

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE DE MOULOU D MAMMERI DE TIZI OUZOU

Faculté des Sciences

Département de Mathématiques



Mémoire de fin d'études

En vue de l'obtention du diplôme de Master en Recherche Opérationnelle

Intitulé :

*Broadcast domination efficace dans les graphes
série-parallèle et chaînes P_n*

Étudié par :

CHABANE CHAUCHE Lydia

ARKAT Ferroudja

Soutenu le 04/10/2021 devant le jury :

| | | | |
|----------------|-------|-----------|------------|
| Mr BELHADJ. A. | M.A.A | U.M.M.T.O | President |
| Mr KASDI. K | M.A.A | U.M.M.T.O | Examineur |
| Mr AOUANE. M. | M.A.A | U.M.M.T.O | Rapporteur |

Promotion 2020/ 2021

Remerciements

Nous tenons à remercier en premier lieu, notre encadreur Mr. AOUANE, tout d'abord pour avoir accepté de nous encadrer, ensuite pour sa modestie et gentillesse. Il ne saura jamais assez notre gratitude pour tout ce qu'il a pu nous apprendre et fait découvrir.

Toute notre gratitude va également à Mr. SADI, pour nous avoir enseigné les bases de la Théorie des Graphes, et fait découvrir la branche que nous avons le plus apprécié de notre parcours universitaire.

Nous adressons nos plus sincères remerciements à MM. BELHADJ et KASDI pour avoir accepté d'examiner notre mémoire.

Nous vous adressons un immense merci à vous tous, membres du Département de Mathématiques, que vous soyez Enseignants, Doctorants ou même Administratifs, pour avoir contribué à notre formation en général, et à la réalisation de ce modeste travail en particulier.

Nous n'oublions non plus de remercier nos familles et proches pour tout le soutien qu'ils nous ont porté tout au long de ces années d'études.

Lydia, Ferroudja ♡

Dédicaces

Ce travail est dédié à ma très chère famille que j'aime tant, à mes parents ♡, je suis hyper reconnaissante de vous avoir dans ma vie, sans vous je ne serais jamais là où je suis aujourd'hui.. Merci pour votre confiance, AMOUR, encouragements et pour tous vos sacrifices... ♡

Comme je le dédie également à mes chères soeurs Ferroudja et Tita spécialement, qui a toujours été à mes cotés dans les bons comme dans les mauvais moments, mon frère et ma petite nièce adorée Ahlem. ♡

Je n'oublie non plus de dédier un très grand merci à mes amis, à Idir ♡ qui m'a toujours encouragé et porté un énorme soutien mental, ainsi qu'à mon très cher ami Joseph ♡ pour sa disponibilité et son aide dans la partie de programmation, à ma chère binôme Ferroudja ♡.. Et à toute ma promotion pour les bons moments que j'ai vécu auprès de vous.

A toi ma Maman, A toi mon Papa

Lydia ♡

Dédicaces

Je dédie ce modeste travail à mes très chers parents, à la mémoire de mon père qu'Allah l'accueille dans son vaste paradis. A ma chère mère qui est ma source d'amour et du courage, qui a sacrifié pour que je puisse arriver jusque là et qui m'a beaucoup soutenu durant mon cursus

A mon frère Amar et sa femme Sara

A mes sœurs Lili, Fatima, Malika, Zhour, Dehbia , Zahra

A mon homme Saadi et ma belle famille

A ma tante Sadia

A toute ma famille

A ma binôme Lydia avec laquelle j'ai eu le plaisir de partager ce travail

A tous mes amis de la promotion du département des Mathématiques.

Ferroudja ♥

Table des matières

| | |
|--|-----------|
| Introduction générale | 7 |
| 1 Concepts de base | 9 |
| 1.1 Introduction | 9 |
| 1.2 Théorie des graphes | 9 |
| 1.2.1 Concepts de base | 10 |
| 1.2.2 Connexité | 15 |
| 1.2.3 Représentation d'un graphe en machine : | 15 |
| 1.3 Quelques classes de graphe | 17 |
| 2 Théorie de la complexité, Domination, Broadcast dominant | 19 |
| 2.1 Introduction | 19 |
| 2.2 La théorie de la complexité | 19 |
| 2.2.1 Définitions : | 19 |
| 2.2.2 Concept de base | 20 |
| 2.2.3 Problème de décision et d'optimalité | 21 |
| 2.2.4 La notation de grand O : | 21 |
| 2.2.5 Classes de complexité P et NP | 21 |
| 2.2.6 Classes de complexité NP-Complet | 22 |
| 2.3 La domination | 22 |
| 2.3.1 La domination dans les graphes | 22 |
| 2.3.2 Généralisation de la domination | 23 |
| 2.4 Broadcast | 24 |
| 2.4.1 Introduction | 24 |
| 2.4.2 Définitions et terminologies | 25 |
| 2.4.3 Quelques résultats fondamentaux | 27 |
| 3 Recherche de Broadcast dominant dans les graphes Séries-Parallèle | 28 |
| 3.1 Introduction | 28 |
| 3.2 Résultats fondamentaux | 28 |
| 3.2.1 Bornes sur le nombre de broadcast dominant | 29 |
| 3.3 Broadcast dominant dans les graphes série-parallèle | 29 |

| | | |
|----------|--|-----------|
| 3.3.1 | Introduction : | 29 |
| 3.3.2 | Broadcast dans les graphes séries-parallèles | 30 |
| 3.3.3 | Initialisation | 32 |
| 3.3.4 | Opération en série : | 33 |
| 3.3.5 | Opération en parallèle : | 34 |
| 3.3.6 | L'algorithme | 35 |
| 3.3.7 | Exemple d'application : | 37 |
| 4 | Etude de la chaîne P_n et Implementation | 43 |
| 4.1 | Introduction | 43 |
| 4.2 | Broadcast efficace dans les Chaînes P_n | 43 |
| 4.2.1 | Invariants de broadcast domination dans la classe chaîne | 43 |
| 4.2.2 | Broadcast dominant efficace d'une chaîne P_n | 44 |
| 4.3 | Implémentation | 51 |
| 4.3.1 | Introduction | 51 |
| 4.3.2 | Univers de travail : Langage C | 51 |
| 4.3.3 | Exemples d'application | 51 |
| | Identification des problèmes | 51 |
| 4.3.4 | Implémentation | 52 |
| | Problème 1 : | 53 |
| | Problème 2 : | 55 |
| | Conclusion générale | 57 |
| | Bibliographie | 57 |
| | Résumé | 59 |

Introduction générale

Sans que l'on en soit forcément conscient, la recherche opérationnelle (R.O) prend de nos jours le carrefour des différentes sciences et technologies. Née pendant la seconde guerre mondiale, dans le but de fournir des techniques d'optimisation des ressources militaires, ses premières applications avaient trait aux opérations militaires d'où justement le qualificatif "opérationnelle". Depuis l'apparition de l'ordinateur, les techniques de la (R.O) se sont développées et son champ d'application s'est élargi à d'autres domaines tels : le Marketing et le commerce, l'économie, et fait son entrée dans le monde de l'entreprise après les années 50.

La (R.O) est une discipline des mathématiques appliquées qui fait partie de l'aide à la décision, pouvant être défini tel un ensemble de techniques rationnelles orientées vers la recherche du meilleur choix et ceci par la modélisation et l'optimisation afin d'aboutir au meilleur résultat possible, ce domaine fait appel aux raisonnements mathématiques (logique, probabilités, analyse) ainsi qu'à la modélisation, l'optimisation.

La théorie de graphes est une branche des mathématiques, qui se rattache à la programmation linéaire, la programmation convexe, et à la topologie. Elle est un support très puissant à la modélisation et la résolution d'une grande famille de problèmes rencontrés en optimisation. Ainsi, parmi les problèmes les plus connus en théorie des graphes, on en cite "la domination", ainsi que celui traité dans ce mémoire qui n'est rien d'autre que la recherche "du broadcast domination" dans une certaine classe de graphes.

Le problème du broadcast domination est une généralisation du problème de la dominance. Introduit en 2003 par *D. J. Erwin*, consiste à trouver une fonction f définie sur l'ensemble des sommets V d'un graphe G connexe, non orienté à valeurs entières positives telles que : $f(v) \leq e(v)$, pour tout sommet v de G .

Si f est telle que : $f(v) > 0$, pour tout sommet v de G , ou il est à distance $\leq f(u)$ d'un autre sommet u de G tel que $f(u) > 0$, alors on dira que f est un broadcast dominant. Ce dernier devient optimal s'il réalise le minimum de la somme des coûts de cette fonction f sur l'ensemble des sommets de G . Ainsi, ce dernier (minimum) va réaliser l'optimalité du broadcast dominant et va être appelé : le nombre du broadcast domination, et sera noté $\gamma_b(G)$.

L'objet principal de ce travail est l'étude du nombre de broadcast dominant optimal dans la classe des Graphes séries-parallèles, ainsi que dans la chaîne P_n . Après une introduction, le mémoire va s'articuler sur quatre chapitres :

- Dans le premier chapitre nous introduisons un certain nombre de définitions et notions fondamentales en théorie des graphes, celles-ci vont être utilisées tout au long de ce mémoire.

- Le deuxième chapitre est consacré essentiellement à l'approche algorithmique et aux principaux concepts de la complexité algorithmique, ce qui nous permet de donner enfin, quelques éléments essentiels du problème de dominance, et ceux du broadcast dominant optimal.

- Ce n'est qu'en troisième chapitre, que nous traitons enfin notre problème principal, de la recherche du broadcast dominant optimal dans une classe spécifique des graphes, celle des graphes séries- parallèles, tout en exposant un algorithme linéaire permettant de déterminer le nombre de broadcast dans cette classe.

- Dans le quatrième et dernier chapitre, nous énonçons les différents résultats des invariants de la domination et du Broadcast dominant de la chaîne P_n . Nous étudions par la suite le Broadcast dominant efficace dans cette famille, en terminant le chapitre par une implémentation qui sera faite sur le Langage C.

- Nous terminons notre travail avec une conclusion générale et quelques perspectives.

Chapitre 1

Concepts de base

1.1 Introduction

Ce premier chapitre est dédié entièrement, aux concepts de base de la Théorie des Graphes. En effet, il est essentiel de définir le vocabulaire et les outils de base, assurant la bonne compréhension des notions introduites dans les chapitres ci-après. En outre, dans le but de rendre le travail plus utile, nous avons décidé d'associer chacune des notations inédites introduites, par sa traduction anglo-saxonne mise entre parenthèses.

1.2 Théorie des graphes

Au XVIII (18ème) siècle, les premiers travaux de la théorie des graphes ont été présenté par le mathématicien suisse " *Leonhard Euler*", à qui les notables de sa ville "Königsberg" demandèrent s'il était possible de parcourir les ponts de la ville en passant sur chacun des sept ponts exactement une seule fois (voir Figure 1.1), puis y retourner au point de départ.

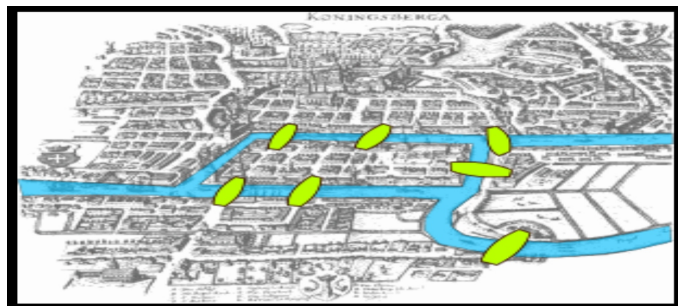


FIGURE 1.1 – La ville des 7 ponts

Ce genre de problèmes est très souvent rencontré dans les problèmes de tournées, du type facteur ou ramassage de déchets ménagers ; dans lesquels il faut parcourir les rues d'une ville de façon optimale, "Euler" a trouvé la solution en 1736 [6]. Néanmoins, la théorie des graphes a été

ensuite utilisé également dans l'étude de plusieurs autres problèmes dont le problème de coloriage de cartes/graphes, ou même celui de marche du cavalier sur l'échiquier.

Depuis son fondement, la théorie des graphes n'a cessé de s'étendre, après avoir été considérée telle une branche non-indépendante des mathématiques, à partir du XX (20ème) siècle, elle a ensuite été appliquée, notamment dans l'étude des circuits électriques, en chimie, en psychologie en science sociale, et en biologie, etc. Enfin, les graphes constituent une méthode de modélisation et de résolution, qui permet l'optimisation et l'aide à la décision, la stratégie, dans la représentation des réseaux de communication, réseaux routiers, etc.

1.2.1 Concepts de base

Définition 1.1 : Graphe non orienté

- Un graphe non orienté est noté $G(V, E)$ avec $V \neq \emptyset$, où :
 $V = (v_1, v_2, \dots, v_n)$ est l'ensemble des sommets du graphe supposé non vide et $E = (e_1, e_2, \dots, e_m)$ l'ensemble des arêtes.

On pose $|V| = n$ nombre de sommets de G dit ordre de G et $|E| = m$ nombre d'arêtes de G dit largeur de G .

Si $E = \emptyset$, alors G est dit un graphe sans arêtes.

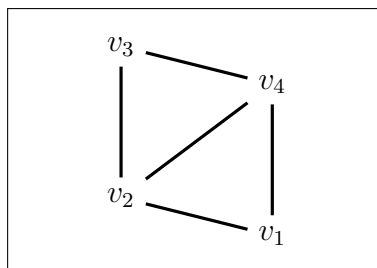


FIGURE 1.2 - Graphe non-orienté

- Deux sommets sont dits adjacents s'ils sont reliés par une même arête, et deux arêtes sont dites adjacentes si elles ont une extrémité commune.

- Dans un graphe, on peut avoir deux (ou plusieurs) arêtes qui ont les mêmes extrémités, elles sont appelées arêtes parallèles. On définit alors la multiplicité de G , $m(G)$ comme étant le maximum d'arêtes liant deux mêmes sommets. Si $m(G) = p$ alors G est dit un p -graphe.

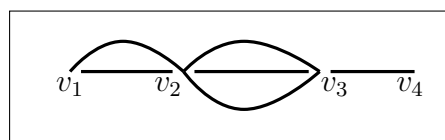


FIGURE 1.3 - 3-graphe

- On peut aussi avoir une arête telle que ses deux extrémités soient identiques, c'est-à-dire $v = v$, dans ce cas elle est appelée une boucle.

- Un graphe qui a des arêtes parallèles et des boucles est appelé un pseudographe. Ainsi, un graphe est dit simple s'il ne contient ni d'arêtes parallèles ni de boucles.

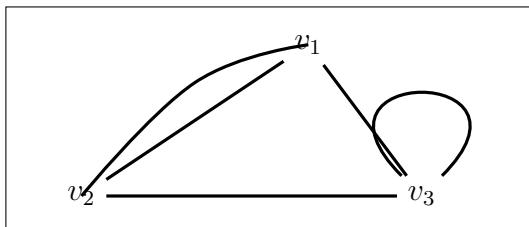


FIGURE 1.4 - Pseudographe (2-graphes)

Définition 1.2 : Graphe orienté

- Un graphe orienté est noté $G(V, U)$ avec $V \neq \emptyset$, obtenu depuis le graphe $G = (V, E)$ muni d'une orientation de ses arêtes où :

$U = (u_1, u_2, \dots, u_m)$ est l'ensemble de ses arcs.

- Si $u = (v, w) \in U$, alors l'arc est orienté de v vers w .

- Soient T et I deux applications des extrémités initiales et terminales de U dans V . Si $u = (v, w)$ est un arc de G , on dit que v est l'extrémité initiale de u notée $I(u) = v$, et w est l'extrémité terminale de u notée $T(u) = w$.

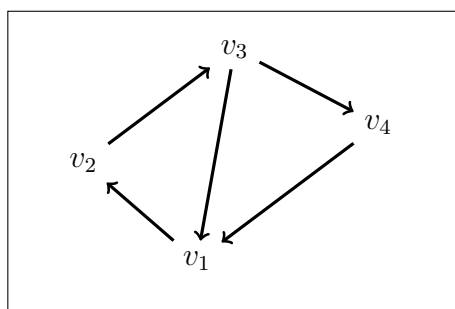


FIGURE 1.5 - Graphe Orienté

Définition 1.3 : Incidence, Adjacence

Incidence :

Soit $u = (v, w) \in U$ un arc de G , alors u est incident à w vers l'extérieur, et w est incident à u vers l'intérieur. Notons ainsi les ensembles suivants :

$X^+(v) : \{ \text{L'ensemble des arcs incidents vers l'extérieur au sommet } v, \text{ c'est-à-dire les arcs qui "partent" de } v \}.$

$X^-(v) : \{ \text{L'ensemble des arcs incidents vers l'intérieur au sommet } v, \text{ c'est-à-dire les arcs qui "arrivent" à } v \}$.

