

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université M'hamed Bougara Boumerdes
Faculté des Sciences
Département de Mathématiques



Mémoire Présenté
Pour L'Obtention Du Diplôme De Master
En Recherche Opérationnelle
Option : Recherche Opérationnelle et Mathématique de Gestion
(ROMAGE)

Par : BAROUD SARA
Et : BAI AI AMINA

**Problème d'Affectation du Trafic Urbain :
formulation et résolution**

Sous la direction de :

M ^{me} H.Harfouche	M.A. classe/ A	Président	à l'UMBB - Boumerdes.
M ^{me} A.Cheikh	M.A. classe/ A	Encadreur	à l'UMBB - Boumerdes.
M ^r A.Laouzai	Doctorant	Co-Encadreur	à l'UMBB - Boumerdes.
M ^{me} N.REGASS	M.A. classe/ A	Examineur	à l'UMBB - Boumerdes.

Année Universitaire 2015 – 2016

Remerciements

*Nous remercions tout d'abord ALLAH, le tout puissant de nous avoir illuminé
et ouvert les portes du savoir et nous avoir donné la volonté et le courage
d'élaborer ce travail.*

*Ce travail n'aurait pu être effectué sans l'accord, le soutien et l'aide de plusieurs
personnes.*

Nous tenons à remercier profondément :

*Nos Encadreurs Mme CHEIKH Asma et Mr, LAOUZAI Abdelfettah. Pour leurs aides,
leurs bienveillances, leurs générosité, et leurs précieux conseils durant la réalisation
de ce mémoire malgré leurs lourdes charges.*

*Nous exprimons nos sincères remerciements à : Mme H.Harfouche
pour l'honneur qu'elle nous fait de présider le jury de soutenance.*

*Nous témoignons notre reconnaissance à : Mme N.REGASS
pour avoir acceptée d'examiner ce travail.*

*Enfin, il nous tient à cœur d'exprimer notre reconnaissance à toute personne
ayant participé de près ou de loin à la réalisation de ce travail.*

Dédicaces

A Ma mère

A Mon père

A Mes frères Hamza et Walid

A Ma sœur Yousra

A Mon fiancé Sofiane

A Ma grande famille

A Mon binôme et amie Amina

A Tous mes enseignants

Je dédie mon travail

Sara

Dédicaces

Je dédie ce travail :

A ma très chère maman, pour ses encouragements, soutiens et ses prières pour que je réussisse dans ma vie, merci mère.

A mon très cher père, pour ses encouragements, son soutien, surtout pour son sacrifice afin que rien n'entrave le déroulement de mes études, merci père.

A mes chères sœurs et frères et leurs enfants

A toute Ma famille

A mon binôme et amie : Sara

A tous mes amis pour leur amitié, leurs encouragements, leurs soutiens

Tous ceux qui j'aime et je respecte

Amina

Table des matières

<i>Introduction générale</i>	2
1 Congestion et affectation du trafic dans les réseaux urbain	4
Introduction	4
1.1 Congestion	4
1.1.1 Types de congestions	5
1.1.2 Causes de la congestion	5
1.1.3 Conséquences de la congestion du trafic routier	6
1.2 Prévoir les lieux de la congestion en milieu urbain	7
1.2.1 Affectation du trafic et planification des voyages	8
1.2.2 Les modèles d'affectation du trafic	9
2 Théories des graphes et programmation mathématique	11
Introduction	11
2.1 Éléments de la théorie de graphe	11
2.1.1 Qu'est-ce qu'un graphe	11
2.1.2 Éléments d'un graphe orienté	12
2.1.3 Représentation d'un graphe	14
2.1.4 Réseau	15
2.1.5 Flot dans un réseau	15
2.1.6 Le problème du plus court chemin	16
2.1.7 Algorithmes de recherche de plus court chemin	17
2.1.8 Algorithme De DIJKSTRA-MOORE (1959)	17
2.2 Élément de la programmation mathématique	19
2.2.1 Éléments de base	19
2.2.2 Éléments de convexité	24
2.2.3 Éléments d'extrémum	26
2.2.4 Notions sur les algorithmes itératifs	28
2.2.5 Programmation mathématique	28
2.2.6 Problème d'optimisation non linéaire	30
2.2.7 Résolution des modèles non linéaires	31
2.2.8 L'existence et l'unicité	31
2.2.9 Caractérisation de la solution(Conditions d'optimalité)	31
2.2.10 Méthodes des résolutions	33
2.2.11 La méthode de Frank-Wolfe	34
Conclusion	36

3	Problème d'affectation du trafic urbain (PATU)	37
	Introduction	37
3.1	Réseau de trafic urbain (le terrain de problème)	37
3.1.1	Les paramètres du réseau routier	37
3.1.2	Réseau de trafic routière	39
3.2	La notion d'équilibre	40
3.2.1	Notion d'équilibre dans l'économie	40
3.2.2	Notion d'équilibre dans le domaine du trafic urbain	41
3.3	Problématique	41
3.4	Les conditions de Wardrop	42
3.4.1	L'équilibre usager de Wardrop	43
3.5	Formulation Mathématique	43
3.5.1	Model de Beckmann 1956	43
3.5.2	Caractérisation de la solution sur le modèle de Beckman	48
3.5.3	Exemple illustratif	54
	Conclusion	60
4	Méthode de résolution de PATU	61
	Introduction	61
4.1	Méthode des directions admissibles	62
4.1.1	La méthode de Franck-Wolfe	63
4.1.2	Exemple illustratif	68
	Conclusion	82
5	Implémentation	83
	Introduction	83
5.1	Choix du langage	83
5.2	La Fonction BPR (Bureau of Public Roads (1964))	85
5.3	Exemple d'application (Instance de Benchmark)	85
5.4	Application sous MATLAB	87
	Conclusion	92
	Conclusion générale	93
	Bibliographie	93
	Annexe	95

Introduction générale

Un réseau de transport typique offre plusieurs options à un usager pour effectuer un déplacement entre une origine et une destination. Avant d'effectuer un déplacement, le voyageur doit prendre une décision de choix afin de trouver l'option la plus adéquate pour arriver à sa destination. En revanche, les choix de déplacement déterminent l'état du trafic.

L'état de trafic est donc un résultat des comportements des voyageurs dans les axes du réseau. Un modèle d'affectation de voyage doit reproduire la répartition de la demande sur le réseau qui se compose d'une part de l'infrastructure routière, des lois d'écoulement du trafic et d'autre part par la demande de transport ainsi que le choix des usagers et les actions d'exploitation. Dans un tel modèle, on représente le système de répartition du trafic comme un marché de déplacement : l'infrastructure et l'ensemble des mesures d'exploitation sont représentées comme étant l'offre, les besoins de déplacement entre les points sont représentés comme étant la demande de déplacement, le choix des automobilistes est la rencontre offre-demande. Les modèles de choix consistent à trouver comment les usagers choisissent leurs options de déplacement en considérant les conditions du trafic. Autrement dit, ils cherchent la distribution optimale des usagers, de sorte que chaque voyageur soit dans l'option qui lui convient le plus.

Le but de cette étude est de présenter et analyser un modèle permettant d'élaborer une meilleure affectation des véhicules sur les axes du réseau routier.

Notre étude se place dans les hypothèses suivantes :

1. Toutes les caractéristiques du réseau ainsi que les données de trafic sont supposées constantes dans le temps (cas statique).
2. Nous utilisons un modèle d'affectation de trafic particulièrement simple afin de pouvoir interpréter simplement les résultats obtenus et mettre en œuvre l'algorithme efficace.

Pour mener à bien notre travail nous avons élaboré le plan suivant :

- Le premier chapitre est consacré à un aperçu général sur la raison principale (congestion) de la naissance de notre problème (PATU), puis à donner l'utilité de ce problème.
- Le deuxième chapitre introduit les notions fondamentales nécessaires pour l'étude du problème considéré.
- Dans le troisième chapitre nous cernons notre problématique, puis nous présentons la formulation mathématique du problème sous forme d'un problème d'optimisation

continue.

- Dans le quatrième chapitre nous d'écrivons une méthode numérique de résolution du problème donné.
- Le dernier chapitre consacré à l'implémentation de la méthode de Frank Wolfe sous le logiciel de programmation Matlab, nous avons par la suite exécuté nos résultats sur des instances de test prédéfinies appelées "Benchmark".
- Enfin nous terminons cette étude par une conclusion générale.

1

Congestion et affectation du trafic dans les réseaux urbains

Introduction

Dans ce chapitre introductif nous allons présenter dans un premier lieu, le phénomène de la congestion et de l'embouteillage dans un milieu urbain, les problèmes, les causes et les conséquences liés à ce phénomène, puis en second lieu nous allons voir, dans le cadre d'analyser, prévoir, ou minimiser ces effets, comment le problème d'affectation du trafic intervient.

1.1 Congestion

La congestion est un phénomène mondial, sans cesse grandissant, sur la majorité des routes et d'autoroutes des grandes villes et métropoles. Peu d'organisations responsables de la gestion de réseaux routiers ont mis en place les moyens adéquats pour mesurer et analyser le phénomène de façon méthodique.

Ce phénomène est un facteur important pour l'économie et pour l'ensemble des usagers des réseaux de transport. Ses impacts se traduisent par une augmentation du temps de déplacement, de la consommation de carburant, de la pollution, du stress, des risques d'accidents et d'incidents. De plus, la congestion réduit le bassin de main-d'œuvre et l'accessibilité à des activités économiques.

La combinaison de plusieurs facteurs tels le comportement des usagers "le choix d'itinéraire optimal" les données démographiques, l'aménagement du territoire et les coûts du logement, influe sur les niveaux de congestion et la rendent énorme et anarchique.

1.1.1 Types de congestions

Il existe deux types de congestions, congestions récurrentes et non récurrentes.

- Les congestions récurrentes apparaissent régulièrement, le plus souvent aux abords des grands centres urbains, aux heures de pointe, quotidiennement ou lors de grands départs en vacances. Elles deviennent prédominantes dès la moitié du siècle dernier avec l'augmentation de la demande de trafic.
- Les congestions non récurrentes, sont créées par des réductions temporaires et non régulières de capacité liées à des accidents, des travaux sur la chaussée, des véhicules en panne ou des objets divers tombés sur la chaussée.



FIGURE 1.1 – Congestions récurrentes et non récurrentes

1.1.2 Causes de la congestion

Pour comprendre la congestion du trafic routier, il faut garder présent à l'esprit que c'est un phénomène qui survient lorsque la demande (le nombre de véhicules qui cherchent à utiliser une infrastructure donnée) est supérieure à la capacité de cette infrastructure. Si la demande excède la capacité, alors des véhicules seront ralentis à l'entrée de l'infrastructure, formant ainsi un bouchon. Ces véhicules excédentaires seront à chaque instant plus nombreux qu'à l'instant précédent.

Comme chaque véhicule occupe une certaine longueur de voie, la longueur de la file d'attente ne fera que croître en proportion du nombre de véhicules présents dans cette file d'attente.

On voit de ce qui précède que la congestion est un phénomène évolutif, à la fois

dans le temps et dans l'espace. Or, en raison du caractère maillé de l'infrastructure, cet allongement de la file d'attente peut la conduire à atteindre des points de choix d'itinéraire, et ainsi congestionner des parties du réseau où circulent des véhicules qui n'utiliseront pas l'infrastructure à l'origine de la congestion.

- La première cause est tout simplement un volume de circulation supérieur à la capacité routière : les voitures sont trop nombreuses et le trafic s'arrête à chaque goulot d'étranglement (rétrécissement de la route, voies qui se rejoignent, croisements de routes). Cet accroissement en mobilité constitue une des causes principales d'apparition de plus en plus fréquente du phénomène de congestion.
- La seconde cause est liée aux comportements des conducteurs : Quand les voitures sont proches les unes des autres, il suffit qu'un conducteur freine brutalement pour faire freiner en cascade tous les véhicules qui le suivent et propager ainsi un ralentissement, avoir même un arrêt du trafic sur une certaine longueur. Quand le premier conducteur redémarre, de nouvelles voitures sont venues s'agglutiner et le bouchon se déplace vers l'arrière comme une onde, par un effet de dominos.
- La troisième cause est liée aux incidents inattendus : "les accidents, ...".

1.1.3 Conséquences de la congestion du trafic routier

Elles sont nombreuses et peuvent être classées en trois catégories : économiques, sociétales et environnementales.

1. Conséquences économiques : D'un point de vue économique, la congestion du trafic routier entraîne :

- Une perte de compétitivité d'une région, d'un pays ou d'un continent, du fait des congestions qui bloquent les artères de communication routière,
- Une perte de productivité, non seulement à cause des retards des personnes se rendant sur les lieux de leur travail, mais aussi en raison des livraisons tardives des approvisionnements ou des services rendus avec des retards importants annulés.

2. Conséquences sociétales : Les usagers des routes et d'autoroutes se plaignent amèrement des heures qu'ils perdent dans les embouteillages parce qu'elles empiètent sur le temps qu'ils peuvent consacrer à leur travail et à leur famille.

La congestion du trafic routier aura deux impacts importants sur :

1. La qualité de vie des usagers, car les conducteurs soumis à des embouteillages fréquents sont sujet à des problèmes de stress, d'anxiété et d'énerverment, entraînant un accroissement du risque d'accident,
2. Le pouvoir d'achat des usagers, car la congestion du trafic a des conséquences directes sur l'augmentation de la consommation énergétique. Dans le contexte actuel d'envolée des prix du pétrole, ceci engendre une nette diminution du pouvoir d'achat des usagers.

3. Conséquences environnementales : Enfin, d'un point de vue environnemental, la congestion du trafic routier a un impact néfaste et négatif à cause de l'augmentation de la pollution qui engendre des coûts économiques et écologiques en croissance exponentielle.

L'érosion de la productivité causée par les pertes de temps dues aux bouchons, l'aggravation de la pollution de l'air et de l'eau, l'augmentation du bruit et la dégradation de la qualité de vie sont autant de conséquences de l'augmentation de la demande de transport terrestre.

On peut donc conclure que le phénomène de congestion du trafic routier est un problème socio-économico-environnemental crucial qui exige de rechercher des solutions efficaces et rapides.

1.2 Prévoir les lieux de la congestion en milieu urbain

Depuis les années 50, les préoccupations sociales, économiques et environnementales ont considérablement modifié l'approche de l'exploitation de la route en ville.

Désormais, la construction de nouvelles voies n'étant plus que rarement considérée, l'optimisation du fonctionnement des infrastructures existantes est une priorité pour tout gestionnaire de réseau routier pour faire face à la croissance permanente du trafic et des encombrements.

Les infrastructures les plus importantes en capacité de transport et en niveau de service, sont conçues et dimensionnées afin d'écouler un certain volume de trafic.

Quand ce volume croît, ou aux périodes de forte concentration comme les pointes journalières en milieu urbain au moment des migrations domicile-travail et les pointes saisonnières en milieu interurbain au moment des migrations touristiques, des phénomènes de congestion se mettent en œuvre, ce qui dégradent la qualité de service et même, en cas de saturation, réduisent la capacité d'écoulement.

La congestion, et sa forme aigüe : la saturation, font l'objet d'une gestion dynamique du trafic par les exploitants des réseaux : intensification des fréquences de desserte en transports public, intervention rapide sur incident en transport public comme en transport privé, mesures d'exploitation comme l'orientation du flot sur certains itinéraires, ou la réduction de la vitesse limite afin d'homogénéiser le flot ou de le retarder, et même des mesures à caractère économique comme la modulation tarifaire.

Dans ce cadre, les exploitants routiers cherchent aujourd'hui à mieux mesurer, comprendre et prévoir l'écoulement du trafic, notamment dans ses aspects dynamiques. Ce regard attentif nouveau leur permet, par la mise en place de mesures de régulation du trafic, d'assurer une réponse efficace aux difficultés d'exploitation observées.

Ces enjeux sont considérables sur le plan économique, mais aussi très complexe : le trafic est formé d'un grand nombre de déplacements, aux lieux d'origine, de destination et de passage variés, avec divers comportements de circulation qui distinguent notamment les différents types de véhicules (poids lourds, voitures, etc.). En effet, le comportement

d'un poids lourd sur la voie se distingue nettement de celui d'une voiture (accélération, gabarit, vitesse de pointe,...).

