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

UNIVERSITE MOULOUD MAMMERI, 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

Thème

Sur le nombre de broadcast domination dans les graphes en bloc

Présenté par

AHCENE Samir

Devant le jury d'examen composé de :

B.SADI	M.C.A	UMMTO	Président
M.AOUANE	M.A.A	UMMTO	promoteur
F.RABIA	M.C.A	UMMTO	Examinatrice
K.KASDI	M.A.A	UMMTO	Examinateur

Soutenu le 15/ 09/ 2013

Remerciements

Avant tout, je remercie Allah le tout puissant de m'avoir donné la force, le courage et surtout la volonté nécessaire pour la réalisation de ce modeste travail.

Je voudrais aussi remercier mon promoteur, pour sa responsabilité au cours de ce mémoire et pour ses conseils qui m'ont aidé petit à petit à comprendre le problème. Ses orientations et ses remarques m'ont également permit à mieux contrôler mes travaux et à surmonter les difficultés.

Je remercie également Mr SADI pour l'honneur qu'il me fait en présidant le jury de ce mémoire. Mes remerciements s'adressent aussi à Mme RABIA et Mr KASDI pour l'intérêt qu'ils ont manifesté à ce mémoire en acceptant de le corriger.

Je ne peux oublier de remercier tous ceux qui ont de près ou de loin contribués à la réalisation de ce modeste travail, ainsi qu'à ma formation.

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Découlant directement de cette faculté qu'a l'homme à choisir le meilleur des outils qu'il possède pour sa fin, l'origine de l'optimisation remonte donc à très loin. On optimise presque dans toutes les sciences mais le noyau de l'optimisation peut être considéré comme étant la recherche opérationnelle. Celle-ci vise essentiellement la recherche des techniques rationnelles d'analyse et de résolution des problèmes rencontrés principalement dans l'activité socio-economique et industrielle et visant en général l'élaboration des décisions les plus efficaces pour l'obtention de meilleurs résultats. Pour de tels objectifs deux approches sont souvent utilisées pour tenter de résoudre ces problèmes, l'approche polyédrale et l'approche algorithmique. Tandis que la première approche est basée sur la recherche de l'enveloppe convexe, dite polytope, des contraintes du problème, la seconde vise essentiellement l'élaboration d'un système de symboles et de procédés de calcul, connu plus sous le nom d'Algorithme qui transforme les données du problème en sa solution. La résolution par l'une ou l'autre des deux approches suppose d'abord une modélisation du problème. Par cette dernière on sous entend un modèle mathématique facilement généralisable à tous les cas, sous lesquels peut se présenter le problème. La théorie des graphes est connue comme étant un puissant outil de modélisation et de résolution des problèmes rencontrés en optimisation. C'est dans celle-ci que ce situe notre problème, qui n'est autre que la recherche du nombre de broadcast domination dans les graphes en bloc.

Introduit en 2001 par D. J. ERWIN pour généraliser le problème de la domination en introduisant le concept de broadcast domination, qui est une fonction f définie sur l'ensemble des sommets d'un graphe connexe, non orienté et non pondéré à valeurs entières telle que $f(v) \leq$ excentricité de v, pour tout sommet v du graphe. Si cette fonction est telle que pour tout sommet v, $f(v) > 0$ ou il est à distance $\leq f(u)$ d'un autre sommet u du graphe tel que $f(u) > 0$, le broadcast est dominant.
Le broadcast dominant optimal est le broadcast dominant réalisant le minimum de la somme des valeurs de cette fonction sur l'ensemble des sommets du graphe. Ce minimum est dit nombre de broadcast domination et est noté $\gamma_b(G)$.

Ce mémoire est divisé en quatre chapitres qui sont comme suit :

Le premier chapitre de ce mémoire est consacré essentiellement aux définitions préliminaires des notions les plus utilisées dans ce dernier. Nous introduisons un certain nombre de notions fondamentales en théorie des graphes, en particulier les principaux concepts et résultats utilisés dans ce travail.

Dans le second chapitre on traite de ce qui est de l'approche algorithmique et des principaux concepts de la complexité algorithmique pour donner enfin certains éléments essentiels du problème de domination ainsi que celui du broadcast dominant optimal.

Nous abordons dans le troisième chapitre, le problème de recherche du broadcast dominant optimal dans la classe des graphes en bloc en exposant un algorithme linéaire en nombre d'arêtes et sommets ($O(n + m)$), pour la détermination du broadcast dominant optimal dans cette classe de graphes.

Le quatrième chapitre sera consacré à l'implémentation des résultats développés dans chapitre trois en utilisant le logiciel MATLAB et nous exécutant notre programme sur un exemple pratique dans les réseaux de communication en informatique.

Nous terminons notre travail par une petite conclusion générale.

Chapitre 1

Concepts de base

1.1 Introduction

Ce premier chapitre fournit au lecteur les bases en théorie des graphes, nécessaires à la bonne compréhension des notions abordées dans la suite de ce mémoire. Ce chapitre est consacré à la théorie des graphes, il reprend ainsi ce qu'on peut retrouver dans la majeure partie des ouvrages d'introduction à ce sujet. Nous nous sommes notamment efforcés à employer les notations et noms les plus usuels en français ; toutefois nous avons jugé utile d'adjoindre, pour chaque terme nouvellement introduit, sa traduction anglophone, mise en évidence par une emphase et entre parenthèses.

1.2 Théorie des graphes

L'histoire de la théorie des graphes débute peut-être avec les travaux d'Euler au XVIIIe siècle et trouve son origine dans l'étude de certains problèmes, tels que celui des ponts de Königsberg, les habitants de Königsberg se demandaient s'il était possible, en partant d'un quartier quelconque de la ville, de traverser tous les ponts sans passer deux fois par un même pont et de revenir à son point de départ, ou celui de la marche du cavalier sur l'échiquier ou le problème de coloriage de cartes. La théorie des graphes s'est alors développée dans diverses disciplines telles que la chimie, la biologie, les sciences sociales. . . etc. Depuis le début du XXe siècle, elle constitue une branche à part entière des mathématiques, grâce aux travaux de König, Menger, Cayley puis de Berge et d'Erdös [15]. De manière générale, un graphe permet de représenter la structure, les connexions d'un ensemble complexe en exprimant les relations de couple entre ses éléments : réseau de communication, réseaux routiers, interaction de diverses espèces animales, circuits électriques . . . etc. Les graphes constituent donc une méthode de modélisation et de résolution d'une grande variété de problèmes, par l'étude et la recherche de sous ensembles remarquables de sommets ou d'arêtes.

1.2.1 Concepts de base

Intuitivement un graphe (graph) est un ensemble de points ou sommets (vertices) dont certaines paires sont reliées, formant ainsi les extrémités (endpoints) d'une arête (edge). Plus formellement:

Définition 1.2.1. ***Un graphe non orienté G*** est un couple (V_G,E_G) où :
•V_G est un l'ensemble des sommets.
•$E_G \subseteq \{\{x,y\},x \in V_G,y \in V_G\}$ est l'ensemble des arêtes.

Lorsque le contexte ne laissera aucune place à l'ambiguïté quant au graphe auquel il sera fait référence, nous écrirons plus simplement V et E. Le nombre de sommets d'un graphe, ou son ordre (order), est égal donc à $|V|$, est plus généralement noté n, le nombre d'arêtes est quant à lui noté m et est égal à $\mid E \mid$.
Notons que la définition 1.2.1 laisse la possibilité pour une arête d'avoir ses deux extrémités identiques, une telle arête est appelée boucle (loop). De plus, pour de nombreuses applications, il peut être utile d'avoir plusieurs arêtes ayant les mêmes extrémités , on parle alors d'arêtes multiples (multiple edges). Un graphe ayant des arêtes multiples est un multigraphe (multigraph) , s'il contient également des boucles, on parle de pseudographe (pseudograph) (figure 1.1). Un graphe ne contenant ni boucles ni arêtes multiples est qualifié de simple (simple graph)

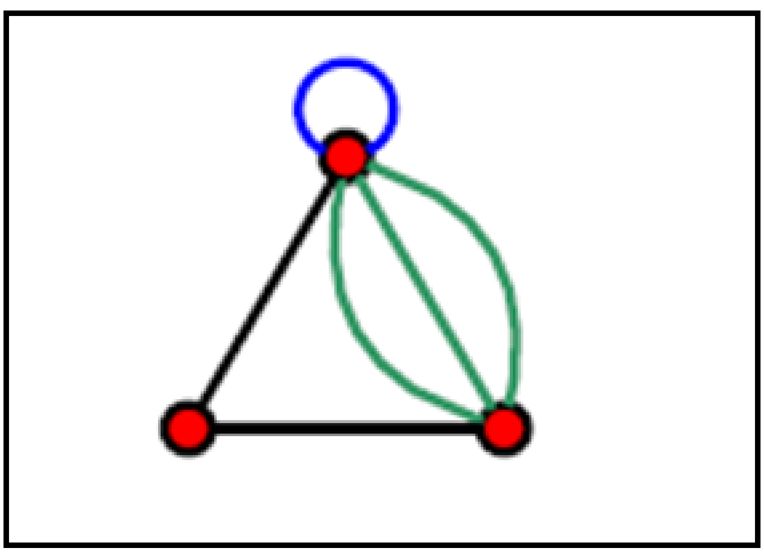


FIG. 1.1 – ***Un exemple de pseudographe, avec des arêtes multiples (en vert) et une Boucle (en bleu).***

Il est également courant de rencontrer des graphes pour lesquels on distingue l'extrémité initiale de l'extrémité terminale des arêtes, c'est-à dire que l'on ne considère plus des paires mais des couples de sommets ; on parle alors d'arcs (arcs) et non plus d'arêtes. Un tel graphe est qualifié d'orienté (directed graph ou digraph) (figure 1.2).

Dans la suite, sauf mention contraire explicite, tous les graphes que nous considérerons seront simples et non orientés.

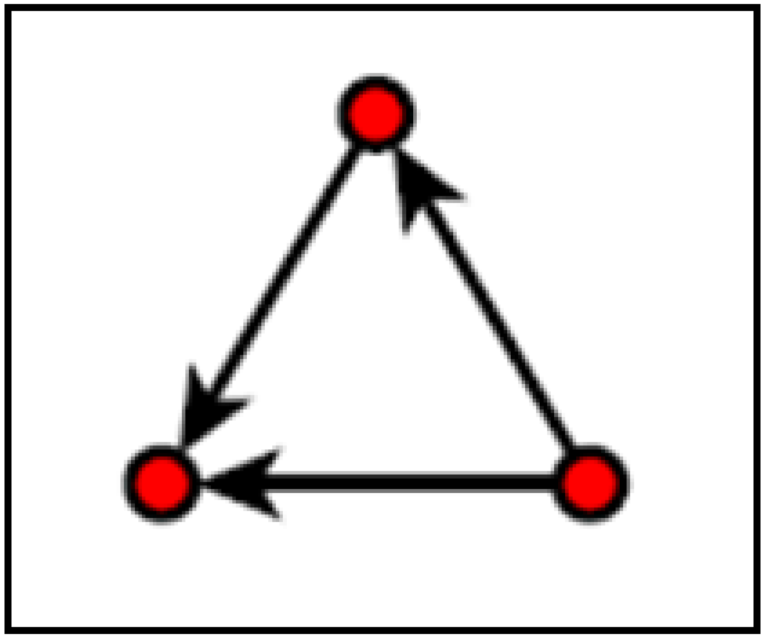


FIG. 1.2 – ***Un exemple de graphe orienté.***

Notons de plus, qu'il est courant d'attribuer des noms ou des numéros aux sommets et/ou aux arêtes d'un graphe. Un tel procédé est appelé étiquetage (labeling). Ainsi, nous parlerons souvent " du sommet u" ou " du sommet v " d'un graphe. Ceci nous autorise donc à parler de l'arête $\{u,v\}$pour l'évoquer, si elle existe. Une arête ayant pour extrémités les sommets u et v , afin d'alléger les notations, nous nous autoriserons même à écrire plus simplement " l'arête uv ". Enfin, nous serons fréquemment amenés à considérer des graphes pondérés (weighted graphs), c'est-à-dire muni d'une fonction de poids sur les sommets et/ou les arêtes.

. Complémentaire

Quand on recherche les propriétés d'un graphe, il est parfois plus simple d'étudier son complémentaire :

Définition 1.1. ***Le complémentaire*** (complement) d'un graphe G est le Graphe noté $\overline{G}$ défini par :

•$V_{\overline{G}} = V_G$,

• l'arête uv $(u \neq v) \in E_{\overline{G}}$ si et seulement si $uv \notin E_G$.