Remarque 1.2.1 : Une boucle (v, v) est un élément à la fois de $X^+(v)$ et de $X^-(v)$.

Adjacence :

- Si $G(V, E)$ est un graphe non orienté, avec $(v, w) \in E \Leftrightarrow v$ et w sont adjacents.
- Si $G(V, U)$ est un graphe orienté, avec $(v, w) \in U \Leftrightarrow w$ est adjacent à v .

Définition 1.5 : Voisinage, Degré d'un sommet

Voisinage d'un sommet :

- Soit $G(V, E)$, le voisinage ouvert d'un sommet v de G noté $N_G(v)$ est l'ensemble :

$$N_G(v) = \{w \in V / vw \in E\}$$

- Le voisinage fermé d'un sommet v est l'ensemble :

$$N_G[v] = N_G(v) \cup \{v\}$$

- Le voisinage ouvert d'un ensemble S noté $N(S) = \bigcup_{v \in S} N(v)$.

Degré d'un sommet :

- Le degré d'un sommet v noté $d_G(v)$ est égal au nombre d'arêtes incidentes à v (une boucle est comptée deux fois).

- Si G est simple et orienté, on a

$$d_G(v) = |N(v)| = |N^+(v)| + |N^-(v)|$$

avec : d_G^+ demi degré intérieur, d_G^- demi degré extérieur.

Définition 1.6 : Successeur, Prédécesseur

Si $(v, w) \in U$ de G , on dit que w est un successeur de v , et v est un prédécesseur de w .

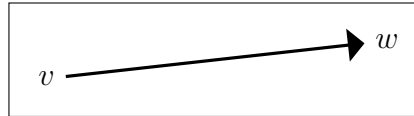
- L'ensemble des successeurs est donc l'ensemble noté :

$$\Gamma^+(v) = \{w \in V : (v, w) \in U\}$$

- L'ensemble des prédécesseurs est noté par :

$$\Gamma^-(v) = \{w \in V : (w, v) \in U\}$$

- Dans cet exemple, v est le prédécesseur de w , et w est le successeur de v , pour $u = (v, w) \in U$.

FIGURE 1.6 - L'arc $u=(v,w)$

Définition 1.7 : Chaîne, cycle

• Une Chaîne P_n , $n > 0$ est une séquence successive de sommets et d'arêtes de la forme $P_n=(v_0, e_1, v_1, e_2, v_2, \dots, e_{n-1}, v_n)$ dont v_0 et v_n sont les extrémités initiale et finale de P_n .

On pose $|P_n|$ le nombre des arêtes de P_n , dite la taille de P_n ou sa longueur.

Il existe deux types de chaînes : La chaîne simple est une chaîne ne passant pas deux fois par une même arête e , c'est-à-dire dont toutes les arêtes sont distinctes. La chaîne élémentaire est une chaîne ne passant pas deux fois par un même sommet, c'est-à-dire dont tous les sommets sont distincts également. voici une chaîne qui est simple et élémentaire à la fois :

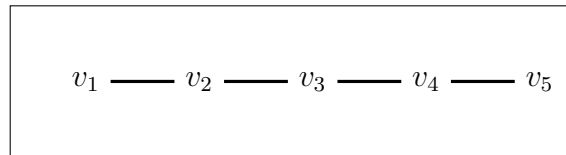


FIGURE 1.7 - Un exemple de chaîne de longueur 4

- Un cycle est une chaîne fermée dont les extrémités coïncident, on la note $(v_0, v_1, v_2, \dots, v_k = v_0)$. Un graphe est dit acyclique s'il ne possède pas de cycle.

Dans l'exemple de la FIGURE 1.8 le cycle est de taille 6.

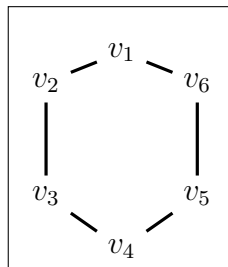
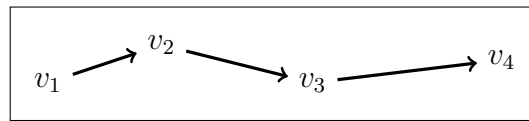


FIGURE 1.8 - Exemple de cycle

Définition 1.8 : Chemin, circuit

- Un chemin est une chaîne dans toutes les arêtes sont dans le même sens.

FIGURE 1.9 - Exemple de chemin, orienté de v_1 vers v_4

- Ainsi un circuit est un cycle dont les arêtes sont toutes orientées dans le même sens.

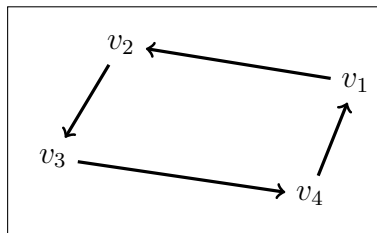


FIGURE 1.10 - Exemple de circuit

Définition 1.9 : Stable (stable set or independant set)

Un stable est un sous-ensemble de sommets deux à deux non adjacents.

Définition 1.10 : Distance

La distance entre deux sommets v et w notée $d(v, w)$ est la longueur de la plus courte chaîne reliant v à w dans G .

Définition 1.11 : Excentricité

L'excentricité d'un sommet $v \in V$ noté $e(v)$ est la plus grande distance entre le sommet v et n'importe quel autre sommet du graphe c-à-d :

$$e(v) = \max_{w \in V} d(v, w)$$

Définition 1.12 : Diamètre

Le diamètre d'un graphe G noté $diam(G)$ et l'excentricité maximum sur tous les sommets de G c-à-d :

$$diam(G) = \max_{v \in V} e(v)$$

Définition 1.13 : Rayon

Le rayon d'un graphe G noté $rad(G)$ est l'excentricité minimum sur tous les sommets de G c-à-d :

$$rad(G) = \min_{v \in V} e(v).$$

1.2.2 Connexité

Définitions :

Définition 1.14 : Connexité

$G = (V, E)$ est dit connexe (connected), si pour toute paire de sommets v et w dans V : v et w sont reliés par une chaîne.

Définition 1.15 : Composante connexe

$\forall v \in V, \forall w \in V : vRw \iff$ il existe une chaîne reliant v et w , R est une relation d'équivalence sur $V \times V$.

Définition 1.16 : Graphe connexe

- Un graphe connexe est un graphe tel que pour toute paire v, w , il existe une chaîne qui relie v et w .

- G est connexe = G est un seul bloc = on ne lève pas la main en le dessinant ; intuitivement, le nombre de composantes connexes est égale exactement au nombre de fois qu'on lève la main quand on le dessine.

- Un ensemble/ un sommet d'articulation (cuset / cutvertex) est un ensemble/ sommet dont la suppression engendre la division du graphe et le rend non-connexe.

- Une composante connexe d'un sommet est une classe d'équivalence $v = \{w \in V / vRw\}$, donc G est connexe si et si seulement si, il admet une seule composante connexe.

Définition 1.17 : Forte connexité

G est dit fortement connexe si $\forall v \in V \exists$ un chemin de v vers $w, \forall w \in V$.

Définition 1.18 : Boule ouverte, fermée

Si $G(V, E)$ est un graphe, alors soient la boule ouverte (fermée respectivement) de centre v et de rayon r :

$$B(v, r) = \{w \in V, d(v, w) < r\}$$

,

$$\overline{B(v, r)} = \{w \in V, d(v, w) \leq r\}$$

.

1.2.3 Représentation d'un graphe en machine :

Si G est un graphe, on peut représenter G en machine, entre autres, par sa matrice associée ; d'adjacence ou d'incidence.

Définition 1.19 : la matrice associée

Une matrice associée à G est une matrice $n \times n$, où chaque ligne et chaque colonne correspondent à un sommet de G , d'éléments a_{ij} où

$$a_{ij} = \begin{cases} 1, & \text{si l'arc } (i, j) \in U; \\ 0, & \text{sinon.} \end{cases}$$

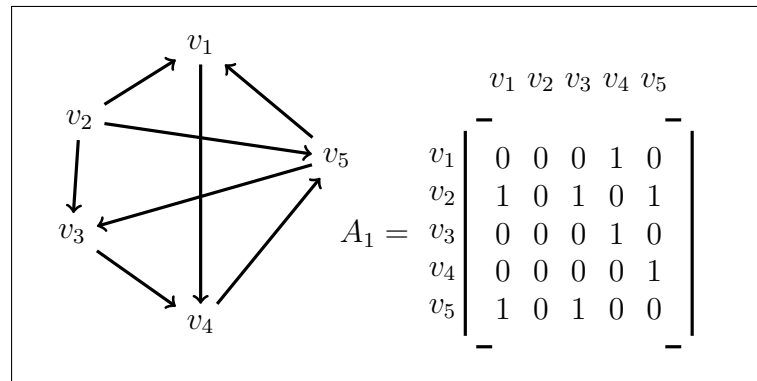


FIGURE 1.11 - $G(V, U)$ et sa matrice associée

Définition 1.20 : la matrice d'adjacence

La matrice d'adjacence du graphe $G = (V, E)$ est une matrice $n \times n$, ses éléments peuvent prendre deux valeurs 1 ou 0. Chaque ligne et chaque colonne correspondent à un sommet du graphe. Ainsi chaque élément de la matrice indique la relation qui existe entre deux sommets

$$a_{ij} = \begin{cases} 1, & \text{si } (i, j) \in E; \\ 0, & \text{sinon.} \end{cases}$$

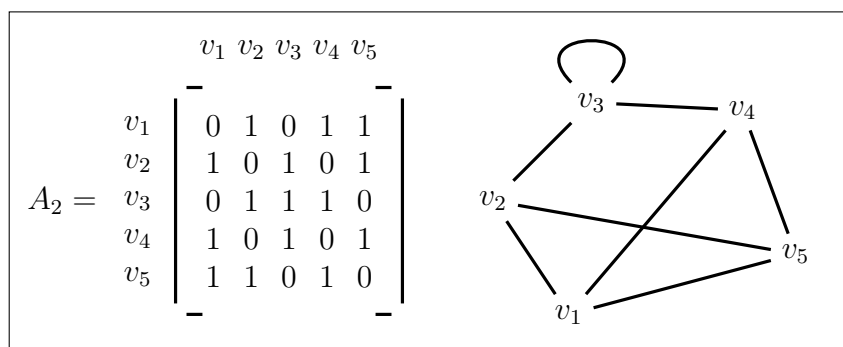


FIGURE 1.12- $G(V, E)$ et sa matrice d'adjacence

Définition 1.21 : la matrice d'incidence aux arcs

La matrice d'incidence aux arcs d'un graphe $G = (V, U)$ est une matrice $n \times m$, ses éléments sont 1, 0 ou -1. Chaque ligne de la matrice est associée à un sommet et chaque colonne à un arc. Tout élément de la matrice indique la relation entre un sommet et un arc comme suit

$$a_{ij} = \begin{cases} 1, & \text{si } I(u_j) = v_i; \\ -1, & \text{si } T(u_j) = v_i; \\ 0, & \text{sinon.} \end{cases}$$

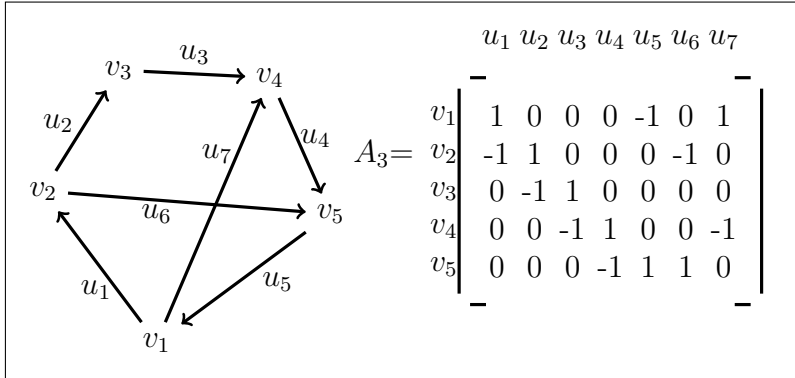


FIGURE 1.13- $G(V, U)$ avec sa matrice d'incidence

1.3 Quelques classes de graphe

Voici les graphes les plus utilisés en théorie des graphes, dont l'un d'entre eux nous sera utile dans la suite.

Sous-graphe (Subgraph) :

Un graphe $G' = (V', E')$ est un sous-graphe du graphe $G = (V, E)$ si $V' \subseteq V$ et $E' \subseteq E$ et toutes les arêtes de E' ont leurs extrémités dans V' .

Graphe complet (complet graph) :

Un graphe complet est un graphe simple noté K_n , tel quel $\forall v, w$ tq $(v, w) \in E$.

Un arbre (tree) :

Un arbre est un graphe connexe sans cycle, exemple,

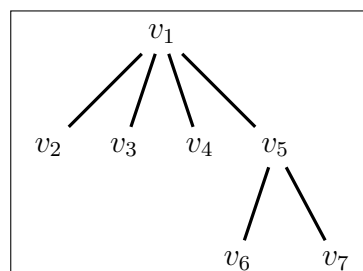


FIGURE 1.14 - Arbre/ Tree

Dans un arbre un sommet de degré 1 est dit pendant, ou bien une feuille; $(v_2, v_3, v_4, v_6, v_7)$ dans l'exemple ci-dessus.

Corollaire : Soit n le nombre de sommets d'un graphe G , et m le nombre de ses arcs alors :

G est connexe $\Rightarrow m \geq n - 1$.

G est sans cycle $\Rightarrow m \leq n - 1$.

$\Rightarrow G$ un arbre implique que : $m = n - 1$.

Les graphes séries- parallèles (SP-Graphs) :

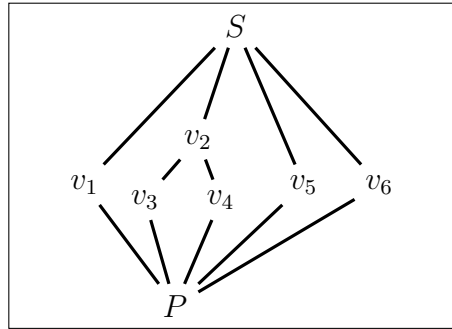


FIGURE 1.15 - Graphe série-parallèle

Les SP-Graphes appartiennent à la famille des multi-graphes. A l'inverse du graphe simple, un multi-graphe autorise des liens multiples appelés liens parallèles (multiple edges ou parallel edges). Un graphe série-parallèle noté $G(V, (s, t), E)$ admet deux sommets spéciaux qui sont la source s et le puits t , et est formé à partir de deux opérations de composition de base ; la composition en série et la composition en parallèle.

La composition en série de deux graphes série-parallèle G_1 et G_2 revient à combiner le puits de G_1 c-à-d pG_1 avec la source de G_2 c-à-d sG_2 , ainsi on obtient le graphe $:G = s(G_1, G_2)$. La composition en parallèle de G_1 et G_2 quant à elle, revient à combiner les deux puits des deux graphes G_1 et G_2 ensemble, et les deux sources également, ainsi on obtient le graphe série-parallèle final $G = p(G_1, G_2)$.

Ces graphes peuvent servir à modéliser des circuits électriques en série et en parallèle.

Chapitre 2

Théorie de la complexité, Domination, Broadcast dominant

2.1 Introduction

Dans ce second chapitre, nous exposons un petit résumé de la théorie de la complexité des algorithmes en s'appuyant sur les notions fondamentales des classes de problèmes : la classe P (Polynomial) et la classe NP (Non deterministic polynomial), ainsi que la classe $NP - Complet$ (exponential). Ensuite, nous touchons à quelques définitions de la domination. Et finalement nous présentons quelques résultats concernant le problème du broadcast dominant.

