

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche  
Scientifique

Université Mouloud Mammeri de Tizi-Ouzou

Faculté des sciences

Département de Mathématiques

Mémoire de Magistère

Spécialité

Mathématique

Option

Recherche Opérationnelle et optimisation

**Thème**

**Quelques sous-classes des graphes**

**$B_1$ -orientables**

par

**Merzak Taflis**

Soutenue devant le jury composé de :

Mr. Mohamed AIDENE, Professeur, UMMTO,.....Président  
Mr. Bachir SADI, maître de conférence (A), UMMTO,.....Rapporteur  
Mr. Méziane AIDER, Professeur, USTHB (Alger),.....Examinateur  
Mr. Brahim OUKACHA, Maître de conférence(A), UMMTO,..Examinateur  
Mme. Isma BOUCHEMAKH, Professeur, USTHB (Alger),....Examinatrice

*A mes parents.*

# Remerciements

Mes premiers remerciements s'adressent tout naturellement à mon encadreur, Monsieur Sadi Bachir, qui a accepté la lourde tâche que représente la lecture critique de ce document. Je remercie aussi l'ensemble des membres du jury pour leur présence et pour l'intérêt qu'ils portent à mes travaux.

Je tiens à saluer tous ceux qui m'ont accompagné dans mes travaux et ont répondu toujours présents pour une aide ou un encouragement (parmi eux Belhadj Aziz, Sadani Idir, Talem Djamel et Hamid Karim). Je n'oublie pas bien sûr tous les professeurs qui ont suivi mon évolution depuis l'école fondamentale jusqu'à aujourd'hui.

Je termine donc en pensant à tous les membres de ma grande famille, et en particulier, mes parents, à qui, je dédie ce travail.

# Table des matières

<b>Table des matières</b>	<b>iv</b>
<b>Introduction générale</b>	<b>1</b>
<b>Quelques notions de base</b>	<b>3</b>
<b>1 Reconnaissance des graphes <math>B_1</math>-orientables</b>	<b>6</b>
1.1 Introduction et définitions . . . . .	6
1.2 Les classes d'implication de $G$ . . . . .	8
1.3 Construction des classes d'implication de $G$ . . . . .	9
1.4 Indépendance entre les classes d'implication . . . . .	9
1.5 L'algorithme d'orientation . . . . .	12
1.5.1 Principe de l'algorithme . . . . .	12
1.5.2 Algorithme ORAL . . . . .	13
1.5.3 Complexité de ORAL . . . . .	13
1.5.4 Exemple d'application . . . . .	14
<b>2 Les graphes triangulés</b>	<b>16</b>
2.1 Introduction . . . . .	16
2.2 Enoncé du problème . . . . .	17

2.3	Rappel . . . . .	17
2.4	L'algorithme 1 . . . . .	25
2.4.1	Principe de l'algorithme . . . . .	25
2.4.2	Algorithme 1 . . . . .	25
2.4.3	Preuve de l'algorithme : . . . . .	26
2.4.4	Complexité de l'algorithme : . . . . .	26
2.4.5	Exemple d'application . . . . .	27
<b>3</b>	<b>Les graphes d'arcs circulaires</b>	<b>29</b>
3.1	Introduction et définitions . . . . .	29
3.2	Caractérisation des graphes d'arcs circulaires . . .	31
3.3	Les graphes d'arcs circulaires sont des graphes $B_1$ -orientables . . . . .	35
3.3.1	Exemple d'application . . . . .	38
	<b>Conclusion et Perspectives</b>	<b>39</b>
	<b>Bibliographie</b>	<b>39</b>

# Introduction générale

La théorie des graphes est un très vaste domaine, en évolution constante tant du point de vue des recherches fondamentales que de celui des applications. Concernant la recherche fondamentale, il faut citer les travaux de P. Seymour et de ses collaborateurs qui ont démontré récemment la conjecture des Graphes Parfaits formulée par Claude Berge, amélioré la preuve du théorème dit des quatre couleurs et surtout développé la théorie des mineurs de graphes.

D'autre part, les applications sont très nombreuses. Elles justifient une recherche importante en algorithmique. L'importance des réseaux de transport et de communication, qui sont d'ailleurs de plus en plus des réseaux évolutifs (pour faire fonctionner les connections des utilisateurs à des serveurs de téléphonie mobile), explique le foisonnement des problématiques.

Dans d'autres domaines tels que la chimie, la biologie ou encore les réseaux sociaux des modèles à base de graphes sont utilisés quotidiennement en recherche.

Dans l'étude des graphes, nous nous intéressons souvent à prendre ces derniers par classes ; tous les graphes qui ont une propriété commune, nous les mettons dans la même classe.

L'utilité de ce fait est surtout dans la théorie de la complexité. Un problème qui est difficile à résoudre du point de vue algorithmique

mique dans une classe, peut être facile dans une autre contenue dans cette dernière.

Dans ce mémoire, nous nous sommes intéressés à la classe des graphes  $B_1$ -orientables.

Après la définition de cette classe par J.P.Spinrad, en 1997, et la conjecture énoncée à ce propos, par Urrutia, affirmant qu'un graphe  $B_1$ -orientable est le graphe d'intersection de sous-arbres d'un graphe planaire [17], B.Sadi a écrit un algorithme polynomial de reconnaissance de cette classe, dans [16].

Notre travail est organisé en trois chapitres :

Dans le premier chapitre, nous avons rappelé la reconnaissance de cette classe qui a été donnée dans [16].

Dans le deuxième , nous avons développé un algorithme polynomial basé sur Lex-BFS dans la reconnaissance des graphes triangulés [15]. Cet algorithme montre que les graphes triangulés sont des graphes  $B_1$ -orientables.

Dans le troisième chapitre, nous avons donné une démonstration théorique, en prouvant que les graphes d'arcs circulaires, forment une sous-classe de celle des graphes  $B_1$ -orientables.

Finalement, chacun des deux derniers chapitres présente un aspect différent de la théorie des graphes : le premier est plutôt algorithmique, alors que le second est plutôt de nature théorique.

# Quelques notions de base

Voici quelques définitions et notations que l'on considère comme étant de base pour comprendre la suite.

- Un graphe  $G$  est un couple  $(X, E)$  de deux ensembles finis et disjoints. Les éléments de l'ensemble  $X = \{x_1, x_2, \dots, x_n\}$  sont appelés les sommets de  $G$ , et  $E \subseteq X \times X$  est l'ensemble des liens entre ses sommets, ces éléments  $\{e_1, e_2, \dots, e_m\}$  de  $E$  sont appelés arêtes de  $G$ . Cette notation suppose que les arêtes (les liens) ne sont pas orientées. Si les arêtes sont orientées, on note le graphe  $G = (X, U)$ , où  $U = \{u_1, u_2, \dots, u_m\}$  l'ensemble des arcs du graphe.

Posons  $n = |X|$  et  $m = |E|$ .

- On note par  $xy$  (avec  $x, y \in X$ ), l'arête  $e \in E$ , on dit que  $x$  et  $y$  sont les extrémités de  $e$ , que  $e$  est incidente en  $x$  et en  $y$ .
- On note par  $(x, y)$  (avec  $x, y \in X$ ), l'arc  $u \in U$ , on dit que  $x$  est l'extrémité initiale et  $y$  l'extrémité terminale de  $u$ , que  $y$  est un successeur de  $x$  tandis que  $x$  est un prédécesseur de  $y$ ,  $y$  est un voisin de  $x$ , s'il est son successeur ou son prédécesseur. On dit aussi que  $u$  est incident extérieur en  $x$ , si  $x$  est extrémité initiale de  $u$  tandis que  $u$  est incident intérieur en  $y$ , si  $y$  est extrémité terminale de  $u$ .
- On note par  $\Gamma^+(x) = \{y \in X / y \text{ est un successeur de } x\}$ , l'ensemble des successeurs de  $x$ .



- On note par  $\Gamma^-(x) = \{y \in X / y \text{ est un prédécesseur de } x\}$ , l'ensemble des prédécesseurs de  $x$ .
- $\Gamma(x) = \Gamma^+(x) \cup \Gamma^-(x) =$  l'ensemble des voisins de  $x$ , noté  $N(x)$ .

Successeur et prédécesseur sont des notions orientées, par contre, la notion de voisin est non orientée.

- Deux sommets sont adjacents s'ils sont reliés par une arête. On note par  $adj(x) = \{y \in X / xy \in E\}$  l'ensemble des sommets adjacents à  $x$ .
- Deux arêtes sont adjacentes au sommet  $y$ , s'ils ont  $y$  comme extrémité commune.
- Une chaîne est une séquence alternée de sommets et d'arêtes, débutant et finissant par des sommets, telle que chaque arête est incidente aux sommets qui l'encadrent dans la séquence. Une arête ne doit pas intervenir plusieurs fois dans la séquence contrairement à un sommet.
- Le premier et le dernier sommets sont appelés extrémités de la chaîne.
- La longueur de la chaîne est égale au nombre d'arêtes qui la composent.
- Un cycle est une chaîne dont les extrémités coïncident.
- Un graphe est dit complet si toute paire de sommets de ce graphe est une arête.
- Un sous-graphe d'un graphe  $G$ , est un graphe composé de certains sommets de  $G$ , ainsi que toutes les arêtes qui relient ces

sommets.

