

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOULOUD MAMMERI, TIZI-OUZOU

FACULTE DES SCIENCES

DEPARTEMENT DE MATHEMATIQUES



MEMOIRE DE MASTER

SPECIALITE: MATHEMATIQUES

OPTION: RECHERCHE OPERATIONNELLE

Présenté par:

SAIDA BRIKI et HADJILA AZZOUT

Thème:

Etude des graphes B2-orientables

Soutenu le 13 juillet 2015

Devant le jury d'examen composé de:

M. AOUANE
B. SADI
K. KASDI

M. A. A
M. C. A
M. A. A

U.M.M.T.O
U.M.M.T.O
U.M.M.T.O

Président.
Promeuteur.
Examineur.

Remerciements

Louange à ALLAH, le miséricordieux. Sans lui, rien de tout cela n'aurait pu être.

Nous tenons d'abord à remercier M^r SADI B., notre encadreur à l'UMMTO, pour nous avoir proposé ce thème; nous lui sommes très reconnaissantes pour ses remarques, ses conseils et surtout sa disponibilité. Nous le remercions pour nous avoir confié ce travail et pour le temps qu'il nous a consacré.

Nos plus vifs remerciements vont aussi à tous les membres du jury pour avoir accepté d'honorer par leur jugement notre travail, et pour leur participation au jury.

Nous tenons aussi à remercier tous les enseignants du département de Mathématiques qui ont répondu toujours présents pour une aide ou un encouragement.

Enfin, nous remercions toutes les personnes ayant contribué de près ou de loin au bon accomplissement de notre travail.

Dédicaces

A la mémoire de mon très cher père, qui a toujours cru en moi et qui m'a toujours encouragée, je ne saurais exprimer mon regret de ne pas l'avoir parmi nous aujourd'hui.

Que Dieu lui accorde sa miséricorde et l'accueille dans son vaste paradis;

A mon adorable et chère mère que j'aime énormément, pour sa présence et son soutien, et surtout pour la confiance qu'elle a toujours eu en moi, j'espère qu'elle trouvera dans ce modeste travail toute ma reconnaissance et tout mon amour ;

A mon très cher et futur mari Rais ainsi que toute sa famille;

A mes frères Khelifa et Smail;

A mes soeurs :

Malika et son mari Hafidh;

Djamila, son mari Meziane et leurs filles Katia, Lydia et Tinhinane;

Fazia, son mari Kader et leurs enfants Sabrina, Moumouh, Oussama et Rayane;

Sabiha, son mari Smail et son enfant yanis;

A mes tantes et oncles;

A mes cousins et cousines;

A tous mes amis et à tous ceux qui me sont chers;

A Hadjila et sa famille.

Je dédie ce mémoire.

M^{elle} Saida BRIKI

Dédicaces

Je dédie ce modeste travail :

A mes très chers parents, qui sont la source de mon éducation, mon savoir et mes principes.

A mon cher frère Farid.

A ma grand mère.

A mes oncles et tantes, cousins et cousines.

A toute ma famille.

A mes amies Fatiha, Kamilia, Sara, Somia et Nadia.

A Saida et sa famille.

A tous ceux qui me sont chers.

M^{elle} Hadjila AZZOUT

Table des matières

Table des matières	1
Introduction générale	3
1 Quelques concepts de base sur les graphes	5
1.1 Introduction	5
1.2 Définitions	5
1.2.1 Graphe	5
1.2.2 Successeur (prédécesseur)	6
1.2.3 voisin	7
1.2.4 sommets adjacents	7
1.2.5 Arcs adjacents et arêtes adjacentes	7
1.2.6 Sous-graphe	7
1.2.7 Clique	7
1.2.8 Chaîne	8
1.2.9 Cycle	8
1.2.10 Chemin	8
1.2.11 circuit	9
1.2.12 Cycles adjacents en une arête	9
1.2.13 Cycles adjacents en un sommet	10
1.3 Quelques graphes particuliers	10
1.3.1 Graphe symétrique	10
1.3.2 Graphe planaire	10
1.3.3 Graphe complet	11
1.3.4 Graphe de comparabilité	11
2 Les graphes B1-orientables	13
2.1 Introduction	13
2.2 Définitions	13
2.2.1 Graphe B1-orientable	13
2.3 Recherche de l'orientation U	16
2.4 Les classes d'implication d'un graphe	16
2.4.1 Définition	17

2.4.2	Construction des classes d'implication de G	17
2.4.3	Indépendance entre les classes d'implication	20
2.5	L'algorithme d'orientation	20
2.5.1	Le principe de l'algorithme	20
2.5.2	L'algorithme ORAL	21
2.5.3	Complexité de ORAL	21
2.6	Les graphes triangulés	23
2.6.1	Définintion	23
2.6.2	Ordre d'élimination simplicial	24
2.7	L'algorithme	27
2.7.1	Principe de l'algorithme	27
2.7.2	L'algorithme	27
2.7.3	Preuve de l'algorithme	28
2.7.4	Complexité de l'algorithme	28
3	Les graphes B2-orientables	30
3.1	Introduction	30
3.2	Définitions	30
3.2.1	Graphe B2-orientable	30
3.2.2	Caractérisation des graphes B2-orientables	32
	Conclusion générale	39
	Bibliographie	40

Introduction générale

La théorie des graphes est devenue aujourd'hui une des branches les plus florissantes de l'algèbre moderne; elle constitue un domaine des mathématiques et de l'informatique. Elle est aussi très vaste et en évolution constante tant du point de vue des recherches fondamentales que celui des applications. Pour ce qui est de la recherche fondamentale, l'un des travaux les plus répandus de Seymour est le théorème Fort des graphes parfaits, formulé par Claude Berge [1].

Par ailleurs, des algorithmes élaborés pour résoudre des problèmes concernant les objets de cette théorie ont de nombreuses applications dans tous les domaines liés à la notion de réseau (réseaux ferroviaires ou routiers, réseaux de communication,...) et dans d'autres domaines tels que la biologie, la chimie, etc...

Cette théorie constitue l'un des instruments les plus utilisés et les plus efficaces pour résoudre des problèmes discrets posés en recherche opérationnelle.

Un des problèmes de graphes est celui de l'orientation de ses arêtes. On s'intéresse souvent à orienter un graphe non orienté $G=(X, E)$, c'est-à-dire remplacer chaque arête $e=xy \in E$; $x,y \in X$ par un arc $u=(x,y)$ ou $u=(y,x)$. Orienter les arêtes d'un graphe telles que $(x,y) \in U \Rightarrow (y,x) \in U$; $x,y \in X$, définit le type de graphe symétrique et dans la classe des graphes parfaits, les graphes de comparabilité sont définis par une orientation transitive; soit: $\forall x,y,z \in X, (x,y) \in U \text{ et } (y,z) \in U \Rightarrow (x,z) \in U$. Le graphe $G=(X,U)$ devient un graphe orienté. Chaque orientation du graphe définit un type de graphe ou une classe de graphe.