2.2 La théorie de la complexité

2.2.1 Définitions :

Définition 2.1 : Algorithme

La notion d'algorithme de résolution d'un problème est une procédure constituée d'une succession finie d'opérations élémentaires qui transforme les données fixées du problème se présentant sous n'importe quelle forme, en une solution. Elle couvre une grande partie des problèmes rencontrés dans la vie quotidienne, en Mathématiques en particulier et dans les autres sciences en général.

Définition 2.2 : Théorie de la complexité

La théorie de la complexité est une branche des mathématiques et de l'informatique ayant pour cadre l'étude de la difficulté des problèmes algorithmiques et de l'espace-temps, utilisé en parcourant le chemin menant à la solution, ensuite les classer en fonction de cette difficulté. Pour évaluer théoriquement l'efficacité d'un algorithme résolvant un problème \mathcal{P} , on compte le nombre d'opérations élémentaires que cet algorithme effectue pour le résoudre.

Définition 2.3 : La machine de Turing

La théorie de la complexité est basée sur un modèle de calcul abstrait ; une machine de Turing, celle-ci est un ruban infini muni d'une tête de lecture et d'écriture qui va lire l'instance (les données du problème), et écrire en même temps jusqu'à l'obtention d'une solution.

Définition 2.4 : Complexité algorithmique

Pour chaque algorithme il existe deux types de complexité, la complexité spatiale et la complexité temporelle.

Complexité spatiale d'un algorithme :

La complexité spatiale d'un algorithme est l'évaluation de l'espace mémoire utilisé pour l'exécution d'un algorithme donné. Chaque algorithme a une complexité spatiale propre à lui.

Dans ce qui suit on ne s'intéressera qu'à la complexité temporelle.

Complexité temporelle d'un algorithme :

C'est la mesure faite sur le temps d'exécution, utile pour aboutir à la solution avec l'algorithme, Celle-ci est spécifique à chaque algorithme.

Suite à cette dernière complexité, on distingue trois classes de problèmes, et pour bien définir ces classes on aura besoin des définitions suivantes :

2.2.2 Concept de base**Définitions 2.5 : Un problème[4]**

Un problème \mathcal{P} est une question générale possédant des paramètres dont la solution dépend de la valeur de ces paramètres, et il est généralement inconnu. Un exemple de problème est obtenu en spécifiant des valeurs particulières pour tous les paramètres de problème.

Définitions 2.6 : Une instance d'un problème

Une instance d'un problème \mathcal{P} est obtenue en affectant une valeur à chacun de ses paramètres.

Définitions 2.7 : La taille d'une instance

La taille d'une instance désigne généralement la valeur d'un ou de plusieurs de ces paramètres.

Exemple 2.1 : Prenons le problème du voyageur de commerce (ou TSP/Traveling salesman problem), étant donné un ensemble de villes séparées par des distances connues, ce problème consiste à trouver le plus court chemin qui relie toutes les villes, en ne passant qu'une seule fois par chaque ville. Une instance du TSP est donc un ensemble de n points (représentant les villes) définis chacun par un couple de coordonnées et la taille de cette instance est $2n + 1$ (il faut une case mémoire pour chaque coordonnée des n points et une autre pour stocker l'entier n).

2.2.3 Problème de décision et d'optimalité

Définition 2.8 : problème de décision (problème de reconnaissance) [4]

Un problème de décision est un problème dont les deux seules réponses possibles sont soit vraie soit faux, autrement dit la solution de ce problème est seulement vrai ou faux, une fois que la réponse est obtenue (oui ou non) le problème est alors résolu. Un problème de reconnaissance n'est pas, de ce fait un problème d'optimisation combinatoire mais, à tout problème d'optimisation combinatoire on peut faire associer un problème de reconnaissance en suivant la définition suivante.

Définition 2.9 : Problème d'optimisation [4]

Un problème d'optimisation consiste à déterminer la meilleure solution parmi toutes les solutions réalisables (minimum global ou maximum global).

2.2.4 La notation de grand O :

La notation grand O est une métrique permettant de décrire le temps d'exécution des algorithmes, mesuré en terme d'accroissement de la taille des données en entrée. On écrit $O(n)$ ou n est une variable qui représente la taille des données en entrée. On dit que la complexité d'un algorithme est en $O(f(n))$ s'il existe deux nombre c et n_0 tels que le temps mis par l'algorithme est borné supérieurement par $c_f(n)$ à partir d'un certain rang n_0 de n .

2.2.5 Classes de complexité P et NP

La classe P : [4] Un problème \mathcal{P} est dit dans la classe P s'il existe un algorithme le résolvant dont le temps d'exécution de cet algorithme est borné par un polynôme, en la taille n des données de \mathcal{P} . Autrement dit, un problème $\mathcal{P} \in P$ s'il existe un algorithme A tel que $\mathfrak{T}(I_A(n)) \in o(q(n))$, où $\mathfrak{T}(I_A(n))$ est le temps de résolution d'une instance I du problème \mathcal{P} par l'algorithme A et $q(n)$ est un polynôme en n . P est dite la classe des problème polynomiaux.

La classe NP : [4] Un problème de décision est dit NP si et seulement si on peut vérifier en un temps polynomial qu'une donnée est à réponse vrai.

Remarque 2.1 : NP ne veut pas dire non polynomial, mais non déterministe.

Exemple 2.2 : Problème de recherche d'un stable de cardinalité k . Considérons la version " décision " du problème du stable : étant donné un graphe G et un entier positif k , existe-t-il un stable de taille au moins k ? Ce problème est clairement dans NP : si l'on dispose d'un ensemble S de sommets, on peut vérifier en temps polynomial que $|S| \geq k$ et que S est stable (par exemple en examinant la liste d'adjacence de chaque sommet de S). Intuitivement, les problèmes de la classe NP sont ceux que l'on peut résoudre en énumérant l'ensemble des solutions possibles (méthode " brutale ") et en les testant à l'aide d'un algorithme polynomial. Naturellement, si on peut résoudre un problème avec un algorithme polynomial, on peut aussi vérifier en temps polynomial que la

solution fournie est bien une solution, par conséquent $P \subset NP$.

2.2.6 Classes de complexité NP-Complet

[4] Soit \mathcal{P} un problème de décision, on dira que :

$\mathcal{P} \in NP$ -complet si :

i- $\mathcal{P} \in NP$.

ii- Tout problème de reconnaissance appartenant à la classe NP se transforme polynomialement à \mathcal{P} .

Notons que P est la classe des problèmes les plus faciles et NP -complet est la classe des problèmes les plus difficiles.

Bien entendu, la classe P est incluse dans la classe NP mais $NP \subset P$ reste une conjecture redoutable, toujours ouverte. Celle-ci peut être représenté dans la représentation de la Figure 2.1.

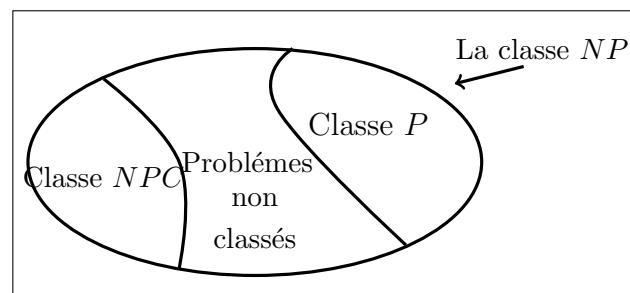


FIGURE 2.1 - Classes de complexité

La NP -complétude est une propriété très importante du fait que, si on aurait un algorithme polynomial pour un problème NP -complet alors on aurait un pour tous les problèmes de la classe NP ce qui donnerait $P=NP$. Montrer qu'un problème est NP -complet est chose très importante du fait qu'une fois l'appartenance d'un problème à la classe NP -complet est établie, l'existence d'un algorithme polynomial résolvant celui-ci devient peu probable mais non impossible.

2.3 La domination

Dans cette partie nous présentons quelques définitions de base et quelques paramètres liés à la domination.

2.3.1 La domination dans les graphes

Soit $G(V, E)$ un graphe,

Sommet dominant :

Un sommet v est dit dominant d'un sous ensemble A de sommets ne contenant pas v si $A \subseteq N(v)$, autrement dit, $\forall w \in A, w$ est adjacent à v .

Ensemble dominant :

- Un ensemble dominant d'un graphe $G = (V, E)$ est un ensemble $S \subseteq V$ de sommets tel que tout sommet de $V - S$ est adjacent au moins à un sommet de S , c'est-à-dire $N[S] = V$.

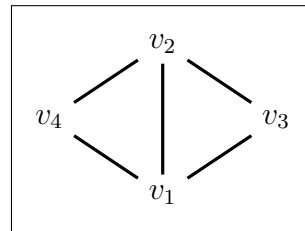
- Un dominant est un ensemble de sommets qui domine tous les autres sommets.

Exemple 2.3 : Dans le $G(V, E)$ ci-dessous on a :

$N(v_1) = \{v_2, v_3, v_4\}$ et $N[v_1] = \{v_1, v_2, v_3, v_4\} = V, \Rightarrow$ donc $S = \{v_1\}$ est un dominant.

De même $S' = \{v_4, v_3\}$ est un dominant mais $|S| < |S'|$.

FIGURE 2.1 Graphe dominé



Nombre de domination optimal : $\gamma(G)$ est le nombre de domination optimal du graphe G , s'il est égale à la cardinalité minimum d'un ensemble de domination c-à-d

$$\gamma_G = \text{Min}\{|S|, S \text{ ensemble dominant de } G \}.$$

2.3.2 Généralisation de la domination

- $S \subseteq V$ est dominant si $\forall v \in V$, soit : - $v \in S$;

- Soit $v \in \bar{S}$, et $\exists w \in S$ tel que $(v, w) \in E$.

$$\iff$$

$$N[S] = V \iff S \cup \{w \in \bar{S} / (v, w) \in E\}.$$

$$\iff$$

Autrement dit, si f est une application $f \equiv \mathbb{I}_S(v) = \begin{cases} 1 \text{ si } v \in S; \\ 0 \text{ sinon.} \end{cases}$

- Un ensemble S est dominant si $\forall v \in V$: - Soit $f(v) = 1$;

- Soit $f(v) = 0$ et $\exists w \in S$ tel que $f(w) = 1$

,

et $(v, w) \in E$, Donc $d(v, w) = 1$.

Ainsi $\gamma(G) = \min_{S \text{ dominant}} \sum_{v \in V} \mathbb{I}_S(v) = \min_{S \text{ dominant}} \{ \sum_{v \in V} \mathbb{I}_S(v), S \text{ dominant} \}$.

Domination double : [5]

- Soit $G = (V, E)$ un graphe simple, $S \subseteq V$ est un dominant double de G si, $\forall v \in S$, on a : $(N[v] \cap S) \geq 2$, c'est-à-dire le sommet $v \in S$ possède au moins un voisin dans S , ou bien $v \in (V - S)$ possède au moins deux voisins dans S .

- Le nombre de domination double noté $\gamma_2(G)$, c'est le cardinal minimum d'un ensemble dominant double de G .

Domination totale :

- Un ensemble S est dominant total d'un graphe G si tout sommet de V est adjacent à un sommet de S , c'est-à-dire $V = N[S]$.

- Le nombre de domination totale noté $\gamma_t(G)$ est le cardinal minimum d'un ensemble dominant total. Dans la domination total, les sommets de l'ensemble S sont eux aussi dominés par un autre sommet de l'ensemble.

Exemple 2.4 :

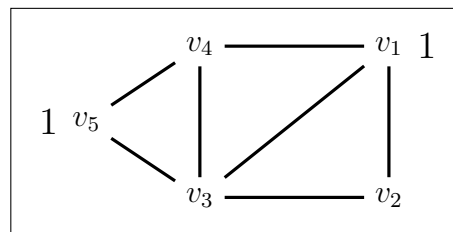


FIGURE $G(V, E)$

$V = \{v_1, v_2, v_3, v_4, v_5\}$, $S = (v_1, v_5)$ est un ensemble dominant-total, car :

$N[S] = N(S) \cup \{S\} = \{v_2, v_3, v_4\} \cup \{v_1, v_5\} = V$, ce qui fait que $\forall v \in V$, v est adjacent à un sommet de S .

2.4 Broadcast

2.4.1 Introduction

Une application du problème de la domination standard avait été présentée par "liu" en 1968, dont l'ensemble dominant représentait un ensemble de villes ayant des stations de diffusion, où chaque ville pouvait entendre une station de radio-diffusion placée dans cette ville, ou bien dans une ville voisine. Même si ce modèle était limité par le fait que toutes les stations de radio-diffusion

soient supposées avoir la même intensité du signal (puissance de transmission), mais n'empêche qu'il ait engendré plusieurs variantes [3].

Récemment (2003), "Erwin" a introduit le concept du broadcast domination qui n'est rien d'autre qu'une généralisation du "problème de la domination standard", tel que : les stations de radio-diffusion (c-à-d les sommets de l'ensemble dominant) sont autorisés à avoir des puissances de transmission différentes, et pas forcément les mêmes, autrement dit, on peut bien associer aux sommets v_i dans V des images de différentes puissances de domination ($f(v) > 0$) tel que tout sommet est sur une distance $f(v)$ d'un autre.

Le broadcast est optimale uniquement s'il cherche à minimiser la somme des coûts, ou des puissances assignées à tous les sommets du graphe G . Ces coûts sont généralement considérés justement comme les valeurs $f(v)$ [3].

Depuis l'introduction du problème de la domination, la plupart des problèmes de domination qui sont "exponentielles" de base (NP-hard) sur des graphes généraux, ont donné quelques indications sur le fait que le broadcast dominant optimal pourrait également être (NP-hard) pour les mêmes graphes. Par la suite, en 2003 "Blair" a donné des algorithmes en temps polynomial pour le broadcast dominant optimal de quelques classes de graphes dont on en cite : les arbres (Trees), les graphes d'intervalles (Interval Graphs), et les graphes séries-parallèles (Series Parallel Graphs)[9].

Dans cette partie nous montrons qu'en fait, assez étonnamment, le broadcast dominant appartient à la classe P (problème facile/ polynomial).

Il s'agit d'un modèle de diffusion plus réaliste que le problème de domination standard, puisque les émetteurs ne sont pas, en général, identiques. Par exemple, les stations de radio FM sont un émetteur avec un ERP (Effective Radiated Power) plus élevé pouvant ainsi transmettre davantage, mais il est plus coûteux à être construit et exploité. Sur cette base, le broadcast domination cherche à calculer une fonction de broadcast à valeurs entières f sur les sommets, de sorte que chaque sommet du graphe est à une distance qui vaut au plus $f(v)$ d'un sommet v tel que $f(v) > 0$. En utilisant ceci, nous donnons un algorithme polynomial calculant le broadcast dominant optimal de graphes quelconques[3].

2.4.2 Définitions et terminologies

Définition 2.9 : Broadcast

Soit $G = (V, E)$ un graphe, et soit $f : V \rightarrow \{0, 1, \dots, Diam(G)\}$.

- On dit que f est un broadcast de G si $\forall v \in V, 0 \leq f(v) \leq e(v)$.
- Si $f(v) \geq 1$, tout sommet de $B_f(u)$ est dominé par u , avec $B_f(u)$: la boule de centre u et rayon $f(u)$.
- L'ensemble des sommets dominateurs est défini par : $V_f = \{v \in V / f(v) \geq 1\}$.
- L'ensemble des sommets dominés est défini par : $V_0 = \{v \in V / f(v) = 0\}$.

- Pour tout sommet $v \in V$ l'ensemble des sommets dominants de v est défini par : $H_f(v) = \{u \in V_f, v \in B_f(u)\}$, avec $B_f(u) = \{v \in V, d(u, v) < f(u)\}$.

Définition 2.10 : Broadcast dominant

- On dit que f est un broadcast dominant si : $\forall u \in V, u \in V_f (f(u) > 0)$, ou bien $u \in V_0$ et $\exists v \in V_f$ tel que $d(u, v) \leq f(v)$.
- Le coût d'un broadcast dominant f est le nombre $C_f = \sum_{v \in V} f(v)$.

Définition 2.11 : Broadcast dominant optimal

- Le nombre du broadcast dominant optimal noté $\gamma_b(G)$ est le plus petit coût d'un broadcast dominant dans G , $\gamma_b(G) = \text{Min} \{C_f, \text{où } f \text{ est dominant}\}$.
- Si G n'est pas connexe alors $\gamma_b(G) = \sum_{C_i \text{ composante connexe}} \gamma_B(C_i)$.

