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

MEMOIRE DE MASTER 2
en
MATHEMATIQUES

Option :
Méthodes et modèles de décision

Thème :

Algorithmes linéaires du nombre de broadcast domination de quelques classes de graphes

Présenté par

SADI Aris

Soutenu de vant le jury :

B.OUKACHA...... Président
M.AOUANE ...... Rapporteur
K.KASDI .... Examinateur

Année universitaire 2014/2015.

Remerciements

Je tiens tout d'abord à remercier mon encadreur, Mr Aouane, pour sa disponibilité et ses conseils qui m'ont été d'une grande aide afin de comprendre petit à petit le problème. Je remercie également Mr OUKACHA qui me fait l'honneur de présider le jury. Mes remerciements s'adressent aussi à Mr KASDI pour l'intérêt manifesté à examiner ce mémoire.

Table des matières

INTRODUCTION GENERALE

Qu'est-ce que la recherche opérationnelle ?

La Recherche Opérationnelle (RO) est au confluent de l'informatique, des mathématiques appliquées, de la gestion et du génie industriel. L'objet de cette discipline est de fournir des bases rationnelles à la prise de décisions, habituellement dans un but de contrôle ou d'optimisation (améliorer l'efficacité, diminuer les coûts, etc.). On utilisera par exemple des techniques de RO pour : " gérer les soins de santé dans les hôpitaux ; " organiser les services policiers ou ambulanciers ; " planifier l'utilisation et gérer la production d'énergie ; " planifier des systèmes de livraison ou de transport en commun ; " gérer la production, les stocks et la distribution de produits usinés ; " concevoir des systèmes de communication et des systèmes informatiques ; " établir des horaires de travail, de cours ou des calendriers sportifs ; " choisir des politiques économiques et financières. Qu'est-ce que la RO-AD ? Le Recherche Opérationnelle (RO) est la discipline des méthodes scientifiques utilisables pour élaborer de meilleures décisions. Elle permet de rationaliser, de simuler et d'optimiser l'architecture et le fonctionnement des systèmes de production ou d'organisation. La RO propose des modèles conceptuels pour analyser des situations complexes et permet aux décideurs de faire les choix les plus efficaces grâce à : " une meilleure compréhension des problèmes, " une vision complète des données, " la considération de toutes les solutions possibles, " des prédictions prudentes de résultats incluant une évaluation des risques, " des outils et des méthodes modernes d'aide à la décision. La RO apparaît comme une discipline carrefour associant les mathématiques, l'économie et l'informatique. Elle est par nature en prise directe sur l'industrie et joue un rôle-clé dans le maintien de la compétitivité. Les apports de la RO sont visibles tout autour de nous et dans les domaines les plus divers : de l'organisation des lignes de production de véhicules à la planification des missions spatiales, de l'optimisation des portefeuilles bancaires à l'aide au séquençage de l'ADN, voire dans la "vie de tous les jours" dans l'organisation des produits recyclables, l'organisation des ramassages scolaires ou la couverture satellite des téléphones portables. . .

La théorie des graphes, est connue comme étant un puissant outil de modélisation et de réesolution des problèmes rencontrés dans les différentes branches de la recherche opérationnelle, c'est dans celle-ci que se situe notre problème, qui n'est autre que la recherche du nombre de broadcast domination dans les graphes d'intervalles. 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 G = (V,E) dans 0,1,...,diam(G) telle que f(v)$\leq$ excentricité de v, pour tout v dans V .Si cette fonction est telle que pour tout v dans V , f(v) $\geq$ 0 ou il est la distance (f(u)) d'un autre sommet u de V 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 du broadcast domination

Chapitre 1

DEFINITIONS DE BASE ET NOTATIONS

1.1 Introduction :

La théorie des graphes est un outil puissant de modélisation et de résolution de problémes concrets. A l'origine, la théorie des graphes était présentée comme une curiosité mathématique ; Euler lors d'une de ses promenades nocturnes a voulu tracer un itinéraire circulaire dans la ville de Königsberg. Partant d'un point donné, il voulut visiter les sept ponts de cette ville (disposés selon le schéma ci-dessous) une seule fois seulement, puis retourner à son point de départ.

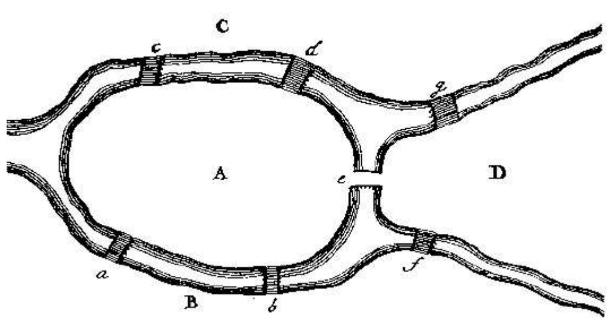


FIGURE 1.1 – les sept ponts de Konigsbeg

Les points A, B, C et D sont des rives.
Ensuite la théorie des graphes a été utilisée pour modéliser des circuits électriques (Kirch-

hoff), puis de nombreuses applications dans différents domaines tels : la chimie, la psychologie, etc.

1.2 Concepts de base

1.2.1 Défnitions générales

Les graphes sont des concepts mathématiques utilisés pour modéliser des relations binaires entre des objets d'un même ensemble. Ils sont fréquemment utilisés pour modéliser des systèmes qui se présentent sous la forme d'un réseau. Il existe deux types de graphes : les graphes orientés et les graphes non orientés.

1.2.2 Graphe orienté :

Un graphe orienté est un couple (X, U), où X est l'ensemble des sommets du graphe et U, l'ensemble de ses arcs.Xet Usont finis.

L'arc est une relation entre deux sommets , dotée d'une orientation :

Si $u = (x, y)$ est un arc de U, avec x , $y \in X$, la relation est orientée de x vers y.

Le graphe G est noté $G = (X, U)$.