En particulier, notre objectif, à travers cette étude, est de nous intéresser à la classe des graphes B1 et B2-orientables.

Pour bien mener notre travail, nous avons structuré notre mémoire en trois chapitres :

- Dans le premier chapitre, nous avons exposé les concepts de base de la théorie des graphes.
- Dans le deuxième chapitre, nous définissons les graphes B1-orientables et les résultats obtenus sur ces derniers.
- Le troisième chapitre est consacré à la présentation des graphes B2-orientables. On y

développe notre modeste contribution.
Et nous terminons par une conclusion générale.

Chapitre 1

Quelques concepts de base sur les graphes

1.1 Introduction

Ce chapitre est consacré en premier lieu à la présentation de quelques notions et définitions de base de la théorie des graphes. Ces derniers sont des modèles pour représenter des relations entre objets (graphe orienté) ou des relations symétriques entre eux (graphe non orienté). Le graphe est devenu un modèle puissant pour la résolution des problèmes.

1.2 Définitions

1.2.1 Graphe

Un graphe est défini par un ensemble de sommets et un ensemble de relations entre ces sommets.

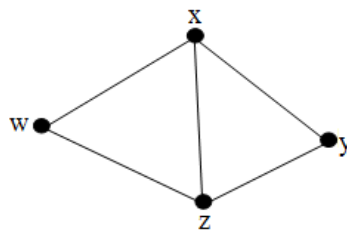


FIG. 1.1 – Graphe $G=(X,E)$

On distingue deux types de graphes :

- **Graphe orienté:** Un graphe orienté est défini par un ensemble de sommets non vide X et un ensemble d'arcs U finis reliant ces sommets. Ce graphe est noté $G=(X,U)$; $n=|X|$ est l'ordre du graphe.

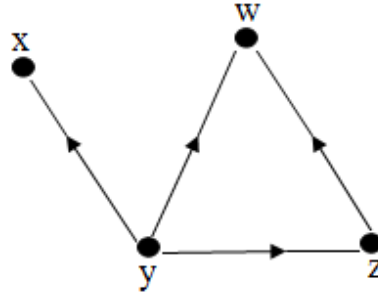


FIG. 1.2 – *Graphe orienté*

- **Graphe non orienté:** Lorsque l'orientation des arêtes n'est pas invoquée, le graphe est dit non orienté. Il est noté $G=(X,E)$

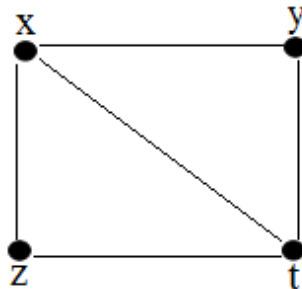


FIG. 1.3 – *Graphe non orienté*

$x,y \in X$, si $u=(x,y)$ est un arc de U , x est l'**extrémité initiale** de u et y son **extrémité terminale**.
 $x,y \in X$, $e=xy$ est une arête de E .

1.2.2 Successeur (prédécesseur)

On dit que x_2 est un **successeur** de x_1 , s'il existe un arc $u=(x_1,x_2)$.
 $\Gamma_G^+(x)$ est l'ensemble des successeurs de x .

x_1 , est alors, le **prédécesseur** de x_2 . $\Gamma_G^-(x)$ est l'ensemble des prédécesseurs de x .

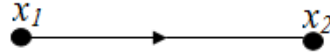


FIG. 1.4 – $L'arc (x_1, x_2)$

1.2.3 voisin

Un sommet y est un voisin du sommet x si $y \in \Gamma_G^+(x)$ ou $y \in \Gamma_G^-(x)$.
 Dans un graphe non orienté, y est voisin de x s'il existe une arête $e=xy$ dans ce graphe.
 L'ensemble des voisins $V(x) = \{y \in X : (x, y) \in E\}$.

1.2.4 sommets adjacents

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 .

1.2.5 Arcs adjacents et arêtes adjacentes

Deux arcs ou deux arêtes sont dits adjacents s'ils ont une extrémité commune.

1.2.6 Sous-graphe

Un sous -graphe d'un graphe G , est un graphe composé de certains sommets de G , ainsi que de toutes les arêtes existant entre ces sommets.

1.2.7 Clique

Une clique de G est un sous -graphe complet de G ou tous les sommets sont deux à deux adjacents. Elle est dite maximale s'il n'existe pas une autre clique qui la contient.

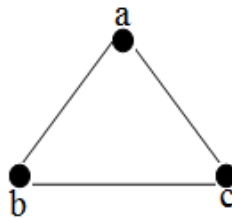


FIG. 1.5 – La clique $G=(X, E)$

1.2.8 Chaîne

Une chaîne joignant deux sommets dans un graphe G est une suite de sommets reliés par des arêtes telle que, deux sommets successifs sont reliés par une arête. Elle est notée par : $(x_0, x_1, \dots, x_k)$.

La longueur de la chaîne est égale au nombre d'arêtes qui la composent.



FIG. 1.6 – *Chaîne a b c d*

Une chaîne est dite **simple** si elle passe, au plus, une seule fois par chaque arête.

1.2.9 Cycle

Un cycle est une chaîne simple dont les deux extrémités coïncident.

On le note par : $(x_0, x_1, \dots, x_k = x_0)$.

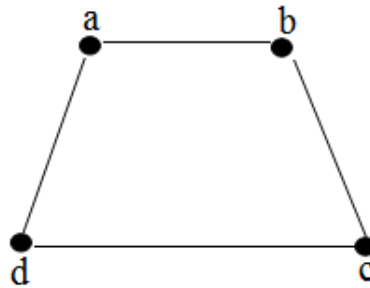


FIG. 1.7 – *Cycle a b c d*

1.2.10 Chemin

Soit $G = (X, U)$ un graphe. Un chemin de G , est une suite de sommets reliés successivement par des arcs orientés dans le même sens.

On note un tel chemin par : $(x_0, x_1, \dots, x_k)$.

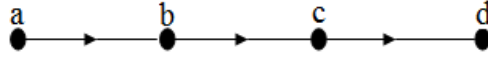


FIG. 1.8 – *Chemin a b c d*

Un **chemin** est dit **simple** s'il passe une seule fois par ses arcs.

1.2.11 circuit

Un circuit est un chemin dont les deux extrémités sont confondues.

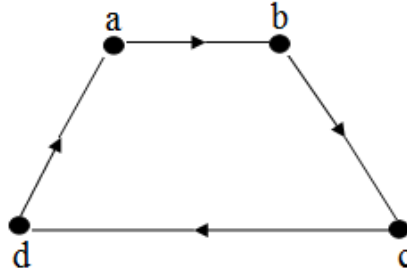


FIG. 1.9 – *circuit a b c d*

1.2.12 Cycles adjacents en une arête

Une corde est une arête reliant deux sommets non consécutifs d'un cycle ou d'une chaîne. On note C_k , le cycle de longueur k .

Deux cycles C_k et $C_{k'}$, $k \geq 4$; $k' \geq 4$ sans corde sont adjacents en une arête si $C_k \cap C_{k'} = \{e\}$; e arête de E .

