

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



MÉMOIRE

Master Académique

pour obtenir le diplôme de Master délivré par

Université de Mostaganem

Spécialité "Modélisation, Contrôle et Optimisation"

présenté et soutenu publiquement par

Gorine LEILA

le 14 Juin 2018

Problème du P-médian dans une classe de graphes

Encadeur : **Mr Abloui HOCINE (UNIVERSITÉ DE MOSTAGANEM, ALGÉRIE)**

Jury

Amir Abdessamad,	Professeur	Président (Université de Mostaganem, Algérie)
Bahnes Nacera,	MAA	Examinatrice (Université de Mostaganem, Algérie)

**LABORATOIRE DE MATHÉMATIQUES PURES ET APPLIQUÉES
FACULTÉ DES SCIENCES EXACTES ET DE L'INFORMATIQUE (FSEI)
Chemin des Crêtes (Ex-INES), 27000 Mostaganem, Algérie**

**M
A
S
T
E
R**

Liste des figures

1.1	Un graphe orienté	4
1.2	Exemple d'un chemin	5
1.3	Exemple d'un graphe circuit	5
1.4	Graphe non connexe	7
1.5	Graphe complet	8
4.1	Un arbre	21
4.2	Exemple des roues	34
4.3	Un cactus	40
4.4	Le graphe de cactus et l'arbre induit correspondant	42

Remerciements

Je remercie tout d'abord mon dieu le tout puissant de m'avoir donné le courage, la patience et la volonté durant toutes mes années d'études. Mes sincères remerciements s'adressent à mon encadreur Mr. Ablouï Hocine, professeur à l'université de Mostaganem, qui m'a fait l'honneur de m'encadrer en Master et de juger ce travail.

Un grand merci pour votre disponibilité, votre implication et vos judicieux conseils. Veuillez trouver ici l'expression de ma profonde et sincère reconnaissance.

Je voudrais remercier les membres du jury, Melle. Bahnes Nacera d'avoir bien voulu présider mon travail, et Mr. Amir Abdessamad d'avoir fait partis de ces membres.

Mes remerciements s'adressent également à tous mes enseignant(e)s en Licence et en Master.

Je tiens à remercier particulièrement, Bouksara Naima, Gorine Fatima et Benguanouna Asmaa pour tous les conseils et pour toutes les corrections effectuées.

Je remercie infiniment tous les étudiants de ma promotion et en particulier de ma spécialité Modélisation Contrôle et Optimisation qui m'ont tant soutenu et encouragé.

Enfin un grand merci à ma famille Gorine et surtout mon père et ma mère, aussi la famille Ferraoun de m'avoir le courage pour mener à terme ce travail.

A ma famille et mes amies.

Table des matières

Remerciements	3
Liste des abréviations	2
1 Notions sur les graphes	4
1 Introduction	4
2 graphe orienté	4
3 graphe non orienté	6
2 Introduction du problème du p-médian	9
1 Présentation du problème	9
2 Formulation mathématique du problème du p-médian	9
3 Application pratiques	10
4 complexité	10
3 Méthodes de Résolution	11
1 Méthodes approchées	11
2 Heuristique de Teitz et Bart	11
3 Heuristique de Maranzana	15
4 Méthode exacte	17
5 Dualité lagrangienne	17
6 L'algorithme du sous-gradient	19
4 Cas particuliers	21
1 Problème du 1-médian dans un arbre	21
2 Problème du 2-médian dans un Arbre	27
3 Problème du 1-médian dans une cycle	32
4 Problème du 1-médian dans une roue	34
5 Problème du 1 -médian dans un cactus	40
5 Les résultats numériques	45
1 Introduction	45
2 Algorithme de Goldman	45
3 Algorithme de recherche du 1-médian dans un cycle	47
4 Algorithme de recherche du 1-médian dans une roue	48

Liste des abréviations

- Gen-A** : Algorithme de génération d'un arbre.
- Gen-C** : Algorithme de génération d'un cycle.
- Gen-R** : Algorithme de génération d'un roue.
- AG** : Algorithme de Goldman.
- MC** : Méthode de recherche du 1-médian dans un cycle.
- MR** : Méthode de recherche du 1-médian dans une roue.
- MSG** : Méthode de sous gradient.

Introduction

L'une des premières personnes qui a étudié les problèmes de localisation est Alfred Weber en 1903. Il s'est intéressé au problème de la localisations dans le plan d'une usine par rapport à deux ressources et un magasin de vente.

les problème de localisation ont énormément d'applications pratiques : que ce soit en physique pour ne citer que le problème de la localisation du centre de gravité d'un solide, en chimie, en économie, en logistique, en informatique, en biologie et dans bien d'autres domaines encore.

Kariv et Hakimi 1980 [9], ont montré que le problème de P-médian est NP-dur (difficile), et il ont montré que le problème devient polynomiale si le graphe se réduit à un arbre.

L'objectif de notre travail est d'étudier des classes particulières de problèmes de localisation : nous nous intéressons aux problèmes du 1-médian et de 2-médian dans un arbre, un cycle, une roue 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 pour terminer le chapitre par le théorème sur la complexité algorithmique du problème du p-médian dans un graphe.

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 : nous donnons des méthodes efficaces pour déterminer le 1-médian dans un arbre (algorithme de Goldman), dans un cycle, une roue 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 Introduction

Nous distinguons deux grandes familles de graphes : les graphes orientés, et les graphes non orientés. Ci-dessous, nous donnons quelques rappels sur les graphes[13].

2 graphe orienté

Un graphe orienté est un graphe qui est défini par la donné :

- D'un ensemble fini V (appelé ensemble des sommets).
- D'un ensemble fini E (appelé ensemble d'arcs).
- De deux applications I et T qui sont définies de $E \rightarrow V$ qui définissent, respectivement, l'extrémité initiale et terminale.

Exemple :

Soit $G=(V,E)$ un graphe orienté telle que $V=(1, 2, 3, 4, 5)$ et $E=(e_1, e_2, e_3, e_4, e_5, e_6, e_7)$.

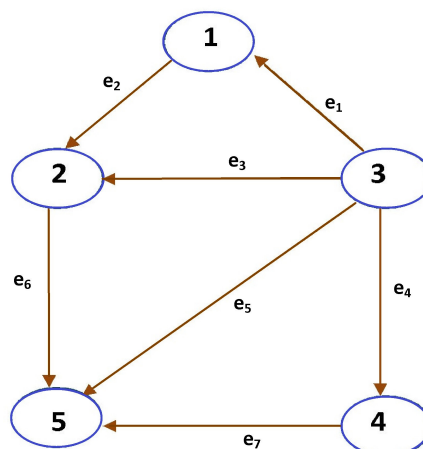


FIGURE 1.1 – Un graphe orienté

2.1 Définition (boucle)

En général, on note par $e = (x,y)$, un arc dont les extrémités sont x et y . Si les extrémités x et y de e coïncident, e est dit boucle .

2.2 Définition (ordre)

L'ordre N d'un graphe est le cardinal de l'ensemble de ses sommets. On note l'ordre de G par : $N = |V|$.

2.3 Définition (adjacent)

Si $e=(x,y)$, x et y sont dit adjacents.

2.4 Chemin

Un chemin de x à y est une séquence d'arcs (e_1, e_2, \dots, e_s) de E telle que : $I(e_1) = x$, $T(e_s) = y$ et enfin, $T(e_k) = I(e_{k+1}) \forall k, k = 1, 2, \dots, s - 1$.

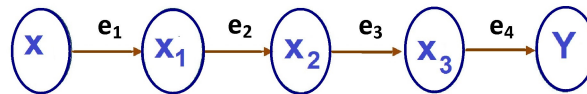


FIGURE 1.2 – Exemple d'un chemin

Le chemin sera dit circuit si l'extrémité initiale du premier arc de la séquence coïncide avec l'extrémité terminale de la séquence.

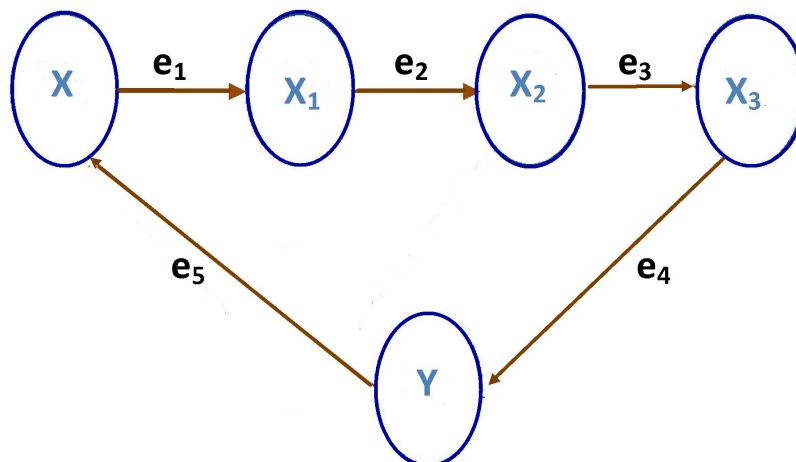


FIGURE 1.3 – Exemple d'un graphe circuit

2.5 Chemin élémentaire

Un chemin est dit élémentaire si en le parcourant, on passe une fois par chacun de ses sommets.

2.6 Chemin simple

Un chemin est dit simple si en le parcourant, on passe une fois par chacun de ses arcs.

Remarque

Tout chemin élémentaire est Simple.

2.7 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 multigraphe.

3 graphe non orienté



Un graphe non orienté **G** est défini par la donnée :

- D'un ensemble fini **E** (appelé ensemble des sommets).
- D'un ensemble fini **V** (appelé ensemble d'arêtes).
- D'une application **e** qui est définie $e : E \rightarrow \{\{a, b\} / a \in V, b \in V\}$. **e(u)** définit les extrémités de l'arête **u** de **E**.

3.1 Chaîne

Une chaîne reliant deux sommets **x** et **y** est une séquence d'arcs ou d'arêtes (e_1, e_2, \dots, 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 Comparaison

Orienté	Non orienté
Arc	Arête
	
Chemin	Chaîne
Circuit	Cycle

3.3 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 .

3.4 Graphes connexes

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

Exemple Par exemple, le graphe non orienté suivant :

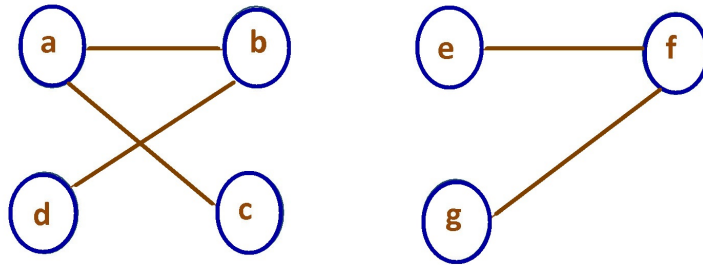


FIGURE 1.4 – Graphe non connexe

n'est pas connexe car il n'existe pas de chaîne entre les sommets a et e . En revanche, le sous graphe construit sur l'ensemble des sommets $\{a, b, c, d\}$ est connexe.

3.5 composante connexe

Une composante connexe d'un graphe G est tout sous graphe G' de G qui est connexe.

- Un graphe est dit connexe si et seulement si il admet une unique composante connexe. Par exemple, le graphe précédent est composé de 2 composantes connexes : la première est le sous-graphe induit par les sommets $\{a, b, c, d\}$, et la seconde est le sous-graphe induit par les sommets $\{e, f, g\}$.

3.6 graphe complet

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

Exemple : les graphe suivants sont complet :

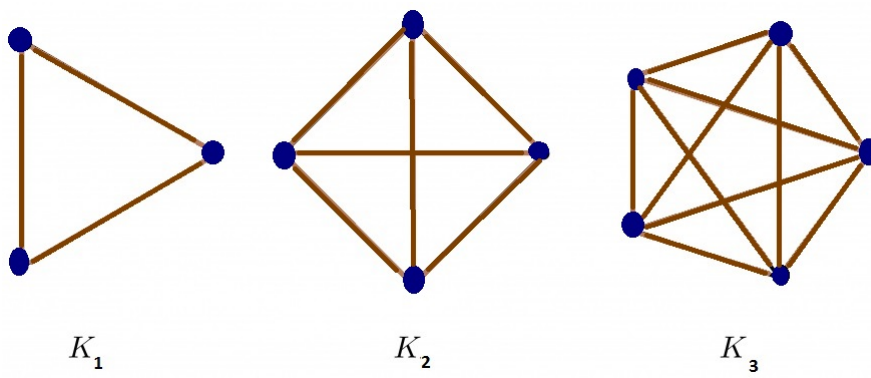


FIGURE 1.5 – Graphe complet

Chapitre 2

Introduction du problème du p-médian

1 Présentation du problème

Le problème du p-médian dans un graphe est un problème qui consiste à localiser p installations (appelées médians) pour servir n demandes situées sur l'ensemble des sommets du graphe. La localisation des p-centres doit être faite de telle sorte à ce que la somme des distances de chaque demande à son installation la plus proche soit la plus petite possible.

