

2M11.3469.1

Université de Montréal

Reconnaissance Des Graphes Policier-Gagnants

par
Benoit Thériault

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Août, 2006

© Benoit Thériault, 2006.



QA

76

US4

2007
V.005

2007 2-2 1-3

Direction des bibliothèques

AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Reconnaissance Des Graphes Policier-Gagnants

présenté par:

Benoit Thériault

a été évalué par un jury composé des personnes suivantes:

Sylvie Hamel
président-rapporteur

Gena Hahn
directeur de recherche

Alain Tapp
membre du jury

Mémoire accepté le 30 octobre 2006

RÉSUMÉ

Les jeux de poursuite sur les graphes ont leur source dans des problèmes réels et il en existe plusieurs variantes. Afin de bien les étudier, on le fait dans l'abstrait, et c'est la direction prise dans ce mémoire. Notre étude porte sur les propriétés des graphes dans une situation de recherche, cette dernière étant une forme particulière de jeu de poursuite : la recherche est un jeu où des policiers tentent de capturer un voleur/intru sur les sommets d'un graphe. Différentes familles de graphes seront étudiées dans ce contexte, et on retrouve au nombre de celles-ci les tournois, les graphes à cordes et plusieurs autres. Des propriétés importantes pour chaque famille seront présentées, étant parfois accompagnées de résultats nouveaux. Parmi ceux-ci se trouvent de nouveaux résultats sur les tournois et plusieurs preuves seront présentées de façon détaillée. Un algorithme permettant d'étudier ces graphes dans le contexte de la recherche sera aussi décrit et analysé. Plusieurs versions de cet algorithme seront comparées, et des tests de performance seront présentés. Cette partie constitue le coeur de ce mémoire.

Mot-Clés : police, voleur, recherche, graphe, poursuite

ABSTRACT

Pursuit games on graphs come from real life problems and have many variants. To study them, it is useful to do so in the abstract and that is what this thesis does. We focus on properties of the graphs on which a searching game is played. Searching is a type of pursuit game where cops are trying to catch a robber on the vertices of a graph. Different families of graphs such as tournaments, chordal graphs, and many others will be studied in that context. Important properties will be presented for each family, sometimes accompanied by new results. Among those are new results on tournaments, with their detailed proofs. An algorithm helpful to study graphs in a searching context will also be presented and analyzed. Lastly, different versions of this algorithm will be compared.

Keywords: cop, robber, pursuit, graph, cop-win, searching

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES TABLEAUX	vii
LISTE DES FIGURES	viii
NOTATION	x
REMERCIEMENTS	xi
INTRODUCTION	xii
CHAPITRE 1 : NOTIONS DE GRAPHS	1
1.1 Voisinage	3
1.2 Chaînes et cycles	4
1.3 Arbres, arborescences	5
1.4 Distance	6
1.5 Sous-graphes	7
1.6 Homomorphismes	8
CHAPITRE 2 : JEUX DE POURSUITES	10
2.1 Variantes	10
2.1.1 Sweeping	10
2.1.2 Contraintes sur la mobilité	11
2.1.3 Visibilité variable	11
2.1.4 Autres	11
2.2 Définitions pour le jeu de recherche	12

CHAPITRE 3 : OPÉRATIONS SUR LES GRAPHERS	17
3.1 Union sur un sous-graphe commun	17
3.2 Construction de Hirsh	18
3.3 Produits de graphes non-réflexifs	19
3.3.1 Produit cartésien	19
3.3.2 Produit fort	20
3.3.3 Produit catégorique	21
3.4 Produits de graphes réflexifs	22
 CHAPITRE 4 : GRAPHERS À PONTS ET À CORDES	 24
4.1 Propriétés des graphes à cordes	24
4.2 Propriétés des graphes à ponts	26
 CHAPITRE 5 : TOURNOIS	 31
5.1 Propriétés générales	31
5.2 Tournois réflexifs	33
5.3 Tournois non-réflexifs	36
5.4 Tournois non 2-policiers-gagnants	39
 CHAPITRE 6 : ALGORITHME	 41
6.1 Di-graphe de déplacement	41
6.2 Principe de l'algorithme	43
6.3 Version 1	46
6.4 Version 2	46
6.5 Version 3	49
6.6 Justifications	50
6.7 Analyse des résultats	54
 CHAPITRE 7 : CONCLUSION	 60
 BIBLIOGRAPHIE	 61

LISTE DES TABLEAUX

6.1	Résultats de la version 1	57
6.2	Résultats de la version 2	58
6.3	Résultats de la version 3	59

LISTE DES FIGURES

1.1 Un di-graphe D et son graphe sous-jacent G	2
1.2 Un graphe non-réflexif G_1 , un graphe partiellement réflexif G_2 et un graphe réflexif G_3	2
1.3 Un di-graphe non-réflexif D_1 , un di-graphe partiellement réflexif D_2 et un di-graphe réflexif D_3	3
1.4 Un homomorphisme d'un graphe G vers un graphe H , et une rétraction sur ce dernier, résultant en H'	9
2.1 Un di-graphe D policier-gagnant, un graphe G 2-policiers-gagnant et G_2 , le graphe de Peterson, 3-policiers-gagnant.	13
3.1 Exemple d'une union sur un sous-graphe commun	17
3.2 Exemple d'une construction de Hirsh	19
3.3 Exemple d'un produit cartésien $K_3 \square K_3$	20
3.4 Exemple d'un produit fort $K_3 \boxtimes K_3$	21
3.5 Exemple d'un produit catégorique $K_3 \times K_3$	22
3.6 Exemple d'un produit catégorique sur des graphes réflexifs sans modification de la définition du produit.	23
4.1 Graphe G	27
4.2 Illustration des quatre cas possibles	28
5.1 Exemple de tournoi	34
5.2 Exemple de tournoi	35
5.3 Exemple de tournoi	37
5.4 Exemple de tournoi	38
5.5 Tournoi de Paley d'ordre 7.	40
6.1 Exemple du voisinage d'un sommet de C_M	43
6.2 Exemple du voisinage d'un sommet de R_M	43

6.3	Exemple du voisinage d'un sommet de C_M dans la nouvelle construction	47
6.4	Exemple du voisinage d'un sommet de R_M dans la nouvelle construction	48
6.5	Exemple d'une matrice d'adjacence	55

NOTATION

- $N_G(v)$ $\{u \mid \{v, u\} \in E(G)\}$
 $N_D^+(v)$ $\{u \mid (v, u) \in A(D)\}$
 $N_D^-(v)$ $\{u \mid (u, v) \in A(D)\}$
 $N_G(K)$ $\{u \in V(G) \setminus K \mid \{v, u\} \in E(G), v \in K\}$, où $K \subseteq V(G)$
 $N_G[K]$ $\{u \mid \{v, u\} \in E(G), v \in K\} \cup K$, où $K \subseteq V(G)$
 $N_D^-(K)$ $\{u \in V(D) \setminus K \mid (u, v) \in A(D), v \in K\}$, où $K \subseteq V(D)$
 $N_D^-[K]$ $\{u \mid (u, v) \in A(D), v \in K\} \cup K$, où $K \subseteq V(D)$
 $N_D^+(K)$ $\{u \in V(D) \setminus K \mid (v, u) \in A(D), v \in K\}$, où $K \subseteq V(D)$
 $N_D^+[K]$ $\{u \mid (v, u) \in A(D), v \in K\} \cup K$, où $K \subseteq V(D)$
 $d_H(u, v)$ Distance de u à v dans H , où H est un graphe ou un di-graphe.
 $u \prec v$ u précède v dans l'ordre donné.
 $e(v)$ excentricité de v .
 $d(v)$ degré de v dans un graphe.
 $d^+(v)$ degré sortant de v dans un di-graphe.
 $d^-(v)$ degré entrant de v dans un di-graphe.
 $\Delta(G)$ degré maximum d'un sommet dans le graphe G .
 $\delta(G)$ degré minimum d'un sommet dans le graphe G .
 $\Delta^-(D)$ degré entrant maximum d'un sommet dans le di-graphe D .
 $\Delta^+(D)$ degré sortant maximum d'un sommet dans le di-graphe D .
 $\delta^-(D)$ degré entrant minimum d'un sommet dans le di-graphe D .
 $\delta^+(D)$ degré sortant minimum d'un sommet dans le di-graphe D .

REMERCIEMENTS

J'aimerais particulièrement remercier Geña Hahn, mon directeur de maîtrise.
Ses conseils, sa patience et son soutien ont rendu ce mémoire possible.

INTRODUCTION

Dans un réseau informatique, un virus se propage, devenant très nuisible. Dans un complexe militaire, la sécurité a été compromise alors qu'un intrus s'est introduit dans le bâtiment. Qu'ont en commun ces situations ? Dans chaque cas, la situation peut être formulée comme un jeu de poursuite. Les variantes sont nombreuses et possèdent toutes des particularités intéressantes, mais dans cette étude, nous nous intéresserons plus particulièrement à une certaine forme de ces jeux qui représente bien la première situation. Ce cas où un groupe de policiers tentent d'attraper un voleur sur les sommets d'un graphe ou di-graphe, est connu sous le nom de **recherche**, mais plus connu sous le nom de **searching**. Dans ce jeu, les joueurs connaissent la position de tous les autres joueurs. Cette version du jeu a été étudiée en premier par Nowakowski et Winkler [17], ainsi que par Quilliot [18], de façon indépendante, pour des graphes implicitement réflexifs.

Afin de permettre au lecteur de se familiariser avec les notations et définitions nécessaires à la lecture de cet ouvrage, un survol des principales définitions de base sur les graphes sera fait dans les premières pages.

Une fois ces bases établies, le cœur du sujet sera abordé. Le jeu de poursuite qui nous intéresse sera expliqué en détail : les règles régissant les déplacements des joueurs et les conditions nécessaires pour qu'un joueur soit déclaré gagnant seront notamment expliquées. De plus, quelques théorèmes importants qui permettent de catégoriser les graphes en fonction du gagnant seront démontrés. Quelques autres variantes seront aussi survolées, mais très brièvement.

Viendront ensuite quelques opérations simples qu'il est possible d'effectuer sur des graphes, ainsi que l'analyse des graphes en résultant. Ces opérations n'avaient pas été étudiées auparavant dans le contexte d'un jeu de poursuite tel que celui qui sera étudié.

Plusieurs résultats intéressants provenant de divers articles sur les graphes à ponts et les graphes à cordes suivront dans une section qui leur est consacrée.

L'avant-dernier chapitre portera sur les tournois, une famille de graphes orientés. La majorité des résultats présentés dans ce chapitre sont inédits et seront par conséquent expliqués en détails, mais certaines propriétés bien connues seront aussi démontrées.

Finalement, la section la plus importante de cet ouvrage étudiera en profondeur un algorithme développé par G. Hahn et G. MacGillivray [14]. L'algorithme sera présenté en détail et plusieurs preuves viendront le justifier. Une partie de ces preuves sont directement tirées de l'article mentionné précédemment, tandis que l'autre partie est constituée de preuves dues à mon travail. Ces dernières viendront compléter celles de l'article. Plusieurs résultats de tests effectués sur différentes implémentations de cet algorithme seront aussi présentés.

CHAPITRE 1

NOTIONS DE GRAPHERS

Quoi que cette section devrait contenir suffisamment d'information pour permettre au lecteur de bien comprendre les chapitres suivants, certains ouvrages plus complets pourraient être utiles. Les livres *Graph Theory* [10] de R. Diestel et *Introduction to Graph Theory* [19] de Douglas B. West en sont deux, et devraient être consultés pour plus d'information.

Soit X et Y des ensembles d'objets, et $k \in \mathbb{N}$ avec $k \leq |X|$. Nous allons définir $\binom{X}{k}$, une collection de sous-ensembles de taille k de l'ensemble X :

$$\binom{X}{k} = \{ Y \subseteq X \mid |Y| = k \}$$

Un graphe **non-orienté** $G = (V, E)$ est composé d'un ensemble de sommets V et d'un ensemble E d'arêtes $\{u, v\}$, où $\{u, v\} \in E \subseteq \binom{V}{2}$. Son équivalent **orienté** $D = (V, A)$ est quant-à lui composé d'un ensemble de sommets V et d'un ensemble A d'arcs (u, v) où $(u, v) \in A \subseteq V^2 \setminus \{(v, v) \mid v \in V\}$. Nous noterons souvent $V(G)$ l'ensemble de sommets de G et $E(G)$ son ensemble d'arcs, en particulier lorsque plusieurs graphes seront utilisés. Pour un di-graphe, $A(G)$ représentera l'ensemble d'arcs. Afin d'alléger la lecture, nous considérerons qu'un **graphe** est un graphe non-orienté, et qu'un **di-graphe** est un graphe orienté.

Notons que les définitions précédentes ne permettent pas d'arêtes/arcs multiples entre deux sommets. Nous nous limiterons à ces définitions car tous les graphes/di-graphes de ce document sont **simples**, c'est-à-dire sans arêtes/arcs multiples. De plus, notons qu'ils seront finis, c'est-à-dire que la taille de $V(G)$ ou $V(D)$ est dans \mathbb{N} .

Un graphe est **complet** s'il contient toutes les arêtes possibles ($E = \binom{V}{2}$) et

nous dirons qu'il est **vide** s'il n'en contient aucune ($E = \emptyset$). Le graphe complet sur n sommets est noté K_n . Un di-graphe est dit **complet** si son graphe sous-jacent est complet et vide pour la même raison qu'un graphe. Lorsqu'il sera fait référence au **graphe sous-jacent** d'un di-graphe, ce sera pour désigner le graphe sur le même ensemble de sommet avec $\{u, v\} \in E$ si $(u, v) \in A$ ou $(v, u) \in A$ (voir Fig 1.1).

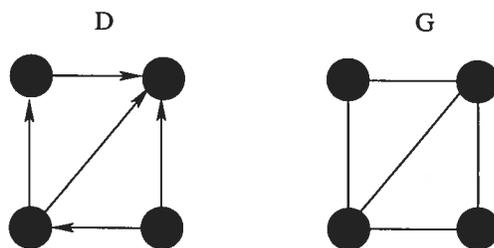


FIG. 1.1 – Un di-graphe D et son graphe sous-jacent G .

Les descriptions faites précédemment ne permettent pas de boucles sur les sommets. Nous appellerons ces graphes/di-graphes **non-réflexifs** (voir Fig 1.2). On élargit la définition en permettant les boucles, et ces graphes/di-graphes seront appelés **partiellement réflexifs**, et, s'il y a une boucle pour chaque sommet, **réflexifs** (voir Fig 1.3). Nous avons donc :

- $E \subseteq \binom{V}{2}$ pour un graphe non-réflexif
- $E \subseteq \binom{V}{2} \cup \{\{v, v\} \mid v \in V\}$ pour un graphe partiellement réflexif ou réflexif

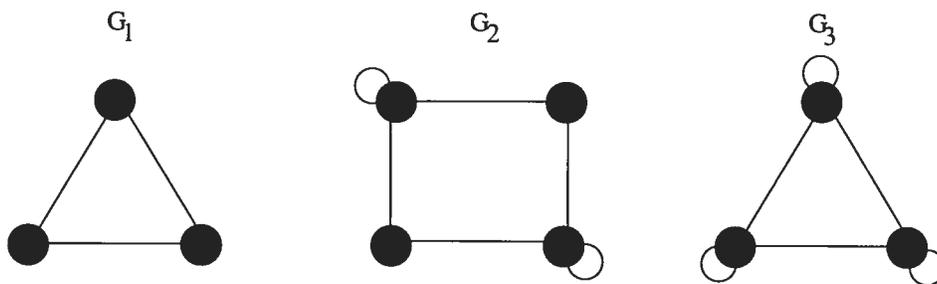


FIG. 1.2 – Un graphe non-réflexif G_1 , un graphe partiellement réflexif G_2 et un graphe réflexif G_3 .

- $A \subseteq V^2 \setminus \{(v, v) \mid v \in V\}$ pour un di-graphe non-réflexif.
- $A \subseteq V^2$ pour un di-graphe partiellement réflexif ou réflexif

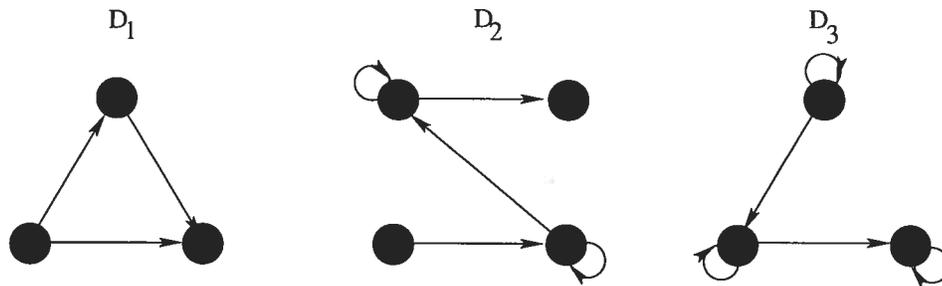


FIG. 1.3 – Un di-graphe non-réflexif D_1 , un di-graphe partiellement réflexif D_2 et un di-graphe réflexif D_3 .

