serait une chaîne, deuxièmement  $P$  contient au moins un élément isolé, autrement, on aurait  $MinP <_{D(P)} MaxP$  et  $D(P)$  ne serait pas une antichaîne. D'où,  $d_{G(P)}(0_P, 1_P) = 2$ . □

Pour  $n \geq 2$ , soit  $\mu = (0_{D(P)}, A_1, A_2, \dots, A_n, 1_{D(P)})$  un plus court chemin entre  $0_{D(P)}$  et  $1_{D(P)}$  dans  $G(D(P))$ . Par définition, et d'après la remarque 4.3, on a  $A_1 \in MinD(P)$  et  $A_2 \notin MinD(P)$ . Ceci montre que  $MinP <_{D(P)} A_2$  (voir la remarque 4.2), et donc l'antichaîne maximale  $MinP$  ne peut pas appartenir au plus court chemin allant de  $0_{D(P)}$  à  $1_{D(P)}$ . Pour les mêmes raisons, l'antichaîne  $MaxP$  ne peut pas être dans ce plus court chemin.

**Lemme 4.5.** *Si  $P$  est un poset tel que  $d_{G(P)}(0_P, 1_P) > 2$ , alors :*

$$d_{G(P)}(0_P, 1_P) = d_{G(D(P))}(0_{D(P)}, 1_{D(P)}) + 1.$$

*Démonstration.* Soit  $\mu = (0_P, a_1, \dots, a_n, 1_P)$  un plus court chemin entre  $0_P$  et  $1_P$  dans  $G(P)$ . Alors  $n > 1$  car  $d_{G(P)}(0_P, 1_P) > 2$ , et d'après le lemme 4.4,  $D(P)$  n'est pas une antichaîne. De plus,  $D(P)$  n'est pas une chaîne car  $a_2 \notin \text{Min}P$  (remarque 4.3) et donc il existe une antichaîne  $A$  dans  $P$  tel que  $\{a_1, a_2\} \subseteq A$  et les antichaînes  $A$  et  $\text{Min}P$  sont incomparables dans  $D(P)$ . Il en résulte que  $\mu' = (0_{D(P)}, A_1, \dots, A_{n-1}, 1_{D(P)})$ , où  $\{a_i, a_{i+1}\} \subseteq A_i$ , pour  $i = 1, \dots, n-1$  est un chemin dans  $G(D(P))$ . Par conséquent,

$$d_{G(D(P))}(0_{D(P)}, 1_{D(P)}) \leq d_{G(P)}(0_P, 1_P) - 1$$

Par ailleurs, soit  $\mu' = (0_{D(P)}, A_1, \dots, A_{n-1}, 1_{D(P)})$  un plus court chemin entre  $0_{D(P)}$  et  $1_{D(P)}$  in  $G(D(P))$ , et soit  $a_1 \in A_1 \cap \text{Min}P$ . Alors, d'après le lemme 4.1, pour  $i = 2, \dots, n-1$ ,  $\exists a_i \in A_i$  tel que  $a_{i-1} \parallel_P a_i$ . Notons que  $a_{n-1} \notin \text{Max}P$ , sinon le chemin  $\mu'' = (0_{D(P)}, B_1, \dots, B_{n-2}, 1_{D(P)})$ , où  $\{a_i, a_{i+1}\} \subseteq B_i$ , pour  $i = 1, \dots, n-2$  serait plus court que  $\mu'$ . Alors, d'après le lemme 4.1,  $\exists a_n \in \text{Max}P$  tel que  $a_{n-1} \parallel_P a_n$ , et donc  $(0_P, a_1, \dots, a_{n-1}, a_n, 1_P)$  est un chemin dans  $G(P)$ . Ceci implique que

$$d_{G(P)}(0_P, 1_P) \leq d_{G(D(P))}(0_{D(P)}, 1_{D(P)}) + 1$$

Par conséquent,

$$d_{G(P)}(0_P, 1_P) = d_{G(D(P))}(0_{D(P)}, 1_{D(P)}) + 1.$$

□

**Théorème 4.6.** *Soit  $P$  un poset. Si  $P$  n'est ni antichaîne ni une somme linéaire de posets, alors :*

$$cdev(P) = d_{G(P)}(0_P, 1_P)$$

*Démonstration.* Notons que pour tout poset  $P$ ,  $cdev(P) = 1 + cdev(D(P))$ .

Par induction sur  $d_{G(P)}(0_P, 1_P)$ , si  $d_{G(P)}(0_P, 1_P) = 2$  et puisque  $P$  n'est pas une antichaîne alors  $D(P)$  est une antichaîne ayant au moins deux éléments (voir le lemme 4.4), et donc  $cdev(D(P)) = 1$ . Par conséquent,

$$cdev(P) = 1 + 1 = d_{G(P)}(0_P, 1_P)$$

Supposons que pour tout poset  $P$  tel que  $d_{G(P)}(0_P, 1_P) = m \geq 2$ ,  $cdev(P) = d_{G(P)}(0_P, 1_P)$ . Soit  $Q$  un poset tel que  $d_{G(Q)}(0_Q, 1_Q) = m + 1$ . D'après le lemme 4.5, nous avons  $d_{G(Q)}(0_Q, 1_Q) = 1 + d_{G(D(Q))}(0_{D(Q)}, 1_{D(Q)})$ , donc  $d_{G(D(Q))}(0_{D(Q)}, 1_{D(Q)}) = m = cdev(D(Q))$  (d'après l'hypothèse). Il en résulte que

$$cdev(Q) = 1 + cdev(D(Q)) = 1 + d_{G(D(Q))}(0_{D(Q)}, 1_{D(Q)}) = d_{G(Q)}(0_Q, 1_Q).$$

□



**Remarque 4.4.** Notons que, d'après les théorèmes 4.3 et 4.6, si  $P$  est une somme linéaire des posets  $P_1, P_2, \dots, P_h$  telle que il existe  $i$  pour lequel  $P_i$  n'est pas une antichaîne, alors :

$$cdev(P) = \max\{d_{G(P_i)}(0_{P_i}, 1_{P_i}), 0 \leq i \leq h\}$$

où  $G(P_i)$  est le graphe de  $P_i$  comme il est défini précédemment. Si  $P$  est une somme linéaire des antichaînes  $A_1, \dots, A_h$  (dans ce cas  $P$  est appelé ordre faible [91]), le Théorème 4.3 implique que

$$cdev(P) = cdev(A_1) = \dots = cdev(A_h) = 1$$

(le même résultat a été trouvé dans la référence [90]).