1.2.1 Affectation du trafic et planification des voyages

La planification fait l'objet d'études au caractère analytique poussé, appelés des études de prévisions de trafic, et réalisés au moyen d'un instrument technique dédié : le modèle d'affectation du trafic, ou sont représentées :

- L'offre de transport en termes d'infrastructures et de services localisées dans un territoire d'étude, avec pour chaque service des caractères physiques et économiques en temps, en prix et en qualité.
- La demande de déplacement, conçue comme des flux d'utilisateurs allant des zones d'origine vers les zones de destination auxquels on associe des comportements économiques.
- L'équilibre entre l'offre et la demande, par simulation des choix usagers et du fonctionnement technique du réseau, qui détermine les volumes et les coûts généralisés des services sur le marché des transports.

Comme souligné par Sheffi [21], le flux de véhicules présents à un moment donné sur un axe dans une zone urbaine dépend de beaucoup de décisions individuelles.

Les usagers du réseau doivent décider s'ils se déplacent ou non, à quelle heure ils partent, quel mode de transport ils utilisent (voiture particulière ou transport public) et enfin leur itinéraire pour gagner leur destination.

D'une part, ces décisions dépendent, en partie tout au moins, du niveau de congestion du réseau et de la localisation des points de congestion. D'autre part, la congestion en tout point du réseau est fonction du nombre de véhicules présents en ce point et donc des décisions individuelles des usagers du réseau.

Le problème d'affectation du trafic est donc sert à étudier et modéliser les interactions entre congestion et décisions de déplacement pour obtenir les flux de véhicules résultants sur chacun des axes du réseau de transport urbain.

L'utilité de ce genre de modèle de simulation du trafic est de pouvoir prédire l'impact de différents scénarios de transport. En effet, une fois calculés les flux de véhicules sur chacun des axes du réseau de transport urbain, on peut en déduire une série de mesures résultantes telles que, par exemple : la congestion mesurée par le temps d'accès moyen à chaque zone, la pollution de l'air par émission de gaz de combustion ou le niveau du bruit.

Dans notre étude, nous allons nous concentrer sur la prévision de la distribution des flux sur les axes du réseau à partir de certains données pour pouvoir analyser la congestion.

Afin de déterminer ces flux pour chacun des axes du réseau, on partira de la remarque suivante : la congestion s'accroît avec le flux de véhicules sur l'arc, mais, d'autre part, la congestion fait dévier les véhicules des axes encombrés. Les flux résulteront donc d'un équilibre entre ces deux mouvements en sens opposé.

1.2.2 Les modèles d'affectation du trafic

Les modèles d'affectations et les études de prévisions du trafic sont désormais couramment répandus dans les pays développés, grâce à une ingénierie-conseil spécialisée.

Il existe deux grands types d'affectations de trafic, qui présentent chacun son avantage et son inconvénient.

1. Modèles statiques :

Ces modèles réduisent les temps chronologiques à une période fictive de référence durant laquelle les variables de trafic sont considérées constantes : cela n'empêche pas de modéliser des temps de parcours et certains aspects de la congestion, mais cela réduit la portée des simulations à la planification du réseau et à la conception de schémas de circulation.

En 1952, Wardrop définit deux principes d'équilibres pour l'approche statique :

- Le premier principe correspond à un optimum individuel où chaque individu utilise l'itinéraire qui lui procure le coût généralisé (réduit le temps de parcours) le plus faible pour un couple origine-destination (*od*) donné.
- Le second principe est fondé sur l'optimum social où le partage des usagers entre les itinéraires concurrents se fait de telle sorte que le coût social généralisé (pour l'ensemble du système) soit minimum.

Dans le cas général ces deux équilibres ne sont pas équivalents.

En 1956, Beckmann et al [3]. ont proposé une formulation mathématique pour l'équilibre statique entre offre et demande de déplacement sur un réseau de transport, cela pour chacun des deux principes de Wardrop, dans le cas d'une classe d'usagers et d'une demande élastique. De plus, Beckmann a indiqué un algorithme pour calculer l'équilibre. Cette approche a été largement utilisée et développée, jusqu'au début des années 1990, avec des extensions à plusieurs classes de trafic, aux réseaux de transports collectif, au choix modal, à d'autres choix de transports (localisation, distribution spatiale entre relations (*od*)).

2. Modèles dynamiques

Ces modèles ont pris leur essor dans les années 1990, profitant des progrès importants des moyens informatiques. Dans les modèles dynamiques, la congestion, et par conséquent le choix d'itinéraire, varie au cours du temps. Dans certains modèles, le choix de l'horaire de départ est également considéré.

De manière succincte, les modèles dynamiques peuvent être décrits comme suit : les usagers font face à un double choix, celui de l'heure de départ et celui de l'itinéraire. Le coût généralisé associé à un trajet est la somme pondérée du temps de déplacement et d'une fonction des pénalités dues aux arrivées précoces

ou tardives. Chaque individu choisit l'heure de départ et l'itinéraire qui minimisent cette fonction de coût généralisé. Un équilibre dynamique est atteint lorsqu'aucun usager ne peut modifier son itinéraire ou son heure de départ en vue de réduire strictement son coût généralisé.

2

Théories des graphes et programmation mathématique

Introduction

Ce chapitre est reparti en deux parties, la première partie consacrée à la théorie des graphes, la seconde réservée à la programmation mathématique. Il sera consacré aux notions, aux rappels et aux définitions des théorèmes et des algorithmes utilisés le long de ce travail.

2.1 Éléments de la théorie de graphe

2.1.1 Qu'est-ce qu'un graphe

Un graphe est défini par un couple $G = (X, E)$ tel que :

- X est un ensemble fini de sommets $X = \{x_1, x_2, \dots, x_n\}$.
- E est un ensemble des arêtes $E = \{e_1, e_2, \dots, e_m\}$.

Où :

$(|X| = n)$ est le nombre de sommets de ce graphe appelé l'ordre de graphe .
 $(|E| = m)$ est le nombre des arcs de ce graphe appelé la taille de graphe .

Un graphe peut être orienté ou non.

2.1.1.1 Graphe orienté [17]

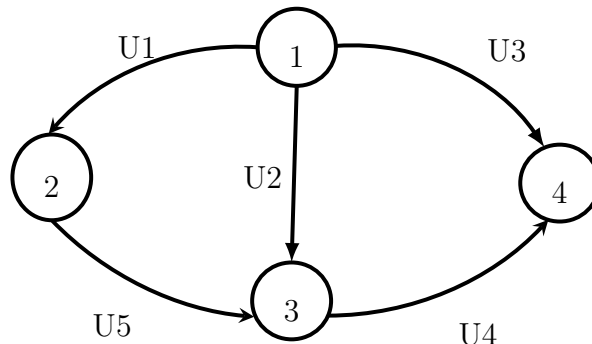
Un graphe orienté est un graphe fini $G = (X, U)$ défini par :

- L'ensemble fini des sommets : $X = \{x_1, x_2, \dots, x_n\}$.
- L'ensemble fini des arcs : $U = \{u_1, u_2, \dots, u_m\}$.

Dans un graphe orienté, les couples $(x_i, x_j) \in U$ sont orientés, c'est à dire que (x_i, x_j) est un couple ordonné, où x_i est le sommet d'extrémité initiale, et x_j le sommet d'extrémité terminal.

Un couple (x_i, x_j) est appelé un arc, et est représenté graphiquement par $x_i \rightarrow x_j$.

exemple :



Le graphe orientée $G = (X, U)$

On a :

$X = \{1, 2, 3, 4\}$ et $U = \{u_1 = (1, 2), u_2 = (1, 3), u_3 = (1, 4), u_4 = (3, 4), u_5 = (2, 3)\}$.

2.1.1.2 Graphe non orienté[17]

Un graphe non orienté est un graphe fini $G = (X, E)$ défini par :

- L'ensemble fini des sommets : $X = \{x_1, x_2, \dots, x_n\}$.
- L'ensemble fini des arêtes : $E = \{e_1, e_2, \dots, e_m\}$.

Dans un graphe non orienté, les couples $\{x_i, x_j\} \in E$ ne sont pas orientés, c'est à dire que (x_i, x_j) est équivalent à (x_j, x_i) .

Une paire $\{x_i, x_j\}$ est appelée une arête, et est représentée graphiquement par $x_i - x_j$.

2.1.2 Eléments d'un graphe orienté

2.1.2.1 Successeur [24]

Soit $G = (X, U)$ un graphe orienté où x_1 et x_2 deux sommets $\in X$, on dit que x_2 est un successeur de x_1 s'il existe un arc ayant x_1 comme extrémité initiale et x_2 comme extrémité terminale c'est-à-dire \exists un arc $u = (x_1, x_2) \in U$.

$\Gamma^+(x_1)$: est l'ensemble des successeurs d'un sommet x_1 .

2.1.2.2 Prédécesseur [24]

Soit $G = (X, U)$ un graphe orienté où x_1 et x_2 deux sommets $\in X$, on dit que x_1 est un prédécesseur de x_2 s'il existe un arc ayant x_1 comme extrémité initiale et x_2 comme extrémité terminale c'est-à-dire \exists un arc $u = (x_1, x_2) \in U$.

$\Gamma^-(x_2)$: est l'ensemble des prédécesseurs d'un sommet x_2 .

2.1.2.3 Racine [24]

On appelle racine dans un graphe orienté un sommet r tel qu'il existe un chemin de r à tout autre sommet du graphe.

2.1.2.4 Source [13]

On appelle source un sommet de départ s où l'ensemble de prédécesseurs de ce sommet est vide c'est-à-dire $\Gamma^-(s) = \emptyset$.

2.1.2.5 Puits [13]

On appelle puits un sommet d'arrivée d où l'ensemble de successeurs de ce sommet est vide c'est-à-dire $\Gamma^+(d) = \emptyset$.

2.1.2.6 Chemin [17]

On appelle chemin dans un graphe orienté $G = (X, U)$, une suite alternée de sommets et d'arcs : $p = (x_0, u_1, x_1, \dots, x_{k-1}, u_k, x_k)$ tel que :
Pour tout $1 \leq i \leq k$, le sommet x_{i-1} est une extrémité initiale de l'arc u_i et le sommet x_i est son extrémité terminale.

2.1.2.7 Chemin simple [4]

Un chemin p est simple si chaque arc $u \in U$ du chemin est empruntée une seule fois.

2.1.2.8 Circuit [17]

Un circuit est un chemin dont les extrémités coïncident.

2.1.2.9 Circuit absorbant [13]

Un circuit absorbant est un circuit dont la somme des longueurs de ses arcs < 0 .

2.1.2.10 Sous graphe [24]

Soit un graphe $G = (X, U)$ quelconque.
Soit $A \subset X$, alors le sous graphe engendré par A est le graphe G_A dont les sommets sont les éléments de A et dont les arcs sont les arcs de G ayant leurs deux extrémités dans A .

2.1.2.11 Arbre [4]

Un arbre est un graphe orienté $G = (X, U)$ tel que :

- G est sans circuit.
- Il existe une racine r de l'arbre G .
- Tous les autres sommets ont exactement 1 prédécesseur.

2.1.2.12 Arborescence [4]

Une arborescence est un arbre comportant un sommet particulier r , nommé racine de l'arborescence à partir duquel il existe un chemin unique vers tous les autres sommets.

2.1.3 Représentation d'un graphe

Il y a plusieurs manières pour représenter un graphe, nous allons citer les plus efficaces.

2.1.2.1 Matrice d'adjacence [17]

Le graphe $G = (X, U)$ est simplement représenté par un tableau à 2 dimensions de taille $n * n$ (Voir la définition d'une matrice dans partie 2 page 21) où n l'ordre du ce graphe

La matrice d'adjacence fait correspondre les sommets d'extrémité initiale des arcs (placés en ligne dans la matrice) aux sommets d'extrémité terminal (placés en colonne), dans le formalisme matrice booléenne, l'existence d'un arc (x_i, x_j) se traduit par la présence d'un 1 à l'intersection de la ligne i et de la colonne j , l'absence d'arc par la présence d'un 0 (avec $1 \leq i, j \leq n$).

Soit $G = (X, U)$ un graphe orientée, la matrice d'adjacence est représentée comme suit :

$$M_{i,j} = \begin{cases} 1, & \text{si } (x_i, x_j) \in U ; \\ 0, & \text{Sinon.} \end{cases}$$

$$\text{Telque } 1 \leq i, j \leq n$$

2.1.2.2 Liste d'adjacence [27]

Cette représentation consiste en un tableau G de n listes chaînées, Pour chaque sommet x_i du graphe, on aura sa liste d'adjacence $G[i]$ qui sera composée des successeurs de x_i dans le cas orienté .

Soit $G = (X, U)$ un graphe orienté, on associe à chaque sommet x_i , la liste des sommets x_j tels que $(x_i, x_j) \in U$.

Exemple :

l'application sur le graphe précédent

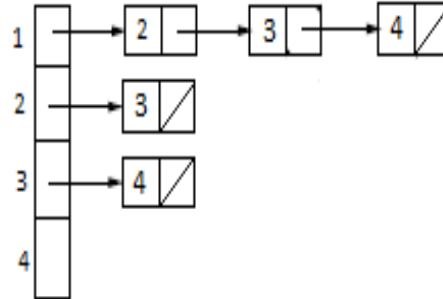


FIGURE 2.1 – liste d’adjacence

2.1.4 Réseau

Un réseau est un graphe orienté $G = (X, U)$ auquel on associe une fonction $L : U \rightarrow IR$ à chaque arc fait correspondre à sa "longueur, temps de parcourt ou le poids", on note $G = (X, U, L)$ un tel réseau.[13]

2.1.5 Flot dans un réseau

Définition [13] :

Soit $G = (X, U, L)$ un réseau d’ordre n et de taille m , où s est la source et d est le puits, à chaque sommet $x \in X$ on associe un ensemble d’arc :

- $\Gamma^+(x)$:l’ensemble de successeurs de x .
- $\Gamma^-(x)$:l’ensemble de prédécesseurs de x .

À chaque arc $u \in U$, on associe un nombre réel, noté φ_u appelé flux de l’arc u .

Soit $G_0 = (X, U^0, L)$ un graphe réduit de G avec un arc du sommet puits vers le sommet source.

$\varphi' = [\varphi_0, \varphi_1, \varphi_2, \dots, \varphi_m]$ est **un flot** sur G^0 **si et seulement si** :

1. La somme des flux sortants est égale à la somme des flux entrants(La condition de conservation des flux au niveau de chaque sommet) :

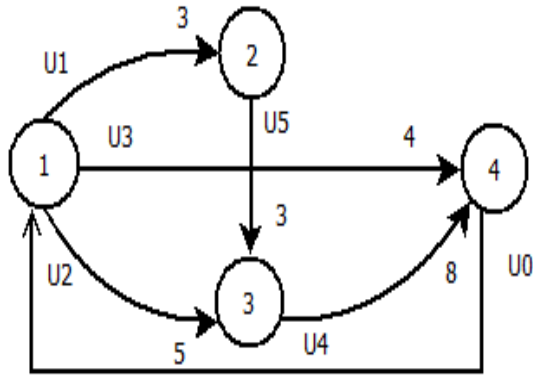
$$\sum_{x_j \in \Gamma^+(x_i)} \varphi(x_i, x_j) = \sum_{x_i \in \Gamma^-(x_j)} \varphi(x_i, x_j) \quad \forall i, j \in n, i \neq j.$$

2. la somme des flux sortants sur s est égale à la somme des flux entrants sur d et égale à la valeur de flot $v(\varphi')$:

$$\sum_{x_j \in \Gamma^+(s)} \varphi(s, x_j) = \sum_{x_i \in \Gamma^-(d)} \varphi(x_i, d) = v(\varphi') \quad \forall i, j \in n$$

Exemple :

Soit $G^0 = (X, U^0, L)$ un réseau où :
 $X = \{1, 2, 3, 4\}$ $s = 1$ et $d = 4$.
 $U^0 = \{u_0, u_1, u_2, u_3, u_4, u_5\}$.



Exemple de réseau avec arc retour

$$\sum_{x_j \in \Gamma^+(1)} \varphi(1, x_j) = \varphi(1, 2) + \varphi(1, 4) + \varphi(1, 3) = 3 + 4 + 5 = 12 \dots \dots \dots (1)$$

$$\sum_{x_j \in \Gamma^-(4)} \varphi(x_i, 4) = \varphi(1, 4) + \varphi(3, 4) = 4 + 8 = 12 \dots \dots \dots (2)$$

On a : (1)=(2) alors $\varphi' = [\varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5]$ est un flot sur G^0 de valeur 12.

2.1.6 Le problème du plus court chemin

Le problème du plus court chemin compte parmi les problèmes les plus classiques de la théorie des graphes et les plus importants dans leurs applications.

Ce problème peut être posé de la façon suivante :

2.1.4.1 Définition [17]

Etant donné un graphe $G = (X, U, L)$ orienté valué, on associe a chaque arc $u = (x_i, x_j)$ un nombre réel $l(u)$, appelé la longueur ou le poids de l'arc u .

