MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITÉ ABDELHAMID BEN BADIS DE MOSTAGANEM
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE
DÉPARTEMENT DE MATHÉMATIQUES ET INFORMATIQUE
FILIÈRE : MATHÉMATIQUES

UNIVERSITE
Abdelhamid Ibn Badis
MOSTAGANEM

MÉMOIRE DE FIN D'ÉTUDES

Pour l'Obtention du Diplôme de Master en Mathématiques délivré par

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

présenté par :

Eman EZZIANE

problème du P-médian dans une classe de graphe

soutenu publiquement juin 2019 devant le jury composé de :

Encadreur :	Hocine ABLAOUI	(université de Mostaganem,Algérie)
Président :	Maghnia HAMOU MAAMAR	MMA (université de Mostaganem,Algérie)
Examinatrice :	Nacera BAHNES	MMA (université de Mostaganem,Algérie)

Année Universitaire : 2018 / 2019

MASTER

Remerciements

Je remercie tout d'abord mon dieu le tout puissant de m'avoir donné le courage, la patience et la volenté durant toutes mes années d'études. Je tiens à exprimer toute ma reconnaissance à mon encadreur de mémoire, M. ABLOUI. Je la remercie de m'avoir encadré, orienté, aidé et conseillé.

J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté de me rencontrer et de répondre à mes questions durant mes recherches.

Je remercie mes très chers parents M. Farouk et Mme. Fariha , qui ont toujours été là pour moi. Je remercie mon frère M. Said , pour leur encouragements.

Enfin, je remercie mes amis Amel , Yasmine , Badra , kaouter qui ont toujours été là pour moi. Leur soutien inconditionnel et leurs encouragements ont été d'une grande aide.

À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Table des matières

Liste des tableaux

Table des figures

Liste des abréviations

GEN-CYCLE	:	Algorithme de génération d'un cycle.
GEN-ARBRE	:	Algorithme de génération d'un arbre.
GEN-CACTUS	:	Algorithme de génération d'un cactus.
AG	:	Algorithme de Goldman.

Introduction

Alfred Weber a été le premier à avoir étudié le problème de la localisation au début du 20^{eme} siècle : il s'est intéressé à la localisation dans le plan d'une usine par rapport à deux ressources et un magasin de vente .Il a proposé des méthodes rudimentaires de résolution du problème.
à la fin des années 70 , Kariv et Hakimi ont étudié le problème de la localisation dans un graphe. Ils ont montré que le problème est NP-dur et que la localisation se trouve toujours sur les sommets du graphes .
Notre travail consiste à étudier les problèmes particuliers du 1-médian (voire du 2-médian) dans des classes du graphes (les arbres , les cycles et les cactus).Nous présentons le problème du 1-médian paramétrique dans un cycle et un cactus . Nous avons divisé notre travail comme suit :
au premier chapitre nous donnons quelques rappels sur la théorie des graphes pour faciliter la compréhension de ce mémoire à un lecteur qui n'est pas familiarisé avec la théorie des graphes.
Le deuxième chapitre est consacré à une introduction du problème du p-médian dans un graphe : une formulation mathématique sous forme d'un programme linéaire est donné, une liste non exhaustive d'applications pratiques est dressée . et on termine ce chapitre en donnant le théorème, du à Kariv et Hakimi , sur la complexité algorithmique du problème du p-médian dans un arbre .
Le troisième chapitre traite des méthodes de résolution exactes et approchées du problème du p-médian dans un graphe.
Au quatrième chapitre nous nous intéressons aux cas particuliers du problème du 1- médian parametrique : nous donnons des méthodes efficaces pour déterminer le 1-médian parametrique dans un cycle et un cactus. Au dernier chapitre nous donnons des résultats numériques Nous terminons ce travail par une conclusion en indiquant quelques perspectives.

Chapitre 1

Notions sur les graphes

1 Historique

La théorie des graphes a été révélée au grand public lors d'une exposition en 1736 sur les curiosités mathématiques .Euler a exposé le problème des ponts de Konigsberg qui consiste à trouver un itineraire pour passer par tous les ponts une et une seule fois .Il montra que se problème n'a pas de solutions .

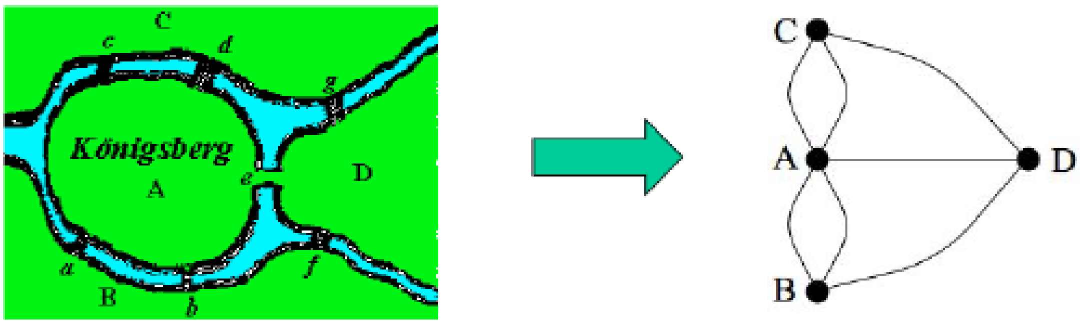


FIGURE 1.1 – le problème des ponts.

kirchhoff (1847) a utilisé des graphes pour étudier les circuits éléctriques . Cayley(1860),en étudiant le problème d'énumération a fait appel à la théorie des graphes

2 Graphes orientés

Un graphe fini $G = (V, E)$ est défini par la donnée :
- d'un ensemble des sommets $V = \{v_1, v_2, ..., v_n\}$ (en général fini) .
- d'un ensemble fini d'arcs $E = \{e_1, e_2, ..., e_m\}$.
- d'une application :

$$\begin{aligned} I : E &\longrightarrow V \times V. \\ e &\longrightarrow I(e) \text{ :extrémité initiale de } e. \end{aligned}$$

- et d'une application :

$$\begin{array}{rcl} T : E & \longrightarrow & V \times V. \\ e & \longrightarrow & T(e) \text{ :extrémité finale de } e. \end{array}$$

On appelle ordre d'un graphe le nombre de sommets n de ce graphe.
Si $e = (x, y)$ alors x et y sont dit adjacents et Si les extrémités x et y de e coïncident alors e est dit boucle .

2.1 chemin

Un chemin de x à y est une séquence d'arcs $(e_1, e_2, ...e_s)$ de E telle que :
$I(e_1) = x$, $T(e_s) = y$ et enfin, $T(e_k) = I(e_{(k+1)})$ $\forall k, k = 1, 2, ..., s_{i-1}$.
- Un chemin est dit élémentaire si en le parcourant, on passe par chaque sommet du graphe en au plus une fois .
-Un chemin est dit simple si en le parcourant, on passe par au plus une fois par chaque arc du graphe.
-Le chemin sera dit circuit si l'extrémité initiale du premier arc de la séquence coïncide avec l'extrémité terminale de la séquence.

2.2 Définition (graphe simple)

Un graphe est dit simple si entre toute paire de sommets a et b du graphe il existe au plus un arc qui relie a et b. Un graphe qui n'est pas simple est appelé multiple.

3 Graphes non orientés

Un graphe fini $G = (V, E)$ est défini par la donnée :
- d'un ensemble des sommets $V = \{v_1, v_2, ..., v_n\}$ (en général fini) .
- d'un ensemble fini d'arêtes $E = \{e_1, e_2, ..., e_m\}$.
- d'une application e qui est définie par :

$$\begin{array}{rcl} e : E & \longrightarrow & \{(a, b) / a \in V \text{et } b \in V\}. \\ u & \longrightarrow & e(u) \text{ :ensemble d' extrémités de } u. \end{array}$$

3.1 chaîne

Une chaîne reliant deux sommets x et y est une séquence d'arcs ou d'arêtes $(e_1, e_2, ..., e_s)$ de E telle que :une extrémité de e_1 est x et une extrémité de e_s est y et enfin e_k et e_{k+1} sont adjacents pour tout k variant de 1 à $s - 1$.

3.2 sous-graphe et Graphe partiel

Soit $G = (V, E)$ un graphe. Soient :$X \subset V$ et $U \subset E$:
- On appelle sous graphe construit sur X, le graphe G_x dont l'ensemble des sommets est X et l'ensemble des arêtes (arcs)est l'ensemble des arêtes (arcs)qui ont leurs deux extrémités dans X.
- On appelle graphe partiel construit sur U , le graphe G_u=(X, U).
- On appelle sous graphe partiel construit sur X et U , le graphe H dont l'ensemble des sommets est X et l'ensemble des arêtes (arcs) sont les arêtes de U qui ont leurs deux extrémités dans X.