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 défini 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[4] :

$$(P) \left\{ \begin{array}{l} \text{Min}(z) = \sum_{i \in V} \sum_{j \in E} w_i d_{ij} x_{ij} \quad (2.1) \\ \sum_{i \in V} x_{ij} = 1 \quad \forall j \in V \quad (2.2) \\ \sum_{i \in V} x_{ii} = p \quad (2.3) \\ x_{ij} \leq x_{ii} \quad \forall i, \forall j \in V \quad (2.4) \\ x_{ij} \in \{0, 1\}, \forall i, \forall j \in V \quad (2.5) \end{array} \right.$$

ou

- d_{ij} : la plus courte distance entre les sommets i et j .

- w_i : le poids du sommet i .

On définit les variables par :

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

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

2.1 Interprétation de contraintes :

- (2.1) Représente le fonction objectif.
- (2.2) Assure que toutes les demandes sont servies par un et un seul serveur.
- (2.3) Le nombre de médians est fixé à p .
- (2.4) Une demande j est servi par i si une installation a été ouverte en i .
- (2.5) Nature binaire des variable x_{ij} .

3 Application pratiques

Les applications pratiques du problème du p-médian sont nombreuses, nous citerons entre autres et sans être exhaustif[12] :

- Hakimi (1964) a modélisé la localisation optimale des centres de communication dans un réseau de télécommunication par un problème du 1-médian.
- Le problème de la localisation optimale de comptes bancaires peut être formulé par un problème du p-médian (Fisher et Nemhauser (1977)).
- La localisation des usines et d'entrepôts (Maranzana (1964)).
- la localisation des lieux de tenu de conférences .

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 cout 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 (ou Heuristique), comme son nom l'indique, est une méthode qui permet l'obtention d'une solution approchée d'un problème donné.

En ce qui concerne le problème du p-médian (comme tout problème d'optimisation) on distingue deux grandes classes de méthodes :

- Les méthodes spécifiques au problème du p-médian. Nous citerons les méthodes de MARANZANA (1964) et celle de Teitz et Bart (1968). Ces méthodes sont efficaces pour des problèmes de taille raisonnable.
- Les méthodes méta-heuristiques qui sont utilisées pour des problèmes de grande taille. Nous avons la méthode du recuit simulé, l'algorithme génétique, la méthode de colonie de fourmis ...etc.

2 Heuristique de Teitz et Bart

C'est l'une des premières heuristiques développées pour le problème p -médian et Teitz et Bart (1968)[12] ont été les premiers à proposer une heuristique d'interchange-ment pour le problème p -médian. L'heuristique commence par générer une solution réalisable, puis tente d'améliorer la solution actuelle grâce à une opération de permutation, qui permet d'interchanger un médian avec un non médian. Le processus se poursuit jusqu'à ce qu'aucune amélioration de la valeur de la fonction objectif ne soit possible.

2.1 Le principe de l'heuristique de Teitz et Bart

- 1- Prendre une solution quelconque de départ(choisir p médians).
- 2- Calculer le coût total de la solution actuelle.
- 3- Choisir un médian **a** de la solution courante et un non médian **b**.
- 4- Interchanger **a** et **b** puis Calculer le coût total de la nouvelle solution.
- 5- Si le nouveau coût est inférieur à l'ancien, remplacez **b** par **a**, sinon gardez **a** dans la solution.

ALGORITHME DE TEITZ ET BART

Entrée : $G = (X; U), W, p, d :$

Initialisation

choisir, aléatoirement, l'ensemble $S = \{x_1, x_2, \dots, x_p\}$ de p sommets de X .

Calculer : $Q(S) = \sum_{k=1}^n W(x_k) d(x_k, S)$

Poser : $S_0 = \{x_{p+1}, x_2, x_3, \dots, x_p\}$

Tant que ($S \neq S_0$)

Poser : $S \leftarrow S_0$.

Pour tout $j = p + 1, p + 2, \dots, n :$

Pour tout $i = 1, 2, \dots, p :$

Calculer : $Q(S_0 \cup \{x_j\} \setminus \{x_i\}) = \sum_{k=1}^n W(x_k) d(x_k, (S_0 \cup \{x_j\} \setminus \{x_i\}))$

Calculer : $\Delta_{ij} = Q(S_0) - Q(S_0 \cup \{x_j\} \setminus \{x_i\})$

Fin pour tout

Déterminer Δ_{ioj} par $\Delta_{ioj} = \max_{x_i \in S_0} (\Delta_{ij})$

Si $\Delta_{ioj} \leq 0$

| $S \leftarrow S_0$

Sinon

| $S \leftarrow S_0 \cup \{x_j\} \setminus \{x_i\}$

| $Q(S) \leftarrow Q(S_0 \cup \{x_j\} \setminus \{x_i\})$

Fin Si

Fin pour tout

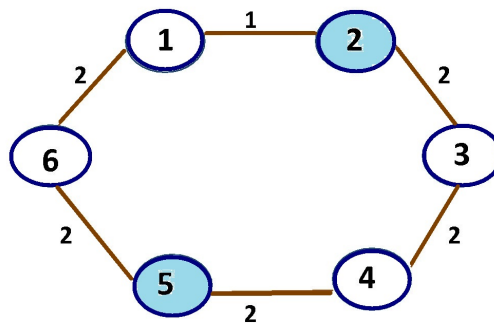
Fin tant que

Sortie : S

Fin de l'algorithme

2.2 Exemple

soit le graphe suivant :



Initialisation

Soit $p=2$.

Si on prend $S = \{3, 6\}$.

$$V - S = \{1, 2, 4, 5\}.$$

$$Q(S) = 2 + 2 + 0 + 2 + 2 + 0 = 8.$$

1^{ère} itération

$$x_i = 3.$$

$$x_j = 1.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(1, 6) = 0 + 1 + 3 + 4 + 2 + 0 = 10.$$

$$\Delta_{3,1} = 8 - 10 = -2.$$

$$x_i = 6.$$

$$x_j = 1.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(3, 1) = 0 + 1 + 0 + 2 + 4 + 2 = 9.$$

$$\Delta_{6,1} = 8 - 9 = -1.$$

$$\Delta_{i \circ 1} = \Delta_{6,1} = -1.$$

marquer "1" avec "+".

2^{ème} itération

$$x_i = 3.$$

$$x_j = 2.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(2, 6) = 1 + 0 + 2 + 4 + 2 + 0 = 9.$$

$$\Delta_{3,2} = 8 - 9 = -1.$$

$$x_i = 6.$$

$$x_j = 2.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(3, 2) = 1 + 0 + 0 + 2 + 4 + 3 = 10.$$

$$\Delta_{6,2} = 8 - 10 = -2.$$

$$\Delta_{i \circ 2} = \Delta_{3,2} = -1.$$

marquer "2" avec "+".

3^{ème} itération

$$x_i = 3.$$

$$x_j = 4.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(4, 6) = 2 + 3 + 2 + 0 + 2 + 0 = 9.$$

$$\Delta_{3,4} = 8 - 9 = -1.$$

$$x_i = 6.$$

$$x_j = 4.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(3, 4) = 3 + 2 + 0 + 0 + 2 + 4 = 11.$$

$$\Delta_{6,4} = 8 - 11 = -3.$$

$$\Delta_{i \circ 4} = \Delta_{3,4} = -1.$$

marquer "4" avec "+".

4^{ème} itération

$$x_i = 3.$$

$$x_j = 5.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(5, 6) = 2 + 3 + 4 + 2 + 0 + 0 = 11.$$

$$\Delta_{3,5} = 8 - 11 = -3.$$

$$x_i = 6.$$

$$x_j = 5.$$

$$Q(S \cup \{x_j\} \setminus x_i) = Q(3, 5) = 3 + 2 + 0 + 2 + 0 + 2 = 9.$$

$$\Delta_{6,5} = 8 - 9 = -1.$$

$$\Delta_{i \circ 5} = \Delta_{6,5} = -1.$$

marquer "5" avec "+".

La solution est $S = \{3, 6\}$ et l'objectif est égal à 8 :

La solution S obtenue (à partir de la solution initiale $S_0 = (1, 2)$).

Si l'on choisit la solution de départ $S_0 = (3, 5)$, la procédure s'arrête

en nous donnant une solution optimale $S = (2, 5)$ avec un l'objectif égal à 7.

3 Heuristique de Maranzana

Dans l'heuristique de localisation et allocation (Maranzana 1964)[12], les installations sont localisées initialement en p points choisis au hasard, les utilisateurs sont affectés à la installation la plus proche, et un problème 1-médian est résolu pour chaque ensemble d'utilisateurs affectés à une même installation. La procédure est ensuite itérée avec les nouvelles localisations des installations jusqu'à ce qu'il n'y ait plus de changements dans les affectations.

Soit l'algorithme de Maranzana de façon formelle :

ALGORITHME DE MARANZANA

Entrée : $G = (X; U); W; p : d.$

Initialisation

choisir, aléatoirement, l'ensemble $M = \{x_1, x_2, \dots, x_p\}$ de p sommets de X .

Poser : $S = \{x_1, x_2, \dots, x_p\}$.

Tant que($M \neq S$)

Poser : $S \leftarrow M$.

Pour tout $i = 1, 2, \dots, p$, faire :

Calculer l'ensemble P_i par :

$p_i = \{x \in X / d(x_i, x) \leq d(x_k, x) \text{ pour tout } k = 1, 2, \dots, p\}$.

Fin pour tout

Pour tout $i = 1, 2, \dots, p$

Déterminer le médian m_i de P_i par :

$m_i = \operatorname{argmin}_{y_i \in P_i} \sum_{x \in P_i} W(x) d(y_i, x)$

Fin pour tout

poser : $M = \{m_1, m_2, \dots, m_p\}$

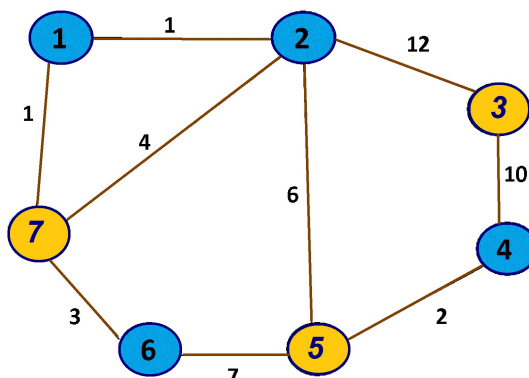
Fin tant que

Sortie : M

Fin de l'algorithme

Exemple Soit le graphe suivant :

Soient : $X = \{1, 2, 3, 4, 5, 6, 7\}$, $P = 3$, telle que le poids de chaque sommet est égale à 1.



supposons $S = 1, 2, 3$

1^{ère} itération :

$$P_1 = \{1, 6, 7\}$$

$$P_2 = \{2, 4, 5\}$$

$$P_3 = \{3\}$$

$$P_1 \begin{cases} C(1) = 1 + 4 = 5 \\ C(6) = 3 + 4 = 7 \\ C(7) = 1 + 3 = 4 \end{cases} \Rightarrow m_1^* = 7$$

$$P_2 \begin{cases} C(2) = 8 + 6 = 14 \\ C(4) = 2 + 8 = 10 \\ C(5) = 3 + 4 = 7 \end{cases} \Rightarrow m_2^* = 5$$

$$P_3 \begin{cases} C(3) = 0 \end{cases} \Rightarrow m_3^* = 3$$

$$M = \{7, 5, 3\}$$

$$S = \{7, 5, 3\}$$

2^{ème} itération :

$$P_1 = \{7, 1, 6, 2\}$$

$$P_2 = \{5, 4\}$$

$$P_3 = \{3\}$$

$$P_1 \begin{cases} C(1) = 1 + 1 + 4 = 6 \\ C(2) = 1 + 2 + 5 = 8 \\ C(6) = 4 + 5 + 3 = 12 \\ C(7) = 1 + 2 + 3 = 6 \end{cases} \Rightarrow m_1^* = 7$$

$$P_2 \begin{cases} C(4) = 2 \\ C(5) = 2 \end{cases} \Rightarrow m_1^* = 5$$

$$P_3 \begin{cases} C(3) = 0 \end{cases} \Rightarrow m_3^* = 3$$

$$M = \{7, 5, 3\}$$

S=M. Stop

Donc : M = {7, 5, 3}

4 Méthode exacte

Une méthode exacte permet de trouver une solution optimale à un problème donné. Toutefois, ces méthodes peuvent devenir rapidement couteuses en temps d'exécution, notamment pour les problèmes NP-difficiles. En effet, le temps de traitement et la complexité du problème sont généralement liés (plus c'est complexe, plus le temps d'exécution sera important). Ci-dessous, quelques méthodes exactes parmi les plus connues :

- Dualité Lagrangienne.

5 Dualité lagrangienne

Notons par

textbf X l'ensemble des solutions de (P) qui vérifient les contraintes (2.3), (2.4) et (2.5).

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

Soit $\lambda = (\lambda_j)_{j=1, \dots, n}$ le multiplicateur de Lagrange associé aux contraintes de type (2.2).