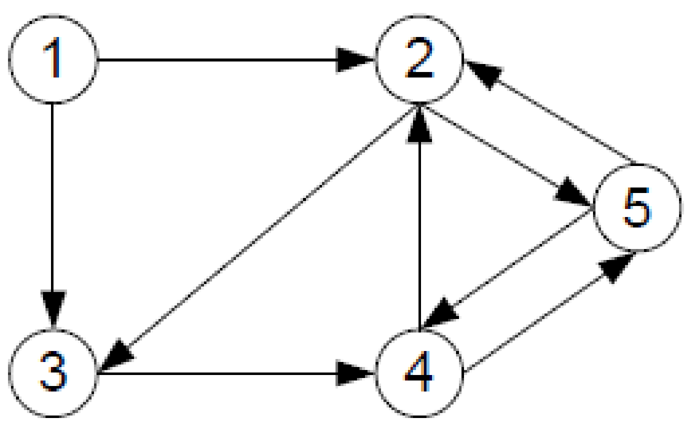


FIGURE 1.2 – Graphe orienté

1.2.3 Graphe non orienté :

Un graphe non orienté est défini par le couple (X, E), où X est l'ensemble des sommets du graphe et E, l'ensemble de ses arêtes .

Si x , y sont en relation, cette dernière est décrite par l'arête e=(xy).

Ici, le sens de la relation n'est pas invoqué.

Les figures 1.2 et 1.3 représentent, respectivement, un exemple d'un graphe orienté et un exemple d'un graphe non orienté.

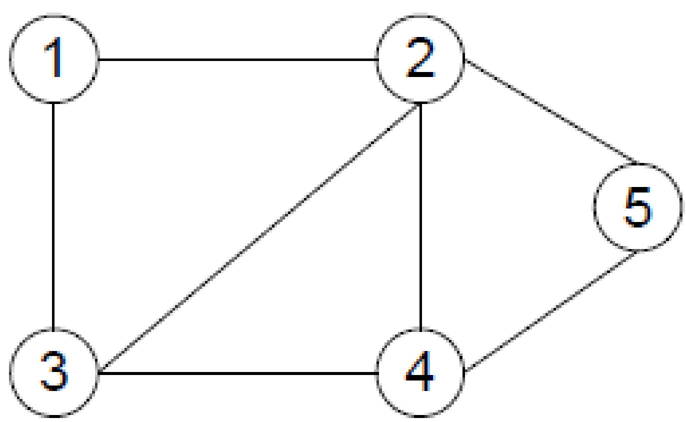


FIGURE 1.3 – Graphe non orienté

1.2.4 Arcs adjacents, arêtes adjacentes :

Deux arcs (resp.arêtes) sont adjacents (resp.adjacentes) en un sommet x, si x est une extrémité commune aux deux.

1.2.5 Arc incident à un sommet :

Un arc u est incident à un sommet x, si x est extrémité de u

.

u est incident-extérieur à x si $I(u) = x$.

u est incident-intérieur à x si $T(u) = x$.

I et T sont les applications extrémité initiale et terminale de e l'arc u.

1.2.6 Graphe symétrique :

$G = (X, U)$ est symétrique si : $\forall$ x,$y \in X$, $(x, y) \in U \Rightarrow (y, x) \in U$

1.2.7 Graphe anti-symétrique :

G est antisymétrique si :

$\forall$ x,$y \in X$, $(x, y) \in U \Rightarrow (y, x)$ n'est pas dans U.

1.2.8 Graphe complet

Un graphe $G = (X, U)$ est dit complet si $\forall$ $x, y \in X$, $\exists$ $u = (x, y) \in U$ ou $u = (y, x)$ $\in U$

Un graphe $G = (X, E)$ est dit complet si $\forall$ $x, y \in X$, $\exists$ e=(xy) $\in E$.

Si $\mid X \mid$=n, alors G est appelé clique d'ordre n ou une n-clique ; il est noté alors K_n

1.2.9 Graphe simple et graphe multiple

Un graphe simple est un graphe sans boucle ni arcs (arêtes) multiples.
Dans le cas contraire, c'est-à-dire, si des boucles ou des arcs (arêtes) multiples sont autorisés, on dira alors que le graphe est multiple. On définit ainsi, la multiplicité d'un graphe orienté par le nombre maximum d'arcs ayant la même extrémité initiale et la même extrémité terminale. Soit p ce nombre, on dit alors que G est un p-graphe.

$$p = max\{u \in U/I(u) = x \quad et \quad T(u) = y\}$$

1.2.10 Prédécesseurs, successeurs et voisins d'un sommet :

Un sommet $y \in X$ est prédécésseur d'un sommet $x \in X$ s'il existe, dans G, un arc u=(y, x)

Un sommet $y \in X$ est successeur d'un sommet $x \in X$ s'il existe, dans G, un arc u=(x, y)

$\Gamma^-(x)$=$\{y \in X$ /$\exists$ u=(y,x) $\in U$ $\}$. $\Gamma^-(x)$ est l'ensemble des prédecésseurs de x

$\Gamma^+(x)$=$\{y \in X$ /$\exists$ u=(x,y) $\in U$ $\}$. $\Gamma^+(x)$ est l'ensemble des successeurs de x

Un sommet $y \in X$ est voisin d'un sommet $x \in X$, si y est prédécesseur ou successeur de x.

On note par $N(x)$, l'ensemble des voisins de x. Ainsi $N(x)$=$\Gamma^+(x) \bigcup \Gamma^-(x)$.

1.2.11 excentricité, rayon

Le rayon d'un graphe est l'excentricité minimale de ses sommets, c'est-à-dire la plus petite distance à laquelle puisse se trouver un sommet de tous les autres. Le centre d'un graphe est formé de l'ensemble de ses sommets d'excentricité minimale.

L'excentricité maximale est appelée diamètre.

La distance entre deux sommets dans un graphe est définie par la longueur d'un plus court chemin entre ces deux sommets.