- Une clique de  $G$  est un sous-graphe complet de  $G$ . Elle est dite maximale s'il n'existe pas une autre clique qui la contient.

# Chapitre 1

## Reconnaissance des graphes $B_1$ -orientables

### 1.1 Introduction et définitions

Dans ce chapitre, nous proposons la reconnaissance donnée dans [16] pour cette classe de graphes. Soit  $G = (X, E)$  un graphe non orienté.  $G$  est  $B_1$ -orientable s'il admet une orientation  $U$  de ses arêtes telle que :  $u, v, w \in X, (u, v) \in U$  et  $(w, v) \in U \Rightarrow uw \in E$ . On note  $G_o = (X, U)$ , le graphe orienté. Pour tout  $x \in X$ , on note  $N(x)$ , l'ensemble des voisins de  $x$ . Soit  $\Gamma$  une relation binaire sur les arcs de  $U$ , définie par :  $a, b \in N(x), a \neq b, (a, x)\Gamma(x, b) \Leftrightarrow ab \notin E$ .

Lorsque  $(a, x)\Gamma(x, b)$ , on dit que  $(a, x)$  implique  $(x, b)$ . Ceci veut dire que si  $(a, x) \in U$ , alors  $(x, b) \in U$ . Il est clair que  $(a, x)\Gamma(x, b) \Leftrightarrow (b, x)\Gamma(x, a)$ .

La recherche de l'orientation  $U$  ressemble à la recherche de l'orientation transitive dans les graphes de comparabilité (voir Gallai

[6] et Golubic [9]).

Pour une meilleure compréhension, nous donnons les remarques et exemple suivants :

**Remarque 1.1.** Dans les graphes  $B_1$ -orientables, supposons qu'on commence l'orientation des arêtes par orienter la première arête  $ab$ , de  $a$  vers  $b$ . Si la  $B_1$ -orientation n'existe pas, ceci ne veut pas dire que le graphe n'est pas  $B_1$ -orientable. Il est possible d'obtenir une  $B_1$ -orientation si l'arête  $ab$  est orientée de  $b$  vers  $a$  (ce que nous verrons plus loin).

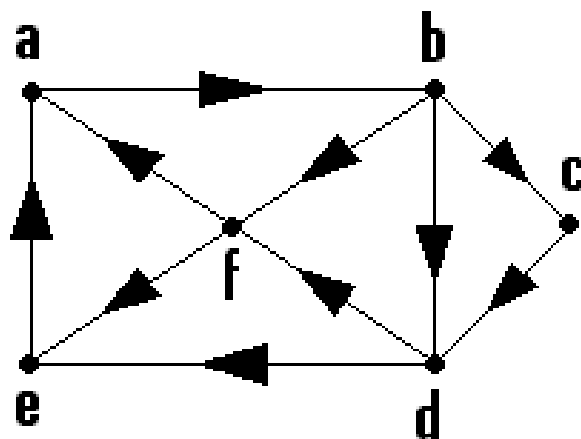


FIG. 1.1 – Après la définition des graphes  $B_1$ -orientables,  $U = \{(a, b), (b, c), (b, d), (b, f), (c, d), (d, e), (d, f), (e, a), (f, a), (f, e)\}$ , est une  $B_1$ -orientation

Dans la suite, une approche similaire à celle de Golubic est utilisée, comme dans [9]. On considère  $G$  comme un graphe symétrique et on étudie une relation entre ses arcs.

## 1.2 Les classes d'implication de $G$

Deux arcs  $(a, b)$  et  $(c, d)$  appartiennent à la même classe d'implication si et seulement si il existe une suite composée d'arcs telle que  $(a, b) = (a_0, a_1)\Gamma(a_1, a_2)\Gamma\dots\Gamma(a_k, a_{k+1}) = (c, d), k \in N$ . Une telle suite est appelée une  $\Gamma$  – chaîne de  $(a, b)$  vers  $(c, d)$ . Si  $C_{(a,b)}$  est la classe d'implication commençant par  $(a, b)$ , on note par  $C_{(a,b)}^{-1}$  la classe d'implication commençant par  $(b, a)$ . Notons par  $C_{(a,b)}^{-1}$ , la classe d'implication  $C_{(b,a)}$ ;  $\|C_{(a,b)}\|$  représente l'ensemble des arcs de  $G$  qui sont contenus dans  $C_{(a,b)}$ . Ensuite, notons par  $C_i$ , la classe d'implication numéro  $i$ , de  $G$ .

**Remarque 1.2.** Soit  $(x, y)$  un arc appartenant à  $C_{(a,b)}$ .  $(x, y) \in C_{(a,b)}$  n'implique pas, nécessairement,  $(y, x) \in C_{(a,b)}^{-1}$ .

**Autre exemple :** Si  $G$  est une clique, toute orientation des arêtes de  $G$  est une  $B_1$ -orientation.

**Définitions :**

- Une classe d'implication de  $G$  possède une *contradiction* si elle contient, simultanément, les arcs  $(x, y)$  et  $(y, x)$ .
- Une classe d'implication est dite *compatible* si elle ne contient aucune contradiction ; sinon, elle est dite *incompatible*.

### 1.3 Construction des classes d'implication de $G$

Le principe de l'algorithme qui calcule les différentes classes d'implication de  $G$  est le suivant :

- 1) Choisir, arbitrairement, un arc  $(a, b)$ , puis calculer  $C_{(a,b)}$ . Posons  $C_{(a,b)} = C_1$ .
- 2) Choisir un arc  $(x, y)$  non encore orienté, puis calculer  $C_{(x,y)}$ . Posons  $C_{(x,y)} = C_2$ .
- 3) Répéter 2), tant qu'il existe, dans  $G$ , une arête non orientée. L'ordre des classes d'implication  $C_1, C_2, \dots, C_k$  est l'ordre des arcs choisis dans 1), 2), 3).

**Lemme 1.1.** Soient  $C_1, C_2, \dots, C_k$ , les classes d'implication de  $G$ , dans le même ordre que précédemment. Alors,  $\sum_{i=1}^k \|C_i\| = |E|$ .

*Démonstration.* 1) Il est clair que  $\sum_{i=1}^k \|C_i\| \leq |E|$  car, les seules arêtes orientées sont celles de  $E$ .

2) Toute arête de  $E$  est soit, forcée par une autre arête, soit elle est la première arête d'une classe d'implication. Donc,  $E \leq \sum_{i=1}^k \|C_i\|$ .  $\square$

### 1.4 Indépendance entre les classes d'implication

Soient  $C_1, C_2, \dots, C_k$ , les différentes classes d'implication de  $G$ , produites par la construction précédente.

On dira que  $C_i$  contredit  $C_j$ ,  $i \neq j$ , si  $(a, b) \in C_i$  et  $(b, a) \in C_j$ , avec  $ab \in E$ .

Nous démontrons, dans la suite, que  $C_i$ ,  $i = 1, 2, \dots, k$ , ne peut pas contredire  $C_j$ ,  $j \neq i$ . La raison est qu'un arc est mis dans une classe d'implication seulement s'il est forcé d'avoir cette orientation dans la classe.

**Lemme 1.2.** Soient  $C_i$  et  $C_j$  deux classes d'implication de  $G$ . Alors  $C_i$  ne peut forcer un arc de  $C_j$ , pour  $j \neq i$ .

En effet, sinon, si l'arc  $(x, y) \in C_j$  et il existe un arc  $(c, d) \in C_i$  tel que  $(c, d) \Gamma (x, y)$ , alors  $(x, y) \in C_i$ .

**Lemme 1.3.** La classe d'implication  $C_{k+1}$  ne contredit pas l'orientation d'un arc  $(x, y)$  contenu dans une autre classe d'implication  $C_1, C_2, \dots, C_k$ , formée auparavant sans contradiction.

*Démonstration.* Supposons que  $C_{k+1} = C_{(a,b)}$  contredit  $C_i$ ,  $i = 1, 2, \dots, k$ , et que  $(x, y) \in C_i$ . Considérons la plus petite suite d'implications dans  $C_{k+1}$ , de  $(a, b)$  jusqu'à l'élément  $(y, x)$ . On a :

$$(a, b) \Rightarrow (b, c) \Rightarrow \dots \Rightarrow (z, y) \Rightarrow (y, x).$$

$(z, y) \Rightarrow (y, x)$  implique  $(x, y) \Rightarrow (y, z)$ . Ceci montre que  $(y, z) \in C_i$ . Comme  $(z, y)$  est dans la plus petite suite d'implications précédente, alors  $(y, z)$  ne peut appartenir à  $C_i$  (sinon, cette suite d'implications ne serait pas la plus courte).  $\square$

**Lemme 1.4.** L'existence d'une classe d'implication incompatible est indépendante de l'ordre dans lequel les classes d'impli-

cation de  $G$  ont été calculées.