Le problème du plus court chemin entre deux sommets s (source ou origine) et d (puit ou destination) du graphe consiste à déterminer ,parmi tous les chemins allant de s à d ,un chemin ,noté p dont la longueur totale :

$$l(p) = \sum_{u \in U} l(u) \quad \text{soit minimal.}$$

Interprétation de $l(p)$: coût de transport,dépense de construction,temps nécessaire de parcours,....

2.1.4.2 Condition d'existence [17]

Le problème du plus court chemin à une solution si et seulement s'il n'existe pas dans le graphe de circuit de longueur strictement négative (circuit absorbant) pouvant être atteint à partir de l'origine s .

2.1.4.3 Les application du problème de plus court chemin

Le problème de la recherche du plus court chemin dans un graphe se rencontre dans de nombreuses applications. On peut citer entre autres :

- Les problèmes d'optimisation de réseaux (routiers, télécommunications).
- Les problèmes de tournées.
- Les problèmes de programmation dynamique à états discrets et temps discret.
- Certaines méthodes de traitement numérique du signal, de codage et de décodage de l'information.
- Les problèmes de labyrinthe et de recreations mathématiques.

2.1.7 Algorithmes de recherche de plus court chemin

Le problème de plus court chemin peut être résolu de nombreuses manières. tout dépend des hypothèses émises sur la structure du réseau, Lorsque le réseau est sans circuit on appliquera l'algorithme de Bellman Ford, Lorsque le réseau n'a que des longueurs positives ou nulles (mais éventuellement avec des circuits), on utilisera la méthode de Dijkstra. Enfin, dans un réseau sans hypothèse particulière, on dispose d'un algorithme général.

Parmi ces algorithmes ,nous nous intéressons a l'algorithme de Dijkstra.

2.1.8 Algorithme De DIJKSTRA-MOORE (1959)

En 1959, Dijkstra et Moore ont développé un algorithme pour calculer les plus courts chemins d'un sommet a tous les autres sous l'hypothèse ou tous les coûts sont positifs.

2.1.8.1 Principe de l'algorithme [17]

L'algorithme prend en entrée un graphe orienté $G = (X, U, L)$ pondéré par des réels positifs L et un sommet racine r , il s'agit de construire progressivement un sous-graphe dans lequel sont classés les différents sommets par ordre croissant de leur distance minimale au sommet de départ, La distance correspond à la somme des poids des arcs empruntés.

Au départ, on considère que les distances de chaque sommet au sommet de départ sont infinies sauf pour le sommet de départ pour lequel la distance est de 0, Le sous-graphe de départ est l'ensemble vide.

Au cours de chaque itération, on choisit en dehors du sous-graphe un sommet de distance minimale et on l'ajoute au sous-graphe. Ensuite, on met à jour les distances des sommets voisins de celui ajouté. La mise à jour s'opère comme suit : la nouvelle distance du sommet voisin est le minimum entre la distance existante et celle obtenue en ajoutant le poids de l'arc entre sommet voisin et sommet ajouté à la distance du sommet ajouté.

On continue ainsi jusqu'à épuisement des sommets (ou jusqu'à sélection du sommet d'arrivée).

2.1.8.2 Applications de l'algorithme

L'algorithme de Dijkstra trouve son utilité dans le calcul des itinéraires routiers où le poids des arcs pouvant être la distance (pour le trajet le plus court), le temps estimé (pour le trajet le plus rapide), la consommation de carburant et le prix des péages (pour le trajet le plus économique).

2.1.8.3 L'algorithme : [17]

Algorithme 1 L'algorithme de Dijkstra-moore

Entrées: Un graphe $G = (X, U, L)$, un sommet s (racine);

Sorties: Un tableau d de marquage;

```

1: pour tous les sommet  $v$  faire
2:    $d[v] \leftarrow \infty$ ;
3: fin pour
4:  $d[s] \leftarrow 0$ ;
5:  $A \leftarrow X$ ;
6:  $T \leftarrow \emptyset$ ;
7: tantque  $A \neq \emptyset$  faire
8:   choisir  $v$  dans  $A$  tel que  $d[v]$  soit minimum
9:    $T \leftarrow T \cup \{v\}$ ;
10:   $A \leftarrow A - \{v\}$ ;
11:  pour tous les  $u \in \Gamma^+(v)$  faire
12:    si  $d[u] > d[v] + l[v, u]$ ; alors
13:       $d[u] \leftarrow d[v] + l[v, u]$ ;
14:    finsi
15:  fin pour
16: fin tantque

```

2.1.8.3 Complexité [27]

L'algorithme de Dijkstra comporte ($|X| = n$) étapes correspondant au marquage de chacun des sommets, pour une étape il nous faut calculer les marquages de tous les sommets non encore marqués, en scannant chacun de leurs voisins, ce qui montre que cette opération se fait en temps $O(|E| = m)$, d'où la complexité totale de l'algorithme est polynomiale d'ordre $O(n.m)$.

2.2 Élément de la programmation mathématique

Cette partie du chapitre sera consacré à la programmation mathématique, nous allons donner en premier lieux les définitions de bases nécessaires puis, en deuxième lieux les éléments théoriques afin de caractériser un programme mathématique et plus précisément le cas non linéaire.

2.2.1 Éléments de base

2.2.1.1 Éléments de topologie

1. Distance, espace métrique [25] :

Soit S un ensemble de \mathbb{R}^n .

1.1 Distance :

On appelle distance sur S qu'on la note l'application :
 $d : S * S \rightarrow \mathbb{R}_+$ vérifiant les axiomes : $\forall x, y, z \in S$

1. $(d(x, y) = 0) \Leftrightarrow (x = y)$.
2. $d(x, y) = d(y, x)$ (symétrique)
3. $d(x, z) \leq d(x, y) + d(y, z)$ (inégalité triangulaire)

1.2 Espace métrique :

Un espace métrique est un couple (S, d) ou d est une distance sur l'ensemble S .

2. Norme, espace vectoriel normés :

Soit E un espace vectoriel

2.1 Norme :

On appelle norme sur E , une application habituellement notée $\|\cdot\|$:

$$\begin{aligned} \mathbb{R}^n &\mapsto \mathbb{R}_+ \\ x &\mapsto \|x\| \end{aligned}$$

Vérifiant pour tout $x, y \in \mathbb{R}^n$ et tout $\lambda \in \mathbb{R}$:

1. $(\|x\| = 0) \Leftrightarrow (x = 0)$
2. $\|\lambda x\| = |\lambda| * \|x\|$ (homogénéité)
3. $\|x + y\| \leq \|x\| + \|y\|$ (inégalité triangulaire)

2.2 Espace vectoriel normé :

Un espace vectoriel normé est un couple $(E, \|\cdot\|)$ ou E est un espace vectoriel sur \mathbb{R}^n et $\|\cdot\|$ est une norme sur E .

3. Ouverts, fermés [20] :

Soit (E, d) un espace métrique

3.1 Boule :

Soit $x \in E$ et soit $r \in \mathbb{R}_+^*$. On appelle boule ouverte (resp. fermée) de centre x et de rayon r l'ensemble noté :

$$B(x, r) = \{y \in E, d(x, y) < r\}$$

$$(resp. \bar{B}(x, r) = \{y \in E, d(x, y) \leq r\})$$

3.2 Ouvert :

On dit qu'un ensemble O de E , est un ouvert si pour chaque $x \in O$ il existe une boule B de centre x contenue dans O :

$$\forall x \in O, \exists r > 0, B(x, r) \subset O$$

3.3 Fermé :

On dit qu'un ensemble F de E , est fermé si son complémentaire dans E est un ouvert de \mathbb{R}^n .

2.2.1.2 Eléments d'algèbre linéaire[12]

1. Matrice, vecteurs :

Soit $n, m \in \mathbb{N}$, une matrice d'ordre $n * m$ a coefficients dans \mathbb{R} est un tableau a deux dimensions, ayant n lignes et m colonnes, représentée sous la forme suivante :

$$A(I, J) = (a_{ij})_{i \in I, j \in J} = \begin{pmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1m} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nm} \end{pmatrix}$$

Où $I = \{1, 2, \dots, n\}$ et $J = \{1, 2, \dots, m\}$ représentent respectivement l'ensemble des indices des lignes et colonnes de A .

Pour des calculs pratiques, la matrice A se note aussi :

$$A = (A_1, A_2, \dots, A_j, \dots, A_m) = \begin{pmatrix} A_1^t \\ A_2^t \\ \cdot \\ \cdot \\ A_n^t \end{pmatrix}$$

$$\text{Où } A_j = A(I, j) = \begin{pmatrix} A_{1j} \\ A_{2j} \\ \vdots \\ \vdots \\ A_{nj} \end{pmatrix}$$

est le $j^{\text{ème}}$ vecteur-colonne de dimension n .

$A_i^t = a(i, J) = (a_{i1}, a_{i2}, \dots, a_{im})$ est le $i^{\text{ème}}$ vecteur-ligne de dimension m .

Remarques :

1. Le terme $(.^t)$ désigne la transposé.
2. La matrice A est dite carrée dans le cas ou $n = m$; de plus, si $A = A^t$, la matrice est dite symétrique.
3. Notons qu'un vecteur-colonne de dimension n peut être considéré comme une matrice d'ordre $(n * 1)$, tandis qu'un vecteur-ligne de dimension n peut être considéré comme une matrice d'ordre $(1 * n)$.

2. Indépendances et dépendance linéaire :

Les vecteurs a_1, a_2, \dots, a_k sont dits linéairement indépendants si :

$$\forall \lambda_i \in \mathbb{R} \quad \lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_k a_k = 0 \Rightarrow \lambda_1 = \lambda_2 = \dots = \lambda_k = 0$$

Les vecteurs $a_i, i = 1, \dots, k$ sont dits linéairement dépendants s'il existe une combinaison linéaire nulle de ces vecteurs, avec des scalaires $\lambda_i, i = 1, \dots, k$, non tous nuls.

3. Mineurs principaux d'une matrice :

Soit A une matrice carrée symétrique d'ordre n . Un mineur principal d'ordre k est le déterminant de la sous matrice de A d'ordre k obtenue en supprimant les $(n - k)$ lignes $(i_1, i_2, \dots, i_{n-k})$ et les $(n - k)$ colonnes $(j_1, j_2, \dots, j_{n-k})$ correspondantes dans A .

4. Mineurs principaux diagonaux d'une matrice :

Le mineur principal diagonal d'ordre k noté D_k de la matrice A est le déterminant de la sous matrice de A obtenue en supprimant les $(n - k)$ dernières lignes et les $(n - k)$ dernières colonnes de A .

5. Matrice (semi) définie positive (semi) définie négative :

Définition 01 :

Soit A une matrice carrée symétrique d'ordre n

- On dit que A est une matrice définie positive si :
 $X^T A X > 0 : \forall X \in \mathbb{R}^n ; X \neq 0$.
- On dit que A est une matrice semi- définie positive si :
 $X^T A X \geq 0 : \forall X \in \mathbb{R}^n$.
- On dit que A est une matrice définie négative si :
 $X^T A X < 0 : \forall X \in \mathbb{R}^n ; X \neq 0$.
- On dit que A est une matrice semi-définie négative si :
 $X^T A X \leq 0 : \forall X \in \mathbb{R}^n$.

Avec : $X^T A X = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j$.

Définition 02 :

Soit $\lambda \in \mathbb{R}^n$, λ est appelée valeur propre de A si $\det(A - \lambda I) = 0$ ou \det désigne le déterminant.

Propriétés :

Soit A une matrice carrée d'ordre n .

- A est définie positive si et seulement si $-A$ est définie négative.
- A est définie positive si et seulement si les mineurs principaux diagonaux sont positives ; ($D_k > 0 ; k = 1, \dots, n$)
- A est semi-définie positive si et seulement si tous les mineurs principaux sont positives ou nuls.
- A est définie positive si et seulement si toutes les valeurs propres de A sont strictement positives.
- A est semi-définie positive si et seulement si toutes les valeurs propres de A sont positives ou nuls.
- A est définie négative si et seulement si toutes les valeurs propres de A sont strictement négatives.
- A est semi-définie négative si et seulement si toutes les valeurs propres de A sont négatives ou nuls.
- A est définie négative si et seulement si $(-1)^k D_k > 0$ (c-à-d : ses n mineurs principaux diagonaux D_k sont alternativement < 0 (si k impaire) et > 0 (si k paire)).
- A est semi-définie négative si et seulement si tous ses mineurs principaux d'ordre k sont alternativement ≤ 0 (si k impaire) et ≥ 0 (si k paire).

6. Gradient :

Le gradient d'une fonction de \mathbb{R}^n dans \mathbb{R} est le vecteur constitué des dérivées partielles de la fonction par rapport à chaque variable :

$$\nabla f(x) = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{pmatrix}$$

La dérivée partielle de f par rapport à x_j est obtenue en dérivant f en considérant les autres variables comme des constantes.

3. Généralité sur les fonctions [26] :

3.1 Continuité :

Soient $S \subset \mathbb{R}^n$, $a \in S$ et $f: S \rightarrow \mathbb{R}$ une fonction.

- On dira que f est continue au point a si l'on a : $\lim_{x \rightarrow a} f(x) = f(a)$.
- On dira que f est continue sur S si elle est continue en tout point de S .

3.2 Dérivabilité :

Soient $S \subset \mathbb{R}^n$, $a \in S$ et $f: S \subset \mathbb{R} \rightarrow \mathbb{R}$ une fonction.

La fonction f est dérivable au point a ($a \in S$) s'il existe un réel noté $f'(a)$ tel que :

$$\lim_{h \rightarrow 0, h \neq 0} \frac{f(a+h) - f(a) - hf'(a)}{h} = 0$$

3.3 Différentiabilité des fonctions numérique :

Soit $f: S \rightarrow \mathbb{R}$, S un ouvert de \mathbb{R}^n et a un point de S .

- On dit que f est une fonction différentiable au point a s'il existe une application linéaire $L: \mathbb{R}^n \rightarrow \mathbb{R}$ telle que :

$$\lim_{h \rightarrow 0, h \neq 0} \frac{f(a+h) - f(a) - L(h)}{\|h\|} = 0$$

- f est différentiable en S si elle est différentiable en tous point de S .
- Si f est différentiable en a (resp. sur S) alors elle est continue en a (resp. sur S).

3.4 Fonction coercive :

La fonction f est dite coercive (ou infinie à l'infinie) si :

$$\lim f(x) = \infty \text{ quand } \|x\| \rightarrow \infty$$

3.5 Classe d'une fonction :

Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$.

On dit que f est de classe C^k sur $S : (f^k(s))$ si toutes ces dérivées partielles existent et sont continues sur S .

3.6 Point critique :

Soit $f : S \rightarrow \mathbb{R}$ une fonction de classe C^1 sur une partie S de \mathbb{R}^n .

On dit que $x^* \in S$ est un point critique de f si toutes les dérivées partielles s'annulent en x^* (équivalent à dire que le gradient de f est nul en x^* , équivalent à dire aussi que la différentielle de f est nulle en x^*).

3.7 Point selle (ou point col) :

Si f est une fonction définie sur une fonction définie sur un ouvert O de \mathbb{R}^n à valeurs dans \mathbb{R} , différentiable.

On dit que x^* est un point selle si x^* est un point critique et si f ne présente pas d'extrémum local en x^* .

2.2.2 Éléments de convexité

Le concept de convexité est d'une importance remarquable dans l'étude des problèmes d'optimisations.

2.2.2.1 Ensembles convexes [27]

Un ensemble $S \subseteq \mathbb{R}^n$ est dit convexe si et seulement si :

$$\forall (x, y) \in S^2 : (\lambda x + (1 - \lambda)y) \in S \quad \forall \lambda \in [0, 1].$$

En d'autre terme, si deux points x et y appartiennent à S , alors le segment joignant x et y est entièrement contenu dans S .

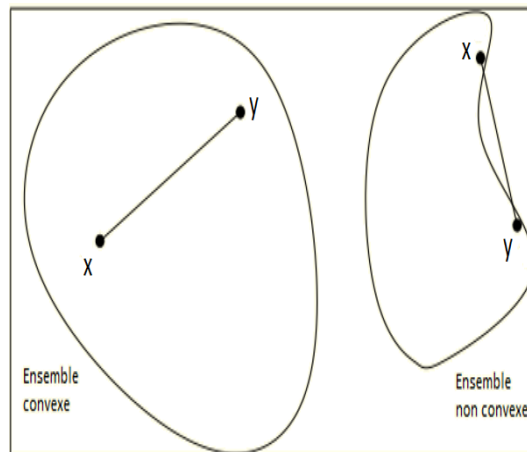


FIGURE 2.2 – Exemple sur l'ensemble convexe

L'ensemble de gauche est convexe car il contient le segment reliant n'importe quels deux de ses points. L'ensemble de droite n'est pas convexe.