1.2.12 Degré d'un sommet

Soit G=(X, U) un graphe orienté. On a :

- Le demi-degré extérieur d'un sommet x est égal au nombre d'arcs ayant le sommet x comme extrémité initiale, on dit aussi le nombre d'arcs incidents extérieurs au sommet x. On le note :
$$d^+(x) = |\{u \in U / I(u) = x\}|$$
- Le demi-degré intérieur d'un sommet x est égal au nombre d'arcs ayant le sommet x comme extrémité terminale, on dit aussi le nombre d'arcs incidents intérieurs au sommet x. On le note :
$$d^-(x) = |\{u \in U / T(u) = x\}|$$

– Le degré d'un sommet x est le nombre d'arcs ayant x comme extrémité initiale ou terminale, on dit aussi le nombre d'arcs adjacents à x. On le note :

$$d(x) = d^+(x) + d^-(x)$$

1.3 Représentation matricielle d'un graphe :

Soit un graphe G =(X, U) contenant n sommets et m arcs ($|X| = n$ et $|U| = m$), on associera à G, trois types de matrices :

1.3.1 La matrice d'adjacence :

La matrice d'adjacence du graphe G=(X, U) est une matrice $n*n$, ses éléments prennent 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 :

– 1 signifie que les deux sommets sont reliés par un arc orienté.

– 0 signifie que les deux sommets ne sont pas reliés par un arc . orienté.

1.3.2 La matrice associée :

La matrice associée d'un graphe G=(X, U) est une matrice n*n, où chaque ligne et chaque colonne correspondent à un sommet du graphe. Un élément de la matrice associée indique le nombre d'arcs orientés dans le même sens reliant deux sommets.

1.3.3 La matrice d'incidence aux arcs :

La matrice d'incidence aux arcs d'un graphe G=(X, U) est une matrice n*m ; ses éléments prennent les valeurs 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 :

– +1 signifie que le sommet est l'extrémité initiale de l'arc.

– -1 signifie que le sommet est l'extrémité terminale de l'arc.

– 0 signifie qu'il n'existe pas de relation entre le sommet et l'arc.

1.3.4 chaine

Soit G=(X, U) un graphe. Une chaine joignant deux sommets x_0 et x_k dans G est une suite de sommets tels que deux sommets successifs sont reliés par une chaine. On la note (x_0,x_1,..,x_k). x_0 et x_k sont les extrémités de la chaine commune.

On la note : (x_0, x_1,x_2,...,x_k) et on dit que x_0 et x_k sont les extrémités de la chaine.

1.3.5 Chemin

C'est la suite de sommets et d'arcs (x_i,u_i,x_{i+1}....,x_k,u_k,x_{k+1},....x_j) tels que x_k = I(U_k) et x_{k+1} = T(u_k)

1.3.6 Cycle :

Un cycle est une chaine simple dont les extrémités coïncident. On le note (x_0, x_1,x_2,..., $x_k = x_0$).

1.3.7 Circuit :

Un circuit est un chemin dont les extrémités sont confondues. On le note par (x_0, x_1,x_2,..., $x_k = x_0$) .

1.3.8 connexité

Un graphe est connexe si $\forall$ x,y $\in X$,il existe une chaine entre x et y.

1.3.9 Composantes connexes

On appelle composante connexe un ensemble de sommets qui ont, deux à deux, la relation de connexité. De plus, tout sommet en dehors de la composante n'a pas de relation de connexité avec les sommets de cette composante.

1.3.10 Graphe connexe

Un graphe G=(X, U) est dit connexe si tous ses sommets ont deux à deux la relation de connexité ; autrement dit, si G contient une seule composante connexe.

Un graphe est connexe .

$$\Longleftrightarrow$$

Il possède une seule composante connexe.

1.3.11 Forte connexité

On définit la forte connexité dans un graphe par une relation entre deux sommets de la manière suivante :

Deux sommets x et y ont une relation de forte connexité

$$\Longleftrightarrow$$

Il existe un chemin de x à y et un chemin de y à x ; ou bien x=y.

1.3.12 Composantes fortement connexes

On appelle composante fortement connexe un ensemble de sommets qui, ont, deux à deux, la relation de forte connexité.De plus, tout sommet en dehors de la composante n'a pas de relation de forte connexité avec aucun élément de cette composante.

1.3.13 Graphe réduit

On appelle graphe réduit du graphe G=(X, U), le graphe G_r= (X_r, U_r) où :

- Les sommets sont représentés par les composantes fortement connexes C_i du graphe G.
- Un arc (C_i,C_j) $\in U_r$, s'il existe $(x, y) \in U$ tel que $x \in C_i$ et $y \in C_j$

1.3.14 Graphe fortement connexe

Un graphe G est dit fortement connexe si tous ses sommets ont, deux à deux, la relation de forte connexité ; autrement dit si G contient une seule composante fortement connexe.

Chaine eulèrienne

Une chaine est eulèrienne si elle utilise toutes les arêtes une et une seule fois

.

Cette définition est valable pour cycle , chemin et circuit.

1.3.15 Chaine hamiltonienne

Une chaine est hamiltonienne si elle utilise tous les sommets une et une seule fois .

Cette définition est valable pour cycle , chemin et circuit.

1.4 Arbres et arborescences :

Définition 1.1. Un arbre est un graphe connexe et sans cycle.

Proposition 1.4.1. *Soit n le nombre de sommets d'un graphe G=(X, E), et m le nombre de ses arcs alors*

- G est connexe ⇒ m ≥ n-1
- G est sans cycle ⇒ m ≤ n-1

Corollaire : G un arbre ⇒ $m = n - 1$.

Poids d'un arbre :

$$\text{P} : \text{E} \rightarrow \mathbb{R}$$
$$\text{e} \rightarrow \text{P(e)}$$

P(e) : est le poids de l'arête e.

- Si A arbre P(A)=$\sum_{e \in A}$P(e) ; P(A)=poids de l'arbre A
- A^* sera dit arbre de poids minimum si :

Définition 1.2. Une forêt est un graphe dont chaque composante connexe est un arbre ; c'est-à-dire un graphe sans cycle.