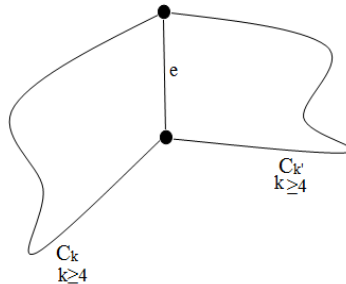


FIG. 1.10 – $C_k \cap C_{k'} = \{e = (x, y)\}$

1.2.13 Cycles adjacents en un sommet

Deux cycles C_k et $C_{k'}$, $k \geq 4$; $k' \geq 4$ sans corde sont adjacents en un sommet si $C_k \cap C_{k'} = \{x\}$; x sommet de X .

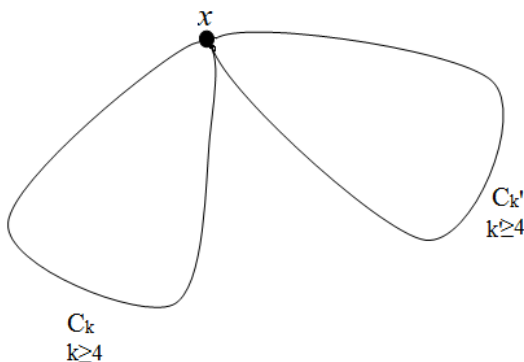


FIG. 1.11 – $C_k \cap C_{k'} = \{x\}$

1.3 Quelques graphes particuliers

1.3.1 Graphe symétrique

Un graphe $G=(X,U)$ est dit symétrique si, $\forall x,y \in X$, $(x,y) \in U \Rightarrow (y,x) \in U$.

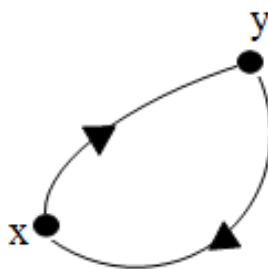


FIG. 1.12 – *Graphe symétrique*

1.3.2 Graphe planaire

Une représentation planaire d'un graphe $G=(X,E)$ est une représentation de G , dans le plan, telle que ses arêtes ne s'intersectent qu'en des sommets de G .

Le graphe G est **planaire** s'il admet une représentation planaire.

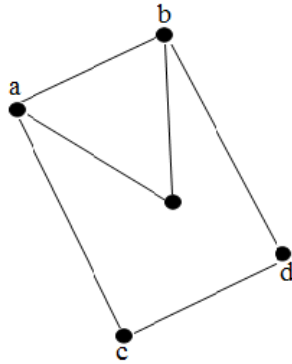


FIG. 1.13 – *Graphe planaire*

1.3.3 Graphe complet

Un graphe complet est un graphe dont les sommets sont deux à deux adjacents.

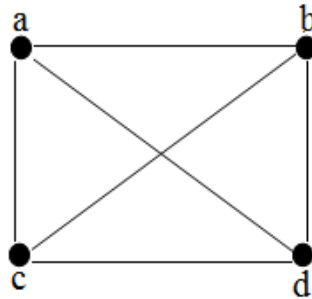


FIG. 1.14 – *Graphe complet*

1.3.4 Graphe de comparabilité

Un graphe $G=(X,E)$ est de comparabilité s'il existe une orientation U de ses arêtes telle que: $\forall x,y,z \in X; (x,y) \in U \text{ et } (y,z) \in U \Rightarrow (x,z) \in U$.

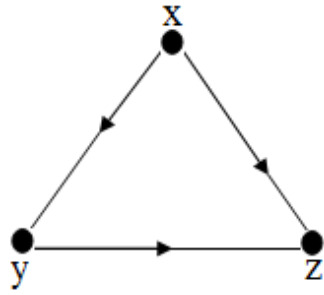


FIG. 1.15 – *Graphe de comparabilité*

Chapitre 2

Les graphes B1-orientables

2.1 Introduction

Dans ce chapitre, nous définissons et étudions la B1-orientation. Nous présentons des propriétés et quelques résultats établis sur la B1-orientation. Après l'étude de cette classe de graphes et l'algorithme de reconnaissance de cette dernière, nous donnons un algorithme polynomial montrant que la classe des graphes triangulés est incluse dans celle des B1-orientables, ainsi que l'algorithme d'orientation de ces graphes.

2.2 Définitions

2.2.1 Graphe B1-orientable

Soit $G=(X,E)$ un graphe non orienté, on cherche **une orientation** U des arêtes de E telle que $\forall a,b,c \in X, (c,a) \in U \text{ et } (b,a) \in U \Rightarrow bc \in E$

U est la B1-orientation.

Le graphe orienté $G^*=(X,U)$ est le **graphe B1-orientable**.

Exemple:

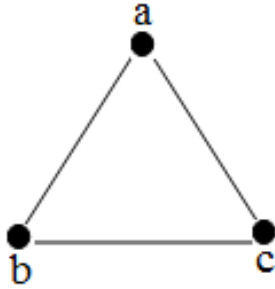


FIG. 2.1 – *Graphe G non orienté*

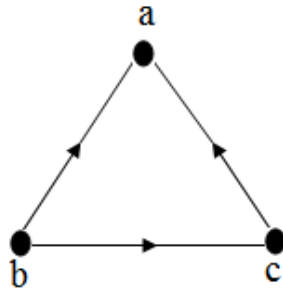
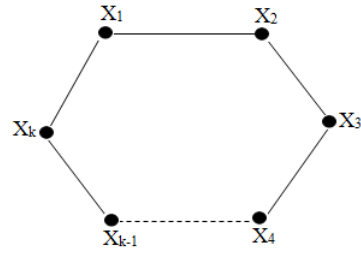


FIG. 2.2 – *Graphe B1-orientable de G*

Lemme 2.1. [2] Dans une B1-orientation, tout cycle C_k ; $k \geq 4$, sans corde, d'un graphe $G=(X,E)$ est orienté suivant un circuit.

Démonstration:

Soit un cycle C_k ; $k \geq 4$, sans corde.



Supposons que l'orientation de C_k n'est pas selon un circuit. Alors, il existe deux arcs $(x_i, x_{i+1}) \in U$ et $(x_{i+2}, x_{i+1}) \in U$. Pour que cette orientation soit une B1-orientation, il faut que $x_i x_{i+2}$ appartienne à E .

$x_i x_{i+2}$ est une corde de C_k ; or C_k n'a pas de corde.

2.3 Recherche de l'orientation U

Le principe de la recherche de l'orientation U est basé sur l'orientation d'une première arête ab , de a vers b . Si toutes les arêtes sont orientées, alors la B1-orientation est trouvée. Sinon, l'arête ab est orientée de b vers a . On oriente, alors toutes les autres arêtes. Si on y arrive, la B1-orientation est trouvée, sinon, le graphe n'est pas B1-orientable.