2.2.2.2 Fonctions convexes [27]

Soient $S \subset \mathbb{R}^n$ un ensemble convexe et $f : S \rightarrow \mathbb{R}$ une fonction.

– La fonction f est dite convexe sur S si et seulement si :

$$\forall (x, y) \in S^2 : f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda)f(y) \quad \forall \lambda \in [0, 1].$$

– f est dite strictement convexe sur S si l'inégalité ci-dessus est toujours stricte $\forall x \neq y, \forall \lambda \in]0, 1[$.

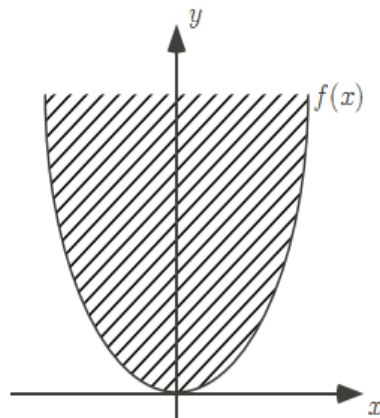


FIGURE 2.3 – Exemple de fonction convexe

2.2.2.3 Convexité en terme de matrice hessienne

Théorème [22]

Soit $S \subset \mathbb{R}^n$ un ensemble convexe et $f : S \rightarrow \mathbb{R}$ une fonction.
Et soit f est de classe c^2 sur S ,

- f est convexe sur $S \Leftrightarrow$ Pour tout $x \in S$, la matrice hessienne $\nabla^2 f(x)$ est semi-définie positive.
- f est strictement convexe \Leftrightarrow Pour tout $x \in S$, la matrice hessienne $\nabla^2 f(x)$ est définie positive.

2.2.3 Eléments d'extrémum

Soient l'ensemble $S \in \mathbb{R}_n$ et une fonction $f : S \rightarrow \mathbb{R}$.
Les minimums (maximums) locaux et globaux de f sur S sont définis de la manière suivante :

2.2.3.1 Minimum (Maximum) local [12]

On dit que x^* est un minimum local (resp.maximum local) de f sur S s'il existe $\epsilon > 0$ tel que :

$$f(x^*) \leq f(x) \text{ (resp. } f(x^*) \geq f(x)), \forall x \in S \text{ avec } \|x - x^*\| < \epsilon.$$

On dit que x^* est un minimum local strict (resp.maximum local strict) de f sur S s'il existe $\epsilon > 0$ tel que :

$$f(x^*) < f(x) \text{ (resp. } f(x^*) > f(x)), \forall x \in S \text{ avec } \|x - x^*\| < \epsilon/x \neq x^*.$$

Où $\|v\|$ désigne la deuxième norme du vecteur v .

2.2.3.2 Minimum (Maximum) global [12]

On dit que x^* est un minimum global (resp.maximum global) de f sur S si :

$$f(x^*) \leq f(x) \text{ (resp. } f(x^*) \geq f(x)), \forall x \in S.$$

On dit que x^* est un minimum global strict (resp.maximum global strict) de f sur S s'il existe $\epsilon > 0$ tel que :

$$f(x^*) < f(x) \text{ (resp. } f(x^*) > f(x)), \forall x \in S.$$

Exemple

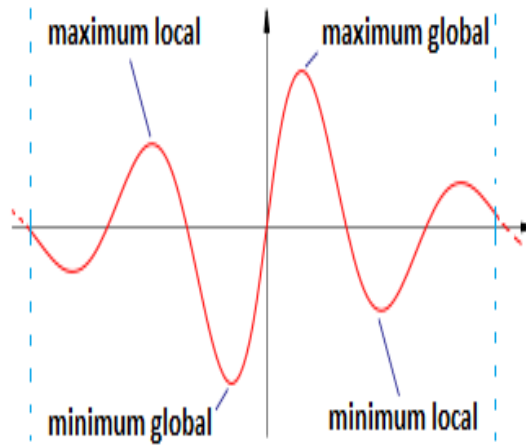


FIGURE 2.4 – graphe sur l'extremum

Remarque :

Dans le cas d'une fonction objectif convexe, il n'y a pas de distinction entre minimum local et global : tout minimum local est également global, comme l'établit le théorème suivant :

Théorème [11]

Soit $f : X \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ une fonction convexe définie sur l'ensemble convexe S . Alors, tout minimum local de f sur S est également un minimum global. Si f est strictement convexe, alors il existe au plus un minimum global de f .

2.2.3.3 Extremum en terme de matrice Hessienne

Théorème 01[9]

Soit $f : S \subset \mathbb{R}^n \rightarrow \mathbb{R}$ et $x_0 \in S$ un point critique de f . On suppose que la Hessienne $\nabla^2 f(x_0)$ existe. Alors :

- Si $\nabla^2 f(x_0)$ est définie positive alors $f(x_0)$ est un minimum local.
- Si $\nabla^2 f(x_0)$ est définie négative alors $f(x_0)$ est un maximum local.
- Si $\nabla^2 f(x_0)$ est indéfinie alors x_0 n'est ni un maximum local ni un minimum local : x_0 est un point selle.

2.2.4 Notions sur les algorithmes itératifs

2.2.4.1 Définition [6]

Considérons le problème suivant :

$$(P) = \begin{cases} \text{Min } f(x) \\ x \in S \end{cases}$$

Un algorithme itératif permettant de résoudre ce problème est un processus itératif générant une suite de vecteurs x^0, x^1, \dots, x^n de x en fonction d'une séquence d'instructions et d'une condition d'arrêt.

La production d'un vecteur x^{k+1} à partir d'un vecteur x^k (telle que x^k et x^{k+1} appartiennent à X) constitue une itération de l'algorithme.

Un tel algorithme est dit de descente, si le coût du vecteur généré à l'itération $k + 1$ est strictement inférieur au coût du vecteur généré à l'itération k , c'est-à-dire si $f(x^{k+1}) < f(x^k), \forall k \geq 0$

Etant donnée un vecteur x^k , et en appliquant la séquence d'instructions de l'algorithme, nous obtenons un nouveau vecteur x^{k+1}

2.2.4.2 Convergence d'un algorithme [6]

On dit qu'un algorithme itératif est globalement convergent si quelque soit le point de départ x^0 choisi, la suite $(x^k)_k$ engendrée converge vers un point satisfaisant une conditions d'optimalité du problème étudié.

2.2.5 Programmation mathématique

2.2.5.1 Définition [14]

Un programme mathématique dans \mathbb{R}^n est un problème d'optimisation sous contraintes de la forme :

$$(P) = \begin{cases} \text{Optimiser } f(x) \\ \text{Sous les contraintes} \\ x \in S \subset \mathbb{R}^n \end{cases}$$

Avec :

$x = (x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ est un vecteur des inconnues du problème.

La fonction f est appelée la fonction objectif ou fonction économique.

$x \in S$ sont les contraintes du problème (P) .

Un problème de programmation mathématique est dit linéaire s'il s'agit de la maximisation (ou de la minimisation) d'une fonction linéaire sous des contraintes purement linéaires, il s'agit donc d'un programme mathématique de la forme :

$$(PL) = \begin{cases} Opt(f) = C^t x \\ \text{Sous les contraintes} \\ Ax \leq b \\ x \geq 0 \end{cases}$$

Où : $C \in \mathbb{R}^n, b \in \mathbb{R}^m, x \in \mathbb{R}^n$ et A est une matrice de taille $m * n$.

l'ensemble des solutions réalisables est : $S = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\}$

Remarque : L'algorithme de simplexe ou la méthode des points intérieures permettent de résoudre les problèmes linéaires

Un problème de programmation mathématique est dit non linéaire si l'objectif ou les contraintes (ou les deux) sont des fonctions non linéaires.

Remarque : On dit que les contraintes sont non linéaires si au moins l'un des contraintes sont non linéaires.

2.2.5.1 Programmes mathématiques convexes

Théorème [19]

On dit que le problème (P) est convexe si :

- Le domaine de solution réalisable (S) est convexe,
- La fonction objectif f est convexe sur S .

Nombreux sont les problèmes de décision qui se ramènent à un modèle de programmation mathématique, dans ce travail nous nous sommes intéressés au cas où la fonction objectif à optimiser est non linéaire avec des contraintes (d'égalité et d'inégalité) linéaire, Ce qui nécessite la définition de quelques notions de programmation non linéaire.

2.2.6 Problème d'optimisation non linéaire

On distingue deux classes d'optimisation non linéaire :

2.2.6.1 Optimisation non linéaire sans contraintes[14]

Il s'agit de résoudre le problème :

$$(P) = \begin{cases} \text{Opt } f(x) \\ x \in \mathbb{R}^n \end{cases}$$

Où $f(x)$ est une fonction non linéaire quelconque définie sur \mathbb{R}^n .

Exemple :

Soit

$$(P) = \begin{cases} \text{Opt } f(x,y)=x^2 + y^2 + xy \\ (x, y) \in \mathbb{R}^2 \end{cases}$$

2.2.6.2 Optimisation non linéaire avec contraintes[14]

Dans ce cas le programme mathématique est de la forme :

$$(PC) = \begin{cases} \text{Opt } f(x) \\ x \in S \end{cases}$$

Où $S \subset \mathbb{R}^n$ et $f : S \rightarrow \mathbb{R}$.

Plus précisément, on s'intéresse à des problèmes (PC) de la forme générale :

$$(PC) = \begin{cases} \text{Min } f(x) \\ Sc : \\ h_i(x) = 0 \dots \dots \dots (1) \\ g_j(x) \leq 0 \dots \dots \dots (2) \end{cases}$$

Où $f : \mathbb{R}^n \rightarrow \mathbb{R}$ et $f : S \rightarrow \mathbb{R}$.

Soit S l'ensemble des points réalisables de (PC) :

$S = \{x \in \mathbb{R}^n ; h_i(x) = 0 \text{ et } g_j(x) \leq 0 ; i = 1, \dots, p ; j = 1, \dots, m\}, S \neq \emptyset$.

Avec :

1. La fonction f, h, g sont non linéaires et supposées continues et différentiables au moins une fois.
2. L'équation (1) désigne les contraintes d'égalité et l'équation (2) les contraintes d'inégalité.

2.2.7 Résolution des modèles non linéaires

Pour résoudre un modèle mathématique il faut passer par la réponse aux trois questions suivantes :

1. Existe-il une solution ? Cette solution est-elle unique ?
2. Comment caractériser cette (ou ces) solution(s) ?
3. Comment approcher de manière efficace une solution ?

Dans le reste du chapitre, nous allons présenter les éléments de réponses pour chaque question :

2.2.8 L'existence et l'unicité

Théorème 1 (existence) [1]

Si S est fermé et f est continue et coercive sur S alors le problème (PC) admet au moins une solution optimale globale $x^* \in S$.

Théorème 3 (unicité) [14]

Si l'ensemble S est convexe et f est une fonction strictement convexe sur S alors le problème (PC) admet au plus une solution optimale globale.

2.2.9 Caractérisation de la solution (Conditions d'optimalité)

2.2.9.1 Pourquoi avons-nous besoin des conditions d'optimalité ?

Afin d'analyser ou de résoudre de manière efficace un problème d'optimisation, il est fondamental de pouvoir disposer des conditions d'optimalité,

En optimisation mathématique, les conditions d'optimalité sont un ensemble d'équations, d'inéquations (c'est-à-dire des inégalités) et d'expressions diverses (par exemple, la copositivité de matrices) vérifiées par une solution d'un problème d'optimisation (on parle alors de conditions nécessaires d'optimalité) ou qui permettent d'affirmer qu'un point qui les vérifie est solution du problème d'optimisation considéré (on parle alors de conditions suffisantes d'optimalité), Ces expressions analytiques de l'optimalité sont utiles entre autres pour :

- Calculer les solutions d'un problème d'optimisation,
- Vérifier l'optimalité d'un point donné,
- Concevoir des algorithmes de résolution.

Pour écrire les conditions d'optimalité, on a besoin de quelques définitions :

Définition 1 :(Lagrangien généralisé)[14]

On appelle Lagrangien généralisé ou fonction de Lagrange généralisée associée au problème (PC) la fonction : $L : \mathbb{R}^n * \mathbb{R}^p * \mathbb{R}^m \rightarrow \mathbb{R}$

$$L(x) = L(x, \lambda, \mu) = f(x) + \sum_{i=1}^p \lambda_i h_i(x) + \sum_{j=1}^m \mu_j g_j(x)$$

Les vecteurs $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_p) \in \mathbb{R}^p$ et $\mu = (\mu_1, \mu_2, \dots, \mu_m) \in \mathbb{R}^m$ sont appelés les vecteurs des multiplicateurs de Lagrange.

Définition 2 :(Contrainte active)[14]

Une contrainte d'inégalité $g_k(x) \leq 0$ est dite active en x^* si $g_k(x^*) = 0$.
Il découle immédiatement de cette définition que $h_i(x) = 0$ sont actives $\forall i \in \overline{1, n}$.

Définition 3 :(qualification des contraintes)[14]

Les contraintes du problème (PC) sont qualifiées au point $x^* \in S$ si les gradients des contraintes actives en x^* :

$$\{\nabla h_i(x^*), i = 1, \dots, p\} \cup \{\nabla g_j(x^*); j = 1, \dots, m\} \text{ active } \}$$

sont linéairement indépendants.

2.2.9.2 Les conditions d'optimalité[15]

1. Condition nécessaire de premier ordre :

Théorème :(CN d'optimalité/Conditions de Karush-Kuhn-Tucker (1951)(KKT)) :

Soient :

- f, g et h trois fonctions non linéaires différentiables en x^* .
- Les contraintes du problème (PC) qualifiées au point x^* .

Si $x^* \in S$ est un point de minimum local de f sur S alors il existe $\lambda^* \in \mathbb{R}^p$ et $\mu^* \in \mathbb{R}^q$ tels que :

$$\begin{cases} \nabla_x L(x^*, \lambda^*, \mu^*) = 0 \\ h_i(x^*) = 0, & i = 1, \dots, p \\ \mu_j^* g_j(x^*) = 0, & j = 1, \dots, m. \\ \mu_j^* \geq 0, & j = 1, \dots, m \end{cases} \quad (**)$$

2. Condition suffisante de deuxième ordre :

Soit x^* un point admissible du problème (PC).
Supposons le problème (PC) convexe et les fonctions f, g et h sont différentiables en x^* .

S'il existe des multiplicateurs $(\lambda^*, \mu^*) \in \mathbb{R}^p \times \mathbb{R}^m$ tels que les conditions de Karush-Kuhn-Tucker soient vérifiées, alors x^* est un point de minimum global de f sur S .

2.2.10 Méthodes des résolutions

La plupart des méthodes existantes en programmation non linéaire sous contraintes peut se rattacher à deux familles des méthodes de résolution :

1. **Méthode analytique** [8]

Cette méthode fait partie des méthodes d'analyse.

En mathématiques, une méthode analytique ou appelée aussi une méthode de forme fermée difficile à résoudre et il donne des solutions exactes .

La méthode analytique est tout calcul ou bien toute méthode qui se basent sur des formules, des relations et des techniques mathématiques sans avoir recours aux procédures algorithmiques.

Exemple :

La résolution de système de KKT fait partie des méthodes analytiques.

2. **Méthode numérique** [16]

Méthode numérique dite aussi l'analyse numérique (numerical analysis en anglais) est une branche des mathématiques appliquées s'intéressant au développement d'outils et de méthodes numériques pour le calcul d'approximations de solutions de problèmes de mathématiques, qu'il serait difficile (voire impossible) d'obtenir par des moyens analytiques (méthode analytique) .

En pratique, l'analyse numérique se propose d'étudier les propriétés mathématiques des algorithmes et leur mise en œuvre (programmation) dont l'objectif de : concevoir et d'étudier des méthodes de résolution de certains problèmes mathématiques (en général issus de la modélisation de problèmes réels), et dont on cherche à calculer la solution ou son approximation à l'aide d'un ordinateur.

Choix de méthode :

La plupart des méthodes numériques appartiennent a deux types :

- (a) Méthodes directes (ou primales).
- (b) Méthodes utilisant la notion de dualité.