Définition 1.3. Un sommet s d'un graphe G est une racine (resp.une $anti - racine$ s'il existe un chemin joignant s à chaque sommet du graphe G (resp. Joignant chaque sommet de G à s) à l'exception du sommet lui-même.

Définition 1.4. Un graphe G=(X, U), avec $n = |X|$ ≥2 sommets. G est une arborescence de racine s si :

- G un arbre
- s est une racine de G

Remarque 1.4.1. Une arborescence est un arbre mais la réciproque est fausse.

– G est arbre
– s est une anti-racine de G

Remarque 1.1. Si on inverse le sens des arcs d'une arborescence, on obtient une anti arborescence.

1.4.1 Graphes d'intervalles

Un graphe d'intervalle est le graphe d'intersection d'un ensemble d'intervalles de la droite réelle. Chaque sommet du graphe d'intervalle représente un intervalle de l'ensemble, et une arête relie deux sommets lorsque les deux intervalles correspondants s'intersectent.

Chapitre 2

Théorie de la complexité, Domination et Broadcast.

2.1 Introduction

Dans ce chapitre nous donnons un bref aperçu de la théorie de la complexité des algorithmes en insistant sur les notions fondamentales de classes de problèmes P (polynomial) et NP (Non-déterministe Polynomial). Et puis nous présentons, quelques définitions de la dominance.En suite nous donnons quelques résultats concernant le problème du broadcast dominant.

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 intrinsèque des problèmes algorithmiques, et qui 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 concrêt ; 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 "plus difficiles".

2.2.1 Concepts de base

Un problème est une question générale possédant des paramètres dont la valeur n'est pas connue.

Une instance d'un problème est obtenue en affectant une valeur à chacun de ses paramètres.

La taille d'une instance désigne généralement la quantité de cases mémoires nécessaires pour décrire les paramètres.

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 onc 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 notion de "grand O", puis les classes de complexité P et NP qui en découlent.

2.2.2 Problèmes de décision et d'optimisation

Définition 2.2.1. Un problème de décision est un problème auquel la réponse est un oui ou un non.

Définition 2.2.2. Un problème d'optimisation consiste à déterminer la meilleure solution parmi toutes les solutions réalisables.

2.2.3 Complexité en temps et en espace

La complexité temporelle d'un algorithme correspond au nombre d'instructions élémentaires (opérations arithmétiques, comparaisons, affectations...) effectuées au cours de son exécution.

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 du 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,on définit 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 ne 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 que :

Définition 2.2.3. Notation grand O

$f(n) = O(g(n))$ (" $f(n)$ *est en grand* $O(g(n))$") *quand* $N \to \infty$*si et seulement si* $\exists M > 0,\ n_0 \in \mathbb{R}$ *tel que* $\forall n \geq n_0\ |f(n)| \leq M|g(n)|$

intuitivement, ceci signifie qu'à partir de n_0 et à un facteur constant près, F ne croît pas plus rapidement que g.

Exemples 2.2.2. soit $f(n) = 6n^4 - 2n^3 + 5$. Choisissons $n_0 = 1$:

$$|6n^4 - 2n^3 + 5| \leq 6n^4 - 2n^3 + 5$$

$$\leq 6n^4 + 2n^4 + 5n^4$$
$$= 13n^4$$

$$= 13|n^4|$$

Ainsi, en prenant $M = 13$, on a $f(n) = O(n^4)$. Autrement dit, à un facteur constant près, $f(n)$ 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 P et NP

Définition 2.2.4. Un problème de décision p est dans la classe P si, pour chacune de ses instances, dont la taille est notée 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.

Exemples 2.2.3. Considérons la version "décision" du problème du stable : étant donné un graphe. 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).Posons le problème suivant : 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 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 $P \subset NP$.[1]

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 2.2.5. 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 2.2.6. 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 via 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.[7]

2.3 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

Pour des raison pratiques, nous noterons un graphe $G = (V, E)$ au lieu de $G = (V, U)$.

Un ensemble dominant d'un graphe $G = (V, E)$ est un ensemble $S \subseteq V$ de sommets tels que tout sommet de V est adjacent à,au moins, un sommet de S. Plus formellement, S vérifie :
$V \subseteq N(S)$ ou encore $V = N[S]$. Le nombre dominant $\gamma(G)$ 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.

Domination totale

La domination totale est l'une des principales variantes de la domination. Elle a été introduite par Cockayne, Dawes et Hedetniemi .

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 $\gamma_t(G) \geq \gamma(G)$.

2.4 Broadcast

Le broadcast dominant a été présenté par Erwin en 2002, et il est une généralisation du problème de domination standard, 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 optimal dominant cherche à minimiser la somme des puissances assignées aux sommets du graphe. Depuis la présentation de ce problème, la complexité informatique a été ouverte, et la croyance générale a été que, il pourrait être NP-dur. Dans cette partie, nous rappelons que le broadcast optimal dominant est réellement dans P , et nous donnons un algorithme en temps polynômial pour résoudre le problème sur les graphes quelconques, ainsi que sa complexité. Depuis l'introduction de ce problème , beaucoup des paramètres du graphe reliés par 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 une broadcast station placée dans cette ville ou bien dans une autre ville voisine. Erwin 2002 présenta le problème de broadcast optimal, qui est plus réaliste dans le sens où 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 cher à construire et utiliser. En conséquence, le problème du broadcast dominant optimal demande à évaluer la fonction de puissance F sur les sommets, telle que chaque sommet du 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 minimisée. Depuis, la plupart des problèmes dominants intéressants sont de classe NP-durs sur les graphes quelconques ; ceci a donné une certaine indication que la domination optimale d'un Broadcast pourrait également être NP-dur pour les graphes quelconques. Après ceci, en 2003, Blair et al. ont donné un algorithme en temps polynômial pour la recherche du nombre de broadcast dominant pour les graphes d'intervalles, et les graphes séries-parallèle . Puis, en 2006, on trouve un algorithme polynomial pour la détermination d'un broadcast dominant optimal pour un graphe quelconque.