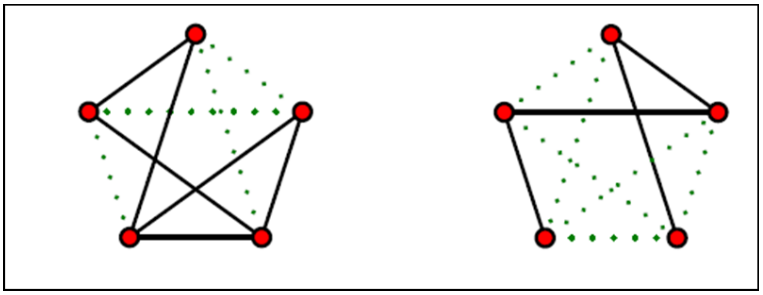


FIG. 1.3 – ***Un graphe et son complémentaire.***

- **Voisinage et degré**

Deux sommets v et w formant les extrémités d'une même arête sont dit adjacents (adjacent vertices) ou voisins (neighbors), ce que l'on note $v \sim w$ (et $v \nsim w$ dans le cas contraire). On définit alors le voisinage d'un sommet par :

Définition 1.2. ***Le voisinage ouvert*** (open neighborhood) d'un sommet v, noté $N(v)$, est l'ensemble des sommets qui lui sont adjacents.
Le voisinage fermé (closed neighborhood) d'un sommet v, noté $N[v]=N(v) \cup \{v\}$.
Un sommet n'ayant aucun voisin est qualifié d'isolé (isolated vertex).

On définit également le voisinage d'un ensemble de sommets

Définition 1.3. Le voisinage ouvert d'un ensemble de sommets S, noté $N(S)$, est l'ensemble $N(S) = \bigsqcup_{v \in S} N(v)$.

Définition 1.4. Le degré (degree) d'un sommet v, noté $d(v)$, est égal au nombre ($|N(v)|$) de ses voisins. Le plus petit degré d'un graphe G est noté $\delta(G)$ et le plus grand degré est noté $\Delta(G)$, ils sont respectivement égaux au degré d'un sommet ayant le moins et le plus de voisins.

- **Graphe adjoint d'un graphe**

Nous dirons également que deux arêtes sont adjacentes si elles possèdent une extrémité commune. On peut alors définir le " graphe d'adjacence des arêtes " de G, ou graphe adjoint de G comme suit:

Définition 1.5. Le graphe adjoint (line graph) d'un graphe $G = (E,V)$ est le graphe noté $L(G)$ tel que :
•$V_{L(G)}=E_G$,
• deux sommets de $L(G)$ sont adjacents si et seulement si les arêtes correspondantes dans G sont adjacentes.

Définition 1.6. Une chaîne (path) d'ordre n, notée P_n, est une séquence successive de sommets adjacents de type $c = (v_1,v_2,....v_{n+1}) / u_i u_{i+1} \in E$.
Le nombre d'arêtes d'une chaîne est la longueur de celle-ci.

. Connexité

La connexité d'un graphe est une mesure importante de sa robustesse quand on le considère comme un réseau (réseau de transport, réseau informatique.. .) .

Définition 1.7. Un graphe est connexe (connected) si pour toute paire de sommets il existe une chaîne les reliant.

La connexité définit une relation d'équivalence sur l'ensemble des sommets et chacune des classes d'équivalence est appelée composante connexe (connected component) du graphe. Autrement dit, un graphe est connexe si et seulement s'il ne contient qu'une seule composante connexe. Intuitivement, le nombre de composantes connexes correspond au nombre de fois qu'on léve la main quand on le dessine. Un résultat classique énonce que :

Lemme 1.1. *Pour tout graphe G, G est connexe ou $\overline{G}$ est connexe*

Définition 1.8. Un sommet (resp. ensemble) d'articulation (cutvertex) (resp. cutset)) est un sommet (resp. ensemble de sommets) tel que sa suppression rend le graphe non connexe.

Définition 1.9. La distance d'un sommet u à un sommet v dans un graphe connexe G notée $d(u,v)$ est la longueur de la plus courte chaîne qui relie u à v.

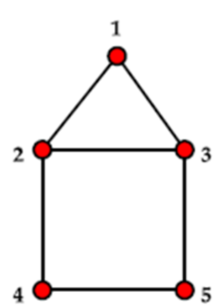
Définition 1.10. L'excentricité $e(v)$d'un sommet v dans un graphe connecte est définie par $max_{(u \in V)} d(u,v)$.

Définition 1.11. Le rayon $r(G)$ d'un graphe est son excentricité maximale.

Remarque 1.1. Un sommet v est central si $e(v) = r(G)$.
Le graphe complet G_P est le graphe de P sommets deux à deux adjacents.

1.2.2 Représentations des graphes en machine

Définition 1.12. ***La matrice d'adjacence*** (adjacency matrix) d'un graphe non orienté d'ordre n est la matrice carrée A de taille n telle que a_{ij}=1 s'il existe une arête entre les sommets i et j, et $a_{ij} = 0$ sinon.



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

FIG. 1.4 – ***Un graphe (la maison) et sa matrice d'adjacence.***

Les représentations par matrices ont de nombreuses propriétés. Par exemple, une matrice d'adjacence permet de tester l'adjacence de deux sommets en une seule opération (il suffit de regarder la valeur de l'entrée correspondante). Elles sont cependant mal adaptées pour représenter des graphes peu denses (c'est-à-dire ayant peu d'arêtes en regard du nombre de sommets), car elles sont alors très creuses (c'est-à-dire contiennent beaucoup de zéros). Pour ces graphes, on préfère souvent une représentation par listes d'adjacence.

Définition 1.13. ***Une liste d'adjacence*** (adjacency list) est une structure de données dans laquelle on associe à chaque sommet sa liste de voisins.

Ainsi, on ne stocke que les informations " utiles " concernant l'adjacence dans le graphe, ce qui permet un gain d'espace mémoire non négligeable par rapport à une représentation par matrice. En revanche, déterminer si deux sommets sont adjacents requiert en général plus d'opérations avec une liste d'adjacence qu'avec une matrice d'adjacence, puisqu'il se peut, qu'il faille parcourir la liste entière des voisins d'un sommet.

$$\begin{gathered}1 \rightarrow 2 \rightarrow 3\\ 2 \rightarrow 1 \rightarrow 3 \rightarrow 4\\ 3 \rightarrow 1 \rightarrow 2 \rightarrow 5\\ 4 \rightarrow 2 \rightarrow 5\\ 5 \rightarrow 3 \rightarrow 4\end{gathered}$$

Représentation par listes d'adjacence du graphe de la figure 1.4.

1.2.3 Sous-graphes

Définition 1.14. ***Un graphe*** $G' = (V', E')$ est un sous-graphe (subgraph) du graphe $G = (V,E)$ Si $V' \subseteq V$ et $E' \subseteq E$ et toutes les arêtes de E' ont leurs extrémités dans V', G' est alors un sur-graphe (supergraph) de G.

Définition 1.15. ***Un sous-graphe couvrant*** (ou parfois graphe partiel) (spanning subgraph) de $G = (V,E)$ est un sous-graphe $G'=(V,E')$ obtenir de G en supprimant uniquement des arêtes)

Définition 1.16. Le sous-graphe de $G = (V,E)$ induit par $V' \subseteq V$ (induced subgraph), noté $G[V']$, est le sous-graphe ayant pour ensemble de sommets V' et pour ensemble d'arêtes toutes les arêtes ayant leurs deux extrémités dans V'.

1.2.4 Stables et cliques

Définition 1.17. ***Un stable*** (stable set or independent set of graph G) est un ensemble de sommets deux à deux non adjacents.

Une clique(clique) est un ensemble de sommets deux à deux adjacents.
Ainsi une clique dans un graphe est un stable dans son complémentaire et inversement. De plus, tout graphe induit par une clique étant un graphe complet, les deux termes sont le plus souvent confondus ; de même, un graphe sans arête est régulièrement appelé stable

Les stables et les cliques interviennent dans de très nombreux problèmes pratiques. Pour la plupart de ces problèmes, on modélise la situation à l'aide d'un graphe, et la solution consiste à déterminer le plus grand stable ou la plus grande clique du graphe : on parle alors de stable ou clique maximum. Il convient d'ailleurs de veiller au point de vocabulaire suivant, légèrement différent de l'usage ≪ *normal* ≫

Définition 1.18. Un ensemble (de sommets, d'arêtes. . .) est **minimal** (resp. **maximal**) pour une propriété p si la suppression (resp l'ajout) d'un élément n'a plus la propriété p .
Un ensemble (de sommets, d'arêtes. . .) est **minimum** (resp. **maximum**) pour une propriété p s'il est de cardinalité minimale (resp. maximale) pour la propriété p.

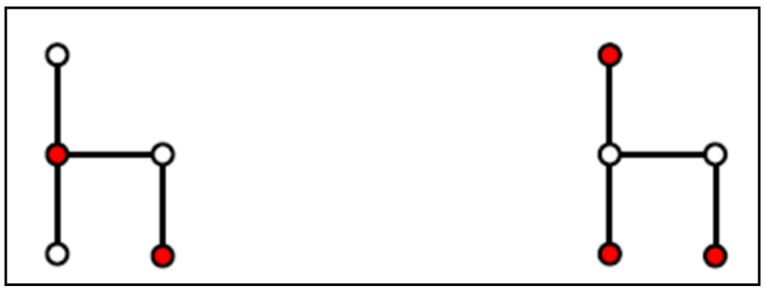


FIG. 1.5 – ***(a) Un stable maximal (b) Un stable maximum***

Il est facile de voir qu'un ensemble maximum est aussi maximal ; mais un ensemble maximal n'est pas nécessairement maximum. Par exemple, le stable formé par les sommets rouges sur la figure 1.5(a) est maximal (il est en effet impossible d'ajouter à ce stable un autre sommet pour former un stable de cardinalité supérieure), mais il n'est pas maximum (puisqu'il est possible de trouver un autre stable, de cardinalité supérieure (figure. 1.5(b))).

1.2.5 Quelques classes de graphes

Nous présentons ici quelques classes de graphes particulières qui occupent une place majeure en théorie des graphes et que nous rencontrerons souvent par la suite, avec leurs propriétés et les liens entre elles.
· Graphes parfaits
Les graphes parfaits ont été introduits par Berge qui essayait de résoudre un problème de théorie de la communication.

· Graphes bipartis, arbres et forêts
Les graphes bipartis, sont définis comme étant les graphes dont l'ensemble de sommets peut être partitionné en deux stables (ou, de façon équivalente, les graphes 2-coloriables). Il est intéressant de noter que les graphes bipartis peuvent également être caractérisés comme suit :
Proposition 1.1. [15] *Un graphe est biparti si et seulement s'il ne contient aucun cycle de longueur impaire.*

Définition 1.19. Une **forêt** (forest) est un graphe sans cycle.
Un **arbre** (tree) est un graphe connexe sans cycle.

Définition 1.20. ***Isthme*** (cut edge) Arête de G telle que le nombre de composantes connexes de G augmente de 1 si on la supprime.

Définition 1.21. ***Blocs graphes*** Un bloc de G est un sous-graphe connexe sans sommet déconnectant maximal.

Les définitions suivantes sont équivalentes :

1. Un graphe est un graphe en bloc s'il est connexe et chaque bloc est une clique.
2. Un graphe est un graphe en bloc s'il peut être construit d'un arbre en remplaçant chaque arête par une clique de taille arbitraire, les cliques ont tout au plus un sommet en commun.
3. Un graphe est un graphe en bloc si pour chaque quatre sommets (u,v,w,x), les deux plus grandes sommes parmi les distances suivantes :
   (a) $d(x,y) + d(z,w)$
   (b) $d(x,z) + d(y,w)$
   (c) $d(x,w) + d(y,z)$
   Sont égales.

4. Un graphe est un graphe en bloc si pour chaque quatre sommets (u,v,w,x) :
   $d(u,v)+ d(w,x) \leq \max\{(d(u,w)+ d(v,x), d(u,x)+ d(v,w)\}$.