Définition 2.12 : Broadcast efficace

Un broadcast domination est efficace si chaque sommet est dominé par un seul sommet, autrement dit $B_f(u) \cap B_f(v) = \phi, \forall u, v \in V_f$.

Définition 2.13 : Graphe de domination

- Le graphe de domination est le graphe $G_f(V_f, E_f)$ obtenu en associant à chaque boule $B_f(v)$ un sommet $v, \forall v \in V_f$. Deux sommets v et w sont adjacents dans G_f s'il existe un sommet v' de $B_f(v)$, et un sommet w' de $B_f(w)$ telle que $(v', w') \in E$
- Le graphe domination est toujours une chaîne ou un cycle, ce qui se prouve avec l'efficacité de f .

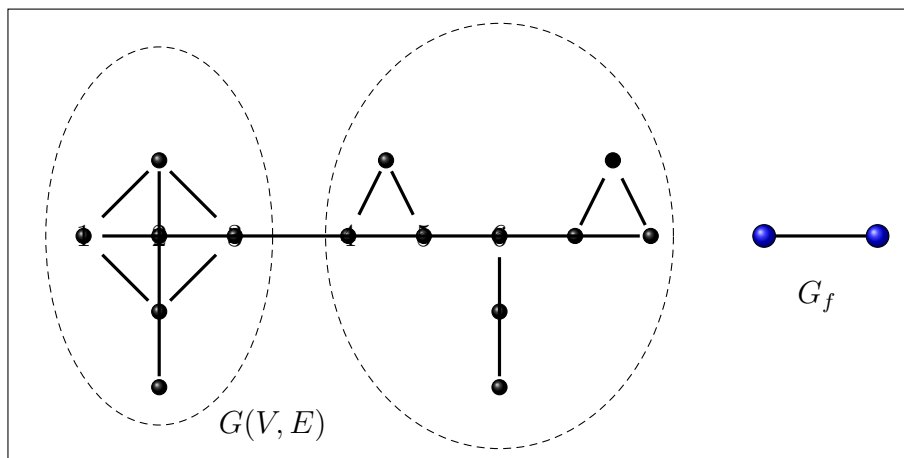


FIGURE 2.2 Graphe de domination qui est une chaîne

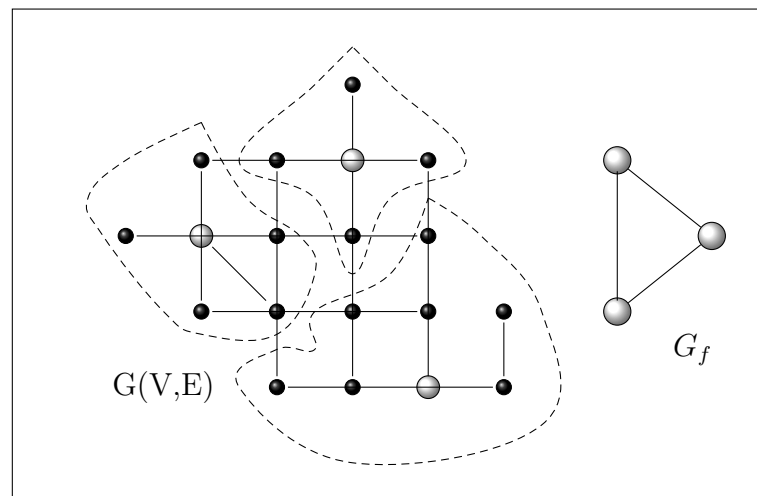


FIGURE 2.3 Graphe de domination qui est un cycle

2.4.3 Quelques résultats fondamentaux

Dans [13] D'après Dunbar et autres, chaque graphe G a un Broadcast dominant optimal efficace, c'est ce que ce Lemme propose de prouver.

Lemme 2.1 : [13] Pour tout Broadcast dominant non efficace d'un broadcast f dans G , il y a un broadcast dominant efficace F dans G tel que $|V_F| < |V_f|$ et $C_F = C_f$.

Lemme 2.2 : [14] Soit f un broadcast efficace dominant dans G . Si le graphe de domination G_f a un sommet de degré > 2 , il y a alors un broadcast dominant efficace F sur G tels que $|V_F| < |V_f|$ et $C_F = C_f$.

Lemme 2.3 : [14] Pour tout graphe G , il y a un broadcast dominant optimal efficace f tels que le graphe de domination G_f est une chaîne ou un cycle.

Lemme 2.4 : [14] Dans tout Broadcast dominant optimale f d'un graphe G , s'il y a deux sommets voisins v, w de V tels que $f(v) > 0$ et $f(w) > 0 \Rightarrow f(v) = f(w)$.

Chapitre 3

Recherche de Broadcast dominant dans les graphes Séries-Parallèle

3.1 Introduction

Dans ce chapitre, nous allons présenter un algorithme de recherche d'un broadcast dominant optimal dans la classe des graphes Séries-parallèle. Cependant, la construction de l'algorithme passe par un certain nombre de nouveaux résultats structuraux, sur le broadcast domination, enfin nous terminons le chapitre par un exemple d'un graphe série-parallèle pour lequel nous allons calculer la valeur $\gamma_b(G)$.

3.2 Résultats fondamentaux

[15] Pour tout graphe G et tout entier fixé k avec $1 \leq k \leq rad(G)$, le nombre de k -distance domination, noté $\gamma_k(G)$ est la plus petite cardinalité d'un ensemble de sommets S de G tel que tout sommet de G est à distance inférieure ou égale à k d'au moins un sommet de S .

Proposition 3.1 : [15] Pour tout graphe G ,

$$\gamma_b(G) \leq \min\{k\gamma_k(G) : 1 \leq k \leq rad(G)\}$$

En appliquant la proposition 3.1 pour $k = 1$ et $k = rad(G)$, on obtient :

$$\gamma_b(G) \leq rad(G) \text{ et } \gamma_b(G) \leq \gamma(G).$$

Corollaire 3.1 : [15] Dunbar et al. prouvent l'inégalité suivante, Pour un graphe G

$$\gamma_b(G) \leq \min\{rad(G), \gamma(G)\}$$

3.2.1 Bornes sur le nombre de broadcast dominant

Une borne supérieure évidente pour le nombre de domination est le nombre de sommets de graphe qui est égale à n , et puisqu'il faut au moins un sommet pour dominer un graphe, nous avons alors pour tout graphe G d'ordre n :

$$1 \leq \gamma_b(G) \leq n$$

Un graphe peut atteindre cette borne inférieure, si et seulement si il possède un sommet de degré $n - 1$, et il atteint la borne supérieure n si et seulement si $G = \bar{K}_n$, c'est-à-dire G est un graphe sans arêtes.

Théorème 3.1 [4] Si G est un graphe sans sommets isolés, alors :

$$\gamma(G) \leq \frac{n}{2}$$

3.3 Broadcast dominant dans les graphes série-parallèle

3.3.1 Introduction :

[3] Dans ce qui suit nous décrivons un algorithme de programmation dynamique pour calculer une fonction du broadcast à coût minimum qui domine un graphe série-parallèle.

Tout au long de cette section, nous supposons que le graphe donné est un graphe série-parallèle.

De manière informelle, une classe de graphe récursive est une classe dont laquelle n'importe quel membre assez grand peut être formée en joignant successivement les membres les plus petits de la classe à des sommets spécifiques appelés terminaux (bornes), un graphe qui admet une définition récursive a toujours deux sommets, une source et un puits . Les graphes séries-parallèle sont 2-arbres partiels, une classe de graphes récursifs avec 2 terminaux. Nous utilisons $t_L(G)$ et $t_R(G)$ pour désigner respectivement la borne gauche et la borne droite de G . Chaque membre de la classe peut ensuite être décomposé en graphe de base, généralement considères comme un seul bord, c'est-à-dire le K_2 . Cette décomposition est spécifiée par un arbre de décomposition qui identifie deux descendants pour chaque noeud non-frontal de l'arbre (sans feuille), montrant ainsi comment le graphe série-parallèle avait été construit, et une opération associée qui combine ces descendants pour créer le parent. Les graphes séries-parallèles peuvent être reconnus et un arbre de décomposition correspond peut être construit en temps linéaire.

Soit $G = (V, \{t_L(G), t_R(G)\}, E)$ et soit $G_j = (V, \{t_L(G_j), t_R(G_j)\}, E)$ pour $j = 1, 2$ des graphes 2-terminaux. On définit l'opération en série comme $s(G_1, G_2) = G$ si $t_L(G_1) = t_L(G)$, $t_R(G_1) = t_L(G_2)$ et $t_R(G_2) = t_R(G)$. Cette opération associe $t_R(G_1)$ à $t_L(G_2)$ puis le sommet résultant perd son statut de terminal. L'opération jackknife gauche, $left(G_1, G_2)$ est la même que la série, sauf que le sommet associé conserve son statut terminal, devenant $t_R(G)$, et $t_R(G_2)$ perd son statut terminal. L'opération droite jackknife $right(G_1, G_2) = G$ est analogue. Enfin, on définit l'opération parallèle comme $p(G_1, G_2) = G$ si $t_L(G_1) = t_L(G_2) = t_L(G)$ et $t_R(G_1) = t_R(G_2) = t_R(G)$. Notons que dans chaque cas, le graphe résultant G à deux terminaux.

Exemple 3.1 :

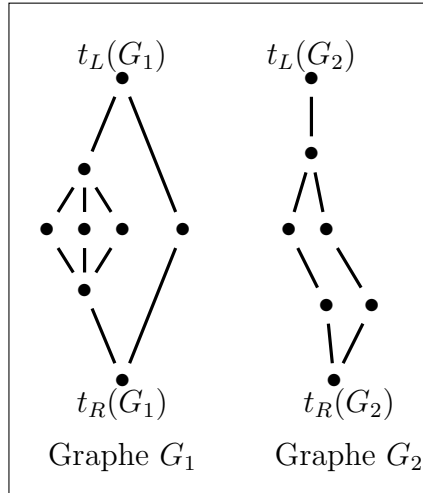


FIGURE 3.1 - Graphes G_1 et G_2

Les compositions en série et en parallèle des graphes G_1 et G_2 de la FIGURE 3.1 donnent les graphes série-parallèle finaux G de la FIGURE 3.2 :

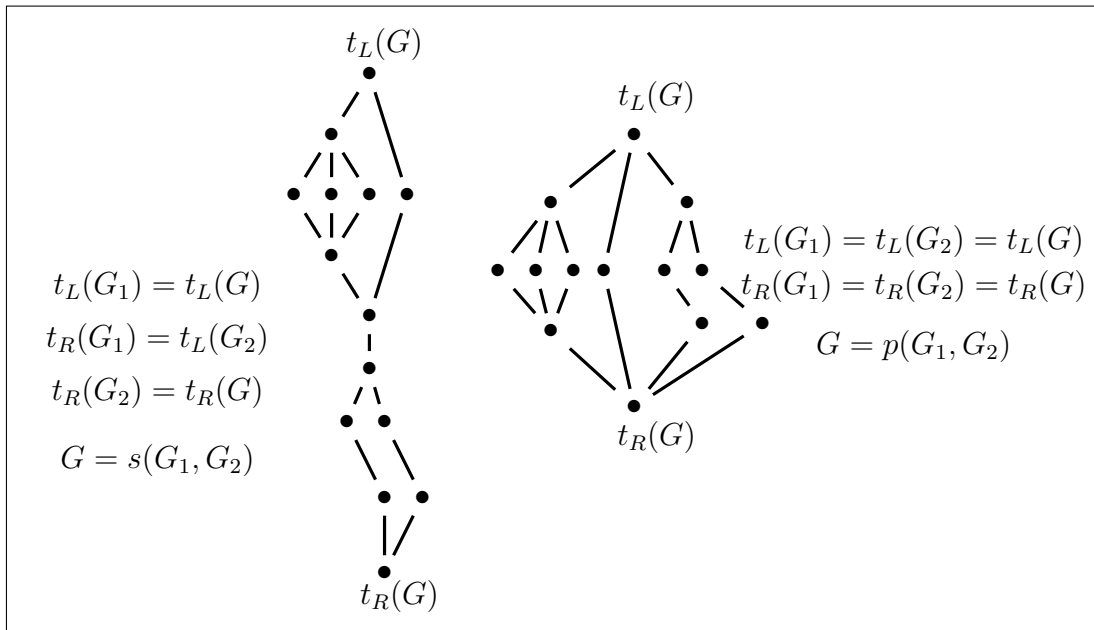


FIGURE 3.2 - Compositions en série et en parallèle

3.3.2 Broadcast dans les graphes séries-parallèles

[3] Soit f une fonction du broadcast $f : V \rightarrow \{0, 1, 2, \dots, r\}$, on dit que la condition de la domination (dom_L, dom_R) existe si les conditions suivantes sont vérifiées :

- ◇ f domine G_i sauf les sommets à distance $-dom_k$ de $t_k(G_i)$, $K \in \{L, R\}$ quand $dom_k < 0$.
- ◇ et f dominerait par le chemin hypothétique de longueur dom_k joint à $t_k(G_i)$ lorsque $dom_k \geq 0$.

Si $dom_k < 0$ on dit que $t_k(G_i)$ est sous-dominé et si $dom_k \geq 0$ on dit que $t_k(G_i)$ est sur-dominé. La définition de la condition de dominance autorise des parties de G_i d'être dominé au noeud courant dans l'arbre de décomposition uniquement lorsqu'il y a une exigence indiquée (la valeur négative de dom_k) qui exige que cette sous-dominance soit corrigée à un point plus proche de la racine de l'arbre de décomposition.

On peut maintenant définir $P_{G_i}(dom_L, dom_R)$ comme étant le coût minimum de la fonction broadcast, étant donné que la condition de la domination (dom_L, dom_R) existe dans G_i . P sera soit N, L, R ou B (neither, left, right, or both) terminaux ayant un broadcast non nul originaux. Par exemple $L_{G_3}(1, -2)$ représente le coût minimum d'une fonction broadcast en G_3 qui fournit une sur-dominance de 1 au terminal gauche, a une sous-dominance de -2 au niveau terminal droite, a un broadcast provenant de $t_L(G)$, et n'a pas de broadcast provenant de $t_R(G)$.

Les propriétés suivantes du broadcast dominant optimal seront utiles pour établir l'exécution d'un algorithme série-parallèle.

Lemme 3.1 : [3] Soit f la fonction du broadcast dominant optimal d'un graphe série-parallèle, et on considère $G_i = (V_i, \{t_L(G_i), t_R(G_i)\}, E)$ de la décomposition de G . Soit $d_{G_i}(t_L, t_R)$ la distance dans G_i entre $t_L(G_i)$ et $t_R(G_i)$.

1• si f appliquée a G_i est telle que : $dom_L = f(t_L(G_i)) > 0$ et $dom_R < 0 = f(t_R(G_i))$ alors

$$-(dom_R + d_{G_i}(t_L, t_R) + 1) < f(t_L(G_i))$$

2• si f appliquée a G_i est telle que : $dom_R = f(t_R(G_i)) > 0$ et $dom_L < 0 = f(t_L(G_i))$ alors

$$-(dom_L + d_{G_i}(t_L, t_R) + 1) < f(t_R(G_i))$$

Preuve : Nous ne démontrons que la première implication du Lemme 3.1, puisque la preuve de la deuxième est analogue. Supposons que la première implication soit fausse pour certains graphes G_i . Alors $-(dom_R + d((t_L(G_i), t_R(G_i)) + 1) \geq f(t_L(G_i))$, et la sous-dominance au sommet $t_R(G_i)$ exige une dominance pour tout sommet de G qui est déjà dominé par le broadcast provenant de $t_L(G_i)$. Mais après $f(t_L(G_i))$ peut être mis à 0 tout en maintenant la domination du broadcast dans G , contrairement à l'optimalité de f . •

Lemme 3.2 : [3] Pour tous broadcast dominant optimal f d'un graphe $G = (V, E)$, s'il existe deux voisins $v, w \in V$ tel que $f(v) > 0$ et $f(w) > 0$, alors $f(v) = f(w)$.