3.3 Graphes connexes

Un graphe est dit connexe si entre deux sommets quelconques a et b, il existe une chaîne C qui les relie .

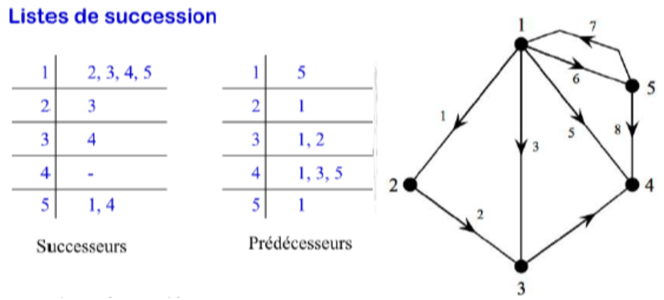
3.4 graphe complet

Un graphe simple G = (V, E) est dit complet si tous les sommets de G sont adjacents.

4 Représentation des graphes

Il existe plusieurs manières de représenter un graphe : Une représentation appropriée d'un graphe dépend du problème concret à étudier . Dans ce qui suit nous citons quelques représentations dles plus utilisées :

1. Liste de successions (ou prédécesseurs) :



1	2, 3, 4, 5
2	3
3	4
4	-
5	1, 4

1	5
2	1
3	1, 2
4	1, 3, 5
5	1

FIGURE 1.2 – Liste de successions (ou prédécesseurs).

2. Représentation par matrice :
   (a) Matrice d'adjacence :
   Considérons un graphe G = (V, E) comportant n sommets. la matrice adjacence, M, de G est définie par :
   $$m_{ij} = \begin{cases} 1 & si \quad (i, j) \in E \\ 0 & sinon \end{cases}$$
   Cette représentation peut être intéressante pour des graphes simples .
   (b) Matrice d'incidence :
   Considérons un graphe orienté sans boucles G = (V, E) comportant n sommets $x_1, x_2, ..., x_n$ et m arêtes $e_1, e_2, ..., e_m$.On appelle matrice d'incidence, M, de G la matrice de type n*m donnée par :
   $$m_{ij} = \begin{cases} 1 & si & x_i = I(e_j) \\ -1 & si & x_i = T(e_j) \\ 0 & si & sinon \end{cases}$$
   Pour un graphe non orienté sans boucles ,la matrice d'incidence est définie par :
   $$\begin{cases} 1 & si \quad x_i \text{ est une éxtremité de } e_j \\ 0 & sinon \end{cases}$$
   (c) Matrice associé :
   Dans le cas où le graphe est multiple , cette représentation est utilisée.

Chapitre 2

Introduction du problème du p-médian

1 Présentation du problème

Le problème du p-médian dans un graphe (supposé , sans perte du généralité, simple et sans boucles) consiste à déterminer l'installation de p centres de services (appelés médians) pour servir les n demandes situées sur les n sommets .
Cette installation est faite de façon à minimiser le coût total de transport .

2 Formulation mathématique du problème du p-médian

Soit G = (V, E) un graphe que l'on suppose, sans perte de généralité, simple et sans boucles.
A tout sommet v on associe un nombre non négatif $w(v)$ (appelé poids du sommet v).
A toute arête e est associée un nombre non négatif $c(e)$ (appelé cout de l'arête e).
On définit la distance $d(x, y)$ entre deux sommets x et y par la plus courte distance entre x et y dans G.
Le problème du p-médian peut être formulé par un programme linéaire (P) suivant :

$$\mathrm{Min(z)} = \sum_{i \in \mathrm{V}} \sum_{j \in \mathrm{E}} w_i d_{ij} x_{ij}. \tag{2.1}$$

$$\sum_{i \in \mathrm{V}} x_{ij} = 1 \qquad \forall j \in \mathrm{V}. \tag{2.2}$$

$$\sum_{i \in \mathrm{V}} x_{ii} = p. \tag{2.3}$$

$$x_{ij} \leq x_{ii} \qquad \forall i \in \mathrm{V}, j \in \mathrm{E}. \tag{2.4}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i \in \mathrm{V}, \forall j \in \mathrm{E}. \tag{2.5}$$

où
- d_{ij} : la plus courte distance entre les sommets i et j.
- w_i : le poids du sommet i .
on défini les variables par :

$$x_{ij} = \begin{cases} 1 & \text{si j est servi par le sommet i}^{1}. \\ 0 & \text{sinon.} \end{cases}$$

$$x_{ii} = \begin{cases} 1 & \text{si i est choisi comme médian.} \\ 0 & \text{sinon.} \end{cases}$$

2.1 Interprétation de contraintes

-(2.1) Représente la fonction objectif .
-(2.2) signifie que toutes les demandes sont servies par un et un seul serveur .
-(2.3) Le nombre des médians est fixe à p .
-(2.4) Une demande j est servi par i si un médian a été ouverte en i .
-(2.5) Nature binaire des variables x_{ij} .

3 Application pratiques

Le problème du p-médian a des applications pratiques dans tous les domaines .Que ce soit en informatique ,en biologie ,en chimie, en economie et dans beaucoups d'autre domaines comme le domaine militaire .Nous citons quelques applications dans ce qui suit :

3.1 Le problème des centres de communication dans les réseaux de telecommunication

L'information qui transite dans un réseau général doit passer par un certain nombre de commutateurs $S_1, ..., S_p$. Cette information peut être le contenu d'un message dans un réseau de telecommunication.
Nous imaginons un réseau de telecommunication comme un graphe $G = (V, E)$, où les sommets représentent des centres télèphoniques et les arêtes des connexions potentielles entre ces centres telephoniques. Chaque sommet v est associe avec un poids w qui designe le nombre de fils (lignes) nécessaires pour connecter le sommet v à un commutateur, et chaque arête (u, v) on associè le coût unitaire d'utilisation de cette connexion.
Le problème est de trouver l'installation qui minimise la somme totale de la longueur des lignes entre les sommets et les commutateurs.

3.2 Le problème des lieux d'ouverture de comptes bancaires

Le nombre de jours nécessaires pour encaisser un chéque dans une banque, située dans une ville i ,depend de la ville j où le chéque a été deposé. Par conséquent, une compagnie qui doit régler ses factures à plusieurs clients a intérêt à ouvrir des comptes dans plusieurs endroits stratégiques afin de maximiser ses fonds disponibles. Elle aimerait payer un client de la ville i à partir d'une banque d'une ville j qui maximise le nombre de jours nécessaires pour l'encaisser.

3.3 Le problème des proxys web

Dans ce cas, le réseau est supposé être une arborescence T avec un sommet racine r. L'ensemble des sommets de T est note par V(T). Les arcs de T sont diriges des feuilles vers la racine. Un sommet v de T fait une demande de service, comme la connexion a une adresse http. Cette demande se propage du sommet v vers la racine à travers l'arborescence. Cette propagation s'arrete quand un proxy[2] est rencontre. Le problème revient à

2. Un proxy est un programme servant d'intermédiaire pour accéder à un réseau (géneralement Internet) .

placer un nombre fixe de proxys dans les sommets de T de sorte que la distance parcourue par les demandes faites au niveau de chaque sommet soit minimale.

4 Complexité

Le problème du P-médian dans un graphe est un problème difficile à résoudre. kariv et Hakimi[9] ont montré que même si le graphe G vérifie :
- Le graphe est planaire.
- Le poids de chaque sommet est égale à 1.
- le coût de chaque arête est égale à 1.
- Le degré de chaque sommet est au maximum égale à 3.
Le problème du p-médian est NP-dure (NP-hard).

Chapitre 3

Méthodes de Résolution

1 Méthodes approchées

Une méthode approchée (où heuristique) , est une méthode qui permet l'obtention d'une solution réalisable, dans un certain sens, assez proche de l'optimum (ces méthodes peuvent être utilisées dans le cas où seule une solution approchée est désirée) d'un problème donné.
Plusieurs heuristiques ont été proposées comme la méthode qui a été développée par MARANZANA (1964) et celle de Teitz et Bart (1968). Ces méthodes sont efficaces pour des problèmes de taille raisonnable.