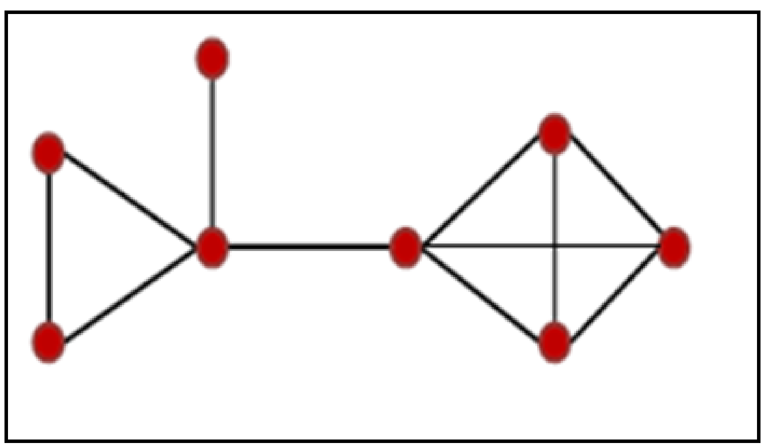


Fig. 1.6 – ***Exemple d'un graphe en bloc***

Proposition 1.2. *Deux blocs d'un graphe ont au plus un sommet commun.*

Remarque 1.2. xy est un bloc de $G \Longleftrightarrow xy$ est un isthme.

Remarque 1.3. L'ensemble des blocs forme une partition de l'ensemble des arêtes du graphe

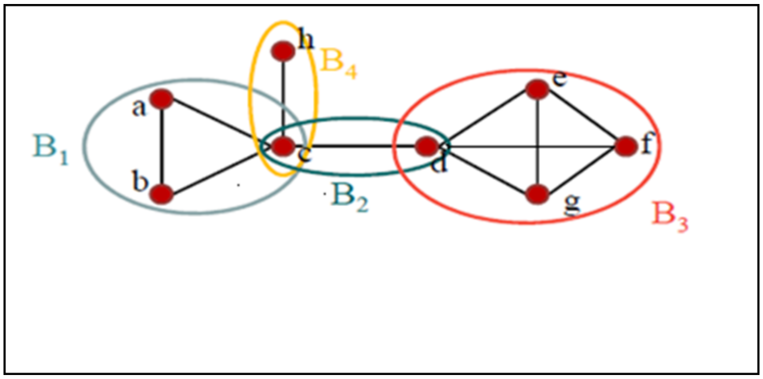


FIG. 1.7 – ***Exemple d'un ensemble de blocs qui forment une partition de l'ensemble des arêtes du graphe***

Chapitre 2

Complexité, Domination et Broadcast

2.1 Introduction

Dans ce chapitre nous donnons un bref aperçu de la théorie de la complexité des algorithmes en introduisant les notions fondamentales de classes de problèmes P (polynomial) et NP (Non-déterministe Polynomial) puis nous présentons très rapidement quelques définitions de la dominance, ensuite nous donnons quelques résultats et notions sur le problème du broadcast dominant optimal ainsi que sa complexité dans les graphes quelconques.

2.2 Complexité

La théorie de la complexité est une branche des mathématiques et de l'informatique ayant pour cadre l'étude de la difficulté intrinsèque des problèmes algorithmiques et vise à classer ces problèmes en fonction de cette difficulté. Ici, les mots " complexité " et " difficulté " ne se rapportent pas à la mise au point d'un algorithme de résolution ou aux concepts avancés auxquels il peut faire appel (comme une structure de données élaborée), mais plutôt à la quantité de ressources à utiliser pour résoudre le problème. En ce qui nous concerne, les ressources consistent en le temps que met un algorithme à résoudre le problème (c'est sa complexité temporelle) et l'espace mémoire qu'il utilise au cours de son exécution (sa complexité spatiale), mais il existe d'autres mesures de complexité, comme le nombre de portes logiques à utiliser pour la réalisation d'un circuit, ou encore la quantité d'information à transmettre dans le cadre de la théorie de la complexité de la communication. Le principal atout de la théorie de la complexité est que ces grandeurs sont exprimées indépendamment de tout dispositif physique concret ; au contraire, elle est basée sur un modèle de calcul abstrait, généralement une machine de Turing, ce qui permet de comparer facilement l'efficacité de deux algorithmes en s'affranchissant de considérations telles que la vitesse du processeur. On est quasiment sûrs aujourd'hui que certains problèmes nécessitent, pour leur résolution, des algorithmes dont le temps de calcul est bien supérieur à celui d'autres problèmes, et c'est en ce sens que l'on dira qu'ils sont $\ll$ plus difficiles $\gg$.

2.2.1 Concepts de base

Définition 2.2.1. Un problème est une question générale possédant des paramètres dont la solution depend de la valeur de ces paramétres et il est généralement inconnue.
Une instance d'un problème est obtenue en affectant une valeur à chacun de ses paramètres. La taille d'une instance désigne généralement la la valeur d'un ou de plusieurs de ses paramétres.

Exemples 2.2.1. Le problème du voyageur de commerce (ou TSP pour Traveling salesman problem) consiste, étant donné un ensemble de villes séparées par des distances connues, 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).
Nous donnons maintenant des définitions plus formelles, des notions abordées dans le paragraphe précédent. On commence par définir la notion de complexité et la notation " grand O ", puis les classes des problèmes P et NP qui en découlent.

2.2.2 Problèmes de décision et d'optimisation

Définition 2.2.2. Un problème de décision est un problème auquel la réponse est oui ou non.

Définition 2.2.3. Un problème d'optimisation consiste à déterminer la meilleure solution parmi toutes les solutions réalisables (minimum global ou maximum global).

2.2.3 Complexité en temps et en espace

Définition 2.2.4. La complexité temporelle d'un algorithme correspond au nombre d'instructions élémentaires (opérations arithmétiques, comparaison, affectation. . .) effectuées au cours de son exécution.
Définition 2.2.5. La complexité spatiale d'un algorithme correspond au nombre de cases mémoires occupées par les données manipulées par l'algorithme au cours de son exécution.

En général, le temps d'exécution dépend de la taille de l'instance du problème considéré, en particulier, plus une instance est grande, plus le problème demandera de temps pour être résolu. Par exemple, si l'on considère un algorithme de tri d'un tableau d'entiers, le nombre d'instructions et l'espace occupé ne seront pas les mêmes si l'on travaille sur un tableau de 10 entiers ou sur un tableau de 10 000 entiers. On exprime donc le temps (ou toute autre mesure de complexité) en fonction de la taille d'une instance générale du problème

considéré, souvent notée n. En algorithmique des graphes, en fonction du problème traité, on choisit généralement le nombre n de sommets, ou le nombre m d'arêtes du graphe. Mais même sur des instances de même taille, une complexité peut varier d'une instance à une autre : par exemple, rechercher une valeur dans un tableau demande plus de temps dans un tableau dont les éléments sont désordonnés que dans le même tableau trié. Aussi définit-on généralement une complexité en considérant la pire instance possible parmi toutes les instances de taille n, c'est-à-dire celle demandant le plus de ressources. Dans la suite, nous nous intéresserons qu'à la complexité temporelle des problèmes étudiés.
La notation grand O, dite aussi symbole de Landau, décrit le comportement asymptotique d'une fonction, exprimé à l'aide d'une autre fonction généralement plus simple. Plus formellement, nous dirons :

Définition 2.2.6. ***(Notation grand O)***
$F(n)$=O(g(n))(F est un grand $O(g)$)quand N$\rightarrow \infty$ si et seulement si $\exists$M$>$ 0,$n_0 \in N$ telque $\forall n \geq n_0 \mid F(n) \mid \leq M \mid g(n) \mid$
Intuitivement, ceci signifie qu'à partir de n_0 et à un facteur constant près, F ne croît pas plus rapidement que g.

Exemple 2.1. Soit $F(n)$=$6n^4 - 2n^3$+5. Choisissons n_0=1. Alors pour tout $n \geq 1$

$$\begin{aligned} \mid 6n^4 - 2n^3 + 5 \mid &\leq 6n^4 - 2n^3 + 5 \\ &\leq 6n^4 + 2n^4 + 5n^4 \\ &= 13n^4 \\ &= 13 \mid n^4 \mid \end{aligned}$$

Ainsi, en prenant M=13, on a $F(n) = O(n^4)$. Autrement dit, à un facteur constant près, F ne croît pas plus rapidement que n^4. Il est facile de voir qu'un polynôme $P(n)$ de degré k est toujours en $O(n^k)$.

2.2.4 Classes de complexité $PetNP$

Définition 2.1. Un problème de décision p est dans la classe P si, pour chacune de ses instances, de taille n, il existe un réel positif k tel qu'il peut être résolu par un algorithme de complexité temporelle $O(n^k)$ c'est-à-dire qu'il peut être décidé en temps polynomial. Les problèmes de la classe P sont dits faciles. Ce sont ceux que l'on sait résoudre efficacement.

Définition 2.2. Un problème de décision est dans la classe NP si l'on peut vérifier en temps polynomial qu'une solution pour une instance donnée est valide (ce que l'on appelle un certificat du oui).
Notons que si un problème est dans la classe NP, alors on ne sait pas est ce que parcequ'il n'existe pas d'algorithme polynomial pouvant le résoudre ou parcequ'on ne peux pas trouver un tel algorithme.

Exemple 2.2. 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 $\mid S \mid \geq k$ 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 $P \subset NP$.

2.2.5 Problèmes $NP-$complets

Une réduction est une transformation d'un problème en un autre , ceci permet de capturer la notion informelle de " problème au moins aussi difficile qu'un autre problème ". Plus précisément, si un problème X peut être résolu en utilisant un algorithme permettant de résoudre un problème Y, c'est que X n'est pas plus difficile que Y, on dit alors que X se réduit à Y. Par exemple, le problème consistant à élever un nombre au carré se réduit au problème plus général de multiplication de deux nombres (ici, aucune transformation n'est nécessaire). Une réduction est polynomiale lorsque le processus de transformation peut se faire en temps polynomial.

Définition 2.3. Un problème X est difficile pour une classe de problèmes C, ou C-difficile, si tout problème de C se réduit à X. Autrement dit, il n'existe pas de problème de C plus difficile que X, puisque tout algorithme résolvant X résout aussi n'importe quel problème de C. En particulier, les problèmes difficiles pour la classe NP forment la classe de problèmes NP-difficiles. Lorsque, de plus, X appartient lui aussi à la classe C, on dit qu'il est complet pour C, ceci signifie que X est l'un des problèmes les plus difficiles de C (il peut en effet y avoir plusieurs problèmes de même difficulté).

Définition 2.4. Un problème de décision est NP- complet s'il vérifie les deux conditions suivantes :
• il appartient à la classe NP
• tous les problèmes de la classe NP se ramènent à celui-ci par une réduction polynomiale.

Tout problème d'optimisation combinatoire dont le problème de décision associé est NP-complet est NP-difficile. Le problème du stable maximum dans un graphe quelconque est NP-difficile [9]

2.3 La Domination

Dans cette partie, nous présentons les définitions principales liées à la domination. On pourra trouver des notions plus complètes dans les deux livres de Haynes, Hedetniemi et Slater.

2.3.1 Les variantes de la domination

•Ensemble dominant

Un ensemble dominant d'un graphe $G = (V,E)$ est un ensemble $S \subseteq V$ de sommets tels que tout sommet de $V \backslash S$ est adjacent au moins à un sommet de S.
Plus formellement, S vérifie :
$V \backslash S \subseteq N(S)$ ou encore $V = N[S]$. Le nombre de Domination $\gamma(G)$ du graphe G est le cardinal minimum d'un ensemble dominant. On dit qu'un sommet domine un autre sommet si les deux sommets sont voisins. Un dominant est donc un ensemble de sommets qui domine tous les autres sommets. Nous rappelons qu'un empilement d'un graphe G est un ensemble de sommets S tels que les ensembles $N[x]$, $x \in S$ sont disjoints deux ‘à deux. Le nombre d'empilement $\rho(G)$ est le cardinal maximum d'un empilement de G.