Définition 2.4.1. Une fonction $F : V \to \{0, 1, ..., diam(G)\}$ est un broadcast sur G, si $\forall v \in$V$1 \leq F(v) \leq e(v)$.
Le coût d'un broadcast $C_F = \sum_{v \in V} F(v)$
L'ensemble de broadcast dominateurs définie par F, c'est l'ensemble $V_F = \{v \in V | F(v) \geq 1\}$, l'ensemble F dominé est $V_0 = V - V_F$.

Définition 2.4.2. un broadcast est dit dominant si pour chaque sommet $u \in V$, $\exists v \in V_F |\ d(u, v) \leq F(v)$

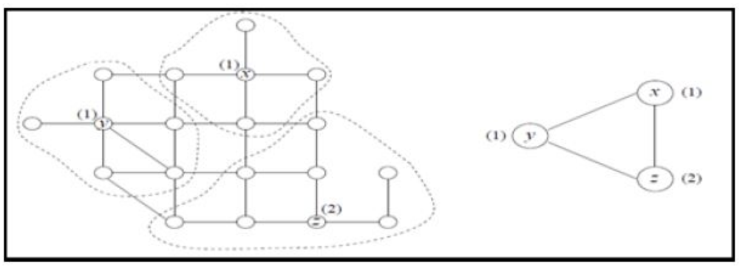
Définition 2.4.3. Le coût d'un broadcast dominant de G est noté C_F(V) également comme $C_F(G)$, et nous référons à lui le coût F dans G. Notons qu'il y a toujours un Broadcast dominant de coût $rad(G)$.

Définition 2.4.4. Le nombre de broadcast de 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*$\}$.

Définition 2.4.5. Si $C_F(G) = \gamma_b(G)$ alors nous appelons F broadcast 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'étudie du problème aux graphes connexes.

Définition 2.4.6. Un broadcast de domination est efficace si chaque sommet est dominé par exactement un seul sommet.

Définition 2.4.7. Soit F un 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 Heggernes et Lokshtanov ont prouvé ce qui suit pour un broadcast F dominant efficace dans G, si G_F a un sommet du degré plus de 2 alors il y a un broadcast F' dominant efficace sur G, tels que $|V_{F'} < V_F$ et $C_{F'}(G) = C_F(G)$.



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

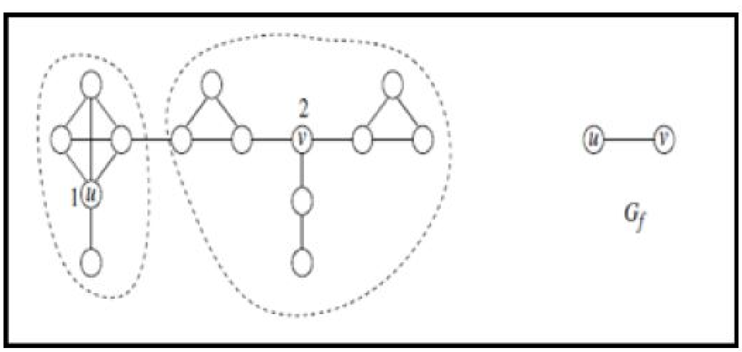


Figure.2.2 : Un graphe en blocs G avec un broadcast F clairsemé est montré du côté gauche, où les dominateurs du broadcast F sont u *et* v, $F(u) = 1$ et $F(v) = 2$. Les boules

$B(v, 1)$ et $B(v, 2)$ sont indiquées sur G et du côté droit, G_F est montré.

2.4.1 La structure du broadcast optimal dominant

Dunbar et al. prouvent que chaque graphe a un Broadcast dominant optimal qui est efficient. En particulier, le Lemme suivant est implicite de la preuve de ce résultat.

Lemme 2.4.1. *Dunbar et al. ont montré que pour tout 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 $V_{F'} < V_F$ et $C_{F'} = C_F$.*

Lemme 2.4.2. *Soit F un broadcast efficace dominant 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 $V_{F'} < V_F$ $C_{F'} = C_F$.*

Lemme 2.4.3. *Pour tout graphe G, il y a un broadcast dominant optimal efficace F sur G tel que le graphe de domination G_F est un chemin ou un cycle.*

Lemme 2.4.4. *Dans tout Broadcast dominant optimal F d'un graphe $G = (V, E)$, s'il y a deux voisins $v, w \in V$ tels que $F(v) > 0$ et $F(w) > 0, alors \Rightarrow F(v) = F(w)$.*

Chapitre 3

Recherche de broadcast dans quelques classes de graphes.

3.1 Introduction

Dans ce chapitre, nous allons voir deux algorithmes de recherche de broadcast optimal, l'un dans les arbres, et le second dans les graphes d'intervalles.

3.2 Les arbres

3.2.1 Introduction

Pour cette classe de graphes, nous verrons un algorithme de complexité $O(n^h)$ *qui recherche un broadcast efficace optimal contenant une racine* t *et de hauteur* h.

La racine de T *peut être désignée comme étant le sommet dont l'extrémité est minimale, aussi appelé le centre de* T.

3.2.2 L'algorithme

Algorithm *Tree Broadcast Domination (**TBD**)*

input : *A tree* T *rooted at a center vertex* v_r.