Exemple :

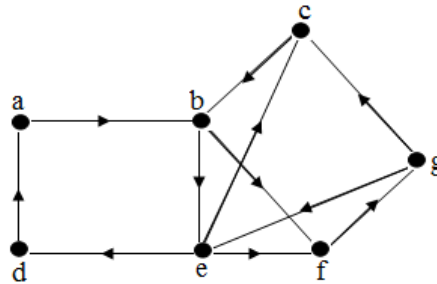


FIG. 2.3 – $(a,b) \in U$. U n'est pas une B1-orientation

Si $(a,b) \in U$ alors U n'existe pas

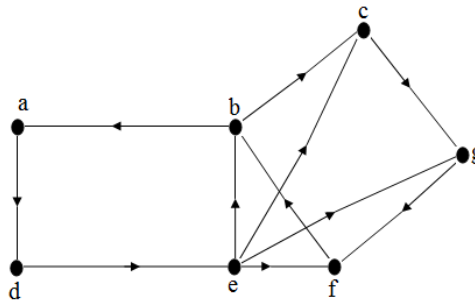


FIG. 2.4 – $(b,a) \in U$. L'orientation U est une B1-orientation

Si $(b,a) \in U$ alors

$U = ((b,a), (a,d), (d,e), (e,b), (e,c), (e,g), (e,f), (f,b), (b,c), (c,g), (g,f))$ est une B1-orientation.

2.4 Les classes d'implication d'un graphe

Forçage: Une arête ab " force " une arête bc si l'orientation de ab oblige bc à avoir une orientation bien définie.

2.4.1 Définition

On note C_i , la classe d'implication de G , avec $e \in E$;

$$C_i(e) = \{e \cup \{e \in E \mid e \text{ est forcé par } e\}\}$$

- La classe d'implication qui commence par l'arc (a, b) , on la note $C_{(a,b)}$.
- la classe d'implication qui commence par l'arc (b,a) , on la note $C_{(a,b)}^{-1}$.
- $\|C_{(a,b)}\|$ représente l'ensemble des arcs de G qui sont contenus dans $C_{(a,b)}$.
- La classe d'implication numéro i , de G , On la note C_i .

2.4.2 Construction des classes d'implication de G

Pour calculer les différentes classes d'implication de G , on applique le principe de l'algorithme suivant [2]:

- 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 $C_1, C_2, \dots, C_n$ est l'ordre des arcs choisis dans 1), 2), 3).

Lemme 2.2. [2] Soient $C_1, C_2, \dots, C_k$, les classes d'implication de G , dans cette ordre. 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, $|E| \leq \sum_{i=1}^k \|C_i\|$.

Exemple:

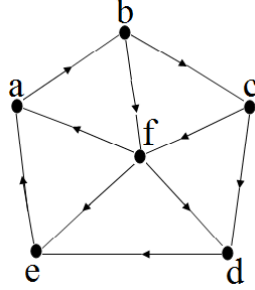


FIG. 2.5 – *Les classes d'implication dans le graphe B1-orientable*

$$C_{(a,b)} = \{(a,b), (b,c), (c,d), (d,e), (e,a)\} = C_1$$

$$C_{(c,f)} = \{(c,f), (f,e), (f,a)\} = C_2$$

$$C_{(b,f)} = \{(b,f), (f,d)\} = C_3$$

$$\sum_{i=1}^3 |C_i| = 10 = |E|.$$

sont les trois classes d'implication de G , qui prouvent la B1-orientation de G .

$C_{(a,b)} \cup C_{(c,f)} \cup C_{(b,f)}$ comporte l'ensemble des arcs qui sont contenus dans G .

Définitions

- Une classe d'implication de G **possède une contradiction**, si elle contient, simultanément, les arcs (a,b) et (b,a) .

Un exemple est donné dans la FIG 2.2.

$C_{(a,b)} = \{(a,b), (b,c), (c,g), (g,f), (f,b), (b,a), \dots\}$, possède une contradiction, car elle contient, simultanément, les arcs (a,b) et (b,a) .

- Si une classe d'implication de G **possède une contradiction**, alors elle est dite **incompatible**.

Exemple:

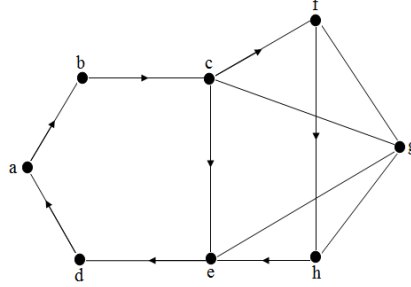


FIG. 2.6 – $(a,b) \in U$. U admet une contradiction

$C_{(a,b)} = \{(a,b), (b,c), (c,e), (e,d), (d,a), (f,h), (h,e), \dots\}$ a une contradiction.

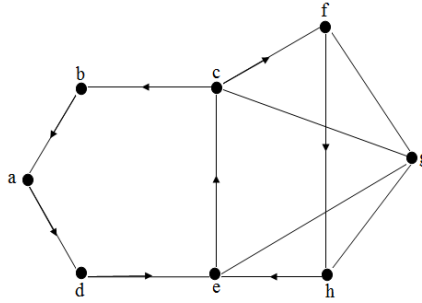


FIG. 2.7 – $(b,a) \in U$. U admet une contradiction

$C_{(b,a)} = \{(b,a), (a,d), (d,e), (e,c), (c,b), (f,h), \dots\}$ a une autre contradiction.

- Si une classe d'implication de G **ne possède pas une contradiction**, alors elle est dite **compatible**.

Exemple:

Les classes d'implication $C_{(b,a)}$ et $C_{(b,c)}$ de la FIG 2.3 sont compatibles.

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

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

car ces deux classes ne contiennent pas de contradiction.

2.4.3 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 , $i=1, \dots, k$, 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 2.3. : L'existence d'une classe d'implication **incompatible** est indépendante de l'ordre dans lequel les classes d'implication 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 orientation, et l'orientation de l'arc (b,a) implique une autre orientation. 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 2.1. 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 B1-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 2.3); cela signifie que dans n'importe quel ordre des classes d'implication, on a une classe d'implication incompatible. Donc G n'est pas B1-orientable.

2.5 L'algorithme d'orientation

2.5.1 Le principe de l'algorithme

On prend une arête quelconque dans $E_i = E$, et on construit une classe d'implication C_i à partir de l'arête $x_i y_i$,

Si C_i a une contradiction. On néglige C_i , on construit C_i^{-1} à partir de l'arête $y_i x_i$,

- Si C_i et C_i^{-1} ont une contradiction alors G n'est pas B1-orientable.

- Si C_i n'a pas une contradiction et C_i^{-1} n'a pas une contradiction, faire $E_{i+1}=E_i-C_i$ (resp. $E_{i+1}=E_i-C_i^{-1}$)
- Si C_i ou C_i^{-1} n'a pas de contradiction, alors G est B1-orientable.

L'algorithme polynomial de reconnaissance de la classe B1-orientable, qui est écrit dans [3] est le suivant:

2.5.2 L'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 B1-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 B1-orientable
 - (4.b) sinon, faire : $i \leftarrow i+1$; aller en (1).