Le problème lagrangien[1] est donné par :

$$(L_\lambda) \begin{cases} v(L_\lambda) = \min(L(\lambda, x)) \\ x \in X \end{cases} \quad (3.1)$$

où L est la fonction de Lagrange.

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

Le dual lagrangien, (D), est donné par :

$$(D) \begin{cases} v(D) = v(L_{\lambda^*}) = \max v(L_\lambda) \\ \lambda \geq 0 \end{cases} \quad (3.3)$$

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

$$(L_\lambda)_{i=1, n} \begin{cases} v(L_\lambda)_i = \sum_{j=1}^n (d_{ij} - \lambda_j) x_{ij} \\ \text{sous les contraintes (2.4) et (2.5)} \end{cases} \quad (3.4)$$

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

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

Soit I_* l'ensemble des indices qui correspondent aux \mathbf{p} plus petites valeurs de δ_i :

$$I^* = \text{l'indice des } \mathbf{p} \text{ plus petites valeurs de } \delta_i \quad (3.6)$$

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

$$x_{ii}^* = \begin{cases} 1 & \text{si } i \in I^* \\ 0 & \text{sinon} \end{cases} \quad (3.7)$$

Et pour $i \neq j$

$$x_{ij}^* = \begin{cases} 1 & \text{si } i \in I^* \text{ et } d_{ij} - \lambda_j \leq 0 \\ 0 & \text{sinon} \end{cases} \quad (3.8)$$

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

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

6 L'algorithme du sous-gradient

Soit $x = (x_{ij})_{ij} \in [0, 1]^{n^2}$ et notant la fonction objectif de **(P)** par :

$$f(x) = \sum_{i=1}^n \sum_{j=1}^n d_{ij} \cdot x_{ij} \quad (3.10)$$

Et la contrainte (2.3) par :

$$g_j(x) = 1 - \sum_{i=1}^n x_{ij} \text{ pour } j = 1, 2, \dots, n \quad (3.11)$$

On remarque que la fonction $v(L_{(\cdot)}) : \lambda \longrightarrow v(L_\lambda)$, donnée par la relation (3.9), n'est pas nécessairement différentielle. Mais elle est concave et continue sur l'ensemble convexe fermé \mathbb{R}_+^n , d'où la fonction $v(L_{(\cdot)})$ a un sous-gradient ζ pour chaque $\bar{\lambda} \in \mathbb{R}_+^n$, c'est à dire

$$v(L_{(\lambda)}) \leq v(L_{(\bar{\lambda})}) + \zeta(\lambda - \bar{\lambda}) \quad (3.12)$$

La méthode du sous-gradient est fréquemment utilisé lorsqu'un sous-gradient peut-être facilement calculé. c'est notre cas : le sous gradient est donné par :

$$\zeta = g(x^*(\bar{\lambda})) \quad (3.13)$$

D'où $x^*(\bar{\lambda})$ est la solution du problème $(L_{\bar{\lambda}})$. cette procédure consiste en une méthode itérative où les multiplicateurs lagrangiens sont mis à jour comme suit :

$$\lambda_j^{(k+1)} = \max(0, \lambda_j^{(k)} + \theta^{(k)} \cdot \zeta_j^{(k)}) \text{ pour } j = 1, 2, \dots, n \quad (3.14)$$

$\lambda^{(k+1)}$ est la projection de $\lambda^{(k)} + \theta^{(k)} \cdot \zeta^{(k)}$ dans \mathbb{R}_+^n , $\zeta^{(k)}$ est le sous-gradient de la fonction $\lambda \longrightarrow v(L_\lambda)$ au point $\lambda^{(k)}$ est donné par :

$$\zeta_j^{(k)} = 1 - \sum_{i \in I^*} x_{ij}^* \text{ pour tout } j = 1, 2, \dots, n \quad (3.15)$$

D'où (x_{ij}^*) est défini par (3.7) et (3.8), $\theta^{(k)}$ est le pas de déplacement qui défini comme suit :

$$\theta^k = \frac{\pi(v(D) - v(L_{(\lambda^k)}))}{\|\zeta^{(k)}\|^2} \quad (3.16)$$

Où $\pi \in]0, 2[$

Soit l'algorithme du sous-gradient [1] de façon formelle :

Algorithme 1 : sous-gradient

Initialisation :

Déterminer une valeur initiale de Z_{ub} (borne supérieure de (P)).

(On peut utiliser l'heuristique de Tietz et Bart ou autre).

poser : $Z_{lb} = -\infty$

(Z_{lb} : Meilleure borne inférieure trouvée durant la procédure.)

Choisir une valeur initiale du multiplicateur Lagrangien $\lambda^{(0)}$

$\lambda_j^{(0)} = \min_{i \neq j} (d_{ij})$ pour tout $j = 1, 2, \dots, n$

Choisir $\epsilon > 0$ et le nombre d'itérations maximum $k_{max} > 0$:

Soit $k = 0$:

Tant que ($k < k_{max}$) :

Calculer : $I_{(k)}^*$, $(x_{ij}^*)^{(k)}$ et $V(L_{(\lambda^{(k)})})$

Donnés par les relations (3.6), (3.7), (3.8), (3.9) et (3.12)

Actualiser Z_{lb} en posant : $Z_{lb} = \max(Z_{lb}, V(L_{(\lambda^{(k)})}))$

Calculer : $Z = \sum_{j=1}^n \min_{i \in I_{(k)}^*} (d_{ij})$

Mettre à jour la valeur de Z_{ub} par : $Z_{ub} = \min(Z, Z_{ub})$

Si ($(Z_{ub} - Z_{lb}) > \epsilon$)

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 :

$\lambda_j^{(k+1)} = \max(0, \lambda_j^{(k)} + \theta^{(k)} \zeta_j^{(k)})$ pour tout $j = 1, 2, \dots, n$

$k = k + 1$

Sinon

Soit $k = k_{max}$ (La solution courante est optimale).

Fin Si.

Fin tant que

Fin de l'algorithme.

Remarque

L'application de la méthode du sous-gradient ne garantit pas l'optimalité de la solution et une procédure de sélection et évaluation progressive (Brunch and bound) est nécessaire pour l'obtention d'une solution optimale. Pour des problèmes de grandes tailles (de l'ordre de milliers de nœuds) l'application de méthodes S.E.P. (brunch and Bound) est très couteuse. Très souvent cette dernière est utilisée pour avoir une "bonne solution" qui servira de point de départ à d'autres méthodes approchées.

Chapitre 4

Cas particuliers

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 karivet Hakimi ont donné un algorithme polynomial, de type programmations dynamique, Pour le problème du p-médian.

En 1971, Goldman a proposé un algorithme très efficace pour la résolution du 1-médian dans un arbre. Oudjit et Mirchandani ont donné un algorithme de résolution du 2-médian dans un arbre cet algorithme a été amélioré par la suite. Dans ce mémoire nous nous penchons particulièrement sur le problème du 1-médian dans un cycle, une roue et un cactus.

Définition d'un Arbre

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

Exemple

le graphe suivant est un arbre d'ordre 7 :

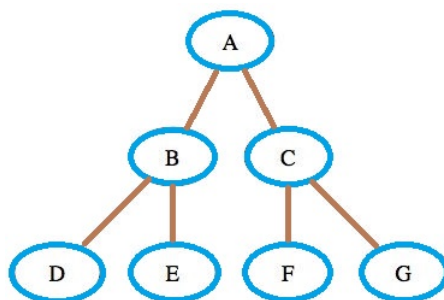


FIGURE 4.1 – Un arbre

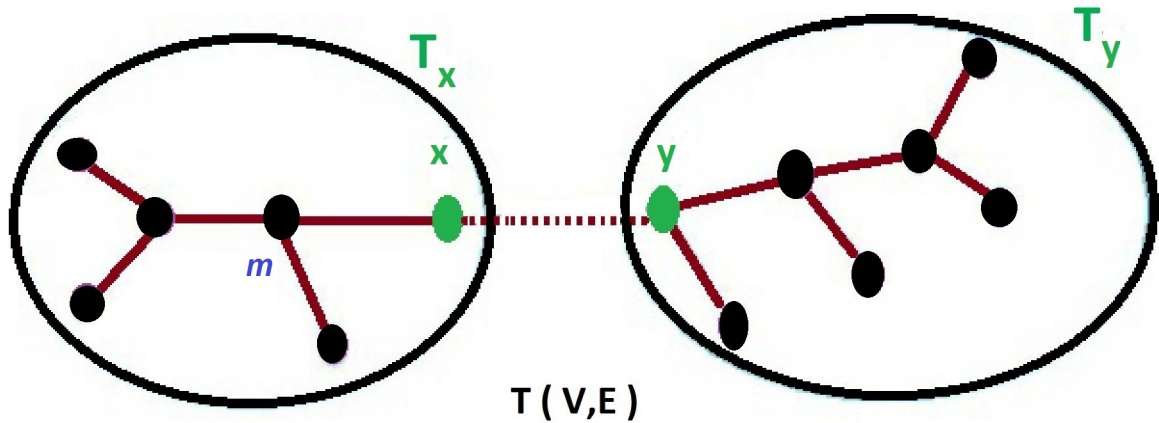
1 Problème du 1-médian dans un arbre

Nous présentons ci-dessous, l'algorithme de Goldman (1971)[2] qui permet la résolution du problème du 1-médian dans un arbre.

Cet algorithme, connu comme algorithme de la majorité de Goldman, est basé sur deux lemmes qui stipulent que la localisation du médian ne dépend pas des distances entre les

sommets mais uniquement des poids des sommets du graphe.

Soient $T=(V, E)$ un arbre et $e=(x, y)$ une arête de E . On note par $T_x = (V_1, E_1)$ et $T_y = (V_2, E_2)$ les deux sous arbre de T , contenant respectivement les sommets x et y , obtenus par suppression de l'arête (x, y) .



Soit m un 1-médian de T , et pour tout sous-arbre $H=(X, U)$ de T on définit le poids de H par : $W(H) = \sum_{x \in X} W(x)$

1.1 Lemme

Les deux propositions suivantes sont équivalentes[6] :

1. $m \in T_x$
2. $W(T_x) \geq W(T_y)$

1.2 Lemme

Si $W(T_x) \geq W(T_y)$ alors[6] :

Chercher un 1-médian m dans T équivaut à déterminer le 1-médian m dans T_x où le poids du sommet x sera augmenté du poids $W(T_y)$ du sous-arbre T_y .

1.3 Corollaire

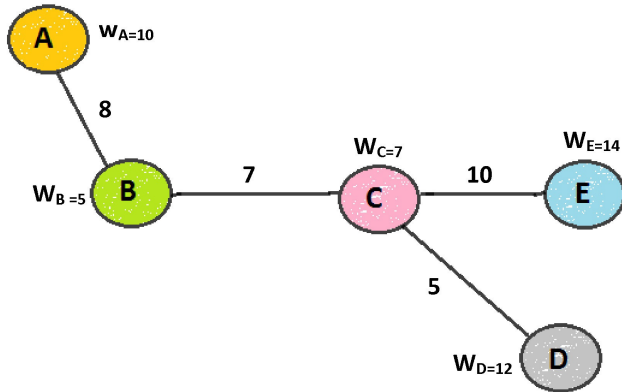
Les deux propositions suivantes sont équivalentes [6] :

1. $m \in T_x$
2. $W(T_x) \geq W(T)/2$

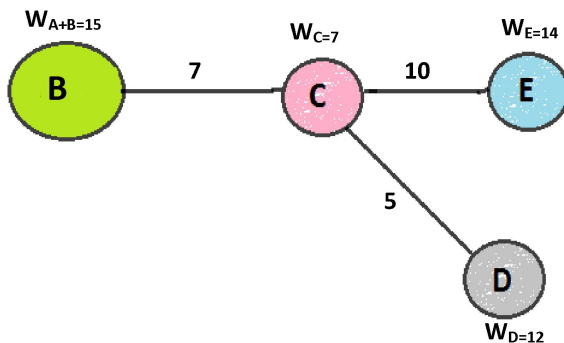
L'algorithme de Goldman est une méthode qui est basée sur la suppression d'arête. On démarre à partir d'un sommet pendant v . Si le poids de ce sommet est supérieur à la moitié de poids total de l'arbre, alors v est le médian cherché. Sinon contracter v avec le sommet adjacent, u , de v , et augmenter le poids de v du poids de u . On continue le processus jusqu'à obtention du médian.

1.4 Exemple

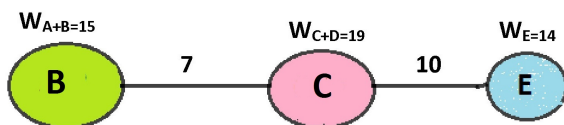
Ci-dessous, nous donnons un exemple d'arbre d'ordre 5 dont le poids total est de 48.