Il reste à montrer que "cdev" est un invariant de comparabilité, i.e. si  $P$  et  $Q$  sont des posets avec un même graphe de comparabilité, alors

$$cdev(P) = cdev(Q)$$

Pour cela, il suffit de montrer qu'on peut déduire l'un des posets à partir de l'autre comme il est indiqué dans le lemme suivant.

**Lemme 4.7.** (D. Kelly [61]) Si  $P$  et  $Q$  sont deux posets finis avec un même graphe de comparabilité, alors  $P$  peut être transformé en  $Q$ , en remplaçant un nombre fini de sous-ordres autonomes par leurs duaux.

**Remarque 4.5.** Soit  $S$  un sous-ordre autonome de  $P$ , et soit  $Q$  le poset obtenu en inversant  $S$  dans  $P$ .

1. Par définition de  $S$ , si  $S \cap \text{Min}P \neq \emptyset$  (resp.  $S \cap \text{Max}P \neq \emptyset$ ) alors  $S^d \cap \text{Min}Q \neq \emptyset$  (resp.  $S^d \cap \text{Max}Q \neq \emptyset$ );
2. Si  $S \neq P$  et  $\mu = (0_P, a_1, \dots, a_n, 1_P)$  est un plus court chemin de  $0_P$  à  $1_P$  dans  $G(P)$  alors  $|\mu \cap S| \leq 1$ . En effet, par exemple, si  $a_i, a_j$  dans  $S$ , avec  $i < j$  alors  $a_{i-1}$  et  $a_j$  (ou  $a_i$  et  $a_{j+1}$ ) sont incomparables et donc  $\mu' = \mu \setminus \{a_i\}$  (ou  $\mu'' = \mu \setminus \{a_j\}$  respectivement) serait un chemin de  $0_P$  à  $1_P$  avec une longueur plus petite.

**corollaire 4.8.** Le paramètre "cdev" est un invariant de comparabilité.

*Démonstration.* Soit  $S$  un sous-ordre autonome dans un poset  $P$ , et soit  $Q$  le poset obtenu par inversion de  $S$  dans  $P$ . Pour montrer que  $cdev(P) = cdev(Q)$ , il suffit de montrer que pour tout plus court chemin  $\mu$  dans  $G(P)$  de  $0_P$  à  $1_P$  il existe un chemin dans  $G(Q)$  de  $0_Q$  à  $1_Q$  avec une longueur, au plus, égale à celle de  $\mu$ .

Soit  $\mu = (0_P, a_1, \dots, a_n, 1_P)$  un plus court chemin entre  $0_P$  et  $1_P$  dans  $G(P)$ .

1. Si  $S = P$ , alors  $Q = P^d$  et donc  $\mu' = (0_Q, a_n, a_{n-1}, \dots, a_2, a_1, 1_Q)$  est un chemin de  $0_Q$  à  $1_Q$  dans  $G(Q)$ .
2. Si  $S \neq P$ , il y a deux cas :
  - (a) Si  $\mu \cap S = \emptyset$  ou  $\mu \cap S = \{a_i\}$ , avec  $a_i$  ni minimal, ni maximal dans  $P$  (autrement dit,  $a_i \neq a_1$  et  $a_i \neq a_n$ ), alors  $\mu' = (0_Q, a_1, \dots, a_n, 1_Q)$  forme un chemin dans  $G(Q)$ ;

- (b) Si  $\mu \cap S = \{a_1\}$  (resp.  $\mu \cap S = \{a_n\}$ ), alors d'après la remarque 4.5, il existe un élément  $s$  dans  $MinQ \cap S^d$  (resp. dans  $MaxQ \cap S^d$ ) tel que  $\mu' = (O_Q, s, a_2, \dots, a_n, 1_Q)$  (resp.  $\mu'' = (0_Q, a_1, \dots, s, 1_Q)$ ) forme un chemin dans  $G(Q)$ .

□

## Conclusion du chapitre 4

Nous avons vu dans ce chapitre que pour tout ensemble ordonné fini, la suite d'ensembles ordonnés développée sur ses antichâmes maximales converge vers un ordre total au bout d'un nombre fini d'itérations. Pour tout ordre fini et en utilisant son graphe de comparabilité, nous avons montré comment calculer ce nombre d'itération nécessaire pour atteindre l'ordre total sans passer par le calcul explicite de tous les termes de la suite.

# Chapitre 5

## $P$ -couplage

Dans ce dernier chapitre, nous introduisons la notion du  $P$ -couplage qui généralise un peu la notion classique du couplage dans un graphe biparti. Après la définition et la caractérisation d'un  $P$ -couplage, nous montrons que le calcul d'un  $P$ -couplage maximum peut se ramener au calcul d'une clique maximum dans un graphe particulier. Nous donnons à la fin quelques cas dans lesquels le problème du  $P$ -couplage peut être résolu en un temps polynômial.

### 5.1 Définition et caractérisation d'un $P$ -couplage

Dans toute la suite de ce chapitre,  $P = (E, \leq_P)$  est un poset,  $G = (X, Y, E')$  est un graphe biparti associé à  $P$  tel que  $|E| = |E'|$  et  $f : E \rightarrow E'$  est une application bijective qui à tout élément  $e$  de  $P$  associe une arête  $e'$  dans  $G$ . Rappelons qu'un couplage dans un graphe est un ensemble d'arêtes deux à deux non adjacentes. Ci-dessous, nous introduisons la notion de  $P$ -couplage qui généralise celle du couplage, considéré dans un graphe biparti, et nous verrons qu'un couplage dans un graphe biparti peut être vu comme un  $P$ -couplage particulier.

**Définition 5.1.** On appelle  $P$ -couplage ou couplage généralisé de  $G$  tout ensemble d'arêtes  $M$  tel que, pour toute antichaine maximale  $A$  de  $P$ , la restriction de  $M$  aux arêtes de  $f(A)$  sont deux à deux disjointes. Autrement dit, un ensemble d'arêtes  $M \subseteq E'$  est un  $P$ -couplage si et seulement si pour toute antichaine maximale  $A$ ,  $M \cap f(A)$  est un couplage dans  $G$ .