Les méthodes primales se caractérisent par le fait qu'elles opèrent directement sur le problème donné, elle engendrent une séquence de solutions (c'est-à-dire de points satisfaisant toutes les contraintes de problème) en assurant une décroissance monotone de la fonction à minimiser, elle présentent donc un avantage important : si le processus itératif est interrompu, elle procurent une solution approchée satisfaisant toutes les contraintes au problème . par contre, elle ont souvent l'inconvénient d'être délicates à mettre en œuvre, et la propriété de convergence globale est parfois difficile à garantir.

Les méthodes duales sont plus robustes et la convergence globale est souvent plus facile à obtenir, en contrepartie elle présente l'inconvénient de ne pas nécessairement garantir l'obtention d'une solution primale réalisable.

Nous nous contenterons ici d'étudier une méthode de premier type, c'est la méthode de linéarisation.

Principe des méthodes de linéarisation [15] :

Le succès de la méthode du simplexe pour la résolution des programmes linéaires, jusqu'à des tailles très importantes, incité des nombreux auteurs à développer des méthodes dites de linéarisation.

Le principe général en est simple : on remplace la résolution de problème non linéaire par la résolution d'une suite de programmes linéaires approximant, en un certain sens, le problème donné.

Ces méthodes utilisent essentiellement

L'approximation tangentielle :

Une fonction non linéaire continument différentiable f est remplacée au voisinage de $x = x^k$ par la fonction linéaire :

$$G = f(x^k) + \nabla h^T(x^k)(x - x^k) \text{ (équation du plan tangent en } x^k \text{ au graphe de la fonction } f \text{ ou l'approximation de Taylor limitée à l'ordre 1)}$$

Parmi les méthodes de linéarisation tangentielle, nous décrivons la méthode de Frank et Wolfe (1956) qui s'applique au cas des contraintes linéaires.

2.2.11 La méthode de Frank-Wolfe

En 1956, Frank et Wolfe ont développé un algorithme pour la résolution d'un problème d'optimisation quadratique qui a été généralisé a l'optimisation non linéaire convexe.

2.2.9.1 Principe de la méthode [14]

Soit le problème non linéaire suivant dont les contraintes sont purement linéaires :

$$(PC) = \begin{cases} \text{Min } f(x) \\ Ax \leq b \\ x \geq 0 \end{cases}$$

En effet, c'est le type de problème que nous allons rencontrer pour la résolution du problème d'affectation du trafic.

Une itération de la méthode comporte les étapes suivantes :

a) Calcul de la direction : On calcule la direction de recherche en résolvant un problème linéaire obtenu en remplaçant la fonction objectif par son approximation linéaire (Étapes 1 et 2 ci-dessous).

b) Calcul du pas : On calcule le pas à faire dans la direction en minimisant la fonction $f(x)$ le long de la direction déterminée en a) (pas 3 et 4 ci-dessous).

2.2.9.2 L'algorithme [3]

Algorithme 2 Algorithme de Franck-Wolfe

- 1: Comme seul l'objectif est non linéaire, on va remplacer $f(x)$ par son approximation de Taylor limitée à l'ordre 1 :

$$f(x) \sim f(x^k) + \nabla f(x^k)(x - x^k) = f(x^k) + \sum_{j=1}^n \frac{\partial f(x^k)}{\partial x_j} (x_j - x_j^k)$$

En éliminant les constantes, cela revient à minimiser la fonction linéaire :

$$g(x) = \sum_{j=1}^n \frac{\partial f(x^k)}{\partial x_j} x_j = \sum_{j=1}^n c_j x_j$$

- 2: On peut donc résoudre le problème linéaire :

$$(PCL) = \begin{cases} \text{Min } g(x) = \sum_{j=1}^n c_j x_j \\ Ax \leq b \\ x \geq 0 \end{cases}$$

Par l'algorithme du Simplexe. Soit x_{LP} sa solution optimale.

- 3: On détermine le pas dans la direction :

$$x^{k+1} = x^k + \alpha_k (x_{LP} - x^k).$$

avec $\alpha_k \in [0, 1]$ qui minimise cette fois la vraie fonction :

$$h(\alpha) = f(x^k + \alpha_k (x_{LP} - x^k))$$

On obtient un problème de minimisation unidimensionnelle.

- 4: Critère d'arrêt : si x^{k+1} et x^k sont suffisamment proches, stop.
-

2.2.9.3 Convergence de la méthode [14]

On suppose que f est continûment différentiable et que l'une des conditions est vérifiée :

- Le polyèdre $S = \{x/Ax = b; x \geq 0\}$ est bornée ;
- f est coercive.

Alors pour tout point de départ $x_0 \in S$, la méthode de Frank et wolf converge vers un optimum local du problème (PC).

Conclusion

Ce chapitre nous a permis de définir toutes les notions fondamentales dont nous avons besoin dans notre travail, dans le chapitre suivant nous allons détailler le problème d'affectation du trafic urbain.

3

Problème d'affectation du trafic urbain (PATU)

Introduction

Ce chapitre (Le cœur de notre travail) sera consacré à la définition du problème d'affectation du trafic urbain (PATU), et à la formulation mathématique du problème sur lequel nous allons appliquer les caractéristiques de la solution étudiées dans chapitre précédent.

3.1 Réseau de trafic urbain (le terrain de problème)

3.1.1 Les paramètres du réseau routier

Pour formuler le problème d'affectation, un réseau routier sera défini par les paramètres suivants :

a. Origine [23] :

C'est un sommet qui désigne le point de départ du voyage d'un usager, c-à-d le lieu où l'usager entre sur le réseau routier réel.

b. Destination [23] :

C'est un sommet qui désigne le point d'arrivée du voyage d'un usager, c-à-d le lieu où l'usager quitte la circulation une fois parvenu à sa destination qui correspond à une sortie du réseau réel.

c. Itinéraire [23] :

Est un chemin continue d'arc reliant deux sommets entre eux, respectivement

l'origine et la destination de l'itinéraire.

d. Fonction de performance (temps de parcours) [3] :

Soit $G = (X, U, T)$ un réseau,

Le temps de parcourt de l'arc u ($u \in U$) est une fonction qui dépend de nombre d'usagers (flux) qui traverse cette arc (plus l'arc est chargée, plus le temps de traversée augmente).

Elle est croissante depuis le temps de traversée à vide jusqu'à une asymptote verticale correspondant à la capacité de l'arc.

$$t_u = t_u(\varphi_u)[3]$$

Avec (φ_u) et (t_u) représentent respectivement le flux et le temps de traversée de l'arc u .

Illustration graphique :

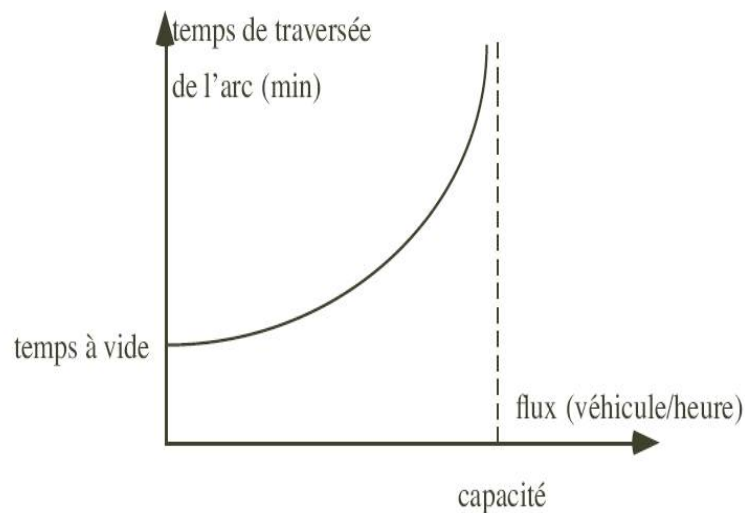


FIGURE 3.1 – Fonction de performance

f. Demandes origine-destination [3] :

C'est une donnée essentielle pour les modèles d'affectation de trafic, elle est modélisé par une matrice origine-destination notée q_{od} tel que les lignes représentent les sommets d'origines, et les colonnes représentent les sommets destination, représentant le taux de voyages (nombre de véhicules) désirent, partant de l'origine o , rejoindre la destination d .

La matrice origine-destination est une matrice variante selon une unité de temps (souvent l'heure) qui est calculée à l'aide d'une estimation de trafic selon des différentes techniques.

Dans notre étude, nous supposons que cette matrice est une donnée (on ne s'intéresse pas a la méthode d'estimation) fixée (dans une heure précisé).

3.1.2 Réseau de trafic routière

Un réseau de trafic est un graphe orienté $G = (X, U, T)$ et $X_{od} \subseteq X$ telle que :

- L'ensemble des sommets X représentent les croisements des rues .
- L'ensemble des arcs U representent les liaisons routières.
- L'ensemble T représente ici les temps de parcourt des arcs ($t_u; \forall u \in U$).
- X_{od} :l'ensemble des sommets origine et destination c-à-d $X_{od} = O \cup D$ tel que :
 - (a) O et D représentent respectivement l'ensemble des origines et l'ensemble des destinations.
 - (b) od : est une paire (couple) origine-destination ($od \in OD$).
 - Avec OD :l'ensemble des couples origine-destination .

Exemple :

Soit $G = (X, U, T)$ et X_{od} un réseau routier où :

$$X = \{1, 2, 3, 4, 5, 6\}$$

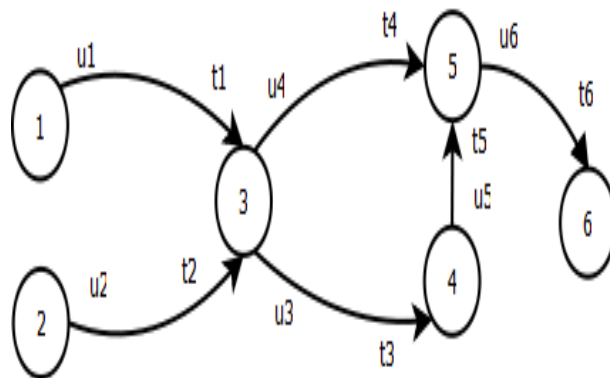
$$U = \{u_1, u_2, u_3, u_4, u_5, u_6\}.$$

$$T = \{1, 2, 2 + \varphi_3, 1 + 2\varphi_4, 3, 2\}.$$

$$X_{od} = \{1, 2, 6\}$$

$$O = \{1, 2\} \text{ et } D = \{6\}.$$

$$OD = \{(16), (26)\}.$$



Exemple de réseau de trafic

3.2 La notion d'équilibre

C'est quoi un équilibre ?

L'équilibre est l'état de repos, position stable d'un système obtenue par l'égalité de deux forces, de deux poids qui s'opposent ; Mettre les plateaux d'une balance en équilibre. C'est une notion qui est utilisée dans de nombreux domaines.

3.2.1 Notion d'équilibre dans l'économie

La notion d'équilibre entre offre et demande sur un marché est bien connue en économie,

- le comportement des producteurs sur un marché est caractérisé par une fonction d'offre qui exprime le montant de bien produit en fonction du prix (plus le prix est élevé, plus la production est importante).
- Le comportement des consommateurs est caractérisé par la fonction de demande qui exprime la quantité totale demandée en fonction du prix (plus le prix est élevé, plus la demande est faible).

Dans les conditions courantes d'échange sur un marché, nous observons que les quantités offertes augmentent et que les quantités demandées diminuent lorsque les prix s'élèvent. A l'inverse, les quantités offertes diminuent et que les quantités demandées augmentent lorsque les prix baissent.

La figure (3.4) : illustre le point d'équilibre entre l'offre et la demande.

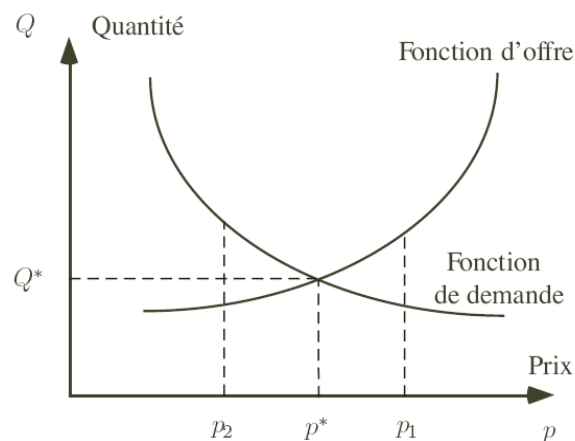


FIGURE 3.2 – Point d'équilibre du marché.

En règle générale l'excès de l'offre amène les offreurs à baisser leurs prix sous l'effet de la concurrence. A l'inverse un excès de la demande par rapport à l'offre provoque

la concurrence entre les acheteurs et conduit à l'augmentation des prix.

Dans les deux cas, la variation du prix se poursuit jusqu'à correction complète du déséquilibre entre l'offre et la demande. Le prix se stabilise lorsque l'offre est égale à la demande ; on parle alors de prix d'équilibre (c'est le point d'intersection des courbes (p^*, Q^*) dans la figure)

3.2.2 Notion d'équilibre dans le domaine du trafic urbain

L'équilibre en trafic urbain est l'ajustement entre l'offre du trafic et la demande des déplacements, selon les principes de Wardrop (1952) (c'est ce que nous allons étudier dans la suite de ce chapitre).

- **L'offre** : est représenté par le réseau routier (par route, par l'infrastructure routière et autoroutière).
- **La demande** : est donnée par le nombre de véhicules qui veulent se rendre d'un point de départ vers un point d'arrivée. on représente généralement ceci par une matrice origine-destination qui indique en position (o, d) le nombre de véhicules désirant se rendre de o vers d tel que o le sommet origine et d est le sommet destination.

3.3 Problématique

Comme souligner précédemment, les voyageurs entre les différentes paires origine-destination du réseau du trafic, choisissent leurs itinéraires de sorte qu'ils réduisent au minimum leur temps de parcours. Ce comportement est égoïste, et cela produit des congestions dans plusieurs parties du réseau.

Etant donnée une matrice de demande q_{od} fixe en un temps fixe (cas statique), et un réseau routier, supposant que les voyageurs sont identiques et connaissent parfaitement le réseau, le PATU consiste à trouver la meilleure manière pour distribuer les voyageurs, afin que chaque automobiliste soit affecté au plus court chemin menant à sa destination, Il s'agit donc de répartir la demande de déplacement sur le réseau en tenant compte du choix des usagers et l'offre de systèmes de transports.

Le problème de l'affectation du trafic urbain (PATU) est donc un outil pour modéliser les interactions entre congestions et décisions de déplacement des usagers pour obtenir les flux de véhicules résultants sur chacun des axes du réseau de trafic urbain.

L'importance du problème se situe, d'une part, dans la modélisation des comportements des usagers et d'autre part, dans la modélisation de l'écoulement du trafic consistant à représenter le choix individuel des usagers en termes de déplacement.

L'hypothèse sur le comportement des usagers est que les usagers sont rationnels chacun cherchant à maximiser son propre intérêt, C'est-à-dire minimiser son coût de déplacement. Cependant, les usagers n'ont pas d'information parfaite sur l'état du trafic sur le réseau et dans ce cas, la décision de déplacement sera en terme du coût de déplacement (précisément le temps de parcourt) est en fonction du choix de tous les usagers. L'argument principal de ce problème est donc se manifeste par l'interaction entre le comportement des usagers et le système de trafic.

En bref, position du problème est de l'objectif de notre travail déterminer comment les voitures se répartissent sur le réseau. En d'autre terme, nous déterminerons le flot optimale sur chacun des axes du réseau (et, en conséquence, le temps minimal de traversée) en passant par le PATU.

Deux questions sous-jacentes se décèlent d'elles-mêmes et auxquelles nous devons apporter des réponses :

- Comment modéliser le problème d'affectation du trafic urbain sous forme d'un problème d'optimisation mathématique ?
- Comment mettre en œuvre numériquement l'équilibre ou la solution du PATU.

C'est, en quelque sorte, nous allons apporter les éléments de repenses aux questions de résolution analytique et numérique du PATU.

3.4 Les conditions de Wardrop

En 1952, Wardrop suppose que les usagers sont économiquement parfaits : ils maîtrisent et connaissent parfaitement le réseau et en perçoivent parfaitement les coûts des itinéraires, qu'ils sont rationnels, recherchent leur plus court chemin et le choisissent sans contrainte extérieure. Il émet l'hypothèse supplémentaire que tous les usagers sont identiques.

En mathématiques, le principe de Wardrop s'énonce ainsi : pour chaque couple d'origine-destination, les chemins utilisés ont le même coût généralisé et celui-ci est inférieur aux coûts des chemins non utilisés. Si, pour un couple origine destination, plusieurs itinéraires sont utilisés, leurs coûts généralisés sont égaux.