1.1 Voisinage

Pour une arête $e = \{u, v\}$ ou un arc $a = (u, v)$, nous dirons que u et v sont **adjacents**, et que e et a sont **incidents** à u et v . De plus, dans le cas d'un arc, u est un **prédécesseur** de v , et ce dernier un **successeur** de u .

Le **voisinage** d'un groupe de sommets dans G est l'ensemble de sommets auxquels ses sommets sont adjacents. De façon générale, nous parlerons de **voisinage fermé** d'un groupe de sommets si on inclut celui-ci, et de **voisinage ouvert** sinon.

Si $K \subseteq V(G)$, alors dans un graphe :

- Voisinage ouvert : $N_G(K) = \{u \in V(G) \setminus K \mid \{v, u\} \in E(G), v \in K\}$
- Voisinage fermé : $N_G[K] = \{u \mid \{v, u\} \in E(G), v \in K\} \cup K$

Et dans le cas d'un di-graphe D :

- Voisinage entrant ouvert : $N_D^-(K) = \{u \in V(D) \setminus K \mid (u, v) \in A(D), v \in K\}$
- Voisinage entrant fermé : $N_D^-[K] = \{u \mid (u, v) \in A(D), v \in K\} \cup K$
- Voisinage sortant ouvert : $N_D^+(K) = \{u \in V(D) \setminus K \mid (v, u) \in A(D), v \in K\}$
- Voisinage sortant fermé : $N_D^+[K] = \{u \mid (v, u) \in A(D), v \in K\} \cup K$

Dans le cas où K ne contient qu'un seul sommet, par exemple $K = \{v\}$, et où un graphe G ou un di-graphe D aura été défini, nous noterons plus simplement les voisinages $N(v)$, $N[v]$, $N^-(v)$, $N^-[v]$, $N^+(v)$ et $N^+[v]$ respectivement.

Le **degré** $d(v)$ d'un sommet dans un graphe sera défini comme la taille de son voisinage ouvert, c'est-à-dire $d(v) = |N(v)|$. Pour un sommet dans un di-graphe, nous devons définir deux degrés, soit le **degré entrant** et le **degré sortant**. Le degré entrant est égal à la taille de $N^-(v)$ et est noté $d^-(v)$, tandis que le degré sortant est défini par $d^+(v) = |N^+(v)|$. Le **degré maximum** d'un graphe est défini par $\Delta(G) = \max\{d(v) \mid v \in V(G)\}$. Dans un di-graphe, nous parlerons plutôt de **degré entrant maximum** $\Delta^-(D)$ et de **degré sortant maximum** $\Delta^+(D)$. Et évidemment, dans le cas orienté comme non-orienté, des définitions analogues pour le minimum existent. Le **degré minimum** de G est noté $\delta(G)$, et ceux de D sont notés $\delta^-(D)$ et $\delta^+(D)$.

Un graphe où le degré de chaque sommet est le même est appelé **régulier**, de même que pour un di-graphe dont tous les sommets ont le même degré sortant et le même degré entrant. De plus, un di-graphe sera dit **presque régulier** si $|V(G)|$ est pair, et si pour tout sommet v , $\{d^+(v), d^-(v)\} = \{\lfloor |E(G)|/|V(G)| \rfloor, \lceil |E(G)|/|V(G)| \rceil\}$.

1.2 Chaînes et cycles

Une **chaîne** dans un graphe est une suite de sommets distincts $v_0v_1\dots v_k$ où $\{v_i, v_{i+1}\}$ est une arête dans E pour $0 \leq i < k$. La longueur d'une chaîne est le nombre d'arêtes qui la composent. Une chaîne est un **cycle** si $\{v_0, v_k\} \in E(G)$. Pour la version orientée, nous parlerons de **chemin**. La séquence de sommets distincts $v_0v_1\dots v_k$ est donc un chemin si $(v_i, v_{i+1}) \in A$ pour $0 \leq i < k$. La **longueur** d'un chemin est égale au nombre d'arcs dont il est composé. Si v_0 et v_k sont adjacents, nous parlerons d'un **circuit**. La notation C_n sera quelques fois utilisée pour

désigner un cycle de longueur n . De plus, un graphe formé par un cycle de longueur n , et d'un sommet central adjacent à tous les sommets du cycle, sera désigné par W_n .

Si, pour un cycle C dans un graphe G , $|C| = |V(G)|$, alors ce cycle et ce graphe seront dits **hamiltoniens**.

Dans le cas où entre toute paire de sommet d'un graphe il existe une chaîne, ce graphe sera dit **connexe**. S'il y a k chaînes distinctes, c'est-à-dire qui ont pour seuls sommets communs leurs extrémités, nous dirons qu'il est **k -connexe**. Pour un di-graphe, il sera qualifié de connexe si son graphe sous-jacent l'est. Si pour toute paire u, v de sommets de D , il existe un chemin de u à v , alors le di-graphe est **fortement connexe**.

1.3 Arbres, arborescences

Si un graphe ne contient pas de cycle, ce graphe est une **forêt**. Si de plus ce graphe est connexe, il est un **arbre**. Dans le cas d'un di-graphe, nous parlerons d'une **arborescence** si tous les sommets du di-graphe sont de degré entrant 1, à l'exception d'un sommet qui n'a aucun voisin entrant que nous appellerons **racine**. Notons que dans un arbre, une racine peut être définie de façon arbitraire et que chaque choix d'une telle racine définit alors implicitement une arborescence car il suffit d'orienter la chaîne unique (garantie par le lemme suivant) de la racine vers v , pour tout v .

Lemme 1.3.1. Un graphe G est un arbre si et seulement si pour chaque paire de sommets de G , il existe une chaîne unique qui relie ces sommets.

Démonstration. Supposons que G est un arbre. Alors il existe une chaîne unique entre chaque paire de sommets, car G est connexe et deux chaînes ou plus impli-

querait que G contient un cycle, ce qui contredirait la définition même d'un arbre. Supposons maintenant qu'il existe une chaîne unique entre chaque paire de sommet de G . De toute évidence, il ne peut exister aucun cycle. De plus, G est connexe car il existe une chaîne entre chaque paire de sommets. Donc G est bel et bien un arbre. \square

Soit r la racine d'un arbre, et $r = v_0v_1\dots v_n = v$ une chaîne reliant r à un sommet v . Cette chaîne existe et est unique par le lemme précédent. Pour $0 < i \leq n$, le sommet v_{i-1} est le **parent** de v_i , et v_i est un **enfant** de v_{i-1} . De plus, r et les sommets v_j tels que $0 \leq j < i$ sont des **ancêtres** de v_i et v , et nous dirons que v et v_i sont leurs **descendants**. Ces définitions sont aussi valides dans le cas orienté, si on remplace la chaîne $r = v_0v_1\dots v_n = v$ par un chemin. Dans tous les cas, un sommet sans enfant sera désigné par le terme **feuille**.

1.4 Distance

La **distance** entre une paire de sommets u, v dans un graphe G est notée $d_G(u, v)$ et est définie par la longueur d'une plus courte chaîne reliant u et v . Il est aisé de constater que $d_G(u, v) = d_G(v, u)$. La distance entre u et v dans un di-graphe sera définie par la longueur d'un plus court chemin de u à v . Il est important de remarquer que dans un di-graphe, la distance de u à v n'est pas nécessairement la même que celle de v à u . De façon générale, la notation $d(u, v)$ sera utilisée pour représenter $d_G(u, v)$ ou $d_D(u, v)$ lorsqu'un graphe G ou un di-graphe D aura été préalablement défini. Dans tous les cas, la distance d'un sommet à lui-même est toujours 0.

L'**excentricité** d'un sommet v , notée $e(v)$ est définie par $e(v) = \max\{d(v, u) \mid u \in V(G)\}$. L'excentricité maximum des sommets de V dans un graphe définit quant-à elle le **diamètre** du graphe.

Soit v un sommet d'un graphe connexe G . Alors pour $0 \leq i \leq e(v)$, nous définirons une partition des sommets de G :

$$V_i(v) = \{ u \mid u \in V(G), d(v, u) = i \}$$

1.5 Sous-graphes

Dans un graphe G , nous appellerons **sous-graphe** un sous-ensemble d'arêtes de E et les sommets qui les définissent.

Pour un sous-graphe H :

- $E(H) \subseteq E(G)$
- $V(H) = \{ u \mid \{u, v\} \in E(H) \}$

Un sous-graphe $H = (V', E')$ **induit** par V' dans $G = (V, E)$ est un graphe tel que $V' \subseteq V$ et $E' = \{ \{u, v\} \mid u \neq v \in V' \} \cap E$, et sera quelques fois noté $G(V')$. Dans le cas où $V = V'$ et $E' \subseteq \{ \{u, v\} \mid u, v \in V' \} \cap E$ nous parlerons d'un graphe **partiel**.

Nous allons maintenant définir quelques notations utilisées pour les sous-graphes de graphes. Soit $X \subseteq V(G)$, un sous-ensemble des sommets de G et $Y \subseteq E(G)$, un sous-ensemble des arêtes de G . On définit alors les sous-graphes suivants :

- $G - X = (V(G) \setminus X, E(G) \cap (V(G) \setminus X)^2)$. Ce sous-graphe est donc le sous-graphe induit sur les sommets de G sans les sommets de X . Nous utiliserons aussi $G - \{v\}$ ou $G - v$ dans le cas où X est constitué d'un seul sommet.
- $G - Y = (V(G), E(G) \setminus Y)$. Comme précédemment, nous utiliserons aussi $G - \{e\}$ ou $G - e$ dans le cas où Y est constitué d'une seule arête.

Si les distances entre les sommets d'un sous-graphe H sont les même que celles dans le graphe G , c'est-à-dire si $d_H(u, v) = d_G(u, v)$ pour toute paire u, v de sommets de H , ce sous-graphe sera dit **isométrique**.

Une **composante connexe** est un sous-graphe connexe d'un graphe ou di-graphe telle qu'aucun autre sous-graphe connexe ne la contient. On peut donc dire qu'une composante connexe est un sous-graphe connexe maximal par rapport à l'inclusion.

Une notion intéressante et utile pour ce travail est celle de convexité. Nous dirons qu'un sous-graphe H d'un graphe G est **convexe** si toutes les chaînes les plus courtes entre les paires de sommets de H sont contenues dans le sous-graphe. C'est-à-dire que pour toute paire de sommets $u, v \in V(H)$, et pour toute plus courte chaîne $u = x_0x_2...x_m = v$ dans G , $x_i \in V(H)$ pour $0 \leq i \leq m$.

Remarque 1.5.1. Les notations et définitions analogues à ceux précédentes pour les di-graphes sont quasi-identiques, il suffit de remplacer les arêtes par des arcs.

Soit H un sous-graphe induit d'un graphe G . Ce sous-graphe sera qualifié de **clique** s'il est complet, et d'**anti-clique** s'il est vide.

1.6 Homomorphismes

Soit G et H des graphes. Un **homomorphisme** est une fonction définie par $\phi : V(G) \rightarrow V(H)$, telle que si deux sommets sont adjacents dans G , alors leur images le sont dans H (voir Fig 1.4). Ou, plus formellement :

$$\{u, v\} \in E(G) \Rightarrow \{\phi(u), \phi(v)\} \in E(H)$$

Rappelons qu'une fonction $f : X \rightarrow Y$ est **surjective** si pour tout $y \in Y$, il existe $x \in X$ tel que $f(x) = y$. Nous dirons qu'elle est **injective** si $f(x) = f(x')$ implique que $x = x'$. Une fonction qui possède ces deux propriétés, sera qualifiée de **bijective**.

Deux graphes G et H sont dit **isomorphes**, noté $G \simeq H$, s'il existe un homomorphisme $\phi : V(G) \rightarrow V(H)$, tel que ϕ est bijectif et $\{u, v\} \in E(G)$ si et seulement si $\{\phi(u), \phi(v)\} \in E(H)$.

Une **rétraction** $\rho : V(G) \rightarrow V(H)$, où H est un sous-graphe de G , est un homomorphisme tel que pour tout $x \in V(H)$, $\rho(x) = x$ (voir Fig 1.4).

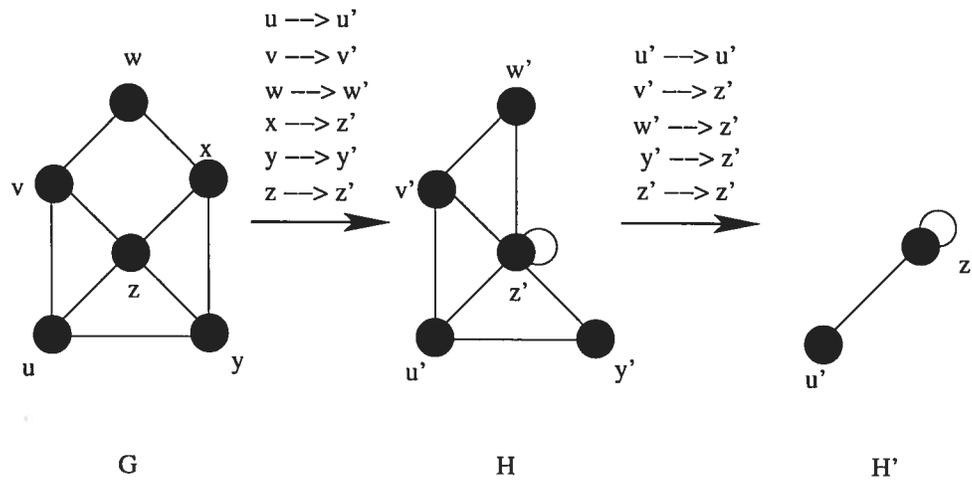


FIG. 1.4 – Un homomorphisme d'un graphe G vers un graphe H , et une rétraction sur ce dernier, résultant en H' .

CHAPITRE 2

JEUX DE POURSUITES

Comme il a été mentionné au début de cet ouvrage, de nombreuses versions du jeu de poursuite existent. Mais cet étude ne détaillera que le jeu de recherche. Ce jeu se déroule sur un graphe G et le fonctionnement est simple : les policiers choisissent chacun un sommet initial dans G , ces sommets n'étant pas obligatoirement distincts. Le voleur choisit ensuite son sommet initial. Le jeu peut maintenant commencer ! Chaque policier doit effectuer un déplacement à partir de son sommet actuel u et choisit par conséquent un sommet destination v tel que $\{u, v\} \in E(G)$. Le voleur fait ensuite de même, et le jeu continue ainsi en alternance. Les policiers sont déclarés vainqueurs s'ils réussissent à attraper le voleur, c'est-à-dire si à un certain moment, un policier occupe le même sommet que le voleur. Si le voleur peut éviter cette situation en tout temps, alors il sera déclaré vainqueur. Le jeu sur un di-graphe est le même, avec pour exception que les arcs ne permettent que des déplacements dans une direction, contrairement aux arêtes.

2.1 Variantes

2.1.1 Sweeping

Le jeu tel que décrit précédemment permet aux joueurs d'occuper les sommets, sans permettre l'occupation des arêtes/arcs. Un tel jeu est intéressant et a des applications réelles, mais dans le cas où un voleur et un policier seraient engagés dans une chasse à l'homme dans la vie de tous les jours, les joueurs pourraient se trouver sur un arc/arête, c'est-à-dire au beau milieu d'un couloir plutôt que seulement aux intersections, sans savoir la position de l'adversaire. Une autre version du jeu existe donc, appelée **sweeping**. Le but des policiers dans cette variante n'est pas de capturer le voleur uniquement sur les sommets, mais bien de "nettoyer" le graphe

ou di-graphe de jeu, en s'assurant qu'aucun voleur ne se cache sur les arêtes/arcs ou les sommets. Cette version a été étudiée dans nombre d'articles ([4], [3], [1]), mais étant donné la complexité de ce problème, plusieurs études et résultats portent sur les arbres plutôt que sur les graphes quelconques. Daniel Dyer [11] a quant-à lui fait une étude approfondie et obtenu plusieurs résultats intéressants pour les graphes et di-graphes dans sa thèse de doctorat.

2.1.2 Contraintes sur la mobilité

D'autres variantes consistent en la recherche sur des sommets, mais en donnant certaines contraintes sur les déplacements des policiers. Les policiers pourraient par exemple être contraints à occuper un sous-graphe connexe en tout temps. Ou bien pour la sécurité des policiers, il pourrait être nécessaire pour ceux-ci de ne jamais se déplacer seuls, c'est-à-dire d'avoir au moins un autre policier sur les sommets adjacents. La version **tandem** en est un exemple, et limite les déplacements de deux policiers de façon à ce qu'ils soient toujours à une distance inférieure à 2 l'un de l'autre. Clarke et Nowakowski ont étudié cette variante dans plusieurs articles tels que [8], [9].