**Exemple 5.1.** La figure 5.1 montre un poset  $P$ , un graphe biparti  $G$  et un isomorphisme  $f$  qui est défini par

$$f(i) = e_i, \text{ pour } i = 1, \dots, 7$$

L'ensemble de toutes les antichânes maximales de  $P$  est  $D(P) = \{A_1 = 12, A_2 = 14, A_3 = 17, A_4 = 23, A_5 = 25, A_6 = 34, A_7 = 37, A_8 = 45, A_9 = 567\}$ . On peut vérifier que les ensembles des arêtes  $M_1 = \{e_2, e_4, e_7\}$ ,  $M_2 = \{e_2, e_4, e_5, e_6\}$  sont deux  $P$ -couplages dans  $G$ . Par contre, l'ensemble  $M = \{e_3, e_5, e_7\}$  n'est pas un  $P$ -couplage

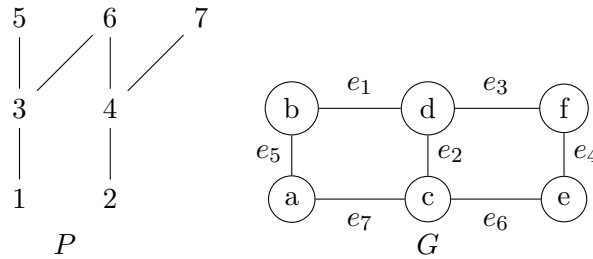


FIGURE 5.1 – Éléments d'une instance d'un problème du  $P$ -couplage

car  $M \cap f(A_9) = \{e_5, e_7\}$  n'est pas un couplage.

Le problème du  $P$ -couplage est un problème d'optimisation combinatoire qui consiste à trouver un  $P$ -couplage  $M$  dans  $G$  avec un maximum d'arêtes possible. Notons par  $\eta(G)$  le cardinal d'un  $P$ -couplage maximum. Le problème de décision associé est défini comme suit :

---

PROBLEME DU  $P$ -COUPLAGE MAXIMUM

**Instance** : Un poset  $P = (E, \leq)$ , un biparti  $G = (X, Y, E')$  avec  $|E| = |E'|$  et  $k > 0$ .

**Question** : Existe-t-il un  $P$ -couplage  $M$  dans  $G$  vérifiant  $|M| \geq k$  ?

---

Par définition, il est clair que tout couplage de  $G$  est aussi un  $P$ -couplage, c'est-à-dire un ensemble d'arêtes deux à deux disjointes définit toujours un  $P$ -couplage. Le théorème suivant donne une condition nécessaire et suffisante pour qu'un ensemble d'arêtes soit un  $P$ -couplage

**Théorème 5.1.** *Un ensemble d'arêtes  $M$  est un  $P$ -couplage dans  $G$  si et seulement si, quel que soient  $u, v \in M$*

$$u \cap v = \emptyset \text{ ou } f^{-1}(u) \sim_P f^{-1}(v).$$

*Démonstration.* Si  $M$  n'est pas un  $P$ -couplage, alors il existe une antichaîne maximale  $A$  telle que  $M \cap f(A)$  n'est pas un couplage, ceci implique qu'il existe deux arêtes  $u, v \in f(A) \cap M$  vérifiant  $u \cap v \neq \emptyset$ . Autrement dit, il existe  $u, v \in M$  tels que  $u \cap v \neq \emptyset$  et  $f^{-1}(u), f^{-1}(v) \in A$ , c'est-à-dire il existe  $u, v \in M$  tels que  $u \cap v \neq \emptyset$  et  $f^{-1}(u) \parallel_P f^{-1}(v)$ .

Réciproquement, supposons qu'il existe  $u, v \in M$  tel que  $u \cap v \neq \emptyset$  et  $f^{-1}(u) \parallel_P f^{-1}(v)$ . Puisque les éléments  $f^{-1}(u) \parallel_P f^{-1}(v)$ , alors il existe une antichaîne maximale  $A$  contenant  $f^{-1}(u)$  et  $f^{-1}(v)$ , et donc  $f(A)$  contient  $u$  et  $v$ . Par conséquent,  $u, v \in f(A) \cap M$  et  $u \cap v \neq \emptyset$ . D'où  $M$  n'est pas un  $P$ -couplage.  $\square$

**corollaire 5.2.** *Si  $\kappa(P)$ ,  $\nu(G)$  désignent respectivement le nombre d'éléments dans une chaîne maximum et dans un couplage maximum, alors*

1.  $\kappa(P) \leq \eta(G)$  ;
2.  $\nu(G) \leq \eta(G)$ .

*Démonstration.* Est une conséquence directe du théorème 5.1. □

Nous allons voir maintenant que le calcul d'un  $P$ -couplage maximum peut se ramener au calcul d'une clique maximum dans un graphe associé à  $P$  et  $G$ .

**Définition 5.2.** Soit  $H = (V(H), E(H))$  le graphe défini comme suit : les sommets de  $H$  sont les arêtes de  $G$  ; deux sommets  $u, v$  sont adjacents dans  $H$  si et seulement si,  $u, v$  comme des arêtes de  $G$ , peuvent appartenir simultanément à un  $P$ -couplage.

Dans la figure 5.2, nous avons le graphe  $H$  associé à  $P$  et  $G$  représentés dans la figure 5.1. Il est facile de se convaincre que tout  $P$ -couplage dans  $G$  est une clique dans  $H$ , et vice versa.

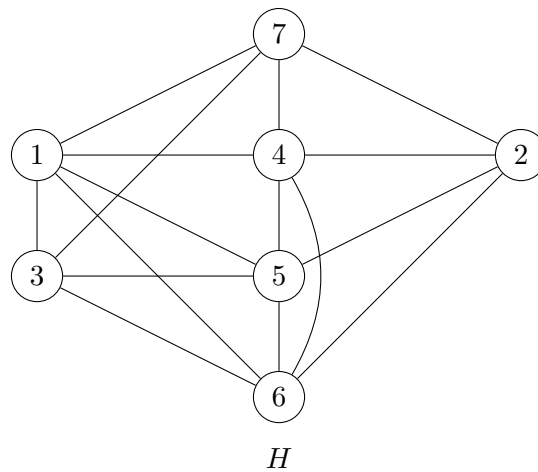


FIGURE 5.2 – Réduction d'un problème du  $P$ -couplage au problème de la clique

- Remarque 5.1.**
1. Le long de ce chapitre,  $H$  est le graphe associé à  $P$  et  $G$  selon la définition 5.2.
  2. Par abus de langage, un élément de  $P$  est vu comme une arête quand il est considéré dans le graphe  $G$ , et il est vu comme un sommet quand il est considéré dans le graphe  $H$ . Ainsi, selon le théorème 5.1 et la définition 5.2, deux sommets  $u, v$  sont adjacents dans  $H$  si et seulement si  $u, v$  sont disjoints dans  $G$  ou comparables dans  $P$ .