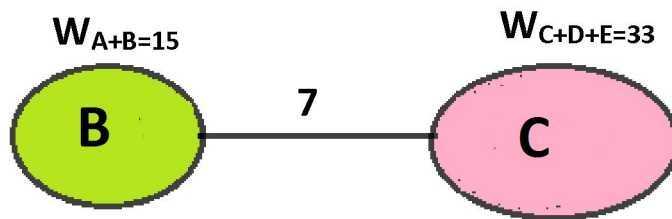
On démarre à partir du sommet **A** dont le poids est égal à 10. 10 étant inférieure à $48/2 = 24$. Donc on supprime le sommet **A** et le poids de **B** (sommet adjacent à **A**) sera augmenté de 10 (poids de **A**) pour atteindre la valeur de 15.



Le sommet **D** (par exemple) est un sommet pendent de poids égal à 12, donc inférieure à 24 (moitié de poids total de **T**), Alors le sommet **D** sera Supprimé et le poids du sommet **C** (sommet adjacent à **D**) sera augmenté de 12 pour devenir 19.



Le sommet **E** est pendent, son poids est de 14, donc inférieur à 24. **E** sera supprimé et le poids de **C** sera augmenté de 14 pour atteindre le poids de 33.



On s'aperçoit alors que le poids de C est égal à 33, valeur qui est supérieure à 24, d'où le sommet C est le sommet cherché : C'est le médian de l'arbre.

Comme conséquence directe de ces résultats, on en déduit un algorithme (du à Goldman) [6] en $O(n)$ de recherche d'un médian m dans un arbre. Cet algorithme s'énonce comme suit :

ALGORITHME DE GOLDMAN

Entrée : $T=(V, E), W$.

Initialisation

$$W(T) = \sum_{k=1}^n W(v_k) \text{ (poids de l'arbre T).}$$

Trouvé \leftarrow 0

Tant que [(|V| \geq 2) et trouvé = 0]

choisir une feuille e.

Si ($W(e) < W(T)/2$)

 Soit v le sommet adjacent à e.

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

Sinon

 Trouvé \leftarrow 1.

$m \leftarrow e$.

Fin si

Fin Tant que

Si [(|V| = 1)]

 Le sommet de V est le médian cherché.

Fin si

Sortie : m

Fin de l'algorithme

Nous avons programmé l'algorithme de Goldman en langage Matlab. Nous avons testé cet algorithme sur des exemples que nous avons généré aléatoirement. Ci-dessous nous donnons un algorithme de génération aléatoire d'un arbre.

Algorithme de génération d' un arbre (Gen-A)

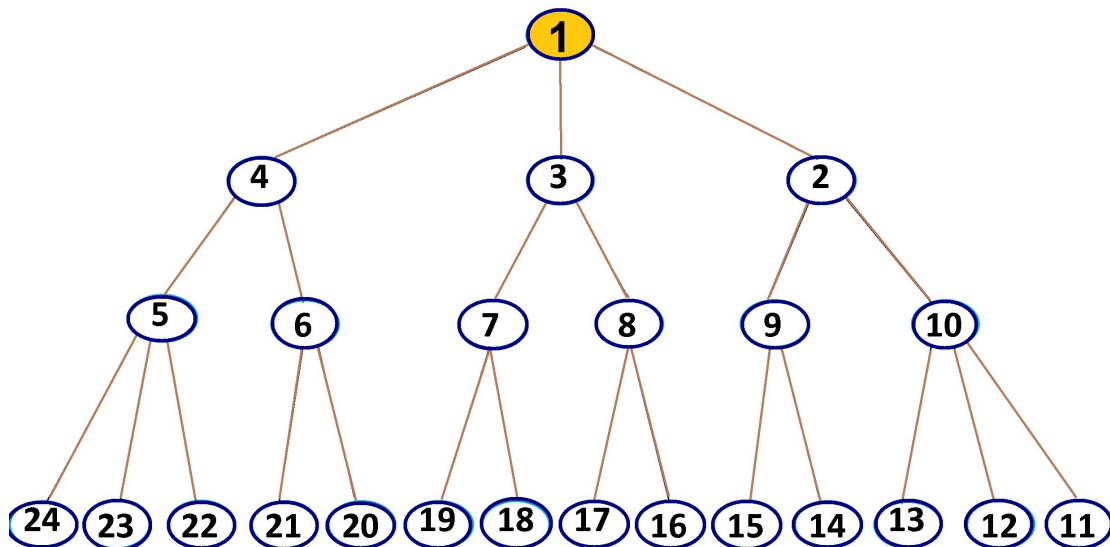
Début

-**introduire** de façon équiprobable les sommets de T dans une liste V_a .
b-Prendre p aléatoire dans $[1,N]$.
Pour $k=1,N$.
 Prendre un sous ensemble E_k de V_a de façon aléatoire et soit $h_k = |E_k|$.
 choisir un sommet quelque r_k de E_k .
 Faire $V_a \leftarrow V_a - E_k$.
 Si $|V| = 0$ aller en **a**.
Fin pour
a Continue.
Pour $k=1,p$.
 Introduire r_k dans V_a .
Fin pour
Si la liste V est vide aller en **b**.

Fin

Exemple

Soit $T(V,E)$ l'arbre telle que : $V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24\}$
 $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{22}, e_{23}\}$.
On cherche le 1-médian de cet arbre.



Déterminer le 1-médian de l'arbre T (voire la figure ci-dessus) où le coût de chaque arête égale à 1 et le poids de chaque sommet est définit comme suit :

- | | |
|----------|------------|
| $W(1)=1$ | $W(13)=13$ |
| $W(2)=2$ | $W(14)=14$ |
| $W(3)=3$ | $W(15)=15$ |
| $W(4)=4$ | $W(16)=16$ |
| $W(5)=5$ | $W(17)=17$ |
| $W(6)=6$ | $W(18)=18$ |
| $W(7)=7$ | $W(19)=19$ |

$W(8)=8$ $W(20)=20$
 $W(9)=9$ $W(21)=21$
 $W(10)=10$ $W(22)=22$
 $W(11)=11$ $W(23)=23$
 $W(12)=12$ $W(24)=24$

On donne les résultats sous forme de tableau suivant :

	e	W(e)	m	v	W(v)	V=
1^{ère} itération	24	W(24)=24		5	W(5)=29	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23}
2^{ème} itération	23	W(23)=23		5	W(5)=52	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22}
3^{ème} itération	22	W(22)=22		5	W(5)=74	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21}
4^{ème} itération	21	W(21)=21		6	W(6)=27	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
5^{ème} itération	20	W(20)=20		6	W(6)=47	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19}
6^{ème} itération	19	W(19)=19		7	W(7)=26	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18}
7^{ème} itération	18	W(18)=18		7	W(7)=44	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}
8^{ème} itération	17	W(17)=17		8	W(8)=25	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16}
9^{ème} itération	16	W(16)=16		8	W(8)=41	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}
10^{ème} itération	15	W(15)=15		9	W(9)=24	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14}
11^{ème} itération	14	W(14)=14		9	W(9)=38	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
12^{ème} itération	13	W(13)=13		10	W(10)=23	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}
13^{ème} itération	12	W(12)=12		10	W(10)=35	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}
14^{ème} itération	11	W(11)=11		10	W(10)=46	V= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
16^{ème} itération	10	W(10)=46		2	W(2)=48	V= {1, 2, 3, 4, 5, 6, 7, 8, 9}
17^{ème} itération	9	W(9)=38		2	W(2)=86	V= {1, 2, 3, 4, 5, 6, 7, 8}
18^{ème} itération	8	W(8)=41		3	W(3)=44	V= {1, 2, 3, 4, 5, 6, 7}
19^{ème} itération	7	W(7)=44		3	W(3)=88	V= {1, 2, 3, 4, 5, 6}
20^{ème} itération	6	W(6)=47		4	W(4)=51	V= {1, 2, 3, 4, 5}
21^{ème} itération	5	W(5)=74		4	W(4)=125	V= {1, 2, 3, 4}
22^{ème} itération	4	W(4)=125		1	W(1)=126	V= {1, 2, 3}
23^{ème} itération	3	W(3)=88		1	W(1)=214	V= {1, 2}
24^{ème} itération	2	W(2)=86	1	1	W(1)=300	V= {1}

Donc le médian de l'arbre **T** est le sommet 1.

2 Problème du 2-médian dans un Arbre

Le principe de l'algorithme de la recherche du 2-médian dans un arbre est basé sur la méthode de suppression d'arêtes : la méthode consiste à considérer $(n - 1)$ suppressions d'arêtes . Pour chaque suppression on engendre deux sous-arbres T_a et T_b dans lesquels on résout, par l'algorithme de Goldman, le problème de la recherche du 1-médian. La méthode nécessite un nombre d'opérations de l'ordre de $O(n^3)$: Le but de l'algorithme que nous allons présenter ci-dessous est de diminuer au maximum le nombres de suppressions d'arêtes pour en écarter des suppressions inutiles[11].
Soit de manière formelle la méthode de suppression d'arêtes :

2.1 Méthode de suppression d'arêtes

Soit $l \in E$ et soit T_a et T_b les deux sous-arbre de T obtenus par la suppression de l'arête l . Par l'algorithme de Goldaman, on détermine le médian m' de T_a et m'' de T_b pour déterminer, $J(m', m'')$, la valeur de l'objectif du couple $(m'$ et m'') donner par :

$$J(m_1, m_2) = \sum_{i \in T_1} w(i).d(m_1, i) + \sum_{i \in T_2} w(i).d(m_2, i)$$

La méthode de suppressions d'arête consiste à considérer tous les suppressions d'arête. Le 2-médian (m_1, m_2) de T est le couple de sonnets m', m'' qui réalise le minimum de $J(m', m'')$.

2.2 Propriétés du 2-médian

Soit $T = (V, E)$ un arbre, m médian de T et (m_1, m_2) un 2-médian de T .

2.2.1 Théorème[11]

Soit m un 1-médian de T et (m_1, m_2) un couple 2-médian de T et on note par $P[m_1, m_2]$ un plus court chemin de m_1 à m_2 alors $m \in P[m_1, m_2]$.

Preuve

On utilise un raisonnement par l'absurde.

Supposons que $m \notin P[m_1, m_2]$

On a deux cas :

(i) $m \in T_a$.

(ii) $m \in T_b$.

cas (i) : $m \in T_a$

d'après lemme 1 on a :

$$W(T_m) \geq W(T - T_m) > W(T_a - T_m) \quad [m \in T_m \text{ et } T_a \subset T$$

mais $W(T_a - T_m) \geq W(T_m) \quad [m(T_a) \in T_a - T_m]$.

Donc contradiction.

On fait le même raisonnement pour $m \in T_b$.

2.2.2 Corollaire [11]

Soit m un 1-médian de T et (m_1, m_2) un couple 2-médian de T alors

$$m_1 \notin P[m, m_2] - \{m\}$$

2.2.3 Corollaire [11]

Soient $m, m' \in V$ deux 1-médians distincts de T et (m_1, m_2) un couple 2-médian de T alors

$$P[m, m'] \subset P[m_1, m_2].$$

2.2.3 Corollaire[11]

Si m_1 sert uniquement $T_1 \subset T_{v_1}$ alors :

- $W(T_{v_1}) \leq \frac{1}{2}W(T - T_1) \iff m_2 = m.$
- $W(T_{v_2}) \geq \frac{1}{2}W(T - T_1) \iff m_2 \in T_{v_2}.$
- $W(T_{v_2}) = W(T_j)$ pour $v_j \in V_m - \{v_1, v_2\} \implies m_2 = m$
- $W(T_{v_2}) \leq \sum_{v_j \in S} W(T_{v_j})$ pour $S \subset V_m - \{v_1, v_2\} \implies m_2 = m.$
- $W(T_{v_2}) < W(T_{v_1} - T_1) \implies m_2 = m.$

2.3 Algorithme de Oudjit-Mirchandani[3]

Pour déterminer le 2-médian, on applique la méthode de suppression d'arête, on considère $(n-1)$ suppressions d'arêtes.

En exploitant les propriétés du 1-médian m et 2-médian $(m_1$ et $m_2)$ dans un arbre (cités ci-dessus), Oudjit et Mirchandani arrivent à réduire le nombre de suppressions d'arêtes.

Soit $E_j = \{e_{ij}\}$, l'ensemble des feuilles de T_{v_j} , $i = 1, 2, \dots, |E_j|$; $j = 1, 2, \dots, f.$

Dans cette procédure, on suppose que le médian m est un candidat pour être un des deux 2-médian m_1 , et pour déterminer l'autre médian m_2 , on considère pour toute feuille e_{ij} la partition associée Q_{ij} que l'on définira par :

$$Q_{ij} = \{v \in T / d(e_{ij}, v) \leq d(v, m)\}.$$

Soit T_{ij} le sous-arbre de T ne contenant pas m et est constitué par Q_{ij}^1 et soit m_{ij}^1 son médian (en utilisant la procédure de Goldman).

On répété ce même procédure pour le couple (m, m_{ij}^1) pour trouver Q_{ij}^2 et T_{ij}^2 puis m_{ij}^2, \dots etc. Le processus s'arrête quand la partition ne change pas.