Preuve : Supposons le contraire. Sans perte de généralité, supposons $f(v) > f(w) > 0$ ou f est un broadcast dominant optimal de G avec $(v, w) \in E(G)$. Alors tout sommet dominé par le broadcast provenant de w est également dominé par le broadcast provenant de v . Mais alors $f(w)$ peut être mis à 0 tout en maintenant la domination du broadcast dans G , contrairement à l'optimalité de f . •

Étant donné un graphe série-parallèle G , l'algorithme calcul les valeurs de $N_{G_i}, L_{G_i}, R_{G_i}$ et B_{G_i} pour chaque graphe G_i dans un arbre de décomposition de G , de manière ascendante. Puisque la

fonction du broadcast dominant optimal aura $f(v) \leq r$ pour tout $v \in V$, nous restreignons les conditions de dominance de $-r$ à r . La sous-section suivante décrit comment calculer les tableaux N, L, R et B pour les feuilles. Les sous-sections en dessous présentent des relations récursives qui permettent le calcul de $P_{G_i}(dom_L, dom_R)$ pour $P \in \{N, L, R, B\}$ pour chaque graphe sans-feuilles G_i dans l'arbre de décomposition. Les équations décrites dans les trois sous-sections sont résumées dans les tableaux 1, 2 et 3.

3.3.3 Initialisation

[3] Rappelons que chaque feuille de l'arbre de décompositions d'un graphe série-parallèle correspond à un K_2 . La discussion suivante sur la façon d'initialiser les tableaux des coûts $N_{K_2}, L_{K_2}, R_{K_2}, B_{K_2}$ est résumé avec les équations du tableau 1.

Pour le cas de N_{K_2} (i.e., $f(t_L(K_2)) = f(t_R(K_2)) = 0$), aucun broadcast ne provient de l'intérieur du K_2 . Ensuite, conduire à un broadcast de domination valable du graphe série-parallèle, la condition de domination du K_2 doit indiquer une demande qui finira par dominer le K_2 . Les conditions de dominations indiquant une telle demande sont identifiées dans l'observation suivante.

Observation 3.1 : [3] Soit f le broadcast dominant d'un graphe série-parallèle et considérons $K_2 = (V_{K_2}, \{t_L(K_2), t_R(K_2)\}, E_{K_2})$ correspondant à une feuille dans la décomposition de G . Si f est appliqué a K_2 est telle que $f(t_L(K_2)) > 0$ et $f(t_R(K_2)) = 0$, alors $dom_R + dom_L \leq -2$.

On va utiliser la valeur ∞ pour représenter des conditions de dominations invalides. Tous les autres cas sont mis à 0, ce qui représente un coût nul lorsqu'aucun broadcast ne provient de K_2 . Rappelons que L_{K_2} représente $f(t_L(K_2)) > 0$ et $f(t_R(K_2)) = 0$, dans le cas ou le coût dû au K_2 est simplement $f(t_L(K_2))$. Ainsi nous devons entrer $f(t_L(K_2))$ pour toutes les conditions de domination ou $dom_L = f(t_L(K_2))$ et dom_R avec dom_L conduisent à un broadcast dominant valide du graphe final G .

Observation 3.2 : [3] Soit f le broadcast dominant d'un graphe série-parallèle et considérons $K_2 = (V_{K_2}, \{t_L(K_2), t_R(K_2)\}, E_{K_2})$ qui correspond à une feuille dans la décomposition de G . Si f appliqué à K_2 est telle que $f(t_L(K_2)) > 0$ et $f(t_R(K_2)) = 0$, alors $dom_R < dom_L$. Cette Observation nous donne la première ligne de l'équation pour L_{K_2} , la deuxième ligne est résultat du Lemme 3.2. L'initialisation de R_{K_2} est analogue.

Intuitivement, On doit définir $B_{K_2}(dom_L, dom_R) = \infty$ à chaque fois que la condition de la domination rend superflue l'une des valeurs de dom_L et dom_R . Les conditions dans lesquelles cela se produit on été formalisées dans le Lemme 1.3.3.2.

Tableau 1 :Initialisation pour l'algorithme SPBD

$$N_{K_2}(dom_L, dom_R) = \begin{cases} 0 & dom_L + dom_R \leq -2 \\ \infty & dom_L + dom_R > -2 \end{cases}$$

$$\begin{aligned}
 L_{K_2}(dom_L, dom_R) &= \begin{cases} \infty & dom_R \geq dom_L \\ \infty & dom_R \leq -(dom_L + 2) \\ dom_L & \text{sinon} \end{cases} \\
 R_{K_2}(dom_L, dom_R) &= \begin{cases} \infty & dom_L \geq dom_R \\ \infty & dom_L \leq -(dom_R + 2) \\ dom_L & \text{sinon} \end{cases} \\
 B_{K_2}(dom_L, dom_R) &= \begin{cases} dom_L + dom_R & (dom_L = dom_R > 0) \\ \infty & \text{sinon} \end{cases}
 \end{aligned}$$

3.3.4 Opération en série :

[3] Considérons un noeud G dans l'arbre de décomposition formé par $G = s(G_1, G_2)$, et considérons le calcul de L_G . Notons que les configurations autorisées avec un broadcast à $t_L(G)$ et aucun broadcast à $t_R(G)$; doivent provenir soit de B_{G_1} et L_{G_2} , soit de L_{G_1} et N_{G_2} . Les autres paires n'ont pas le broadcast originaire spécifié, ou bien sont incompatibles dans le sens où $t_R(G_1)$ ne peut être associé à $t_L(G_2)$ car l'un a un broadcast originaire et l'autre non.

Pour le cas de B_{G_1} et L_{G_2} , $t_L(G_1)$ et $t_L(G_2)$ ont un broadcast identique i . La formule compare simplement ces options pour toutes les valeurs i pertinentes et utilise la moins chère comme valeur candidate pour $L_G(dom_L, dom_R)$.

Notons le coût du broadcast originaire à $t_R(G_1) = t_L(G_2)$ est soustrait puisque son coût est représenté deux fois, à la fois dans le coût de domination de G_1 et dans le coût de domination de G_2 , mais il n'appartient qu'une seule fois dans le coût de domination de G .

Pour le cas de L_{G_1} et N_{G_2} aucune soustraction n'est nécessaire puisque $t_R(G_1)$ et $t_L(G_2)$ n'ont pas un broadcast d'origine. Maintenant soit G_1 fournit une sur-dominance qui doit au moins couvrir toutes les sous-dominance dans G_2 , ou vice versa, comme représenté par les deux dernières lignes de la formule $L_G(dom_L, dom_R)$. Une fois tous ces cas sont considérés, $L_G(dom_L, dom_R)$ se voit attribuer le coût minimum d'une fonction du broadcast en G avec une condition de dominance (dom_L, dom_R) qui a un broadcast provenant de $t_L(G)$ mais ne provenant pas de $t_R(G)$.

Des arguments similaires justifient les récursions données par N_G , R_G et B_G .

Tableau 2 : $G = s(G_1, G_2)$

$$N_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} R_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, j) + N_{G_2}(-i - 1, dom_R) \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, -i - 1) + N_{G_2}(j, dom_R) \end{cases}$$

$$L_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} B_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, j) + N_{G_2}(-i-1, dom_R) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, -i-1) + N_{G_2}(j, dom_R) \end{cases}$$

$$R_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} R_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, j) + R_{G_2}(-i-1, dom_R) \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, -i-1) + R_{G_2}(j, dom_R) \end{cases}$$

$$B_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} B_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, j) + R_{G_2}(-i-1, dom_R) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, -i-1) + R_{G_2}(j, dom_R) \end{cases}$$

3.3.5 Opération en parallèle :

[3] Considérons le calcul de $L_G(dom_L, dom_R)$ lorsque $G = p(G_1, G_2)$. Lorsque $dom_L \geq 0$, nous devons comparer les configurations dans G_1 et G_2 , où les deux graphes ont un broadcast d'origine approprié à t_L , et où un graphe fournit dom_R à t_R . L'autre graphe peut alors fournir entre $-(dom_R + 1)$ et dom_R à t_R . Garantissant ainsi que le graphe résultant fournira dom_R à t_R . L'autre moitié de $L_G(dom_L, dom_R)$ c'est-à-dire lorsque $dom_R < 0$ est calculé de manière analogue.

Des arguments similaires justifient les récursions données par N_G, R_G et B_G .

Tableau 3 : $G = p(G_1, G_2)$

$$B_G(dom_L, dom_R) = B_{G_1}(dom_L, dom_R) + B_{G_2}(dom_L, dom_R) - dom_L - dom_R$$

$$L_G(dom_L, dom_R) = \min \begin{cases} \min_{-(dom_R+1) \leq i \leq dom_R} \left\{ \begin{array}{l} L_{G_1}(dom_L, dom_R) + L_{G_2}(dom_L, i) - dom_L \\ L_{G_1}(dom_L, i) + L_{G_2}(dom_L, dom_R) - dom_L \end{array} \right\} & \text{if } dom_R \geq 0 \\ \min_{dom_R \leq i \leq -(dom_R+1)} \left\{ \begin{array}{l} L_{G_1}(dom_L, dom_R) + L_{G_2}(dom_L, i) - dom_L \\ L_{G_1}(dom_L, i) + L_{G_2}(dom_L, dom_R) - dom_L \end{array} \right\} & \text{if } dom_R < 0 \end{cases}$$

$$R_G(dom_L, dom_R) = \min \begin{cases} \min_{-(dom_L+1) \leq i \leq dom_L} \left\{ \begin{array}{l} R_{G_1}(dom_L, dom_R) + R_{G_2}(i, dom_R) - dom_R \\ R_{G_1}(dom_L, i) + R_{G_2}(dom_L, dom_R) - dom_R \end{array} \right\} & \text{if } dom_L \geq 0 \\ \min_{dom_L \leq i \leq -(dom_L+1)} \left\{ \begin{array}{l} R_{G_1}(dom_L, dom_R) + R_{G_2}(i, dom_R) - dom_R \\ R_{G_1}(i, dom_R) + R_{G_2}(dom_L, dom_R) - dom_R \end{array} \right\} & \text{if } dom_L < 0 \end{cases}$$

$$N_G(dom_L, dom_R) = \min \left\{ \begin{array}{l} \min_{-(dom_L+1) \leq i \leq dom_L; -(dom_R+1) \leq j \leq dom_R} \left\{ \begin{array}{l} N_{G_1}(dom_L, dom_R) + N_{G_2}(i, j) \\ N_{G_1}(dom_L, j) + N_{G_2}(i, dom_R) \\ N_{G_1}(i, dom_R) + N_{G_2}(dom_L, j) \\ N_{G_1}(i, j) + N_{G_2}(dom_L, dom_R) \end{array} \right\} \text{ si } dom_L, dom_R \geq 0 \\ \min_{-(dom_L+1) \leq i \leq dom_L; dom_R \leq j \leq -(dom_R+1)} \left\{ \begin{array}{l} N_{G_1}(dom_L, dom_R) + N_{G_2}(i, j) \\ N_{G_1}(dom_L, j) + N_{G_2}(i, dom_R) \\ N_{G_1}(i, dom_R) + N_{G_2}(dom_L, j) \\ N_{G_1}(i, j) + N_{G_2}(dom_L, dom_R) \end{array} \right\} \text{ si } dom_L \geq 0 \text{ et } dom_R < 0 \\ \min_{dom_L \leq i \leq -(dom_L+1); -(dom_R+1) \leq j \leq dom_R} \left\{ \begin{array}{l} N_{G_1}(dom_L, dom_R) + N_{G_2}(i, j) \\ N_{G_1}(dom_L, j) + N_{G_2}(i, dom_R) \\ N_{G_1}(i, dom_R) + N_{G_2}(dom_L, j) \\ N_{G_1}(i, j) + N_{G_2}(dom_L, dom_R) \end{array} \right\} \text{ si } dom_L < 0 \text{ et } dom_R \geq 0 \\ \min_{dom_L \leq i \leq -(dom_L+1); dom_R \leq j \leq -(dom_R+1)} \left\{ \begin{array}{l} N_{G_1}(dom_L, dom_R) + N_{G_2}(i, j) \\ N_{G_1}(dom_L, j) + N_{G_2}(i, dom_R) \\ N_{G_1}(i, dom_R) + N_{G_2}(dom_L, j) \\ N_{G_1}(i, j) + N_{G_2}(dom_L, dom_R) \end{array} \right\} \text{ si } dom_L, dom_R < 0 \end{array} \right.$$

3.3.6 L'algorithme

[3] Nous avons défini des initialisations pour des feuilles dans un arbre de décomposition pour un graphe série-parallèle, et des équations récursives pour calculer le coût du broadcast dominant optimal pour les deux opérations en série et en parallèle. Les formules pour les opérations jacknife sont une adaptation directe de celles pour l'opération en série. Dans l'intérêt de l'espace, elles sont omises. Un algorithme pour calculer le coût d'une solution optimale pour un graphe entier G trouverait simplement le coût minimum pour lequel il n'y a pas de sous-dominance dans les tableaux $\{N, L, R, B\}$ à la racine de l'arbre de décomposition. Les détails de cet algorithme, que nous appelons **SPBD** sont données ci-dessous. C'est donc, simple pour redescendre dans l'arbre pour trouver la fonction du broadcast dominant réelle f qui correspond à ce coût optimal.

Algorithme series-parallel Broadcast domination (SPBD)

Input : A series-parallel graph G and a decomposition tree T_d for G .

Output : $\gamma_b(G)$

$r = \text{rad}(G)$;

for each leaf G_i in T_d **do**

use the formulas in the table.1 to compute N_{G_i} , L_{G_i} , R_{G_i} and B_{G_i} .

repeat

$G_i =$ an unprocessed node in T_d whose children have processed ;

If $G_i = s(G_1, G_2)$ **then**

use the formulas in the table.2 to compute N_{G_i} , L_{G_i} , R_{G_i} and B_{G_i} .

Else if $G_i = p(G_1, G_2)$ **then**

use the formulas in the table.3 to compute N_{G_i} , L_{G_i} , R_{G_i} and B_{G_i} .

until the root of T_d has been processed ;

$x = \infty$

for $i = 0$ **to** r **do**

for $j = 0$ **to** r **do**

$x = \min\{x, N_G(i, j), L_G(i, j), L_G(i, j), L_G(i, j)\}$;

$\gamma_b(G) = x$;

Les résultats suivants vérifient que l'algorithme **SPBD** est correct et donnent une borne supérieure sur le temps d'exécution de l'algorithme.

Lemme 3.3 : [3] Chaque fonction d'un broadcast minimal possible sur un graphe série-parallèle G est considérée lors de l'exécution de l'algorithme **SPBD**

Preuve : Par Observation 3.2, Lemme 3.1 et Lemme 3.2, les seules fonctions du broadcast possible sur les feuilles qui ne sont pas incluses dans l'initialisation sont celles qui ne peuvent pas conduire à une solution optimale G . Ainsi, au stade initial de l'algorithme chaque fonction du broadcast minimal possible est représenté. A chaque noeud non-feuille de l'arbre de décomposition, chaque configuration combinée de G_1 et G_2 est considéré pour la rétention. Une configuration combinée n'est éliminée que lorsqu'une autre configuration combinée sans un plus mauvais coût est représentée dans la même cellule $P_G(dom_L, dom_R)$ pour $P \in \{N, L, R, B\}$.

Théorème 3.2 : [3] Étant donné un graphe série-parallèle, l'algorithme **SPBD** calcule une fonction de broadcast dominant optimal de G en temps $O(nr^4)$.

Preuve : L'exactitude découle du Lemme 3.3. Pour la complexité temporelle, notons qu'il n'y a pas plus de 2 noeuds dans l'arbre de décomposition d'un graphe série-parallèle. Pour chaque noeud, nous remplissons quatre $(2r + 1) \times (2r + 1)$ tableaux. Remplir les tableaux N_G pour une opération parallèle prend le plus de temps, nécessitant une minimisation sur les valeurs $O(r^2)$. Ainsi, le temps requis par SP est $O(nr^4)$.