**Lemme 5.3.** *Le graphe  $H$  ne peut pas contenir un trou impair de longueur 7, c'est-à-dire  $H$  est sans  $C_7$ .*

*Démonstration.* Supposons le contraire et soit  $\mu = (x_1, x_2, x_3, x_4, x_5, x_6, x_7)$  un trou de longueur 7, voir la figure 5.3.

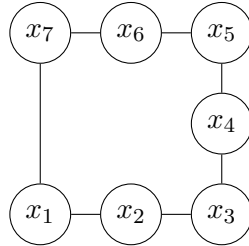


FIGURE 5.3 – Trou de longueur 7

Rappelons que, d'après le théorème 5.1, deux sommets  $u, v$  sont non adjacents dans  $H$  si et seulement si ils sont non disjoints dans  $G$  et incomparables dans  $P$ . Dans le cycle  $\mu$ , les sommets  $x_1, x_3$  et  $x_5$  sont deux à deux non adjacents, donc ils sont deux à deux non disjoints dans  $G$  et deux à deux non comparables dans  $P$ . Ainsi ces trois sommets forment une étoile dans  $G$  et une antichaine dans  $P$ . Le sommet  $x_7$  n'est pas adjacent aux sommets  $x_3$  et  $x_5$  dans  $\mu$ , donc l'arête  $x_7$  intersecte les arêtes  $x_3, x_5$  dans  $G$  et forment ainsi une étoile (voir la figure 5.4).(a) ; et dans ce cas  $x_7$  et  $x_1$  qui sont adjacents dans  $H$  et non disjoints dans  $G$  doivent être comparables dans  $P$ , voir la figure 5.4.(b).

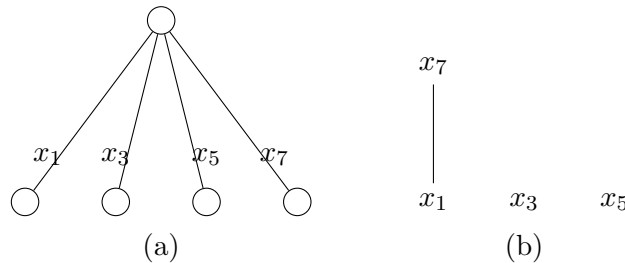


FIGURE 5.4 – Le graphe  $H$  est sans  $C_{2k+1}, k \geq 3$

Le sommet  $x_2$  n'est pas adjacent aux sommets  $x_5$  et  $x_7$  dans  $\mu$ , donc l'arête  $x_2$  intersecte les arêtes  $x_5, x_7$  dans  $G$  et forment ainsi une étoile (voir la figure 5.5).(a) ; et dans ce cas  $x_2$ , qui est adjacent à  $x_1$  et  $x_3$  dans  $\mu$ , doit être comparable à  $x_1$  et  $x_3$  dans  $P$ , voir la seule configuration possible donnée par la figure 5.5.(b).

Le sommet  $x_4$  n'est pas adjacent aux sommets  $x_1, x_2$  et  $x_7$  dans  $\mu$ , donc l'arête  $x_4$  intersecte les arêtes  $x_1, x_2$  et  $x_7$  dans  $G$  et forment ainsi une étoile (voir la figure 5.6).(a) ; et dans ce cas  $x_4$ , qui est adjacent à  $x_3$  et  $x_5$  dans  $H$ , doit être com-

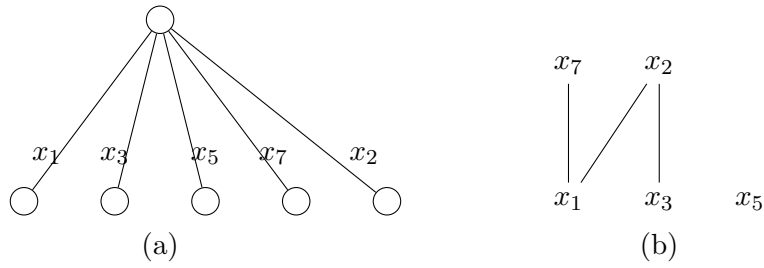


FIGURE 5.5 – Le graphe  $H$  est sans  $C_{2k+1}$ ,  $k \geq 3$

parable à  $x_3$  et  $x_5$  dans  $P$ , voir la seule configuration possible donnée par la figure 5.6.(b).

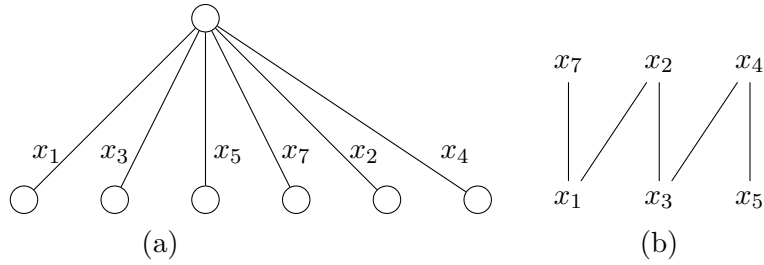


FIGURE 5.6 – Le graphe  $H$  est sans  $C_{2k+1}$ ,  $k \geq 3$

Il nous reste le sommet  $x_6$ , et pour les mêmes raisons que celles évoquées ci-dessus, ce sommet forme une étoile dans  $G$  avec les sommets du cycle  $\mu$ . Par ailleurs, dans  $P$ , l'élément  $x_6$  doit être comparable aux éléments  $x_5$  et  $x_7$ , et incomparable aux éléments  $x_1, x_2, x_3$  et  $x_4$ . Pour rajouter l'élément  $x_6$  à la configuration de la figure 5.6.(b), nous avons trois possibilités qui sont illustrées par la figure 5.7.

1. Si  $x_7 \leq_P x_6$  et  $x_5 \leq_P x_6$  (voir la figure 5.7.(a)), on aurait  $x_1 \leq_P x_6$ ; mais ceci est une contradiction avec le fait que  $x_1$  et  $x_6$  ne sont pas adjacents dans  $\mu$ .
2. Si  $x_6 \leq_P x_7$  et  $x_6 \leq_P x_5$  (voir la figure 5.7.(b)), on aurait  $x_6 \leq_P x_4$ ; et ceci est une contradiction avec le fait que  $x_4$  et  $x_6$  ne sont pas adjacents dans  $\mu$ .
3. Si  $x_6 \leq_P x_7$  et  $x_5 \leq_P x_6$  (voir la figure 5.7.(c)), on aurait  $x_5 \leq_P x_7$ ; et ceci est une contradiction avec le fait que  $x_5$  et  $x_7$  ne sont pas adjacents dans  $\mu$ .