Le coût généralisé est une formule mathématique prenant en compte tous les critères de calcul du chemin (temps, distance, énergie utilisée, capacité , etc.). Il est principalement utilisé pour l'étude des réseaux de transport connaissant un niveau de congestion élevé.

Remarque : dans notre travail le coût généralisé est réduit au temps de parcourt.

L'hypothèse de l'utilisateur parfaitement rationnel englobe celle régissant le principe de choix individuel défini précédemment car non seulement les usagers cherchent à minimiser le coût de leur déplacement, mais ils y parviennent effectivement.

Pour Wardrop, l'équilibre se définit par le fait que tous les usagers d'un même couple od subissent tous le même coût, qui est le coût minimal possible. Il existe deux principes de wardrop :

1^{er} principe de wardrop [16] :

" À l'équilibre, les coûts générales sur chaque itinéraire donné sont égaux et inférieures aux coûts générales sur les itinéraires non utilisés "

2^{ème} **principe de wardrop [16] :**

" L'équilibre, bien qu'il corresponde à la min. du coût individuel généralisé, ne conduit pas forcément à la min. du total du coût généralisé pour l'ensemble des usagers "

Wardrop en 1952 déclare dans son premier principe que le réseau atteint un état d'équilibre, appelé équilibre usager.

3.4.1 L'équilibre usager de Wardrop

Le premier principe de wardrop signifie que si l'itinéraire p du couple $od \in X_{od}$ est utilisée, alors son coût est minimale :

$$q_p^{od} > 0 \Rightarrow C_p^{od} = C_{min}^{od} \forall p, od$$

Avec :

C_p^{od} : coût de l'itinéraire p .

C_{min}^{od} : coût minimal sur l'ensemble des itinéraires du couple od .

Cette situation est bien un état d'équilibre car aucun usager ne subit un coût de déplacement qu'il peut diminuer. En vertu du choix individuel, les usagers n'ont donc pas intérêt à changer d'itinéraire.

Il est à remarquer que la relation précédente n'est qu'une implication, et non une équivalence : un itinéraire qui a un coût minimal n'est pas nécessairement empruntée.

3.5 Formulation Mathématique

La plupart des modèles d'affectation statique du trafic routier sont formulés sur la base de premier principe de (Wardrop 1952),

Parmi ces modèles d'affectation statique, nous nous intéressons au modèle déterministe homogène de WARDROP-BECKMANN,

En 1956, Beckmann et al [3] ont été les premiers à proposer une formulation mathématique non linéaire pour l'équilibre statique entre offre et demande de déplacement sur un réseau de trafic.

3.5.1 Model de Beckmann 1956

On considère un réseau routier $G = (X, U, T)$, où l'ensemble X représente les sommets, et l'ensemble U représente les arcs sur lequel chaque arc $u \in U$, a un

temps de parcours associé $t = t_u(\varphi_u)$ qui dépend de flux total φ_u sur l'arc u (fonction de performance de l'arc u).

Notons O l'ensemble des sommets origine ($|O| = s$) et D l'ensemble des sommets destination ($|D| = t$), où la matrice origine destination est noté q , autrement dit q_{od} est le taux de voyage entre l'origine o et la destination d exprimé en véhicules par unité de temps (souvent l'heure).

Pour modéliser le problème d'affectation de trafic urbain sous forme d'un modèle mathématique, on a besoin d'illustrer quelques relations :

1- Le cout de parcourt :

Soit p un chemin quelconque entre l'origine o et la destination d dans le réseau $G = (X, U)$.

Le cout de parcours (ici réduit au temps de parcours) d'un chemin p est la somme des temps de parcours des arcs successifs du chemin entre o et d :

$$c_p = \sum_{u \in U} t_u \delta_{u,p} \quad (3.1)$$

Avec $\delta_{u,p}$ est la matrice d'incidence *arc/chemin* indiquant à quelle chemin p un arc u appartient. elle est définie comme suit :

$$\delta_{u,p} = \begin{cases} 1, & \text{si l'arc } u \text{ appartient au chemin } p \\ 0, & \text{Sinon.} \end{cases}$$

La forme matricielle :

La relation (3.1) se réécrit matriciellement par :

$$C = T\Delta$$

Où :

- $C = (c_1, c_2, \dots, c_r)$ est le vecteur ligne résultant dans les éléments sont les couts des chemins $p_i, i = \overline{1, r}$, avec r est le nombre de chemins reliant o et d dans le réseau $G = (X, U, T)$.
- $T = (t_1, t_2, \dots, t_m)$ est un vecteur linge dans les éléments sont les temps de parcourt des arcs $U_j, j = \overline{1, m}$, avec m est la taille de réseau $G = (X, U, T)$.
- $\Delta = \delta_{u,p}$ est la matrice d'incidence *arc/chemin* .

2- La relation entre les flux des arcs $u \in U$ et les flux des chemins p :

Soit φ_u un flux sur un arc u dans le réseau $G = (X, U, T)$, φ_u sera décomposé en la somme des flux des différents chemins p qui passent par cet arc :

$$\varphi_u = \sum_p z_p \delta_{u,p} \quad \forall u \in U \quad (3.2)$$

Avec : z_p : note le flux du chemin p .
 $\delta_{u,p}$: est la matrice d'incidence *arc/chemin* .

La forme matricielle :

La relation (3.2) se réécrit matriciellement par :

$$\varphi = \Delta Z$$

Où :

- $\varphi = \begin{pmatrix} \varphi_{u_1} \\ \varphi_{u_2} \\ \vdots \\ \varphi_{u_m} \end{pmatrix}$ est le vecteur colonne résultant, dont les éléments sont les flux des arcs $u_j, j = \overline{1, m}$ avec , m est la taille du réseau $G = (X, U, T)$.

- $\Delta = \delta_{u,p}$ est la matrice d'incidence *arc/chemin* .

- $Z = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_r \end{pmatrix}$ est un vecteur colonne, dont les éléments sont les flux des chemins p_k ,
 $k = \overline{1, r}$, avec r est le nombre des chemins dans le réseau $G = (X, U)$.

3- La condition de Conservation du flux :

La demande total entre o et d (od) est égale la somme des flux sur tous les chemins p reliant la paire od .

$$\sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \quad (3.3)$$

Où $\delta'(p, od)$ est une matrice d'incidence *chemins/paires* indiquant à quelle paire od un chemin p appartient, elle est définie comme suit :

$$\delta'(p, od) = \begin{cases} 1, & \text{si } p \in P_{od} \\ 0, & \text{Sinon.} \end{cases}$$

Avec : P_{od} est l'ensemble des chemins reliant od .

La forme matricielle :

La relation (3.3) se réécrit matriciellement par :

$$Z^T \Delta' = q$$

Où :

- $Z^T = (z_1, z_2, \dots, z_r)$ est un vecteur ligne dans les éléments sont les flux des chemins p_k , $k = \overline{1, r}$, avec r est le nombre des chemins reliant o et d dans le réseau $G = (X, U, T)$.
- $\Delta' = \delta'(p, od)$ est la matrice d'incidence *chemins/paires*
- $q = (q_{od_1}, q_{od_2}, \dots, q_{od_v})$ est un vecteur ligne dans les éléments sont les taux de voyages entre l'origine et la destination od_i , $i = \overline{1, v}$ où v le nombre de couples origine-destination.

Modèle mathématique :

Beckmann et al. [16] ont montré que le flot correspondant à un équilibre d'utilisateur peut être obtenu en résolvant le problème de programmation non linéaire suivant :

$$(P) = \begin{cases} \text{Min}f(\varphi) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w) dw \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \end{cases} \quad [3]$$

Par remplacement des φ_u par leur valeur (d'après la relation (3.2)) le problème devient :

$$(P) = \begin{cases} \text{Min}f(\varphi) = \sum_{u \in U} \int_0^{\sum_p z_p \delta_{u,p}} t_u(w) dw \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \dots \dots \dots (1) \\ z_p \geq 0 \quad \forall p \dots \dots \dots (2) \end{cases}$$

Interprétation du modèle :

– La fonction objectif est la somme des intégrales des fonctions de performance des arcs, supposer continu et de classe c^2 , cette fonction est purement mathématique ,elle n'a pas d'interprétation économique [16].

– Le premier ensemble des contraintes (1) représente les équations de conservation du flux, ces contraintes signifient que la somme des flux sur tous les chemins p reliant la paire od est égale au taux du voyage entre o et d .

Remarque :l'ensemble des contraintes (1) comporte l contraintes ($|od| = l$).

– Le deuxième ensemble des contraintes (2) décrit les contraintes de non-négativité, indiquant simplement que les flux des chemins doivent être positive ou nuls(le nombre de voyageurs en réalité) c'est-à-dire elle assure que la solution soit physiquement réaliste.

Remarque :l'ensemble des contraintes (2) comporte r contraintes où r est le nombre des chemins reliant od dans le réseau $G = (X, U, T)$.

Le modèle de Beckmann et al (1956) se place dans les deux hypothèses suivantes :

– **Hypothèse 1** : Nous supposons que le temps de traversée d'un arc n'est fonction qu'une du flux de cet arc, Mathématiquement :

$$\frac{\partial t_u(\varphi_u)}{\partial \varphi_{u'}} = 0, \forall u \neq u' \quad t.q \quad u, u' \in U \tag{3.4}$$

– **Hypothèse 2** : Les fonctions de performance des arcs sont strictement croissantes :

$$\frac{\partial t_u(\varphi_u)}{\partial \varphi_u} > 0, \forall u \in U \tag{3.5}$$

3.5.2 Caractérisation de la solution sur le modèle de Beckman

3.5.2.1 Analyse de modèle de Beckmann

Soit (P) le modèle de Beckmann suivant :

$$(P) = \begin{cases} \text{Min} f(\varphi) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w) dw \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \dots \dots \dots (1) \\ z_p \geq 0 \quad \forall p \dots \dots \dots (2) \end{cases} \quad [3]$$

L'ensemble des solutions réalisables est donnée par :

$$S = \{ \varphi_u \geq 0 \quad \text{telque} \quad \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \quad \text{et} \quad z_p \geq 0 \quad \forall p \}$$

1- Non linéarité :

Le modèle de Beckmann se classe dans la famille des modèles mathématiques non linéaire, en effet, il s'agit d'une fonction objectif non linéaire (la somme des intégrales des fonctions de performance des arcs) avec des contraintes purement linéaires.

2- Qualification des contraintes :

Théorème : [14]

Pour que la condition de qualification des contraintes (QC) soit vérifiée en tout point $x \in X$, il suffit que toutes les contraintes soient linéaires (Karlin 1959) .

D'après ce théorème la condition de qualification des contraintes est vérifiée dans le modèle de Beckmann (P) .

3- Convexité :

Nous rappelons qu'un programme mathématique est convexe si et seulement si :

1. L'objectif est convexe.
2. Le domaine de solutions réalisables est aussi convexe.

Est-ce que le domaine réalisable est convexe ?

Comme le domaine de solutions réalisables est linéaire, donc le polyèdre du domaine de solution est convexe.

Est-ce que la fonction objective est convexe ?

D'après théorème de convexité en terme de matrice Hessienne [22] :

f est convexe $\Leftrightarrow \forall S$, la matrice Hessienne $\nabla^2 f_\varphi$ est semi-définie positive.

La matrice Hessienne s'écrit comme suit :

$$\nabla^2 f_\varphi = \left(\frac{\partial^2 f}{\partial \varphi_i \partial \varphi_j}(\varphi) \right)_{i=1, \dots, n; j=1, \dots, n} = \begin{pmatrix} \left(\frac{\partial^2 f}{\partial \varphi_1^2} \right) (\varphi) & \left(\frac{\partial^2 f}{\partial \varphi_1 \partial \varphi_2} \right) (\varphi) & \cdot & \cdot & \cdot & \left(\frac{\partial^2 f}{\partial \varphi_1 \partial \varphi_n} \right) (\varphi) \\ \left(\frac{\partial^2 f}{\partial \varphi_2 \partial \varphi_1} \right) (\varphi) & \left(\frac{\partial^2 f}{\partial \varphi_2^2} \right) (\varphi) & \cdot & \cdot & \cdot & \left(\frac{\partial^2 f}{\partial \varphi_2 \partial \varphi_n} \right) (\varphi) \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \left(\frac{\partial^2 f}{\partial \varphi_n \partial \varphi_1} \right) (\varphi) & \left(\frac{\partial^2 f}{\partial \varphi_n \partial \varphi_2} \right) (\varphi) & \cdot & \cdot & \cdot & \left(\frac{\partial^2 f}{\partial \varphi_n^2} \right) (\varphi) \end{pmatrix}$$

Les dérivées du premier ordre :

$$\frac{\partial f(\varphi)}{\partial \varphi_u} = \frac{\partial \sum_{u''} \int_0^{\varphi_{u''}} t(w).dw}{\partial \varphi_u} = \sum_{u''} \frac{\partial \int_0^{\varphi_{u''}} t(w).dw}{\partial \varphi_u} = t_u \quad \forall u \in UCar :$$

$$\frac{\partial \int_0^{\varphi_{u''}} t(w).dw}{\varphi_u} = \begin{cases} t_u(\varphi_u) & \text{Si } u = u'' \\ 0 & \text{Sinon} \end{cases}$$

Les dérivées du second ordre :

D'après l'hypothèse 1 :

$$\frac{\partial^2 f(\varphi)}{\partial \varphi_u \partial \varphi_{u'}} = \frac{\partial t_u(\varphi_u)}{\partial \varphi_{u'}} = \begin{cases} t'_u(\varphi_u) & \text{Si } u = u' \\ 0 & \text{Sinon} \end{cases}$$

Ce qui donne une matrice Hessienne des dérivées partielles secondes diagonale suivant :

$$\nabla^2 f_u = \begin{pmatrix} t'_1(\varphi_1) & \cdot & \cdot & \cdot & \cdot \\ \cdot & t'_2(\varphi_2) & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & t'_n(\varphi_n) \end{pmatrix}$$

Cette matrice est définie positive car elle est diagonale et tous ses éléments diagonaux sont strictement positifs par l'hypothèse 2, donc la fonction objective f est strictement convexe ce qui implique que f est convexe.

3.5.2.2 l'existence de solution :

D'après le théorème de l'existence ([1]), pour montrer que le problème (P) admet au moins une solution optimale il faut vérifier ces conditions :

1. Le domaine des solutions réalisable S est fermé.
2. f est continue sur S .
3. f est coercive sur S .

Vérification :

1. Nous avons le domaine de solutions réalisables S est un polyèdre fermé (les contraintes sont linéaires et forment un polyèdre).
2. la fonction $f(\varphi_u)$ est continue sur le domaine S (voir section précédent Interprétation du modèle).
3. $f(\varphi_u) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w) dw$ est une fonction croissante, en effet :
Le temps de parcourt de l'arc u ($t_u(\varphi_u)$) est une fonction croissante (fonction de performance), et nous avons l'intégrale d'une fonction croissante est une fonction croissante.
Donc lorsque φ_u tend vers l'infini, $f(\varphi_u)$ aussi tend vers l'infini $\implies \lim f(\varphi_u) = \infty$ lorsque $\|\varphi_u\| \rightarrow \infty \implies f$ est une fonction coercive.

Les trois conditions (1), (2) et (3) sont vérifiées donc le problème (P) admet au moins une solution optimale globale.

3.5.2.3 Unicité de la solution

D'après le théorème de l'unicité [14], pour montrer que la solution est unique il faut montrer que la fonction objective f est convexe et le domaine de définition est strictement convexe, ce que nous avons déjà montré dans la partie précédente (analyse du problème de Beckmann) donc le problème de Beckmann admet une solution unique.

3.5.2.4 Condition d'optimalité :

Nous allons montrer que le flot qui résout ce problème (P) satisfait aussi les conditions de Kuhn et Tucker.

Pour vérifier les conditions d'optimalité, nous formulons le Lagrangien correspondant à (P) :

$$\text{Le modèle}(P) \iff \begin{cases} \text{Min} f(\varphi) = \sum_u \int_0^{\varphi_u} t_u(w) dw \\ q_{od} - \sum_p z_p \delta'_{p,od} = 0 & \forall od \\ -z_p \leq 0 & \forall p \end{cases}$$

$$L(z, \mu, \lambda) = f(\varphi(z)) + \sum_{od} \lambda_{od} (q_{od} - \sum_p z_p \delta'_{p,od}) - \sum_p \mu_p z_p$$

λ_{od} : Le 1^{er} multiplicateur de Lagrange associé à l'équation de conservation du flux de la paire origine-destination (première contrainte du modèle).

μ_p : Le 2^{ème} multiplicateur de Lagrange associé à la 2^{ème} contrainte du modèle.