3.3.7 Exemple d'application :

Dans cette partie, on utilisera l'exemple ci-dessous :

$$G = s(G_1, G_2), \text{ avec } G_1 = G_2$$

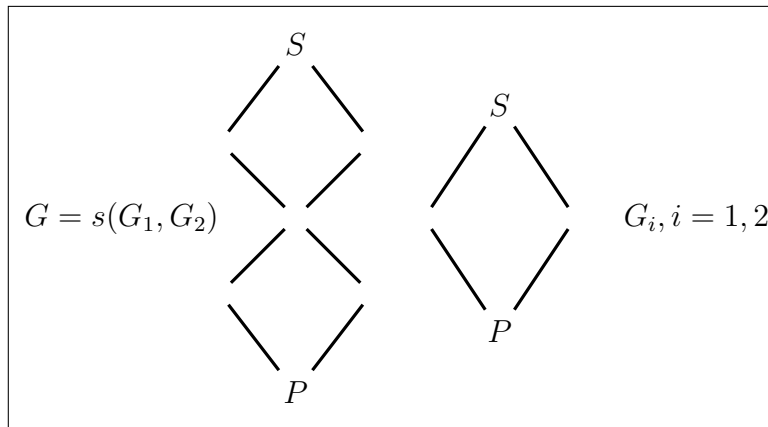


FIGURE 3.3 - $G = s(G_1, G_2)$

La condition de dominance pour notre graphe est telle que :

$$(dom_L, dom_R) = (1, 1)$$

Notons que le rayon de G est égal à : $r(G) = 2$ et que $G_1 = G_2$

$$\diamond \text{ Initialisons les coûts pour les deux graphes } G_1 \text{ et } G_2 : \begin{cases} 1, & L_{G_1} = L_{G_2} = 2; \\ 2, & R_{G_1} = R_{G_2} = 2; \\ 3, & B_{G_1} = B_{G_2} = 2; \\ 4, & N_{G_1} = N_{G_2} = 2. \end{cases}$$

◇ Ensuite, on va chercher la valeur du coût pour le graphe G en utilisant le tableau 3.2
 $G = s(G_1, G_2)$ tel que :

$$P_G = \{L_G, R_G, B_G, N_G\}$$

Premièrement $L_G(dom_L, dom_R)$:

$$L_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} B_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i & ..(1) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, j) + N_{G_2}(-i - 1, dom_R) & ..(2) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, -i - 1) + N_{G_2}(j, dom_R) & ..(3) \end{cases}$$

On a $r = 2 \Rightarrow r - 1 = 1$, ainsi que $dom_K = 1$, avec $K = \{L, R\}$. Et d'après le tableau 3.2 on obtient alors :

$$L_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq 2} B_{G_1}(1, i) + L_{G_2}(i, 1) - i & ..(1) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, j) + N_{G_2}(-i - 1, 1) & ..(2) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, -i - 1) + N_{G_2}(j, 1) & ..(3) \end{cases}$$

Traitons les équations une par une :

Equation (1)

$$\min_{1 \leq i \leq r=2} B_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i$$

deux itérations seulement se présentent : $\begin{cases} \text{Cas1} : i = 1, & B_{G_1}(1, 1) + L_{G_2}(1, 1) - 1 = 2 + 2 - 1 = 3; \\ \text{Cas2} : i = 2, & B_{G_1}(1, 2) + L_{G_2}(2, 1) - 1 = 3 + 2 - 2 = 3. \end{cases}$

Démonstration

- Cas 1 : $B_{G_1}(1, 1)$ représente le coût minimal d'un broadcast qui vaut 2 (d'après l'initialisation) provenant depuis le terminal gauche $t_L(G_1)$, et un autre depuis de terminal droit $t_R(G_1)$, qui fournissent tous les deux une sur-dominance de 1 de chaque borne. $L_{G_2}(1, 1)$ qui est une broadcast d'un coût minimal qui provient uniquement de la borne gauche du graphe G_2 appelée $t_L(G_2)$ qui vaut 2 fournissant ainsi une sur-dominance depuis les deux bornes terminales de G_2 .

- Cas 2 : $B_{G_1}(1, 2)$ est un coût minimal d'un broadcast provenant de chacune des bornes terminales du graphe G_1 , il vaut 2, et fournit une sur-dominance de 1 a gauche, une autre de 2 a droite. et le $L_{G_2}(2, 1)$ vaut également 2 et fournit deux sur-dominances de 2, 1 successivement sur les bornes gauche et droite.

Equation (2), (3)

Remarque 3.1 : ($dom_K \leq 0$) ne vérifie pas la condition de dominance (condition invalide) ce qui fait que :

les equations (2) et (3) vont tendre vers l'infini ∞ .

Démonstration : Pour l'équation (2) :

$$\min_{0 \leq i \leq j \leq r-1=1} L_{G_1}(dom_L, j) + N_{G_2}(-i-1, dom_R)$$

trois cas se présentent :

$$\begin{cases} Cas1 : i = j = 0, & L_{G_1}(1, 0) + N_{G_2}(-1, 1) = 2 + \infty \rightsquigarrow \infty ; \\ Cas2 : i = 0, j = 1, & L_{G_1}(1, 1) + N_{G_2}(-1, 1) = 2 + \infty \rightsquigarrow \infty ; \\ Cas3 : i = j = 1, & L_{G_1}(1, -2) + N_{G_2}(1, 1) = \infty + 2 \rightsquigarrow \infty . \end{cases}$$

- Pour le cas 1 : $i = j = 0$

Le coût $L_{G_1}(1, 0)$ représente le coût minimal d'un broadcast provenant de $t_L(G_1)$ qui vaut 2 d'après l'initialisation des coûts de G_i , tel qu'il fournit une sur-dominance de 1 depuis le terminal gauche $t_L(G_1)$, et ne fournit aucun broadcast du terminal droit), ensuite $N_{G_2}(-1, 1)$, d'après observation 1.3.3.1 lui ne vérifie pas la condition de dominance (car $dom_L < 0$). Ce qui fait tendre l'équation vers l'infini.

• Pareil pour les Cas 1 et Cas 3, les coûts minimaux $N_{G_2}(-1, 1)$ et $L_{G_1}(1, -2)$ respectivement tendent vers l'infini car ils ne vérifient pas la condition de dominance, ainsi les deux equations tendent également vers ∞ .

Pour l'équation (3) : Démonstration analogue que l'équation (2).

Conclusion 3.1 $L_G(dom_L, dom_R) = 3 = \min \begin{cases} \min\{3, 3\} \\ \infty; \\ \infty. \end{cases}$

Deuxièmement $R_G(dom_L, dom_R)$:

$$R_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} R_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i & ..(1) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, j) + N_{G_2}(-i-1, dom_R) & ..(2) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, -i-1) + N_{G_2}(j, dom_R) & ..(3) \end{cases}$$

Ce qui nous donne en remplaçant $r = 2 \Rightarrow r - 1 = 1$ et $dom_k = 1$:

$$R_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq 2} R_{G_1}(1, i) + B_{G_2}(i, 1) - i & ..(1) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, j) + N_{G_2}(-i-1, 1) & ..(2) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, -i-1) + N_{G_2}(j, 1) & ..(3) \end{cases}$$

Equation (1)

$$\min_{1 \leq i \leq r=2} R_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i$$

on a deux cas : $\begin{cases} Cas1 : i = 1, & R_{G_1}(1, 1) + B_{G_2}(1, 1) - 1 = 2 + 2 - 1 = 3; \\ Cas2 : i = 2, & R_{G_1}(1, 2) + B_{G_2}(2, 1) - 2 = 2 + 3 - 2 = 3. \end{cases}$

Démonstration

• Pour le Cas 1 : $R_{G_1}(1, 1)$ représente le coût minimal d'un broadcast qui vaut 2 (d'après l'initialisation) provenant depuis le terminal gauche $t_R(G_1)$, fournissant ainsi une sur-dominance de 1 sur $t_L(G_1)$ et sur $t_R(G_1)$. $B_{G_2}(1, 1)$ est un broadcast qui vaut également 2 dont les valeurs des sur-dominances qu'il fournit sont exactement les mêmes que le broadcast précédent sur les deux bornes.

• Pour le Cas 2 : Pareil pour $R_{G_1}(1, 2)$ et $B_{G_2}(2, 1)$, sauf qu'eux, fournissent deux sur-dominances de 1, 2 et 2, 1 respectivement.

Equation (2), (3) Elles tendent vers l'infini.

Remarque 3.2 : même argument que dans le premier cas de $L_G(dom_L, dom_R)$.

$$\text{Conclusion 3.2 } R_G(dom_L, dom_R) = 3 = \min \begin{cases} \min\{3, 3\} \\ \infty; \\ \infty. \end{cases}$$

Troisièmement $B_G(dom_L, dom_R)$:

$$B_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} B_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i & ..(1) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, j) + R_{G_2}(-i - 1, dom_R) & ..(2) \\ \min_{0 \leq i \leq j \leq r-1} L_{G_1}(dom_L, -i - 1) + R_{G_2}(j, dom_R) & ..(3) \end{cases}$$

Ce qui nous donne en remplaçant $r = 2$ et $dom_k = 1$:

$$B_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r=2} B_{G_1}(1, i) + B_{G_2}(i, 1) - i & ..(1) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, j) + R_{G_2}(-i - 1, 1) & ..(2) \\ \min_{0 \leq i \leq j \leq 1} L_{G_1}(1, -i - 1) + R_{G_2}(j, 1) & ..(3) \end{cases}$$

Equation (1)

$$\min_{1 \leq i \leq r=2} B_{G_1}(dom_L, i) + B_{G_2}(i, dom_R) - i$$

on a deux cas :
$$\begin{cases} Cas1 : i = 1, & B_{G_1}(1, 1) + B_{G_2}(1, 1) - 1 = 2 + 2 - 1 = 3; \\ Cas2 : i = 2, & B_{G_1}(1, 2) + B_{G_2}(2, 1) - 2 = 3 + 3 - 2 = 4. \end{cases}$$

Equation (2), (3)elles tendent vers l'infini.

Remarque 3.3 : même argument que dans le premier cas de $L_G(dom_L, dom_R$

Conclusion 3.3 $B_G(dom_L, dom_R) = 3 = \min \begin{cases} \min\{3, 4\} \\ \infty; \\ \infty. \end{cases}$

Quatrièmement $N_G(dom_L, dom_R)$:

$$N_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq r} R_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i & ..(1) \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, j) + N_{G_2}(-i - 1, dom_R) & ..(2) \\ \min_{0 \leq i \leq j \leq r-1} N_{G_1}(dom_L, -i - 1) + N_{G_2}(j, dom_R) & ..(3) \end{cases}$$

on a $r = 2 \Rightarrow r - 1 = 1$, ainsi que $dom_K = 1$, avec $K = \{L, R\}$. Et d'après le tableau 3.2 on obtient alors :

$$N_G(dom_L, dom_R) = \min \begin{cases} \min_{1 \leq i \leq 2} R_{G_1}(1, i) + L_{G_2}(i, 1) - i & ..(1) \\ \min_{0 \leq i \leq j \leq 1} N_{G_1}(1, j) + N_{G_2}(-i - 1, 1) & ..(2) \\ \min_{0 \leq i \leq j \leq 1} N_{G_1}(1, -i - 1) + N_{G_2}(j, 1) & ..(3) \end{cases}$$

Equation (1)

$$\min_{1 \leq i \leq r=2} R_{G_1}(dom_L, i) + L_{G_2}(i, dom_R) - i$$

deux itérations seulement se présentent :
$$\begin{cases} Cas1, & R_{G_1}(1, 1) + L_{G_2}(1, 1) - 1 = 2 + 2 - 1 = 3; \\ Cas2, & R_{G_1}(1, 2) + L_{G_2}(2, 1) - 2 = 2 + 2 - 2 = 2. \end{cases}$$

Démonstration

- Le Cas 1 : $R_{G_1}(1, 1)$ représente le coût minimal d'un broadcast qui vaut 2 (d'après l'initialisation) provenant depuis le terminal droit $t_R(G_1)$, fournit une sur-dominance de 1 de chaque borne. $L_{G_2}(1, 1)$ qui est une broadcast d'un coût minimal qui provient uniquement de la borne gauche du graphe G_2 appelée $t_L(G_2)$ qui vaut 2 fournissant ainsi une sur-dominance depuis les deux bornes terminales de G_2 .

• Et pour le Cas 2 : $R_{G_1}(1, 2)$ est un coût minimal d'un broadcast provenant de la borne droite du graphe G_1 , il vaut 2, et fournit une sur-dominance de 1 à gauche, une autre de 2 à droite, et le $L_{G_2}(2, 1)$ vaut également 2 et fournit deux sur-dominances de 2, 1 successivement sur les bornes gauche et droite.

Equation (2), (3)

Remarque 3.4 : ($dom_K \leq 0$) ne vérifie pas la condition de dominance ce qui fait que : les equations (2) et (3) vont tendre vers l'infini ∞ .

Remarque 3.5 : même arguments que précédemment.

Conclusion 3.4 $N_G(dom_L, dom_R) = 2 = \min \begin{cases} \min\{3, 2\} \\ \infty; \\ \infty. \end{cases}$

Conclusion générale

$$P_G(dom_L, dom_R) = \min\{L_G(dom_L, dom_R), R_G(dom_L, dom_R), B_G(dom_L, dom_R), N_G(dom_L, dom_R)\} \\ = \min\{3, 3, 4, 2\} = 2$$

donc

$$P_G(dom_L, dom_R) = N_G(dom_L, dom_R) = 2$$

c'est-à-dire :

$$\gamma_b(G) = N_G(dom_L, dom_R) = 2, dom_K = 1, k = 1, 2$$

Remarque 3.6 :

L'Algorithme de Broadcast domination dans les Graphes Series-Parallèle **SPBD** étudié dans ce chapitre nécessite beaucoup de temps pour la programmation vu, la complexité des calculs mentionnés dans les tableaux 2 et 3. Ceci est une bonne perspective de ce travail.

Chapitre 4

Etude de la chaîne P_n et Implementation

4.1 Introduction

Dans ce chapitre, nous allons étudier le Broadcast dominant efficace pour la chaîne P_n , et sa relation avec la domination, ensuite nous cherchons un broadcast efficace qui a le moins de sommets pour les deux cas de la chaîne P_n : pair et impair. Nous terminons ainsi notre chapitre par une implementation sur le logiciel **C**, en donnant quelques résultats finaux.

4.2 Broadcast efficace dans les Chaînes P_n

4.2.1 Invariants de broadcast domination dans la classe chaîne

Théorème 4.1 : [4] Pour tout entier $n > 2$, $rad(P_n) = \lfloor \frac{n}{2} \rfloor$

Proposition 4.1 : Bornes inférieures du nombre de broadcast domination

Dans [11], Erwin détermina une borne inférieure pour le nombre de broadcast domination d'un graphe arbitraire qui généralise l'inégalité $\gamma_b(G) \geq \lceil \frac{diam(G)+1}{3} \rceil$ connue pour le nombre de domination.

Théorème 4.2 : [11] Pour tout graphe G , on a

$$\gamma_b(G) \geq \lceil \frac{diam(G) + 1}{3} \rceil$$

Corollaire 4.1 : [11] Pour tout entier $n \geq 2$,

$$\gamma_b(P_n) = \gamma(P_n) = \lceil \frac{n}{3} \rceil$$

Remarque 4.1 :

$$\gamma_G(P_n) = \lceil \frac{\text{diam}_G + 1}{3} \rceil = \lceil \frac{n - 1 + 1}{3} \rceil = \lceil \frac{n}{3} \rceil$$

Théorème 4.3 : Pour tout $n \geq 2$,

$$\text{diam}(P_n) = n - 1$$

4.2.2 Broadcast dominant efficace d'une chaîne P_n

4.2.2.1 - Broadcast dominant efficace booléen

Soit P_n une chaîne d'ordre $n \geq 2$, définie par $P_n = (v_1, v_2, \dots, v_n)$. Et soit f une fonction de broadcast dominant efficace pour P_n définie telle que : $f : V \longrightarrow \{0, 1, \dots, \text{Diam}P_n\}$. Dans cette section, nous nous intéressons au broadcast booléen, autrement dit, la fonction broadcast prend ses valeurs dans $\{0, 1\}$ et ceci dans le but d'établir une relation entre le broadcast efficace d'une chaîne et son nombre de domination. Il est à noter qu'en généralement, ces deux invariants ne sont pas liés dans un graphe quelconque, même si on dispose d'algorithmes efficaces pour leur détermination.

Théorème 4.4 : Un Broadcast dominant f est dit efficace si : $B(v, f(v)), B(w, f(w)) = \phi$. C'est-à-dire qu'il n'y a pas d'intersection entre les deux boules de centres v, w respectivement.