*Démonstration.* Soient  $C_1, C_2, \dots, C_k$ , les classes d'implication de  $G$ , dans cet ordre. Supposons qu'il existe  $i \in 1, 2, \dots, k$  tel que  $C_i$  soit incompatible. Posons  $C_i = C_{(a,b)}$  et  $(a, b)$  un arc de  $G$ .  $C_{(a,b)}$  incompatible veut dire que l'orientation de l'arc  $(a, b)$  implique une contradiction, et l'orientation de l'arc  $(b, a)$  implique une autre contradiction. Deux raisons pour cela :

- 1)  $(a, b)$  ou  $(b, a)$  sont les orientations de l'arête  $ab$ .
- 2) L'arc  $(a, b)$  ou l'arc  $(b, a)$  est, sûrement, dans une certaine classe d'implication, dans tout ordre dans lequel on a considéré les classes d'implication de  $G$ . □

**Corollaire 1.1.** Soient  $C_1, C_2, \dots, C_k$ , les classes d'implication de  $G$ , prises dans un certain ordre.

S'il existe  $i \in \{1, 2, \dots, k\}$  tel que  $C_i$  soit incompatible, alors  $G$  n'est pas  $B_1$ -orientable.

*Démonstration.* L'incompatibilité de  $C_i$  ne dépend pas du choix de l'ordre dans lequel on considère les classes d'implication de  $G$  (voir lemme 1.4) ; cela signifie que dans n'importe quel ordre des classes d'implication, on a une classe d'implication incompatible. Donc  $G$  n'est pas  $B_1$ -orientable. □

**Théorème 1.1.**  $G = (X, E)$  est  $B_1$ -orientable si et seulement si toutes les classes d'implication, prises dans un certain ordre sont compatibles.



*Démonstration.*  $\implies$ ) Supposons qu'il existe une classe d'implication  $C_{(a,x)}$  incompatible, commençant par  $(a, x)$ . Donc il existe un arc  $(b, y) \in C_{(a,x)}$  tel que  $(a, x)\Gamma(b, y)$  et  $(a, x)\Gamma(y, b)$ , et il existe un arc  $(c, z) \in C_{(a,x)}^{-1}$  tel que  $(x, a)\Gamma(c, z)$  et  $(x, a)\Gamma(z, c)$ ; dans les deux cas, il ya une impossibilité. Comme l'arête  $ax$  possède seulement deux orientations  $a \rightarrow x$  ou  $x \rightarrow a$ , alors  $G$  ne peut admettre une  $B_1$ -orientation (voir lemme 1.4 et corollaire).

$\impliedby$ ) Soient  $C_1, C_2, \dots, C_k$ , les classes d'implication de  $G$ . Tous les arêtes de  $C_i, i = 1, 2, \dots, k$ , sont orientés sans contradiction. Comme  $\sum_{i=1}^k \|C_i\| = |E|$ , alors cette orientation est une  $B_1$ -orientation.  $\square$

## 1.5 L'algorithme d'orientation

### 1.5.1 Principe de l'algorithme

On choisit arbitrairement, une arête  $e_i = x_i y_i$  dans  $E_i = E$ , et on construit une classe d'implications  $C_i$ , commençant avec  $(x_i, y_i)$ , tant que  $C_i$  n'a pas de contradiction. Si, dans  $C_i$ , il existe une contradiction, on abandonne  $C_i$  et on considère  $C_i^{-1}$  commençant avec  $(y_i, x_i)$ , tant que  $C_i^{-1}$  n'a pas de contradiction.

- Si  $C_i$  et  $C_i^{-1}$  ont une contradiction, le graphe  $G$  n'est pas  $B_1$ -orientable.
- Si  $C_i$  (resp.  $C_i^{-1}$ ) n'a pas de contradiction, faire :  $E_{i+1} = E_i - C_i$  (resp.  $E_{i+1} = E_i - C_i^{-1}$ ), et recommencer dans  $E_{i+1}$ .

Si  $\forall i = 1, 2, \dots, k$ ,  $C_i$  ou  $C_i^{-1}$  n'a pas de contradiction, le graphe  $G$  est  $B_1$ -orientable.

### 1.5.2 Algorithme ORAL

Soient  $C_1, C_2, \dots, C_k$ , les classes d'implication de  $G$ .

- (0) Initialiser  $i = 1$ ;  $E_1 = E$ .
- (1) Choisir une arête  $e_i = x_i y_i \in E_i$ .
- (2) Soit  $C_i$  la classe d'implication de  $E_i$ , commençant avec  $(x_i, y_i)$ 
  - (a) Si  $C_i$  contient une contradiction, faire : considérer  $C_i^{-1}$  de  $E_i$ , commençant avec  $(y_i, x_i)$ .  
-Si  $C_i^{-1}$  contient une contradiction, faire :STOP ;  $G$  n'est pas  $B_1$ -orientable  
Sinon, faire :  $C_i \leftarrow C_i^{-1}$
  - (b) Sinon, faire :
- (3)  $E_{i+1} = E_i - C_i$
- (4) (4.a) Si  $E_{i+1} = \emptyset$ ;  $k = i$ , STOP ;  $G$  est  $B_1$ -orientable  
(4.b) Sinon, faire :  $i \leftarrow i + 1$  ; aller en (1).

### 1.5.3 Complexité de ORAL

Pour chaque sommet  $x \in X$ , **ORAL** visite tous les voisins de  $x$  puis les teste, deux à deux, s'ils ne sont pas reliés par une arête. Alors, la complexité est  $n.C_{\Delta}^2 = n.\Delta(\Delta - 1)/2 \approx O(n.\Delta^2)$ , où  $n = |X|$  et  $\Delta$  est le degré maximum de  $G$ .

### 1.5.4 Exemple d'application

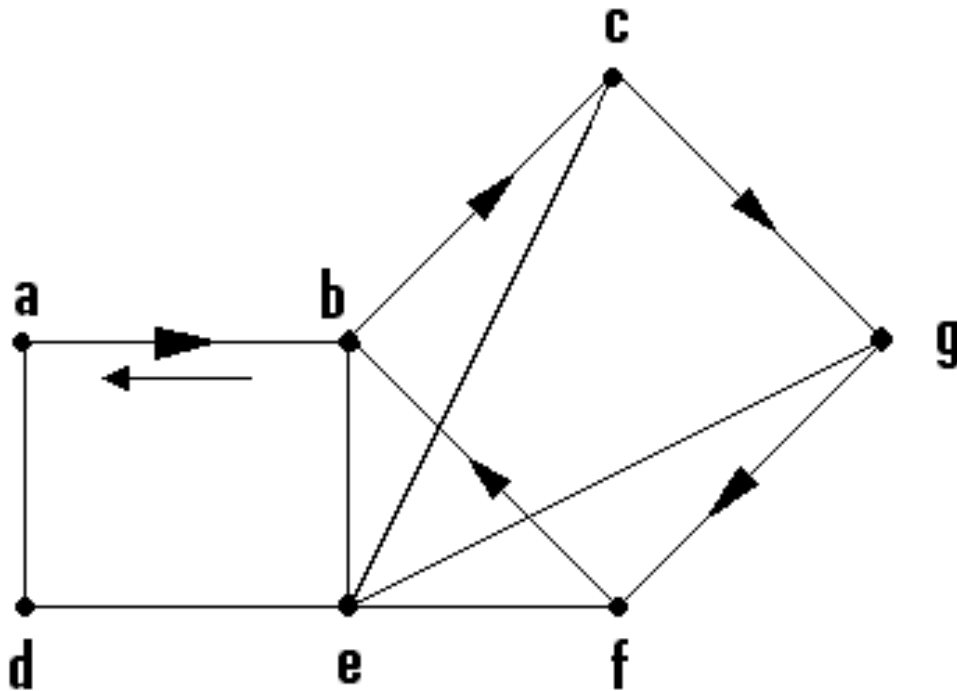


FIG. 1.2 –

Si  $(a, b) \in U$  : (Voir fig 1.2)

$(a, b)\Gamma(b, c)\Gamma(c, g)\Gamma(g, f)\Gamma(f, b)\Gamma(b, a)\Gamma\dots$

On remarque que  $C_{(a,b)}$  possède une *contradiction*, car elle contient, simultanément, les arcs  $(a, b)$  et  $(b, a)$ , donc  $U$  n'existe pas.

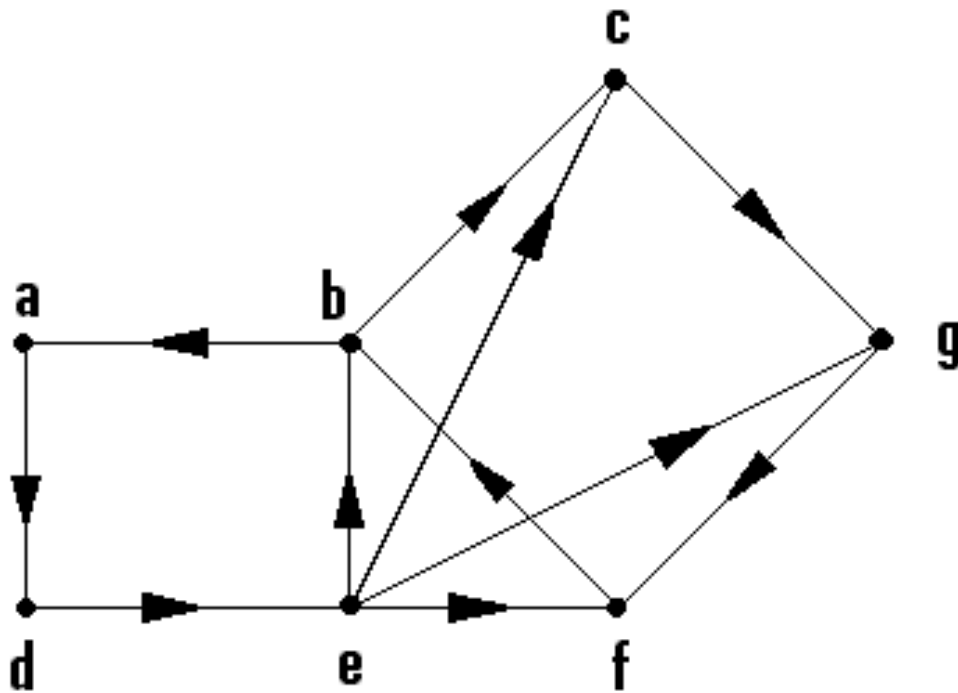


FIG. 1.3 –

Si  $(b, a) \in U$  : (Voir fig 1.3)

$$C_{(b,a)} = ((b, a), (a, d), (d, e), (e, b), (e, f), (e, g), (e, c)).$$

$$C_{(b,c)} = ((b, c), (c, g), (g, f), (f, b))$$

$C_{(b,a)} \cup C_{(b,c)}$  comporte l'ensemble des arcs de  $G$ .