• Domination totale

La domination totale est l'une des principales variantes de la domination. Elle a été introduite par Cockayne, Dawes et Hedetniemi dans [4]. Un dominant total d'un graphe $G = (V,E)$ sans sommet isolé est un ensemble $S \subseteq V$ de sommets tel que tout sommet de V est adjacent à un sommet de S. Plus formellement, S vérifie $V = N(S)$ Le nombre de domination totale $\gamma_t(G)$ du graphe G est le cardinal minimum d'un dominant. Dans la domination totale, contrairement à la domination simple, il faut que les sommets sélectionnés dans l'ensemble soient eux aussi dominés par un autre sommet de l'ensemble. C'est pourquoi il n'est pas possible de trouver un dominant total dans un graphe contenant un sommet isolé, car celui-ci ne peut pas être dominé . De plus, on remarque qu'un dominant total est nécessairement un dominant, et $\gamma_t(G) \geq \gamma(G)$.[15]

2.4 Broadcast

La notion de broadcast a été introduit par Erwin en 2001 pour généraliser le problème de la domination, tel que les différents sommets peuvent être assignés différentes puissances de domination. Le broadcast dominant assigne une puissance de nombre entier $F(v) \geq 0$ ¸ à chaque sommet v du graphe donné, tel que chaque sommet du graphe est sur la distance $F(v)$ d'un certain sommet v ayant $1 \leq F(v) \leq e(v)$.
Le broadcast dominant optimal cherche à minimiser la somme des puissances assignées aux sommets du graphe. Depuis l'introduction de ce problème, la complexité informatique a été ouverte, et on pensait plus qu'il est NP-difficile.
Dans ce qui suit, on présente les principales notions liées à ce problème et nous demontrons que le problème du broadcast dominant optimal est dans la classe P , et nous donnons un algorithme en temps polynômial (voir annexe 2) pour le résoudre sur des graphes quelconques.
Depuis son introduction [1,14], beaucoup de paramètres du graphe liés à la domination ont été présentés et étudiés. Le problème de domination standard peut être vu comme un moyen de représenter un ensemble de villes ayant des broadcast stations, où chaque ville

peut entendre un broadcast station placée dans cette ville ou bien dans une autre ville voisines. Erwin 2002 [7] présenta le problème de broadcast optimal, qui est plus réaliste dans le sens que les diverses stations de broadcast sont autorisées à transmettre aux différentes puissances. Des stations par radio FM sont distinguées par leur fréquence de transmission et par leur PRE (Puissance Rayonnée Efficace).
Un émetteur avec un plus haut PRE peut transmettre plus loin, mais il est plus coûteux à construire et utiliser.
En conséquence, le problème du broadcast dominant optimal consiste à évaluer la fonction de puissance F sur les sommets, telle que chaque sommet de graphe est à la distance au plus $F(v)$ d'un certain sommet v ayant $1 \leq F(v) \leq e(v)$, et la somme des puissances est minimisé.
Depuis, la plupart des problèmes dominants intéressants sont de classe NP-difficile sur les graphes quelconques, ceci a donné une certaine indication que la domination optimale d'un Broadcast pourrait également être NP-difficile pour les graphes quelconques. Après ceci, en 2003 Blair et autres ont donné un algorithme en temps polynômial pour la recherche du nombre de broadcast dominant pour les graphes d'intervalle, et les graphes séries-parallèle [3] . Puis en 2006 dans [14] on trouve un algorithme polynomial pour la détermination d'un broadcast dominant optimal pour un graphe quelconque.

2.4.1 Définitions et terminologie

Définition 2.5. ***Broadcast***
Une fonction $F : V \to \{0,1,\ldots\ldots..diam(G)\}$ est un broadcast sur G, si
$\forall v \in V/0 \leq F(v) \leq e(v)$.
Le coût d'un broadcast C_F=$\sum_{v\in V} F(v)$
L'ensemble de broadcast dominateurs définie par F, est l'ensemble:
V_F=$\{v \in V \mid F(v) \geq 1\}$, l'ensemble $F\cdot$ dominés est $V_0 = V - V_F$

Définition 2.6. ***Broadcast dominant***
Un broadcast est dit dominant si pour chaque sommet $u \in V$,$\exists v \in V_F$ telque $d(u,v) \leq F(v)$

Définition 2.7. ***Le coût d'un broadcast dominant***
Nous notons $C_F(V)$ également comme $C_F(G)$, et nous référons à lui le coût F dans G .Notons qu'il y a toujours un Broadcast dominant de coût $rad(G)$.

Définition 2.8. ***Le nombre de broadcast dominant***
Le nombre de broadcast domination, $\gamma_b(G)$ est le plus petit coût d'un broadcast dominant dans G, $\gamma_b(G)$=Min$\{C_F$ où F est un broadcast dominant dans $G\}$

Définition 2.9. ***Broadcast dominant optimal***

Si $C_F(G)$=$\gamma_b(G)$ alors nous appelons F broadcast dominant optimal dans G. Pour un graphe non connexe, un broadcast dominant optimal est l'union des Broadcasts de domination optimaux de ses composants connexes, Ceci justifie la restriction de l'étude du problème aux graphes connexes.

Définition 2.10. ***Broadcast dominant efficace***

Un broadcast de domination est efficace si chaque sommet est dominé par un seul sommet.

Définition 2.11. ***Graphe de domination***

Soit F broadcast de domination efficace sur $G = (V,E)$, on définit un graphe de domination G_F=(V_F,E_F) tel qu'il existe une arête entre deux sommets $u,v \in V_F$ Si et seulement si $B_G(u,F(u))$ est adjacente à $B_G(v,F(v))$ dans le graphe G , Figure.2.1. Heggernes et Lokshtanov [14] ont prouvé ce qui suit pour un broadcast F dominant efficace dans G, si G_F a un sommet de degré supérieur a 2 alors il y a un broadcast dominant F' efficace sur G, tels que $\mid V_{F'} \mid < \mid V_F \mid$ et $C_{F'}(G)$=$C_F(G)$.

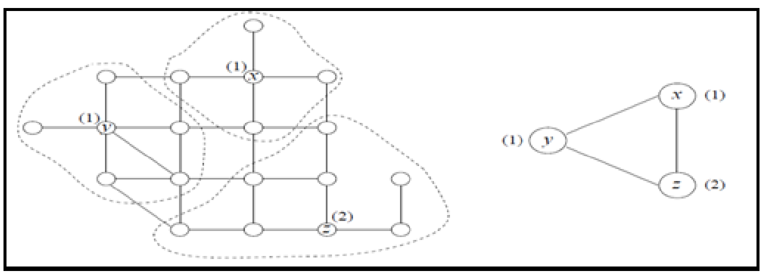


FIG. 2.1 – ***l'exemple d'un graphe F de broadcast dominant efficace F***

La figure 2.1 illustre l'exemple d'un graphe G de broadcast dominant efficace F. Pour chaque sommet v avec $F(v) \geq 1$, sa puissance $F(v)$ est montrée entre parenthèses, et les courbes à tiret indiquent les boules $B(v,F(v))$. Pour tous les autres sommets w, $F(w) = 0$. Du côté droit, le graphe correspondant de domination G_F, et le poids de chaque sommet est montré entre parenthèses.

Définition 2.12. ***Broadcast clairsemé***
on définit un broadcast clairsemé G comme étant un broadcast F optimal tels que V_F est réduit au minimum. Par suite, de ce qui précède, si F est un broadcast optimal clairsemé sur G, alors F est efficace et G_F est une chaîne ou un cycle.

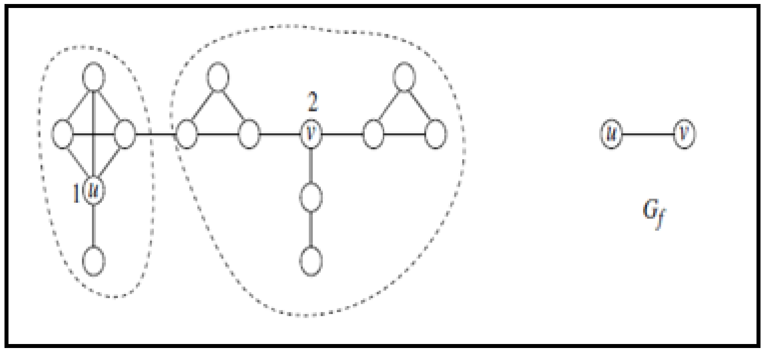


FIG. 2.2 – ***Un graphe de bloc G avec un broadcast F clairsemé***

Figure.2.2 : Un graphe en bloc G avec un broadcast F clairsemé est montré du côté gauche, où les dominateurs du broadcast F sont u et v, avec $F(u) = 1$ et $F(v) = 2$. Les boules $B(u,1)$ et $B(v,2)$ sont indiqués sur G et du côté droit est dessiné G_F.

2.4.2 La structure de broadcast optimal dominant

Dans [6] , Dunbar et autres prouvent que chaque graphe a un Broadcast dominant optimal qui est efficace. En particulier, le lemme suivant est la preuve de ce résultat.

Lemme 2.1. [6] *Pour tout Broadcast dominant non efficace d'un Broadcast F dans un graphe* $G = (V,E)$ *il y a un Broadcast dominant efficace* F' *dans* G *tel que* $\mid V_{F'} \mid < \mid V_F \mid$ *et* $C_{F'}$=C_F.

Lemme 2.2. [12] *Soit* F *un broadcast non dominant efficace dans* $G = (V,E)$ *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* $\mid V_{F'} \mid < \mid V_F \mid$ *et* $C_{F'}$=C_F.

Lemme 2.3. [12]

Pour tout graphe G*, il y a un broadcast dominant optimal efficace* F *sur* G *tels que le graphe de domination* G_F *est une chaîne ou un cycle.*

Lemme 2.4. [12]

Dans tout Broadcast dominant optimale F *d'un graphe* $G = (V,E)$*, s'il y a deux voisins* v,w *de* V *tels que* $F(v) > 0$ *et* $F(w) > 0 \Longrightarrow F(v)$=$F(w)$

Lemme 2.5. [12] *La complexité de l'algorithme* ***RMPBD*** *sur un graphe G avec n sommets est* $O(n^4\ )$ *.*

2.4.3 algorithme pour le cas général

[12] (voir l'annexe2) L'ideé de l'algorithme est comme suit:
soit x n'importe quel sommet de G. Pour chaque k entre 1 et rad $(G\ )$ tels que $G' = G(V \backslash B(x,k))$ est connexe ou vide, nous parcourons l'algorithme de plus courte chaîne de broadcast dominant (**RMPBD** [12](voir l'annexe1)) sur G'.

De cette façon, nous considérons tous les Broadcast dominant F dont les graphes de domination correspondant sont des chaînes ou des cycles. L'avantage de cette approche est sa simplicité. L'inconvénient est que nous considérons également beaucoup de cas qui ne font pas correspondre à une chaîne ou à un cycle, que nous avons détectés avec un algorithme plus long et plus compliqué. Cependant, ces cas inutiles ne menace pas l'exactitude de l'algorithme, et leur détection ne diminue pas la limite asymptotique du temps.

Théorème 2.1. [12] *L'algorithme* ***OBD****(Broadcast Dominant Optimal(voir annexe)) calcule un broadcast dominant optimal d'un graphe quelconque.*

Théorème 2.2. [12] *La complexité de l'algorithme* ***OBD*** *sur un graphe quelconque* G *à* n *sommets est* $O(n^6)$.

Preuve. D'abord l'algorithme **OBD** trouve le rayon de G ce qui peut être fait en temps $O(n^3)$ pour chaque itération de la boucle intérieure nous découvrons si G est connexe, et l'algorithme fait un appel à l'algorithme (**RMPBD**). La première de ces tâches peut être faite en temps $O(n+m)$, alors que la seconde est faite en $O(n^4)$ par le Lemme 2.4 La boucle intérieure répéter $O(n^2)$ des périodes, et la preuve est complète.

Chapitre 3

Broadcast domination dans les graphes en bloc

3.1 Introduction

Dans ce chapitre, nous présentons un algorithme linéaire pour la recherche d'un broadcast dominant optimal dans les graphes en bloc. Les graphes en bloc sont des superclasses d'arbres dans lesquels chaque cycle est une clique.
Cependant, la manière de construction de notre algorithme passe par un certain nombre de nouveaux résultats structuraux sur les broadcast domination, que nous allons voir en premier.

3.2 Les Propriétés générales d'un broadcast dominant

Nous commençons par une remarque simple qui nous permettra de remplacer les dominateurs broadcast d'un sous-graphe induit avec d'autres, sans affecter le reste des dominateurs de broadcast.

Remarque 3.1. Soit $G = (V,E)$ un graphe, $A \subseteq V$, et G'=$G[A]$ Si G admet un broadcast F dominant tel que $\cup_{a\in V_f \cap A} B(a,F(a)) = A$, et F' est un broadcast optimal dans G', alors G admet un broadcast $F^\star$ défini comme suit :

$$F^\star(v) = \begin{cases} F'(v) & si \quad v \in A \\ F(v) & si \quad v \in V\backslash A \end{cases}$$

Les prochains lemmes et corollaires nous fourniront des outils pour juger si un broadcast dominant donné est clairsemé.