□

**corollaire 5.4.** *Le graphe  $H$  ne peut pas contenir de trou impair de longueur supérieure ou égale à 7.*

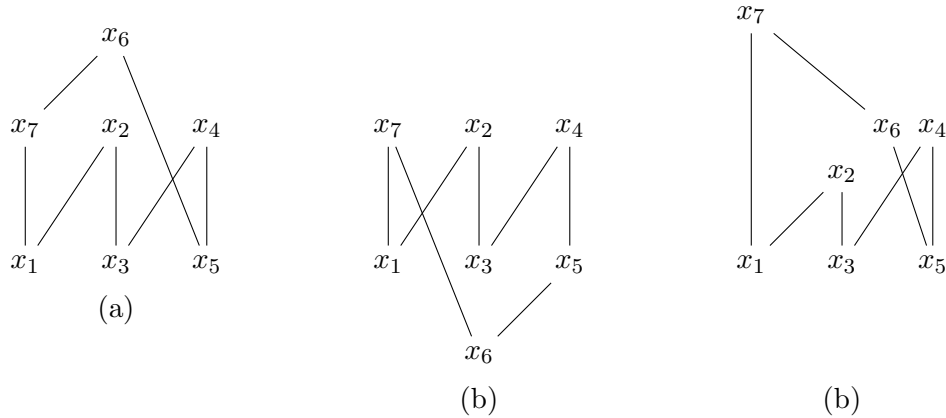


FIGURE 5.7 – Le graphe  $H$  est sans  $C_{2k+1}, k \geq 3$

## 5.2 Études de cas particuliers

### 5.2.1 Le poset $P$ est une antichaîne

**Proposition 5.5.** *Si  $P$  est une antichaîne, alors tout  $P$ -couplage est un couplage.*

*Démonstration.* Puisque  $P$  est une antichaîne, et d’après le théorème 5.1, un ensemble d’arêtes  $M$  est un  $P$ -couplage si et seulement si  $\forall u, v \in M, u \cap v = \emptyset$ , c’est-à-dire  $M$  est un  $P$ -couplage si et seulement si il est un couplage.  $\square$

Ainsi, si  $P$  est une antichaîne, alors un  $P$ -couplage maximum est un couplage maximum dans le graphe biparti  $G$ , et pour le calculer, on peut utiliser l’algorithme d’Edmond [29].

### 5.2.2 Le poset $P$ est faible

Rappelons qu’un ordre faible (weak order) est un ordre obtenu par composition séries d’antichaînes. Ainsi, les seules antichaînes d’un ordre faible sont ses niveaux.

**Proposition 5.6.** *Si  $P$  est un ordre faible, alors le  $P$ -couplage maximum est égal à l’union des couplages maximums correspondant à chaque niveau de  $P$ .*

*Démonstration.* Soient  $N_0, N_1, \dots, N_k$  les niveaux de  $P$ , et pour  $i = 0, \dots, k$ , soit  $M_i$  le couplage maximum induit par les éléments du niveau  $N_i$  dans  $G$ . Il s’agit de montrer que

$$\mathcal{M} = \bigcup_{i=0}^{i=k} M_i$$

est un  $P$ -couplage maximum.



1. Montrons que  $\mathcal{M}$  est un  $P$ -couplage.  
Soient  $u, v$  deux arêtes dans  $\mathcal{M}$ . Nous avons deux cas
  - (a) S'il existe  $i$  tel que  $u, v \in M_i$ , alors  $u$  et  $v$  sont disjoints dans  $G$  car  $M_i$  est un couplage, et donc elles peuvent appartenir à un même  $P$ -couplage.
  - (b) Si  $u \in M_i, v \in M_j$ , avec  $i \neq j$ , alors  $u$  et  $v$  appartiennent à deux niveaux différents dans  $P$ . Puisque  $P$  est un ordre faible, alors  $u$  et  $v$  sont comparables dans  $P$ , et donc elles peuvent appartenir à un même  $P$ -couplage. Par conséquent  $\mathcal{M}$  est un  $P$ -couplage.
2. Montrons maintenant que  $\mathcal{M}$  est maximum.  
Soit  $\mathcal{M}'$  un  $P$ -couplage maximum, et pour  $i = 0, \dots, k$ , posons

$$M'_i = N_i \cap \mathcal{M}'$$

Ainsi, pour  $i = 0, \dots, k$ ,  $M'_i$  est un couplage qui correspond à  $N_i \cap \mathcal{M}'$ ; et puisque, par hypothèse,  $M_i$  est un couplage maximum correspondant à  $N_i, i = 0, 1, \dots, k$ , alors, pour  $i = 0, \dots, k$ ,

$$|M'_i| \leq |M_i|$$

donc

$$\sum_{i=0}^k |M'_i| \leq \sum_{i=0}^k |M_i|$$

et ceci implique

$$\mathcal{M}' \leq \mathcal{M}$$

Du fait que  $\mathcal{M}'$  est maximum, on en déduit que

$$\mathcal{M}' = \mathcal{M}$$

Par conséquent,  $\mathcal{M}$  est maximum. □

Ainsi, si  $P$  est un ordre faible ayant les niveaux  $N_0, N_1, \dots, N_k$ , alors pour calculer un  $P$ -couplage maximum, il suffit d'appliquer l'algorithme d'Edmond pour calculer un couplage maximum pour chaque sous-graphe biparti  $G_i$  qui est induit par le niveau  $N_i, i = 0, \dots, k$ ; et calculer ensuite l'union de ces couplages.

### 5.2.3 Le graphe $G$ est une étoile

Rappelons qu'une **étoile** (star) est un graphe biparti complet  $K_{1,p}$ .

**Proposition 5.7.** *Si le graphe biparti  $G$  est une étoile, alors le nombre d'éléments dans un  $P$ -couplage maximum est égal au nombre d'éléments dans une chaîne maximum dans  $P$ .*

*Démonstration.* Si  $G$  est une étoile, ces arêtes sont deux à deux non disjointes, et ceci implique que les éléments d'un  $P$ -couplage doivent être deux à deux comparables dans  $P$ , et donc un  $P$ -couplage maximum est une chaîne maximum dans  $P$ . □

## **Conclusion du chapitre 5**

Une instance d'un problème du  $P$ -couplage est constituée par un ensemble ordonné  $P$  et un graphe biparti  $G$  ayant autant d'arêtes que d'éléments dans  $P$ . En terme de complexité algorithmique, nous ne connaissons pas encore la classe de problèmes contenant ce problème. Par ailleurs, nous avons donné une condition nécessaire et suffisante pour qu'un ensemble d'arêtes constitue un  $P$ -couplage, nous avons montré que ce problème se réduit polynomialement en un problème de la clique et donnons enfin quelques classes de posets et graphes pour lesquelles ce problème est polynômial.