Les condition du premier ordre :

$$\begin{cases} \frac{\partial L(z, \lambda, \mu)}{\partial z_p} = 0 \\ \mu_p z_p = 0 \\ \mu_p \geq 0 \end{cases}$$

On a $\forall p \in P_{od}$

$$\begin{aligned} \frac{\partial L(z, \lambda, \mu)}{\partial z_p} &= \frac{\partial(f(\varphi(z)) + \sum_{od} \lambda_{od} (q_{od} - \sum_{p' \in P_{od}} z_{p'} \delta'_{p',od}) - \sum_{p' \in P_{od}} \mu_{p'} z_{p'})}{\partial z_p} \\ &= \frac{\partial(f(\varphi(z)))}{\partial z_p} + \frac{\partial(\sum_{od} \lambda_{od} (q_{od} - \sum_{p' \in P_{od}} z_{p'} \delta'_{p',od}))}{\partial z_p} - \frac{\partial(\sum_{p' \in P_{od}} \mu_{p'} z_{p'})}{\partial z_p} \\ &= L_1 + L_2 - L_3 \end{aligned}$$

Calculons chacun des termes de la dérivée partielle du Lagrangien par rapport à z_p :

$$L_1 = \frac{\partial(f(\varphi(z)))}{\partial z_p} = \sum_{u \in U} \left(\frac{\partial f}{\partial \varphi_u} \right) \left(\frac{\partial \varphi_u}{\partial z_p} \right)$$

Par remplacement de f et φ respectivement par leurs équations :

$f(\varphi) = \sum_{u'} \int_0^{\varphi_{u'}} t_{u'}(w) dw$ et $\varphi = \sum_{p'} z_{p'} \delta_{u,p'}$ (voir relation 3.2) nous obtenons :

$$L_1 = \sum_{u \in U} \left(\frac{\partial \sum_{u' \in U} \int_0^{\varphi_{u'}} t_{u'}(w) dw}{\partial \varphi_u} \right) \left(\frac{\partial \sum_{p' \in P_{od}} z_{p'} \delta_{u,p'}}{\partial z_p} \right) \dots\dots\dots (eq1)$$

Nous introduisons la formule de la dérivée suivante $((u + v)' = u' + v')$ dans l'équation (eq1) pour obtenir :

$$L_1 = \sum_{u \in U} \left(\sum_{u' \in U} \frac{\partial \int_0^{\varphi_{u'}} t_{u'}(w) dw}{\partial \varphi_u} \right) \left(\sum_{p' \in P_{od}} \frac{\partial z_{p'} \delta_{u,p'}}{\partial z_p} \right)$$

Ayant obtenu une dérivée suivie d'une intégrale l'équation L_1 devient :

$$L_1 = \sum_{u \in U} \sum_{u'} t_{u'}(\varphi_{u'}) \left(\sum_{p' \in P_{od}} \delta_{u,p'} \right)$$

$$L_1 = \sum_{u \in U} (t_u)(\delta_{u,p}) = c_p \text{ (A partir de relation (3.1))}$$

d'autre part :

$$L_2 = \frac{\partial (\sum_{od} \lambda_{od} (q_{od} - \sum_p z_p \delta'_{p,od}))}{\partial z_p}$$

de plus ;

$$L_2 = \frac{\partial (\sum_{od} (\lambda_{od} q_{od} - \lambda_{od} \sum_p z_p \delta'_{p,od}))}{\partial z_p}$$

Ce qui donne :

$$L_2 = \frac{\partial (\sum_{od} (\lambda_{od} q_{od} - \sum_p \lambda_{od} z_p \delta'_{p,od}))}{\partial z_p}$$

Par la suite

$$L_2 = \frac{\partial (\sum_{od} \lambda_{od} q_{od} - \sum_{od} \sum_p \lambda_{od} z_p \delta'_{p,od})}{\partial z_p}$$

A partir de l'application de cette loi de la dérivée $((u + v)' = u' + v')$ on trouve :

$$L_2 = \frac{\partial \sum_{od} \lambda_{od} q_{od}}{\partial z_p} - \frac{\partial \sum_{od} \sum_p \lambda_{od} z_p \delta'_{p,od}}{\partial z_p}$$

On remarque que : $\frac{\partial \sum_{od} \lambda_{od} q_{od}}{\partial z_p} = 0$ car $\sum_{od} \lambda_{od} q_{od}$ représente une constante par rapport à z_p donc on obtient :

$$L_2 = - \frac{\partial \sum_{od} \sum_p \lambda_{od} z_p \delta'_{p,od}}{\partial z_p}$$

A partir de l'application de cette loi de dérivée $((u + v)' = u' + v')$ on trouve :

$$\begin{aligned} L_2 &= - \sum_{od} \sum_p \frac{\partial(\lambda_{od} z_p \delta'_{p,od})}{\partial z_p} \\ &= - \sum_{od} \sum_p \lambda_{od} \delta'_{p,od} = -\lambda_{od} \end{aligned}$$

Où od est la paire à laquelle appartient le chemin p .

Et :

$$L_3 = \frac{\partial(\sum_{p' \in P_{od}} \mu_{p'} z_{p'})}{\partial z_p}$$

A partir de l'application de cette loi de dérivée $((u + v)' = u' + v')$ on trouve :

$$L_3 = \sum_{p' \in P_{od}} \frac{\partial(\mu_{p'} z_{p'})}{\partial z_p} = \mu_p$$

Donc les conditions du premier ordre s'écrivent :

$$\begin{cases} c_p - \lambda_{od} - \mu_p = 0 \Rightarrow (\mu_p = c_p - \lambda_{od}) \\ \mu_p z_p = 0 \\ \mu_p \geq 0 \end{cases}$$

On peut éliminer μ_p du système pour obtenir :

$$\begin{cases} c_p - \lambda_{od} \geq 0 & \dots(1) \\ z_p(c_p - \lambda_{od}) = 0 & \dots(2) \end{cases}$$

Remarquons que :

les conditions (2) impliquent que :

$$\text{si } z_p > 0, \text{ alors } c_p = \lambda_{od}$$

car : $z_p > 0$ donc a partir de (2) on aura $(c_p - \lambda_{od}) = 0$ alors $c_p = \lambda_{od}$

D'autre part, les inéquations (1) impliquent que :

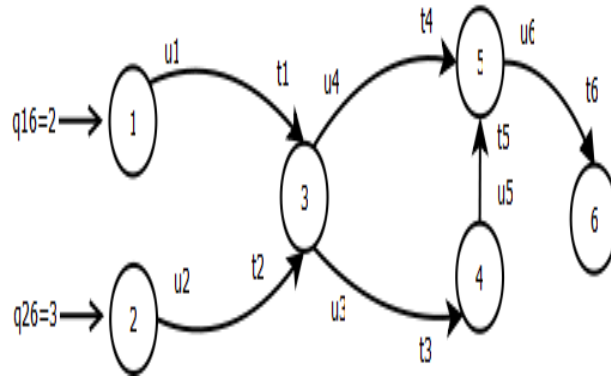
$$\text{si } z_p = 0, \text{ alors } c_p \geq \lambda_{od}$$

$$\begin{cases} c_p - \lambda_{od} \geq 0 \\ z_p(c_p - \lambda_{od}) = 0 \end{cases} \implies \begin{cases} c_p - \lambda_{od} \geq 0 \\ 0 \cdot (c_p - \lambda_{od}) = 0 \end{cases} \implies \{ c_p - \lambda_{od} \geq 0 \Rightarrow c_p \geq \lambda_{od}$$

Ce sont précisément les conditions d'équilibre où le multiplicateur de Lagrange λ_{od} représentant le temps minimum entre l'origine o et la destination d .

3.5.3 Exemple illustratif

Soit $G = (X, U, T)$ le réseau du trafic urbain suivant :



Réseau de trafic $G=(X,U,T)$.

Dans cet exemple :

- Nous allons formuler le problème d'affectation de trafic urbain PATU sous forme d'un modèle mathématique non linéaire (modèle de Beckmann).
- Nous allons chercher l'équilibre de Wardrop par l'utilisation des conditions de Kuhn et Tucker .

Les données :

L'offre :

- $X = \{1, 2, 3, 4, 5, 6\}$ $n = 6$
- $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ $m = 6$
- $OD = \{(16), (26)\}$ c'est-à-dire on a deux paire origine-destination.
- Les fonctions de performances des arcs sont donnée par :

$$t_1 = 1; t_2 = 2; t_3 = 1 + 2\varphi_3; t_4 = 2 + \varphi_4; t_5 = 3; t_6 = 2$$

Avec $\varphi_{u_j} = \varphi_j$ est le flux de l'arc u_j $j = \overline{1, m}$.

La demande :

La demande origine-destination est donnée par la matrice :

$$q_{od} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

C'est-à-dire : $O = \{1, 2\}$ et $D = \{6\}$ $q_{16} = 2$ et $q_{26} = 3$

a Formulation mathématique de problème de Beckmann :

Pour déterminer le modèle mathématique, Nous allons formuler les trois relations que nous avons étudié précédemment :

On a quatre chemins :

$$\begin{aligned} p_1 &= (u_1 - u_4 - u_6) \\ p_2 &= (u_1 - u_3 - u_5 - u_6) \\ p_3 &= (u_2 - u_4 - u_6) \\ p_4 &= (u_2 - u_3 - u_5 - u_6) \end{aligned}$$

1- Les couts de ces chemins donnée par :

D'après la relation (3.1) (voir la forme matricielle) les couts des chemins sont calculés par :

$$C = (1, 2, 1 + 2\varphi_3, 2 + \varphi_4, 3, 2) \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} = (5 + \varphi_4, 7 + 2\varphi_3, 6 + \varphi_4, 8 + 2\varphi_3)$$

Donc :

$$\begin{cases} c_1 = 5 + \varphi_4 \\ c_2 = 7 + 2\varphi_3 \\ c_3 = 6 + \varphi_4 \\ c_4 = 8 + 2\varphi_3 \end{cases} \dots\dots\dots(1)$$

2- La relation entre les flux des arcs $u \in U$ et les flux des chemins p_i tel que $i = \overline{1, 4}$ est donnée par :

D'après la relation (3.2) (voir la forme matricielle) les couts des chemins sont calculés par :

$$\varphi = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \end{pmatrix} = \begin{pmatrix} z_1 + z_2 \\ z_3 + z_4 \\ z_2 + z_4 \\ z_1 + z_3 \\ z_2 + z_4 \\ z_1 + z_2 + z_3 + z_4 \end{pmatrix}$$

Donc :

$$\begin{cases} \varphi_1 = z_1 + z_2 \\ \varphi_2 = z_3 + z_4 \\ \varphi_3 = z_2 + z_4 \\ \varphi_4 = z_1 + z_3 \\ \varphi_5 = z_2 + z_4 \rightarrow \varphi_5 = \varphi_3 \\ \varphi_6 = z_1 + z_2 + z_3 + z_4 = \varphi_1 + \varphi_2 = \varphi_3 + \varphi_4 \end{cases} \dots\dots\dots(2)$$

3- la conditin de Conservation du flux est donée par :

D'après la relation (3.3) (voir la forme matricielle) les taux de voyages entre *o* et *d* sont calculés par :

$$(z_1, z_2, z_3, z_4) \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{pmatrix} = (2, 3)$$

Donc :

$$\begin{cases} z_1 + z_2 = 2 \\ z_3 + z_4 = 3 \end{cases} \dots\dots\dots(3)$$

Le programme associer par la formulation de Beckmann et al :

$$(P) = \begin{cases} \text{Min}f(\varphi) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w)dw \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \\ \text{Min} f(\varphi) = \int_0^{\varphi_1} 1.dw + \int_0^{\varphi_2} 2.dw + \int_0^{\varphi_3} (1 + 2w_3).dw + \int_0^{\varphi_4} (2 + w_4).dw + \int_0^{\varphi_5} 3.dw + \int_0^{\varphi_6} 2.dw \\ \quad = \varphi_1 + 2\varphi_2 + \varphi_3^2 + \varphi_3 + \frac{1}{2}\varphi_4^2 + 2\varphi_4 + 3\varphi_5 + 2\varphi_6 \\ S.c \\ z_1 + z_2 = 2 \\ \quad \quad z_3 + z_4 = 3 \\ z_1 \geq 0 \\ \quad z_2 \geq 0 \\ \quad \quad z_3 \geq 0 \\ \quad \quad \quad z_4 \geq 0 \end{cases}$$

D'après la relation(2), on remplace les flux d'arcs $\varphi_i(i = \overline{1,6})$ par les valeurs des flux des chemin correspondants , on aura le programme équivalent "flux – chemin" :

$$(P) = \begin{cases} \text{Minf}(\varphi) = (z_1 + z_2) + 2(z_3 + z_4) + (z_2 + z_4)^2 + (z_2 + z_4) + \frac{1}{2}(z_1 + z_3)^2 + 2(z_1 + z_3) \\ \quad 3(z_2 + z_4) + 2(z_1 + z_2 + z_3 + z_4) \\ \\ S.c \\ z_1 + z_2 = 2 \\ \quad z_3 + z_4 = 3 \\ z_1 \geq 0 \\ \quad z_2 \geq 0 \\ \quad \quad z_3 \geq 0 \\ \quad \quad \quad z_4 \geq 0 \end{cases}$$

b L'équilibre de Wardrop par l'utilisation des conditions de Kuhn et Tucker :

Pour écrire les conditions de Kuhn et Tucker, écrivons le Lagrangien correspondant :

$$\begin{aligned} L(Z, \mu, \lambda) &= f(\varphi(z)) + \sum_{od} \lambda_{od}(q_{od} - \sum_p z_p \delta'_{p,od}) - \sum_p \mu_p z_p \\ &= (z_1 + z_2) + 2(z_3 + z_4) + (z_2 + z_4)^2 + (z_2 + z_4) + \frac{1}{2}(z_1 + z_3)^2 + 2(z_1 + z_3) + 3(z_2 + z_4) + \\ & 2(z_1 + z_2 + z_3 + z_4) + \lambda_{16}(2 - z_1 - z_2) + \lambda_{26}(3 - z_3 - z_4) - \mu_1 z_1 - \mu_2 z_2 - \mu_3 z_3 - \mu_4 z_4. \end{aligned}$$

Où :

- λ_{16} , λ_{26} et μ_i ($i = \overline{1,4}$) : sont les multiplicateurs de Lagrange.
- z_k ($k = \overline{1,4}$) est le flux sur le chemin p .