2.1.3 Visibilité variable

Il est aussi possible de varier le degré de visibilité. Dans le cas habituel sur lequel nous travaillons, la visibilité est complète et les joueurs connaissent en tout temps la position des autres joueurs, tandis que dans le cas du sweeping, elle est nulle. Les règles peuvent être changées de façon à ce que les joueurs puissent voir à une distance de 1 par exemple, tel qu'étudié dans [16].

2.1.4 Autres

De nombreuses autres règles, dispositifs, et autres peuvent aussi être ajoutées. Certains, comme N. Clarke [7], se sont entre autres appliqués à étudier des situations où des alarmes et des caméras sont utilisées.

2.2 Définitions pour le jeu de recherche

Une **stratégie** pour k policiers est une fonction s définie par :

$$s : V(G)^{k+1} \rightarrow V(G)^k$$

C'est-à-dire une fonction qui associe à chaque configuration possible des positions des joueurs, la nouvelle configuration que les policiers doivent obtenir en se déplaçant. Donc, si les c_i représentent les positions des policiers et r celle du voleur, on ne peut avoir $f((c_1, \dots, c_k, r)) = (c'_1, \dots, c'_k)$ que si chaque policier peut se déplacer sur son nouveau sommet, donc que $\{c_i, c'_i\} \in E(G)$ ou $(c_i, c'_i) \in A(D)$, pour $0 \leq i \leq k$.

Une stratégie est **gagnante** pour les policiers s'il existe une position initiale qui permet à ceux-ci de capturer le voleur en un nombre fini de tours, quelle que soit la stratégie du voleur.

Définition 2.2.1. Un graphe/di-graphe G sera dit **policier-gagnant** si un policier possède une stratégie gagnante. Nous dirons **k -policiers-gagnant** dans le cas où k policiers sont suffisants pour avoir une telle stratégie. Tout graphe k -policiers-gagnant est donc aussi k' -policiers-gagnant si $k' \geq k$ (voir Fig 2.1).

Définition 2.2.2. Un graphe/di-graphe G sera dit **voleur-gagnant** si le voleur peut éviter d'être capturé par le groupe de k policiers. Tout graphe/di-graphe voleur-gagnant avec k policiers est donc aussi voleur-gagnant avec k' policiers si $0 \leq k' \leq k$.

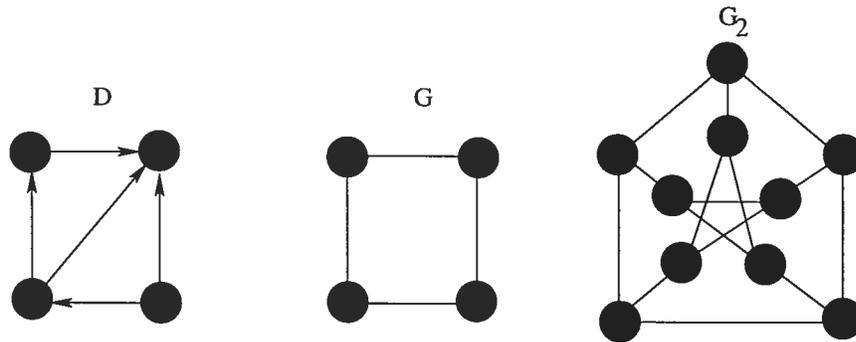


FIG. 2.1 – Un di-graphe D policier-gagnant, un graphe G 2-policiers-gagnant et G_2 , le graphe de Peterson, 3-policiers-gagnant.

Évidemment, les cas où un graphe/di-graphe n'est pas connexe sont moins intéressants. Si par exemple il y a 3 composantes connexes, le nombre de policiers nécessaires est égal à la somme des policiers nécessaires pour capturer le voleur sur chaque composante connexe. En effet, puisque les déplacements entre deux composantes connexes sont impossibles, il faut au minimum un policier sur chaque composante, autrement le voleur choisirait une composante sans policier comme position initiale et ne serait jamais capturé. De plus, le nombre de policiers sur chaque composante doit être suffisant pour capturer le voleur sur cette composante. Autrement, le voleur choisira une composante qui ne remplit pas cette condition et sera gagnant puisque les policiers ne peuvent changer de composante. C'est pour cette raison que nous ne nous intéresserons qu'aux cas connexes.

Voici un premier théorème utile à la caractérisation des graphes policier-gagnants (la définition du produit fort se trouve à la page 20) :

Théorème 2.2.3. (Nowakowski et Winkler [17]) Soit G le rétract d'un produit fort fini de graphes policier-gagnants. Alors G est policier-gagnant.

Démonstration. Nous allons premièrement montrer que le produit fort H de graphes

$\{G_1, G_2, \dots, G_n\}$ est policier-gagnant. Le policier peut utiliser sa stratégie sur G_1 pour attraper le voleur sur celui-ci. Il peut ensuite suivre les mouvements du voleur dans G_1 tout en utilisant sa stratégie sur G_2 pour capturer le voleur sur G_2 , et ainsi de suite. Il est donc possible au policier d'attraper le voleur sur H .

Soit G un rétract de H et f la rétraction telle que $f : H \rightarrow G$. Puisque H est policier-gagnant, le policier a une stratégie pour capturer le voleur. Le policier peut utiliser sur G l'image de sa stratégie sur H et ainsi capturer le voleur. \square

Définition 2.2.4. Dans un graphe G , un sommet u est appelé **dominé** si il existe un sommet v tel que $N[u] \subseteq N[v]$. Le sommet v sera quant-à lui appelé **dominant**.

Les graphes réflexifs sur lesquels un seul policier est gagnant ont été caractérisés par Nowakowski et Winkler dans les théorèmes suivants :

Théorème 2.2.5. (Nowakowski et Winkler [17]) Soit G un graphe réflexif fini et $u, v \in V(G)$ deux sommets distincts tels que $N[u] \subseteq N[v]$. Alors G est policier-gagnant si et seulement si $G' = G - \{u\}$ l'est aussi.

Démonstration. Supposons que G est policier-gagnant. Le sous-graphe G' peut être considéré comme un rétract de G puisque u peut être rétracté sur v . Donc G' est policier-gagnant.

Supposons maintenant que G' est policier-gagnant. Pour jouer sur G , le policier utilise la même stratégie que sur G' . Il est facile de voir que le policier n'a pas besoin de u car il lui suffit d'être sur v pour pouvoir atteindre n'importe quel sommet qu'il pourrait atteindre à partir de u . Il en est de même pour le voleur, mais supposons que celui-ci a besoin d'aller sur le sommet u . Cela signifie que le policier se trouve sur un des sommets du voisinage de v , autrement le voleur aurait pu choisir v . Donc au prochain tour, le policier se déplacera sur v et le voleur se retrouvera coincé puisque tous les voisins de u sont aussi voisins de v . \square

Théorème 2.2.6. (Nowakowski et Winkler [17]) Soit G un graphe réflexif. Alors G est policier-gagnant si et seulement si il existe un ordre des sommets v_1, v_2, \dots, v_n tel que pour $0 \leq i < n$, il existe $j > i$ tel que $N[v_j] \supseteq N[v_i] \cap \{v_i, \dots, v_n\}$. Un tel ordre sera dit **policier-gagnant**.

Démonstration. Supposons que G est policier-gagnant. Il doit donc exister deux sommets distincts u_1, v_1 tels que $N[v_1] \subseteq N[u_1]$, autrement le voleur pourrait toujours s'échapper. Par le théorème précédent, nous savons que $G - \{v_1\}$ est policier-gagnant. On peut donc y trouver une paire de sommets similaire u_2, v_2 , et continuer ainsi jusqu'à ce que le graphe restant soit vide. On obtient donc un ordre v_1, \dots, v_n tel que décrit précédemment.

Supposons maintenant qu'il existe bien un ordre v_1, v_2, \dots, v_n tel que décrit dans l'énoncé. Clairement, le graphe $G(\{v_n\})$ est policier-gagnant. De même pour $G(\{v_{n-1}, v_n\})$, par le théorème précédent puisque $G(\{v_n\}) = G(\{v_{n-1}, v_n\}) - \{v_{n-1}\}$ et qu'il existe un sommet qui domine v_{n-1} . De la même façon, il est possible d'affirmer que $G(\{v_{n-2}, v_{n-1}, v_n\})$ est policier-gagnant, et ainsi de suite. Le graphe G est donc policier-gagnant. \square

Théorème 2.2.7. (Nowakowski et Winkler [17]) Soit G un arbre partiellement réflexif. Alors G est policier-gagnant.

Démonstration. À chacun de ses déplacements, le policier choisi l'unique sommet (par le lemme 2.3.1) qui lui permet de s'approcher du voleur. Comme l'arbre ne contient aucun cycle, le sous-graphe où le voleur peut se déplacer est réduit à chaque tour, et celui-ci ne peut donc s'échapper.

Il est aussi possible de prouver le théorème à l'aide du théorème qui le précède, dans le cas où les sommets internes (qui ne sont pas des feuilles) possède tous une boucle. Soit v_1 une feuille de G , et plaçons celle-ci au début de l'ordre. Puisque la feuille n'a qu'un seul voisin et que celui-ci sera plus loin dans l'ordre, la feuille sera bien dominée. Soit v_2 une feuille dans l'arbre $G - \{v_1\}$. Elle est placée après v_1 dans

l'ordre, et il est donc possible de construire un ordre tel que décrit précédemment en continuant ainsi. G est donc policier-gagnant. \square

CHAPITRE 3

OPÉRATIONS SUR LES GRAPHES

3.1 Union sur un sous-graphe commun

Soit H, H' des graphes partiellement réflexifs, et K un sous-graphe induit commun tel que $V(H) \cap V(H') = K \neq \emptyset$. Nous pouvons donc considérer le graphe G comme l'union de deux graphes sur leur sous-graphe commun K , c'est-à-dire que $V(G) = V(H) \cup V(H')$ et $E(G) = E(H) \cup E(H')$ (voir Fig 3.1).

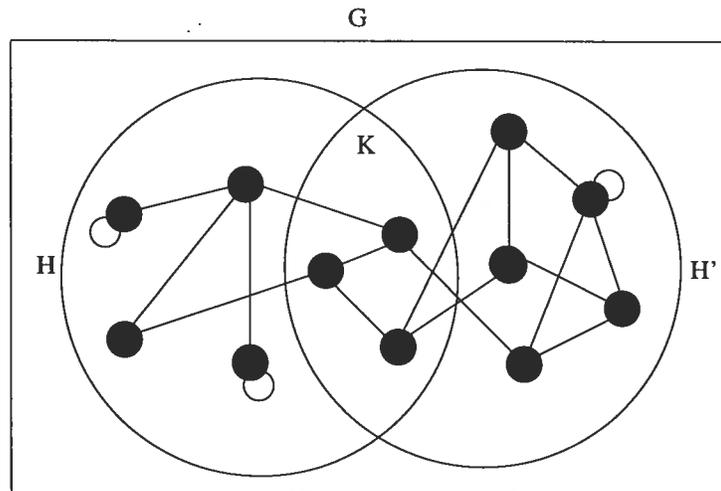


FIG. 3.1 – Exemple d'une union sur un sous-graphe commun

Nous allons définir deux ensembles :

- $Y = \{h'_i \in K \mid N_G(h'_i) \cap V(H') \setminus V(K) \neq \emptyset\}$.
- $X = \{h'_i \in K \mid N_G(h'_i) \cap V(H') \setminus V(K) = \emptyset\}$.

Proposition 3.1.1. Le graphe G est policier-gagnant si les conditions suivantes sont respectées :

- Il existe des ordres policier-gagnants $O_H = (h_1, h_2, \dots, h_n)$ et $O_{H'} = (h'_1, h'_2, \dots, h'_m)$ tels que pour $h_i = h'_s, h_j = h'_t \in K$ et $h_i, h_j \in O_H$, $h'_s, h'_t \in O_{H'}$, on a $i < j$ si et seulement si $s < t$.
- $h_i \in Y$ si $i \geq n + 1 - |Y|$

Démonstration. Nous allons prouver que l'ordre suivant est policier-gagnant :

$$O_G = (h_1, h_2, \dots, h_{n-|Y|}, h'_{i_1}, h'_{i_2}, \dots, h'_{i_y})$$

où $h'_{i_1}, \dots, h'_{i_y}$ est l'ordre $O_{H'}$ limité à $H' - X$. On peut voir que l'on retrouve l'ordre de O_H dans cet ordre. Les sommets $h_1, h_2, \dots, h_{n-|Y|}$ n'ont pas de voisins dans $H' \setminus K$, et ils seront donc couverts par le même sommet que dans O_H où par le même sommet dans $O_{H'}$.

Il reste donc à vérifier que tous les sommets de $h'_{i_1}, h'_{i_2}, \dots, h'_{i_y}$ sont couverts. Nous savons que tous les sommets de X n'ont aucun voisin dans $H' \setminus K$, ils ne peuvent donc couvrir un sommet h'_{i_s} . Donc les sommets h'_{i_s} sont bien couverts même sans les sommets de X . □

3.2 Construction de Hirsh

Soit H et H' deux graphes partiellement réflexifs tels que $V(H) \cap V(H') = \emptyset$. La construction de Hirsh consiste à prendre une paire de sommets adjacents dans chacun des graphes, h_1, h_2 et h'_1, h'_2 . Le nouveau graphe G est ensuite défini de la façon suivante :

$$\begin{aligned} - V(G) &= \{v\} \cup V(H) \cup V(H') \setminus \{h_1, h'_1\} \\ - E(G) &= \{\{h_2, h'_2\}\} \cup E(H) \cup E(H') \setminus \{\{h_1, h_2\}, \{h'_1, h'_2\}\} \end{aligned}$$

Évidemment, puisque le sommet v remplace les sommets h_1 et h'_1 , les arêtes $\{u, h_1\} \in H$ et $\{u', h'_1\} \in H'$ deviendront plutôt $\{u, v\}$ et $\{u', v\}$ (voir Fig 3.2).

Proposition 3.2.1. Soit H et H' deux graphes policier-gagnants. On construit un nouveau graphe G avec la construction de Hirsh. Alors le graphe résultant G n'est pas policier-gagnant.

Démonstration. Soit W_H une plus courte chaîne reliant h_1 et h_2 dans $H - \{h_1, h_2\}$ et $W_{H'}$ une plus courte chaîne reliant h'_1 et h'_2 dans $H' - \{h'_1, h'_2\}$. Les chaînes W_H , $W_{H'}$ et $\{h_2, h'_2\}$ forment un cycle dans G . On sait qu'il n'y a pas de corde entre w et h_2 , ainsi qu'entre w et h'_2 . Donc il n'y a pas de corde entre les sommets de W_H et $W_{H'}$ car les sommets de H et H' ne sont pas reliés sauf pour ceux choisis. Donc le cycle est convexe et de longueur au moins 5. Donc G n'est pas policier-gagnant. \square

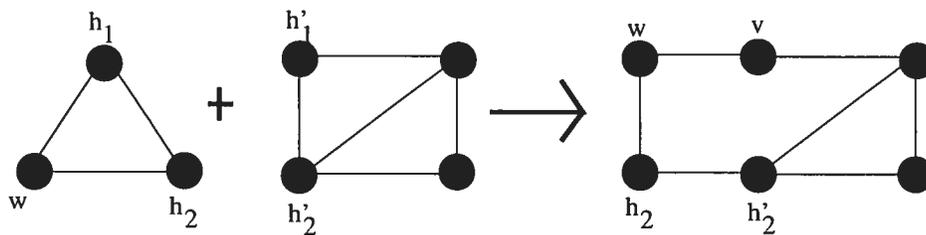


FIG. 3.2 – Exemple d'une construction de Hirsh

3.3 Produits de graphes non-réflexifs

3.3.1 Produit cartésien

Soit H_1 et H_2 deux graphes. Nous allons définir le produit cartésien (voir Fig 3.3), qui sera noté $G = H_1 \square H_2$, par :

- $V(G) = V(H_1) \times V(H_2)$
- $\{(u_1, u_2), (v_1, v_2)\} \in E(G)$ si l'un des 2 cas suivants est vrai :
 - $u_1 = v_1$ et $\{u_2, v_2\} \in E(H_2)$
 - $u_2 = v_2$ et $\{u_1, v_1\} \in E(H_1)$

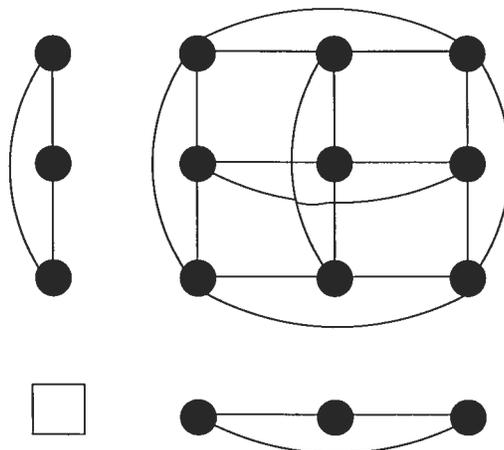


FIG. 3.3 – Exemple d'un produit cartésien $K_3 \square K_3$

Proposition 3.3.1. Soit H_1 et H_2 des graphes non-triviaux. Alors le graphe G défini par le produit $H_1 \square H_2$ n'est pas policier-gagnant.