2.5.3 Complexité de ORAL

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

Application de l'algorithme de ORAL sur l'exemple suivant:

Cas 1:

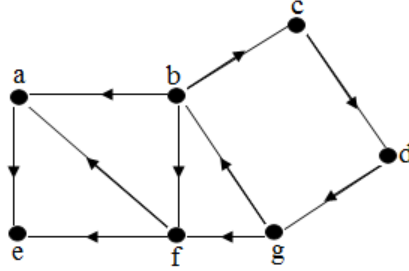


FIG. 2.8 – *Graphe B1-orientable*

Si $(c,d) \in U$

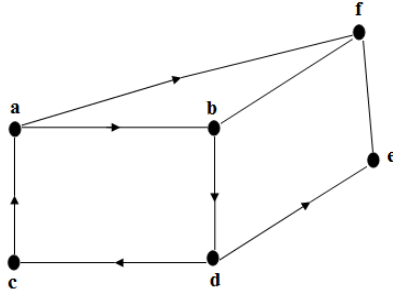
$C_{(c,d)} = \{(c,d), (d,g), (g,b), (b,c), (g,f), (f,e), (b,a), (f,a), (a,e)\}$

$C_{(b,f)} = \{(b,f)\}$

$C_{(c,d)}$ et $C_{(b,f)}$ ne possèdent pas une contradiction.

La B1-orientation U est trouvée, donc, le graphe est **B1-orientable**.

Cas 2:

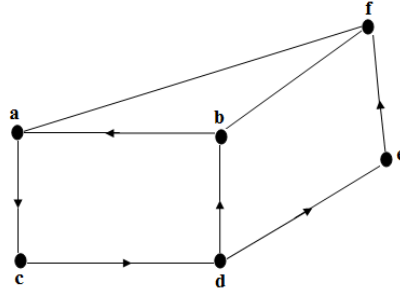


Si $(c,a) \in U$

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

Cette classe possède une contradiction, car, elle contient, simultanément les arcs (a,f) et (f,a)

Cette orientation n'est pas une B1-orientation.



Si $(a,c) \in U$

$$C_{(a,c)}^{-1} = \{(a,c), (c,d), (d,b), (b,a), (d,e), (e,f), \dots\}$$

cette classe possède une contardiction, car $(e,f) \in U$ et $(b,f) \in U$ or $b \notin E$.

Cette orientation n'est pas une B1-orientation

Le graphe n'est pas **B1-orientable** car, il n'admet pas une B1-orientation.

2.6 Les graphes triangulés

L'importance d'une classe de graphes est dans le fait qu'elle contienne des sous-classes de graphes connues ou que, elle même soit contenue dans une plus grande classe.

Nous montrons, dans la suite, que les graphes triangulés sont des graphes B1-orientables.

2.6.1 Définition

Définition 2.1. [4] Un graphe G est dit triangulé si tout cycle de longueur ≥ 4 , admet une corde.

Exemple

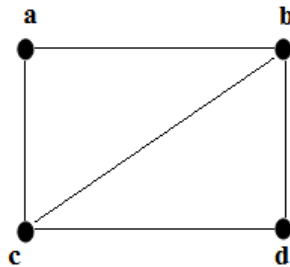


FIG. 2.9 – *Graphe triangulé*

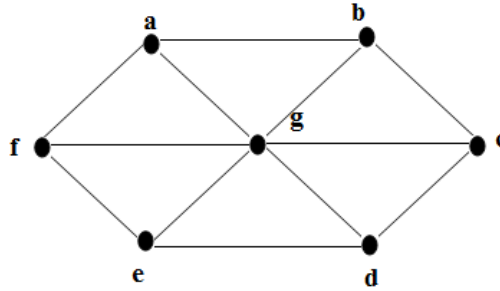


FIG. 2.10 – *Graphe non triangulé*

Sommet simplicial:

$G=(X,E)$, $x \in X$;

x est **un sommet simplicial** si le voisinage de x est une clique. Il est défini, aussi, comme un sommet n'appartenant qu'à une seule clique.

Exemple

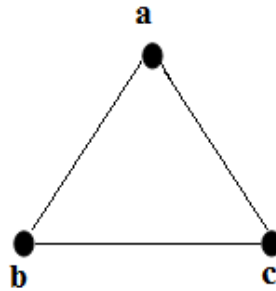


FIG. 2.11 – *a,b,c sont simpliciaux.*

2.6.2 Ordre d'élimination simplicial

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

Exemple:

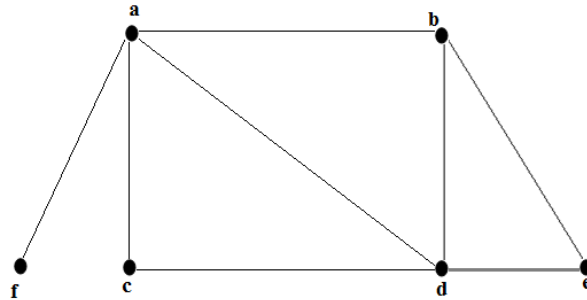


FIG. 2.12 – $\sigma = [f, e, b, a, d, c]$ est un ordre d'élimination simplicial

Lemme 2.4. [5] Tout graphe triangulé $G=(X,E)$ possède au moins un sommet simplicial. si G n'est pas un graphe complet, alors il possède au moins deux sommets simpliciaux non adjacents.

Théorème 2.1. [6] Un graphe est triangulé si et seulement si il existe un ordre d'élimination simplicial.

Démonstration:

$\Rightarrow$) Dans un graphe triangulé, on trouve un sommet simplicial, on le supprime, le graphe obtenu reste un graphe triangulé...

$\Leftarrow$) Si un graphe n'est pas triangulé, alors il possède un cycle C_k ; $k \geq 4$, sans corde, et aucun sommet de ce cycle ne peut être éliminé simplicialement.

Remarques:

Tout sous-graphe d'un graphe triangulé est également triangulé.

Si G est une clique, toute orientation des arêtes de G est une B1-orientable.

Théorème 2.2. [3] Tout graphe triangulé est B1-orientable.

Démonstration:

La démonstration de ce théorème découle de l'algorithme suivant, composé de deux parties. La première est lex-BFS (parcours en profondeur dans un ordre lexicographique), qui consiste en l'énumération des sommets simpliciaux du graphe.

La deuxième partie donne la B1-orientation du graphe.

L'algorithme Lex-BFS qui calcule un ordre d'élimination simplicial, qui est écrit dans [7] est le suivant:

Donnée: Un graphe $G=(X,E)$

Résultat: Un ordre σ des sommets de G

(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 non numéroté de marque maximum dans l'ordre lexicographique ;

$\sigma(i) \leftarrow x$

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

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

Sa **complexité** est $O(n+m)$, où $n=|X|$; $m=|E|$.