Théorème 4.5 : (Dunbar et al. [3]) Il existe pour tout graphe G , un Broadcast dominant optimal efficace.

Algorithme :

L'algorithme ci-dessous recherche un broadcast dominant efficace booléen $f : V \longrightarrow \{0, 1\}$, dans le but de trouver le lien avec la domination. Après, nous allons nous intéresser à la cardinalité minimale d'un ensemble broadcast dominant avec f efficace, c-à-d $\min |V_f|$.

- **Cas 1 :** $n \equiv 0[3]$

```

For  $i = 1$  to  $n$  Do
     $K := 0$ ;
While  $i \leq n$  Do
    If  $i = 2$ 
         $f(v_i) == 1$ ;
         $K = K + 3$ ;
    Else
         $f(v_i) == 0$ ;

```

```
EndIf;  
EndWhile;  
EndFor;
```

- Cas 2 : $n \equiv 1[3]$

```
For  $i = 1$  to  $n$  Do  
     $K := 0$ ;  
    While  $i \leq n$  Do  
        If  $i = 1$   
             $f(v_i) == 1$ ;  
             $K = K + 3$ ;  
        Else  
             $f(v_i) == 0$ ;  
        EndIf;  
    EndWhile;  
EndFor;
```

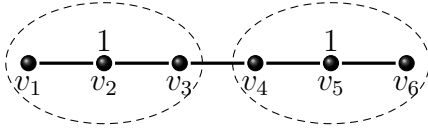
- Cas 3 : $n \equiv 2[3]$

```
For  $i = 1$  to  $n$  Do  
     $K := 0$ ;  
    While  $i \leq n$  Do  
        If  $i = 1$   
             $f(v_i) == 1$ ;  
             $K = K + 3$ ;  
        Else  
             $f(v_i) == 0$ ;  
        EndIf;  
    EndWhile;  
EndFor;
```

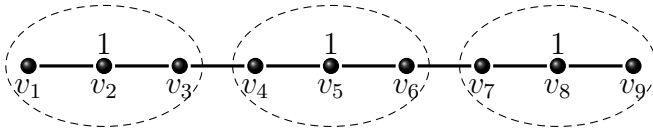
Remarque 4.2 : le cas 3, peut être fait autrement, en commençant par le deuxième sommet au lieu du premier.

Exemples 4.1 :

Premier cas :

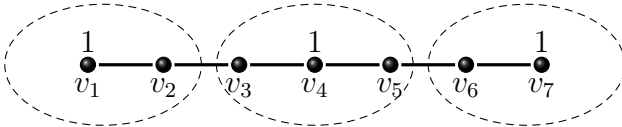


$$P_6 : 6 \equiv 0[3] \Rightarrow \gamma_b(P_6) = \lceil \frac{6}{3} \rceil = 2$$

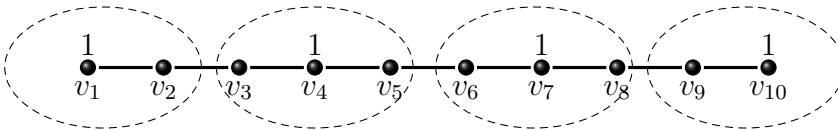


$$P_9 : 9 \equiv 0[3] \Rightarrow \gamma_b(P_9) = \lceil \frac{9}{3} \rceil = 3$$

Deuxième cas :

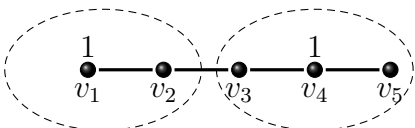


$$P_7 : 7 \equiv 1[3] \Rightarrow \gamma_b(P_7) = \lceil \frac{7}{3} \rceil = 3$$



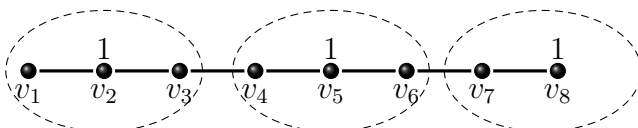
$$P_{10} : 10 \equiv 1[3] \Rightarrow \gamma_b(P_{10}) = \lceil \frac{10}{3} \rceil = 4$$

Troisième cas (méthode 1) :



$$P_5 : 5 \equiv 2[3] \Rightarrow \gamma_b(P_5) = \lceil \frac{5}{3} \rceil = 2$$

Troisième cas (méthode 2)



$$P_8 : 8 \equiv 2[3] \Rightarrow \gamma_b(P_8) = \lceil \frac{8}{3} \rceil = 3$$

En conclusion, on voit bien, que les procédures décrites dans cette section, génèrent bien des broadcasts efficaces booléens dans les chaînes P_n avec une complexité linéaire d'ordre $o(n + m)$, ce qui nous donne un ensemble dominant optimal, donc le nombre de domination $\gamma(P_n)$. En effet, comme le broadcast booléen est efficace, il suffit alors de poser $S = V_f$ donc chaque boule de rayon 1, de centre $f(v_i) = 1$ correspond au voisinage $N[v_i]$ et ces voisinages sont deux à deux disjoints car le broadcast est efficace. De plus, le broadcast booléen est dominant efficace donc chaque sommet v de V appartient à exactement une des boules, autrement dit, il existe un unique sommet w dans

S tel que $v \in V[w]$ ce qui fait la cardinalité minimum de S et assure la dominance de S car il est evident dans ce cas que $N[S] = V$.

4.2.2.2 - Cardinalité minimale de V_f :

Position du problème : Soit f un broadcast dominant efficace de la chaîne P_n , et soit l'ensemble des sommets-broadcast dominants :

$$V_f = \{v \in V, f(v) > 0\}$$

Question : Quel est le broadcast dominant efficace $\gamma_b(P_n)$ qui admet le minimum de sommets-broadcast dominant ? Autrement dit ; on cherche :

$$\gamma_b(P_n), \text{ avec } f \text{ est telle que : } \min |V_f|$$

1. Cas P_n impaire :

Proposition 4.2 : Soient $f \rightarrow \{0, 1, \dots, diam(P_n)\}$ un broadcast dominant et, $P_n = \{v_1, v_2, \dots, v_i, \dots, v_n\}$ une chaîne impaire.

On définit la médiane $Med(P_n)$ comme suit :

$$Med(P_n) = v_i = v_{\lceil \frac{n}{2} \rceil}$$

et on affecte le coût de broadcast dominant pour le sommet qui représente la position de la médiane de notre chaîne tel que ce broadcast soit égal à :

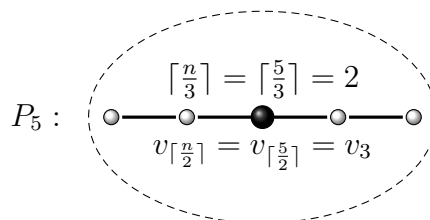
$$f(v_i) = \lceil \frac{n}{3} \rceil \text{ et, } f(v_j) = 0, \forall i \neq j$$

Alors tous les sommets de P_n devraient être dominés.

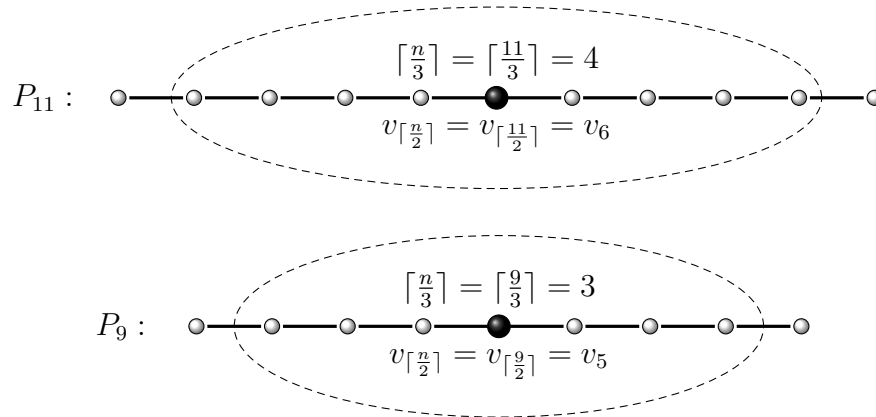
Exemple 4.2 : Soit $P_5 = \{v_1, v_2, v_3, v_4, v_5\}$, la médiane de cette chaîne $Med(P_n)$ est égale à :

$$Med(P_5) = v_i = v_{\lceil \frac{5}{2} \rceil} = v_{\lceil \frac{5}{2} \rceil} = v_3 \implies v_i = v_3$$

et le coût du broadcast dominant est égal à : $f(v_i) = f(v_3) = \lceil \frac{5}{3} \rceil = \lceil \frac{5}{3} \rceil, \implies f(v_3) = 2$.



Exemple 4.3 : Prenons deux autres chaînes P_{11} et P_9 que voici :



Contradiction! le Broadcast de la chaîne P_{11} et celui de P_9 ci-dessus est efficace mais n'est pas dominant.

conclusion 4.2 : Les chaînes P_9 et P_{11} , ne peuvent pas admettre de Broadcast efficace Dominant de cardinalité $|V_f| = 1$.

Conjecture :

On peut conjecturer que toutes les chaînes impaires avec, $n \geq 9$, ne vont pas admettre de Broadcast dominant efficace de cardinalité $|V_f| = 1$, car $\lceil \frac{n}{3} \rceil \neq \lfloor \frac{n}{2} \rfloor$.

De ce fait, il est très intéressant de pencher sur la démonstration de ce résultat.

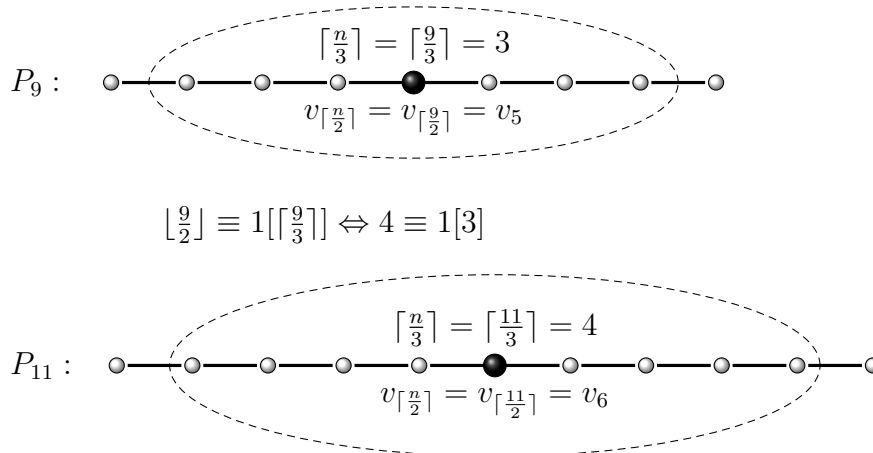
Démonstration :

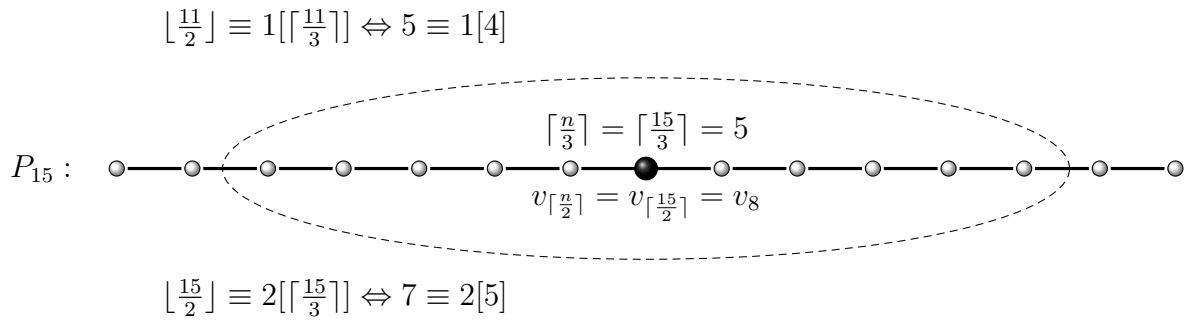
Nous cherchons les cas où le broadcast dominant efficace avec $\min|V_f| = 1$ existe?

Soit i tel que : $\lfloor \frac{n}{2} \rfloor \equiv i \lceil \frac{n}{3} \rceil$, pour toute chaîne $P_n, \forall n$.

Si $i \neq 0, f$ vérifiant cette cardinalité minimale de l'ensemble broadcast dominant n'existe pas.

Exemple 4.4 :





Remarque 4.3 : Uniquement les chaînes P_n d'ordre n impair avec, $n < 9$ qui vérifient :

$\lfloor \frac{n}{2} \rfloor \equiv 0[\lceil \frac{n}{3} \rceil]$, autrement dit $\lfloor \frac{n}{2} \rfloor = \lceil \frac{n}{3} \rceil$, peuvent admettre de broadcast dominant efficace non-booléen de cardinalité 1, $f : V \rightarrow \{0, 1, \dots, diam(P_n)\}$. Ce qui nous conduit à la conclusion suivante :

Conclusion 4.3 : Pour toute chaîne P_n d'ordre n tel que, $1 < n < 9$, la cardinalité minimale d'un ensemble Broadcast dominant V_f , pour tout f broadcast dominant efficace est toujours égale à 1.

$$min|V_f| = 1$$

2. Cas P_n paire :

Proposition 4.3 : Soit $P_n = \{v_1, v_2, \dots, v_i, \dots, v_n\}$ une chaîne paire d'ordre n , alors $|v_f P_n| > 1$.

Contrairement au premier cas (la chaîne impaire), la chaîne d'ordre n pair n'admet pas de médiane $Med(P_n) = v_i$, c'est pour cela que nous allons définir l'interval-médian comme suit :

$$IMed(P_n) = [v_i, v_{i+1}]$$

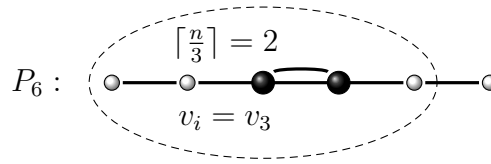
Avec $[v_i, v_{i+1}] = [v_{\frac{n}{2}}, v_{\frac{n}{2}+1}]$

Ensuite, on affecte le coût du broadcast dominant soit pour v_i ou bien pour v_{i+1} :

- Si $f(v_i) := \lceil \frac{n}{3} \rceil \implies v_n$ ne sera pas couvert de ce broadcast.
- Si $f(v_{i+1}) := \lceil \frac{n}{3} \rceil \implies v_1$ ne sera pas couvert donc, $|V_f P_n| > 1$.

Exemple 4.5 : Soit la chaîne d'ordre n pair, P_6 pour laquelle nous souhaitons trouver le $min|V_f|$, telle que, son interval-médian est défini par :

$$IMED(P_6) = [v_i, v_{i+1}] = [v_{\frac{n}{2}}, v_{\frac{n}{2}+1}] = [v_3, v_4]$$

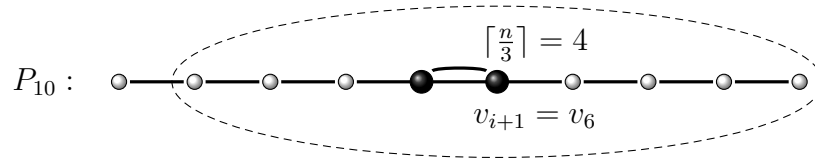


Effectivement, lorsqu'on a posé $f(v_i) := \lceil \frac{n}{3} \rceil$, le dernier sommet v_6 n'est pas broadcast-dominé par f .

- Si $f(v_{i+1}) := \lceil \frac{n}{3} \rceil \implies v_1$ dans ce cas ne sera pas dominé par f .

Exemple 4.6 : Prenons cette fois-ci la chaîne d'ordre n pair, P_{10} pour laquelle nous souhaitons trouver le $\min|V_f|$. On définit son interval-médian est défini par :

$$IMED(P_{10}) = [v_i, v_{i+1}] = [v_{\frac{n}{2}}, v_{\frac{n}{2}+1}] = [v_5, v_6].$$



Effectivement, $f(v_{i+1}) := \lceil \frac{n}{3} \rceil$, le premier sommet v_1 n'est pas broadcast-dominé par f .