Soit (m, m_{ij}) le couple de sommet obtenu à la fin de l'algorithme et soit Q_{ij} est défini par :

$$Q_{ij} = \{v \in T / d(v, m_{ij}) \leq d(v, m)\}.$$

Soit x_{ij} le sommet de Q_{ij} qui le plus proche de m . Soit T_{ij} le sous-arbre formé par les sommets de Q_{ij} . Le processus examine uniquement les suppression d'arête l se trouvant sur $p[m, x_{ij}]$ pour toute feuille e_{ij} .

Nous aurons à examiner tous les suppression d'arêtes qui se trouvent sur T^* qui est défini comme suit :

$$T^* = \cup_{i,j} p[m, x_{ij}] \quad i = 1, 2, \dots, f$$

L'algorithme s'énonce comme suit :

Étape 0 : [initialisation]

Déterminer un médian \mathbf{m} de \mathbf{T} (en utilisant l'algorithme de Goldman), on considéré \mathbf{T} comme arborescence de racine de \mathbf{m} .

Définir T_{v_j} , pour tout $v_j \in V_m$.

Poser :

1) $N \leftarrow V_m$.

2) $T^* \leftarrow \emptyset$.

3) les partition Q_{ij}^0 sont les arêtes adjacentes aux feuilles e_{ij} .

Étape 1 : [choix de sous-arbre T_{v_j}]

Si $N = \emptyset$, alors aller à l'étape 5.

Sinon

1) Choisir $v_i \in N$.

2) Faire : $E \leftarrow E_j$.

3) Faire : $N \leftarrow N - v_j$.

Fin si

Étape 2 : [choix d'une feuille e_{ij}]

Si $E = \emptyset$, alors aller à l'étape 1.

Sinon

1) Prendre $e_{ij} \in E_j$.

2) Faire : $E \leftarrow E_j - e_{ij}$.

3) Faire : $x \leftarrow e_{ij}$.

4) poser $h \leftarrow 1$.

Fin si

Étape 3 : [détermination de Q_{ij}^h et de sa partition, Q_{ij} optimale associée].

Déterminer la partition Q_{ij}^h associe aux sommets candidat (\mathbf{m}, \mathbf{x}) .

Si $Q_{ij}^h = Q_{ij}^{h-1}$, alors la partition optimale $Q_{ij} = Q_{ij}^h$.

Déterminer le sommet x_{ij} correspondant.

Poser $T^* \leftarrow T^* \cup p[m, x_{ij}]$.

Aller à l'étape 2.

Sinon

Soit T_{ij}^h le sous-arbre qui ne contient pas \mathbf{m} , formé par Q_{ij}^h .

Fin si

Étape 4 : [Détermine un 1-médian de T_{ij}^h] .

En utilisant l'algorithme de Goldman, détermine un médian m_{ij}^h de T_{ij}^h .

$x \leftarrow m_{ij}^h$.

$h \leftarrow h + 1$.

Aller à l'étape 3.

Étape 5 : [Détermination de m_1, m_2] .

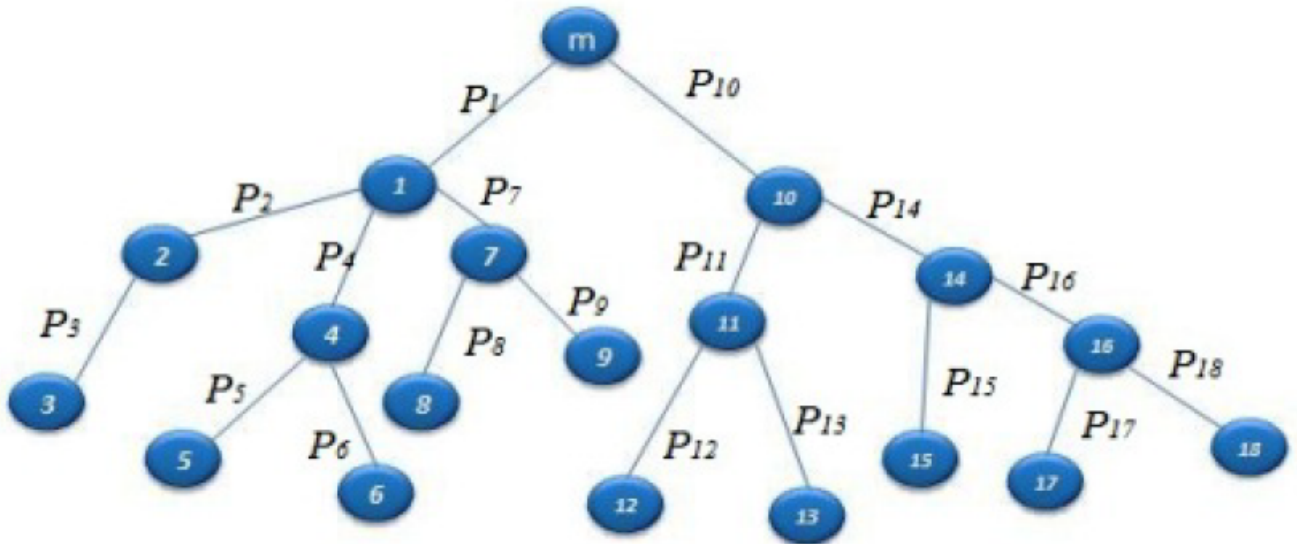
Applique la procédure de suppression des arêtes sur T^* et puis déterminer la suppression optimale et par suite le 2-médian.

Ci dessous, un exemple est donné pour illustrer cet algorithme (exemple cite par Oudjit et Mirchandani)

Exemple

Déterminer le 2-médian d'un arbre T de racine son médian m (voir figure) où le poids de chaque nœud égale à 1 et le coût de chaque arête est égale à 1 aussi. Le poids total de T est de 19, le nombre d'arête est égale à 18.

L'application de la méthode générale de suppression des arêtes nécessite le traitement de 18 suppressions d'arête.



La suppression optimale est P_{10} et les 2-médian est $\{v_1, v_{14}\}$.

Ci dessous, on donnera toutes les étapes de la procédure de Oudjit-Mirchandani : $V_m = \{v_1, v_{10}\}$, le médian de T est noté par m . Par commodité, on notera par $T_{v_i}, \forall v_i$, le sous-arbre de T contenant v_i (v_i étant le sommet de T_{v_i} le plus proche de m)

Supposons qu'à l'étape 1 $T_{v_{10}}$ est choisi.

Soit : $e_{1,10} \equiv x \leftarrow v_{18}$.

Étape 2 :

• $\{m, x\} \leftarrow \{m, v_{18}\}$.

• $Q_{10}^1 \leftarrow P_{14}$.

• $T_{1,10}^1 \leftarrow T_{14}$.

Étape 3 :

• $m_{1,10}^1 \leftarrow v_{16}$.

• $x \leftarrow v_{16}$.

Étape 3 :

• $Q_{1,10}^2 \leftarrow P_{14}$.

• $Q_{1,10} \leftarrow P_{14}$.

• $x_{ij} \leftarrow v_{14}$.

• $T^* \leftarrow P[m, v_{14}]$.

Étape 2 :

• $e_{2,10} \equiv x \leftarrow v_{12}$.

Étape 3 :

• $\{m, x\} \leftarrow \{m, v_2\}$.

• $Q_{2,10}^1 = P_{11}$.

• $T_{2,10}^1 \leftarrow T_{v_{11}}$.

Étape 4 :

- $m_{2,10}^1 \leftarrow v_{11}$.

- $x \leftarrow v_{11}$.

Étape 3 :

- $\{m, x\} \leftarrow \{m, v_{11}\}$.

- $Q_{2,10}^1 \leftarrow P_{10}$.

- $T_{2,10}^1 \leftarrow T_{v_{10}}$.

Étape 4 :

- $m_{2,10}^1 \leftarrow v_{14}$.

- $x \leftarrow v_{14}$.

Étape 3 :

- $Q_{2,10}^3 \leftarrow P_{10}$.

- $Q_{2,10} \leftarrow P_{10}$.

- $x_{ij} \leftarrow v_{10}$.

Finalement on obtient : $T^* \leftarrow P[m, v_{14}] \cup P[m, v_{10}]$. Qui contient les sommets v_1, m, v_{10} , et v_{14} et les trois arêtes qui les relient.

En comparant les valeur de la fonction objective pour les suppression des arêtes dans T^* , on trouve que P_{10} est la suppression optimale des arêtes aboutissant aux 2-médians qui sont localisés à v_1 et v_{14} .

3 Problème du 1-médian dans un cycle

3.1 Définition d'un cycle

Les cycles, ou n -cycles, forment une classe importante de graphes. Un cycle (au sens graphe) C_n est constitué d'un unique cycle élémentaire de longueur n (pour $n \geq 3$). C'est un graphe connexe (que l'on suppose non-orienté) d'ordre n sur 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 $\{e = (v_i, v_{i+1}) / i = 1, 2, \dots, n-1\}$ par $E(C_n)$.
- w_i : le poids du sommet v_i ($i = 1, 2, \dots, n$).
- Et par $l(v_i, v_j)$ le coût de l'arête (v_i, v_j) .

Dans certains cas les indices des sommets peuvent dépasser n et pour simplifier les notations, on prendra (par convention) l'indice i modulo (n) : par exemple le sommet v_{n+1} correspond à v_1 . La longueur du cycle est donnée par :

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

L'algorithme de recherche du 1-médian dans un cycle est constitué de deux parties :

Le calcul de $d(v_i, v_j)$ $i = 1, 2, \dots, n$; peut être obtenu en $O(n)$.

Ensuite, en utilisant un algorithme en $O(n)$, on obtient la valeur de l'objectif de tous les sommets du cycle.

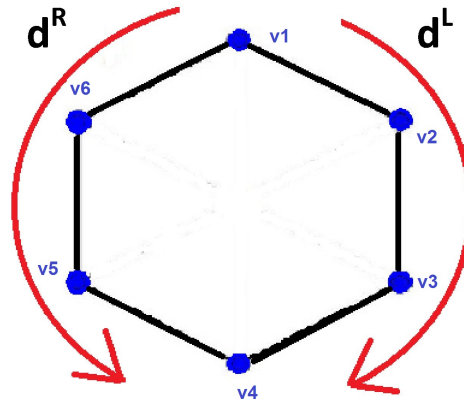
3.2 Partie 1 : calcul de la matrice des plus courtes distances

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

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

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

$d^l(v_i, v_j)$ et $d^r(v_i, v_j)$ correspondent 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 (voir la figure ci-dessous)



La longueur du cycle, L , est définie par : $L = d^l(v_i, v_j) + d^r(v_i, v_j)$.

Nous avons $d^l(v_i, v_j) = d^r(v_j, v_i)$ (pour tous les indices i et j variant de $1, 2, \dots, n$) et le calcul de $d(v_i, v_j)$ est donné par : $d(v_i, v_j) = \min(d^l(v_i, v_j), d^r(v_i, v_j)) \forall i, j = 1, 2, \dots, n$.

Remarque : $d^r(v_i, v_j) = L - d^l(v_i, v_j)$ où L est la longueur de cycle.

3.3 Partie 2 : calcul de la valeur de l'objectif d'un sommet du cycle

Après avoir calculé toutes les distances, nous pouvons calculer, $f(v_i)$, la valeur de l'objectif de chaque sommet v_i du cycle C par :

$$f(v_i) = \sum_{j=1}^n d(v_i, v_j) \cdot w(v_j)$$

Nous avons programmé l'algorithme de recherche du 1-médian dans un cycle en langage Matlab. Nous avons testé cet algorithme avec le procédure de sous gradient sur des exemples que nous avons généré aléatoirement. Ci-dessous nous donnons un algorithme de génération aléatoire d'un cycle d'ordre n .

Algorithme de génération d'un cycle (Gen-C)

```

Entrée : n, k
Faire : pour tout i=1,n
    Faire : pour tout j=1,n
        A(i,j) ← ∞
    Fin pour
Fin pour
Faire : pour tout i=1,n -1
    choisir : aléatoirement la valeur de A(i,i+1) dans l'intervalle [1,k]
    A(i+1,i) ← A(i,i+1)
Fin pour
choisir : aléatoirement la valeur de A(1,n) dans l'intervalle [1,k]
A(n,1) = A(1,n)
Sortie : A
Fin algorithme
    
```

4 Problème du 1-médian dans une roue

4.1 Introduction

Dans cette partie nous présentons un algorithme de recherche du 1-médian dans une roue : L'algorithme détermine de manière récursive la matrice des plus courtes distances ainsi que les valeurs des fonctions objectif de chaque sommet en un temps raisonnable.

4.2 Définition de la Roue

Une roue W_n est un graphe d'ordre $n(n \geq 4)$ constitué d'un cycle C_{n-1} (formé de $n - 1$ sommets) et d'un sommet v_0 "centre" qui est relié à tous les autres sommets du cycle C_{n-1} : Une roue est donc un graphe d'ordre n et ayant $2(n - 1)$ arêtes[8]. Les arêtes qui relient le centre v_0 au reste des sommets du cycle C_{n-1} sont appelées rayons.