1.1 Heuristique de Teitz et Bart

L'heuristique de Teitz et Bart est une procédure itérative qui est basée sur la substitution des sommets : On démarre à partir d'une solution quelconque , on procède à la substitution d'un sommet médian par un autre sommet non médian. La solution est acceptée si le coût engendré est meilleur, sinon on rejette la solution. On continue le processus jusqu'à ce que l'on constate qu'aucune substitution n'est possible.
Soit l'algorithme de Teitz et Bart de façon formelle :

Algorithm 1: ALGORITHME DE TEITZ ET BART

Input: $G = (X, U)$,W,p,d.
Output: S

Initialisation
Choisir, aleatoirement l'ensemble $S = \{x_1, x_2, ..., x_p\}$ de p sommets de X
Calculer : $Q(S) = \sum_{k=1}^{n} W(x_k)d(x_k, S)$
Poser $S_0 = X - S$
poser : $S_{opt} \leftarrow \{x_{p+1}, x_{p+2}, ..., x_n\}$
while $S \neq S_{opt}$ **do**
    poser : $S_{opt} \leftarrow S$
    for $j \leftarrow 1$ ***to*** $|S_0|$ **do**
        for $i \leftarrow 1$ ***to*** $|S|$ **do**
            Calculer :
            $Q(S \cup \{x_j\} \backslash \{x_i\} = \sum_{k=1}^{n} W(x_k)d(x_k, (S \cup \{x_j\} \backslash \{x_i\}))$
            Calculer : $\Delta_{ij} = Q(S) - Q(S \cup \{x_j\} \backslash \{x_i\}$
        Determiner Δ_{ioj} par Δ_{ioj}=$max_{x_i} \in S(\Delta_{ij})$
        if $\Delta_{ioj} \leq 0$ **then**
            Marquer x_j d'un signe +
            else
                $S \leftarrow S \cup \{x_j\} \backslash \{x_i\}$
                $Q(S) \leftarrow Q(S \cup \{x_j\} \backslash \{x_i\})$
                Marquer x_j d'un signe -
return S

1.2 Heuristique du Maranzana

C'est une procédure itérative qui est basée sur le partitionnement de l'ensemble des sommets du graphe.
Elle a été développée par Marazana en (1964). L'algorithme débute par un choix, a priori arbitraire, de p sommets et la partition associée de l'ensemble des sommets restant du graphe. Ces p sommets sont, en fait, pris comme solutions de départ du problème. Dans chaque sous-ensemble composant la partition, on détermine le 1-médian, et l'ensemble de ces médians constituent la nouvelle solution approchée du problème du p-médian dans le graphe. On répète le processus jusqu'à ce que l'on ne puisse plus changer (et donc améliorer).
Soit l'algorithme de Maranzana de façon formelle :

Algorithm 2: ALGORITHME DE MARANZANA

Input: $G = (X, U)$,W,p,d.
Output: M
Initialisation
Choisir, aleatoirement l'ensemble $M = \{x_1, x_2, ..., x_p\}$ de p sommets de X
Poser $S = \{x_1, x_2, x_3, ..., x_p\}$
while $S \neq M$ **do**
  poser : $S \leftarrow M$
  for $i \leftarrow 1$ ***to*** p **do**
    Calculer : l'ensemble P_i par
    $p_i = \{x \in X \backslash d(x_i, x) \leq d(x_k, x)$ pour tout $k = 1, 2, ..., p$
    Determiner le median m_i de P_i par
    $m_i = argmin_{y_i \in p_i} \sum_{x \in p_i} W(x) d(y, x)$
  Poser : $M = \{m_1, m_2, ..., m_p\}$.

2 Méthode exacte

ce sont des méthodes d'énnumération implicite basées sur une exlporation des sommets par séparation et évaluation progressive (Branch and Bound). Elles consistent ,schématiquement, à décrire l'ensemble des solutions sous la forme d'une arborescence qu'on explore sans omettre aucun sommet .
l'évaluation par défault permet ,lorsque l'on possède une " bonne" solution réalisable , de parcourir implicitement toute une branche de l'arborescence sans en examiner chacun des noeuds .

2.1 Dualité lagrangienne

Notons par X l'ensemble des solutions de (P) qui vérifient les contraintes et

$$v(p) = min \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij}$$

Soit $\lambda = (\lambda_j)\ j = 1, ..., n$ le multiplicateur de Lagrange associé aux contraintes de type (2.3)(2.4)(2.5). Le problème lagrangien est donné par :

$$(\mathrm{L}_\lambda) \begin{cases} v(\mathrm{L}_\lambda) & = & min(\mathrm{L}(\lambda, x)) \\ x & \in & \mathrm{X} \end{cases} \tag{3.1}$$

où L est la fonction de Lagrange.

$$\mathrm{L}(\lambda, x) = \sum_{i=1}^{n} \sum_{j=1}^{n} (d_{ij} - \lambda_j) x_{ij} + \sum_{j=1}^{n} \lambda_j : x \in \mathrm{X} \tag{3.2}$$

Le dual lagrangien (D), est donné par :

$$(\mathrm{D}) : \begin{cases} v(\mathrm{D}) = v(\mathrm{L}_\lambda^\star) = max(v(\mathrm{L}_\lambda)) \\ \lambda \geq 0 \end{cases} \tag{3.3}$$

Pour $\lambda \geq 0$ (donné), le problème (L_λ) est résolu. Il suffit de décomposer ce dernier en n problème suivant :

$$\begin{cases} v(L_\lambda)_i = \sum_{j=1}^{n}(d_{ij} - \lambda_j)x_{ij} & \forall i = 1,\dots,n \\ \text{sous contrainte (2.4) et (2.5)} \end{cases} \tag{3.4}$$

La contrainte (2.3) est prise de manière implicite. Pour chaque i, le problème $(L_\lambda)_i$ est résolu par :

$$\delta_i = \sum_{j=1}^{n} min(0, d_{ij} - \lambda_j) \tag{3.5}$$

Soit $I^\star$ l'ensemble des indices qui correspondent aux p plus petites valeurs de δ_i :

$$I^\star = \{\text{ l'indice des p plus petites valeurs de } \delta_i\}. \tag{3.6}$$

La solution $x^\star_{ij}$ du problème $(L_\lambda)_i$ est donnée par :

$$x^\star_{ii} = \begin{cases} 1 & si \quad i \in I^\star \\ 0 & sinon \end{cases} \tag{3.7}$$

et pour i different de j

$$x^\star_{ij} = \begin{cases} 1 & si \quad i \in I^\star \text{et} d_{ij} - \lambda_j \leq 0. \\ 0 & sinon \end{cases} \tag{3.8}$$

D'où : $x^\star \in X$, $\delta_i = v((L_\lambda)_i)$ et

$$v(L_\lambda) = \sum_{i \in I^\star} \delta_i + \sum_{j=1}^{n} \lambda_j. \tag{3.9}$$

Initialisation ;
Determiner une valeur initiale de Zub(borne sup de (P) (on peut utiliser l'heuristique de Teitz et Bart ou autre);
poser :Zub=$-\infty$;
(Zlb :Meilleure borne inférieure trouvée durant la procédure.)
choisir une valeur initiale du multiplicateur Lagrangien $\lambda^{(0)}$ $\lambda_j^{(0)}$=$min_{ij}(d_{ij})pourtout$ j1,2,...,n
Choisir ϵ>0 et le nombre d'iterations maximum k_{max}>0;
Soit k=0;
tant que k<k_{max} **faire**
    Calculer : $I^{\star}_{(k)}$,($x^{\star}_{ij}$),$V(L_{(\lambda^k)})$;
    Donnés par les relations (3.6),(3.7),(3.8),(3.9) et (3.12);
    Actualisez Zlb en posant : Zlb=$max(Zlb, V(L_{(\lambda^k)}))$;
    Calculer : Z=$\sum_{j1}^{n} min_{i \in I^{\star}_{(k)}}(d_{ij})$;
    Mettre à jour la valeur de Zub par : Zub=$min(Z, Zub)$;
    if $((Zub - Zlb)$>$\epsilon)$ **then**
        Calculer le sous-gradient $\zeta^{(k)}$ par la relation(3.15);
        Calculer $\theta^{(k)}$ par la relation (3.16);
        Mettre à jour le multiplicateur Lagrangien par ;
        $\lambda_j^{(k1)}$=$max(0, \lambda_j^{(k)}$+$\theta^{(k)}\zeta_j^{(k)})$ pour tout j=$1, 2, ..., n$;
        k=k+1;
        else
            Soit k=k_max (la solution courante est optimale);
        end
    end
fin
return M;

Algorithme 3 : SOUS-GRADIENT

Chapitre 4

Cas particulier

1 Problème du 1- médian paramétrique dans un cycle

1.1 Définition d'un cycle

Un cycle (au sens de graphe) C_n est constitué d'un unique cycle élémentaire de longueur n ($n \geq 3$). C'est un graphe connexe (que l'on suppose non-orienté) d'ordre n ayant n arêtes. Il est 2-régulier, c'est-à-dire que chacun de ses sommets est de degré 2. On note :
- l'ensemble des arêtes du cycle $E = \{e, e = (v_i, v_{i+1}) / i = 1, 2, ..., n-1\}$.
-w_i : le poids du sommet $v_i (i = 1, 2, ..., n)$ et le poids du sommet v_1 est égal à $w_1 + t$ avec t un réel non négatif .
- et par $l(v_i, v_{i+1})$ le coût de l'arête (v_i, v_{i+1}).
la longueur du cycle est donnée par :

$$L = \sum_{i=1}^{n} l(v_i, v_{i+1})$$

la résolution du problème du 1-médian , pour toute valeur du paramétre t, peut être obtenu en $O(n)$. Le calcul de la valeur optimal de la fonction objectif nécessite $O(n)$ opérations élementaires
Notre but est de résoudre le problème du 1-médian paramétrique pour toute valeur de t et de calculer la valeur de la fonction objectif correspondante .
Notons, la valeur de la fonction objective relative au sommet v_i ($i = 1, 2, ..., n$) , par :

$$f_i(t) = \sum_{j=1}^{n} w_j d(v_i, v_j) + t d(v_i, v_1). \tag{4.1}$$

En utilisant cette notation, alors le problème du 1-médian paramétrique dans un cycle consiste à déterminer la fonction

$$F(t) = \min_{i=\overline{1,n}} f_i(t)$$

et les arguments correspondant , i.e , le sommet pour lequel F est optimal pour le réel donnée $t \geq 0$.
Nous remarquons que la fonction $F(t)$ est concave et est continue par morceau sur au plus n points de discontinuité et notons par : $PD = \{t_1, t_2, ..., t_n\}$ avec $t_1 = 0$ et $k \leq n$ l'ensemble des points de discontinuité .