Application de l'algorithme Lex-BFS

Soit G , le graphe triangulé suivant:

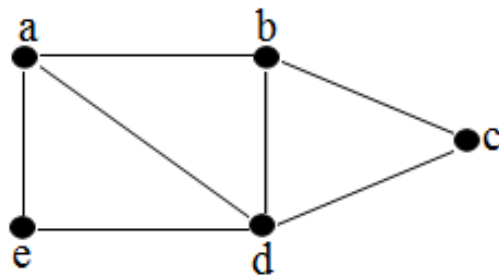


FIG. 2.13 – $\sigma=[e,c,d,b,a]$ est un ordre d'élimination simplicial

sommet	marque	numéro				
a	$\emptyset$	$\emptyset$ 5	$\emptyset$ 5	$\emptyset$ 5	$\emptyset$ 5	$\emptyset$ 5
b	$\emptyset$	{5}	{5} 4	{5} 4	{5} 4	$\emptyset$ 4
c	$\emptyset$	$\emptyset$	{ $\emptyset$,4}	{ $\emptyset$,4,3}	{ $\emptyset$,4,3} 2	{ $\emptyset$,4,3} 2
d	$\emptyset$	{5}	{5,4}	{5,4} 3	{5,4} 3	{5,4} 3
e	$\emptyset$	{5}	{5}	{5,3}	{5,3}	{5,3} 1

L'ordre d'élimination simplicial obtenu est $\sigma=[e,c,d,b,a]$.

2.7 L'algorithme

2.7.1 Principe de l'algorithme

L'algorithme comporte deux parties:

La première partie, c'est Lex-BFS, dont le principe 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 $G_i = G[x_1, \dots, x_n]$ pour tout $1 \leq i \leq n$. Cet ordre existe si le graphe est triangulé.

La deuxième partie, est un fragment ajouté à Lex-BFS, dont l'utilité est de donner au graphe une orientation liée à l'ordre d'élimination simplicial des sommets et 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 $1 \leq i \leq n-1$.

Cette orientation est une **B1-orientation** pour le graphe.

2.7.2 L'algorithme

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

Résultat: Une B1-orientation U de G

(0) $U = \emptyset$

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

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

2.7.3 Preuve de l'algorithme

On suppose que l'orientation trouvée n'est pas B1-orientation, donc $\exists (x,y) \in U$ et $(z,y) \in U$ avec $xz \notin E$. Les arêtes xy , zy sont orientées à l'étape où y est un sommet simplicial; par conséquent, chaque couple dans le voisinage de y forme une arête, y compris xz .

2.7.4 Complexité de l'algorithme

Dans l'instruction 3, 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 insertions d'arcs $(\sigma(i), \sigma(i))$, en ajoutant le nombre d'opérations maximum pour l'union qui est $n-1$. La complexité de l'instruction 3 est $O(n(n-1)+m+(n-1))=O(n^2+m)$.

les instructions 1 et 2 réalisent Lex-BFS qui est $O(n+m)$. Donc la complexité de l'algorithme est $O(n^2+2m) \simeq O(n^2)$.

Application de l'algorithme

Dans la FIG.2.12 on a donné l'ordre d'élimination simplicial pour la première partie de **lex-BFS** qui est $\sigma(i) = \{e, c, d, b, a\}$. Alors pour comprendre bien la deuxième partie, il suffit d'appliquer l'instruction 3 de l'algorithme.

On note l'ensemble des sommets simpliciaux par $\sigma(i)$, et l'ensemble des sommets adjacents par $\sigma(j)$.

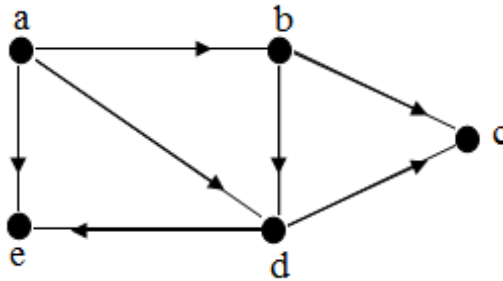


FIG. 2.14 – La B1-orientation du graphe

$U = \emptyset$

i=1:

$\sigma(1) \equiv e; \{ \sigma(j), e \}, \text{ avec } \sigma(j) \in \text{adj}(e), \text{ et } j > 1 \} = \{(a,e), (d,e)\}.$

$U = \emptyset \cup \{(a,e), (d,e)\}.$

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

i=2:

$\sigma(2) \equiv c; \{ \sigma(j), c \}, \text{ avec } \sigma(j) \in \text{adj}(c), \text{ et } j > 1 \} = \{(b, c), (d, c)\}.$

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

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

i=3:

$\sigma(3) \equiv d; \{ \sigma(j), d \}, \text{ avec } \sigma(j) \in \text{adj}(d), \text{ et } j > 1 \} = \{(b, d), (a, d)\}.$

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

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

i=4:

$\sigma(4) \equiv b; \{ \sigma(j), b \}, \text{ avec } \sigma(j) \in \text{adj}(b), \text{ et } j > 1 \} = \{(a, b)\}.$

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

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

D'où la **B1-orientation** cherchée est trouvée.

Chapitre 3

Les graphes B2-orientables

3.1 Introduction

Dans ce chapitre, nous étudions une autre orientation dans les graphes: la B2-orientation. Nous nous inspirons des résultats obtenus sur la B1-orientation, pour caractériser la B2-orientation et arriver à l'algorithme de reconnaissance des graphes B2-orientables. Nous montrons, aussi, par un algorithme polynomial, que la classe des graphes triangulés est incluse dans celle des graphes B2-orientables.

3.2 Définitions

3.2.1 Graphe B2-orientable

Soit $G=(X,E)$ un graphe non orienté, on cherche **une orientation** U^* des arêtes de E telle que $\forall a,b,c \in X, (a,b) \in U^* \text{ et } (a,c) \in U^* \Rightarrow bc \in E$

U^* est la B2-orientation.

Le graphe orienté $G^{**}=(X,U^*)$ est le **graphe B2-orientable**.

Exemple:

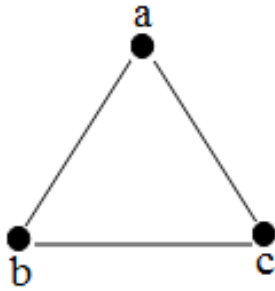


FIG. 3.1 – *Graphe G non orienté*

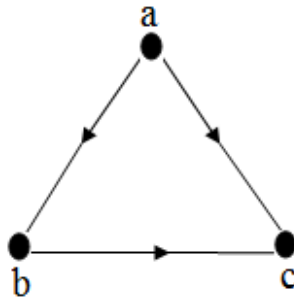
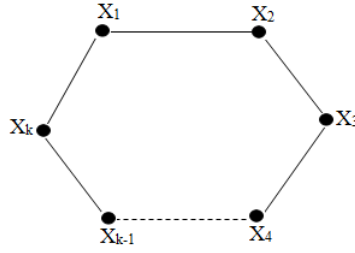


FIG. 3.2 – *Graphe B2-orientable de G*

Lemme 3.1. Dans une B2-orientation, tout cycle C_k ; $k \geq 4$, sans corde, d'un graphe $G=(X,E)$ est orienté suivant un circuit.