Lemme 3.1. [13] *Soit B_1=$B(v,k_v\ )$ et B_2=$B(u,k_u)$ deux boules dans le graphe G.
Si $k_v \leq k_u$ alors il existe un sommet $z \in G$ tel que $B_1 \cup B_2 \subseteq B(z,\ell)$ où*

$$\ell = \begin{cases} k_u & si \quad B_1 \subseteq B_2 \\ \frac{\mathrm{d}_G(u,v)+k_v+k_u}{2} & sinon \end{cases}$$

Preuve. Si B_1 est un sous-ensemble de B_2, $z = u$ et que $B_1 \cup B_2 \subseteq B(z,l)$=$B_2$. Autrement dit, soit z un sommet sur la plus courte chaîne de u à v tel que $d_G\ (u,z)$=$\lceil d_G\ (u,v)+k_v - k_u)/2\rceil$. Alors $d(z,x) \leq k_u$+$d(z,u) = l$, pour chaque sommet x dans B_2, et par conséquent $B_2 \subseteq B(z,\ell)$. Pour chaque sommet y dans B_1,$d(z,y) \leq k_v$+$d(v,z) = k_v$+$d(v,u)$−$d(u,z) \leq \ell$, en conséquence aussi $B_1 \subseteq B(z,\ell)$.

Corollaire 3.1. [13] *Soit $B_1 = B(v,k_v\ )$ et $B_2 = B(u,k_u)$ deux boules dans le graphe G. Si $k_v + k_u \geq d_G(u,v) + 2k$ pour un entier $k \leq$min $\{k_v,k_u\}$, alors il existe un sommet $z \in G$ tel que $B_1 \cup B_2 \subseteq B(z,k_v + k_u - k)$.*

Lemme 3.2. [13]*Soit x,y,z trois sommets dans un graphe G. Soit P_y la plus courte chaîne de x à y dans G, et P_z la plus courte chaîne de x à z dans G. Si $P_y \cap P_z$ contient plus de sommets que x, alors il existe trois (03) nombres entiers k_x,k_y,k_z tels que $B(x,k_x)$est adjacente aux deux boules suivantes $B(y,k_y)$ $B(z,k_z)$, dans ce cas il existe un sommet v dans G tel que :*
$B(x,k_x) \cup B(y,k_y) \cup B(z,k_z) \subseteq B(v,k_x + k_y + k_z)$.

Preuve. S'il existe un sommet quelconque différent de x dans $P_y \cap P_z$, alors la plus courte chaîne de x à ce sommet est une sous-chaîne d'une plus courte chaîne de x à l'un des deux sommets suivants y,z. Par conséquent, le sommet u est adjacent à x dans P_y, en plus u est le sommet le plus proche à y et z . Puisque u et x sont adjacents, on note aussi que $B(x,k_x) \subseteq B(u,k_x + 2)$. Nous observons que $B(u,k_x + 2)$ recouvre $B(z,k_z)$ par le corollaire 3.1, par conséquent il existe là un sommet w tel que $B(u,k_x + 2) \cup B(z,k_z) \subseteq B(w,k_z + k_x + 1)$et cette boule recouvre aussi $B(y,k_y)$ par corollaire 3.1, où il existe un sommet v tel que $B(w,k_z + k_x + 1) \cup B(y,k_y) \subseteq B(v,k_z + k_x + k_y)$.
Par conséquent,$B(x,k_x) \cup B(y,k_y) \cup B(z,k_z) \subseteq B(y,k_y) \cup B(u,k_x + 2) \cup B(z,k_z)$

$$\subseteq B(y,k_y) \cup B(w,k_z + k_x + 1)$$
$$\subseteq B(v,k_z + k_x + k_y)$$

Corollaire 3.2. [13]*Soit F un broadcast clairsemé sur un graphe $G = (V,E\ )$et soit x,y et z trois sommets distincts dans V_F tel que y et z sont des voisinages de x dans G_F et l'intersection d'une plus courte chaîne de x à y dans G et d'une plus courte chaîne de x à z dans G est exactement l'ensemble $\{x\}$.*

3.3 Les propriétés structurales d'un broadcast dominant sur des graphes en bloc

Un graphe G est dit graphe en bloc si les sommets de chaque cycle forment une clique.

Théorème 3.1. [13]*Soient quatre sommets x,y,z et w dans un graphe en bloc G, alors les deux plus grandes sommes parmi les distances suivantes :*

1.$d(x,y)$+$d(z,w)$

$$2.d(x,z)+d(y,w)$$
$$3.d(x,w)+d(y,z)$$

Sont égales.

Chaque cycle dans un graphe en bloc est une clique et n'importe quelle plus courte chaîne contient tout au plus deux sommets d'une clique, il s'en suit qu'une plus courte chaîne entre n'importe quelle paire de sommets est unique dans un graphe en bloc. En particulier, chaque sommet sur une plus courte chaîne entre deux sommets s et t est un sommet d'articulation séparant s de t.

Lemme 3.3. [13]*Si G est un graphe en bloc alors $rad(G) = \lceil diam(G)/2\rceil$*

Preuve. Si G est un graphe complet alors le resultat est trivial. Supposons que G n'est pas complet et soit $P = s, \ldots.., t$ une chaîne diamétrale dans G. Alors il existe un sommet $x \in P$ tel que $\min\{d(s,x),d(t,x)\}=\lceil diam(G)/2\rceil$. Rappel, x est un sommet d'articulation. Nous démontrons que $e(x) \leq \lceil diam(G)/2\rceil$ par contradiction, en supposant qu'il existe un sommet y tel que $d(x,y) > \lceil diam(G)/2\rceil$. Puisque x sépare s et t, et sépare également y du sommet s ou de t. Sans perte de généralité, soit x le séparant de y et t. Dans ce cas, $d(y,t) = d(y,x) + d(x,t)$. Puisque $d(x,y) > \lceil diam(G)/2\rceil$
et $d(x,t) \geq \lceil diam(G)/2\rceil$. Contradiction, $d(y,t) > diam(G)$. Par conséquent nous concluons que : $e(x) \leq \lceil diam(G)/2\rceil$ donc $rad(G) \leq \lceil diam(G)/2\rceil$. Puisque $rad(G) \geq \lceil diam(G)/2\rceil$, donc le rayon de G est exactement $\lceil diam(G)/2\rceil$.

Lemme 3.4. [13] *Soit G un graphe en bloc et s un sommet de G. Alors $e(s) = diam(G)$ si et seulement si G a un sommet x tel que*
$d(x,s) \geq d(x,y)$ pour tous $y \in V(G)$.

Preuve. Comme la direction est insignifiante nous pouvons prendre x comme l'extrémité de la chaîne diamétrale du graphe G, et le sommet s pour l'autre direction, donc $\forall y \in V(G)$, nous avons $d(x,s) \geq d(x,y)$.Soit $P = a, \ldots\ldots\ldots, b$ une chaîne diamétrale dans G, et $d_1 = d(a,b) + d(x,s)$,$d_2 = d(a,x) + d(b,s)$,d_3=$d(a,s) + d(b,x)$. Puisque $d(x,s)$ supérieur ou égale à $d(a,x)$ et à $d(b,x)$, et $d(a,b)$ supérieur ou égale à chaque distance dans G, on remarque aussi que $d_1$ne peut pas être inferieur à d_2 ou d_3. Soit $d_2 \leq d_3$. Par le théorème 3.1, $d_1 = d_3$. Puisque $d(s,x) \geq d(x,b)$ on a aussi $d(a,b) \leq d(a,s)$. Et puisque $d(a,b) = diam(G)$, nous avons également $d(s,a) = diam(G)$, par conséquent $e(s) = diam(G)$.

Remarque 3.2. Si G est un graphe en bloc et F un broadcast clairsemé dans G alors G_F est une chaîne.

Lemme 3.5. [13] *Soit G un graphe en bloc et F un broadcast clairsemé dans G. Pour chaque clique C de taille au moins 3, tous les sommets de la clique C sont dominés par le même dominateur V_F.*

Preuve. Puisque G_F est une chaîne et F efficace (selon les résultats précédents), les sommets de la clique C peuvent être dominés au plus par deux sommets distincts. La clique C contient au moins trois sommets, donc au minimum deux sommet x,y dans C doivent être dominés par le même dominateur z, z$\in V_F$. Puisque dans G, les distances entre les

sommets de la clique C sont égales à 1, et par le théorème 3.1 :
$d_G(w,z) \leq d_G(x,z) = d_G\ (y,z)$ pour chaque sommet $w \in C$. Donc chaque sommet w dans C est dominé exactement par le sommet z.

3.4 Les propriétés algorithmiques d'un broadcast dominant sur des graphes en bloc

Se rappeler, que n'importe quelle clique d'un graphe en bloc s'intersecte en au plus en deux sommets d'une chaîne diamétrale. Soit P une chaîne diamétrale, et $C(P)$ l'union de toutes les cliques, cela intersecte avec P exactement dans deux sommets. Un ensemble $C \subseteq V(G)$ s'appelle le noyau de G si $C = C(P)$ pour une chaîne diamétrale C dans G. Par la série des lemmes de cette partie, on va voir dans la suite que chaque graphe en bloc $G = (V,E)$ admet au moins un broadcast dominant optimal F et un noyau C, tel que chaque dominateur appartient à la fois à V_F et à C. Enfin le lemme 3.11, nous permettra de trouver les dominateurs et leurs poids respectifs dans F , comme sera décrit ci-dessus [13] nous commençons par définir deux opérations. Ces opérations sont illustrées dans la Fig 3.1.

Définition 3.1. Soient $B = B(a_0,k)$ une boule et une chaîne $P = a_0,a_1,\ldots\ldots a_p$ dans un graphe G, et soit ℓ un nombre entier positif $\ell \leq p$, l'élargissement de la boule $B = (a_0,k)$ avec ℓ sommets tout au long de P, est une boule $B(a_\ell,k + l)$. Ceci est également désigné sous le nom de la boule élargie(B,P,ℓ).

Définition 3.2. Soit une boule $B = B(a_0,k)$ donnée et une chaîne $P = a_0,a_1,\ldots\ldots a_p$ dans un graphe G, et un nombre entier positif $\ell \leq min\{k-1,p\}$, le rétrécissement de la boule $B = (a_0,k)$ avec ℓ sommets tout au long de P, est une boule $B(a_\ell,k-\ell)$. Ceci est également désigné sous le nom de la boule rétrécie (B,P,ℓ).

Remarque 3.3. Soit B_1 une boule, si B_2= élargie(B_1,aPb,ℓ) alors B_1= rétrécie$(B_2$,aPb,$\ell)$.

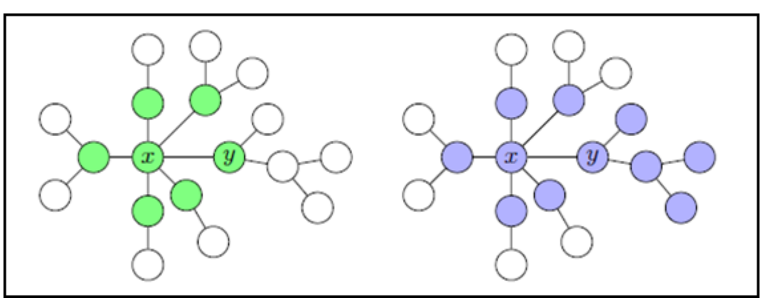


FIG. 3.1 – ***Exemple d'une boule élargie***

Une boule $B_1 = B(x,1)$ est montrée en vert du côté gauche, et la boule élargie

$(B_1,(x,y),1) = B(y,2)$ en bleue

Preuve. $B(v',k')$=elargie (B,P,L) puisque d_G (v $,v')\leq L$, d_G $(v',x)\leq d_G$ $(v',v)+d_G(v,x) \leq k+L=k'$ pour chaque $x \in B$, ainsi pur chaque sommet dans B egalment dans $B(v',k')$.

Lemme 3.6. *Soit P la plus courte chaîne entre deux sommets s et t dans un graphe en bloc G. Soit u un sommet de P et k_1 ,k_2 deux nombres entiers positifs tels que :*
$k_1 < k_2$ et $k_1 \leq d_G$ (u,s).
Étant donné B_1,B_2 deux boules telles queB_1=$B(u,k_2)$ etB_2=rétrécie(B_1,uPs,k_1), si $t \in B_2$ alors $B_1 \cap B(s,d(s,t))$=$B_2 \cap B(s,d(s,t))$.