Remarque 4.1 *on dit que v_i est optimale si et seulement si v_i est optimale pour $t \in [t_s, t_{s+1}]$. autrement dit , on dit que v_i est optimale pour $t > t_k$ si et seulement si v_i est optimale pour $t \geq t_k$*

Pour calculer F(t) de maniére éfficace on doit trier les n droites de la fonction linéaire $f_i(t)$ et calculer les différentes valeurs de $d^r(v_1, v_j)$ et $d^l(v_1, v_j)$ $\forall j = 1, ..., n$ (comme définis en 4.2 et 4.3)

1.2 Partie 1 : calcul de la matrice des plus courtes distances

Avant de décrire l'algorithme, nous allons introduire quelques notations. Soient v_i et v_j deux sommets quelconques de V(C) [1], alors on définit :

$$d^l(v_i, v_j) = \sum_{k=i}^{j-1} l(v_k, v_{k+1}). \tag{4.2}$$

$$d^r(v_i, v_j) = \sum_{k=j}^{i-1} l(v_k, v_{k+1}). \tag{4.3}$$

$d^l(v_i, v_j)$ et $d^r(v_i, v_j)$ correspondant aux longueurs des deux chemins, respectifs, de v_i à v_j en parcourant le cycle dans le sens des aiguilles d'une montre et dans le sens contraire de ces dernières .
La longueur du cycle L , est definie par : $\mathrm{L} = d^l(v_i, v_j) + d^r(v_i, v_j)$.
et le calcul de $d(v_i, v_j)$ est donné par :

$$d(v_i, v_j) = min(d^l(v_i, v_j), d^r(v_i, v_j)) \forall i = 1, 2, ..., n. \tag{4.4}$$

Soient :

$$\mathrm{L} = \{v_j \in \mathrm{V} | d^l(v_1, v_j) \leq d^r(v_1, v_j)\}$$

$$\mathrm{R} = \{v_j \in \mathrm{V} | d^r(v_1, v_j) < d^l(v_1, v_j)\}$$

ces deux ensembles L et R forment une partition de l'ensemble des sommets V(C) et s'écrivent sous la forme $\mathrm{L} = \{v_1, v_2, ..., v_s\}$ et $\mathrm{R} = \{v_{s+1}, v_{s+2}, ..., v_n\}$ s'ils sont non vides .

Remarque 4.2 *si on note $k_i = d(v_1, v_i)$ (pour $i = 1, 2, ..., n$) alors , il est facile de voir que :*

$$k_1 \leq k_2 \leq ... \leq k_{s-1} \leq k_s$$

et

$$k_n \leq k_{n-1} \leq ... \leq k_{s+1}$$

Ci-dessous nous donnons un algorithme pour calculer la matrice des plus courtes distances dans un cycle :

1. V(C) est l'ensemble des sommets du cycle C

Algorithm 4: ALGORITHME POUR DETERMINER LA MATRICE DES PLUS COURTES DISTANCES DANS UN CYCLE.

Input: C
Output: d

Calculer : $L = \sum_{i=1}^{n-1} l(v_i, v_{i+1})$
for $i \leftarrow 1$ ***to*** n **do**
    $p \leftarrow 0$
    Calculer le succsseur s de i
    $p \leftarrow p + C(i, s)$
    $d(i, s) \leftarrow min(p, L - p)$
    $d(i, s) \leftarrow d(s, i)$
    while $s \neq i$ **do**
        Calculer le succsseur k de s
        $p \leftarrow p + C(s, k)$
        $d_{i,k} \leftarrow min(p, L - p)$
        $d(i, k) \leftarrow d(k, i)$
        $s \leftarrow k$
return d

1.3 Partie 2 : calcul de la valeur de la fonction objectif d'un sommet du cycle

à tenir, de ce fait les droites peuvent être triées par rapport à leurs tengentes dans le temps linéaire en fusionnant deux ensembles ordonnés.ce tri, il est montré comment la fonction peut être obtenue .
par suite un algorithme est décrit qui calcule l'enveloppe inférieur dans l'intervalle $[0, \infty)$ de n droites de $y_i(x) = a_i x + b_i$ où $a_i < a_{i+1}$ $\forall i = 1, ..., n-1$.
tout simplemet, nous supposons qu'il n'y a pas de droites avec la même tengente a_i si c'est le cas, la droite avec l'ordonnée à l'origine la plus basse sera intéressante et les autres peuvent être supprimés . on définit d'abord ;

$$\mathrm{F}^j(x) = \min_{i=1,...,j} y_i(x)$$

pour $j \leq n$ notant que $\mathrm{F}^j(x)$ définit uniquement par rapport à ses points du discontinuités PD $= \{x_1 = 0, x_2, ..., x_n\}$ et correspond au tengentes k_l^j sur $[x_l, x_{l+1}]$ pour $l = 1, .., s-1$ et k_s^j sur $[x_l, \infty)$ .De plus, nous appelons une droite active dans l'intervalle $[x_l, x_{l+1}]$ si $y(x) = \mathrm{F}^j(x) \forall x \in [x_l, x_{l+1}]$. notant que l'enveloppe inférieure donné par $\mathrm{F}^n(x)$ l'algorithme proposé explicite le calcule $\mathrm{F}^j(x) \forall j = 1, ..., n$ au début ,on a $\mathrm{F}^1(x) = a_1 x + b_1$ (par définition)à fin de calculer $\mathrm{F}^{j+1}(x)$ car $\mathrm{F}^j(x)$ est déja calculer .
c'est important de trouver l'enveloppe inférieure dû à la droite $y_{i+1}(x) = a_{i+1}x + b_{i+1}$ et la fonction $\mathrm{F}^{j+1}(x)$,on tenir compte que la tengente et le point discontinuité $\mathrm{F}^j(x)$.
à fin de calculer l'intersection $s = (s_x, s_y)$ du $y_{i+1}(x)$ et la droite qui est active sur l'intervalle $[x_1, x_2]$ où x_1 et x_2 sont les deux premiers points de discontinuités du $\mathrm{F}^j(x)$. si $s_x \leq 0$ alors $\mathrm{F}^j(x) \leq y_{i+1}(x) \forall x \geq 0$ donc $\mathrm{F}^{j+1}(x) = \mathrm{F}^j(x)$ d'où on n' est pas intérèssee de recalculer la suite des $y_{i+1}(x)$.

sinon $s_x \in [x_1 = 0, x_2]$ alors s_x devient le nouveau point du discontinuité de $F^{j+1}(x)$, où la tengente sur l'intervalle $[0, s_x]$ est donnée par a_{j+1} . ensuite $F^{j+1}(x) = F^j(x) \forall x \in [s_x, \infty)$.à tenir $F^{j+1}(x) > F^j(x) \forall x \in [x_1 = 0, x_2]$ et on supprime la droite active sur cette intervalle . calculant maintenant l'intersection du point $\acute{s} = (\acute{s}_x, \acute{s}_y)$ du $y_{i+1}(x)$ et la droite qui est active sur intervalle $[x_2, x_3]$ où x_3 le point qui suit , on a donc $\acute{s}_x \geq x_2$ alors si $\acute{s}_x \in [x_2, x_3]$ donc $F^{j+1}(x) = F^j(x)$ d'où on n' est pas intérèssée de recalculer la suite des $y_{i+1}(x)$
cettte procédure doit être répété pour l'intervalle qui convient jusqu'à peut déterminer $y_{j+1} \leq F^j(x)$ pour tout $x \geq 0$
En raison du fait qu'après chaque calcul d'un point d'intersection soit une droite peut être supprimé où un nouveau point du discontinuité peut être ajouté et le calcul de $F_j(x)$ pour certains $j = 1, \ldots, n$ est terminé, il y a au plus $O(n)$ calculs d'une intersection point. Puisque chaque point du discontinuité peut être obtenu en temps constant, l'enveloppe inférieure peut également être calculé en temps linéaire. Ainsi, le problème paramétrique du 1-médian considéré dans un cycle peut être résolu en O (n) .