La  $B_1$ -orientation  $U$  est trouvée.

# Chapitre 2

## Les graphes triangulés

### 2.1 Introduction

Dans ce chapitre et après avoir rappelé la définition des graphes  $B_1$ -orientables [17], nous avons énoncé le problème sous forme d'un théorème disant que les graphes triangulés sont des graphes  $B_1$ -orientables. Pour prouver ce théorème, nous avons développé un algorithme polynomial qui fait des graphes triangulés des graphes  $B_1$ -orientables.

cet algorithme est constitué de deux parties. La première, c'est Lex-BFS, pour cela, nous avons cité les travaux donnés par certains auteurs (voir [9]), sur lesquels ce dernier (Lex-BFS) a été construit, dont le principe est de donner un ordre d'élimination simplicial si et seulement si le graphe est triangulé. La deuxième partie de cet algorithme est un fragment que nous avons ajouté à Lex-BFS, dont le rôle est de donner au graphe une orientation liée à l'ordre d'élimination simplicial des sommets. Cette orien-

tation est une  $B_1$ -orientation pour le graphe.

Dans la suite, nous allons donner plus de détails pour comprendre mieux le fonctionnement de cet algorithme.

## 2.2 Enoncé du problème

**Théorème 2.1.** Tout graphe triangulé est  $B_1$ -orientable.

La démonstration de ce théorème se fait par l'algorithme qu'on propose dans ce travail.

Cet algorithme est basé sur la reconnaissance des graphes triangulés (Lex-BFS).

Mais avant de l'écrire, nous rappelons simplement quelques éléments.

## 2.3 Rappel

Les graphes triangulés sont apparus sous différents noms au début des années 60. Dans la littérature anglo-saxonne, nous trouvons généralement l'appellation "*chordal graphs*" car dans tout cycle, il existe une *corde* (arête joignant deux sommets non consécutifs du cycle). Cette classe de graphes a été très étudiée, elle intervient dans différents domaines : le calcul matriciel et les bases de données, par exemple. De plus, c'est l'une des premières classes découvertes de graphes parfaits au sens de Berge [2].

**Définition 2.1.** (Berge 1960) [2] On note  $C_k$ ,  $k \geq 4$ , le cycle de longueur  $\geq 4$ , sans corde (arête reliant deux sommets non-

consécutifs d'un cycle).

Un graphe  $G = (X, E)$  est triangulé s'il ne contient pas de  $C_k$ .

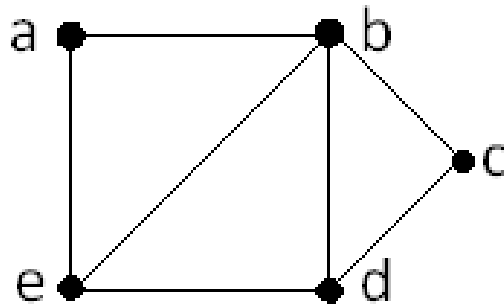


FIG. 2.1 – Ce graphe est triangulé, il ne possède aucun cycle sans corde de longueur  $\geq 4$

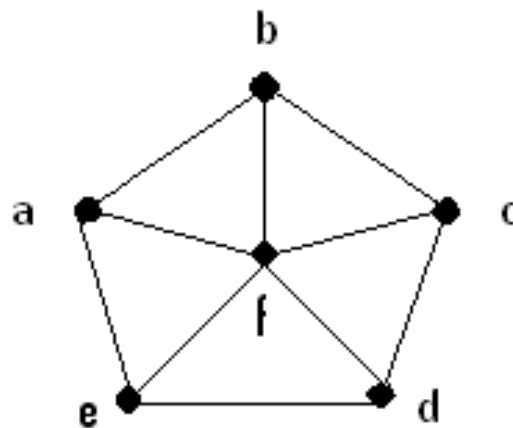


FIG. 2.2 – Ce graphe n'est pas triangulé, car il possède un cycle sans corde à 5 sommets :  $(a, b, c, d, e, a)$ .

**Notation 2.1.** Soit  $G = (X, E)$  un graphe et  $X' \subseteq X$  un sous-ensemble de sommets. Alors  $G[X']$  désigne le sous-graphe induit par les sommets de  $X'$ .

Le premier enseignement issu de la définition des graphes triangulés est que cette classe est une classe de graphes héréditaires. C'est ce qu'affirme le lemme suivant :

**Lemme 2.1.** Soit  $G = (X, E)$  un graphe triangulé. Tout sous-graphe  $H = G[X']$  de  $G$ , où  $X' \subseteq X$ , est un graphe triangulé.

En effet, la suppression d'un quelconque ensemble de sommets  $S$  du graphe  $G$  ne peut en aucun cas créer de  $C_k$ ,  $k \geq 4$ .

**Définition 2.2.** Un sommet  $x$  est simplicial si son voisinage  $N(x)$  est une clique.

**Définition 2.3.** . Un ordre  $\sigma = [x_1, \dots, x_i, \dots, x_n]$  est un ordre d'élimination simplicial si et seulement si pour tout  $1 \leq i \leq n$ , le sommet  $x_i$  est simplicial dans le graphe  $G_i = G[x_i, \dots, x_n]$ .

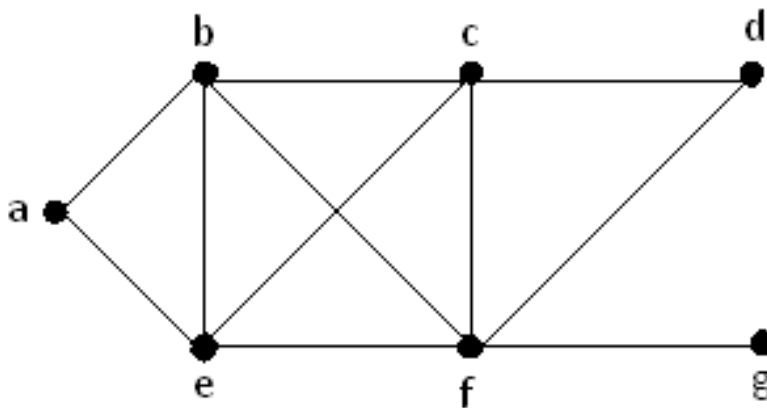


FIG. 2.3 –  $\sigma = [a, g, e, d, b, f, c]$  est un ordre d'élimination simplicial

**Notation 2.2.** Soit  $\sigma = [x_1, \dots, x_i, \dots, x_n]$  un ordre. Dans la suite,  $X_i$  désignera toujours les sommets  $\{x_i, \dots, x_n\}$ . Le sous-graphe induit par les sommets de  $X_i$  sera noté  $G_i = G[x_i, \dots, x_n]$ .



Dirac, en 1961, a montré la propriété fondamentale de l'existence d'un ordre d'élimination simplicial dans les graphes triangulés.

**Lemme 2.2.** (Dirac 1961) [4] Tout graphe triangulé  $G = (X, E)$  possède un sommet simplicial. Si  $G$  n'est pas un graphe complet, alors il possède au moins deux sommets simpliciaux non adjacents.

Fulkerson et Gross, en 1965, ont caractérisé les graphes triangulés de la manière suivante :

**Théorème 2.2.** (Fulkerson et Gross 1965) [5] Un graphe est triangulé si et seulement si il existe un ordre d'élimination simplicial.

*Démonstration.*  $\implies$ ) Cette implication découle de l'application récursive du lemme précédent : Dans un graphe triangulé, on trouve un sommet simplicial, on le supprime, le graphe obtenu reste un graphe triangulé.....

( $\impliedby$  Si un graphe n'est pas triangulé, alors il possède un cycle de plus de 3 sommets sans corde, et aucun sommet de ce cycle ne peut être éliminé simplicialement.  $\square$ )

C'est toujours cette caractérisation (existence d'un ordre d'élimination simplicial) qui a mené à l'algorithme linéaire de reconnaissance des graphes triangulés (Lex-BFS) donné par Rose, Tarjan et Leuker en 1976 [15]. Il construit un ordre d'élimination simplicial à l'envers : il détermine d'abord le dernier sommet  $x_n$

de l'ordre et termine par le premier sommet  $x_1$ . Pour comprendre ce fonctionnement, nous devons utiliser un peu plus finement les conséquences de la propriété de Dirac (lemme 2.2) :

**Lemme 2.3.** Soit  $G = (X, E)$  un graphe triangulé. Pour tout sommet  $x \in X$ , il existe un ordre d'élimination simplicial  $\sigma$  terminant par  $x$ .