Out put : $\gamma_b(T)$.

for $i = h$ *to* h ***do***

$Sum(i) = 0;$

$InfinityCount(i) = 0;$

for *each child* v_k *of* v ***do***

if $cost_{vk}[i] = \infty$***then***

$InfinityCount(i) = InfinityCount(i) + 1;$

else

$Sum(i) = Sum(i) + cost_{vk}[i];$

for$i = h$ *to* $1do$

$cost_v[i] = Sum(i+1);$

$cost_v[0] = \infty;$

if$InfinityCount(0) \leq 1$ ***then***

for *each child* vk *of* v ***do***

if $cost_{vk}[0] = \infty$ ***then***

$cost_v[0] = \min\{cost_v[0],\ cost_{vk}[1] + Sum(0)\};$

else if$InfinityCount(0) = 0$ ***then***

$cost_v[0] = \min\{cost_v[0], cost_{vk}[1] + Sum(0) - cost_{vk}[0]\};$

for $i = 1$ *to* $h - 1$ ***do***

$BestChild(i) = \infty;$

if$InfinityCount(-i) \leq 1$ ***then***

for *each child* v_k *of* v ***do***

if $cost_{vk}[-i] = \infty$ ***then***

$BestChild(i) = \min\{BestChild(i), cost_{vk}[i+1] + Sum(-i)\}$ *;*

else if $InfinityCount(i) = 0$ ***then***

$BestChild(i) = \min\{BestChild(i), cost_{vk}[i+1] + Sum(-i) - cost_{vk}[-i]\};$

$cost_v[i] = \min\{(i + Sum(-i)), BestChild(i)\}$

$cost_v[h] = h + Sum(-h);$

$\gamma_b(G) = \min_{0 \leq i \leq h}\{cost_{vk}[i]\};$

3.3 recherche de la plus courte chaine

Algorithme *de recherche de la plus courte chaine dans un graphe d'intervalles.*

$R = 0;\ P_k = v_i;\ P = \{p\}$

while $P_k \notin N(v_j)$ ***faire***

Choisir P_{k+1} *tel qu'il soit un interval dans* $N(P_k)$ *avec la plus grande extrémité.*
$P \cup \{P_k + 1\}$; $k = k + 1$.
$P = P \cup \{v_j\}$.

Corollaire *Soit* f *une fonction broadcast dominant dans* G *avec* $v_i \in v_f$.
Si v_j *est entendu par* v_i, $(i < j)$, *alors tout* v_k *peut être entendu par* v_i $\forall i < k < j$

3.4 Graphes d'intervalles

3.4.1 Graphes parfaits

Soit $G = (X, E)$ *un graphe.On note par*

$\alpha(G)$, *le nombre de stabilité de* G

$\theta(G)$, *une partition minimale en cliques de* G

$\omega(G)$, *la taille d'une plus grande clique de* G

$\gamma(G)$, *le nombre chromatique de* G.

– *Le graphe* G *est dit* α*-parfait si* $\alpha(G)$=$\theta(G)$ *et* γ*-parfait si* $\gamma(G)$=$\omega(G)$

Proposition *: Un graphe* G *est* α*-parfait si et seulement si il est* $\gamma(G)$*-parfait*

Caractérisation *: Un graphe est parfait s'il est* α*-parfait ou* $\gamma(G)$*-parfait.*

3.4.2 Graphes d'intervalles

Les graphes d'intervalles sont une sous-classe de la classe des graphes parfaits. L'importance de cette classe, en plus de son utilité pratique (cas sociologiques, élucider un meurtre...), la partie théorique des graphes d'intervalles est importante. Le calcul d'un *certain nombre d'invariants connu NP-complet dans le cas général du graphe, se retrouve polynomial dans cette sous-classe comme dans celle des graphes parfaits.*

Un graphe d'intervalle est le graphe d'intersection d'un ensemble d'intervalles de la droite réelle. Chaque sommet du graphe d'intervalle représente un intervalle de l'ensemble, et une arête relie deux sommets lorsque les deux intervalles correspondants s'intersectent. Le calcul d'un stable maximum, de la taille d'une plus grande clique, de la partition minimale en clique ou encore le calcul du nombre chromatique d'un graphe deviennent polynomiaux dans cette classe.

Définition *: Le graphe* G=(X, E) *est dit d'intervalles propre si aucun intervalle n'est contenu dans l'autre.*

Autre caractérisation : G=(X, E) *est dit d'intervalles propre s'il ne contient pas de* $K_{1,3}$.

Exemple pratique d'utilisation des graphes d'intervalles

Dans une université , chaque étudiant doit se rendre une fois par jour à la bibliothèque . A la fin de la journée, on demande à chaque étudiant qui il a rencontré, et l'on se propose de déterminer leur ordre d'arrivée à la bibliothèque. On trace un graphe dont les sommets représentent les étudiants, deux sommets étaient joints si les étudiants se sont rencontrés à la bibliothèque. On obtient le graphe représentatif d'une famille d'intervalles : chaque intervalle représente l'intervalle de temps pendant lequel un étudiant est resté à la bibliothèque , et deux intervalles s'intersectent si et seulement si les étudiants s'y sont rencontrés.

Caractérisation des graphes d'intervalles

Définitions *:*

- *On appelle trou, un cycle* C_k, $k \geq 4$, *sans corde. Une corde est une arête joignant deux sommets non consécutifs d'un trou.*

– *Un graphe G est triangulé s'il n'admet pas de trou.*

Propriété *: Un graphe d'intervalles est triangulé*

.

– *Un graphe G=(X, E) est appelé un graphe de comparabilité s'il est possible, en orientant ses arêtes, d'en faire le graphe (X, U) d'une relation d'ordre , c'est-à-dire avec : $(x, y) \in U$, $(y, z) \in U \Rightarrow (x, z) \in U$ (transitivité)*

Propriété *: Si G est le graphe représentatif d'une famille d'intervalles , son complémentaire $\overline{G}$ est un graphe de comparabilité*

Corollaire *: G est d'intervalles $\Leftrightarrow$ G est triangulé et $\overline{G}$ de comparabilité*

Recherche du Broadcast optimal dans les graphes d'intervalles

L'algorithme de recherche du broadcast optimal, est divisé en deux étapes majeures. La première consiste à calculer les solutions radiales , et la deuxième consiste à introduire ces dernières dans l'algorithme, suivant ses intructions.