Démonstration. Si H_1 ou H_2 ne contient pas d'arêtes, alors G n'est pas connexe et donc voleur-gagnant. Si au contraire les deux graphes possèdent au moins une arête, alors G contiendra un 4-cycle convexe. Donc, G est par le fait même voleur-gagnant. \square

3.3.2 Produit fort

Soit H_1 et H_2 deux graphes. Nous allons définir le produit fort (voir Fig 3.4), qui sera noté $G = H_1 \boxtimes H_2$, par :

- $V(G) = V(H_1) \times V(H_2)$
- $\{(u_1, u_2), (v_1, v_2)\} \in E(G)$ si l'un des 3 cas suivants est vrai :
 - $u_1 = v_1$ et $\{u_2, v_2\} \in E(H_2)$
 - $u_2 = v_2$ et $\{u_1, v_1\} \in E(H_1)$
 - $\{u_1, v_1\} \in E(H_1)$ et $\{u_2, v_2\} \in E(H_2)$

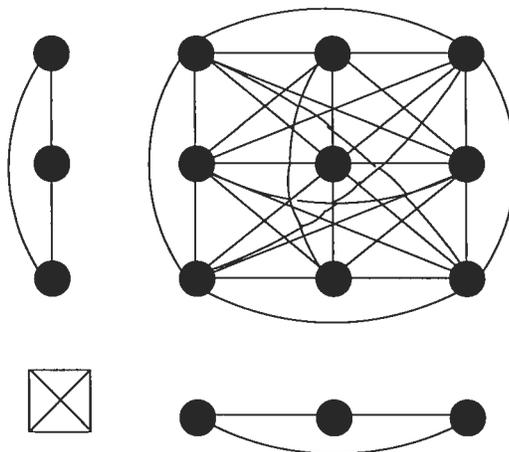


FIG. 3.4 – Exemple d'un produit fort $K_3 \boxtimes K_3$

Proposition 3.3.2. Soit H_1 et H_2 des graphes non-triviaux. Alors, le graphe G défini par le produit $H_1 \boxtimes H_2$ est policier-gagnant si H_1 et H_2 le sont.

Démonstration. Supposons que H_1 et H_2 sont policier-gagnants. Le policier peut commencer par poursuivre le voleur sur les sommets de H_1 , jusqu'à ce que la position du policier soit (u, x) et celle du voleur (u, y) . Une fois cet état atteint, le policier peut utiliser sa stratégie sur H_2 , tout en suivant les mouvements du voleur dans H_1 . \square

3.3.3 Produit catégorique

Soit H_1 et H_2 deux graphes. Nous allons définir le produit catégorique (voir Fig 3.5), qui sera noté $G = H_1 \times H_2$, par :

- $V(G) = V(H_1) \times V(H_2)$
- $\{(u_1, u_2), (v_1, v_2)\} \in E(G)$ si $\{u_1, v_1\} \in E(H_1)$ et $\{u_2, v_2\} \in E(H_2)$

Proposition 3.3.3. Soit H_1 et H_2 des graphes. Alors, le graphe $G = H_1 \times H_2$ n'est pas policier-gagnant.

Démonstration. Si H_1 ou H_2 a moins de 3 sommets ou n'est pas connexe, alors G

n'est pas connexe et est donc voleur-gagnant. Si les deux graphes sont connexes et de taille supérieure à 3, alors il existe $x_1, y_1, z_1 \in H_1$, et $x_2, y_2, z_2 \in H_2$ tels que $\{x_1, y_1\}, \{y_1, z_1\} \in E(H_1)$ et $\{x_2, y_2\}, \{y_2, z_2\} \in E(H_2)$. Donc, dans G , les sommets $(y_1, x_2), (z_1, y_2), (y_1, z_2), (x_1, y_2)$ formeront un 4-cycle convexe, et par conséquent G ne peut pas être policier-gagnant. \square

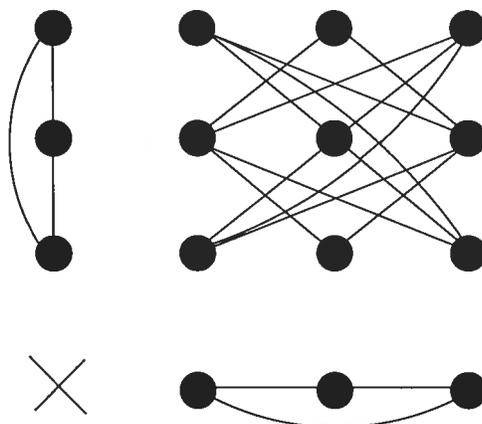


FIG. 3.5 – Exemple d'un produit catégorique $K_3 \times K_3$

3.4 Produits de graphes réflexifs

Les trois produits définis précédemment peuvent aussi être appliqués sur des graphes réflexifs, mais d'une façon un peu différente. Car si les produits ne sont pas redéfinis, les résultats seront quelque peu bizarres, comme l'illustre l'exemple de ci-dessous (voir Fig 3.6).

Soit H_1, H_2 des graphes réflexifs, et H'_1, H'_2 ces mêmes graphes sans les boucles.

Les produits sur H_1 et H_2 seront définis par :

- $V(H_1 \times H_2) = V(H_1 \square H_2) = V(H_1 \boxtimes H_2) = V(H_1) \times V(H_2)$
- $E(H_1 \times H_2) = E(H'_1 \times H'_2) \cup \{(x, y), (x, y) \mid x \in V(H_1), y \in V(H_2)\}$
- $E(H_1 \square H_2) = E(H'_1 \square H'_2) \cup \{(x, y), (x, y) \mid x \in V(H_1), y \in V(H_2)\}$

$$- E(H_1 \boxtimes H_2) = E(H'_1 \boxtimes H'_2) \cup \{(x, y), (x, y) \mid x \in V(H_1), y \in V(H_2)\}$$

Les graphes résultant de ces produits ne sont pas policier-gagnants, sauf pour le produit fort, car il est facile de voir que la réflexivité n'a aucun effet sur les preuves faites précédemment.

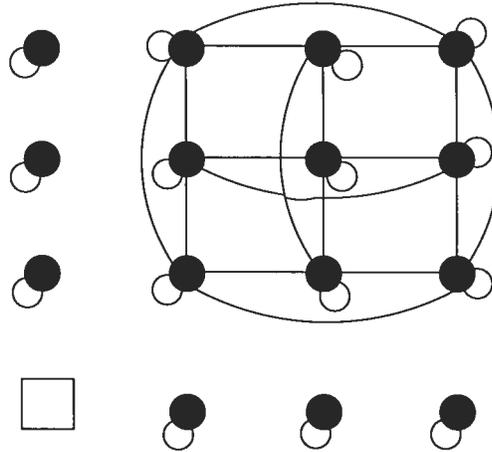


FIG. 3.6 – Exemple d'un produit catégorique sur des graphes réflexifs sans modification de la définition du produit.

CHAPITRE 4

GRAPHES À PONTS ET À CORDES

Nous allons commencer par définir deux types de graphes :

Définition 4.0.1. Si un graphe G ne contient aucun cycle C isométrique de longueur supérieure à 3, alors G est un **graphe à ponts**. C'est-à-dire que pour tout cycle C de G de longueur supérieure à 3, il existe deux sommets u, v du cycle C tels que $d_G(u, v) < d_C(u, v)$.

Définition 4.0.2. Si dans un graphe G tout cycle C de longueur supérieure à 3 contient deux sommets $u, v \in C$ tels que $\{u, v\} \notin E(C)$ et $\{u, v\} \in E(G)$, alors on dit que G est un **graphe à cordes**

Remarque 4.0.3. Les graphes à cordes et à ponts de ce chapitre seront tous connexes.

4.1 Propriétés des graphes à cordes

Proposition 4.1.1. (Diestel [10]) Si un graphe G est à cordes, alors tout sous-graphe induit de G est aussi à cordes.

Démonstration. Soit H un sous-graphe induit de G et C un cycle isométrique de longueur supérieure à 3 dans H . Ce cycle existe aussi dans G et doit donc avoir au moins une corde dans G sinon G ne serait pas à cordes. Cette corde est aussi dans H puisque H est induit, et C n'est pas isométrique. Contradiction. \square

Définition 4.1.2. Un sommet v dans un graphe G est **simplicial** si $N(v)$ est une clique.

Nous allons maintenant définir un nouveau type de sous-graphe qui sera utile pour les propositions à venir. Soit G un graphe, r un de ses sommets, et X une

composante connexe de $V_{e(r)}(r)$. Nous dirons que G' est le graphe induit par V' , où V est le groupe de sommets formé par r , X et tous les sommets se trouvant dans une plus courte chaîne entre r et un sommet de X .

Remarque 4.1.3. Dans le graphe G' , pour un sommet r et pour $0 < i \leq e(r)$, $V'_{i-1}(r) \cap N_{G'}(V'_i(r)) = V'_{i-1}(r)$ car si un sommet de V'_{i-1} n'a pas de voisin dans V'_i , il n'est pas dans une plus courte chaîne entre r et X , ce qui contredit la définition de G' .

Proposition 4.1.4. (West [19]) Soit G un graphe à cordes, r un de ses sommets et G' un sous-graphe tel que décrit précédemment. Alors dans G' , pour $0 \leq i \leq e(r)$, $V'_i(r)$ est une clique.

Démonstration. Soit $u, v \in V'_i(r)$. Pour $i = 0$ et $i = e(r)$, nous savons que $V'_i(r)$ est une clique. Sinon, il existe une chaîne entre u et v de longueur au moins 2 qui passe par $V'_{i-1}(r)$, et il en est de même pour $V'_{i+1}(r)$. Les sommets u et v doivent donc être adjacents, autrement les deux chaînes formeraient un cycle sans ponts de longueur supérieure ou égale à 4. \square

Proposition 4.1.5. (West [19]) Soit G un graphe à cordes et r un sommet de ce graphe. Alors il existe un sommet simplicial dans les sommets les plus éloignés de r , c'est-à-dire dans $V_{e(r)}(r)$.

Démonstration. Nous allons démontrer que toute composante connexe X de $V_{e(r)}(r)$ contient un sommet simplicial. Pour ce faire, nous allons montrer que $V'_{e(r)}(r)$ dans G' contient un sommet simplicial. Un tel sommet n'a pas de voisin dans $V_{e(r)}(r)$ autre que ceux de X , ni dans $V_{e(r)-1}(r)$ autre que ceux de $V'_{e(r)-1}(r)$ car ce voisin serait sur un plus court chemin entre X et r , et serait donc dans G' . Un sommet de $V'_{e(r)}(r)$ simplicial dans G' l'est donc aussi dans G .

Nous ferons la preuve de notre hypothèse par induction. Pour $|V'| = 2$, la preuve est triviale.

Supposons donc la proposition vraie pour les graphes de taille inférieure à $|V'|$. Si $N_{G'}[V'_{e(r)}(r)]$ est une clique, alors il est évident que $V'_{e(r)}(r)$ contient un sommet simplicial, car pour tout $v \in V'_{e(r)}(r)$, $N_{G'}(v)$ est une clique. Sinon, il doit exister deux sommets $u, v \in N_{G'}[V'_{e(r)}(r)]$ tels que $d_{G'}(u, v) = 2$. De plus, au moins un de ces sommets, supposons v , doit être dans $V'_{e(r)}(r)$ et l'autre dans $V'_{e(r)-1}(r)$ par la proposition 5.1.4. Par l'hypothèse, nous savons qu'un des sommets à distance 2 de u sera simplicial dans $N_{G'}[V'_{e(r)}(r)]$, et par conséquent, simplicial dans G' et dans G . \square

Définition 4.1.6. Dans un graphe G , un **ordre d'élimination simplicial** est un ordre des sommets $\sigma = v_1, v_2, \dots, v_n$ tel que v_j est simplicial dans $G \setminus \{v_1, \dots, v_{j-1}\}$.

Proposition 4.1.7. (West [19]) Un graphe G est à cordes si et seulement si il existe un ordre d'élimination simplicial $\sigma = v_1, v_2, \dots, v_n$.

Démonstration. Soit v un sommet de G . Puisque que nous savons qu'il existe un sommet simplicial dans les sommets les plus éloignés de v , nous pouvons choisir ce sommet u comme premier de l'ordre. Ensuite, il existe un sommet simplicial dans les sommet les plus éloignés de v dans $G \setminus \{u\}$ et ce sommet sera le deuxième de l'ordre. En continuant de cette façon, un ordre d'élimination simplicial est construit. Supposons qu'il existe un ordre d'élimination simplicial, et que le graphe n'est pas à cordes. Il existe donc un cycle C isométrique de longueur supérieure à 3. Or, il existe des sommets $v_i, v_j, v_k \in C$, $i < j < k$, tels que $v_j, v_k \in N(v_i)$. Mais nous savons que le voisinage de v_i est une clique dans $G \setminus \{v_1, v_2, \dots, v_{i-1}\}$. Donc v_j et v_k sont adjacents et forment une corde, alors que nous avons supposé que C est isométrique. Contradiction. \square

4.2 Propriétés des graphes à ponts

Proposition 4.2.1. (Farber et Jamison [12]) Un graphe G est à ponts si et seulement si pour tout ensemble de sommets convexe K , $N[K]$ est convexe.

Démonstration. Supposons que pour tout $K \subseteq V(G)$ convexe, alors $N[K]$ est convexe, mais que G n'est pas à ponts. Il existe donc un cycle isométrique C de longueur au moins 4. Posons $|C| = n$ avec $n \geq 4$. Soit $c, d \in C$ deux sommets opposés dans C , c'est-à-dire que $d_C(u, v) = \lfloor n/2 \rfloor$. Considérons $H = N_{\lceil \frac{n-1}{4} \rceil}[c]$. Cet ensemble H contient au moins la moitié des sommets de C , sans contenir d . Autrement cela impliquerait que le cycle n'est pas isométrique. L'ensemble H contient donc deux sommets u, v tels que $d(u, c) = d(v, c) = \lceil \frac{n-1}{4} \rceil$. Donc, $d_H(u, v) \geq n/2$. Or, il y a un plus court chemin entre u et v qui passe par d mais $d \notin H$. Contradiction car H devrait être convexe.

Prouver l'autre direction de la proposition nécessite beaucoup de travail. Nous laisserons au lecteur le soin d'aller consulter la référence s'il le désire. \square

Proposition 4.2.2. (Chepoi [6]) Soit G un graphe à ponts et un ordre v, v_1, \dots, v_{n-1} des sommets de G tel que pour $1 \leq i < j \leq n$, $d(v, v_i) \leq d(v, v_j)$, c'est-à-dire un ordre obtenu par un algorithme qui parcourt les sommets en largeur d'abord. Soit u et w , avec $u \prec w$, des sommets adjacents tels que $d(v, u) = d(v, w) = k$. Soit $x, y \in V(G)$ tels que x et y sont respectivement les parents de u et w dans l'ordre. Alors ou bien $x = y$, ou bien x et y sont adjacents. De plus, y est adjacent à u .

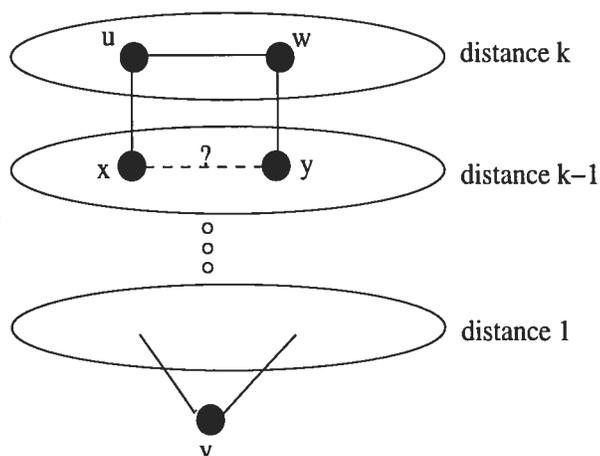


FIG. 4.1 – Graphe G

Démonstration. Nous allons effectuer la preuve par induction sur la distance $k = d(v, u) = d(v, w)$. Si $k = 1$ alors $v = x = y$ et y est bien adjacent à u .

Supposons maintenant que x et y ne sont ni adjacents ni égaux. La distance entre eux est donc supérieure à 1. Or, nous savons que $N_{k-1}[v]$ est convexe, ce qui nous permet de déduire que $d(x, y) = 2$, autrement u et w seraient dans une plus courte chaîne et cela impliquerait que $u, w \in N_{k-1}[v]$. De plus, $uy, wx \notin E(G)$ car u ou w serait encore dans une plus courte chaîne.

Soit z un sommet adjacent à x et y . Puisque G ne contient aucun 4-cycle ou 5-cycle induit, z doit être adjacent à u et w , et $d(v, z) = k - 1$.