1.4 Algorithme de génération d'un cycle.

Pour pouvoir estimer notre algorithme, on génére de façon aléatoire des cycles .Cela consiste, au fait pour notre cas , à choisir aléatoirement n valeurs (qui correspondent aux n coûts des n arêtes). Ci-dessous nous donnons un algorithme de maniére formelle :

Algorithm 5: GEN-CYCLE.

Input: n ,k
Output: C

$C \leftarrow \infty$
for $i \leftarrow 1$ ***to*** n **do**
  for $j \leftarrow 1$ ***to*** n **do**
    if $i = j$ **then**
      $C(i, j) \leftarrow 0$
for $i \leftarrow 1$ ***to*** $n - 1$ **do**
  Choisir : aleatoirement la valeur de $C(i, i+1)$ dans l'intervalle $[1, k]$
  $C(i+1, i) \leftarrow C(i, i+1)$
Choisir : aleatoirement la valeur de $C(1, n)$ dans l'intervalle $[1, k]$

2 problème du 1- médian dans un arbre .

2.1 introduction

Si le problème du p-médian dans un graphe quelconque est un problème NP-dur, il devient théoriquement facile si le graphe se réduit à un arbre. En 1979 kariv et Hakimi ont donné un algorithme polynomial.
En 1971, Goldman a proposé un algorithme trés efficace pour la résolution du 1-médian dans un arbre.

2.2 Définition : D'un arbre.

Un arbre T est un graphe qui est connexe et sans cycle .

2.3 problème du 1-médian dans un arbre.

Avant de donner la procédure qui permet la détermination du médian dans un arbre, nous donnons une extension des lemmes de Goldman.

Dans tout ce qui suit , on notera indifférement le poids d'un sommet u par $w(u)$ ou w_u et on notera par $f(u)$ la valeur de l'objectif du sommet u quand ce dernier est choisi comme médian .

Lemme 4.1 *Soit G=(V,E,w,l) . S'il existe un sommet $v^\star$ Tel que*

$$2w_{v^\star} \geq \sum_{v\in \mathrm{V}} w_v.$$

alors $v^\star$ est un 1-médian de G.

Preuve 4.1 *On démontre que $f(v^\star) \leq f(v)$ pour tout $v \in \mathrm{V}$.*
Soit $v \in \mathrm{V}$. L'inégalité triangulaire implique $d(u,v^\star) - d(u,v) \leq d(v,v^\star)$ pour tout $u \in \mathrm{V}$.ainsi on peut conclure que

$$\begin{aligned}
f(v^\star) - f(v) &= \sum_{u\in \mathrm{V}} w_u d(u,v^\star) - \sum_{u\in \mathrm{V}} w_u d(u,v) \\
&= \sum_{u\in \mathrm{V}} w_u (d(u,v^\star) - d(u,v)) \\
&= \sum_{u\neq v^\star} w_u (d(u,v^\star) - d(u,v)) - w_{v^\star} d(v^{star},v) \\
&\leq \sum_{u\neq v^\star} w_u (d(v,v^\star) - w_{v^\star} d(v^\star,v) \\
&\leq d(v,v^\star)\left(\sum_{u\neq v^\star} w_u - w_{v^\star}\right) \\
&\leq 0.
\end{aligned} \tag{4.5}$$

D'où le resultat .

pour énoncer le prochain lemme, nous introduisons les notations suivantes :
Soient $G = (V, E)$ un graphe et $e = (a, b)$ un isthme [2] de G . Notons par V_a et V_b les deux composantes connexes de G contenant , respctivement, les sommets a et b obtenues par la suppression de l'arête e .On définit :

$$D(a,b) = \sum_{v \in V_a} w_v - \sum_{v \in V_b} w_v$$

Lemme 4.2 *Soit $e = (a, b)$ un isthme et supposons que $D(a, b) \geq 0$, alors il existe un sommet $v \in V_a$ qui est le 1- médian.*

Preuve 4.2 *en utilisant la définition de la fonction objectif, il est facile de voir que*

$$f(a) - f(b) = l(a,b)D(a,b). \tag{4.6}$$

et donc :

$$f(a) - f(b) \geq 0$$

de plus, nous allons prouver que

$$f(b) \leq f(v) \forall v \in V_b. \tag{4.7}$$

Notons que :

$$f(v) = \sum_{u \in V_b} w_u d(u,v) + \sum_{u \in V_a} w_u d(u,b) + \sum_{u \in V_a} w_u d(b,v). \tag{4.8}$$

pour tous $v \in V_b$ Le second terme de droite de l'équation 4.8 est une canstante qui ne dépend pas de v, donc on peut redéfinir le nouveau poids , $\tilde{w}_v$,d'un sommet v de V_b par :

$$\tilde{w}_v = \begin{cases} w_v & \forall v \neq b \in V_b \\ \sum_{v \in V_a} w_v + w_b & v = b \end{cases}$$

il est claire que cela revient à minimiser la nouvelle fonction objectif qui est diminuée de la quantité $\sum_{u \in V_a} w_u d(b,u)$ pour tout $v \in V_b$ et que l'on a : $2\tilde{w}_b \geq \sum_{u \in V_b} \tilde{w}_u$.
En utilisant le lemme 4.1, l' inégalité 4.7 et en combinant 4.6 et 4.7 on en déduit le résultat voulu.

Théorème 4.1 *$v^\star$ est un médian d'un arbre $T = (V, E)$ si et seulement si $D(v^\star, v) \geq 0$ pour tout voisin v de $v^\star$.*

Comme conséquence directe, on en déduit un algorithme (du à Goldman) en $O(n)$ de recherche d'un médian m dans un arbre. Cet algorithme s'énonce comme suit :

2. un isthme est une arête pour lequel le graphe $G - e$

Algorithm 6: ALGORITHME DE GOLDMAN

Input: T,W.

Output: V

Initialisation : $W(T) = \sum_{k=1}^{n} W(v_i)$(poids de l'arbre T)

Trouve $\leftarrow 0$

while ($|V| \geq 2$ *et trouve* $= 0$) **do**

  Choisir une feuille e

  if ($W(e) < W(T)/2$) **then**

    Soit v le sommet adjacent e

    Faire $V \leftarrow V - e$ et $W(v) = W(v) + W(e)$.

  else

    Trouve $\leftarrow 1$

    $m \leftarrow e$

if ($|V|) = 1$ **then**

  Le sommet e est le médian cherché.

3 problème du 1-médian paramétrique dans un arbre.

Lemme 4.3 *Soit* $T = (V, E)$ *un arbre . On suppose que :* $V = \{v_1, v_2, ..., v_n\}, \forall e \in E\ l(e) \geq 0$ *,* $w_i \geq 0\ \forall\ i = 1, 2, ..., n$ *et* $w_1(t) = w_1 + t$ *. (* $t \geq 0$*). notons par :*
-$v^\star \in V$ *un 1- médian de* T *pour* $t = 0$
- $[v^\star, v_1]$ *l'ensemble des sommets qui se trouvent sur la chaîne qui joint* $v^\star$ *et* v_1
Alors pour toute valeur du paramétre t ($t \geq 0$ *) il existe un sommet* v_t *situé entre* v_1 *et* $v_\star$ *qui soit médian de T .*

Preuve 4.3 *(par l'absurde)*
Supposons qu'il existe t' ($t' > 0$*) pour lequel il n'existe pas de sommet* m_t *sur le chemin* $\{v^\star, v_1\}$ *qui soit 1-médian de T .*
Notons par v' *la solution optimale pour laquelle le nombre d'arêtes situées entre* $v^\star$ *et v est minimum et supposons que* (v', v'') *est la premiére arête de ce chemin.Alors on en déduit que* $l(v', v'') > 0$ *. En utilisant le lemme 4.2 on en conclut que* $D(v', v'') > 0$ *et* $f(v') < f(v'')$ *pour* t' *donnée .*
$v^\star \in V_{v''}$ *alors les deux inégalités restent vrai pour tout paramétre non négatif* $\tilde{t} \leq t'$ *et en particulier pour* $\tilde{t} = 0$ *. ceci contredit le fait que* $v^\star$ *est le médian pour* $t' = 0$.