*Démonstration.* Soit  $\gamma$  un ordre d'élimination simplicial de  $G$  et  $\gamma(x) = i$ . A partir de  $\gamma$ , nous allons construire un ordre d'élimination  $\sigma$  tel que  $\sigma(n) = x$  :

Si  $G$  est un graphe complet, il suffit d'échanger  $x$  et  $\gamma(n)$  pour obtenir l'ordre d'élimination simplicial  $\sigma$ .

Si  $i = n$ , alors  $\sigma = \gamma$ . Supposons que  $i \neq n$ . Alors pour tout  $1 \leq j < i$ , prenons  $\sigma(j) = \gamma(j)$ . Par hypothèse  $x$  est simplicial dans le sous-graphe  $G_i$ . Or le lemme 2.2 montre qu'il existe dans  $G_i$  un sommet simplicial  $y \neq x$ . Donc on peut choisir  $\sigma(i) = y$ .

Si  $y$  n'est pas voisin de  $x$  dans  $G_i$ , le voisinage de  $x$  reste inchangé et reste donc une clique. Sinon  $y$  a été supprimé du voisinage de  $x$ . Mais tout sous-graphe d'une clique est une clique. Donc  $x$  reste simplicial dans le sous-graphe  $G_{i+1}$ .  $\square$

Donc pour tout  $i \leq k \leq n$ , nous pouvons appliquer l'argument précédent. Ceci montre que  $\sigma$  peut être construit tel que  $\sigma(n) = x$ .

On peut donc imaginer un algorithme de reconnaissance des graphes triangulés qui commence par choisir le dernier sommet de l'ordre d'élimination simplicial. Il faut juste trouver une règle qui autorise à poursuivre cette stratégie à chaque étape sur le sous-graphe induit par les sommets non numérotés. C'est ce que fait Lex-BFS.

Les sommets sont numérotés dans l'ordre inverse de leur visite. Chaque sommet  $x$  possède une marque,  $marque(x)$ , qui correspond à la liste ordonnée de ses voisins numérotés. A chaque fois qu'un nouveau sommet  $x$  est numéroté, son numéro est ajouté à la fin de  $marque(y)$  pour tout voisin non numéroté  $y$  de  $x$ . Le nouveau sommet choisi est toujours un sommet ayant une marque maximum dans l'ordre lexicographique.

**Algorithme :** Lex-BFS (RTL 1976) [15]

**Donnée :** Un graphe  $G = (X, E)$

**Résultat :** Un ordre  $\sigma$  des sommets de  $G$

(1) Pour chaque sommet  $x \in X$  faire :

$marque(x) \leftarrow \emptyset$

(2) Pour  $i = n$  à 1 faire :

(2-1) Choisir un sommet  $x$  non numéroté de marque maximum dans l'ordre lexicographique ;

$\sigma(i) \leftarrow x$

(2-2) Pour chaque voisin non numéroté  $y$  de  $x$  faire :

$marque(y) \leftarrow marque(y) \cup \{i\}$

**Théorème 2.3.** (RTL 1976) [15] Lex-BFS calcule un ordre d'élimination simplicial si et seulement si le graphe  $G$  est triangulé. Sa complexité est  $O(n + m)$ , où  $n = |X|$  et  $m = |E|$ .

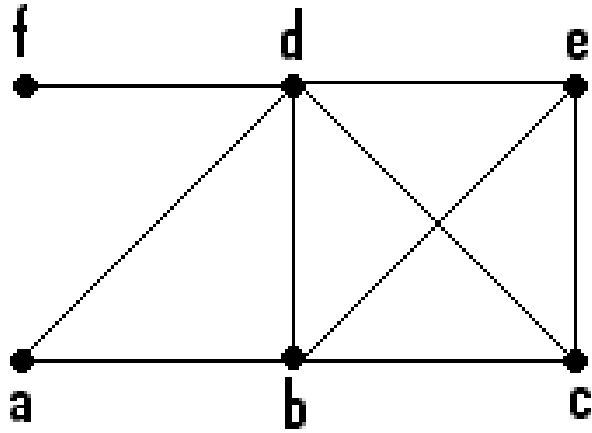


FIG. 2.4 –

$\sigma$	6	5	4	3	2	1
a	$\emptyset$	6	6,5	6,5	$\bullet 6, 5$	
b	$\emptyset$	$\bullet 6$				
c	$\emptyset$	6	$\bullet 6, 5$			
d	$\bullet \emptyset$					
e	$\emptyset$	6	6,5	$\bullet 6, 5, 4$		
f	$\emptyset$	6	6	6	6	$\bullet 6$

Un graphe triangulé  $G$  et la table d'état des marques de chaque sommet aux différentes étapes de Lex-BFS. L'ordre obtenu est  $\sigma = [f, a, e, c, b, d]$  (il est calculé à l'envers). Nous pouvons vérifier que  $\sigma$  est un ordre d'élimination simplicial : par exemple  $e$  est simplicial dans le sous-graphe  $G[e, c, b, d]$ .

## 2.4 L'algorithme 1

### 2.4.1 Principe de l'algorithme

L'algorithme comporte deux parties. La première c'est Lex-BFS dont le rôle est de donner un ordre d'élimination simplicial  $\sigma = [x_1, \dots, x_i, \dots, x_n]$ , où  $x_i = \sigma(i)$  est un sommet simplicial pour le graphe  $G_i = G[x_i, \dots, x_n]$  pour tout  $1 \leq i \leq n$ , alors que pour la deuxième partie, l'idée principale est d'afficher tous les arcs de la forme  $(x_i, y)$ , où  $x_i$  est simplicial dans le graphe  $G_i$  avec  $y \in \text{adj}(x_i)$  pour tout  $1 \leq i \leq n - 1$ .

### 2.4.2 Algorithme 1

**Donnée :** Un graphe  $G = (X, E)$  triangulé non orienté

**Résultat :** Une  $B_1$ -orientation  $U$  de  $G$

(0)  $U = \emptyset$

(1) Pour chaque sommet  $x \in X$  faire :

$\text{marque}(x) \leftarrow \emptyset$

(2) Pour  $i = n$  à 1 faire :

(2-1) Choisir un sommet  $x$  non numéroté de marque maximum dans l'ordre lexicographique ;

$\sigma(i) \leftarrow x$

(2-2) Pour chaque voisin non numéroté  $y$  de  $x$  faire :

$\text{marque}(y) \leftarrow \text{marque}(y) \cup \{i\}$

(3) Pour  $i = 1$  à  $n - 1$  faire :

Si  $\{j, \text{ tel que } \sigma(j) \in \text{adj}(\sigma(i)) \text{ et } j > i\} \neq \emptyset$  alors

$$U = U \cup \{(\sigma(j), \sigma(i))\}$$

### 2.4.3 Preuve de l'algorithme :

On suppose que l'orientation trouvée n'est pas une  $B_1$ -orientation, donc  $\exists(x, y) \in U$  et  $(z, y) \in U$  et  $xz \notin E$ .

Les arêtes  $xy, zy$  sont orientées à l'étape où  $y$  était un sommet simplicial, par conséquent, chaque couple dans le voisinage de  $y$  forme une arête, y compris  $xz$ .

### 2.4.4 Complexité de l'algorithme :

Dans le fragment que nous avons ajouté, pour tout  $i = 1$  à  $n - 1$ ; l'algorithme visite tous les voisins  $\sigma(j)$  de  $\sigma(i)$ , puis les teste, un à un, si  $j > i$ , puis il teste si  $\{j, \text{tel que } \sigma(j) \in \text{adj}(\sigma(i)) \text{ et } j > i\} \neq \emptyset$ , suivi par  $m$  insertion d'arcs  $(\sigma(j), \sigma(i))$ , donc le temps demandé est  $O(n(n - 1) + m)$ , en ajoutant le nombre d'opérations maximum pour l'union qui est  $n - 1$ , la complexité de ce fragment est  $O(n(n - 1) + m + (n - 1)) = O(n^2 + m)$ , avec la complexité de Lex-BFS qui est  $O(n + m)$ , la complexité de l'algorithme est  $O(n^2 + 2m) \simeq O(n^2)$ .