Preuve. soit u'un sommet sur uPs tel que $B(u',k_2-k_1)$=B_2. Il devrait être clair que uPs soit la plus courte chaîne de u à s. Rappelons que u' dans ce cas-ci est un sommet d'articulation séparant u et s. Ainsi, il y a un ensemble de sommets S$\subseteq B(s,d(s,t))$ tel que chaque chaîne de u à un sommet de S passe par le sommet u'.
Chaque chaîne de s à un sommet dans $T = B(s,d(s,t))\backslash$S passe par u'. La chaîne $u'Pu$ est la chaîne la plus courte, puisque c'est une chaîne secondaire de P.
Par conséquent $d(u',u)$=k_1. Pour chaque sommet $v \in S$, nous prenons ainsi $d(u',v)$=$d(u,v)$-k_1. En conséquence chaque sommet dans $B_1 \cap S$est à la distance au plus k_2-k_1 de u', par conséquence $B_1 \cap S \subseteq B_2$.
Rappelons que u' se trouve sur chaque chaîne de s à un sommet de T.puisque chaque sommet $v \in T$ est à une distance au plus $d(s,t)$ de s. Nous obtenons ce qui suit :
$d(v,u')+d(u',s)=d(v,s)\leq d(s,t) = d(s,u')+d(u',t)$. Ainsi chaque sommet dans T est à une distance $d(u',t)\leq k_2-k_1$ de u'. Par conséquence $T \subseteq B_2$, et$B_1 \cap B(s,d(s,t))$=$B_2 \cap B(s,d(s,t))$.

Lemme 3.7. [13] *Soit G un graphe en bloc P=$s_0,\ldots,s_k,\ldots\ldots,s_p$ une chaîne diamétrale dans G et soit $v \neq s_k$ un sommet de G. Si $2k \leq p$ et $s_0 \in B(v,k)$ alors $B(v,k) \subseteq B(s_k,k)$.*

Preuve. Puisque $s_0 \in B(v,k)$, nous constatons que $d(s_0,x)\leq 2k$ pour n'importe quel sommet $x \leq B(v,k)$. Par conséquent, $B(v,k) \subseteq B(s_0,2k)$. Notons que la boule $B(s_k,k)$ est identique à la boule élargie($B(s_0,2k),P,k$). Si $B(s_p,p)$=$V(G)$, alors par lemme 3.6 nous obtenons : $B(s_k,k) = B(s_{2k},2k)$. Nous avons ainsi prouvé que $B(v,k)\subseteq B(s,2k)$=$B(s_k,k)$. toutefois, puisque s_k est sur la plus courte chaîne de s_0 à s_{2K} (la chaîne s_0Ps_{2K}),s_k est un sommet d'articulation séparant les deux sommets s_0 et s_{2K}. Cela veut dire que s_k est sur la plus courte chaîne de v à l'un des deux sommets s_0 ou s_{2K}.
Puisque $d(v,s_0)\leq$d($s_{0=}$,s_{2K}), s_k doit être sur la plus courte chaîne de v à s_{2k}. Donc la distance $d(v,s_{2K})>$d(s_k ,s_{2K})=k. En conséquence, $B(s_k,k)\neq B(v,k)$. La combinaison de cette dernière avec les relations précédentes nous mène à $B(v,k)\subset$B(s_k ,k).

Lemme 3.8. [13] *Soit G un graphe en bloc $P = s,\ldots\ldots\ldots\ldots\ldots,t$ une chaîne diamétrale dans G, si F est un broadcast clairsemé dans G et $\mid V_F \mid>1$ alors les deux Sommets t et s sont dominés par deux sommet distinct dans P.*

Preuve. soit s' un sommet qui domine s. Puisque $\mid V_F \mid$>1 et F optimal, le poids de chaque dominateur doit être inferieur $rad(G)$. On a aussi $F(s')\leq$rad(G)-1=$\lceil d(s,t)/2\rceil$-1 par Lemme3.3. La distance $d(s,t)$ est plus grande que 2($\lceil d(s,t)/2\rceil$-1), ainsi s' ne peut pas dominer les deux sommets s et t. Nous avons par Lemme 3.7, à moins que s' soit dans P, là il existe un sommet x tels que $B\ (s',\ F(s'\ ))\subset B(x,F(s'\ ))$.Cependant, dans ce cas, il doit exister un broadcast optimal non efficace g, tel que$|V_g|$=$|V_F|$. Ceci fait de g un broadcast clairsemé sur G, et g doit être efficace. C'est une contradiction, et par conséquent (s' doit être dans P. La même preuve est applicable pour t.

Lemme 3.9. [13] *Soit G un graphe en bloc et C un noyau de G. Si F est un broadcast clairsemé dans G tel que$|\ V_F\ |> 1$, alors chaque sommet dans C est dominé par un dominateur de C.*

Preuve. soit P=s_0,s_1,........s_p une chaîne diamétrale tel que $C = C(P)$ notons que : si chaque sommet sur P est dominé par un sommet de C, chaque sommet sur $C\backslash P$ est exactement dominé par un sommet de C, par lemme 3.5. Par lemme 3.8, les sommets s_0 et s_p sont dominés par des sommets dont $P \subseteq C$. Soient (s'_0) et (s'_p) deux sommets qui dominent respectivement les sommets s_0 et s_p, et s_i,$s_{(i+1)}$,...... s_j l'ensemble des sommets qui sont dans P et non dominés par (s'_0) ou (s'_p), et soit un sommet x qui domine s_d avec $s_d \in s_iPs_j$. Puisque (s'_0) et (s'_p)sont séparés par s_i ou s_j, les dominateurs de sommets de s_iPs_j doivent être compris entre (s'_0) et (s'_p) (dans G_F). Par conséquent, le degré de x dans G_F doit être égal exactement à 2. Soient x_l et $x_r$deux sommets dans G_F se trouvant respectivement à gauche et à droite du sommet x, on note que C forme l'union de quelques cliques maximales. Par définition de graphe en bloc, chaque composante de $G - C$ est adjacente exactement à un seul sommet de C. Ainsi, si x n'est pas dans C, alors il existe un sommet $u \in C$ qui sépare x de C dans le graphe G. Cependant, la plus courte chaîne de x à x_l etx_r contient le sommet u. Par le lemme 3.2 , ceci implique que u et x doivent être le même sommet, qui est une contradiction.
Par conséquent, si x domine un sommet dans P, alors x doit être dans C.

Lemme 3.10. [13] *Soient G un graphe en bloc et C un noyau de G. Si F est un broadcast clairsemé dans G tel que$|V_F|$ >1, alors$V_F \subseteq C$.*

Preuve. On rappelle que le fait que F est clairsemé, il est efficace et G_F est une chaîne. Soit $P = s_0$,s_1,.............s_p est une chaîne diamétrale dans G tel que $C = C(P)$. Soient deux sommets s_0 et s_p dominés respectivement par s'_0 et s'_p (Lemme 3.8), et (s'_0) et s'_p $\in P$, et par le lemme 3.9, le sommet entre s'_0 et (s'_p)dans G_F il est dans C. Par conséquent, si$V_F \nsubseteq C$, alors il existe un sommet $x \in V_F\backslash C$ tel que x adjacent à s'_0 ou à s'_p . Sans perte de généralité, soit x adjacent à s'_0 dans G_F. Et soit y adjacent à s'_0 dans G sur la chaîne s'_0 Ps_p. Nous avons deux possibilités :

$$d_G\ (x,y)< d_G\ (s'_0,x) \text{ ou } d_G\ (x,y)\geq d_G\ (s'_0\ ,x)$$

Dans le premier cas y sépare s'_0 et x dans G. Cependant, puisque y sépare dans G s'_0 et l'autre voisin de s'_0 dans G_F , nous savons par le lemme 3.2 que F n'est pas clairsemé, contradiction.
Dans le deuxième cas, y sépare x et s_p. En plus $d_G\ (s_p.\ y\ )$=$d_G\ (s_p.s'_0)$-1.
puisque $d_G\ (s_0\ ,\ s'_0\ )\leq F(\ s'_0\ )$ et F est efficace, avec $F(x)$ >0, nous avons
$d_G\ (s'_0\ ,x)$=$F(s'_0)$+$F(x)$+1> $d_G\ (s'_0\ ,s_0\ )$+1. Ceci mène à la contradiction suivante :

d_G (s_p ,x)=d_G (s_p , y)+d_G (x ,y)> d_G (s_p ,(s'_0)+d_G (s'_0 ,s_0) >diam(G). En conséquence, on ne peut pas avoir un sommet x dans $G_F \backslash C$ si F est clairsemé.

Lemme 3.11. [13] *Soit* $P = s_0, \ldots, s_k, \ldots \ldots, s_p$ *une chaîne diamétrale dans un graphe en bloc* G*, et* F *est un broadcast clairsemé dans* G *tel que* $|V_F|$ *>1, et supposons que tous les sommets dans* $B(s_k\ ,k)$ *sont dominés par le même dominateur dans* V_F *pour* $k \leq p/2$ *.Si tous les deux rapports suivants sont vrais alors il existe un broadcast optimal* F' *sur* G *tels que* $F'(s_k) = k$*, et si l'un ou l'autre est faux, donc tous les sommets dans* $B(s_{(k+1)}\ ,k+1)$ *doivent avoir le même dominateur dans* V_F*.*

$$1.d(s_0\ ,x)=2k+1 \Longrightarrow x=s_{(2k+1)}$$
$$2.d(s_p\ ,x)=d(s_p\ ,s_{2k}\ )+1 \Longrightarrow x \in N(s_{2k})$$

Preuve. Soit s_i le sommet qui domine $B(s_k\ ,k)$ du broadcast clairsemé F,$F(s_i\ )=i$, par conséquent i≥k. Par le lemme 3.5, si s_{2K} et $s_{(2K+1)}$ ont un voisin en commun, alors ils doivent avoir le même dominateur, qui implique $i > k$ depuis $s_{(2K+1)} \not\subseteq B(s_k,k)$.
Supposons que (1) est faux c-à-d, il y a un sommet x tels que d(s_0,x)=2k+1 et x≠ $s_{(2k+1)}$, s,x est un voisinage de $s_{(2k+1)}$ et s_{2k}, alors i>k comme discuté ci-dessus,
par conséquent tous les sommets dans $B(s_{(2k+1)},k+1)$ sont dominés par s_i. Supposons que x est non adjacent à $s_{(2k+1)}$ et à s_{2k} , par conséquent s_{2k} sépare $s_{(2k+1)}$ et x avec $i = k$, mais le degré $s_{(i)}$dans G_F est plus grand que 1 ,contradiction car G_F est une chaîne.
Supposons que (2) est faux c.-à-d. il existe un sommet x telque $d(s_p,x)$=$d(s_p,s_{2k}\ )$+1 et $x \not\subseteq$N(s_{2k}).Soit $x' \in C(P)$ un sommet dominateur pour x dans F.
Nous avons $d(s_p,\ x')$+$d(x'$,x)≥ d(s_p,x)=d(s_p,s_{2k})+1≥ $d(s_p,x')$+$d(x',s_{2k}\ )$. Par conséquent s_{2k} est dominé par x'. Cependant, cela vaut également pour $s_{(2k+1)}$, ainsi i>k comme ci-dessus.
Supposons que (1) et (2) sont vrais si $i = k$, alors nous sommes faits, supposer encore $i > k$ soit $R = B(s_p,d(s_p,s_{2k}\ ))$et soit $B = B(s_i,i)$ noter cela d(s_{2K},$s_{(i+k)}$)=$i - k$ et ainsi $s_{2k} \in B'$=rétrécir(B,s_iPs_p,k) par lemme 3.6 ceci implique que $B' \cap R$=$B \cap R$ nous voyons maintenant que $B\backslash$R⊂ $B(s_k,k)$.
Nous pouvons construire un nouveau broadcast F' comme suit :

$$F' = \begin{cases} k\ si\ x = s_k \\ 0\ si\ x = s_i \\ i - k\ si\ x = s_{(i+k)} \\ F(x)\ sinon \end{cases}$$

Par les arguments ci-dessus F' domine les mêmes sommets que F
et clairement $C_{F'}(G)$=$C_F\ (G)$.
(Observons que s_{2k} est dominé par deux sommetss_k et $s_{(i-k)}$, impliqueF' n'est pas efficace).
Le résultat ci-dessus peut être employé pour construire un algorithme avide simple pour le calcul d'un broadcast dominant optimal dans un graphe en bloc :
Aussi longtemps que γ_b (G)<rad(G) il y aura toujours un broadcast clairsemé F tel que $|V_F|$ >1, et une chaîne diamétrale $s_0,s_1,\ldots\ldots..s_p$ tel que chaque sommet dans $B(s_i,i)$ sont dominés par le même dominateur. Cela crée la base pour le lemme 3.11. que un algorithme peut réitérer ainsi par les valeurs possibles de k jusqu'à les deux rapports (1) et (2) du

lemme 3.11 soient satisfaits, Quand ceci se produit, simplement ce que enlever les sommets dominés par $B(s_k,k)$ et répéter, pour chaque itération.