# Conclusion et perspectives

Dans cette thèse, nous avons étudié trois types de problèmes liés au calcul des invariants dans les graphes et ordres.

Le premier problème est celui de décomposition d'un graphe biparti en bicliques. Les graphes bipartis distance héréditaire ont été introduits par Howorka en 1977 ; ils ont été caractérisés par l'existence d'un ordre d'élimination parfait pour les sommets (F. Nicolai [76, 77]). Nous avons montré comment calculer cet ordre efficacement par le parcours en largeur lexicographique (LexBFS). Nous avons aussi, en utilisant le même parcours dans cette classe, établi des algorithmes de complexité temporelle linéaire pour calculer une biclique maximum pour les sommets, une biclique maximum pour les arêtes, une décomposition minimum des arêtes en bicliques et une décomposition minimum des sommets en bicliques. Comme perspectives, il faut appliquer le parcours LexBFS pour calculer efficacement (avec une complexité linéaire) d'autres invariants sur cette même classe de graphes ; généraliser l'application de LexBFS à d'autres classes de graphes et posets bipartis.

Le second problème a été introduit par Kong et Ribinboim. Il consiste, pour un poset  $P$  fini quelconque, de trouver le nombre minimum d'itérations pour transformer une suite d'ordres développée récursivement sur les antichaines maximales de  $P$  sans passer par le calcul explicite des termes de cette suite. En introduisant deux sommets artificiels sur le graphe d'incomparabilité  $Inc(P)$  de  $P$ , notés  $0_P$ ,  $1_P$ , et qui sont connectés respectivement aux éléments minimaux et maximaux de  $P$  dans  $Inc(P)$ , nous avons résolu définitivement ce problème en montrant que ce nombre d'étapes, noté  $cdev$ , est égal à la distance entre les sommets  $0_P$  et  $1_P$ . Nous avons aussi montré que ce paramètre est un invariant de comparabilité.

Le dernier problème est mixte, il est appelé problème de  $P$ -couplage, il généralise le problème classique du couplage. Nous avons donné une caractérisation de ce problème et montrons comment calculer efficacement un  $P$ -couplage pour des posets et graphes particuliers. Comme perspective, il reste à trouver la classe de problèmes contenant celui-ci, ou trouver d'autres classes de posets et de graphes bipartis plus générales pour lesquelles se problème peut être résolu efficacement.

# Bibliographie

- [1] M. Aigner and R. Wille (eds. 1991) Combinatorics of ordered sets, Discrete Mathematics, 88(2-3).
- [2] G. Alexe, S. Alexe, Y. Crama, S. Foldes, P.L. Hammer, B. Simeone : Consensus algorithms for the generation of all maximal bicliques. Discrete Appl. Math. 145(1), 11–21 (2004)
- [3] J. Amilhastre, M. C. Vilarem, and P. Janssen. Complexity of minimum biclique cover and minimum biclique decomposition for bipartite domino-free graphs. Discrete Appl. Math., 86(2-3) : (1998)125-144.
- [4] H.J. Bandelt, A. Henkmann, F. Nicolai, Powers of distance-hereditary graphs, Discrete Math. 145 (1995)37-60.
- [5] H. J. Bandelt, H.M. Mulder, Distance-hereditary graphs J. Comb. Theory (B) 41 1986 182–208
- [6] M. Barbut and B. Monjardet (1970) Ordre et classification, algèbre et combinatoire, tomes I et II, Hachette, Paris.
- [7] C. Berge and J. L. Ramírez Alfonsín. Origins and Genesis. In Ramírez Alfonsín and Reed, pages 1-12 (1994).
- [8] M. Bomhoff, B. Manthey, Bisimplicial edges in bipartite graphs, Discrete Applied Mathematics 161 (2013) 1699–1706.
- [9] K.S. Booth and G.S. Lueker, Testing for the consecutive ones property, interval graphs and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (1976). 335-379.
- [10] A. Brandstad, F.F. Dragan, F. Nicolai, LexBFS-orderings and powers of chordal graphs, Discrete Mathematics 171, 27-42 (1997).
- [11] A. Brandstadt, Special graph classes - a survey, Technical Report SM-DU-199, Schriftenreihe des Fachbereichs Mathematik, 1993.
- [12] A. Brandstädt, V.B. Le., and J. Spinrad. Graph classes : a survey. SIAM, Philadelphia, (1999).
- [13] C. Bron et J. Kerbosch : Finding all cliques of an undirected graph. Communications of the ACM, 16(9) :575-577, (1973).
- [14] N. Caspard, B. Leclerc, B. Monjardet, Finite Ordered Sets : Concepts, Results and Uses. Springer, Berlin Heidelberg New York (2007).

- [15] D. Chakrabarti, S. Papadimitriou, D.S. Modha, and C. Faloutsos. Fully automatic cross-associations. In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 79–88, (2004).
- [16] S.A. Cook : The complexity of theorem-proving procedures. In Proceedings of the third annual ACM symposium on Theory of computing, pages 151-158. ACM, (1971).
- [17] T. Cormen, C. Leiserson, R. Rivest, C. Stein : Introduction à l’algorithmique, cours et exercices, 2<sup>ieme</sup> édition, Dunod, (2004).
- [18] D. G. Corneila, E. Kohler, J. M. Lanlignel, On end-vertices of Lexicographic Breadth First Searches, Discrete Applied Mathematics 158 (2010) 434-443.
- [19] D.G. Corneil, Lexicographic breadth first search—A survey, in : 30th International Workshop on Graph Theory, WG2004, in : LNCS, vol. 3353, Springer, pp. 1-19(2004).
- [20] D.G. Corneil, S. Olariu, L. Stewart, Linear time algorithms for dominating pairs in asteroidal triple-free graphs, SIAM J. Comput. 28 (1999) 1284-1297
- [21] Damaschke, Peter ; Müller, Haiko ; Kratsch, Dieter Domination in convex and chordal bipartite graphs. Inf. Process. Lett. 36, No.5, 231-236 (1990).
- [22] G. Damiand, M. Habib, Ch. Paul A simple paradigm for graph recognition : application to cographs and distance-hereditary graphs Theo. Comp. Sci. 263 99-111 (2001).
- [23] M. Dawande, P. Keskinocak, J.M. Swaminathan, S. Tayur : On bipartite and multipartite clique problems. J. Algorithms 41(2), 388-403 (2001).
- [24] R. Diestel : Graph Theory. Graduate Texts in Mathematics, volume 173. Springer-Verlag, Heidelberg, troisième édition (2005).
- [25] G.A. Dirac, On rigid circuit graphs. In Abhandlungen aus dem Mathematischen Seminar der Universit ”at Hamburg, volume 25, pages 71-76. Springer, (1961).
- [26] F.F. Dragan, Almost diameter of a house-hole-free graph in linear time via LexBFS, Discrete Appl. Math. 95 (1999) 223-239.
- [27] F.F. Dragan, F. Nicolai, A. Brandstadt, LexBFS-orderings and powers of graphs, Proceedings of the WG 96, Lecture Notes in Computer Science, vol. 1197, pp. 166-180(1997).
- [28] O. Duginov, Partitioning the vertex set of a bipartite graph into complete bipartite subgraphs, Discrete Mathematics and Theoretical Computer Science. vol. 16 :3, (2014)203-214.
- [29] J. Edmonds : Paths, trees, and flowers. Canadian Journal of Mathematics, 17(3) : 449-467, (1965).
- [30] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. In SACMAT’08 : Proceedings of the 13th ACM symposium on Access control models and technologies, pages 1-10, New York, NY, USA, (2008). ACM.  
doi :10.1145/1377836.1377838.