Considérons maintenant x', z', y' les parents de x, z, y respectivement. Quatre cas sont possibles :

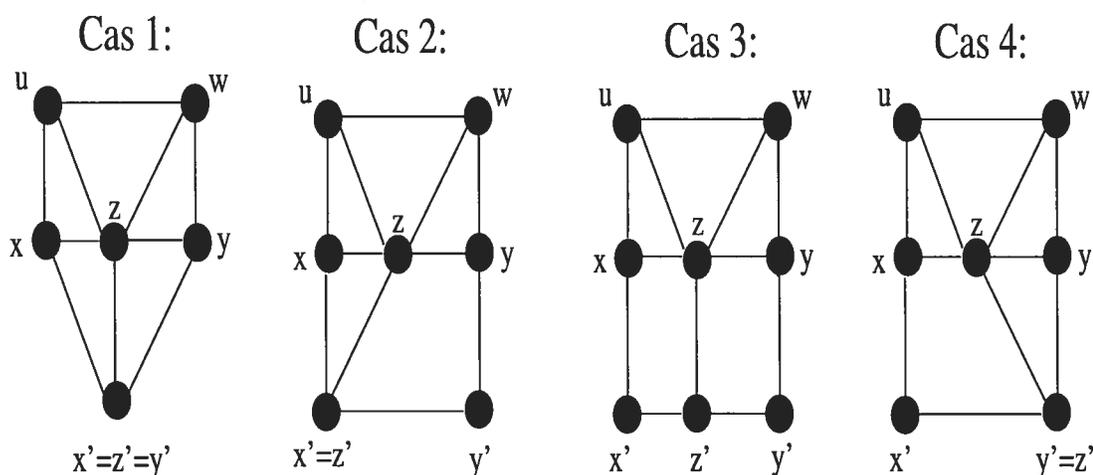


FIG. 4.2 – Illustration des quatre cas possibles

- *Cas 1* : Les sommets x, u, w, y, x' forment un 5-cycle induit. Contradiction car G est un graphe à ponts.
- *Cas 2* : Puisque $x' = z'$, nous avons $z' \prec y'$, ce qui implique que $z \prec y$ et que z est le parent de w . Contradiction.
- *Cas 3* : De par notre hypothèse de départ, $x \prec y \prec z$, autrement z serait

le parent de u ou w . Puisque nous savons que la proposition est vraie pour $k - 1$, cela implique que $x, y \in N(z')$. Un 5-cycle est donc créé comme dans le premier cas. Contradiction.

- *Cas 4* : Pour éviter que $x'xzz'$ ne forment un 4-cycle isométrique, il faut que x et z' soient adjacents ou bien que x' et z le soient. Si $\{x, z'\} \in E(G)$, alors $xuwyz'$ est un 5-cycle isométrique. Tandis que x' ne peut être adjacent à z car il serait son parent. Contradiction.

□

Les théorèmes précédents permettent d'obtenir le résultat suivant :

Théorème 4.2.3. (Anstee et Farber [2]) Tout graphe à ponts G est policier-gagnant.

Démonstration. La preuve qui suit est inspirée de Chepoi [6]. Soit v, v_1, \dots, v_{n-1} un ordre des sommets de G obtenu par un algorithme qui parcourt les sommets en largeur d'abord. Nous allons montrer que tout sommet $w = v_i$, à distance k de v , est dominé par son parent y dans le graphe induit par v, v_1, \dots, v_i . Soit u un voisin de w dans le graphe induit par v, v_1, \dots, v_i . Si $d(v, w) = d(v, u)$ alors par la proposition précédente, y est le parent de u , soit y et u sont adjacents. Sinon, $d(v, y) = d(v, u)$, et dans ce cas, y et u sont dans $N_{k-1}[v]$ qui est convexe. Mais w n'est pas inclus dans cet ensemble et par conséquent y et u doivent être adjacents pour ne pas contredire la convexité. Chaque sommet v_i est donc dominé par son parent dans le graphe induit par v, v_1, \dots, v_i et il suffit d'inverser cet ordre pour obtenir un ordre policier-gagnant. □

Le théorème suivant est très important pour les graphes à ponts car il permet de déduire de nombreuses particularités de ceux-ci. Il permet notamment de déduire facilement que les graphes à ponts sont policier-gagnants. Étant donné la complexité de la preuve, celle-ci ne sera pas détaillée ci-dessous.

Théorème 4.2.4. (Anstee et Farber [2]) Tout graphe à ponts non-trivial contient

une paire de sommets u, v telle que $N[u] \subseteq N[v]$.

Corollaire 4.2.5. (Anstee et Farber [2]) Tout graphe à ponts G contient un sommet $u \in V(G)$ tel que $G - u$ est isométrique. De plus, $G - u$ est à ponts.

Démonstration. Ce résultat est obtenu directement du théorème précédent. Les distances sont conservées en l'absence de u car toute chaîne contenant u peut être modifiée en substituant v à u . Puisque $G - u$ est isométrique, tout cycle dans $G - u$ est isométrique dans G . Il ne peut donc y avoir de cycle isométrique de longueur supérieure à 3 dans $G - u$. \square

Les résultats précédents combinés avec quelques autres résultats permet d'obtenir le théorème suivant :

Théorème 4.2.6. (Anstee et Farber [2]) Un graphe G est à ponts si et seulement si il est policier-gagnant et ne contient pas W_4 ou W_5 induits.

Démonstration. Si G est à ponts, il ne peut contenir W_4 ou W_5 , et il a été prouvé précédemment qu'un tel graphe est policier-gagnant.

Supposons que G est policier-gagnant et ne contient pas W_4 ni W_5 . Il est donc impossible que G contienne un cycle isométrique de longueur supérieure à 3 car autrement il serait voleur-gagnant. \square

CHAPITRE 5

TOURNOIS

Définition 5.0.7. Un **tournoi** est un graphe orienté tel que pour chaque paire de sommets, il existe un, et un seul, arc entre eux. Ou, de façon équivalente, on peut dire qu'un tournoi est obtenu à partir d'un graphe complet en donnant une orientation à chacune de ses arêtes.

5.1 Propriétés générales

Voici une liste de propriétés sur les tournois, valides pour les tournois partiellement réflexifs. Ces propriétés sont bien connues et sont mentionnées dans plusieurs articles ou livres tel que celui de Douglas B. West [19]

Définition 5.1.1. Dans un tournoi T , un sommet k est appelé **roi** si pour tout sommet $v \in V(T)$, $d(v, k) \leq 2$. Un **roi** k est un **empereur** si $N^+[k] = V(T)$.

Proposition 5.1.2. Si k est un sommet de degré sortant maximum dans un tournoi T , alors k est un roi.

Démonstration. Supposons qu'il existe un sommet v tel que $d(k, v) > 2$. Donc, $u \in N^+(v)$ pour tout $u \in N^+(k)$, et le degré sortant de v est supérieur à celui de k car v a aussi k dans son voisinage sortant. Contradiction. \square

Proposition 5.1.3. Dans un tournoi T , un sommet k est un empereur si et seulement si k est l'unique roi dans le tournoi.

Démonstration. Nous allons faire la démonstration par induction sur la taille n du tournoi. Pour $n = 1$, il est évident que la proposition est vraie.

Soit un tournoi T de taille n . Supposons qu'il existe un seul sommet k tel que k est

un roi, mais que k n'est pas un empereur. Soit $H = V(T) \setminus N^+[k]$. Ce sous-graphe contient au moins un sommet, il existe donc $k' \in H$ tel que k' est un roi dans le sous-graphe induit par H . Mais k' est aussi un roi dans T puisqu'il est un roi dans le sous-graphe induit par H , et est adjacent à k et donc à distance 2 de $N^+(k)$. Contradiction.

Si k est un empereur, alors il est clair qu'il ne peut exister un autre roi car aucun autre sommet ne peut l'atteindre. \square

Théorème 5.1.4. Tout tournoi T contient un chemin hamiltonien.

Démonstration. Soit v_0 le sommet de degré sortant maximal dans T . Ce sommet est choisi en premier. On choisit ensuite le sommet v_1 de $N^+(v_0)$ tel que la taille de $N^+(v_1) \cap V(T) \setminus \{v_0\}$ est maximale. On peut donc continuer ainsi pour parcourir tous les sommets et obtenir un chemin hamiltonien $v_0v_1\dots v_n$. Si à une étape quelconque le voisinage sortant du sommet v_i est vide, cela signifie que $i = n$, ou bien qu'il existait un sommet v_j dans le voisinage de v_{i-1} avec une taille de $N^+(v_j) \cap V(T) \setminus \{v_0, v_1, \dots, v_{i-1}\}$ supérieure lorsque ce sommet a été choisi. \square

Théorème 5.1.5. (A. Hill [15]) Un tournoi réflexif ou non-réflexif T est policier-gagnant si et seulement si T contient un empereur.

Démonstration. Si le tournoi ne contient pas d'empereur, le voleur choisit comme sommet de départ un des sommets qui ne se trouve pas dans le voisinage sortant du sommet où se trouve le policier. Ensuite, lorsque le policier se déplace, le voleur peut le suivre en allant sur le sommet qu'il vient de quitter. Le voleur ne sera donc jamais attrapé. \square

Proposition 5.1.6. Si k est un roi et si le degré de v dans le sous-graphe induit par $V(G) \setminus N^+[k]$ est maximal, alors v est un roi dans T .

Démonstration. Nous savons que v est à une distance de 1 ou 2 de tous les sommets dans $V(G) \setminus N^+[k]$ car il est roi dans ce sous-graphe. Mais $k \in N^+(v)$, donc v est à distance 1 de k et à distance 2 de tous les voisins de k . Donc v est bel et bien un roi puisqu'il se trouve à une distance d'au plus 2 de tous les autres sommets. \square

Proposition 5.1.7. Tout tournoi T sans empereur contient un circuit composé de rois.

Démonstration. Avec la proposition précédente, il est clair que tout roi qui n'est pas un empereur a comme prédécesseur au moins un roi. Comme le tournoi est fini, un circuit sera formé. \square

5.2 Tournois réflexifs

Soit v un sommet de degré maximum dans un tournoi T , $H = V(G) \setminus N^+(v)$ et u un sommet de $N^+(v)$ tel que $|N^+(u) \cap H|$ est maximal. Dans les preuves suivantes, nous supposons que H est régulier ou presque régulier, autrement les policiers pourraient choisir v et un sommet de H comme positions initiales de façon à couvrir tout le graphe et gagner en un seul tour.

Proposition 5.2.1. Si dans un tournoi réflexif T de taille n , $\Delta^+(T) > n - 3$, alors le tournoi T est 2-policiers-gagnant.

Démonstration. S'il existe un sommet de degré sortant n ou $n - 1$, alors l'un des deux policiers choisit ce sommet et le deuxième choisit le sommet qui n'est pas couvert s'il y en a un et tout le graphe est couvert. S'il existe un sommet de degré sortant $n - 2$, le premier policier choisit ce sommet. Il reste donc deux sommets u, v qui ne sont pas couverts. Puisque T est un tournoi, alors $(u, v) \in A(T)$ ou $(v, u) \in A(T)$. Le deuxième policier choisit donc le sommet qui est un prédécesseur de l'autre, et couvrira ainsi les deux sommets. Dans tous les cas, le voleur sera

capturé dès le premier tour. □

Proposition 5.2.2. Si dans un tournoi réflexif T de taille n , $\Delta^+(T) = n - 3$, alors le tournoi T est 2-policiers-gagnant.

Démonstration. Suivant les précisions du début de section, nous allons travailler sur le tournoi suivant (Fig 5.1) :

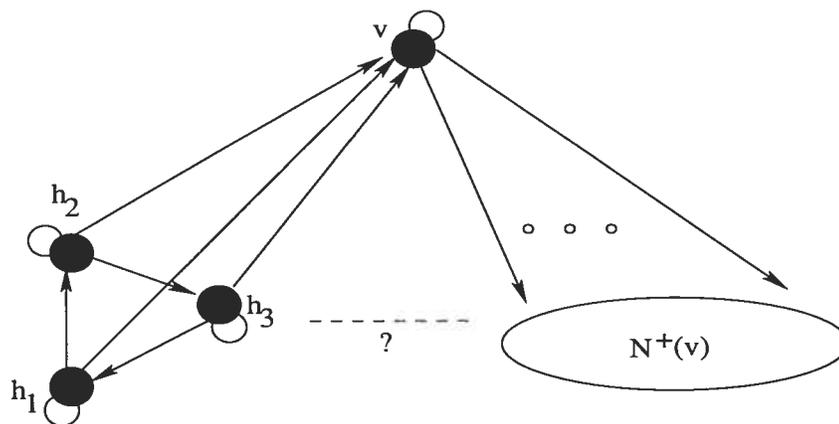


FIG. 5.1 – Exemple de tournoi

Cas possibles :

– $|N^+(u) \cap H| \geq 2$.

Supposons sans perdre de généralité que u couvre h_1 et h_2 .

Dans ce cas, les policiers commencent sur v et h_2 . Le voleur doit donc choisir h_1 comme position initiale. À leur premier tour, les policiers se déplacent sur u et v respectivement. Le seul sommet qui n'est pas couvert par les policiers est donc h_3 , que le voleur ne peut atteindre. Peu importe son déplacement, il sera donc attrapé au prochain tour.

– $|N^+(u) \cap H| = 1$. Les policiers commencent sur h_1 et h_2 et le voleur sur un sommet $w \in N^+(v)$. Nous savons que $|N^+(w) \cap H| \leq 1$. Au moins un des policiers couvre donc le sommet de H sur lequel le voleur peut se déplacer.

Ce policier reste donc sur place, pendant que l'autre se déplace sur v . Il n'y a donc plus de déplacement possible pour le voleur.

□

Proposition 5.2.3. Si dans un tournoi réflexif T de taille n , $\Delta^+(T) = n - 4$, alors T_n est 2-policiers-gagnant.

Démonstration. Suivant les précisions du début de section, nous allons travailler sur le tournoi suivant :

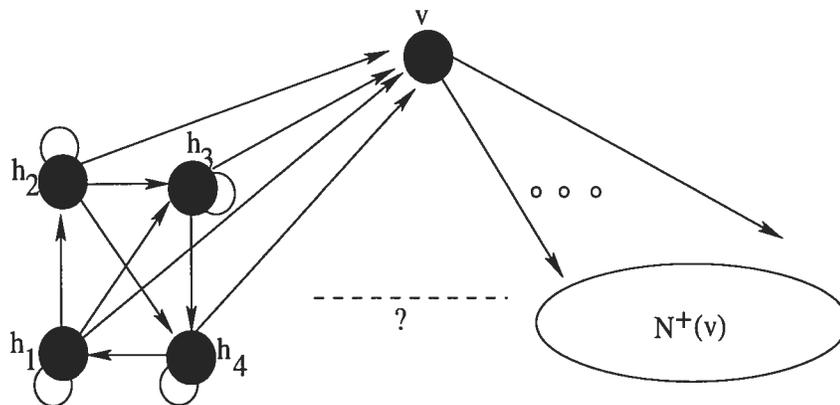


FIG. 5.2 – Exemple de tournoi

Cas possibles :

- $|N^+(u) \cap H| \geq 3$.

Donc $N^+(u) \supseteq N^+[h_3] \cap H$ ou $N^+(u) \supseteq N^+[h_4] \cap H$. Nous allons supposer sans perdre de généralité que le deuxième cas est vrai. Les policiers commencent sur h_1 et v , ce qui ne laisse que le sommet h_4 au voleur. Les policiers se déplacent ensuite sur v et u respectivement, et attraperont le voleur au tour suivant.

- $|N^+(u) \cap H| = 2$.

Les policiers commencent sur h_1 et h_2 et le voleur sur un sommet $w \in N^+(v)$.

Donc $N^+[h_1] \supseteq N^+(w) \cap H$ ou $N^+[h_2] \supseteq N^+(w) \cap H$. Peu importe w , un des sommets de H couvre $N^+(w) \cap H$, et l'un des policiers peut se déplacer sur ce sommet tandis que l'autre se déplace sur v au premier tour. Le voleur sera donc attrapé au tour suivant.

– $|N^+(u) \cap H| = 1$.

Les policiers débutent sur h_1 et h_2 et le voleur sur un sommet $w \in N^+(v)$. Nous savons que $|N^+(w) \cap H| \leq 1$. Un des deux policiers peut atteindre le seul sommet de H où le voleur peut se déplacer. Ce policier reste donc en place tandis que l'autre se déplace sur v , et de cette façon le voleur n'a plus de sommet où se déplacer.

□

Proposition 5.2.4. Soit T un tournoi réflexif de taille $n \leq 10$, alors T est 2-policiers-gagnant.

Démonstration. Tout tournoi de taille inférieure ou égale à 10 contient au moins un sommet v de degré maximum tel que $d^+(v) \geq n - 4$, et nous savons par les propositions précédentes qu'un tel tournoi est 2-policiers-gagnant. □

5.3 Tournois non-réflexifs

Comme pour les tournois réflexifs, nous considérerons dans les preuves suivantes que H est régulier ou presque régulier et que u est le sommet de $N^+(v)$ tel que $|N^+(u) \cap H|$ est maximum.