Notons que la preuve de ce lemme nous mène immédiatement à un algorithme en O(n) du problème paramétrique du 1-médian dans un arbre . Il suffit de considérer uniquement les sommets de la *chaîne* reliant les sommets $v^\star$ et v_1 .
ci-dessous nous présentons un algorithme qui nous donne la *chaîne* reliant les sommets $v^\star$ et v_1 ainsi que les plus courtes distances entre v_1 et ces sommets :

Algorithm 7: Algorithme de recherche de la chaine reliant v1 á v* dans un arbre.

Input: T,med.
Output: d

1 Appliquer une procdure pour déterminer une chaîne partir du sommet v_1
2 **for** $i \leftarrow 1$ ***to*** $|V|$ **do**
3   **if** $V_i = med$ **then**
4         $V = V(1:i)$
5   aller en 9
6 **if** $V_{|V|} \neq med$ **then**
7   Supprimer la derniere arête $u = [V_{|V|-1}, V_{|V|}]$
8   aller en 1
9 **for** $k \leftarrow 1$ ***to*** $|V|$ **do**
10   Actualiser : $p \leftarrow 0$
11   **for** $j \leftarrow 1$ ***to*** $|V| - 1$ **do**
12         $p \leftarrow p + T(V_j, V_{j+1})$
13         $d(k, j+1) \leftarrow p$
14         $d(k, j+1) \leftarrow d(j+1, k)$

3.1 Algorithme de génération aléatoire d'un arbre.

Pour pouvoir estimer notre algorithme, on génère de façon aléatoire des arbres . Pour cela on va se baser sur le théoréme suivant :

Théorème 4.2 *Soit* T = (V, E) *un graphe simple. Les assertions suivantes sont équivalentes :*

1. T *est un arbre .*
2. T *est connexe et chaque arête est un isthme .*

Preuve 4.4 *1)* $\Longrightarrow$ *2) :*
T étant un arbre , donc T est connexe (par definition d'un arbre) . Il reste à montrer que toute arête est un isthme . Soit e = (x, y) *une arête supposons que e n'est pas un isthme alors sa suppression ne déconnecte pas l'arbre et donc il existe une chaîne* C *qui relie x et y . Ceci implique que* $C \cup \{e\}$ *est un cycle de T . Cela est impossible car T est un arbre et ne contient pas de cycle .Donc e est un isthme et on a l'implication directe .*
2) $\Longrightarrow$ *1) :*
T étant connexe donc il reste à montrer que T est sans cycle . Comme toute arête est un isthme donc il n'appartient à aucun cycle de T . Cela veut dire que T ne possède pas de cycle et par suite T est un arbre .

Ci-dessous nous donnons un algorithme de génération aléatoire d'un arbre :

Algorithm 8: GEN-ARBRE

Input: n, k
Output: T

$T \leftarrow \infty$
for $i \leftarrow 1$ ***to*** n **do**
    for $j \leftarrow 1$ ***to*** n **do**
        if $i = j$ **then**
            $T(i, j) \leftarrow 0$
$V \leftarrow \{v_2, ..., v_n\}$
$U \leftarrow \{v_1\}$
while $|V| \neq 0$ **do**
    Choisir arbitrairement v dans V
    Faire : $V \leftarrow V - \{v_i \in V\}$
    Choisir arbitrairment u dans U
    Faire :$U \leftarrow U \cup \{v\}$
    Choisir alatoirement c dans [1,k]
    Poser $T(u, v) \leftarrow c$ et $T(v, u) \leftarrow c$
return T

4 problème du 1-médian paramétrique dans un cactus

4.1 introduction

L'idée de résoudre le problème paramétrique du 1-médian dans un cactus est de réduire les cycles pour obtenir un arbre. Ensuite, le problème est résolu dans l'arbre. Enfin, il est possible d'utiliser la solution de l'arbre pour obtenir une solution au problème d'origine

4.2 Définition

Un cactus est un graphe connexe dans lequel deux cycles élémentaires ont au plus un sommet en commun. De manière équivalente, un cactus est un graphe tel que chaque arête appartient à au plus un cycle. Les cactus sont des graphes planaires.

Exemple 4.1 *Soit* G *le cactus d'ordre 11 et ayant trois cycles* C_1 ,C_2 *et* C_4 *suivant :*

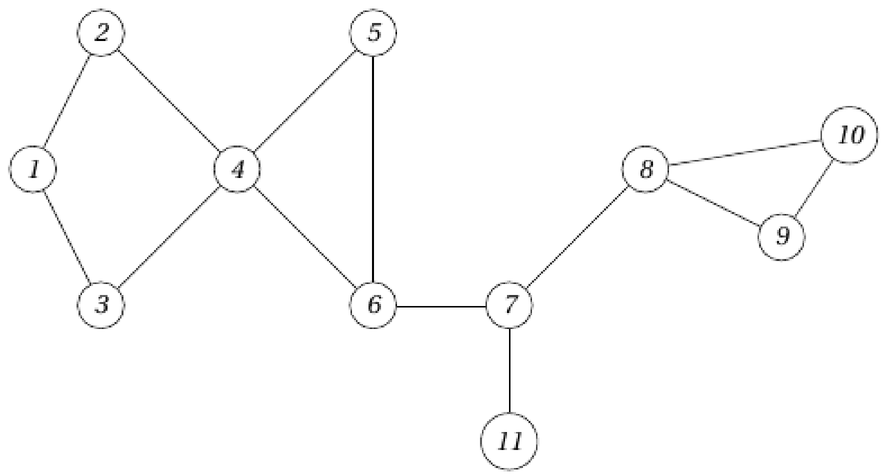


FIGURE 4.1 – La représentation graphique d'un cactus.

Nous présentons une méthode de résolution du problème qui est constituée de deux phases :

4.3 phase 1

Avant de discuter en détail de l'algorithme, nous supposons sans perte de généralité que chaque sommet du cactus est contenu dans au plus un cycle. Si ce n'est pas le cas, c'est-à-dire qu'un sommet v est dans les cycles $C_1, \ldots, C_k$ nous peut introduire k sommets artificiels $v'_1, \ldots, v'_k$ et arêtes (v'_i, v) de longueur 0 pour tous $i = 1, \ldots, k$. Au lieu des arêtes (v, u) du graphe d'origine, nous ajoutons les arêtes (v'_i, u) avec $l(v', u) = l(v, u)$ si u est dans le cycle C_i. Le poids des nouveaux sommets est défini à 0. Soit G un cactus,à ce cactus est associé à l'arbre T = (V, E) où V = $\{C_1, C_2, \ldots, C_K\}$ est formé par l'ensemble des cycles de G , et E par l'ensemble des isthmes de G . Pour $C_i \in V$, on définit $w(C_i) = \sum_{v \in C_i} w(v)$.

Notons que si le sommet dont le poids dépend de t était dans un cycle, le sommet représentant ce cycle ne dépend maintenant de paramètre aussi. Maintenant, le problème paramétrique du 1-médian est résolu sur l'arbre (le nouveau graphe), c'est-à-dire si un sommet de l'arbre est optimal pour certains $t \geq 0$, il est optimal si et seulement si t est dans un intervalle fermé $[t_1, t_2]$ ou dans l'intervalle $[t_1, \infty)$ pour certains $0 \geq t_1 < \infty$.

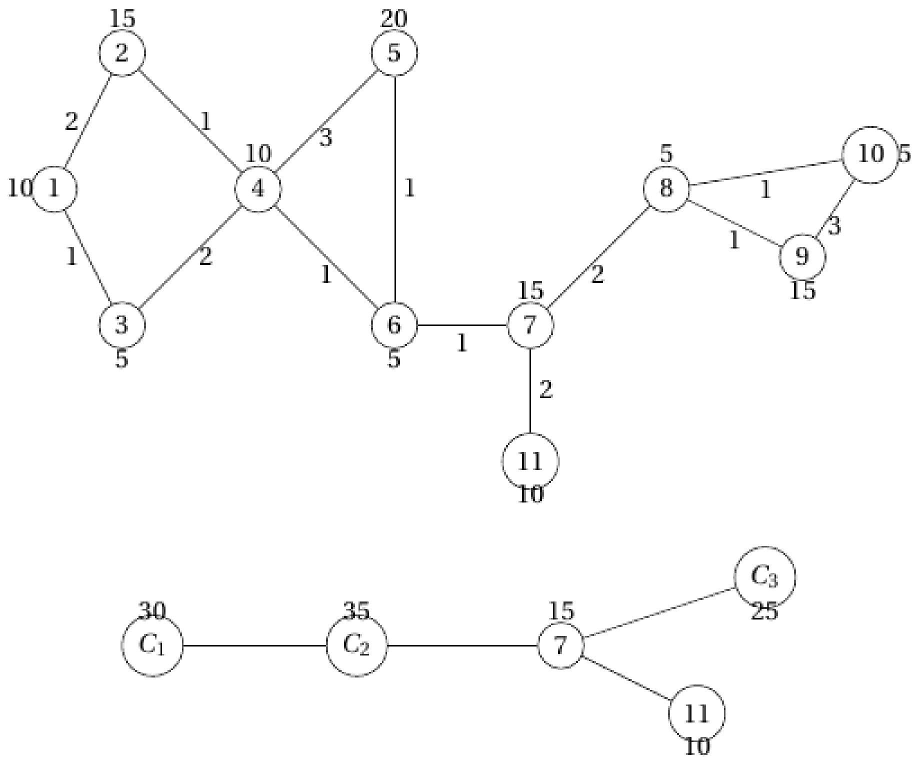


FIGURE 4.2 – Le graphe de cactus et l'arbre induit correspondant.