Exemple

soit $W_4, W_5, W_6, W_7, W_8, W_9$ des roues d'ordre 4, 5, 6, 7, 8, 9.

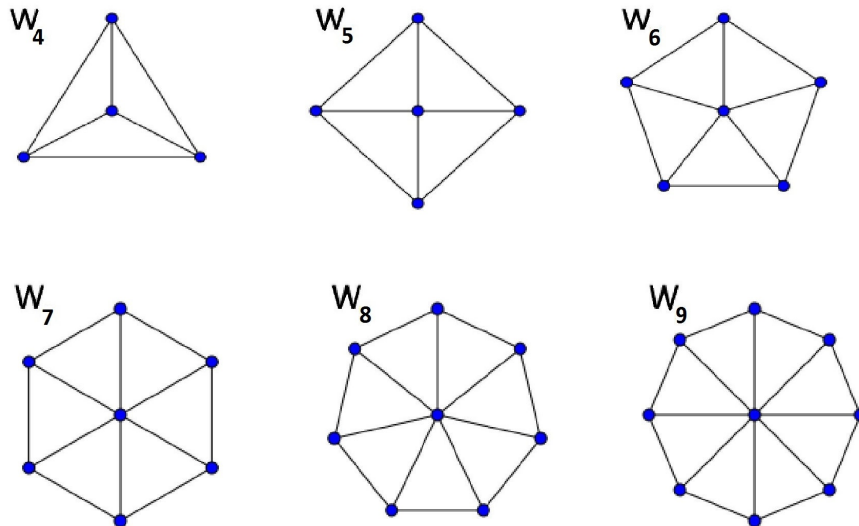


FIGURE 4.2 – Exemple des roues

4.3 Problème du 1-médian dans une roue[8]

Soit $W_{n+1} = (V(W_{n+1}), E(W_{n+1}))$ une roue.

On note par :

- $C_n = (V(C_n), E(C_n))$ le graphe constitué du cycle.
- $V(W_{n+1}) = \{v_0, v_1, v_2, \dots, v_n\}$.
- $V(C_n) = \{v_1, v_2, \dots, v_n\}$.

L'ensemble des arêtes de $E(W_{n+1})$ peut être partitionné en 2 sous ensembles :

- $E(C_n) = \{(v_i, v_{i+1}) / i = 1, 2, \dots, n - 1\} \cup \{(v_n, v_1)\}$: arêtes du cycle.
- $E(S) = \{(v_0, v_i) / i = 1, 2, \dots, n\}$ (appelé ensemble des rayons)

On note par : w_i le poids du sommet v_i ($i = 0, 1, 2, \dots, n$) et par $l(v_i, v_j)$ le coût de l'arête (v_i, v_j) .

Dans certains cas les indices des sommets peuvent dépasser n et pour simplifier les notations on prendra (par convention) l'indice i modulo n (par exemple le sommet v_{n+1} correspond à v_1).

La longueur du cycle est définie par :

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

L'algorithme de recherche du 1-médian dans une roue est composé de deux parties. D'abord, on montre que $d(v_i, v_0)$ ($i = 1, 2, \dots, n$) peut être calculée en $O(n)$. Ces distances sont utilisées pour le calcul de $f(v_0)$, valeur de l'objectif de v_0 . Puis, une procédure, en $o(n)$, est donnée pour trouver la valeur de l'objectif, $f(v_i)$ ($i = 1, 2, \dots, n$), de chaque sommet du cycle.

Enfin, la valeur de l'objectif, $f(v_0)$, est comparée aux valeurs des objectifs, $f(v_i)$ ($i = 1, 2, \dots, n$), des autres sommets pour déterminer la solution optimale (c'est à dire le 1-médian).

1- Calcul de la valeur de l'objectif, $f(v_0)$, de v_0

Pour calculer

$$f(v_0) = \sum_{i=1}^n w_i \cdot d(v_i, v_0)$$

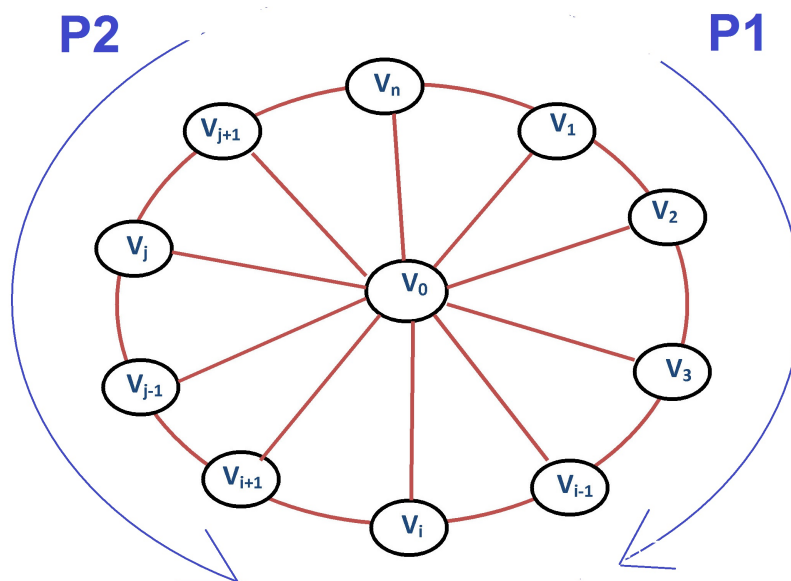
Nous devons d'abord, calculer la longueur d'un plus court chemin entre v_i et v_0 pour tout $i = 1, 2, \dots, n$.

Le plus court chemin de v_i à v_0 est donné soit sous la forme :

$$P1 = (v_i, v_{i+1}, \dots, v_{i+r}, v_0) \text{ pour un certain } r \in \mathbb{N}$$

ou sous la forme :

$$P2 = (v_i, v_{i-1}, \dots, v_{i-s}, v_0) \text{ pour un certain } s \in \mathbb{N}$$



En effet, dans le plus court chemin de v_i à v_0 , les arêtes du cycle sont soit traversées dans le sens des aiguilles d'une montre ou dans le sens inverse et il n'y a qu'un seul rayon. Ceci conduit à l'idée d'introduire deux orientations différentes d'une roue, à savoir W^1 et

W^2 .

Dans le premier cas, tous les arêtes du cycle sont orientés dans le sens contraire des aiguilles d'une montre, alors que dans le second cas l'orientation des arêtes du cycle est inversée. Dans les deux cas, les arêtes qui ne sont pas dans le cycle sont dirigés vers v_0 . Soit $d^j(v_i, v_0)$ la longueur du plus court chemin de v_i à v_0 dans W^j pour $j = 1, 2$ et pour tout $i = 1, 2, \dots, n$.

$$d(v_i, v_0) = \min(d^1(v_i, v_0), d^2(v_i, v_0)) \quad (4.1)$$

Supposons (sans perte de généralité) que :

$$l(v_1, v_0) \leq l(v_i, v_0) \text{ pour } \text{tous } i = 2, \dots, n \quad (4.2)$$

On peut alors conclure que : $d(v_1, v_0) = d^1(v_1, v_0) = d^2(v_1, v_0) = l(v_1, v_0)$. De plus,

$$d^1(v_i, v_0) = \min(l(v_i, v_0), d^1(v_{i-1}, v_0) + l(v_i, v_{i-1})) \quad \forall i = 2, \dots, n$$

et

$$d^2(v_{n+1-i}, v_0) = \min(l(v_{n+1-i}, v_0), d^2(v_{n+2-i}, v_0) + l(v_{n+1-i}, v_{n+2-i})) \quad \forall i = 1, \dots, n-1$$

mène immédiatement à des algorithmes de temps linéaires qui calculent $d^1(v_i, v_0)$ et $d^2(v_i, v_0)$ pour tout $i = 1, \dots, n$.

Pour obtenir $d(v_i, v_0)$ on utilise l'équation (4.1).

Algorithme de recherche du valeur de l'objectif, $f(v_0)$, de v_0

Poser :

$$d_1(v_1, v_0) \leftarrow l(v_1, v_0)$$

$$d^2(v_1, v_0) \leftarrow l(v_1, v_0)$$

$$d(v_1, v_0) \leftarrow (l(v_1, v_0))$$

Faire pour $i=2, N$

$$d^1(v_i, v_0) \leftarrow \min(l(v_i, v_0), d^1(v_{i-1}, v_0) + l(v_i, v_{i-1}))$$

$$d^2(v_{N-i+2}, v_0) \leftarrow \min(l(v_{N-i+2}, v_0), d^2(v_{N-i+1}, v_0) + l(v_{N-i+2}, v_{N-i+1}))$$

Fin pour

$$f(v_0) \leftarrow w(v_1).d(v_1, v_0)$$

Faire pour $i=2, n$

$$d(v_i, v_0) \leftarrow \min(d^1(v_i, v_0), d^2(v_i, v_0))$$

$$f(v_0) \leftarrow f(v_0) + w(v_i).d(v_i, v_0)$$

Fin pour

Sortie : $F(v_0)$

Fin algorithme

2- Calcul de la valeur de l'objectif du sommet du cycle

2.1- Calcul de la valeur de l'objectif du sommet v_1

On suppose que les plus courts distances, $d(v_i, v_0)$ ($i=1, 2, \dots, n$), ont déjà été calculées auparavant. Donc on peut calculer $d^h(v_i, v_1)$ par :

$$d^h(v_i, v_1) = d(v_i, v_0) + d(v_0, v_1).$$

Il y a trois chemins possibles reliant v_i à v_1 :

- Le chemin, C_1 , qui passe par v_0 et sa longueur est égale à : $l(C_1) = d(v_i, v_0) + d(v_0, v_1)$.
- Le chemin, C_2 , qui ne passe pas par v_0 , qui relie v_i à v_1 et en parcourant le cycle dans le

sens trigonométrique. sa longueur est donnée par : $l(C_2) = \sum_{j=1}^{i-1} l(v_j, v_{j+1})$

- Enfin le chemin, C_3 , qui ne passe pas par v_0 , qui relie v_i à v_1 et en parcourant le cycle dans le sens des aiguilles d'une montre et sa longueur est donnée par : $l(C_3) = L - l(C_2)$.

La distance, $d(v_i, v_1)$, est donnée par : $d(v_i, v_1) = \min(l(C_1), l(C_2), l(C_3))$.

2.2- Calcul de la valeur de l'objectif du sommet v_i autres que v_1

Le calcul de la valeur de la fonction objectif des sommets v_i ($i=2,3,\dots,n$) est beaucoup plus sophistiqué pour obtenir une complexité en $O(n)$. Comme précédemment, il y a trois chemins distincts qui relient v_i à v_j : le chemin qui passe par v_0 , celui qui ne passe pas par v_0 et qui parcourt le cycle dans le sens trigonométrique et enfin le chemin qui ne passe pas par v_0 et qui parcourt le cycle de le sens des aiguilles d'une montre.

L'arborescence des plus courts chemins de raciné v_{i-1} est utilisée pour déterminer l'arborescence des plus courts chemins de raciné v_i .

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

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

$$d^r(v_i, v_j) = \sum_{k=j}^{i-1} l(v_k, v_{k+1}). \quad (4.4)$$

$$d^h(v_i, v_j) = d(v_j, v_0) + d(v_0, v_i). \quad (4.5)$$

$d^l(v_i, v_j)$ et $d^r(v_i, v_j)$ correspondent aux longueurs des deux chemins de v_j à v_i qui ne passe pas par v_0 .

La longueur, L , du cycle est égale à : $L = d^l(v_i, v_j) + d^r(v_i, v_j)$

Il est claire que : $d^l(v_i, v_j) = d^r(v_j, v_i)$ et $d(v_i, v_j) = \min(d^l(v_i, v_j), d^r(v_i, v_j), d^h(v_i, v_j))$.

On définit les ensembles suivants par :

$$L(i) = \{v_j : v_j \neq v_i, v_j \in V(C) \setminus \{v_i\} \mid d^l(v_i, v_j) \leq d^r(v_i, v_j), d^l(v_i, v_j) \leq d^h(v_i, v_j)\}. \quad (4.6)$$

$$H(i) = \{v_j \in V(C) \setminus \{v_i\} \mid d^h(v_i, v_j) < d^l(v_i, v_j), d^h(v_i, v_j) \leq d^r(v_i, v_j)\}. \quad (4.7)$$

$$R(i) = \{v_j \in V(C) \setminus \{v_i\} \mid d^r(v_i, v_j) < d^l(v_i, v_j), d^r(v_i, v_j) < d^h(v_i, v_j)\}. \quad (4.8)$$

Notons que les ensembles $L(i)$, $H(i)$ et $R(i)$ définis ci-dessus sont disjoints deux à deux et leur union est $V(C) \setminus \{v_i\}$. Cependant, les ensembles $L(i)$, $H(i)$ et $R(i)$ pourraient être vides pour certains $i = 1, \dots, n$.

En utilisant la définition de ces ensembles, il s'ensuit que $L(i) \cup R(i)$ ne contient que des sommets pour lesquels il existe un plus chemin de ces derniers à v_i qui ne contient pas v_0 . De plus, tous les ensembles définis par les équations (4.6), (4.7) et (4.8) ont une forme particulière parce que les graphes induits par chaque ensemble est un chemin.