- [31] D. Eppstein, Arboricity and bipartite subgraph listing algorithms, *Information Processing Letters*, 51 (1994) 207-211
- [32] D. Eppstein, M. T. Goodrich, and J. Y. Meng, Confluent layered drawings. *Algorithmica*, 47 :439-452, (2007).  
doi :10.1007/s00453-006-0159-8.
- [33] L.Euler Solutio problematis ad geometriam situs pertinentis, *Opera Omnia* 7 (1736), p. 128-140.
- [34] H. Fleischner, E. Mujuni, D. Paulusma, and S. Szeider. Covering graphs with few complete bipartite subgraphs. *Theoretical Computer Science*, 410 :2045–2053, (2009).
- [35] D.R. Fulkerson et O.A. Gross : Incidence matrices and interval graphs, *Pacific Journal of Mathematics*, vol. 15 :pp. 835-855 (1965).
- [36] B. Ganter, R. Wille : *Formal Concept Analysis, Mathematical Foundations*. Springer, Berlin (1996)
- [37] F. Gavril : Minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on computing* Vol. 1, No. 2(1972).
- [38] F. Geerts, B. Goethals, and T. Mielikäinen. Tiling databases. In *Discovery Science*, p. 278-289. Springer, (2004).
- [39] A. Gély, L. Nourine, B. Sadi, Enumeration aspects of maximal cliques and bicliques, *Discrete Applied Mathematics* 157 (7) (2009) 1447-1459,
- [40] F. Glover : Maximum matching in a convex bipartite graph. *Naval Research Logistics*, 14, 313-316 (1967)
- [41] L. Goh, D. Rotem, Recognition of perfect elimination bipartite graphs, *Information Processing Letters*, 15 (1982) 179-182
- [42] M.C. Golumbic *Algorithmic Graph Theory and Perfect Graphs*, Elsevier, Second Edition (2004).
- [43] M.C. Golumbic, C.F. Goss, Perfect elimination and chordal bipartite graphs, *Journal of Graph Theory* 2 (1978) 155-163.
- [44] R.L. Graham, H.O. Pollak, On the addressing problem for loop switching, *Bell System Technical Journal*, 50 (1971) 2495-2519.
- [45] [https ://www.graphclasses.org/index.html](https://www.graphclasses.org/index.html)
- [46] M. Gregory, Stabilité et coloration des graphes sans  $P_5$ , HAL Id : tel-00651941 (2012).
- [47] H. Gruber and M. Holzer. Inapproximability of Nondeterministic State and Transition Complexity Assuming  $P = NP$ . In Tero Harju, Juhani Karhumäki, and Arto Lepistö, editors, *Developments in Language Theory*, volume 4588 of *Lecture Notes in Computer Science*, pages 205-216. Springer, (2007).  
doi :10.1007/978-3-540-73208-2-21.
- [48] M. Habib, Comparability invariants, *Ann. Discrete Math.* 23 (1984) 371-386.

- [49] M. Habib, R. McConnell, C. Paul, L. Viennot LexBFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234, No.1-2, 59-84 (2000). **ISSN** : 0304-3975.
- [50] A. HAJNAL. J. SURANYI, Uber die Auflosung von Graphen in vollstandige Teilgraphen, *Ann. Univ. Sci. Budapest, Edivds Sect. Math.* 1 (1958), 113-121.
- [51] P.L. Hammer, U.N. Peled et X. Sun : Difference graphs. *Discrete Applied Mathematics*, 28(1) :35-44, (1990).
- [52] R.B. Hayward, Weakly triangulated graphs, *Journal of Combinatorial Theory, series B* 39, 200-208 (1985).
- [53] D. Hermelin and G. Manoussakis, Efficient enumeration of maximal induced bicliques, *Discrete Applied Mathematics* (2020).  
<https://doi.org/10.1016/j.dam.2020.04.034>.
- [54] M.H. Heydari, L. Morales, C.O. Shields Jr., and I.H. Sudborough. Computing cross associations for attack graphs and other applications. In *Proceedings of 40th Annual Hawaii International Conference on System Sciences (HICSS 07)*, page 270, (2007).
- [55] A.J. Hoffman, A.W.J. Kolen and M. Sakarovitch, Totally-balanced and greedy matrice, *SIAM Journal of Algebraic Discrete Methods* 6 (1985) 721-730.
- [56] E. Howorka, A characterization of distance-hereditary graphs, *Quarterly Journal of Mathematics Oxford Ser. 2* (26) (1977) 417-420.
- [57] W.L. Hsu, On-line recognition of interval graphs in  $O(m + n \log n)$  time *Lecture Notes in Computer Science* 1120 (1995) 27-38
- [58] T. Jiang, B. Ravikumar, Minimal NFA Problems are hard, *SIAM Journal on Computing* 22, (1993)1117-1141.
- [59] D. Jungnickel, *Graphs, Networks and Algorithms*, Springer, Berlin Heidelberg New York Dordrecht London (2013).
- [60] R.Karp, Reducibility among combinatorial problems, complexity of computer computation 85-103 plenum pres (1972).
- [61] D. Kelly, Invariants of Finite Comparability Graphs, *Order* 3 ,p. 155-158 (1986).
- [62] T. Kloks, D. Kratsch, Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph, *Information Processing Letters* 55(1995)11-16
- [63] T.Y.Kong and P. Ribemboim, Chaining of Partially Ordered Sets. *C. R. Acad. Sci. Paris*, t. 319, Série I, p. 533-537 (1994).
- [64] N. Korte, R.H. Möhring, An incremental linear-time algorithm for recognizing interval graphs *SIAM Journal on Computing* 18 (1989) 68-81.
- [65] M.C. Lin, F. Soulignac, J.L. Szwarcfiter Arboricity, h-index and dynamic algorithms, *Theoretcal Computer Science*, 426 75-90 (2012).
- [66] W. Lipski,F.P. Preparata : Efficient algorithms for finding maximum matchings in convex bipartite graphs and related problems. *Acta Inform.* 15(4), 329-346 (1981)