4.4 phase 2

Si v était obtenu en rétrécissant un cycle C et est optimal pour tout $t' \in [t_1, t_2]$ (ou $[t_1, \infty)$), alors il existe un sommet $v \in$ C dans le cactus qui est optimal pour une valeur de paramètre dans l'intervalle $[t_1, t_2]$ (ou $[t_1, \infty)$). En fait, nous devons résoudre un problème paramétrique du 1-médian dans le cycle C pour tout $t' \in [t_1, t_2]$ (ou $[t_1, \infty)$). Notons les sommets du cycle C par $v_1, ..., v_s$. Ensuite, nous devons calculer à nouveau s droites du $f_i(t)$, une pour chaque sommet du cycle, avec $\sum_{j=1}^{n} w_j d(v_j, v_i)$ c'est-à-dire la valeur de la fonction objectif du problème initial pour $t = 0$ et la tengente $d(v_i, v_1)$ pour $i = 1, ..., s$. Il est montré que l'intersection de ces droites peuvent être calculées en temps linéaire. De plus, il est également possible de calculer les valeurs $d(v, v_1)$ pour tout v dans le cactus en temps linéaire. En particulier, nous pouvons obtenir les tengentes en $O(n)$. nous pouvons résoudre le problème paramétrique dans le cycle C . En conclusion, on peut dire que le problème du 1- médian paramétrique considérée dans le cactus peut être résolu en temps linéaire.

4.5 Algorithme de génération d'un cactus.

Pour pouvoir estimer notre algorithme, on génére de façon aléatoire des cactus .Cela consiste à génére un arbre d'ordre n puis on génére des cycles celon le degré de chaque

sommets d' arbre .
Ci-dessous nous donnons un algorithme de génération aléatoire d'un cactus.

Algorithm 9: GEN- CACTUS

Input: n, k
Output: G
$G \leftarrow \infty$
$T \leftarrow Gen - arbre(n, k)$.
Choisir alatoirement m dans $[1, 6]$
. $C \leftarrow Gen - cycle(m, k)$.
for $i \leftarrow 1$ ***to*** n **do**
    Calculer d le degr du sommet v_i dans T.
    for $r \leftarrow 1$ ***to*** d **do**
        Choisir alatoirement un sommet v dans le cycle C.
        Choisir alatoirement m dans $[1, 6]$
        . $C_t \leftarrow Gen - cycle(m, k)$.
        Choisir alatoirement un sommet u dans le cycle trouvée C_t.
        Choisir alatoirement b dans [1,k]
        Poser $G(u, v) \leftarrow b$ et $G(v, u) \leftarrow b$
        $C \leftarrow C \cup C_t$
return G

Exemple 4.2 *Considérons le cactus,*G *, donné Ci-dessous. On veut déterminer le 1-médian de* G.

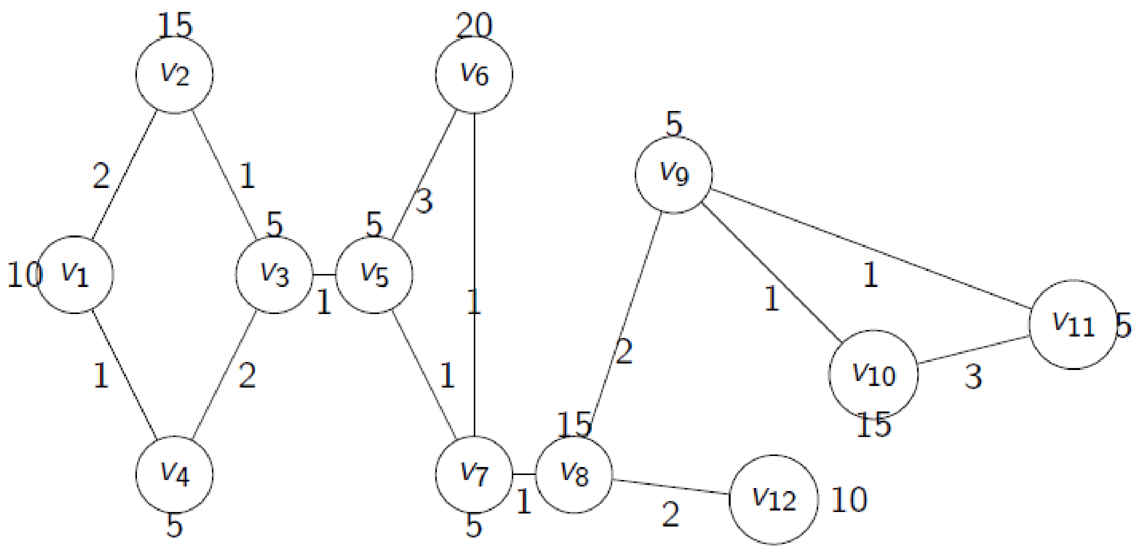


FIGURE 4.3 – Le graphe de cactus.

Phase 1 :(Localiser le Cycle où se trouve le médian .)

Ci-dessous, nous donnons la procédure de calculer le médian pour $t = 0$ (En utilisant l'algorithme de Goldman), dans l' arbre induit d'ordre 5 dont le poids total est de 115 :

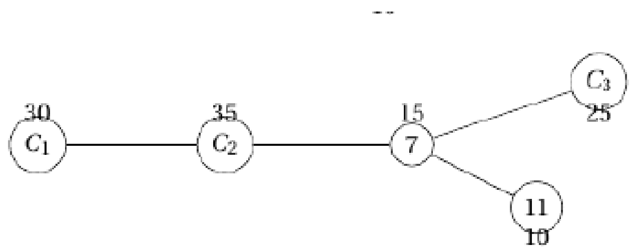
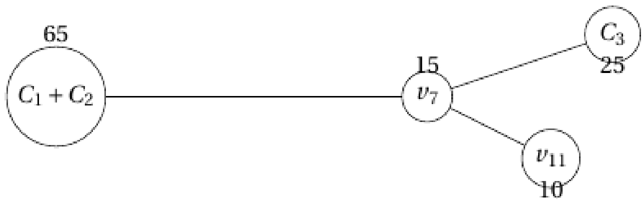
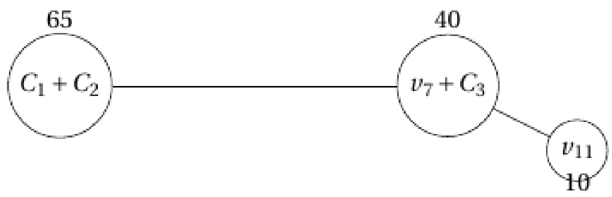


FIGURE 4.4 – L'arbre associé au graphe de cactus.

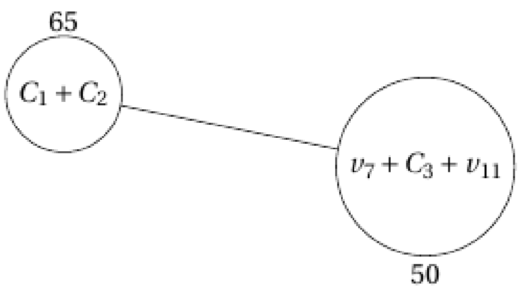
On démarre à partir du sommet C_1 dont le poids est égal à 35. 35 étant inférieure à 115/2 =57.50 . Donc on supprime le sommet C_1 et le poids de C_2 (sommet adjacent à C_1) sera augmenté à 65.



Le sommet C_3 (par exemple) est un sommet pendant de poids égal à 25, donc inférieure à 57.50 (moitié de poids total de T), Alors le sommet C_3 sera Supprimé et le poids du sommet v_7 (sommet adjacent à C_3) sera augmenté de 15 pour devenir 40.



Le sommet v_{11} est pendant, son poids est de 10, donc inférieur à 57.50. v_{11} sera supprimé et le poids de $v_7 + C_3$ sera augmenté de 40 pour atteindre le poids de 50.