Conclusion 4.4 : Les deux exemples ci-dessus, prouvent que, $\forall n \geq 2$, la cardinalité minimale d'un ensemble broadcast dominant V_f , pour f broadcast dominant efficace est toujours supérieur strictement à 1.

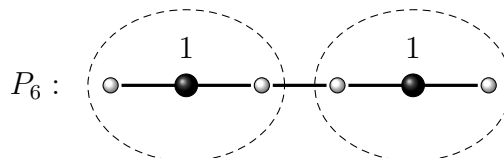
$$\min|V_f| > 1$$

Résultat 4.1 : Pour le cas d'une chaîne P_n d'ordre n pair,

$$\min|V_f| = \gamma_b(P_n),$$

Exemple 4.6 : Soit la chaîne P_6 qui est paire, d'après l'illustration ci-dessous, le résultat est vérifié, en effet

$$\min|V_f| = 2 = \lceil \frac{n}{3} \rceil = 2 \implies \min|V_f| = \gamma_b(P_n)$$



Conclusion 4.5 : A partir du résultat 4.1, on peut conclure que pour toute chaîne P_n d'ordre n pair on a :

$$\gamma(P_n) = \gamma_b(P_n)$$

4.3 Implémentation

4.3.1 Introduction

Dans cette partie, nous allons implémenter des résultats obtenus dans le titre 4.2 pour la chaîne P_n , plus précisément sur le broadcast dominant efficace ainsi que la cardinalité minimale d'un ensemble de sommets broadcast dominants, avec f efficace.

4.3.2 Univers de travail : Langage C

C est un langage de programmation impératif généraliste, inventé au début des années 70. C'est l'un des langages les plus intéressants et utilisés encore de nos jours. En effet, de nombreux langages plus modernes reprennent une syntaxe similaire que lui, ainsi qu'une partie de sa logique.

4.3.3 Exemples d'application

Il s'agit d'implémenter sur le C, en premier lieu les résultats du titre 4.2.1 concernant les chaînes qui peuvent bien admettre de broadcast dominant efficace booléen ($f(v_i) \in \{0,1\}$) entraînant leur relation avec le nombre de domination, ensuite ceux de la partie 4.2.2, traitant la cardinalité minimale de l'ensemble V_f des sommets broadcast dominants pour la chaîne P_n avec n pair ou impair, et f efficace en deuxième lieu.

Identification des problèmes

Problème 1 : Soit $f : V \rightarrow \{0,1\}$ un broadcast dominant efficace booléen, et $P_n = (v_1, v_2, \dots, v_n)$ une chaîne d'ordre n .

◇ Si $n \equiv 0[3]$, (autrement dit, n multiplicateur de 3) alors P_n peut admettre de Broadcast dominant efficace booléen f .

◇ Si $n \equiv 1[3]$ alors P_n peut bien en admettre également.

◇ Sinon ($n \equiv 2[3]$), alors P_n admet de broadcast dominant efficace f booléen.

Et le nombre de ce Broadcast dans tous les cas vaut : $\gamma_b(P_n) = \lceil \frac{n}{3} \rceil$

Il s'agit d'implémenter ce programme sous le langage C, en donnant des exemples explicatifs.

Problème 2 : Soit $f : V \rightarrow \{0, 1, \dots, \text{diam}(G)\}$ un broadcast dominant efficace, et $P_n = (v_1, v_2, \dots, v_n)$ une chaîne d'ordre n pair ou impair.

Nous cherchons à trouver la cardinalité minimale d'un ensemble de sommets broadcast dominants V_f tel que ;

◇ Si n est pair, alors $\min|V_f| > 1$.

◇ Sinon, alors on obtient deux cas :

- Si $n < 9$, alors on a $\min|V_f| = 1$.
- Si $n \geq 9$, alors la chaîne P_n n'admet pas de Broadcast dominant efficace.

4.3.4 Implémentation

Les programmes que nous allons présenter ci-dessous sont une implémentation des deux problèmes déjà identifiés sur le langage **C**.

Problème 1 :

```

1  #include <stdio.h>
2  #include "list.h"
3
4  int main(void)
5  {
6
7      //Définition d'une liste:
8      List mylist = new_list();
9
10     //Remplissage de la liste:
11     int i;
12     printf("Donnez la longueur de la chaîne: \n");
13     scanf("%d", &i);
14
15     for (int x = 1; x <= i; x++)
16     {
17         mylist = push_list(mylist, x);
18     }
19
20     //Affichage de la liste:
21     printf("P%d =", i);
22     print_list(mylist);
23
24     //Nombre d'element de la liste :
25     int n = list_length(mylist);
26
27     //La vérification de Broadcast :
28     if (n % 3 == 0)
29     {
30         printf( "le reste  de la division de %d sur 3 est egale a 0 \n", n);
31         printf("P%d peut admettre de Broadcast efficace booleen \n", i );
32         printf("le nombre de Broadcast dominant efficace de P%d vaut: %d", i, n / 3);
33     }
34
35     else if

```

FIGURE 4.1 – programme du problème 1

```

33     }
34
35     else if
36
37         (n % 3 == 1)
38     {
39
40         printf( "le reste  de la division de %d sur 3 est egal a 1\n", n);
41         printf("P%d peut admettre de Broadcast efficace booleen \n", i );
42         printf("le nombre de Broadcast dominant efficace de P%d vaut: %d", i, n / 3 + 1);
43     }
44
45     else
46
47     {
48         printf("le reste de la division de %d sur 3 egal a 2 \n", n);
49         printf("P%d peut donc admettre de broadcast dominant efficace booleen \n", i);
50         printf("le nombre de Broadcast dominant efficace de P%d vaut: %d", i, n / 3 + 1);
51     }
52 }
53
54
55
56
57 // Effacement de la liste obligatoire dans le C :
58 mylist = clear_list(mylist);
59
60 return 0;
61 }

```

FIGURE 4.2 – programme du problème 1 (suite)

```

CA Invite de commandes
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
6
P6 = {V[1],V[2],V[3],V[4],V[5],V[6],}
le reste de la division de 6 sur 3 est egale a 0
P6 peut admettre de Broadcast efficace booleen
le nombre de Broadcast dominant efficace de P6 vaut: 2
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
9
P9 = {V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],V[9],}
le reste de la division de 9 sur 3 est egale a 0
P9 peut admettre de Broadcast efficace booleen
le nombre de Broadcast dominant efficace de P9 vaut: 3
C:\Users\merie\Desktop\joseph>

```

FIGURE 4.3 – Résultats de P_9 et P_6

```

CA Invite de commandes
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
7
P7 = {V[1],V[2],V[3],V[4],V[5],V[6],V[7],}
le reste de la division de 7 sur 3 est egal a 1
P7 peut admettre de Broadcast efficace booleen
le nombre de Broadcast dominant efficace de P7 vaut: 3
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
10
P10 = {V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],V[9],V[10],}
le reste de la division de 10 sur 3 est egal a 1
P10 peut admettre de Broadcast efficace booleen
le nombre de Broadcast dominant efficace de P10 vaut: 4
C:\Users\merie\Desktop\joseph>

```

FIGURE 4.4 – Résultats de P_7 et P_{10}

```

CA Invite de commandes
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
8
P8 = {V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],}
le reste de la division de 8 sur 3 egal a 2
P8 peut donc admettre de broadcast dominant efficace booleen
le nombre de Broadcast dominant efficace de P8 vaut: 3
C:\Users\merie\Desktop\joseph>programme
Donnez la longueur de la chaîne:
17
P17 = {V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],V[9],V[10],V[11],V[12],V[13],V[14],V[15],V[16],V[17],}
le reste de la division de 17 sur 3 egal a 2
P17 peut donc admettre de broadcast dominant efficace booleen
le nombre de Broadcast dominant efficace de P17 vaut: 6
C:\Users\merie\Desktop\joseph>

```

FIGURE 4.5 – Résultats de P_8 et P_{17}

Problème 2 :

```
Welcome  C list.c  C list.h  C listes.c  X
C listes.c > main(void)
1  #include <stdio.h>
2  #include "list.h"
3
4  int main(void)
5
6
7  //Définition d'une Liste:
8  List mylist = new_list();
9
10 //Remplissage de la liste:
11 int i;
12 printf("Donnez le nombre de caractere de la chaine : \n");
13 scanf("%d", &i);
14
15 for (int x = 1; x <= i; x++)
16 {
17     mylist = push_list(mylist, x);
18 }
19
20 //Affichage de la liste:
21 printf("P%d =", i);
22 print_list(mylist);
23
24 //Nombre d'element de la liste :
25 int n = list_length(mylist);
26
27
28
29 //paire et impaire
30 if (n % 2 == 0)
31 {
32     printf("%d est pair \n", n);
33 }
```

FIGURE 4.6 – Programme associé au problème 2

```
28
29 //paire et impaire
30 if (n % 2 == 0)
31 {
32     printf("%d est pair \n", n);
33     printf(" La cardinalite minimale de l'ensemble des sommets broadcast dominants efficace est strictement superieur a 1", n);
34 }
35 else
36 {
37     if (n < 9)
38     {
39         printf(" %d est impair et inferieur a 9", n);
40         printf(" donc la cardinalite minimale de l'ensemble des sommets broadcast dominants efficace est egale exactement a 1 \n", n);
41     }
42     else
43     {
44         printf(" %d est impair et surperieur ou egal a 9", n);
45         printf(" donc la chaîne associee ne peut pas admettre de Broadcast dominant efficace \n", n);
46     }
47 }
48
49 // Effacement de la liste obligatoir dans le C :
50 mylist = clear_list(mylist);
51
```

FIGURE 4.7 – Programme associé au problème 2 (suite)

```

C:\ Invite de commandes
Microsoft Windows [version 10.0.18363.1556]
(c) 2019 Microsoft Corporation. Tous droits réservés.

C:\Users\merie>cd.\desktop\prob2

C:\Users\merie\Desktop\prob2>gcc *.c -o programme

C:\Users\merie\Desktop\prob2>programme
Donnez le nombre de caraceter de la chaine :
10
P10 ={V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],V[9],V[10],}
10 est pair
La cardinalite minimale de l'ensemble des sommets broadcast dominants efficaces est strictement superieur a 1
C:\Users\merie\Desktop\prob2>

```

FIGURE 4.8 – Résultats pour P_{10}

```

C:\Users\merie\Desktop\prob2>programme
Donnez le nombre de caraceter de la chaine :
7
P7 ={V[1],V[2],V[3],V[4],V[5],V[6],V[7],}
7 est impair et inferieur a 9 donc la cardinalite minimale de l'ensemble des sommets broadcast dominants efficace est egale exactement a 1
C:\Users\merie\Desktop\prob2>

```

FIGURE 4.9 – Résultats pour P_7

```

C:\Users\merie\Desktop\prob2>programme
Donnez le nombre de caraceter de la chaine :
11
P11 ={V[1],V[2],V[3],V[4],V[5],V[6],V[7],V[8],V[9],V[10],V[11],}
11 est impair et surperieur ou egal a 9 donc la chaine associee ne peut pas admettre de Broadcast dominant efficace
C:\Users\merie\Desktop\prob2>

```

FIGURE 4.10 – Résultats pour P_{11}

Conclusion 4.6 : Ainsi s’achève notre chapitre, les résultats obtenus sur le **C** montrent bien la fiabilité ainsi que la justesse du travail que nous avons fait.

Conclusion générale

Dans ce mémoire, nous avons, après avoir rappelé toutes les définitions de base de la Théorie des Graphes, Théorie de la Complexité, et traité le problème de la domination, le modèle de diffusion de base qui a ensuite été généralisé en broadcast domination, autrement dit le problème maître de notre travail, nous avons tenté d'appliquer ce dernier sur deux classes de graphes spécifiques, les graphes série-parallèle et chaînes P_n .

Pour ce faire, le travail s'est déroulé comme suit :

D'abord, nous avons étudié un algorithme pour le broadcast domination optimal pour les graphes série-parallèle (SPBD) muni d'un exemple de la composition des graphes en série bien détaillé montrant le déroulement des calculs présentés par l'algorithme.

Ensuite, nous avons tenté l'étude du broadcast dominant efficace pour les chaînes P_n et ceci en deux sous-parties ; la première traitant la relation entre la domination et le broadcast dominant efficace booléen, pour laquelle nous avons présenté des algorithmes. La deuxième traitant quant à elle la cardinalité minimale d'un ensemble de sommets broadcast dominants, pour laquelle nous avons formulé une conjecture assez intéressante.

Ainsi, le mémoire est achevé, par une implementation des résultats obtenus ci-dessus sur le langage de programmation C pour les deux problèmes de broadcast dominant efficace booléen ($f(v_i) \in \{0, 1\}$), ainsi que la cardinalité minimale de l'ensemble V_f , f efficace.

Nous tenons à préciser que tous nos résultats sur les chaînes sont prouvés, et accompagnés d'exemples explicatifs permettant une meilleure et plus facile compréhension des résultats, tandis que ceux de la classe série-parallèle, nous avons expliqué les démonstrations les plus importantes, ainsi que l'algorithme.

Perspective :

1. Il serait très intéressant de programmer l'algorithme pour la classe série-parallèle.
2. Tenter de démontrer la conjecture.
3. Établir un lien entre la domination et broadcast dans le cas des graphes série-parallèle.

Bibliographie

- [1] C. BERGE, *Théorie des graphes et ses applications*. Dunod, Paris, 1967.
- [2] M.AOUANE, Mémoire de Magister en Mathématiques, *Approche algorithmique pour la résolution de quelques problèmes dans les posets*
- [3] J. R. S. BLAIR, P. HEGGERNES, S. HORTON, and F. MANNE, *Broadcast domination Algorithms for interval graphs, series-parallel graphs, and trees*, Congressus Numerantium, [8- 15], 2004.
- [4] M. R. GAREY, and D. S. JOHNSON, *Computers and Intractability*. Editions Freeman, 1979.
- [5] Gregory MOREL, *Thèse de doctorat Stabilité et coloration des graphes sans P5*, Université de Grenoble, 2006.
- [6] Frédéric MENNIER, *Introduction à la Recherche Opérationnelle*, Université de Paris.
- [7] Mathieu LIEDLOFF, Université Paul, verlaine Metz.
- [8] Alain BRETTO, Alain FAISANT, François HENNECART, *Eléments de théorie des graphes*.
- [9] P. HEGGERNES and D.LOKSHTANOV, Optimal broadcast domination of arbitrary graphs in polynomial time. Disc. Math, 306(24) : [1- 2], 2006.
- [10] B. SADI, *Théorie des graphes*, compléxité algorithmique. Institut d'informatique, Université de Tizi-Ouzou.
- [11] *Algorithmic Graph Theory and Perfect Graphs*, new edition (Annals of discrete Mathematics 57), Martin Charles Golumbic (Eds.)- Butterworth heineman (2004).
- [12] David EPPSTEIN, *Parallel Recognition of Series-parralel graphs*, Departement of Information and Computer Science. University of California, Irvine, CA 92717. October 4, 1990.
- [13] J. E. DUNBAR, D. J. ERWIN, T. W. HAYNES, S. M. HEDETNIEMI, and S. T. HEDETNIEMI, *Broadcasts in graphs*, 2004. Submitted to Disc. Appl. Math.
- [14] P. HEGGERNES, Daniel. L, *Optimal broadcast domination in polynomial time*, Department of Informatics, University of Bergen, N-5020 Bergen, Norway.
- [15] D. J. ERWIN, *Dominating in Graphs*, Bulletin of the institute of combinatorics and applications, 42 (2004) 89-105.
- [16] S. BOUCHOUKA, I. BOUCHEMAKH, E. SOPENA, *Broadcast on Paths and Cycles*, January 30, 2020.

Résumé

Soit G un graphe et P_n une chaîne de longueur n , un ensemble dominant dans G est un sous-ensemble de sommets du graphe G , tel que tout sommet appartient soit à cet ensemble soit à un voisin de celui-ci.

Le problème du broadcast domination est une variante du problème de la domination. Un broadcast f est efficace si tout sommet est f -broadcast dominé exactement par un seul sommet. Dans ce travail, nous définissons et étudions le nombre du broadcast dominant efficace dans les graphes série-parallèle et chaînes P_n , et nous implémentons les résultats sur le langage C .

Mots clés : Domination, Broadcast domination, Broadcast efficace, Graphe série-parallèle, chaîne,