Proposition 5.3.1. Si dans un tournoi non-réflexif T de taille n , $\Delta^+(T) > n - 4$, alors le tournoi T est 2-policiers-gagnant.

Démonstration. Si v est un empereur, le tournoi est 2-policiers-gagnant. Dans le cas contraire, il existe un sommet u tel que $v \in N^+(u)$. Un des policiers choisit

u comme sommet de départ. Puisque v est de degré $n - 3$ ou plus, seulement 2 sommets ou moins ne sont pas dans son voisinage sortant fermé. Supposons que ces sommets sont au nombre de 2, et désignons-les par x, y , où $y \in N^+(x)$. Le deuxième policier choisit un sommet du voisinage entrant de x comme position initiale. Peu importe la position du voleur, les policiers se déplacent sur v et x au premier tour, et couvrent à eux deux tous les sommets. \square

Proposition 5.3.2. Si dans un tournoi non-réflexif T de taille n , $\Delta^+(T) = n - 4$, alors le tournoi T est 2-policiers-gagnant.

Démonstration. Suivant les précisions du début de section, nous allons travailler sur le tournoi suivant :

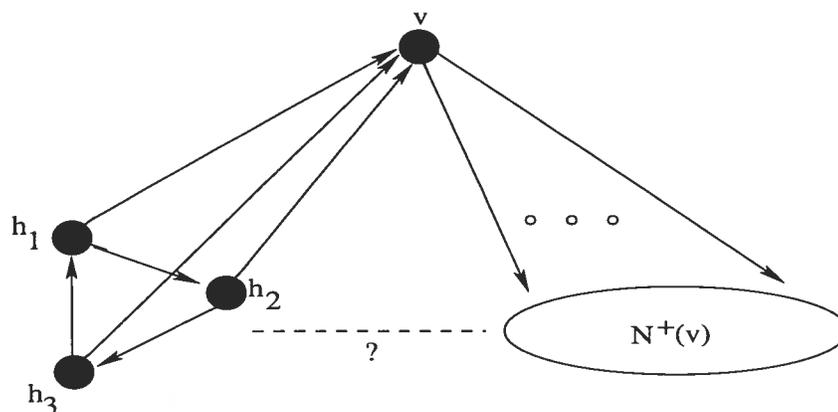


FIG. 5.3 – Exemple de tournoi

Cas possibles :

- $|N^+(u) \cap H| \geq 2$.

La stratégie des policiers est la même que pour le cas $n - 3$ sur un tournoi réflexif.

- $|N^+(u) \cap H| = 1$. Les policiers commencent sur h_1 et h_2 et le voleur sur un sommet $w \in N^+(v)$. Nous savons que $|N^+(w) \cap H| \leq 1$. Au moins un des

policiers couvre le sommet de H sur lequel le voleur peut se déplacer. Ce policier se déplace donc sur ce sommet, pendant que l'autre se déplace sur v . Il n'y a donc plus de déplacement possible pour le voleur et il sera capturé au tour suivant.

□

Proposition 5.3.3. Si dans un tournoi non-réflexif T de taille n , $\Delta^+(T) = n - 5$, alors le tournoi T est 2-policiers-gagnant.

Démonstration. Suivant les précisions du début de section, nous allons travailler sur le tournoi suivant :

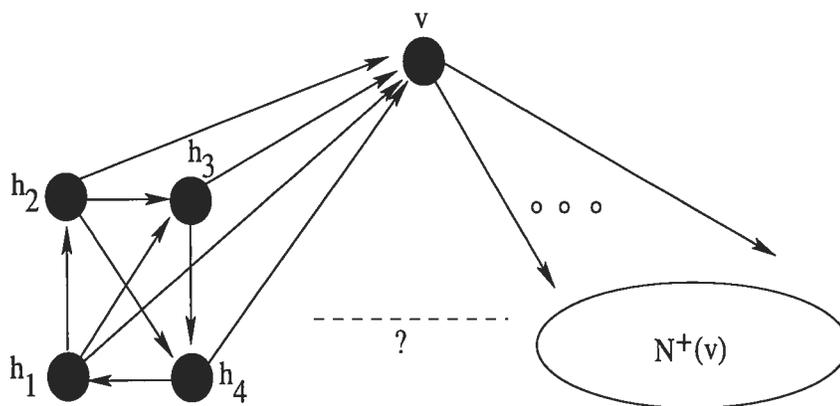


FIG. 5.4 – Exemple de tournoi

Cas possibles :

- $|N^+(u) \cap H| \geq 3$.

La stratégie des policiers est la même que pour le cas $n - 4$ dans un tournoi réflexif.

- $|N^+(u) \cap H| = 2$.

Donc $N^+[h_1] \supseteq N^+(u) \cap H$ ou $N^+[h_2] \supseteq N^+(u) \cap H$. Les policiers commencent sur h_1 et h_4 et le voleur sur un sommet $w \in N^+(v)$. Peu importe w ,

w est couvert dans H par h_1 ou h_2 . Un des policiers se déplace donc sur ce sommet tandis que l'autre se déplace sur v et le voleur sera capturé au tour suivant.

$$- |N^+(u) \cap H| = 1.$$

Les policiers débutent sur h_1 et h_2 et le voleur sur un sommet $w \in N^+(v)$. Nous savons que $|N^+(w) \cap H| \leq 1$. Un des deux policiers peut se déplacer de façon à couvrir le sommet de H que le voleur peut atteindre. L'autre policier se déplace sur v et le voleur n'aura plus de sommet où se déplacer.

□

Proposition 5.3.4. Soit T un tournoi non-réflexif de taille $n \leq 10$, alors T est 2-policiers-gagnant.

Démonstration. Tout tournoi non-réflexif de taille inférieure ou égale à 10 contient au moins un sommet v de degré maximum tel que $d^+(v) \geq n - 5$, et nous savons par les propositions précédentes qu'un tel tournoi est 2-policiers-gagnant.

5.4 Tournois non 2-policiers-gagnants

Définition 5.4.1. Un **tournoi de Paley** d'ordre q est un tournoi avec pour sommets les éléments du corps fini $GF(q)$. L'ordre q doit être une puissance d'un nombre premier telle que $q \equiv 3 \pmod{4}$. Un arc partira de x pour aller jusqu'à y si le résidu quadratique de $x - y$ est non-nul, c'est-à-dire si $x \neq y$ et s'il existe $x \in \mathbb{N}$ tel que $x^2 \equiv x - y \pmod{q}$ (voir Fig 5.5).

Définition 5.4.2. Un tournoi T est **n -existentiellement-clos** si pour tout sous-ensemble S de n sommets de $V(T)$ et pour tout sous-ensemble K de S , il existe un sommet $x \in V(T) \setminus S$ qui est dans le voisinage entrant de tous les sommets de $S \setminus K$ et dans le voisinage sortant de tous les sommets de K .

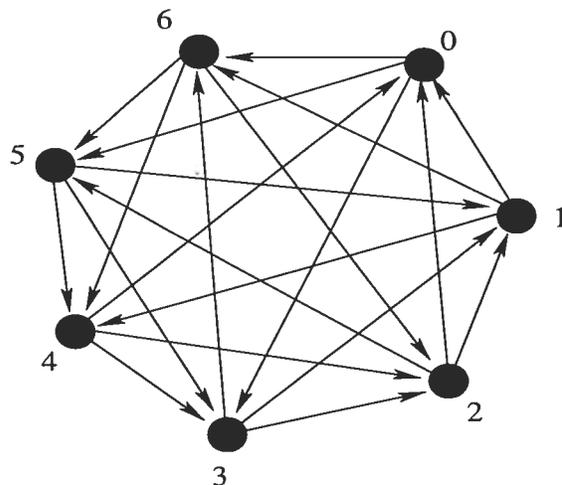


FIG. 5.5 – Tournoi de Paley d'ordre 7.

Donc si un tournoi partiellement réflexif est 3-existentiellement-clos, peu importe la position des 2 policiers et du voleur, il existe toujours un sommet sur lequel le voleur peut se déplacer et qui ne peut être atteint par les policiers. Un tel tournoi ne peut donc être 2-policiers-gagnant.

Le résultat qui nous intéresse est cité dans l'article de Bonato et Cameron [5], et a été prouvé par R.L. Graham et J.H. Spencer [13]. Ce résultat dit que les graphes de Paley, pour un n fixé, sont n -existentiellement-clos lorsque suffisamment grands. Certains tournois sont donc 3-existentially-closed et ne sont pas 2-policiers-gagnants.

CHAPITRE 6

ALGORITHME

Afin de mieux connaître les graphes et d'arriver à découvrir de nouvelles classes de graphes où un policier possède une stratégie gagnante, il est utile de pouvoir disposer d'un algorithme qui peut déterminer en temps polynomial si un graphe donné est policier-gagnant. Un tel algorithme a été décrit par G. Hahn et G. MacGillivray dans [14]. Les algorithmes décrits plus loin sont par contre une élaboration de celui décrit dans l'article mentionné précédemment puisque ces algorithmes sont implémentés de façon à pouvoir simuler la poursuite avec plusieurs policiers.

Nous considérerons de façon générale un graphe orienté D , qui peut représenter tous les types de graphes, orientés ou non, et avec boucles ou non. L'idée derrière cet algorithme est de construire un di-graphe représentant les mouvements possibles des policiers et du voleur sur un di-graphe D . Nous allons démontrer qu'un di-graphe est k -policiers-gagnant si et seulement si un tel di-graphe M_D construit par l'algorithme possède certaines propriétés.

6.1 Di-graphe de déplacement

Le di-graphe de déplacement M_D sera composé d'un ensemble de sommets V_D et d'un ensemble A_D d'arcs. Chaque sommet représente une configuration possible de la position des policiers et du voleur. De plus, chaque sommet aura comme information à qui des policiers ou du voleur il incombe d'effectuer un déplacement, et ces sommets seront respectivement placés dans les ensembles C_M et R_M . Il y aura par conséquent $2|V(D)|^{k+1}$ sommets. Quant-à eux, les arcs de A_D indiqueront s'il est possible, à partir de la configuration d'un sommet $x \in C_M$, d'obtenir la configuration d'un sommet $y \in R_M$, ou vice et versa. Il va sans dire qu'il ne peut

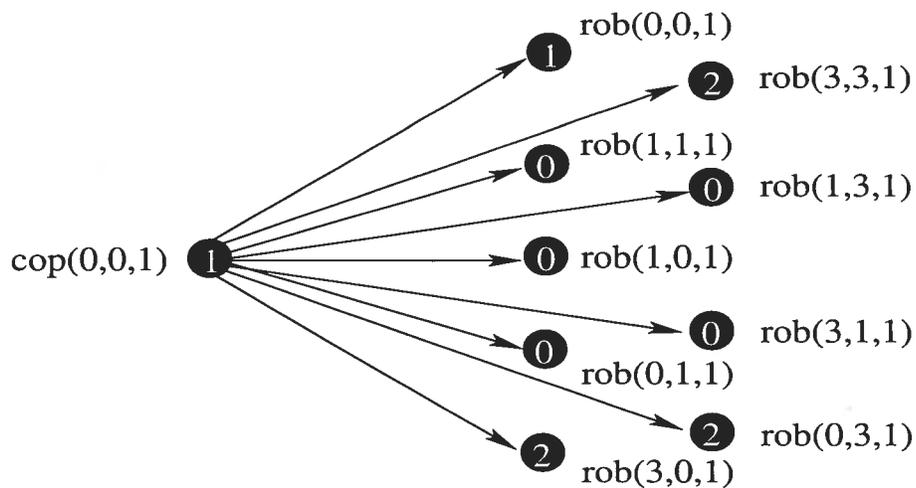
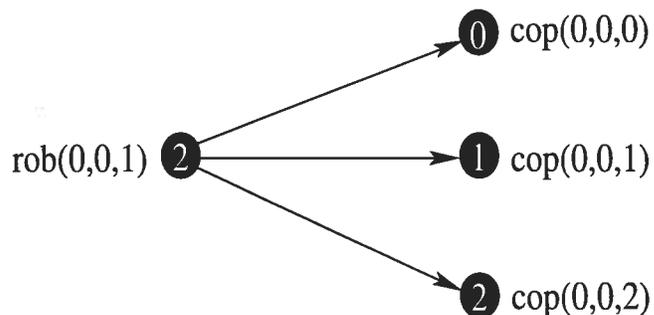
y avoir d'arcs entre deux sommets de C_M ou deux sommets de R_M , puisque les joueurs doivent se déplacer en alternance.

Nous allons définir $M_D = (V_D, A_D)$ comme suit :

- $C_M = V(D)^{k+1} \times \{cop\}$
- $R_M = V(D)^{k+1} \times \{rob\}$
- $V_D = C_M \cup R_M$. Nous utiliserons la notation $cop(c_1, c_2, \dots, c_k, r)$ et $rob(c_1, c_2, \dots, c_k, r)$ pour rendre la lecture plus aisée, plutôt que d'utiliser la forme $(c_1, c_2, \dots, c_k, r, cop)$ par exemple.
- $rob(c'_1, \dots, c'_k, r) \in N^+(cop(c_1, \dots, c_k, r))$ si pour $1 \leq i \leq k$, $(c_i, c'_i) \in A(D)$
- $cop(c_1, \dots, c_k, r') \in N^+(rob(c_1, \dots, c_k, r))$ si $(r, r') \in A(D)$.

Un sommet $cop(c_1, \dots, c_k, r)$ représente les positions c_i des k policiers et la position r du voleur, où il appartient maintenant aux policiers de faire un déplacement. Les arêtes de A_D relient tout simplement une configuration $cop(c_1, \dots, c_k, r)$ où les policiers doivent se déplacer, à une configuration $rob(c'_1, \dots, c'_k, r)$ telle que si les policiers sont positionnés dans D comme l'indique la première configuration, il est possible pour ceux-ci de se déplacer et d'obtenir la deuxième configuration. Des arêtes sont aussi créées des sommets $rob(c_1, \dots, c_k, r)$ jusqu'aux sommets $rob(c_1, \dots, c_k, r')$, si $(r, r') \in A(D)$. C'est-à-dire que le voleur se trouve sur r et qu'un arc sera créé vers chaque sommet de C_M où la position des policiers ne change pas et où r' est un successeur de r .

Les figures 6.1 et 6.2 montrent deux exemples de sous-graphes de M_D , où le di-graphe de jeu est l'équivalent orienté d'un cycle non-orienté et réflexif de 4 sommets. La figure 6.1 représente le voisinage d'un sommet de C_M tandis que la figure 6.2 représente celui d'un sommet de R_M . Les valeurs inscrites dans les sommets représentent les valeurs qui seraient obtenues après le passage de l'algorithme, et la signification de celles-ci sera expliquée sous-peu.

FIG. 6.1 – Exemple du voisinage d'un sommet de C_M FIG. 6.2 – Exemple du voisinage d'un sommet de R_M

6.2 Principe de l'algorithme

Le but de l'algorithme est d'étiqueter chaque sommet avec une valeur représentant le nombre de tours nécessaires aux policiers pour gagner. La valeur associée à chaque sommet sera préalablement initialisée à ∞ , et à 0 pour les sommets où la position dans D d'un des policiers correspond à celle du voleur puisqu'aucun déplacement n'est nécessaire aux policiers pour attraper le voleur. La phase d'initialisation se chargera aussi de changer à 0 la valeur des sommets où le voleur n'a pour seule option que de se déplacer sur un sommet occupé par un policier.

À chaque fois que l'algorithme considère un sommet, il ajuste la valeur de ce sommet en supposant que le joueur qui est dans cette position choisira le déplacement optimal. L'algorithme changera donc la valeur d'un sommet $c \in C_M$ en ajoutant 1 à la valeur du sommet $v \in N^+(c)$ ayant la plus petite valeur. C'est-à-dire que pour le sommet c , il faudra un tour pour se déplacer sur v , et $valeur(v)$ tours pour attraper le voleur. Quant à la valeur d'un sommet $r \in R_M$, elle sera changée pour la valeur maximum des sommets $v \in N^+(r)$, car le voleur choisit le sommet qui lui permettra de demeurer libre pendant le plus grand nombre de tours. Nous obtenons donc :

$$valeur(c) = 1 + \min\{ valeur(u) \mid u \in N^+(v) \}$$

$$valeur(r) = \max\{ valeur(u) \mid u \in N^+(v) \}$$

Pour éviter que l'algorithme n'entre dans une boucle infinie, il est nécessaire de spécifier une condition d'arrêt à celui-ci. Après un certain nombre de boucles, les valeurs ne changeront plus et il est important de détecter cet événement pour signifier à l'algorithme qu'il doit arrêter.

En étudiant les valeurs des sommets, il sera possible de déterminer si le di-graphe D est k -policiers-gagnant grâce à la proposition suivante :

Proposition 6.2.1. (Hahn et MacGillivray [14]) Un di-graphe D partiellement réflexif est k -policiers-gagnant si et seulement s'il existe $(c_1, \dots, c_k) \in V(D)^k$ tel que pour tout $r \in V(D)$, la valeur du sommet $cop(c_1, \dots, c_k, r)$ est finie.