Solutions radiales *: décomposer le graphe en sous-graphes, et trouver au fur et à mesure son rayon et son centre . Autrement dit, pour chaque $G_{i,j}$, $(i < j)$, trouver les $MinRad(i, j)$ respectifs .*

Une fois cette étape terminée , introduire les $MinRad(i, j)$ et dérouler suivant les instructions, où $Opt(1, n)$= Broadcast optimal du graphe d'intervalles G.

pour $i = 1$ à $n - 1$ **_do_**
$Opt(i, i + 1) = \min Rad(i, i + 1)$;
pour $i = 2$ à $n - 1$ **_do_**
pour $j = 1$ à $n - 1$ **_do_**
$\gamma = \min_{k=j+1}^{j+i-2} \{Opt(j, k) + Opt(k + 1, j + i)\}$;
$Opt(j, j + i) = \min\{MinRad(j, j + i), \gamma\}$

Voici ci-dessous l'algorithme de recherche du Broadcast optimal dans les graphes d'intervalles

Introduire toutes les solutions radiales dans un tableau que l'on nommera $MinRad$

pour $i = 1$ *à* $n-1$ ***do***

$Opt(i, i+1) = \min Rad(i, i+1);$

pour $i = 2$ *à* $n-1$ ***do***

pour $j = 1$ *à* $n-1$ ***do***

$\gamma = \min_{k=j+1}^{j+i-2} \{Opt(j,k) + Opt(k+1, j+i)\};$

$Opt(j, j+i) = \min\{MinRad(j, j+i), \gamma\}$

Chapitre 4

Implémentation

4.1 Introduction

Dans ce chapitre, nous implémentons l'algorithme étudié dans ce mémoire, en utilisant le logiciel de programmation C Sharp.L'exemple pratique représente un mini réseau de communication.

4.2 C Sharp

Le C Sharp est un langage de programmation orienté objet, créé par la société Microsoft, et notamment un de ses employés, Anders Hejlsberg.

Il a été créé afin que la plate-forme Microsoft .NET soit dotée d'un langage permettant d'utiliser toutes ses capacités. Il est très proche du Java dont il reprend la syntaxe générale ainsi que les concepts (la syntaxe reste cependant relativement semblable à celle de langages tels que le C++ et le C). Un ajout notable au C Sharp est la possibilité de surcharge des opérateurs, inspirée du C++. Toutefois, l'implémentation de la redéfinition est plus proche de celle du Pascal Objet

4.3 Identification du problème

Un réseau informatique dispose de 7 terminaux ayant des signaux où chaque terminal peut capter un signal qui est envoyé par ce terminal-même, ou bien dans un autre terminal voisin.

Ces terminaux sont reliés entre eux comme suit :

1→ (2,3,4) ; 2→(1,3) ;3→(1,2,4 et 6)
4→(1,3,5 et 6) ;5→(3,4) ;6→(7) ;7→(6).

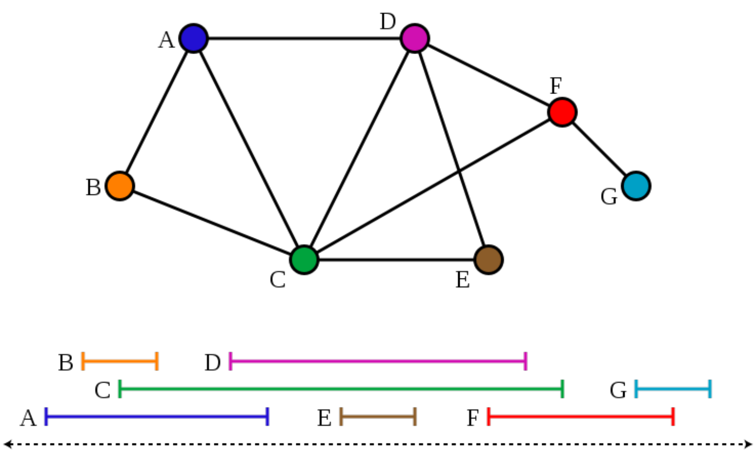
Le problème consiste à déterminer le nombre minimum nécessaire de terminaux susceptibles d'être les plus appropriés à être activés, en minimisant le signal d'intensité global de sorte que tous les terminaux soit alimentés.

4.4 Modélisation du problème

Nous pouvons modéliser le problème précédent sous forme d'un graphe $G = (V, E)$ avec les sommets et les arrêts représentant respectivement les terminaux et le lien entre eux.

| V | =7 (nombre de terminaux).
| E |=10 (nombre de liens enter les terminaux

Nous avons maintenant toutes les données pour construire le graphe corréspondant à notre problème :



Notre problème réside dans la détermination d'un broadcast dominant optimal F sur le graphe G

4.5 Implémentation de l'algorithme de recherche de broadcats optimal dans les graphes d'intervalles

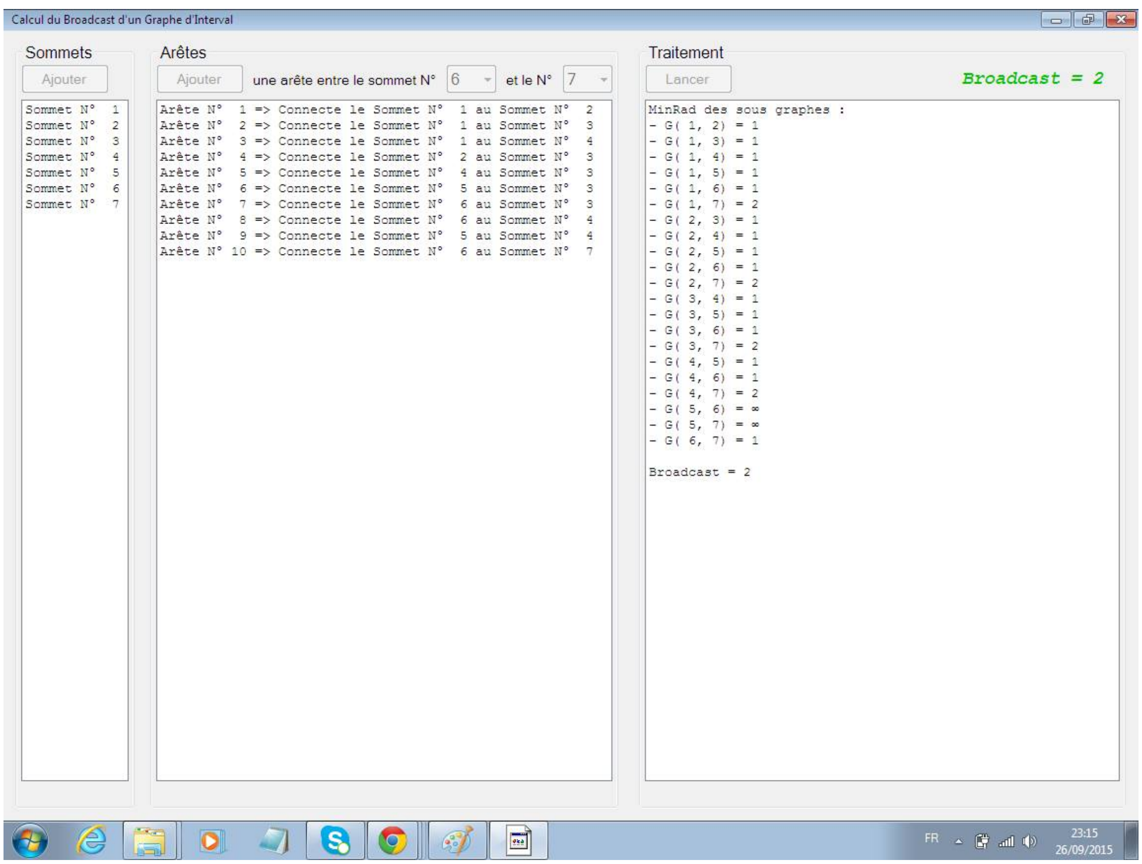
L'interface ci-dessous générée par C Sharp , est décomposée en trois parties :