3.5 Un algorithme en temps linéaire pour calculer un broadcast optimal sur des graphes en bloc

Nous avons maintenant tous les résultats qui nous permettent de présenter notre algorithme

Théorème 3.2. [13] *Un broadcast dominant optimal peut être trouvé en temps linéaire dans les graphes en bloc.*

3.5.1 Explication de l'algorithme

Comme il a été expliqué précédemment, l'algorithme désigne un dominateur extrême gauche du graphe G, selon le lemme 3.11. Il enlève tous les sommets dominés par ce sommet, et trouve le dominateur extrême gauche dans un broadcast optimal efficace pour le sous-graphe restant.
Ceci est répété jusqu'à ce qu'il n'y ait plus de sommet, et chaque sommet est dominé ainsi. Par l'Observation 3.1 et le lemme 3.11, le broadcast résultant est optimal. L'algorithme commence par trouver un sommet t tels que $e(t) = \text{diam}\ (G\ )$, en utilisant Lemme 3.4.Une chaîne diamétrale entre t et un sommet s est utiliseé dans l'algorithme. Cependant, nous n'avons pas besoin de calculer chaque sommet dans la chaîne pour chaque itération. Nous pouvons calculer chaque sommet s_i sur la chaîne $s = s_0,s_1,\ldots\ldots.s_i,..s_p{=}t$ si nous avons besoin sans connaître le reste de la chaîne:Le sommet s_i est équivalent au sommet x tels que d_G (s_0 ,x)=i et d_G (x ,s_p)=p.

Remarque 3.4. nous n'avons pas besoin de changer le sommet t pour chaque itération non plus à chaque fois que le graphe G change.
Chaque sommet à la gauche d'un sommet s_j dans le premier diamétral sera enlevé. Cependant, depuis diam(G)=d(s ,t)=d(s ,s_j)+d(s_j ,t) et le sommet s se trouve à gauche de s_j, aucun sommet vers la droite ne peut être plus loin de s_j que t, et par conséquent e_G (t)=diam(G_0) par Lemme3.4.

3.5.2 Algorithme Broadcast Graphe Bloc

Algorithme Broadcast Graphe Bloc
Entrée Un graphe bloc $G = (V,E)$
Sortie un broadcast optimal sur G
Pour chaque v dans V **faire**
$F(v) \leftarrow 0$
Finpour
$x \leftarrow$un sommet quelconque dans V
$t \leftarrow$un sommet à distance maximum de x
Pour $i \leftarrow 0$ jusque à $|V|$ -1 **faire**
$T[i] \leftarrow$l'ensemble des sommet à distance i de t
Pour $v \in T[i]$ **faire**
D[v] $\leftarrow$i
Finpour
Finpour
Tant que $V \neq \emptyset$ **faire**
$s \leftarrow$un sommet à distance maximum de t dans G
$S[0] \leftarrow \{s\}$
$S[1] \leftarrow N(s)$
$S[2] \leftarrow N(S[1])\backslash\{s\}$
$k \leftarrow 0$
Tant que $|S[2k+2]| > 0$ **faire**
$k \leftarrow k+1$
$S[2k+1] \leftarrow$ Les sommets à distance $2k+1$ de s
$S[2k+2] \leftarrow$ Les sommets à distance $2k+2$ de s
s_{2k}Un sommet dans $S[2k]$ tel que $D[s_{2k}]$=$D[s]-2k$
Si $|S[2k+1]|$ >1 **alors continue**
Si $T[D[s]-2k+1] \subset N(s_{2k})$ **alors sortie**
Fin tant que
$v \leftarrow$ Un sommet dans $S[k]$ tel que $D[v] = D[s] - k$
Si $k > 0$ ou $|S[2k+1]|$ >0 et $|$ $S[2k+2]$ $|$=0 **alors** $k \leftarrow k+1$
$F(v) \leftarrow k$
$V \leftarrow V\backslash B(v,F(v))$ $G \leftarrow G|V|$
Fin tant que
Return F

Cet algorithme de Broadcast domination optimal des graphes en Bloc permet aussi d'obtenir un broadcast dominant efficace.

3.5.3 Exemple

Dans ce tutoriel on utilisera l'exemple suivant

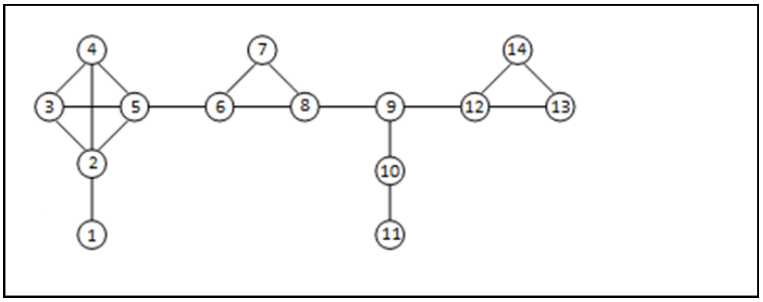


FIG. 3.2 – ***graphe G***

POSITION DU PROBLÈME

On souhaite trouver un broadcast F dominant sur le graphe G les instructions de l'algorithme de Broadcast Graphe Bloc qui permet de résoudre notre problème en temps linéaire sont les suivantes:
On prend le graphe bloc G
Soit V l'ensemble de sommets de graphe G

$$V \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11 \quad 12 \quad 13 \quad 14$$

On doit initialiser le broadcast $F(v)$ à zéro

v	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$F(v)$	0	0	0	0	0	0	0	0	0	0	0	0	0	0

On prend un sommet x quelconque de l'ensemble V, soit $x = 1$
t est le sommet le plus loin de x dans le graphe G Soit t=13
pour $i = 0$ jusqu'à 13, chaque $T[i]$ reçoit les sommets qui sont à distance i de t dans le graphe G, comme le montre le tableau ci-dessus.

$T[i]$	0	1	2	3	4	5	6	7	8	9	10	11	12	13
v	13	12,14	9	8,10	6,7,11	5	2,3,4	1	∅	∅	∅	∅	∅	∅

pour i=0 jusqu'à 13, et chaque $v \in T[i]$, $D[v]$ reçoit i, comme présenté dans le tableau suivant.

v	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$D(v)$	7	6	6	6	5	4	4	2	2	2	4	1	0	1

Comme il a été déclaré précédemment, $|V|$ =14

$V \neq \emptyset$(Donc on va entrer dans la première boucle while de l'algorithme).
s =1, (Le sommet le plus loin de t dans le graphe G)
$S[0]$={1},
$S[1]$={2}, (L'ensemble des voisinages du sommet 1 dans le graphe G).
$S[2]$={3,4,5},(L'ensemble des voisinages des sommets de $S[1]$ que le sommet 1 dans le graphe G).
k=0,
$|S[2]|$ >0 (On doit accéder à la deuxième boucle while de l'algorithme).
k=1
$S[3]$={6}, (L'ensemble des sommets à distance 3 de sommet 1 dans le graphe G).
$S[4]$={7,8}, (L'ensemble des sommets à distance 4 de sommet 1 dans le graphe G).
s_{2k}=5, (Un sommet dans l'ensemble $S[2]$ qui vérifié la condition suivante ($D[s_{2k}]$=$D[s]-2$), effectivement c'est le cas pour le sommet 5, ($D[5] = 5 = D[1] - 2 = 7 - 2 = 5$)).
$|S[3]|$=1, (la première condition (**if**) de la deuxième boucle while de l'algorithme n'est pas vérifier donc on passe au deuxième teste (2^{eme} **if**) certainement dans la même boucle.
$T[D[1] - 2 + 1]$=$T[7 - 1]$=$T[6]$={2,3,4} Comme il été déclaré dans le tableau précédent
$N(5)$={2,4,3,6}, (Les voisinages de sommet 5 dans le graphe G).
$T[6] \subset N((5)$, **donc on sorte de la boucle**.
v=2, (est un sommet dans l'ensemble $S[1]$ qui vérifie la condition suivante ($D[2] = D[1]-2$), est l'ensemble des sommets a distance 4 du sommet 1 dans le graphe G. Donc on doit pas incrémenter k.
$F(2)$ = 1,(on effectue la valeur k au sommet 2). B(2,1)={1,2,3,4,5},(c'est l'ensemble des sommets qui sont dominés par le sommet 2 (le centre de la boule B)).
$V = V \backslash B(2\ ,1)$={6,7,8,9,10,11,12,13,14}
Le reste de graphe G est présenté comme suit

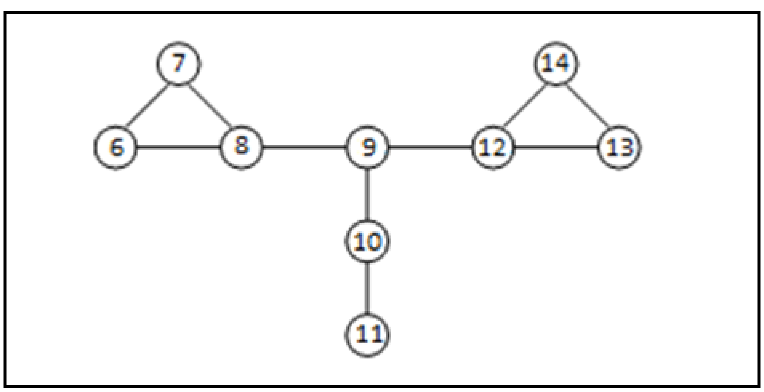


FIG. 3.3 – ***Le reste du graphe G***

$|V| \neq \emptyset$ (Donc on va rentrer dans la première boucle while de l'algorithme).
s=11, (Le sommet le plus loin de t dans le graphe G).
$S[0]$={11}, $S[1]$={10}, (L'ensemble des voisinages du sommet 11 dans le graphe G).

$S[2]$={9},(L'ensemble des voisinages des sommets de $S[1]$ que le sommet 11 dans le graphe G).
k=0,
$|S[2]|$ >0 (On doit accéder à la deuxième boucle while de l'algorithme).
k=1 $S[3]$={12,8}, (L'ensemble des sommets a distance 3 de sommet 11 dans le graphe G).
$S[4]$={13,14,6,7}, (L'ensemble des sommets a distance 4 de sommet 11 dans le graphe G)
s_{2k}=9, (Un sommet dans l'ensemble $S[2]$ qui vérifié la condition suivante ($D[s_{2k}] = D[s]-2$), effectivement c'est le cas pour le sommet 9, ($D[9] = 2 = D[11]-2 = 4-2 = 2$)).
$|S[3]|$=2, (la première condition (**if**) de la deuxième boucle while de l'algorithme est vérifier donc on continue $|S[3]|$ >0 (On doit accéder à la deuxième boucle while de l'algorithme).
k=2
$S[5] = \emptyset$, (L'ensemble des sommets à distance 5 de sommet 11 dans le graphe G).
$S[6] = \emptyset$, (L'ensemble des sommets à distance 6 de sommet 11 dans le graphe G).
s_{2k}=13, (Un sommet dans l'ensemble $S[4]$ qui vérifié la condition suivante ($D[s_{2k}] = D[s]-2k$), effectivement c'est le cas pour le sommet 13, ($D[13] = 0 = D[11]-2 = 4-4 = 0$)).
$|S[5]|$=0, (la première condition (**if**) de la deuxième boucle while de l'algorithme n'est pas vérifier donc on passe au deuxième teste (2^{eme} **if**) certainement dans la même boucle.
$T[D[11]-4+1] = T[1]$={12,14} Comme il été déclaré dans le tableau $N(5)$={2,4,3,6}, (Les voisinages de sommet 5 dans le graphe G).
$T[6] \subset N((5)$, **donc on sorte de la boucle**.
v=9, (Est un sommet dans l'ensemble $S[2]$ qui vérifié la condition suivante ($D[9] = D[11]-4$), Effectivement c'est le cas pour le sommet 9, $|S[6]|$=0, (Est l'ensemble des sommets a distance 6 de sommet 11 dans le graphe G).Et $|S[5]|$=0, (Est l'ensemble des sommets a distance 5 de sommet 11 dans le graphe G). Donc on doit pas incrémenter k.
$F(9)$ =2,(on effectue la valeur k au sommet 9).
$B(9,2)$={6,7,8,9,10,11,12,13,14}
$V = V \backslash B(9,2) = \emptyset$.
$V = \emptyset$ Donc on sorte de la boucle
A la fin de l'exécution de l'algorithme, quand on a déterminé un broadcast optimal on affiche ses valeurs avec les sommets correspondants. $F(v)$le broadcast optimal sur le graphe G est donner dans le tableau suivant

v	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$F(v)$	0	1	0	0	0	0	0	0	2	0	0	0	0	0

3.5.4 La complexité de l'Algorithme Broadcast Graphe Bloc

Le temps d'exécution de l'algorithme est $O(n+m)$:Nous commençons par initialiser les valeurs de F en $O(n)$, et peupler alors T et D dans $O(n+m)$ La boucle principale pourrait ressembler à un cou de bouteille en raison du temps nécessaire pour peupler des entrées de S dans chaque itération. Cependant, chaque entrée $S[i]$ peut être calculé à temps proportionnel à la cardinalité de $N(S[i-1])$. Par conséquent, puisque chaque sommet apparait dans une et seulement une seule entrée, tout le temps de fonctionnement à calculer tous les ensembles dans S, n'est en fait pas plus que $O(n+m)$. Par conséquent, le fonctionnement de tout l'algorithme entier est de $O(n+m)$.