Démonstration. La preuve est claire avec la proposition suivante. □

Proposition 6.2.2. (Hahn et MacGillivray [14]) Soit D un di-graphe et M_D son di-graphe de déplacement :

1. Les policiers possèdent une stratégie optimale en t tours à partir de $c =$

$cop(c_1, \dots, c_k, x)$ si et seulement si $valeur(c) = t$.

2. Le voleur possède une stratégie optimale en t tours à partir de $r = rob(c_1, \dots, c_k, x)$ si et seulement si $valeur(r) = t$.

Démonstration. Nous n'allons nous intéresser qu'aux cas où $t < \infty$ puisque le cas où t est infini est trivial, et procéder par induction sur t . Si $valeur(c) = 0$, alors il est clair que les policiers ont une stratégie en 0 tour puisque si $valeur(c) = 0$, cela implique qu'un policier occupe le même sommet que le voleur. Si $valeur(r) = 0$ alors soit un des policiers occupe le sommet du voleur, ou bien la valeur de v est nulle pour tout $v \in N^+(r)$. Le voleur est donc coincé et les policiers n'auront pas besoin de se déplacer pour l'attraper.

1. Supposons que les policiers ont une stratégie gagnante optimale en t tours à partir de c . Donc, il existe $v \in N^+(c) \subseteq R_M$ tel qu'il existe une stratégie gagnante en $t - 1$ tours. Donc $valeur(v) = t - 1$ et nous pouvons en déduire que $valeur(c) = t$ car autrement, si v n'est pas le sommet du voisinage de c ayant la valeur la plus basse, l'hypothèse selon laquelle la stratégie en t tours à partir de c est optimale est contredite. Supposons maintenant que $valeur(c) = t$, et que par conséquent il existe $v \in N^+(c) \subseteq R_M$ tel que $valeur(v) = t - 1$. Nous savons donc qu'il existe une stratégie optimale à partir de v pour attraper le voleur en $t - 1$ tours. Il est donc clair qu'il y a une stratégie optimale en t tours à partir de c .
2. Supposons que $valeur(r) = t$. Pour tout sommet $v \in N^+(r) \subseteq C_M$, $valeur(v) \leq t$. Donc peu importe le prochain déplacement du voleur, celui-ci sera capturé en t tours ou moins, et sa stratégie optimale consiste à choisir le sommet ayant pour valeur t . Supposons maintenant que le voleur a une stratégie optimale en t tours. Cela implique que pour tout sommet $v \in N^+(r) \subseteq C_M$, les policiers ont une stratégie gagnante optimale en t tours ou moins à partir de v . Donc, pour tout $v \in N^+(r) \subseteq C_M$, $valeur(v) \leq t$, et par conséquent la valeur de r est t car il existe au moins un voisin ayant pour valeur t . \square

Pour tous les algorithmes, nous supposons que M_D a déjà été construit et que la phase d'initialisation a été effectuée.

6.3 Version 1

Algorithme

début

Répéter jusqu'à ce qu'il n'y ait plus de changements :

pour chaque $v \in C_M$

si $valeur(v) = \infty$, changer $valeur(v) = 1 + \min\{ valeur(x) \mid x \in N^+(v) \}$

pour chaque $v \in R_M$

si $valeur(v) = \infty$, changer $valeur(v) = \max\{ valeur(x) \mid x \in N^+(v) \}$

fin

6.4 Version 2

Description de M_D

Dans cette version, le di-graphe de déplacement sera construit différemment afin d'en réduire la taille. Ce deuxième algorithme est donc presque identique au premier. La nouvelle construction a été pensée en fonction de réduire la taille de M_D afin de permettre de tester des graphes plus grands en un temps raisonnable. L'idée derrière ce changement est que pour deux sommets $cop(0, 1, 2, 3, 8)$ et $cop(3, 2, 1, 0, 8)$, les policiers ne sont pas aux mêmes positions individuellement, mais que le groupe de policiers occupe le même ensemble de quatre sommets. Nous pourrions donc considérer ces deux sommets comme équivalents et c'est ce que nous ferons en cessant de nous intéresser à la position de chaque policier, pour plutôt utiliser le nombre de policiers se trouvant sur chaque sommet, sans distinctions entre les policiers.

Plus formellement :

- $C_M = (c_1, \dots, c_n, r) \times \{Cop\}$, $R_M = (c_1, \dots, c_n, r) \times \{Rob\}$, où $c_i \leq k$, $\sum c_i = k$ et c_i est le nombre de policiers présents sur le sommet i , et r le sommet où se trouve le voleur.
- $V_D = C_M \cup R_M$.
- $Rob(c'_1, \dots, c'_n, r) \in N^+(Cop(c_1, \dots, c_n, r))$ si il existe $(u_1, u_2, \dots, u_k), (v_1, v_2, \dots, v_k) \in V^k$ tel que pour $0 \leq j \leq k$, les conditions suivantes sont respectées :
 - $(u_j, v_j) \in A(D)$
 - $|\{u_i \mid u_i = j\}| = c_j$
 - $|\{v_i \mid v_i = j\}| = c'_j$
- $Cop(c'_1, \dots, c'_n, r') \in N^+(Rob(c'_1, \dots, c'_n, r))$ si $r' \in N^+(r)$.

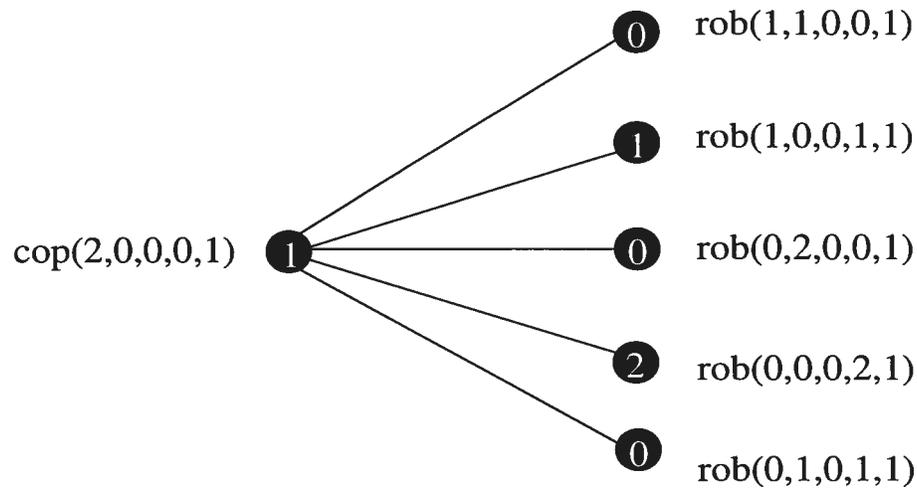


FIG. 6.3 – Exemple du voisinage d'un sommet de C_M dans la nouvelle construction

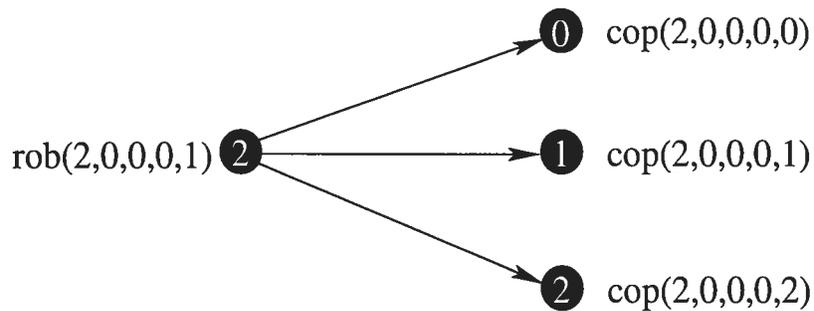


FIG. 6.4 – Exemple du voisinage d'un sommet de R_M dans la nouvelle construction

Algorithme

début

Répéter jusqu'à ce qu'il n'y ait plus de changements :

pour chaque $v \in C_M$

si $\text{valeur}(v) = \infty$, changer $\text{valeur}(v) = 1 + \min\{ \text{valeur}(x) \mid x \in N^+(v) \}$

pour chaque $v \in R_M$

si $\text{valeur}(v) = \infty$, changer $\text{valeur}(v) = \max\{ \text{valeur}(x) \mid x \in N^+(v) \}$

fin

Analyse

La taille du di-graphe de déplacement est considérablement réduite, ce qui permet de faire une économie de temps calcul substantielle. Plus de détails seront donnés dans la section contenant l'analyse des résultats.

Malheureusement, cette nouvelle approche crée aussi son lot de problèmes. Si le nombre de sommets est réduit de beaucoup, il n'en est pas de même pour le nombre d'opérations nécessaires à la création des arcs de ce nouveau di-graphe.

Dans la version précédente, il était facile de vérifier si $r \in R_M$ était dans le voisinage de $c \in C_M$. Il en est tout autrement dans cette nouvelle construction car pour vérifier l'adjacence de ces deux sommets, il faut vérifier de nombreuses possibilités. Toutefois, la complexité pour vérifier si $c \in C_M$ est dans le voisinage de $r \in R_M$ ne change pas.

Pour résoudre ce problème, on associe à chaque sommet de la nouvelle représentation un sommet correspondant dans l'ancienne. Par exemple, si on veut savoir quels sont les voisins de $cop(0,1,0,0,1)$, on prend le sommet $cop(1,4)$ de l'ancienne représentation, on trouve ses voisins et on convertit ceux-ci dans la nouvelle représentation. C'est donc une façon relativement simple et peu coûteuse de résoudre le problème que posait la nouvelle façon de construire M_D .

6.5 Version 3

Il est important de noter que le di-graphe de déplacement utilisé dans cette version est le même que celui de la version 2.

Algorithme

début

pour chaque $v \in C_M$, avec $valeur(v) = 0$

 ajouter v à *File*

pour chaque $v \in R_M$ avec $valeur(v) = 0$

 ajouter v à *File*

tant que(*File* $\neq \emptyset$)

v = premier élément de *File*, enlever v de *File*.

 Si $v \in C_M$, changer $valeur(v) = 1 + \min\{ valeur(x) \mid x \in N^+(v) \}$

 Si $v \in R_M$, changer $valeur(v) = \max\{ valeur(x) \mid x \in N^+(v) \}$

 Si $valeur(v)$ a été changée, ajouter $N^-(v)$ à *File*

fin

Analyse

Cette version utilise la même construction de M_D que celle précédente, mais utilise une stratégie plus optimale pour l'étiquetage des sommets. Cette stratégie ressemble à un parcours en largeur modifié. Plutôt de n'avoir qu'un seul point de départ comme les parcours en largeur traditionnels, ce sera un groupe de sommets qui servira de point de départ, ces sommets étant ceux dont la valeur a été mise à 0 dans la phase initiale. De plus, le parcours sera fait en ordre inverse, c'est-à-dire en examinant les sommets du voisinage entrant ouvert plutôt que ceux du voisinage sortant ouvert. Il sera démontré plus loin qu'avec cette méthode, l'étiquetage se fait en ordre croissant des valeurs.

L'algorithme examine donc un sommet v , tente de changer sa valeur, et ajoute dans la file les sommets de $N^-(v)$ seulement si la valeur de v a pu être changée ou si v a pour valeur 0. En effet, il n'est intéressant d'examiner un sommet que lorsqu'un de ses voisins sortants a été modifié puisque sa valeur ne peut changer sans qu'un changement ait eu lieu dans les sommets de son voisinage sortant. Avec cette nouvelle stratégie, de nombreux sommets ne seront plus parcourus inutilement.

6.6 Justifications

Il est important de noter l'ordre dans lequel l'algorithme parcourt les sommets de M_D . Sans un ordre des sommets tel que choisi, l'algorithme deviendrait beaucoup plus lent. Cet ordre est particulièrement nécessaire lorsque la condition "si $valeur(v) = \infty$ " est utilisée et de plus amples détails seront donnés plus loin. En effet, quoi qu'il ait été impossible de prouver formellement ce fait, il a été constaté lors de quelques tests avec l'algorithme que la valeur d'un sommet peut changer plusieurs fois pendant l'exécution de l'algorithme si l'ordre des sommets n'est pas approprié et si cette condition n'est pas utilisée. Nous nous limiterons donc à prouver plus loin que les valeurs ne peuvent changer lorsque certaines conditions sont respectées.

Si par exemple les sommets étaient placés dans une liste en ordre arbitraire, cela aurait pour effet que lorsque la valeur d'un sommet v serait changée une première fois pour $x < \infty$, cette valeur ne représenterait pas obligatoirement la valeur de la stratégie optimale, c'est-à-dire que $valeur(v) = x$ mais qu'il est possible pour les policiers de gagner en moins de x tours. Sans un ordre approprié des sommets, la condition "si $valeur(v) = \infty$ " empêchera les sommets d'atteindre leur valeur optimale.

Peu importe l'ordre des sommets, l'algorithme dans sa version actuelle est toujours utile pour affirmer si les policiers ont une stratégie gagnante car, pour cela, il suffit de savoir qu'il existe une stratégie finie, sans plus de détails. Pour être certain d'obtenir à nouveau des valeurs optimales, il suffit d'enlever la condition "si $valeur(v) = \infty$ ". Certaines valeurs seront donc changées plusieurs fois avant d'atteindre leur optimum, ce qui aura pour effet de ralentir l'algorithme considérablement.

Pour justifier les deux premiers algorithmes, il est nécessaire de s'attarder à la condition "si $valeur(v) = \infty$ ". Si celle-ci n'est pas utilisée, la validité a été démontrée dans [14]. Mais ne pas utiliser cette condition rendra l'algorithme lent et inefficace puisque beaucoup de sommets seront examinés inutilement. Nous voulons donc utiliser cette condition tout en s'assurant que les valeurs obtenues quand l'algorithme se termine représentent bien la valeur des stratégies optimales. Il sera donc démontré plus loin que l'ordre des sommets choisi remplit cette condition.

Soit D un di-graphe où les policiers sont gagnants sur lequel on lance l'algorithme. Soit $c \in C_M$ un sommet examiné ayant pour valeur ∞ , tel qu'il existe des sommets $u, v \in N^+(c)$ tels que la valeur de v est infinie, et celle de u finie. Nous savons que de tels sommets c, u, v existent car autrement cela signifierait que toutes les valeurs des sommets de R_M ont été changées avant ceux de C_M , ce qui

est impossible puisque pour que la valeur d'un sommet de R_M soit changée, il faut que tous ses voisins aient des valeurs finies.

Soit $t = \min\{ \text{valeur}(x) \mid x \in N^+(c) \}$, où $t < \infty$. La valeur de c sera donc changée à t et pour pouvoir affirmer que la valeur de c est optimale, il faut être certain que $\text{valeur}(v)$ sera supérieure à t lorsque l'algorithme changera la valeur de v , ce qui n'est pas toujours vrai. Il faut donc que les valeurs soient attribuées en ordre croissant, si nous voulons utiliser la condition "si $\text{valeur}(v) = \infty$ " et quand même obtenir des valeurs optimales. Nous prouverons donc à l'aide des propositions suivantes que l'ordre utilisé par les deux premiers algorithmes produit un ordre croissant des valeurs et que par conséquent, les valeurs obtenues sont optimales.

Les trois propositions qui suivent sont nouvelles, et viennent compléter celles de l'article de Hahn et MacGillivray.

Proposition 6.6.1. Si l'algorithme attribue la valeur x dans une boucle, il n'attribuera que des $x + 1$ dans la boucle suivante.

Démonstration. Nous allons démontrer la proposition par induction. Dans la phase initiale, la valeur de certains sommets est mise à 0, et la première boucle n'attribuera donc que des valeurs 1 car les seuls sommets ayant des valeurs finies ont pour valeur 0. Certains sommets de C_M verront donc leur valeur changée à 1, ainsi que quelques sommets de R_M par la suite. Il est évident que les sommets de C_M ne peuvent pas obtenir une valeur différente de 1, et il en est de même pour ceux de R_M car ceux ayant pour valeur 0 ont tous été parcourus dans la phase initiale.

Supposons l'hypothèse vraie et vérifions que lorsque l'algorithme attribue une valeur x , alors la prochaine boucle n'utilisera que des $x + 1$. Soit $c \in C_M$ un sommet dont la valeur est encore ∞ . Cela signifie qu'à la boucle précédente, la valeur de v est infinie pour tout $v \in N^+(c)$. Donc la valeur de c ne sera changée que si l'un

de ses voisins a vu sa valeur changée à x à la fin de la boucle précédente après que celui-ci eut été examiné, et elle sera changée à $x + 1$ si elle l'est.