Démonstration:

Soit un cycle C_k ; $k \geq 4$, sans corde.



Supposons que l'orientation de C_k n'est pas selon un circuit. Alors, il existe deux arcs $(x_{i+1}, x_i) \in U^*$ et $(x_{i+1}, x_{i+2}) \in U^*$. Pour que cette orientation soit une B2-orientation, il faut que $x_i x_{i+2}$ appartienne à E .

$x_i x_{i+2}$ est une corde de C_k ; or C_k n'a pas de corde.

3.2.2 Caractérisation des graphes B2-orientables

La recherche de l'orientation U^* et les classes d'implication dans les graphes B2-orientables se fait exactement de la même manière que pour les graphes B1-orientables. Pour éviter la répétition, nous avons jugé d'omettre de détailler.

En effet, les exemples suivants illustrent bien notre démarche.

La recherche de l'orientation U^* :

Exemple:

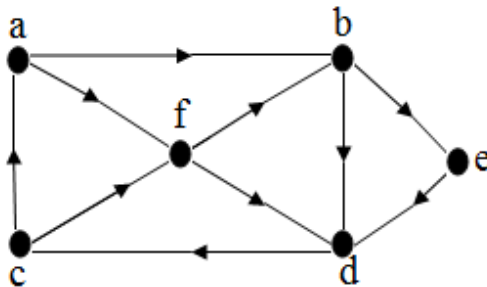


FIG. 3.3 – L'orientation U^* est une B2-orientation

$U^* = ((a,b), (b,d), (d,c), (c,a), (a,f), (f,b), (f,d), (b,e), (e,d), (c,f))$ est une B2-orientation.

La construction des classes d'implication de G :

Exemple:

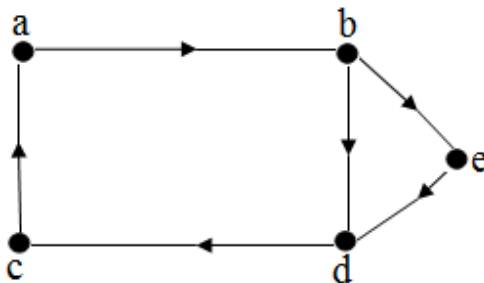


FIG. 3.4 – Les classes d'implication dans le graphe B2-orientable

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

$$C_{(b,e)} = \{(b,e)\}$$

sont les deux classes d'implication de G , qui prouvent la B2-orientation du graphe.

Alors, on remarque que l'algorithme **ORAL** s'applique même pour les graphes B2-orientables.

Application de l'algorithme ORAL sur l'exemple suivant:

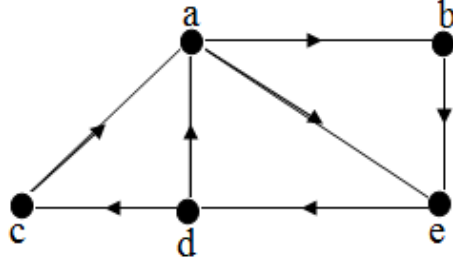


FIG. 3.5 – *Graphe B2-orientable*

Si $(a,b) \in U$ alors,

$$C_{(a,b)} = \{(a,b), (d,a), (c,a)\}$$

$$C_{(d,c)} = \{(d,c), (e,d), (b,e)\}$$

$$C_{(a,e)} = \{(a,e)\}$$

$C_{(a,b)} \cup C_{(d,c)} \cup C_{(a,e)}$ comporte l'ensemble des arcs de G . Donc la B2-orientation U^* est trouvée.

D'où G est **B2-orientable**.

Remarque:

Le théorème 2.2 du chapitre précédent nous donne un algorithme qui montre que les graphes triangulés sont des graphes B1-orientables. On remarque que le théorème reste vrai pour les graphes B2-orientables.

Proposition 3.1. Les graphes triangulés sont B2-orientables.

Démonstration: L'algorithme qui suit montre que les graphes triangulés sont B2-orientables. Après avoir trouvé les sommets simpliciaux, nous appliquons l'algorithme suivant:

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

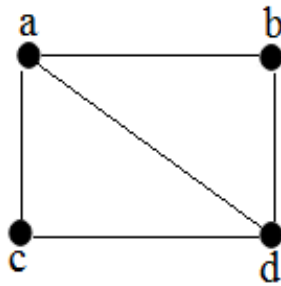
Résultat: Une B2-orientation.

Tant que G possède un sommet, faire

- sélectionner un sommet simplicial x de G
- Orienter, à partir de x , toutes les arêtes de G contenant x
- Supprimer de G le sommet x

Fin tant que.

Application de l'algorithme sur l'exemple suivant:



sommet	marque	numéro			
a	$\emptyset$	$\emptyset$ 4	$\emptyset$ 4	$\emptyset$ 4	$\emptyset$ 4
b	$\emptyset$	$\{\emptyset, 4\}$	$\{\emptyset, 4\}$ 3	$\{\emptyset, 4\}$ 3	$\{\emptyset, 4\}$ 3
c	$\emptyset$	$\{\emptyset, 4\}$	$\{\emptyset, 4\}$	$\{\emptyset, 4\}$	$\{\emptyset, 4\}$ 1
d	$\emptyset$	$\{\emptyset, 4\}$	$\{\emptyset, 4, 3\}$	$\{\emptyset, 4, 3\}$ 2	$\{\emptyset, 4, 3\}$ 2

l'ordre d'élimination obtenu est $\sigma(i)$ est $\{c, d, b, a\}$

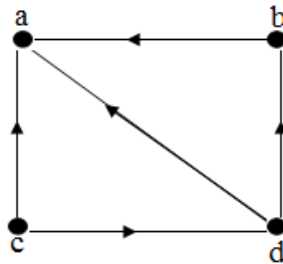


FIG. 3.6 – La B2-orientation du graphe

$$U = \emptyset$$

i=1:

$\sigma(1) \equiv c; \{c, \sigma(j)\}$, avec $\sigma(j) \in \text{adj}(c)$, et $j > 1 \} = \{(c,a),(c,d)\}$.

$$U = \emptyset \cup \{(c,a),(c,d)\}.$$

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

i=2:

$\sigma(2) \equiv d; \{d, \sigma(j)\}$, avec $\sigma(j) \in \text{adj}(d)$, et $j > 1 \} = \{(d,a),(d,b)\}$.

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

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

i=3:

$\sigma(3) \equiv b; \{b, \sigma(j)\}$, avec $\sigma(j) \in \text{adj}(b)$, et $j > 1 \} = \{(b,a)\}$.

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

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

D'où la B2-orientation cherchée est trouvée.

Un isomorphe est une bijection entre les sommets de deux graphes qui préserve les arêtes.

Proposition 3.2. Soit $G=(X,E)$ un graphe non orienté, et soient $C_k; k \geq 4, C_{k'}; k' \geq 4$, deux cycles sans corde.