### 2.4.5 Exemple d'application

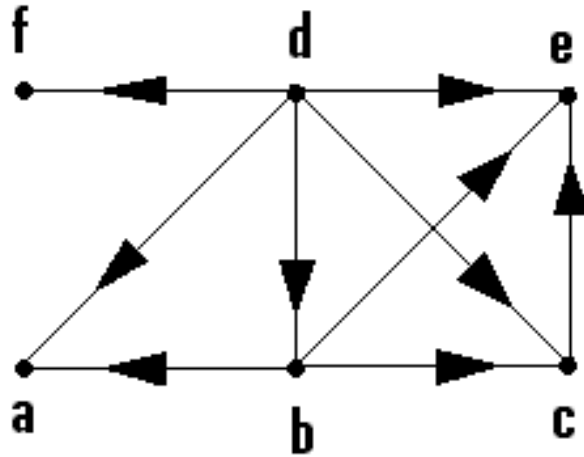


FIG. 2.5 –

Nous appliquons l'algorithme 1 sur le graphe de l'exemple précédent ; Comme Lex-BFS nous a donné un ordre d'élimination simplicial  $\sigma = [f, a, e, c, b, d]$ , alors il suffit d'appliquer la dernière partie de l'algorithme :

**(3)** Pour  $i = 1$  à  $n - 1$  faire :

Si  $\{j, \text{ tel que } \sigma(j) \in \text{adj}(\sigma(i)) \text{ et } j > i\} \neq \emptyset$  alors  $U = U \cup \{(\sigma(j), \sigma(i))\}$



$$U = \emptyset$$

$$\mathbf{i=1} : \sigma(1) \equiv f; \{(\sigma(j), f), \text{ avec } \sigma(j) \in \text{adj}(f), \text{ et } j > 1\} = \{(d, f)\}$$

$$U = \emptyset \cup \{(d, f)\} = \{(d, f)\}$$

$$\mathbf{i=2} : \sigma(2) \equiv a; \{(\sigma(j), a), \text{ avec } \sigma(j) \in \text{adj}(a), \text{ et } j > 2\} = \{(b, a), (d, a)\}$$

$$U = U \cup \{(b, a), (d, a)\} = \{(d, f), (b, a), (d, a)\}$$

$$\mathbf{i=3} : \sigma(3) \equiv e; \{(\sigma(j), e), \text{ avec } \sigma(j) \in \text{adj}(e), \text{ et } j > 3\} = \{(c, e), (b, e), (d, e)\}$$

$$U = U \cup \{(c, e), (b, e), (d, e)\} = \{(d, f), (b, a), (d, a), (c, e), (b, e), (d, e)\}$$

$$\mathbf{i=4} : \sigma(4) \equiv c; \{(\sigma(j), c), \text{ avec } \sigma(j) \in \text{adj}(c), \text{ et } j > 4\} = \{(b, c), (d, c)\}$$

$$U = U \cup \{(b, c), (d, c)\} = \{(d, f), (b, a), (d, a), (c, e), (b, e), (d, e), (b, c), (d, c)\}$$

$$\mathbf{i=5} : \sigma(5) \equiv b; \{(\sigma(j), b), \text{ avec } \sigma(j) \in \text{adj}(b), \text{ et } j > 5\} = \{(d, b)\}$$

$$U = U \cup \{(d, b)\} = \{(d, f), (b, a), (d, a), (c, e), (b, e), (d, e), (b, c), (d, c), (d, b)\}$$

La  $B_1$ -orientation cherchée est :

$$U = \{(d, f), (b, a), (d, a), (c, e), (b, e), (d, e), (b, c), (d, c), (d, b)\}$$

# Chapitre 3

## Les graphes d'arcs circulaires

### 3.1 Introduction et définitions

Ce chapitre est consacré à la classe des graphes d'arcs circulaires. Tucker en 1970 a donné la première caractérisation pour cette classe de graphes. A partir de cette caractérisation, nous avons obtenu un résultat que nous avons proposé sous forme d'un théorème en montrant que les graphes d'arcs circulaires sont des graphes  $B_1$ -orientables.

Les premiers papiers dans lesquels, les graphes d'arcs circulaires sont apparus, ont été écrits en 1964 par Hadwiger, Debrunner et Klee [10]. Tucker, dans sa thèse de Ph.D. en 1969 [19], et aussi dans une série d'articles en 1970 [20,21,22,23,24], a donné quelques propriétés fondamentales pour cette classe de graphes. Depuis leur apparition, ces graphes ont été très étudiés. Aujourd'hui, ils interviennent dans différents domaines tels que la génétique et la circulation routière.

**Définition 3.1.** Considérons, autour d'un cercle  $C$ , une famille  $(A_1, A_2, \dots, A_n)$  d'arcs.

On peut former un graphe  $G$ , dont les sommets  $x_1, x_2, \dots, x_n$  représentent, respectivement, les arcs  $A_1, A_2, \dots, A_n$ . Deux sommets étant adjacents si et seulement si les arcs correspondants s'intersectent.

Un tel graphe est dit graphe d'arcs circulaires.

**Remarque 3.1.** En général, les graphes d'arcs circulaires ne sont pas parfaits, par exemple : les cycles impairs de longueur  $> 3$  et sans corde (voir figure 3.1).

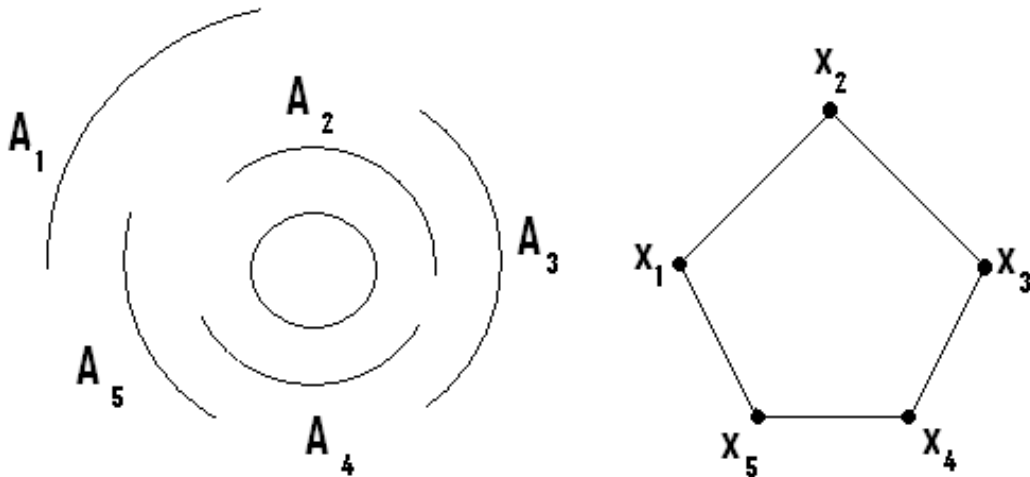


FIG. 3.1 – une représentation d'arcs circulaires du graphe non parfait  $C_5$ .

**Notation 3.1.** Il est prouvé qu'il n'y a aucune différence entre le choix d'arcs ouverts et fermés. Par la suite, nous adopterons la convention d'arcs ouverts. En plus, nous pouvons supposer sans perte de généralité, qu'aucune paire d'arcs n'a une extrémité commune.

### 3.2 Caractérisation des graphes d'arcs circulaires

**Théorème 3.1.** (Tucker 1970) [19] . Un graphe  $G = (X, E)$  non orienté est un graphe d'arcs circulaires si et seulement si, il existe un ordre circulaire  $x_1, x_2, \dots, x_n$  de ses sommets tel que pour tout  $i < j$ , si l'arête  $x_i x_j \in E$ , alors :

$$(2) \begin{cases} \text{Soit, } x_{i+1}, x_{i+2}, \dots, x_j \in \text{adj}(x_i) & (2-1); \\ \text{ou bien, } x_{j+1}, x_{j+2}, \dots, x_i \in \text{adj}(x_j) & (2-2). \end{cases}$$

*Démonstration.*  $\implies$ ) Sur la représentation d'arcs circulaires de  $G$ , imaginons une aiguille, qui commence à tourner d'un point arbitraire du cercle  $C$ , dans le sens des aiguilles d'une montre. (voir figure 3.2)

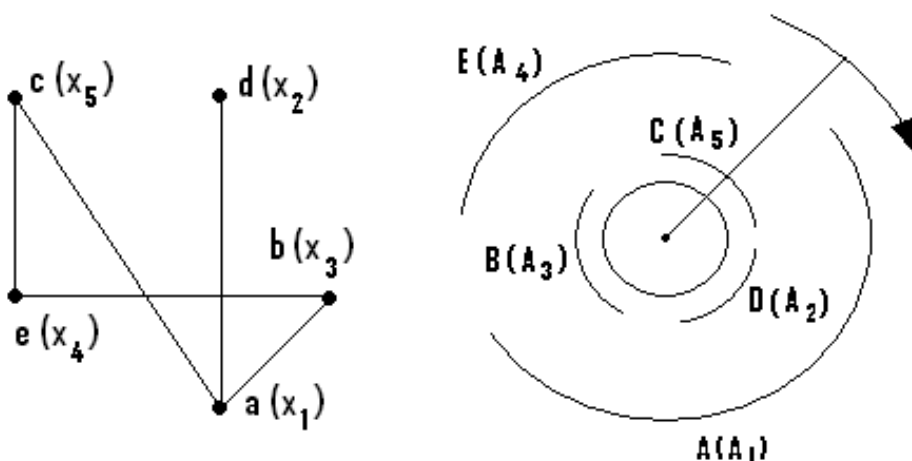


FIG. 3.2 – Les arcs  $A, B, C, D, E$  sont respectivement marqués :  $A_1, A_3, A_5, A_2, A_4$  et les sommets  $a, b, c, d, e$  sont respectivement indexés  $x_1, x_3, x_5, x_2, x_4$ .

A chaque fois que cette aiguille rencontre la première extrémité d'un nouvel arc, nous lui donnons la marque  $A_i$ , et nous indexons par  $x_i$ , le sommet correspondant à cet arc.

Dès que l'aiguille fait son premier tour autour du cercle  $C$ , on aura indexé tous les sommets de  $G$ .