Chapitre 4

Implementation

4.1 Introduction

Dans ce chapitre, nous implémentons l'algorithme étudié dans ce mémoire, avec le MATLAB, puis nous donnons une instance d'un exemple pratique dans les réseaux de communication.

4.2 Univers de travail :

Matlab (Matrix Laboratory)
Matlab dispose d'un langage de programmation basé essentiellement sur le calcul matriciel, avec des fonctionnalités mathématiques et graphiques étendues, ce qui correspond parfaitement à notre cas, puisque l'algorithme que nous présentons manipule essentiellement des matrices.

4.3 Exemples d'application :

Il s'agit d'analyser les résultats obtenus avec l'exécution de l'algorithme (BBG) sur un exemple de réseaux de communication en informatique.

4.3.1 Identification du problème

Un réseau informatique dispose de 10 terminaux où chaque terminal peut capter un signal qui est envoyé par ce terminal même, ou bien par un autre terminal voisin.

Ces terminaux sont reliés entre eux comme suit :
(1 $\longrightarrow$ 2,3,4),(2 $\longrightarrow$ 1,3,4),(3 $\longrightarrow$ 1,2,4),(4 $\longrightarrow$ 1,2,3,5), (5 $\longrightarrow$ 4,6,7),(6 $\longrightarrow$ 5,7), (7 $\longrightarrow$ 6,5),(8 $\longrightarrow$ 9,10,5),(9 $\longrightarrow$ 8,10),(10 $\longrightarrow$ 9,8)

Le problème consiste à déterminer le nombre minimum nécessaire de terminaux susceptibles d'être les plus appropriés à être activés, en minimisant l'intensité du signal global de sort que tous les terminaux soit alimentés.

4.3.2 Modélisation du problème :

On peut modéliser le problème précédent sous forme d'un graphe G=(V,E) où les sommets et les arrêts représentent respectivement les terminaux et le lien entre eux.
$\mid V \mid$=10 (nombre de terminaux).
$\mid E \mid$=14 (nombre de lien enter les terminaux)
Nous avons maintenant toutes les données qui nous permettent de construire le graphe G modélisant le problème.

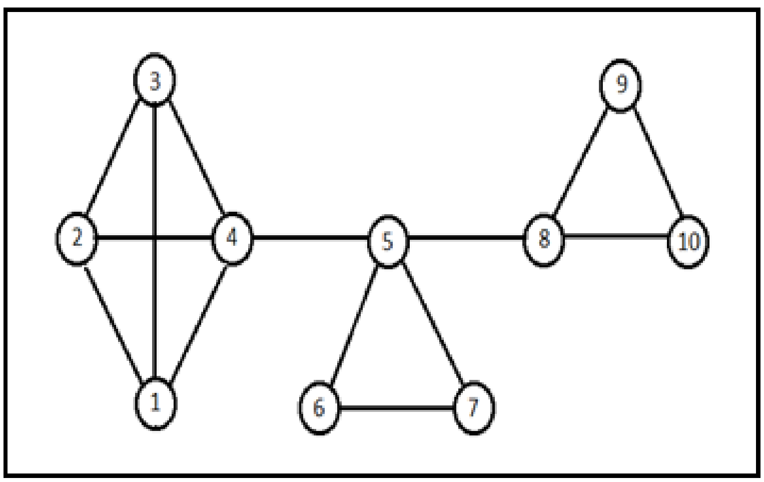


FIG. 4.1 – ***Le graphe G modélisant le problème d'application***

Notre problème repose sur la détermination d'un broadcast F dominant optimal sur le graphe G et pour sa résolution on fait appel à un langage de programmation **MATLAB** sur lequel on a établit le programme de résolution en utilisant l'algorithme broadcast bloc graphe (voir chapitre 3)

4.4 Implémentation de l'algorithme broadcast bloc graphe

Le programme Matlab ci-dessous est une implémentation de l'algorithme broadcast bloc graphe décrit dans le chapitre 3 et permettant l'obtention du broadcast dominant optimal

sur un graphe G en un temps linaire.
• Les trois paramètres d'entrée sont les suivants:
(1) n : Le nombre de sommets de graphe G $n = 10$

(2) M : La matrice d'adjacence de graphe G

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

(3) D : La matrice des distances

$$\begin{pmatrix}
0 & 1 & 1 & 1 & 2 & 3 & 3 & 3 & 4 & 4 \\
1 & 0 & 1 & 1 & 2 & 3 & 3 & 3 & 4 & 4 \\
1 & 1 & 0 & 1 & 2 & 3 & 3 & 3 & 4 & 4 \\
1 & 1 & 1 & 0 & 1 & 2 & 2 & 2 & 3 & 3 \\
2 & 2 & 2 & 1 & 0 & 1 & 1 & 1 & 2 & 2 \\
3 & 3 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 3 \\
3 & 3 & 3 & 2 & 1 & 1 & 0 & 2 & 3 & 3 \\
3 & 3 & 3 & 2 & 1 & 2 & 2 & 0 & 1 & 1 \\
4 & 4 & 4 & 3 & 2 & 3 & 3 & 1 & 0 & 1 \\
4 & 4 & 4 & 3 & 2 & 3 & 3 & 1 & 1 & 0
\end{pmatrix}$$

• À la sortie, on récupère :
Les valeurs de la fonction $F(v)$ (le broadcast optimal sur le graphe G

4.5 Les résultats d'exécution du programme

Après avoir introduit ces trois (03) paramètres dans le programme, comme l'illustre la figure 4.2

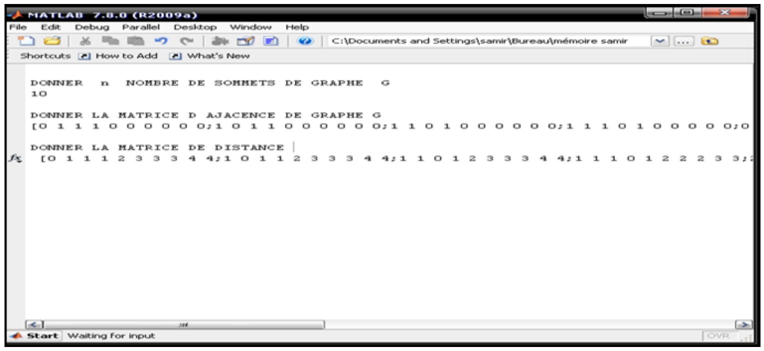


Fig. 4.2 – ***Les trois paramètres d'entrée***

Après l'exécution on a obtenu les résultats illustrés dans la figure 4.3.

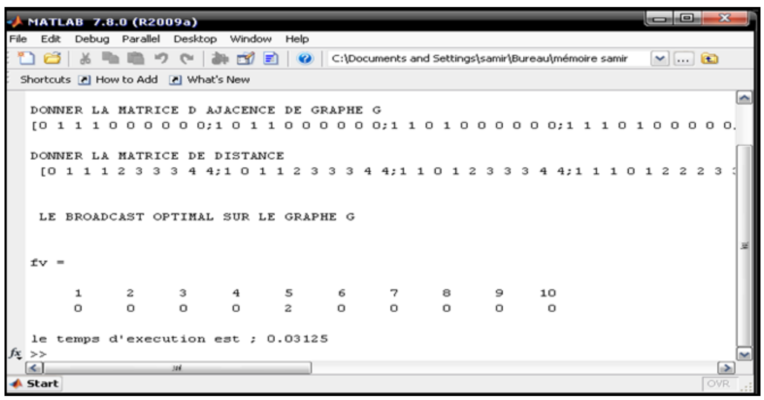


Fig. 4.3 – ***Les résultats d'exécution du programme***

Interprétation 4.1. *Avec notre programme et après avoir introduit les données de l'exemple d'application comme illustré dans la figure 4.2 on obtient la convergence de l'algorithme après (0,03125 sec) qui est affiché au sein de la figure 4.3,* $F(v)$ *est le broadcast dominant optimal sur le graphe* G *avec un seul dominateur qui est le sommet 5, avec* $F(5) = 2$*, donc si on émet un signal d'intensité 2 à partir du terminal 5 on est sûr d'atteindre tous les autres terminaux du réseau qui seront activés et connectés, de plus ce signal est optimal. Autrement dit, il est impossible d'avoir un signal d'intensité inférieure qui activera le réseau.*

Conclusion Générale

Du fait de sa généralisation du concept de la domination, le nombre de broadcast domination jouit d'une grande importance dans la modélisation et la résolution des problèmes d'optimisation combinatoire. Les études faites sur ce dernier ont permis de définir une classe très remarquable d'algorithme polynomiaux, généralement linéaires, pour la recherche de cet invariant, tels que les graphes d'intervalles, les arbres. . . .et très récemment les travaux sur les graphes 2 section des hypergraphes d'intervalles. L'algorithme étudié dans ce mémoire, serait d'une utilité plus grande s'il se généralisait à des classes plus grandes de graphes, tels que les graphes parfaits, les graphes de transitions. . . etc. ce qui et permettrait ainsi de contribuer à la résolution des problèmes de décision.

Bibliographie

[1] C. Berge. Théories de graphes et ses applications. Dunod, Paris, 1976.
[2] C. Berge ; Graphes et hypergraphes. Deuxième édition, Dunod, Paris, 1973.
[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, 169:55 -77, 2004.
[4] E.J. Cockayne, R.M. Dawes et S.T. Hedetniemi, Total domination in Graphs, Networks 10 (1980), 211-219, 1980 Wiley Periodicals, Inc., A Wiley Company.
[5] J. Dabney, B. C. Dean, and S. T. Hedetniemi. A linear-time algorithm for broadcast domination in a tree. Networks, 53(2):160169, 2009.
[6] J. E. Dunbar, D. J. Erwin, T. W. Haynes, S. M. Hedetniemi, and S. T. Hedet- Niemi. Broadcasts in graphs. 2004. Submitted to Disc. Appl. Math.
[7] E. Dunbar, D. J. Erwin, T. W. Haynes, S. M. Hedetniemi, and S. T. Hedetniemi. Broadcasts in graphs. Disc. Appl. Math., 154(1):5975, 2006.
[8] D. J. Erwin. Dominating broadcasts in graphs. Bull. Inst. Comb. Appl. 42:89- 105, 2004.
[9] M. R. Garey and D. S. Johnson. Computers and Intractability. Editions Freeman., 1979.
[10] M. C Golumbic, Algorithmic graph theory and perfect graphs, Academic Press, New York, 1980.
[11] M. Gondran , M. Minoux. Graphes et Algorithmes . Editions : Eyrolles 1995.
[12] P. Heggernes, Daniel. L. Optimal broadcast domination in polynomial time Department of Informatics, University of Bergen, N-5020 Bergen, Norway.
[13] P. Heggernes and Sigve H. S ther Broadcast Domination on block graphs in linear time CSR 2012, LNCS 7353: 177-188, Springer.
[14] P. Heggernes and D. Lokshtanov. Optimal broadcast domination of arbitrary graphs in polynomial time. Disc. Math., 306(24):3267-3280, 2006.
[15] G. Morel, Thèse doctorat Stabilité et coloration des graphes sans p5. Université de Grenoble, 2006.

Annexe1