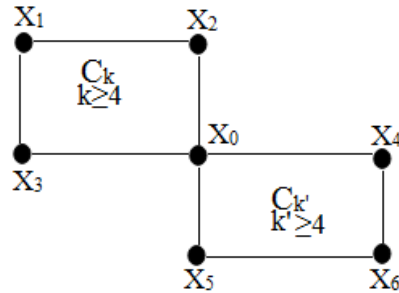
$$(1) C_k \cap C_{k'} = \{x_0\}; x_0 \in X$$

$$(2) C_k \cap C_{k'} = \{e\}; e \in E$$

Alors si G contient une subdivision isomorphe à (1) ou à (2), G n'est ni B1-orientable ni B2-orientable.

Démonstration:

Soit $G=(X,E)$ un graphe non orienté, et soit $C_k; k \geq 4, C_{k'}; k' \geq 4$, deux cycles adjacents en un sommet, sans corde.



Soit un cycle $C_k; k \geq 4$. On oriente l'arête $x_1 x_2$, de x_1 vers x_2 , de telle sorte que C_k soit un circuit.

- Si on oriente le cycle $C_{k'}$ de x_0 vers x_4 , alors, il existe deux arcs $(x_2, x_0) \in U$ et $(x_5, x_0) \in U$, or $x_5 x_2, \notin E$. G n'est pas B1-orientable. Il existe deux arcs $(x_0, x_3) \in U^*$ et $(x_0, x_4) \in U^*$, or $x_3 x_4, \notin E$. G n'est pas B2-orientable.

- Si on oriente le cycle $C_{k'}$ de x_4 vers x_0 , alors, il existe deux arcs $(x_2, x_0) \in U$ et $(x_4, x_0) \in U$, or $x_4 x_2, \notin E$. G n'est pas B1-orientable. Il existe deux arcs $(x_0, x_3) \in U^*$ et $(x_0, x_5) \in U^*$, or $x_3, x_5 \notin E$. G n'est pas B2-orientable.

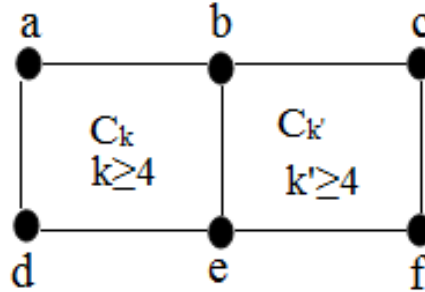
Soit un cycle C_k ; $k \geq 4$. On oriente l'arête $x_2 x_3$, de x_2 vers x_3 , de telle sorte que C_k soit un circuit.

- Si on oriente le cycle $C_{k'}$ de x_0 vers x_4 , alors, il existe deux arcs $(x_3, x_0) \in U$ et $(x_5, x_0) \in U$, or $x_5 x_3, \notin E$. G n'est pas B1-orientable. Il existe deux arcs $(x_0, x_4) \in U^*$ et $(x_0, x_4) \in U^*$, or $x_3 x_4, \notin E$. G n'est pas B2-orientable.

- Si on oriente le cycle $C_{k'}$ de x_4 vers x_0 , alors, il existe deux arcs $(x_4, x_0) \in U$ et $(x_3, x_0) \in U$, or $x_3 x_4, \notin E$. G n'est pas B1-orientable. Il existe deux arcs $(x_0, x_2) \in U^*$ et $(x_0, x_5) \in U^*$, or $x_2, x_5 \notin E$. G n'est pas B2-orientable.

le graphe contenant 1) n'est ni B1, ni B2-orientable.

Soit $G=(X,E)$ un graphe non orienté, et soit C_k ; $k \geq 4$, $C_{k'}$; $k' \geq 4$, deux cycles adjacents en une arête, sans corde.



Soit un cycle C_k ; $k \geq 4$, on oriente l'arête ab , de a vers b , de telle sorte que C_k soit un circuit.

- Si on oriente le cycle $C_{k'}$ de b vers c , alors, il existe deux arcs $(b, e) \in U$ et $(f, e) \in U$, or $bf \notin E$. G n'est pas B1-orientable. Il existe deux arcs $(b, e) \in U^*$ et $(b, c) \in U^*$, or $ec \notin E$. G n'est pas B2-orientable.

- Si on oriente le cycle $C_{k'}$ de c vers b, alors, il existe deux arcs $(a,b) \in U$ et $(c,b) \in U$, or $ac \notin E$. G n'est pas B1-orientable.

Il existe deux arcs $(e,d) \in U^*$ et $(e,f) \in U^*$, or $df \notin E$. G n'est pas B2-orientable.

Soit un cycle C_k ; $k \geq 4$, on oriente l'arête ab, de b vers a, de telle sorte que C_k soit un circuit.

- Si on oriente le cycle $C_{k'}$ de b vers c, alors, il existe deux arcs $(d,e) \in U$ et $(f,e) \in U$, or $df \notin E$. G n'est pas B1-orientable.

Il existe deux arcs $(b,a) \in U^*$ et $(b,c) \in U^*$, or $ac \notin E$. G n'est pas B2-orientable.

- Si on oriente le cycle C_k' de c vers b alors, il existe deux arcs $(c,b) \in U$ et $(e,b) \in U$, or $ec \notin E$. G n'est pas B1-orientable.

Il existe deux arcs $(e,b) \in U^*$ et $(e,f) \in U^*$, or $bf \notin E$. G n'est pas B2-orientable.

le graphe contenant 2) n'est ni B1, ni B2-orientable.

Remarque: Si $G=(X,E)$ est un triangle alors B1 est équivalent à B2.

Exemple:

Si G est une clique, toute orientation des arêtes de G est une B2-orientation.

Conclusion générale

Dans ce mémoire, nous nous sommes intéressées à l'étude des graphes B1 et B2-orientables.

Ce travail nous a permis de nous familiariser avec quelques classes de graphes.

Nous avons donné un algorithme polynomial de reconnaissance de ces classes de graphe et des algorithmes efficaces, qui montrent que les graphes triangulés appartiennent à ces classes. Aussi, nous donnons un procédé de recherche des deux orientations. Il serait intéressant d'implémenter l'ensemble de ces algorithmes.

Bibliographie

- [1] Berge C., Les problèmes de coloration en théorie des graphes, Publ, Inst, Statist, Univ. Paris 9, 1960.
- [2] Taffis M., Quelques sous classes des graphes B1-orientables, mémoire de magistère, 2011.
- [3] Sadi B., B1-orientable graphs, Revue sciences et technologie, Université de constantine.
- [4] Berge C., graphes et hypergraphes, Dunod, 1970.
- [5] Dirac G.A., On rigid circuit graphs, Abh. Math. Sem, Univ. Hambourg, 1961.
- [6] Fulkerson D.R., Gross O.A., Incidence matrices and interval graphs, Pacific Journal Math, 1965.
- [7] Donald J.Rose, R.Endre Tarjan, and George S.Leuker, Algorithm aspects of vertex elimination of graphs, SIAM Journal of computing, 1976.