De cette manière, on donne un ordre circulaire  $x_1, x_2, \dots, x_n$  à tous les sommets de  $G$ .

Clairement,  $x_i$  est adjacent à  $x_j$  si et seulement si leurs arcs correspondants s'intersectent, avec  $1 \leq i < j \leq n$ .

Commençons maintenant (après avoir marqué tous les arcs de la représentation d'arcs circulaires), à tourner l'aiguille à partir d'un point de départ  $m$  du cercle  $C$ , qui est hors de  $A_i \cup A_j$ , et toujours dans le sens des aiguilles d'une montre, avec  $A_i$  et  $A_j$  sont deux arcs quelconques marqués de la représentation d'arcs circulaires).

Dans cette situation, nous avons deux cas :

**Notation :** Dans ce qui vient,  $s_i$  et  $t_i$  (respectivement  $s_j$  et  $t_j$ ) signifient, l'extrémité initiale et l'extrémité terminale de l'arc  $A_i$  (respectivement de l'arc  $A_j$ ), avec  $1 \leq i < j \leq n$ .

**Premier cas :**  $A_i$  se situe avant  $A_j$  (Voir figure 3.3)

A partir du moment où l'aiguille rencontre l'extrémité initiale  $s_j$  de l'arc  $A_j$ , c'est sûrement qu'elle n'a pas encore rencontré l'extrémité terminale  $t_i$  de l'arc  $A_i$ , (car  $A_i$  et  $A_j$  s'intersectent). En plus, à ce moment là, c'est sûrement que cette aiguille avait rencontré tous les arcs  $A_{i+1}, A_{i+2}, \dots, A_{j-1}$  (car les extrémités ini-

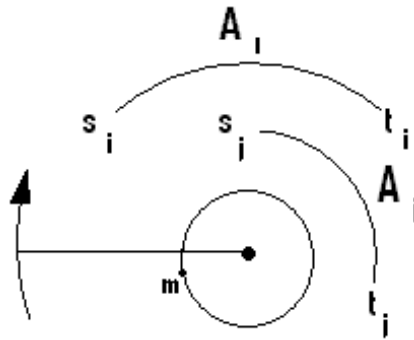


FIG. 3.3 –

tiales de ces arcs se situent tous après celle de  $A_i$  et avant celle de  $A_j$  ).

Donc pour tout  $h \in \{i+1, i+2, \dots, j-1\}$ ,  $A_h$  et  $A_i$  s'intersectent, par conséquent,  $x_h$  et  $x_i$  sont adjacents  $\forall h \in \{i+1, i+2, \dots, j\}$ , d'où (2-1).

**Deuxième cas :**  $A_i$  se situe après  $A_j$  (Voir figure 3.4)

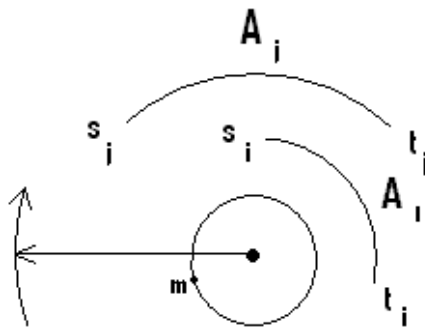


FIG. 3.4 –

A partir du moment où l'aiguille rencontre l'extrémité initiale  $s_i$  de l'arc  $A_i$ , c'est sûrement qu'elle n'a pas rencontré l'extrémité terminale  $t_j$  de l'arc  $A_j$ , (car  $A_i$  et  $A_j$  s'intersectent).

En plus, à ce moment là, c'est sûrement que cette aiguille avait rencontré tous les arcs  $A_{j+1}, A_{j+2}, \dots, A_{i-1}$  (car les extrémités initiales de ces arcs se situent tous après celle de  $A_j$  et avant celle de  $A_i$ ). Donc pour tout  $h \in \{j+1, j+2, \dots, i-1\}$ ,  $A_h$  et  $A_j$  s'intersectent, par conséquent,  $x_h$  et  $x_j$  sont adjacents  $\forall h \in \{j+1, j+2, \dots, i\}$ , d'où (2-2).

$\Leftarrow$ ) Indéxer les sommets de  $G$  comme exigé dans **(2)**. Notre but est de construire une représentation d'arcs circulaires pour  $G$ .

Sur le cercle  $C$ ,  $p_k$  représente l'abscisse curviligne correspondante au sommet  $x_k$ , où  $1 \leq k \leq n$ .

On note que l'ordre croissant de ces abscisses est dans le sens des aiguilles d'une montre.

Pour chaque sommet  $x_i$ , soit  $x_{m_i}$  le premier sommet dans la séquence cyclique  $x_{i+1}, x_{i+2}, \dots, x_i$  qui n'est pas adjacent à  $x_i$ .

Tirer un arc ouvert  $A_i$  dans le sens des aiguilles d'une montre, de  $p_i$  à  $p_{m_i}$ .

Par construction,  $A_i$  croise  $A_j$ , ( $i \neq j$ ) si et seulement si  $\begin{cases} , p_j \in A_i & ; \\ ou, p_i \in A_j & . \end{cases}$

Mais aussi :  $\begin{cases} p_j \in A_i \Leftrightarrow x_{i+1}, x_{i+2}, \dots, x_j \in adj(x_i) & ; \\ , p_i \in A_j \Leftrightarrow x_{j+1}, x_{j+2}, \dots, x_i \in adj(x_j) & . \end{cases}$

Par conséquent, et avec **(2)**, on a :  $\begin{cases} p_j \in A_i & ; \\ ou, p_i \in A_j & . \end{cases} \Leftrightarrow x_i x_j \in E \Leftrightarrow A_i \cap A_j \neq \emptyset.$

□

### 3.3 Les graphes d'arcs circulaires sont des graphes $B_1$ -orientables

Dans cette section, nous avons obtenu un résultat que nous proposons sous forme d'un théorème inspiré de celui de Tucker. Dans ce résultat, nous avons montré que les graphes d'arcs circulaires forment une sous-classe de celle des graphes  $B_1$ -orientables.

**Théorème 3.2.** Pour tout graphe  $G$  d'arcs circulaires non orienté, il existe un ordre circulaire  $x_1, x_2, \dots, x_n$  de sommets de  $G$ , et une orientation  $U$  tels que pour tout  $i < j$ ;

$$U = \begin{cases} (x_i, x_j), \text{ si } x_{i+1}, x_{i+2}, \dots, x_j \in \text{adj}(x_i); \\ (x_j, x_i), \text{ si } x_{j+1}, x_{j+2}, \dots, x_i \in \text{adj}(x_j). \end{cases}$$

Cette orientation  $U$  est une  $B_1$ -orientation pour  $G$ .

*Démonstration.* **L'existence de l'ordre circulaire :**

On peut construire un ordre  $x_1, x_2, \dots, x_n$ , comme il est exigé dans le théorème précédent (Tucker).

D'après le théorème précédent, nous savons que l'existence d'une arête quelconque  $x_i x_j$  où  $i < j$ , obéit aux deux cas suivants :

Premier cas :  $x_i$  et  $x_j$  sont dans la suite  $x_{i+1}, x_{i+2}, \dots, x_j$ , avec  $x_{i+1}, x_{i+2}, \dots, x_j \in \text{adj}(x_i)$ .

Deuxième cas :  $x_i$  et  $x_j$  sont dans la suite  $x_{j+1}, x_{j+2}, \dots, x_i$ , avec  $x_{j+1}, x_{j+2}, \dots, x_i \in \text{adj}(x_j)$ .

Si l'existence de l'arête  $x_i x_j$  suit le premier cas, alors pour la



construction de  $U$ , il suffit d'orienter cette arête du sommet  $x_i$  vers le sommet  $x_j$ , sinon, on l'orienter de  $x_j$  vers  $x_i$ .

De cette manière, on aura construit  $U$ .

D'où l'existence de l'ordre circulaire et de l'orientation de  $U$ .

### La $B_1$ -orientation de $U$ :

Soit  $(x_i, x_h) \in U$  et  $(x_j, x_h) \in U$ , alors :  $x_i x_j \in ? E$  avec  $i, j$ , et  $h \in \{1, 2, \dots, n\}$ .

Sur la représentation d'arcs circulaires de  $G$ , les arcs :  $A_i, A_j$  et  $A_h$  sont correspondants respectivement aux sommets :  $x_i, x_j$  et  $x_h$ . Comme les arêtes  $x_i x_h$  et  $x_j x_h$ , leur sommets sont orconnés de cette manière, alors d'après le théorème de Tucker, les arcs  $A_i, A_j$  se situent avant  $A_h$  (voir fig 3.5).

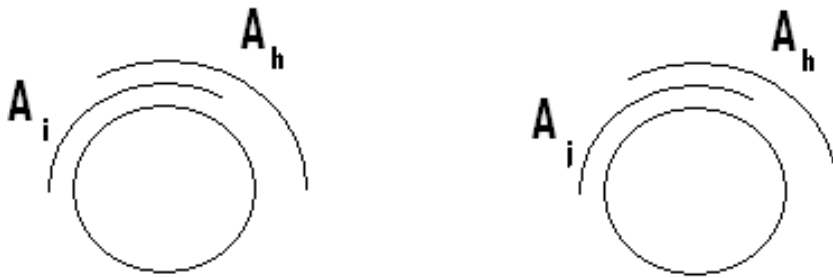


FIG. 3.5 –

Notons que la position des arcs  $A_i$  et  $A_j$ , l'un par rapport à l'autre, ne nous intéresse pas.