Soit $r \in R_M$ un sommet dont la valeur est encore ∞ . Cela signifie qu'à la boucle précédente il existe $v \in N^+(r)$ tel que $valeur(v) = \infty$, après que l'algorithme ait parcouru tous les sommets de C_M . Donc la valeur de r ne peut changer que si l'un de ses voisins ayant une valeur infinie a vu sa valeur changée à $x + 1$ au début de la boucle courante, et elle sera changée à $x + 1$ si elle l'est, car toutes les valeurs attribuées jusqu'à ce moment sont inférieures ou égales à $x + 1$. \square

Deux autres propositions peuvent être déduites de la proposition précédente :

Proposition 6.6.2. Si l'algorithme attribue la valeur $x < \infty$ à un sommet, alors pour tous les sommets $v \in M_D$ tels que $valeur(v) < \infty$, la valeur de v est inférieure ou égale à x .

Proposition 6.6.3. Si l'algorithme attribue la valeur $x < \infty$ à un sommet, cette valeur est optimale et ne peut changer.

Démonstration. Il est clair que la valeur des sommets ayant pour valeur initiale 0 ne peut changer. Il ne peut exister d'autres sommets de C_M dont la valeur sera 0 car $valeur(c) = 1 + \min\{valeur(v) \mid v \in N^+(c)\} \geq 1$. Si un sommet $r \in R_M$ voit sa valeur changer à 0 durant la phase initiale, c'est que la valeur de c est nulle pour tout $c \in N^+(r)$, où chaque c ne peut voir sa valeur changer. Nous pouvons donc affirmer que les sommets ayant pour valeur 0 ne voient leur valeur changer qu'au plus une fois.

Supposons la proposition vraie pour $valeur(v) = n$ et vérifions pour $valeur(v) = n + 1$. Si $v \in C_M$, alors il est clair que la valeur ne peut changer car selon la proposition vue précédemment, les valeurs qui seront attribuées ultérieurement seront supérieures ou égales à $n + 1$ et ne changeront pas le minimum de $\{valeur(r) \mid r \in N^+(v)\}$. De plus, la valeur du sommet u , tel que $u \in N^+(v)$ et $valeur(u) = n$,

ne peut changer par l'hypothèse. Si $v \in R_M$, alors nous savons que pour tout $c \in N^+(v)$, la valeur de c est inférieure ou égale à $n + 1$. Les valeurs inférieures à $n + 1$ de ces sommets du voisinage ne peuvent changer par l'hypothèse d'induction, et nous avons prouvé ci-dessus que pour $c \in C_M$ avec $valeur(v) = n + 1$, la valeur ne peut changer. Donc la valeur de $v \in R_M$ ne peut changer. \square

L'ordre dans lequel l'algorithme parcourt les sommets est donc très important puisque les valeurs doivent être attribuées en ordre croissant. Nous distinguerons trois cas possibles pour l'algorithme, qui résument une partie des justifications :

1. La condition "si $valeur(v) = \infty$ " n'est pas utilisée et l'ordre est quelconque.
Résultat : l'algorithme est lent mais valide.
2. La condition "si $valeur(v) = \infty$ " est utilisée et l'ordre est quelconque.
Résultat : l'algorithme est rapide, mais les valeurs finales obtenues ne sont pas optimales. La validité est conservée puisqu'il est possible d'affirmer si les policiers peuvent gagner sur le di-graphe, car nous pouvons quand même vérifier si toutes les valeurs sont finies. Par contre il ne sera pas possible de trouver une stratégie puisque les valeurs ne sont pas toutes optimales.
3. La condition "si $valeur(v) = \infty$ " est utilisée et l'ordre est croissant.
Résultat : l'algorithme est rapide et valide.

6.7 Analyse des résultats

Graphe1 : Cycle réflexif de longueur 4.

Graphe2 : Graphe de Peterson réflexif.

Graphe3 : Graphe de Peterson réflexif avec 2 arêtes additionnelles.

Graphe4 : Système Triple de Steiner (13), di-graphe non-réflexif.

Graphe5 : Graphe 3-régulier non-réflexif.

Graphe6 : Graphe 4-régulier non-réflexif.

Graphe7 : Graphe 3-régulier non-réflexif.

Graphe8 : Graphe 4-régulier non-réflexif.

Ces 8 graphes ont été utilisés pour tester les 3 versions de l'algorithme. Il faut maintenant définir ce qu'est un Système Triple de Steiner pour comprendre le quatrième graphe.

Un **Système Triple de Steiner**, noté $STS(n)$, est un ensemble X de n éléments, et un ensemble $Y \subseteq \binom{X}{3}$ tel que toute paire d'éléments distincts $a, b \in X$ est contenue par un et un seul des éléments de Y . On peut aussi voir un $STS(n)$ comme une décomposition de $E(K_n)$ en cycles de longueur 3 n'ayant pas d'arêtes en commun. Dans un cas orienté comme le Graphe4, l'ensemble d'arc est décomposé en circuits plutôt qu'en cycles. Ce di-graphe est donc un di-graphe sur 13 sommets où chaque paire de sommets appartient à un et un seul circuit de longueur 3.

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

FIG. 6.5 – Exemple d'une matrice d'adjacence

De façon plus précise, ce di-graphe est défini par la matrice d'adjacence H illustrée dans la figure 6.5. Notons qu'une **matrice d'adjacence** d'un di-graphe de taille n

est une matrice $n \times n$ telle que il existe un arc de i vers j dans le di-graphe si et seulement si $a_{i,j} = 1$.

Notons que les tests ont été effectués sur un processeur AMD Athlon(tm) 64 Processor 4000+ de 2400Mhz, avec 2048Mb de mémoire vive. Bien que rapide, ce système n'a malheureusement pas pu compléter certains tests en un délai raisonnable, et c'est pour cette raison que certaines valeurs sont notées "???". Les temps enregistrés sont le nombre de secondes qui ont été nécessaires à l'algorithme pour construire le di-graphe de déplacement et ensuite attribuer les valeurs. Une étape représente quant à elle un appel à la fonction *valeur(v)* par l'algorithme.

On peut voir au premier coup d'oeil que la première version est de loin la plus lente, tandis que la troisième est de loin la plus rapide. La deuxième version est quant-à elle entre les 2 autres avec des performances moyennes.

Il est clair que le changement fait dans la construction de M_D a porté fruit. Pour un petit nombre de policiers, la différence est minime, mais lorsque M_D est construit en fonction de 3 ou 4 policiers, la différence de taille devient considérable. La taille de $V(M_D)$ sera de $n \frac{n!}{(n-k)!k!}$ comparativement à n^{k+1} précédemment. Prenons un exemple sur un graphe relativement simple, le deuxième graphe. Lorsqu'il est construit de la première façon, on obtient 200,000 sommets et 26 millions d'arcs. Avec la deuxième façon, 14,300 sommets et 794,580 arcs sont obtenus. Cette différence est énorme et se reflète directement sur le temps et le nombre d'étapes. Dans le cas présent, le nombre d'appels faits à la fonction *valeur(v)* est d'un peu plus de 8 milliards dans la première version, contre environ 50 millions dans la deuxième!

Quant à la troisième version, elle est exécutée sur le même di-graphe de déplacement M_D que la deuxième, mais la performance est plusieurs fois plus rapide. Cette amélioration vient du fait qu'il n'y a presque plus de sommets examinés en vain.

TAB. 6.1 – Résultats de la version 1

D	$ V(D) $	$ A(D) $	k	Gagnant	$ V(M_D) $	$ A(M_D) $	Temps	Étapes
Graphe1	4	12	1	voleur	32	96	0.0s	848
Graphe1	4	12	2	policiers	128	768	0.03s	2807
Graphe1	4	12	3	policiers	512	7680	0.18s	21995
Graphe1	4	12	4	policiers	2048	86016	0.53s	188759
Graphe2	10	40	1	voleur	200	800	0.07s	60620
Graphe2	10	40	2	voleur	2000	20000	0.94s	6292100
Graphe2	10	40	3	policiers	20000	680000	24.57s	108507436
Graphe2	10	40	4	policiers	200000	26000000	2337.34s	8640465596
Graphe3	10	42	1	voleur	200	840	0.08s	63157
Graphe3	10	42	2	policiers	2000	21840	0.62s	1553295
Graphe3	10	42	3	policiers	20000	782880	25.86s	113183979
Graphe3	10	42	4	policiers	200000	31536960	2217.1s	9059464815
Graphe4	13	78	1	voleur	338	2028	0.12s	258362
Graphe4	13	78	2	policiers	4394	92274	1.93s	14675510
Graphe4	13	78	3	policiers	57122	6340542	324.13s	1946369360
Graphe4	13	78	3	policiers	???	???	???	???
Graphe5	10	30	1	voleur	200	600	0.06s	45347
Graphe5	10	30	2	policiers	2000	12000	0.64s	1219481
Graphe5	10	30	3	policiers	20000	300000	23.37s	89448170
Graphe5	10	30	4	policiers	200000	8400000	2172.37s	7270597703
Graphe6	10	40	1	voleur	200	800	0.07s	58800
Graphe6	10	40	2	policiers	2000	20000	0.87s	1509376
Graphe6	10	40	3	policiers	20000	680000	26.19s	116315816
Graphe6	10	40	4	policiers	200000	26000000	2272.99s	9093500828
Graphe7	14	42	1	voleur	392	1176	0.1s	189245
Graphe7	14	42	2	policiers	5488	32928	3.07s	22087474
Graphe7	14	42	3	policiers	76832	1152480	550.19s	2283628958
Graphe7	14	42	3	policiers	???	???	???	???
Graphe8	14	56	1	voleur	392	1568	0.14s	247072
Graphe8	14	56	2	policiers	5488	54880	2.84s	21320760
Graphe8	14	56	3	policiers	76832	2612288	570.54s	3282834328

TAB. 6.2 – Résultats de la version 2

D	$ V(D) $	$ A(D) $	k	Gagnant	$ V(M_D) $	$ A(M_D) $	Temps	Étapes
Graphe1	4	12	1	voleur	32	96	0.0s	848
Graphe1	4	12	2	policiers	80	364	0.02s	1370
Graphe1	4	12	3	policiers	160	1116	0.1s	3378
Graphe1	4	12	4	policiers	280	2692	0.19s	7188
Graphe2	10	40	1	voleur	200	800	0.07s	60620
Graphe2	10	40	2	voleur	1100	9690	0.68s	1819458
Graphe2	10	40	3	policiers	4400	100570	1.81s	5903936
Graphe2	10	40	4	policiers	14300	794580	17.13s	50530618
Graphe3	10	42	1	voleur	200	840	0.08s	63182
Graphe3	10	42	2	policiers	1100	10410	0.58s	533520
Graphe3	10	42	3	policiers	4400	108040	1.73s	6050245
Graphe3	10	42	4	policiers	14300	844200	17.01s	52578620
Graphe4	13	78	1	voleur	338	2028	0.15s	258518
Graphe4	13	78	2	policiers	2366	42874	1.01s	3720152
Graphe4	13	78	3	policiers	11830	798239	15.29s	78567104
Graphe4	13	78	4	policiers	47320	11086465	289.25s	1154734619
Graphe5	10	30	1	voleur	200	600	0.07s	45347
Graphe5	10	30	2	policiers	1100	6060	0.49s	444033
Graphe5	10	30	3	policiers	4400	51660	1.66s	5021096
Graphe5	10	30	4	policiers	14300	358990	16.36s	45281502
Graphe6	10	40	1	voleur	200	800	0.08s	58800
Graphe6	10	40	2	policiers	1100	9880	0.58s	519683
Graphe6	10	40	3	policiers	4400	99670	1.76s	6285312
Graphe6	10	40	4	policiers	14300	802240	17.2s	52979465
Graphe7	14	42	1	voleur	392	1176	0.12s	189245
Graphe7	14	42	2	policiers	2940	16968	1.13s	4785025
Graphe7	14	42	3	policiers	15680	203546	24.04s	84839517
Graphe7	14	42	4	policiers	66640	2054010	455.9s	1357712681
Graphe8	14	56	1	voleur	392	1568	0.13s	247072
Graphe8	14	56	2	policiers	2940	27608	1.4s	6851959
Graphe8	14	56	3	policiers	15680	427644	26.15s	124067099
Graphe8	14	56	4	policiers	66640	5449052	502.24s	2177884131

TAB. 6.3 – Résultats de la version 3

D	$ V(D) $	$ A(D) $	k	Gagnant	$ V(M_D) $	$ A(M_D) $	Temps	Étapes
Graphe1	4	12	1	voleur	32	96	0.01s	116
Graphe1	4	12	2	policiers	80	364	0.02s	389
Graphe1	4	12	3	policiers	160	1116	0.06s	825
Graphe1	4	12	4	policiers	280	2692	0.23s	1593
Graphe2	10	40	1	voleur	200	800	0.05s	800
Graphe2	10	40	2	voleur	1100	9690	0.62s	7259
Graphe2	10	40	3	policiers	4400	100570	0.91s	88354
Graphe2	10	40	4	policiers	14300	794580	4.07s	572084
Graphe3	10	42	1	voleur	200	840	0.05s	842
Graphe3	10	42	2	policiers	1100	10470	0.39s	13062
Graphe3	10	42	3	policiers	4400	111140	1.16s	96849
Graphe3	10	42	4	policiers	14300	892090	4.44s	640108
Graphe4	13	78	1	voleur	338	2028	0.09s	2210
Graphe4	13	78	2	policiers	2366	41860	0.71s	49369
Graphe4	13	78	3	policiers	11830	722735	3.15s	628760
Graphe4	13	78	4	policiers	47320	10289396	66.26s	7815685
Graphe5	10	30	1	voleur	200	600	0.04s	536
Graphe5	10	30	2	policiers	1100	6060	0.52s	7658
Graphe5	10	30	3	policiers	4400	51660	0.82s	48239
Graphe5	10	30	4	policiers	14300	358990	2.56s	272641
Graphe6	10	40	1	voleur	200	800	0.04s	760
Graphe6	10	40	2	policiers	1100	9880	0.73s	11848
Graphe6	10	40	3	policiers	4400	99670	0.98s	86791
Graphe6	10	40	4	policiers	14300	802240	4.29s	579427
Graphe7	14	42	1	voleur	392	1176	0.07s	860
Graphe7	14	42	2	policiers	2940	16968	0.65s	22737
Graphe7	14	42	3	policiers	15680	203546	1.91s	203048
Graphe7	14	42	4	policiers	66640	2054010	17.11s	1699975
Graphe8	14	56	1	voleur	392	1568	0.1s	1392
Graphe8	14	56	2	policiers	2940	27608	0.84s	36220
Graphe8	14	56	3	policiers	15680	427644	2.54s	398738
Graphe8	14	56	4	policiers	66640	5449052	32.56s	4299110

CHAPITRE 7

CONCLUSION

C'est ainsi que se termine cette étude sur le jeu de poursuite de recherche. Plusieurs aspects ont été scrutés, mais évidemment, ce sujet en comporte plusieurs autres.

Afin de mieux comprendre les tournois, de nombreux tests ont été faits sur des tournois générés de façon aléatoire, de taille allant jusqu'à 50 sommets. Ces tests ont montré que tous ces tournois étaient 2-policiers-gagnants. De plus, les policiers possédaient toujours une stratégie gagnante qui leur permettait de capturer le voleur en très peu de déplacement, c'est-à-dire moins de 5. Cela semblait donc indiquer que tous les tournois sont 2-policiers-gagnants et il aurait été intéressant de pouvoir démontrer ce fait. Mais des résultats cités par Bonato et Cameron [5] impliquent que certains tournois ne peuvent respecter cette propriété. Il a donc été possible de prouver qu'une partie des tournois sont 2-policiers-gagnants, mais cela n'a malheureusement pas été le cas pour tous les tournois.

Peut-être que tous les tournois qui ne sont pas des tournois de Paley sont 2-policiers-gagnants, et cette possibilité devrait être explorée. D'autres tests pourraient aussi être effectués sur une machine plus performante afin de tester des tournois de plus grande taille et d'observer si ceux-ci sont aussi 2-policiers-gagnants.

Bien que le dernier algorithme implémenté soit performant, des améliorations pourraient sûrement être apportées. L'emphase a plutôt été mise sur l'aspect algorithmique que sur une grande optimisation du code.

Il serait aussi intéressant de vérifier si l'idée derrière cet algorithme peut être utilisée pour développer des algorithmes pour différents jeux de poursuite.

- [13] R.L. Graham and J.H. Spencer. A constructive solution to a tournament problem. *Can. Math. Bull.* 14, 14 :45–48, 1971.
- [14] G. Hahn and G. MacGillivray. A note on k-cop, l-robber games on graphs. *Discrete Mathematics*, 306 :2492–2497, 2006.
- [15] A. Hill. *Several Results on Cops and Robbers*. PhD thesis, Dalhousie University, 2006.
- [16] V. Isler, S. Kannan, and S. Khanna. Randomized pursuit-evasion with limited visibility. In *SODA '04 : Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1060–1069, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.
- [17] R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Mathematics*, 43 :235–239, 1983.
- [18] A. Quilliot. *Problèmes de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hyper-graphes*. PhD thesis, Université de Paris VI, 1983.
- [19] D. West. *Introduction to Graph Theory*. Prentice Hall, second edition, 2001.