- [67] A. Lubiw, Doubly lexical orderings of matrices, *SIAM Journal on Computing*, 5 (1987) 854-879.
- [68] Y. Lu, C. A. Phillips, M. A. Langston, Biclique : an R package for maximal biclique enumeration in bipartite graphs, *BMC Research Notes* (2020) 13 :88.  
doi.org/10.1186/s 13104-020-04955-0.
- [69] R. G. Michael and D. S. Johnson, *Computers and intractability : a guide to the theory of NP-completeness*, Freeman, (1979).
- [70] N. Mishra, D. Ron, R. Swaminathan : On finding large conjunctive clusters. In : *Proceedings of the 16th Annual Conference on Computational Learning Theory*, pp. 448-462 (2003)
- [71] W. Moon et L. Moser : On cliques in graphs. *Israel journal of Mathematics*, 3(1) :23-28, (1965).
- [72] G. Morel, Stabilité et coloration des graphes sans  $P_5$ . *Mathématiques générales [math.GM]*. Université de Grenoble, France (2011), NNT : 2011 GRENM0 42, tel-00651941v3.
- [73] H. Müller, On edge perfectness and classes of bipartite graphs, *Discrete Mathematics*, (149) (1996) 159-187.
- [74] A. S. Nau, G. Markowsky, M. A. Woodbury, and D. Bernard Amos. A mathematical analysis of human leukocyte antigen serology. *Math. Biosciences*, 40(3-4) :243-270, (1978).  
doi :10.1016/0025-5564(78)90088-3.
- [75] I. Nor, D. Hermelin, S. Charlat, J. Engelstadter, M. Reuter, O. Duron, and M. F. Sagot. Mod/Resc parsimony inference : Theory and application. *Information on Computing*, 213 :23-32, (2012).  
doi :10.1016/j.ic.2011.03.008.
- [76] F. Nicolai, *Strukturelle and algorithmische Aspekte distanz-erblicher Graphen und verwandter Klassen*, Dissertation thesis, Gerhard-Mercator-Universität, Duisburg (1994).
- [77] F. Nicolai, *A hypertree characterization of distance-hereditary graphs*, manuscript, GerhardMercator-Universität, Duisburg (1996)
- [78] D. Nussbaum, S. Pu, J.R. Sack, T. Uno, H. Z.Zadeh, Finding Maximum Edge Biclques in Convex Bipartite Graphs, *Algorithmica* (2012) 64 :311-325.
- [79] J. Orlin, Containment in Graph Theory : covering graphs with cliques, *Nederl. Akad. Wetensch. Indag. Math.* 39, (1977) 211-218.
- [80] R. Paige and R.E. Tajan, Three partition refinement algorithms, *SIAM Journal on Computing*, 6 (1987) 973-989.
- [81] A. Pecher, A.K. Wagler, Clique and chromatic number of circular-perfect graphs, *Proceedings of ISCO 2010 - International Symposium on Combinatorial Optimization*, *Elec. Notes in Discrete Mathematics* 36 199-206 (2010)



- [82] R. Peeters, The maximum edge biclique problem is NP-complete, *Discrete Applied Mathematics* 131, 651-654(2003).
- [83] M. Pouzet and D. Richard, *Orders : Descriptions and Roles*, *Annals of Discrete Mathematics*, 23, North-Holland (1984).
- [84] E. Prisner, Bicliques in graphs I : bounds on their number. *Combinatorica*, 20(1) :109-117 (2000).
- [85] I. Rival, *Graphs and Orders*, NATO ASI Series (C)147, D. Reidel, Dordrecht (1985).
- [86] I. Rival, *Combinatorics and Ordered Sets*, American Mathematical Society, *Contemporary Mathematics*, 57 (1986).
- [87] I. Rival, *Algorithms and orders*, NATO ASI Series (C)255, D. Reidel, Dordrecht, (1989).
- [88] D.J. Rose, R.E. Tarjan, and G.S. Lueker : Algorithmic aspects of vertex elimination on graphs, *SIAM Journal on Computing*, vol. 5, p. 266-283(1976).
- [89] D. J. Rose, Triangulated graphs and the elimination process. *Journal of mathematical analysis and application* v. 32, p. 597-609 (1970).
- [90] B. Sadi, Suite d'ensembles partiellement ordonnés, *Arima*, vol. 4, p. 66-71 (2006).
- [91] B. Schröder, *Ordered Sets*, Second Edition, Springer (2016).
- [92] J. Soares, M. Stefanos : Algorithms for maximum independent set in convex bipartite graphs, *Algorithmica* 53(1), 35-49 (2009)
- [93] J. Spinrad, Doubly lexical ordering of dense 0-1 matrices, *Information Processing Letters*. 45 (1993) 229-235.
- [94] J. Spinrad, A. Brandstadt, and L. Stewart. Bipartite permutation graphs. *Discrete Applied Mathematics*, 18 :279-292, (1987).
- [95] G. Steiner, J.S. Yeomans : A linear time algorithm for maximum matchings in convex, bipartite graphs. *Comput. Appl. Math.* 31(12), 91-96 (1996).
- [96] Talem, D., Sadi, B. Invariant Computation in a Poset. *Order* (2021). <https://doi.org/10.1007/s11083-021-09559-2>
- [97] R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13 (1984), 566-579.
- [98] P. Wolper *Introduction à la calculabilité*, Dunod 3ième édition (2006)
- [99] M. Yannakakis : Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10 :310-327, (1981).
- [100] Y. Zhang, C.A. Phillips, G.L. Rogers, E.J. Baker, E.J. Chesler, M.A. Langston, On finding bicliques in bipartite graphs : a novel algorithm and its application to the integration of diverse biological data types, *BMC Bioinform.* 15 (1) (2014) 110.