Si les ensembles sont non vides, ils peuvent s'écrire sous la forme suivante :

$L(i) = \{v_{i+1}, v_{i+2}, \dots, v_{i+s}\}$ et $H(i) = \{v_{i-1}, v_{i-2}, \dots, v_{i-r}\}$ pour certains $s, r \in \mathbb{N}$.

En utilisant les propriétés de base du plus court chemin dans l'arbre, il s'ensuit que si $v_k \in L(i)$ et $v_k \neq v_{i+1}$ alors $v_k \in L(i+1)$. D'autre part, si $v_k \in H(i)$ alors $v_k \notin R(i+1)$, parce

que

$$d^r(v_{i+1}, v_k) = d^r(v_i, v_k) + l(v_i, v_{i+1}) \geq d^h(v_i, v_k) + l(v_i, v_{i+1}) \geq d^h(v_{i+1}, v_k).$$

Ceci implique que si $v_k \in H(i)$ et $v_k \neq v_{i+1}$ alors v_k est soit dans $H(i+1)$ soit dans $L(i+1)$. Si $v_k \in R(i)$ nous ne pouvons tirer aucune information, autrement dit si $v_k \neq v_{i+1}$ alors v_k peut être dans $R(i+1)$, $H(i+1)$ ou $L(i+1)$.

L'idée de l'algorithme est de mettre à jour ces trois ensembles de façon itérative. Le calcul de la valeur du fonction objectif est finalement basé sur ces mises à jour. Notons que si $f(v_i)$, la valeur de la fonction objectif de v_i , alors $f(v_{i+1})$, peut être obtenue par :

$$f(v_{i+1}) = f(v_i) \tag{4.9}$$

$$+ \sum_{v \in R(i) \cap R(i+1)} w(v) \cdot l(v_i, v_{i+1}) \tag{4.10}$$

$$- \sum_{v \in L(i) \cap L(i+1)} w(v) \cdot l(v_i, v_{i+1}) \tag{4.11}$$

$$+ \sum_{v \in H(i) \cap H(i+1)} w(v) \cdot (d(v_0, v_{i+1}) - d(v_0, v_i)) \tag{4.12}$$

$$+ \sum_{v \in H(i+1) \cap R(i)} w(v) \cdot (d^h(v_{i+1}, v) - d^r(v_i, v)) \tag{4.13}$$

$$+ \sum_{v \in L(i+1) \cap R(i)} w(v) \cdot (d^l(v_{i+1}, v) - d^r(v_i, v)) \tag{4.14}$$

$$+ \sum_{v \in L(i+1) \cap H(i)} w(v) \cdot (d^l(v_{i+1}, v) - d^h(v_i, v)) \tag{4.15}$$

$$+ d(v_i, v_{i+1}) \cdot (w_i - w_{i+1}) + (d(v_0, v_{i+1}) - d(v_0, v_i)) \cdot w_0 \tag{4.16}$$

Donc le calcul des différentes valeurs, $f(v_i)$, des objectif des sommets v_i se fait de manière récursif. En utilisant, avec plus de détail, les propriétés des ensembles $L(i)$, $H(i)$ et $R(i)$ qui sont aussi déterminés de manière récurrente, on arrive à trouver un algorithme de recherche du 1-médian dans une roue en $O(n)$.

Nous avons programmé l'algorithme de recherche du 1-médian dans une roue en langage Matlab. Nous avons testé cet algorithme avec le procédure de sous gradient sur des exemples que nous avons généré aléatoirement. Ci-dessous nous donnons un algorithme de génération aléatoire d'une roue d'ordre n .

Algorithme de génération d'une roue (Gen-R)

Entrée n, k

Faire pour tout $i=1, n$

Faire pour tout $j=1, n$

$A(i, j) \leftarrow \infty$

Fin pour

$A(i, i) = 0$

Fin pour

Faire pour tout $i=2, n$

choisir : aléatoirement la valeur de $A(1, i)$ dans l'intervalle $[1, k]$

$A(i, 1) = A(1, i)$

Fin pour

Faire pour tout $i=2, n-1$

choisir : aléatoirement la valeur de $A(i, i+1)$ dans l'intervalle $[1, k]$

$A(i+1, i) = A(i, i+1)$

Fin pour

choisir : aléatoirement la valeur de $A(2, n)$ dans l'intervalle $[1, k]$

$A(n, 2) \leftarrow A(2, n)$

Sortie : A

Fin algorithme

5 Problème du 1 -médian dans un cactus

5.1 Introduction

On s'intéresse au problème de la recherche du 1-médian dans un cactus. Nous présentons une méthode de résolution du problème qui est constituée de deux phases :

- La première phase consiste à localiser le cycle dans lequel se trouve le médian[10] (on utilise l'algorithme de Goldman pour cela).
- La dernière phase se réduit à la résolution d'un médian dans un cycle.

5.2 Définition : Un cactus[10]

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

Soit G le cactus d'ordre 17 et ayant quatre cycles C_1 , C_2 , C_3 et C_4 , dont la représentation graphique est donnée ci-dessous :

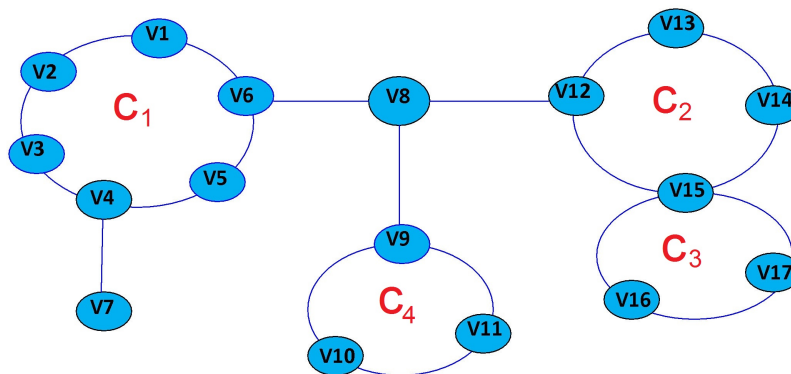


FIGURE 4.3 – Un cactus

5.3 Problème du 1 -médian dans un cactus[10]

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

5.3.1 Définition

Soit $G=(V,E)$ un graphe et soit $e=(a,b)$ une arête de E .

e est dite isthme de G si la suppression de e du graphe fait augmenter le nombre de composantes connexes. Un sommet v de V est dit point d'articulation si sa suppression augmente le nombre de composantes connexes.

Soit G_1 un sous graphe de G alors le poids, $W(G_1)$, de G_1 est égal à la somme des poids des sommets de G_1 .

Notons par G_a et G_b les deux sous graphes de G , obtenus par la suppression de l'isthme e , contenant respectivement les sommets a et b .

5.3.2 Lemme :[6]

Soit $G=(V,E)$ et soit $e=(a,b)$ un isthme de G .

Les deux propositions suivantes sont équivalentes :

1. $m \in G_a$
2. $W(G_a) \geq W(G_b)$

5.3.3 Lemme :[6]

Si $W(G_a) \geq W(G_b)$ alors :

Chercher un 1-médian m dans G équivaut à déterminer le 1-médian m dans G_a où le poids du sommet a sera augmenté du poids $W(G_b)$ du sous-arbre G_b .

5.3.4 Corollaire :[6]

Les deux propositions suivantes sont équivalentes :

1. $m \in G_a$
2. $W(G_a) \geq W(G)/2$

Dans ce qui suit nous allons décrire les deux phases de l'algorithme de recherche du 1-médian dans un cactus.

5.4 La phase 1

Le poids total du cactus, G , et les poids des cycles de ce dernier sont calculés. Le cactus G est représenté comme une arborescence (appelée arbre induit). Plus précisément, nous considérons chaque cycle de G comme un sommet de l'arbre induit. Si les deux cycles dans G ont un sommet commun, alors les deux sommets correspondants dans l'arbre induit sont adjacents.

Si deux cycles ont un sommet en commun, le sommet commun est représenté dans le 'cycle père'. Si un cycle n'a pas de sommets en commun avec d'autres cycles, ce sommet 'cycle' représente tous les sommets du cycle.

Le poids de chaque sommet de l'arborescence est égal à la somme des poids des sommets qui constituent le cycle correspondant dans G .

Par exemple, nous considérons un cactus et son arbre induit dans la figure 1 et supposons que l'arbre est de racine C_2 . Alors :

- C_1 représente v_1, v_2 et v_3 .
- C_2 représente v_4, v_5, v_6 et v_7 .
- C_3 représente v_8, v_9 et v_{10} .

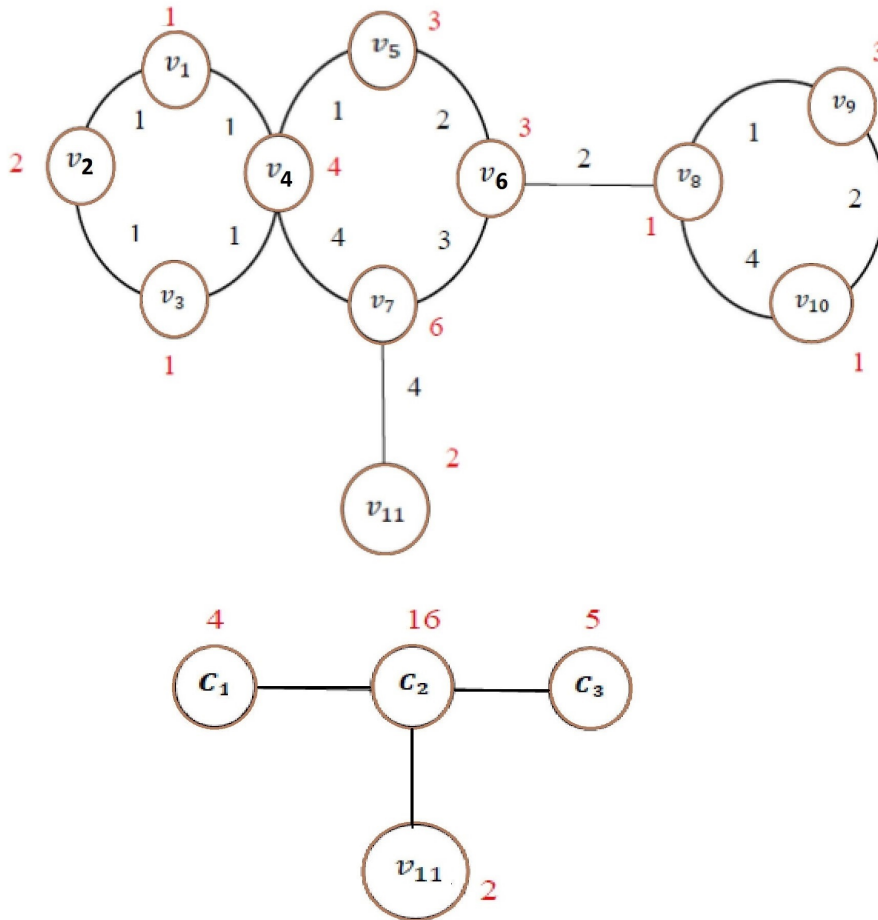


FIGURE 4.4 – Le graphe de cactus et l'arbre induit correspondant

Notons que C_2 est le père de C_1 donc le sommet v_4 est représenté par C_2 .

Par l'algorithme de Goldman, on détermine le médian, \mathbf{m} , de l'arbre induit. Si \mathbf{m} correspond à un sommet du cycle alors ce dernier est le médian de cactus. Si, au contraire, \mathbf{m} correspond à un cycle, \mathbf{C} , tout entier il faut déterminer le médian du cycle " \mathbf{C} " et donc passer à la deuxième phase.

5.4.1 lemme :[10]

Il y a au plus deux cycles dans le cactus \mathbf{G} qui contiennent le 1-médian.

5.5 Phase 2

On applique l'algorithme, décrit précédemment, de recherche du 1-médian dans le cycle \mathbf{C} (trouvé en phase 1)

5.6 Théorème 1

Le problème de la localisation du 1-médian dans un cactus peut être simplement résolu en $O(n)$ [10].

5.7 Exemple 1.

Considérons le cactus, G , donné par FIGURE 4.4. On veut déterminer le 1-médian de G .

Phase 1

L'arbre induit, T , correspondant à G peut être représenté par l'arborescence de racine C_2 . Le 1-médian de T est C_2 (en prenant le soins de considérer que le 1-médian de T peut appartenir au cycle C_1 , qui a v_4 comme sommet commun avec C_2 , si on transfère (*) le poids de v_4 au sommet C_1 de l'arbre T).

Phase 2

On calcule la distance de v_6 aux autres sommets du sous graphe $G(v_6)$ (sous graphe du cactus contenant le sommet v_6 et qui correspond au sous arbre de T de racine v_6) comme suit : - La longueur de C_3 est $l(C_3) = 7$.

- Comme v_8 est un fils de v_6 alors $d(v_6, v_8) = 2$.