La première partie permet à l'utilisateur d'introduire le graphe c-à-d ajouter le nombre de sommets

la deuxième toujours dans le but d'introduire les informations concernant le graphe , demande à l'utilisateur de séléctionner les sommets et leurs voisins directs

.

La troisième partie nous donne les $MinRad(i, j)$ respectifs , et le broadcast optimal



4.6 Interprétation

Après avoir introduit les données de notre exemple d'application , nous obtenons le résultat suivant :le broadcast dominant optimal sur le graphe est 2*. Le sommet 3 pondéré d'une intensité* 2 *réalise* $F(C) = 2$*.Autrement dit , si on émet un signal à partir du terminal* C *,nous sommes sûrs d'atteindre tous les autres terminaux du réseau qui seront activés.Il est impossible d'avoir un signal d'intensité inférieure qui activera le réseau.*

4.7 Autres résultats

4.7.1 Broadcast et transversal (couverture de sommets)

$G = (X, E)$ *un graphe . Un ensemble de sommets* T *et* X *est dit transversal si toute arête* e *de* E *admet, au moins une extrémité dans* T

Le problème classique étant, alors , de calculer un transversal de cardinalité minimale

Définissons une fonction broadcast

$f : X \rightarrow \{0, 1\}$

$x \rightarrow f(x)$

Alors, trouver un broadcast de valeur minimale , c'est trouver un transversal de cardinalité minimale.

4.7.2 Un algorithme linéaire pour la recherche du broadcast optimal dans les graphes d'intervalles

R-Y.Chang et S-L.Peng sont arrivés à obtenir un résultat interressant, il s'agit d'un algorithme linéaire pour la recherche du broadcast optimal dans les graphes d'intervalles.Cet algorithme a été présenté lors d'un workshop sur les mathématiques combinatoires [17] *, en janvier 2010.*

Ces deux chercheurs ont considéré les graphes d'intervalles propres , tout en se penchant sur les propriétés des cliques.

4.8 Conclusion générale

Du fait de la 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'algorithmes polynomiaux, généralement linéaires, tels que les graphes série-parallèls, 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 était généralisé à des classes plus grandes de graphes, tels que les graphes parfaits,. . . etc. Et permettrait ainsi de contribuer à la résoution de certains problèmes de décision.

Mon travail personnel se situe au niveau de la compréhension de l'algorithme calculant le broadcast dans les graphes d'intervalles et ainsi, de l'implémenter. Les résultats obtenus sont satisfaisants.

Bibliographie

[1] Gregory Morel, Thèse de doctorat Stabilité et coloration des graphes sans p 5, Université de Grenoble, 2006.

[2] *C. Berge, Théorie des graphes et ses applications. Dunod, Paris, 1967.*

[3] *C. Berge, Graphes et hypergraphes. Deuxième édition, Bordas, Paris, 1973.*

[4] *Michel Gondran , Michel Minoux, Graphes et Algorithmes , Edit . Eyrolles 1995.*

[5] *M.C Golumbic, Algorithmic graph theory and perfect graphs, Academic Press, New York, 1980.*

[6] *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.*

[7] *M.R. Garey, and D. S. Johnson, Computers and Intractability. Editions Freeman, 1979.*

[8] *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.*

[9] *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.*

[10] *D. J. Erwin, Dominating broadcasts in graphs. Bull. Inst. Comb. Appl., 42 :89- 105, 2004.*

[11] *Pinar.H , Daniel. L, Optimal broadcast domination in polynomial time Department of Informatics, University of Bergen, N-5020 Bergen, Norway.*

[12] *Pinar Heggernes and Sigve H. S Ther, Broadcast Domination on block graphs in linear time CSR 2012, LNCS 7353 : 177-188, Springer.*

[13] *J. Dabney, B. C. Dean, and S. T. Hedetniemi, A linear-time algorithm for broadcast domination in a tree. Networks, 53(2) :160169, 2009.*

[14] *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.

[15] *P. Heggernes and D. Lokshtanov, Optimal broadcast domination of arbitrary graphs in polynomial time. Disc. Math., 306(24) :3267-3280, 2006.*

[16] *N.Belharrat et Collectif .Théorie des Graphes , Edit : Pages bleues, 2004.*

[17] *R-Y.Chang and S-L Peng , A Linear-time Algorithm for Broadcast Domination Problem on Interval Graphs, The 27th Workshop on Combinatorial Mathematics and Computer Theory,2010.*