On s'aperçoit alors que le poids de $C_1 + C_2$ est égal à 65, il est de valeur qui est supérieure à la moitié de poids total de l'arbre , d'où le sommet C_2 est le sommet cherché : C'est le

médian de l'arbre pour $t = 0$.
On déduit alors la chaîne $\{C_1, C_2\}$,où le poids associé à chaque sommet dans cette chaîne sera égale à $\{30 + t, 85\}$.
- âfin de résoudre ce problème, nous ne faisons que prendre en compte les valeurs de la fonction objectif tel que :

$$f_i(t) = \sum_{j=1}^{n} w_j d(v_i, v_j) + t d(v_i, v_1)$$

de ces sommets en fonction du paramètre t. Nous pouvons définir une fonction linéaire pour chaque sommet du chaîne de C_2 et C_1 . On trouve donc :
1-

$$f_2(t) = 30 + t.$$

2-

$$f_1(t) = 85.$$

On déduit alors que C_2 est le médian pour toute $t \in [0, 55[$,et C_1 est le médian pour toute $t \in [55, \infty[$.
Phase 2 :(calcul du 1-médian dans le cycle trouvée.)
La dernière phase se réduit à la résolution d'un médian dans un cycle.Prenons maitenant le cycle C_1 .En calculant $d(v_i, v_j)$ est donné par les équation 4.2 ,4.1 et 4.4 ,on déduit maintenant les droites associé à toutes les sommets de cycle C_1 comme suit :

$$f_2(t) = 40 + 2t.$$

$$f_3(t) = 50 + 3t.$$

$$f_4(t) = 60 + 2t.$$

$$f_1(t) = 70.$$

Finalement,nous calculons que v_2 est le médian pour toute $t' \in [0, 15[$,et v_1 est le médian pour toute $t' \in [15, 55[$.

Chapitre 5

Résultats Numériques

1 introduction

Nous avons programmé les différents algorithmes en langage Matlab, sur un PC de type Asus avec un processeur :Intel(R)Coleron(R) CPU N3060 @ 1.60GHz 1.60 GHz, et de RAM 2.00 Go Pour ce faire, nous avons fait des tests sur des exemples que nous avons généré aléatoirement. Tous les tests numériques confirment les résultats théoriques.

2 Algorithme de recherche du 1-median paramétrique dans un cycle

Nous avons programmé l'algorithme de recherche du 1-médian paramétrique dans un cycle en language Matlab. Nous avons testé cet algorithme en variant $t \in \mathbb{R}_+$ sur 5 exemples d'ordre variant 20 à 5000 que nous avons généré aléatoirement par l'algorithme de génération aléatoire d'un cycle .

Exemple	n	t	médian	CPU
C_1	20	0	7	0.1906
		20	6	0.1132
		101	2	0.1101
		150	1	0.1184
C_2	50	0	9	1.0093
		8.5	7	0.8594
		55	7	0.9022
		115	5	0.7594
		395	1	0.7403
C_3	150	0	142	8.6045
		120	148	8.0235
		150	149	8.3885
		201	150	8.0246
		315	1	8.0994
C_4	560	0	131	182.3185
		55	130	135.5201
		55.5	130	164.5401
		800	100	136.3653
		1225	39	130.5155
C_5	2500	0	12	161.7479
		30	9	164.9806
		100	7	172.6443
		1550	1	173.4503
C_6	5000	0	2030	$1.1907.10^3$
		99	2024	$1.2005.10^3$
		550	1559	$1.1196.10^3$
		2019	1	$1.1187.10^3$

TABLEAU 5.1 – étude du problème 1-médian paramétrique dans un cycle.

3 Algorithme de recherche du 1-median dans un arbre

3.1 Algorithme de Goldman

Nous avons généré aléatoirement cinq arbres que nous notons par T_1, T_2,T_3 ,T_4,T_5. Les poids des sommets sont générés dans l'intervalle [1,50] et les coûts des arêtes sont pris tous égaux à 1 (la recherche du 1-médian ne dépend pas des coûts des arêtes). Le tableau ci-dessous résume les différents temps CPU et les nombres d'itérations de l'algorithme de Goldman sur ces cinqs exemples.

Exemple	n	médian	AG(CPU)	nombres des iterations
T_1	40	8	0.1899	32
T_2	80	6	0.7435	78
T_3	150	118	2.4120	144
T_4	500	61	24.3411	497
T_5	1800	1	304.1923	1784

TABLEAU 5.2 – Résultats de l'algorithme de Goldman.

4 Algorithme de recherche du 1-median paramétrique dans un arbre

on donne quelques résultats de la méthode de recherche du 1-médian paramétrique dans un arbre . Nous avons généré (Ci-dessous) aléatoirement 7 exemples d'arbre d'ordre variant 10 à 5000 en variant toujours le paramétre $t \geq 0$.

Exemple	n	t	médian	CPU
T_1	10	0 75	6 1	0.0883 0.0442
T_2	25	$\forall t \geq 0$	1	0.0874
T_3	80	0 83 200 2000	22 22 22 1	0.7544 0.7067 0.6983 0.8161
T_4	160	0 2019	108 1	2.9020 2.9132
T_5	1000	0 6550 20150	771 771 1	95.3887 95.6561 95.4235
T_6	2500	0 12860 56173	2426 949 1	754.9618 728.1065 552.8857
T_7	5000	0 31742	3306 1	$4.2703.10^3$ $3.8216.10^3$

TABLEAU 5.3 – étude du problème 1-médian paramétrique dans un arbre.

5 Algorithme de recherche du 1-median paramétrique dans un cactus

on donne quelques résultats de la méthode de recherche du 1-médian paramétrique dans un cactus . Nous avons généré (Ci-dessous) aléatoirement 6 exemples de cactus d'ordre variant 27 à 6708 en variant toujours le paramétre $t \geq 0$.

Exemple	n	t	médian	CPU
G_1	27	0	4	0.0533
		25.5	2	0.0089
		101	1	0.0054
G_2	61	0	3	0.0036
		15	1	0.0017
G_3	198	0	2	0.0094
		35.5	1	0.0020
		88	1	0.0018
G_4	540	0	1	0.0033
		2000	1	0.0069
G_5	2273	0	4	0.0084
		50	5	0.0112
G_6	6708	0	5	0.0926
		25	5	0.0036
		50	1	0.0045

TABLEAU 5.4 – étude du problème 1-médian paramétrique dans un cactus.

Conclusion

Nous nous sommes intéressé au problème du 1-médian dans une classe de graphe (arbre , cycle et cactus) et nous avons étudié le cas de la paramétrisation d'un seul sommet et on a vu que le problème reste facile . Des extensions au cas du 2-médian est facile (cela n'a pas été fait faute de temps). Il serait intéressant de généraliser la paramétrisation à tous les sommets du graphe : on peut commencer par des fonctions élémentaires du paramétre t . C'est ce que nous envisageons de faire à l'avenir.

Bibliographie

[1] **A. J. Goldman.** (1971), *Optimal center location in simple network.*, Transp Sci, **5**, 212 – 221.

[2] **Alain. Berto. and Alain. Faisant. and François. Hennecart** (2011), *éLéments de théorie des graphes*, Caen, Saint-étienne.

[3] **Beaumont J.R.** (1987), *Allocation Models and Central Place Theory, Spatial Analysis and Location Allocation Models.*, ed. Ghosh Avijit, Rushton Gerard, Van Nostrand Reinhold.

[4] **Bekada. Karima.** *Méthode de résolution d'un problème d'optimisation combinatoire.*, mémoire master, Université de Mostaganem,(2015).

[5] **Bekkar. Habib et Houidef. AEK.** *Résolution du problème P-médian dans un arbre.*, mémoire master, Université de Mostaganem,(2005).

[6] **Christo. des, N. and beasley. J. E** (1982), *A tree search algorithm for the p-median problems.*, European Journal of Operational Research, **vol , 10**, 196– 204.

[7] **Hatzl.J.** (2007), *Median problems on wheels and cactus graphs*, Computing, **80**, 383 – 388. doi :10.1007/s00607-007-0238-y

[8] **Kariv. O, Hakimi. S.** (1979), *An algorithmic approach to network location problems part II : p-médians.*,SIAM J Appl Math , **27**, 539– 560.

[9] **Gondron et M.** (1995), *Graphe et Algorithmes.*, Eyrolles, Paris.

[10] **Thabet. Naoufel.** *Algorithme de génération de colonnes pour le problème du P médian.*, mémoire master, Université de Montréal,(1998).

Titre

Résumé : Nous avons étudié le problème du 1-médian paramétrique dans trois classes particulières de graphes (arbre, cycle et cactus).

Mots-Clés. Les graphe, p-médian, complexité, dualité lagrangienne, sous-gradient, Algorithme de Goldman,paramétrique, arbre,cycle, cactus.

Title

Abstract : We studied the 1-median problem in three particular classes of graphs (tree, cycle, and cactus).

Key Words. The graph, p-mediane, complexity, Lagrangian duality, subgradient, Goldman algorithm,parmetric, tree, cycle, cactus.