Considérons les deux fils de v_8 , soient v_9 et v_{10} .

Comme $d^l(v_8, v_{10}) = 4 > l(c_3) - d^l(v_8, v_{10}) = 3 \Rightarrow d(v_8, v_{10}) = 3$.

On en déduit que : $d(v_6, v_{10}) = 5$ et $d(v_6, v_9) = 3$.

- La longueur de C_1 est $l(c_1) = 4$.

$d^l(v_4, v_1) = 1 < l(c_1) - d^l(v_4, v_1) = 3 \Rightarrow 1$.

$d^l(v_4, v_3) = 1 < l(c_1) - d^l(v_4, v_3) \Rightarrow d(v_4, v_3) = 1$.

$d^l(v_4, v_2) = 2 \leq l(c_1) - d^l(v_4, v_2) = 2 \Rightarrow d(v_4, v_2) = 2$.

- $d(v_7, v_{11}) = 1$.

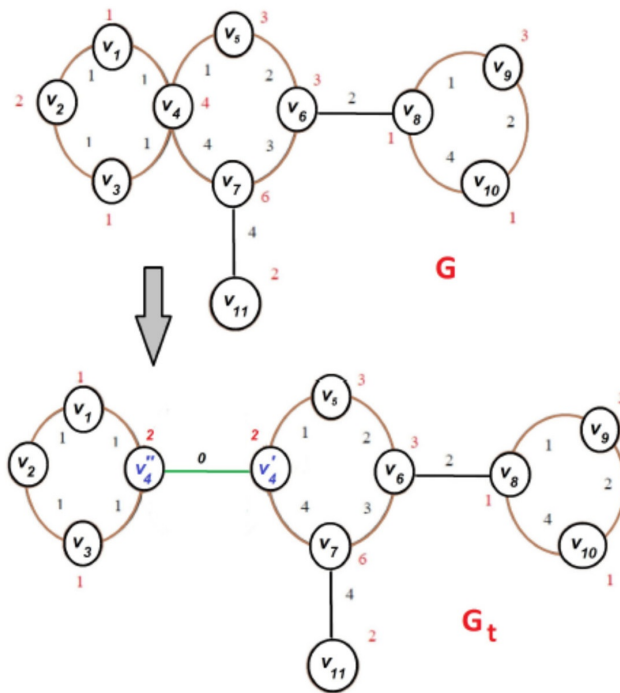
Par calculs élémentaires, nous obtenons les valeurs de l'objectif de chaque sommet du cycle C_2 . Finalement, nous concluons que v_7 est un 1- médian de G .

5.8 Remarque(*)

Le problème de la recherche du 1-médian dans un cactus ne change pas si tout sommet x intersection de deux cycles du cactus est éclaté en deux sommets x' et x'' que l'on relie par une arête $e=(x', x'')$ de cout nul avec $w(x') = w(x'') = w(x)/2$.

Cette manière de procéder permet d'éviter, dans la phase 1 de l'algorithme, d'éventuels transferts de poids.

Dans l'exemple traité ci-dessus, le sommet v_4 est commun à deux cycles. v_4 sera éclaté en deux sommets v'_4 et v''_4 que l'on relie par une arête $e=(v'_4, v''_4)$ de cout nul avec $w(v'_4) = w(v''_4) = w(v_4)/2=2$. Le cactus G peut être remplacé par le cactus G_t suivant :



Chapitre 5

Les résultats numériques

1 Introduction

Nous avons programmé les différents algorithmes (AG, MC, MSG et MR) en langage Matlab, sur un PC de type Acer avec un processeur : Intel(R)Core(TM)2 Duo CPU T6670 @ 2.20GHz 2.20 GHz, et de RAM 3.00 Go

Nous avons procédé à une étude comparative entre la méthode du sous-gradient (MG) et les autres méthodes. Pour ce faire, nous avons fait des tests sur des exemples que nous avons généré aléatoirement. Tous les tests numériques confirment les résultats théoriques (les méthodes (AG), (MC) et (MR) sont des algorithmes en $O(n)$).

2 Algorithme de Goldman

Nous avons généré aléatoirement quatre arbres que nous notons par T_1 , T_2 , T_3 et T_4 . Les poids des sommets sont générés dans l'intervalle $[1,100]$ et les couts des arêtes sont pris tous égaux à 1 (la recherche du 1-médian ne dépend pas des couts des arêtes). Le tableau (Table 1) résume les différents temps CPU et les nombres d'itérations de l'algorithme de Goldman (AG) sur ces quatre exemples.

Arbre	n	médian	AG (CPU)	Nombre des itérations
T_1	30	14	0.0055	28
T_2	50	26	0.077	48
T_3	78	55	0.0109	77
T_4	100	52	0.0087	50

Table 1 : Résultats de l'algorithme de Goldman (AG)

Le tableau (Table 2) résume les résultats donnés par la méthode de Goldman (AG) et l'algorithme du sous-gradient. Les différents temps CPU sont donnés en seconde. On remarque la performance de la méthode de Goldman(AG) par rapport à la méthode du sous-gradient (MSG).

Exemple	n	médian	MSG (CPU)	AG (CPU)
Exemple 1	50	48	0.0308	0.0056
Exemple 2	80	37	0.0569	0.0103
Exemple 3	100	90	0.0898	0.0131
Exemple 4	200	165	0.5626	0.0420
Exemple 5	400	389	4.6106	0.3304
Exemple 6	600	157	15.9406	1.4729
Exemple 7	800	711	38.2698	2.8042
Exemple 8	1000	146	74.6191	5.7306
Exemple 9	1200	782	128.6671	9.6500
Exemple 10	1400	427	202.1371	15.3154
Exemple 11	1800	613	440.5873	32.8481
Exemple 12	2000	1045	597.3881	45.8589
Exemple 13	2500	234	$1.0617 \cdot 10^3$	86.8773
Exemple 14	2800	2182	$1.6637 \cdot 10^3$	134.4917
Exemple 15	3000	2445	$2.2405 \cdot 10^3$	170.6935
Exemple 16	3400	77	$2.7956 \cdot 10^3$	263.6484
Exemple 17	3800	3096	$4.2652 \cdot 10^3$	378.0701
Exemple 18	4000	1709	$5.7498 \cdot 10^3$	462.9118

Table 2 : Étude comparative entre les méthodes (AG) et (MSG)

3 Algorithme de recherche du 1-médian dans un cycle

Nous avons programmé l'algorithme de recherche du 1-médian dans un cycle en langage Matlab. Nous avons testé cet algorithme avec la procédure du sous gradient sur 22 exemples d'ordre variant 20 à 5000 que nous avons généré aléatoirement par l'algorithme de génération aléatoire d'un cycle.

Exemple	n	médian	MC (CPU)	MSG (CPU)
Exemple 1	20	18	0.0017	0.0222
Exemple 2	50	10	0.0028	0.0319
Exemple 3	80	70	0.0056	0.0576
Exemple 4	100	89	0.0112	0.0919
Exemple 5	200	56	0.0479	0.5372
Exemple 6	300	51	0.1942	1.6825
Exemple 7	400	215	0.5732	3.9597
Exemple 8	500	248	1.2234	7.7070
Exemple 9	600	425	2.2446	13.3101
Exemple 10	700	336	3.7320	22.0224
Exemple 11	800	140	5.4631	32.7979
Exemple 12	900	610	7.7951	47.4213
Exemple 13	1000	734	10.6280	63.6133
Exemple 14	1500	992	36.0634	221.2637
Exemple 15	1700	1190	51.8840	313.3605
Exemple 16	1800	570	63.6981	372.3205
Exemple 17	2000	347	87.3722	522.7077
Exemple 18	3000	699	289.4387	$1.8064 \cdot 10^3$
Exemple 19	3500	2529	451.0867	$2.8635 \cdot 10^3$
Exemple 20	4000	825	669.1856	$5.2626 \cdot 10^3$
Exemple 21	4500	3149	958.6835	$6,5572 \cdot 10^3$
Exemple 22	5000	717	$1.444 \cdot 10^3$	$9.1476 \cdot 10^3$

Table 3 : Étude comparative entre les méthodes (MC) et (MSG)

4 Algorithme de recherche du 1-médian dans une roue

Ci-dessous (Tableau 4), on donne quelques résultats de la méthode de recherche du 1-médian dans une roue. Nous avons généré aléatoirement 25 exemples de roue d'ordre variant 20 à 4500.

Exemple	n	médian	MR (CPU)	MSG (CPU)
Exemple 1	20	19	0.0035	0.0223
Exemple 2	40	37	0.0044	0.0276
Exemple 3	80	1	0.0088	0.0586
Exemple 4	100	2	0.0129	0.0931
Exemple 5	200	1	0.0644	0.5085
Exemple 6	300	173	0.3188	1.7181
Exemple 7	400	182	1.0202	4.0144
Exemple 8	500	170	2.1939	7.7399
Exemple 9	600	426	3.8876	13.5307
Exemple 10	700	2	6.1563	21.9649
Exemple 11	800	656	9.3880	32.8844
Exemple 12	900	80	13.2122	46.8588
Exemple 13	1000	301	18.0915	64.4994
Exemple 14	1400	1356	49.9002	176.1659
Exemple 15	1600	1190	74.4308	316.8758
Exemple 16	1800	576	106.3787	384.2556
Exemple 17	2000	463	147.4943	509.0003
Exemple 18	2400	2271	247.2100	896.1089
Exemple 19	2600	1082	316.4402	$1.1553.10^3$
Exemple 20	2800	577	440.5954	$1.5158.10^3$
Exemple 21	3000	121	552.7387	$1.8169.10^3$
Exemple 22	3400	2110	727.4419	$2.6789.10^3$
Exemple 23	3800	1362	971.9471	$3.6048.10^3$
Exemple 24	4000	1282	$1.1463.10^3$	$5.2308.10^3$
Exemple 25	4500	1420	$1.6458.10^3$	$6.5845.10^3$

Table 4 : Étude comparative entre les méthodes (MR) et (MSG)

Comme le montrent si bien les tests numériques (Table 2, Table 3 et Table 4), les résultats théoriques sont confirmés.

Conclusion

Le problème du p -médian est un problème d'optimisation combinatoire très important au vu des nombreuses publications sur le sujet qui paraissent chaque année. Nous nous sommes intéressé au problème particulier du 1-médian dans un cycle, une roue et dans un cactus : nous avons donné des algorithmes en $O(n)$. Nous pensons, facilement, donner une extension de la méthode de la recherche du 1-médian dans un cactus à une classe de graphes qui contient strictement les cactus : On peut envisager de remplacer les cycles du cactus par des roues.

Bibliographie

- [1] **Ablaoui Hocine et Amir Abdessamed** " *Surrogate subgradient for P-median problem*" avril 2018 .
- [2] **A. J. Goldman** , *Optimal center location in simple network*. *Transp Sci* 5,212-221(1971).
- [3] **A. Oudjit et P.B Mirchandani** : , *Localizing 2-medians on probabilistic and deterministic tree networks*, *Networks*, vol. 10, pp. 329-350, 1980.
- [4] **BEAUMONT J.R. (1987)** -*Allocation Models and Central Place Theory, Spatial Analysis and Location-Allocation Models*, ed. Ghosh Avijit, Rushton Gerard, Van Nostrand Reinhold.
- [5] **Bekada Karima** " *Méthode de résolution d'un problème d'optimisation combinatoire*" mémoire .n d'étude, master, Université de Mostaganem, 2015.
- [6] **Bekkar Habib et Houidef AEK** " *Résolution du problème P-médian dans un arbre*" mémoire de fin d'étude, master, Université de Mostaganem, 2005.
- [7] **Christo.des, N and beasley, J. E** " *A tree search algorithm for the p-median problems*" *European Journal of Operational Research* vol 10 (1982) 196-204.
- [8] **J.Hatzl** " *Median problem on wheels and cactus graphs*" *Computing* (2007) 80 : 377-393.
- [9] **Kariv O et Hakimi,S** : : *An algorithmic approach to network location problems, part II : p-médians*. *SIAM J Appl Math* 27, 539-560(1979).
- [10] **Kien Trung Nguyen, Pham Van Chien, Ly Hong Hai, Huynh Duc Quoc** *A Simple Linear Time Algorithm for Computing a 1-Median on Cactus Graphs* ,*Applications and Applied Mathematics : An International Journal (AAM)*. Vol. 12, Issue 1 (June 2017), pp. 70 – 77 .
- [11] **Gondron et M** " *Graphe et Algorithmes*", E ditions Eyrolles, Paris, 1995.
- [12] **Thabet Naoufel** " *Algorithme de génération de colonnes pour le problème du P médian*" mémoire master, Université de Montréal, 1998.
- [13] **WEST D** *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.

Titre

Résumé : Nous avons étudié le problème du 1-médian dans quatre classes particulières de graphes (arbre, cycle, roue et cactus). Une amélioration de l'algorithme dans un cactus a été faite.

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

Title

Abstract : We studied the 1-median problem in four particular classes of graphs (tree, cycle, wheel, and cactus). An improvement of the algorithm in a cactus has been made.

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