Imaginons maintenant, une aiguille qui commence à tourner

dans le cercle, et dans le sens des aiguilles d'une montre, à partir d'un point de départ qui est hors de  $(A_i \cup A_j \cup A_h)$ .

A l'instant  $t_o$  où l'aiguille rencontre l'extrémité initiale de l'arc  $A_h$ , c'est sûrement qu'elle n'a pas encore rencontré l'extrémité terminale de l'arc  $A_i$  (car  $A_i$  et  $A_h$  s'intersectent.....(1).

A ce même instant  $t_0$ , cette aiguille n'a pas rencontré l'extrémité terminale de l'arc  $A_j$  (car  $A_j$  et  $A_h$  s'intersectent.....(2).

De (1) et (2) : au moins à cet instant  $t_o$ , l'aiguille rencontre  $A_i$  et  $A_j$ , donc  $A_i \cap A_j \neq \emptyset \Rightarrow x_i x_j \in E$ , d'où la  $B_1$ -orientation de  $U$ . □

### 3.3.1 Exemple d'application

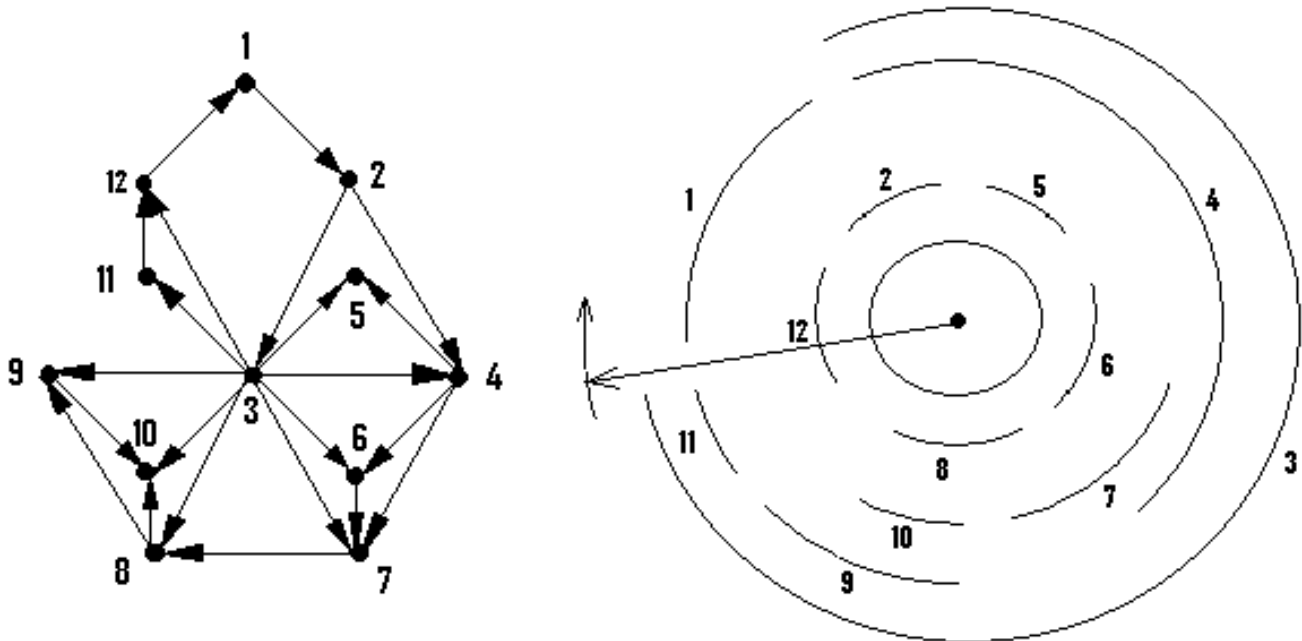


FIG. 3.6 – Cette figure nous montre un graphe d'arcs circulaires muni d'un ordre circulaire  $1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12$ , et d'une  $B_1$ -orientation  $U$ , tels que pour tout  $1 \leq i < j \leq 12$  :

$$U = \begin{cases} (i, j), si & i + 1, i + 2, \dots, j \in adj(i); \\ (j, i), si & j + 1, j + 2, \dots, i \in adj(j). \end{cases}$$

Si on prend, par exemple, l'arc  $(3, 11)$  qui est dans  $U$ , c'est sûrement que  $4, 5, 6, 7, 8, 9, 10, 11 \in adj(3)$ , avec  $U$ , une  $B_1$ -orientation.

# Conclusion et Perspectives

Après avoir rappelé la reconnaissance de la classe des graphes  $B_1$ -orientables qui a été introduite juste récemment, nous avons donné deux démonstrations à deux aspects différents : la première démonstration dont l'aspect est algorithmique est pour montrer que les graphes triangulés sont des graphes  $B_1$ -orientables, alors que la deuxième démonstration dont l'aspect est théorique est pour montrer que les graphes d'arcs circulaires sont des graphes  $B_1$ -orientables.

Pour nos perspectives futures, plusieurs issues sont encore ouvertes. Pour cela, nous posons les questions suivantes :

1. Trouver d'autres sous-classes de cette classe.
2. Il serait intéressant de voir que le calcul de certains paramètres de graphes, difficile dans le cas général, est polynomial sur cette classe.
3. Cette classe n'est-elle pas incluse dans une autre, plus importante ?

# Bibliographie

- [1] J.C. Arditti, Partially ordered sets and their comparability graphs, their dimension and their adjacency, Comptes rendus du Congrès international de combinatoire et théorie des graphes, Orsay 1976 (CNRS Paris 1978).
- [2] C. Berge, Les problèmes de colorations en théorie des graphes, Publ. Inst. Statist. Univ. Paris 9, 1960.
- [3] C. Berge, Graphes et Hypergraphes, Dunod, 1970.
- [4] G.A. Dirac, On rigid circuit graphs, Abh. Math. Sem, Univ. Hamburg, 25, 1961.
- [5] D.R. Fulkerson and O.A. Gross, Incidence matrices and interval graphs, Pacific Journal Math., 15 :835-855, 1965.
- [6] T. Gallai, Transitiv orientierbare graphen, Acta Math. Acad. Sci. Hungar, 18 (1967), 25-66.
- [7] A. Ghouila-Houri, Caractérisation des graphes non orientés dont on peut orienter les arêtes de manière à obtenir le graphe d'une relation d'ordre, C. R. Acad. Sci. Paris, 254 (1962), 1370-1371.

- [8] P.C. Gilmore and A.G. Hoffman, A characterization of comparability graphs and of interval graphs, *Canad. J. Math*, 16 (1964), 539-548.
- [9] M.C.Golumbic, *Algorithms Graph Theory and Perfect Graphs*, Academic Press, New York University, 1980.
- [10] H. Hadwiger, H. Debrunner and V. Klee, *Combinatorial Geometry in the plane*, Holt Rinehardt and Winston, New York, 1964.
- [11] V. Klee, What are the intersection graphs of arcs in a circle?, *American Mathematical Monthly*, 76 (1976) 810-813.
- [12] Christophe Paul, *Parcours en Largeur Lexicographique : Un algorithme de partitionnement, application aux graphes et généralisation*, Thèse de DOCTORAT en informatique, Université Montpellier II, 1998.
- [13] A.Pnueli, A. Lempel and S. Even, Transitive orientation of graphs and indentification of permutation graphs, *Cand. J. Math*, 23 (1971), 160-175.
- [14] Donald J.Rose, Triangulated graphs and elimination process, *J. Math. Anal*, 32 : 597-609, 1970.
- [15] Donald J.Rose, R.Endre Tarjan, and George S.Leucker, Algorithmic aspects of vertex elimination of graphs, *SIAM Journal of computing*, 5(2) : 266-283, June 1976.

- [16] B.Sadi,  $B_1$ -orientable graphs, *Revue sciences et technologie*, Université de constantine.
- [17] J.P.Spinrad, List of open problems, manuscript, 1997.
- [18] M.Taffis et B.Sadi, Les graphes triangulés sont des graphes  $B_1$ -orientables, Contribution au 6 ème colloque sur l'optimisation et les systèmes d'information, (Annaba du 25 au 27 mai 2009).
- [19] A. Tucker, Two characterizations of proper circular-arc graphs, Ph.D. Thesis, Stanford Operation Research Department, Stanford. Calif. USA, 1969.
- [20] A. Tucker, Characterizing circular-arc graphs, *Bull. American mathematical Society*, 76 (1970) 1257-1260.
- [21] A. Tucker, Matrix characterizations of circular-arc graphs, *Pacific Journal of Mathematics*, 39 (1971) 535-545.
- [22] A. Tucker, Structure theorems for some circular-arc graphs, *Discrete Mathematics*, 7 (1974) 167-195.
- [23] A.Tucker, Circular arc-graphs : New uses and a new algorithm, in : Theory and application of graphs, *Lecture Notes in Mathematics*, 642 (1978) 580-589.
- [24] A. Tucker, An efficient test for circular-arc graphs, *SIAM Journal on computing* 9 (1980) 1-24.
- [25] J.R. Walter, Representation of rigid circle graphs, Ph.D. thesis, Wayne State Univ.