Les conditions d'optimalité de Kuhn et Tucker de premier ordre sont écrit comme suit :

$$(KKT) = \begin{cases} \frac{\partial L(\varphi(z), \lambda, \mu)}{\partial z_p} = 0 & \Leftrightarrow (4) \begin{cases} \frac{\partial L}{\partial z_1} = 5 + (z_1 + z_3) - \lambda_{16} - \mu_1 = 0 \\ \frac{\partial L}{\partial z_2} = 7 + 2(z_2 + z_4) - \lambda_{16} - \mu_2 = 0 \\ \frac{\partial L}{\partial z_3} = 6 + (z_1 + z_3) - \lambda_{26} - \mu_3 = 0 \\ \frac{\partial L}{\partial z_4} = 8 + 2(z_2 + z_4) - \lambda_{26} - \mu_4 = 0 \end{cases} \\ \\ \mu_p z_p = 0 & \Leftrightarrow (5) \begin{cases} \mu_1 z_1 = 0 \\ \mu_2 z_2 = 0 \\ \mu_3 z_3 = 0 \\ \mu_4 z_4 = 0 \end{cases} \\ \\ \mu_p \geq 0 & \Leftrightarrow (6) \begin{cases} \mu_1 \geq 0 \\ \mu_2 \geq 0 \\ \mu_3 \geq 0 \\ \mu_4 \geq 0 \end{cases} \end{cases}$$

Par remplacement de l'équation (2) dans l'équation (4) :

$$(4) \iff \begin{cases} 5 + \varphi_3 - \lambda_{16} - \mu_1 = 0 \\ 7 + 2\varphi_4 - \lambda_{16} - \mu_2 = 0 \\ 6 + \varphi_3 - \lambda_{26} - \mu_3 = 0 \\ 8 + 2\varphi_4 - \lambda_{26} - \mu_4 = 0 \end{cases} \iff \begin{cases} \mu_1 = 5 + \varphi_3 - \lambda_{16} \\ \mu_2 = 7 + 2\varphi_4 - \lambda_{16} \\ \mu_3 = 6 + \varphi_3 - \lambda_{26} \\ \mu_4 = 8 + 2\varphi_4 - \lambda_{26} \end{cases}$$

Ainsi a partir de relation (1) :

$$(4) \iff \begin{cases} \mu_1 = c_1 - \lambda_{16} \\ \mu_2 = c_2 - \lambda_{16} \\ \mu_3 = c_3 - \lambda_{26} \\ \mu_4 = c_4 - \lambda_{26} \end{cases} \dots\dots\dots(4')$$

Avec c_p : le cout de parcourt du chemin p entre (1 – 6) et (2 – 6).
On peut éliminer μ_p du système (c-à-d on remplace (4') dans (5) et (6)) pour obtenir :

$$(5) \iff \begin{cases} (c_1 - \lambda_{16})z_1 = 0 \\ (c_2 - \lambda_{16})z_2 = 0 \\ (c_3 - \lambda_{26})z_3 = 0 \\ (c_4 - \lambda_{26})z_4 = 0 \end{cases} \dots\dots\dots(5')$$

$$(6) \iff \begin{cases} c_1 \geq \lambda_{16} \\ c_2 \geq \lambda_{16} \\ c_3 \geq \lambda_{26} \\ c_4 \geq \lambda_{26} \end{cases} \dots\dots\dots(6')$$

Les conditions (5') impliquent que : Si $z_p > 0$ "il existe un flux sur le chemin p " alors :

$$\begin{cases} c_1 = \lambda_{16} \\ c_2 = \lambda_{16} \\ c_3 = \lambda_{26} \\ c_4 = \lambda_{26} \end{cases}$$

On remplace les couts des chemins $c_i(i = \overline{1,4})$ par leurs valeurs d'après la relation (1) :

$$\iff \begin{cases} 5 + \varphi_4 = \lambda_{16} \\ 7 + 2\varphi_3 = \lambda_{16} \\ 6 + \varphi_4 = \lambda_{26} \\ 8 + 2\varphi_3 = \lambda_{26} \end{cases} \dots\dots\dots(\star)$$

Par remplacement (3) dans (2) on obtient :

$$\begin{cases} \varphi_1 = 2 \\ \varphi_2 = 3 \\ \varphi_6 = \varphi_1 + \varphi_2 = \varphi_3 + \varphi_4 = 5 \end{cases}$$

D'après la condition de conservation des flux au niveau du sommet (3) nous avons :

$$\varphi_1 + \varphi_2 = \varphi_3 + \varphi_4 = 5 \implies \varphi_4 = 5 - \varphi_3 \dots \dots \dots (\star')$$

Et nous avons :

$$(\star) \implies 2\varphi_3 + 2 = \varphi_4 \dots \dots \dots (\star'')$$

Par remplacement (\star'') dans (\star') nous obtenons :

$$\varphi_3 = \varphi_5 = 1$$

$$\varphi_4 = 5 - \varphi_3 = 4$$

Donc les flux des arcs qui donnent l'équilibre sont :

$$\varphi = \begin{cases} \varphi_1 = 2 \\ \varphi_2 = 3 \\ \varphi_3 = 1 \\ \varphi_4 = 4 \\ \varphi_5 = 1 \\ \varphi_6 = 5 \end{cases}$$

La fonction objectif :

$$f(\varphi) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w) dw = \varphi_1 + 2\varphi_2 + \varphi_3^2 + \varphi_3 + \frac{1}{2}\varphi_4^2 + 2\varphi_4 + 3\varphi_5 + 2\varphi_6 = 2 + 2 \times 3 + 1 + 1 + \frac{1}{2} \times 4^2 + 2 \times 4 + 3 + 2 \times 5 = 39$$

Illustration des solutions à travers un réseau :

le temps de parcourt des arcs est donné par :

$$T = \begin{cases} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2\varphi_3 = 3 \\ t_4 = 2 + \varphi_4 = 6 \\ t_5 = 3 \\ t_6 = 2 \end{cases}$$

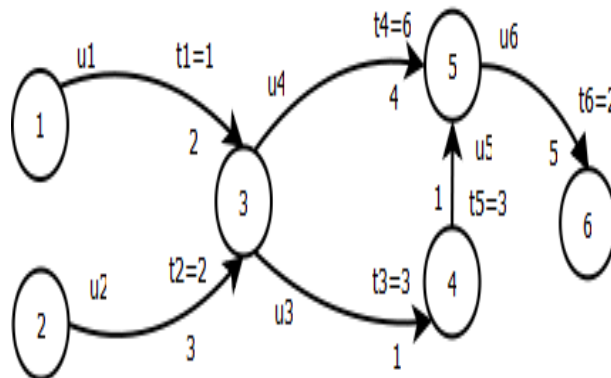


Illustration graphique de solution optimale.

Vérification du premier principe de Wardrop :

À l'équilibre les temps de traversées de tous les chemins reliant l'origine o avec la destination d utilisés sont égaux et inférieur aux coûts des chemins non utilisés.

1. Pour $od = (16)$:

$$\begin{cases} c_{p1} = t(\varphi_1) + t(\varphi_4) + t(\varphi_6) = 2 + 4 + 5 = 10 \\ c_{p2} = t(\varphi_1) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 2 + 3 + 3 + 2 = 10 \end{cases} \implies c_{p1} = c_{p2}$$

2. Pour $od = (26)$:

$$\begin{cases} c_{p3} = t(\varphi_2) + t(\varphi_4) + t(\varphi_6) = 3 + 4 + 2 = 9 \\ c_{p4} = t(\varphi_2) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 3 + 1 + 3 + 2 = 9 \end{cases} \implies c_{p3} = c_{p4}$$

Conclusion

L'application de la méthode analytique sur un réseau de 5 sommets, considéré comme un petit réseau en le comparant à un réseau réel (plus de 500 sommets) ceci a nécessité l'utilisation de beaucoup de variables et de ressources pour la résolution du problème du réseau de l'exemple; pour cela le calcul analytique est considéré trop complexe et fastidieux par rapport au calculs numériques que nous allons détailler dans le chapitre suivant.

4

Méthode de résolution de PATU

Introduction

Résoudre analytiquement un modèle mathématique est parait trop difficile et parfois même impossible. La plus part des partisans et spécialistes dans l'optimisation font appel aux algorithmes et méthodes numériques exacte ou approché pour la résolution. Le PATU est un programme convexe, il est composé d'une fonction non linéaire convexe qui est croissante et de C^2 (classe 2), et des contraintes linéaires. Sa résolution repose sur les méthodes de directions faisables.

Leurs adaptations au PATU se traduisent en la recherche de la meilleure distribution des voyageurs dans le réseau, de sorte que chaque voyageur soit dans le chemin optimal menant vers son destination.

La résolution par la méthode analytique (la méthode de KKT) obtenu par le chapitre précédent est considérée trop complexe et difficile à atteindre (impossible) dans les réseaux réelles, pour cela nous allons voir dans ce chapitre une méthode parmi les méthode de résolution de direction admissible la plus qualifie, la plus simple et la plus facile par la mise en œuvre, puis nous appliquons cette méthode sur un petit exemple (même exemple du chapitre précédent afin de comparer les résultats analytiques et numériques).

4.1 Méthode des directions admissibles

Comme nous avons vu au chapitre précédent, le problème d'équilibre de l'utilisateur (PATU) pouvait être formulé comme étant un problème d'optimisation non linéaire convexe .

$$(P) = \begin{cases} \text{Min}f(\varphi) = \sum_{u \in U} \int_0^{\varphi_u} t_u(w)dw \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \dots \dots \dots (1) \\ z_p \geq 0 \quad \forall p \dots \dots \dots (2) \end{cases}$$

tel que :

φ_u :flux sur l'arc u .

$t_u(\varphi_u)$:le temps de parcours de l'arc u .

z_p :le flux sur le chemin p .

$\delta'_{p,od}$:la matrice d'incidence *chemins/paires* .

q_{od} :la matrice de demande entre l'origine o et la destination d .

Le flux sur l'arc u et le flux sur le chemin p sont reliés par l'équation suivante :

$$\varphi_u = \sum_p z_p \delta_{u,p} \quad \forall u \in U \quad (4.1)$$

L'ensemble des solutions réalisables est donnée par :

$$S = \{ \varphi_u \geq 0 \text{ tel que } \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \text{ et } z_p \geq 0 \quad \forall p \text{ sont vérifiés} \}$$

La résolution de ce problème PATU se fait par différentes méthodes d'optimisation, parmi ces méthodes on cite celles qui se repose sur la direction admissible.

Principe des méthodes de direction admissible :

Cette classe de méthodes résout un problème de minimisation non linéaire en se déplaçant d'un point de S vers un autre point dans S à un coût inférieur. Elles fonctionnent selon le principe suivant :

Etant donné un élément φ^k de S , une direction d^k est générée telle que : pour un $\alpha_k > 0$ et suffisamment petit, les propriétés suivantes sont assurées :

1. $\varphi^k + \alpha_k d^k \in S$.
2. $f(\varphi^k + \alpha_k d^k) < f(\varphi^k)$.

Une fois d^k déterminée, α_k s'obtient par minimisation unidimensionnelle pour que le déplacement dans la direction d^k soit optimal, mais cette fois-ci il est nécessaire d'imposer une borne supérieure sur la valeur de α_k afin de ne pas sortir de S . Cela définit le nouveau point $\varphi^k + 1$, et le processus est recommencé.

Il existe plusieurs stratégies pour la résolution du sous-problème consistant à déterminer d^k . Comme nous allons le voir, il peut être exprimé sous forme d'un programme linéaire. Il existe plusieurs méthodes de directions admissible parmi ces méthodes :

- la méthode de Franck-Wolfe.
- La méthodes de gradient projeté.
- La méthodes de newton et de quasi-newton.

Nous nous intéressons à la méthode de Frank et Wolfe.

4.1.1 La méthode de Franck-Wolfe

Le principe de la méthode repose sur deux étapes obligatoires :

- Le choix d'une direction pour la fonction objectif ;
- Le choix du pas le long de cette direction de descente.

1. Le choix de la direction :

Le choix de la direction est déterminé par

$$d^k = y^k - \varphi^k$$

Nous avons y^k minimise l'approximation linéaire de $f(\varphi)$ en φ^k sous les contraintes linéaires. Ensuite une recherche unidimensionnelle sera effectuée le long de la direction d^k .

On a donc y^k est une solution du problème linéaire suivant :

$$(P) = \begin{cases} \text{Min}_y f(\varphi^k) + \nabla f(\varphi^k)(y - \varphi^k) \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \end{cases}$$

Avec les deux relations suivantes :

Le cout de parcourt d'un chemin p est donnée par :

$$c_p = \sum_{u \in U} t_u \delta_{u,p} \quad (4.2)$$

La relation entre les flux des arcs $u \in U$ et les flux des chemins p est donnée par :

$$y_u = \sum_p z_p \delta_{u,p} \quad \forall u \in U \quad (4.3)$$

Où : $\delta_{u,p}$ est la matrice d'incidence *arc/chemin* indiquant à quelle chemin p un arc u appartient.

Dans cette section nous allons montrer que la résolution de ce problème linéaire est équivalente a la résolution de problème de plus court chemin.

Justification :

$$(P) \iff \begin{cases} \text{Min} f(\varphi^k) + \sum_u \frac{\partial f(\varphi^k)}{\partial \varphi_u} (y_u - \varphi_u) \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \end{cases}$$

Après élimination des constantes, (P) devient :

$$(P) \iff \begin{cases} \text{Min} \frac{\partial f(\varphi^k)}{\partial \varphi_u} y_u = \sum_u t_u^k y_u \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \end{cases}$$

Par remplacement de y_u par la relation (4.2) et d'après (4.3), on obtient :

$$(P) \iff \begin{cases} \text{Min} \sum_u \sum_p t_u^k z_p \delta_{p,od} = \sum_p \sum_u t_u^k \delta_{p,od} z_p = \sum_p c_p z_p \\ \sum_p z_p \delta'_{p,od} = q_{od} \quad \forall od \\ z_p \geq 0 \quad \forall p \end{cases}$$

Donc ce problème peut être décomposé en un problème de plus court chemin avec chargement pour chaque paire origine-destination od . Dont la solution optimale est obtenue en affectant tout q_{od} au chemin p (affectation tout ou rien) dont le coût c_p est minimum .

Remarque :

Dans le cas où plusieurs chemins sont minimaux on choisi n'importe lequel .

$$(P) \iff \begin{cases} \text{Min} \sum_{p \in OD} c_p z_p \\ \sum_{p \in OD} z_p = q_{od} \\ z_p \geq 0 \quad \forall p \in OD \end{cases}$$

Une fois les z_p déterminés, la direction peut être calculée comme suit :

$$y_u^k = \sum_p z_p \delta_{u,p} \quad \forall u \in U \quad d^k = y^k - \varphi^k \quad (4.4)$$

Pour initialiser l'algorithme, on fera une affectation sur base des temps de traversées à vide :

$$t_u^0 = t_u(0) \quad \forall u \in U \quad (4.5)$$

2. Recherche unidimensionnelle :

Après la détermination de la direction y , reste maintenant la seconde partie de la méthode de Franck-Wolfe, à savoir la recherche unidimensionnelle. Nous allons déterminer le pas α_k dans la direction d^k qui minimise :

$$f(\varphi^k + \alpha(y^k - \varphi^k)) \quad (4.6)$$

En limitant α à l'intervalle $[0, 1]$ de manière à ne pas sortir de la région réalisable. Afin de déterminer ce pas α nous avons abordé l'une des méthodes de recherche unidimensionnelle existantes, c'est la méthode de la section dorée.

Avant d'appliquer la méthode de la section dorée il faut vérifier l'existence d'une racine, puis vérifier qu'elle est minimum.

L'existence d'une racine :

- L'existence d'une racine sera déterminée par la vérification de la condition suivante :
- Soit $f(\alpha)$ une fonction objective à minimiser sur l'intervalle $[a; b]$ et Soit $f' : [a, b] \rightarrow \mathbb{R}$ telle que f' est continue sur le segment $[a, b]$;
 - Si $f'(a) * f'(b) \leq 0$ alors f' possède une racine α dans $[a, b]$ ($f'(\alpha) = 0$) et si $f''(\alpha) > 0$ alors α est un minimum .

Principe de La méthode de la section dorée :

Cette méthode tirant son nom de la valeur du paramètre m (nombre d'or) .
Soit $f(\varphi)$ la fonction coût (objectif) que l'on cherche à minimiser sur l'intervalle $[a; b]$. Afin de trouver ce minimum.

On construit une suite décroissante d'intervalles $[a, b]$ qui contiennent tous le minimum α .

Pour passer de $[a, b]$ au nouvel intervalle $[a, b]$, on procède de la manière suivante.
On introduit deux nombres a et b de l'intervalle initiale $[a, b]$ tels que $a < b$.
On calcule les valeurs $(f'(a) * f'(b) < 0)$ pour vérifier si l'intervalle $[a, b]$ contenant le minimum α de f . Puis, on calcule les valeurs $f(c)$ et $f(d)$, deux possibilités se présentent alors à nous.

Si $(f(c) < f(d))$, alors, α se trouve nécessairement à gauche de b Ceci définit alors le nouvel intervalle en posant $a = a$ et $b = d$.

Considérons maintenant que l'inégalité : $(f(c) > f(d))$ est satisfaite.

Dans ce second cas, il est évident que le minimum se trouve cette fois à droite de a . On pose alors : $a = c$ et $b = b$.

A cette fin, on doit valider un critère d'arrêt de la forme : $(b - a) > 2 * tolerance$, où $tolerance$ est l'erreur que l'on se permet sur la solution α du problème.

L'algorithme de la section dorée[14] :

On utilise l'inverse $dore = \frac{\sqrt{5}-1}{2} \simeq 0.618$ du nombre d'or : $\frac{\sqrt{5}+1}{2}$

Algorithme 3 L'algorithme de section doree

Entrées: a,b,tolerance,f;

Sorties: alpha;

```

1: Calculer  $f'$ 
2: Calculer  $f''$ 
3: si  $f'(a) * f'(b) < 0$  alors
4:    $dore \leftarrow \frac{\sqrt{5}-1}{2}$ ;
5:    $c \leftarrow dore * a + (1 - dore) * b$ ;
6:    $d \leftarrow a + b - c$ ;
7:   tantque  $(b - a) > 2 * tolerance$  faire
8:     si  $f(c) < f(d)$ ; alors
9:        $b \leftarrow d$ ;
10:       $d \leftarrow c$ ;
11:       $c \leftarrow a + b - d$ ;
12:     sinon
13:        $a \leftarrow c$ ;
14:        $c \leftarrow d$ ;
15:        $d = a + b - c$ ;
16:     finsi
17:   fin tantque
18: finsi
19:  $\alpha \leftarrow (a + b)/2$ ;
20: si  $f'' > 0$  alors
21:    $\alpha$  est un minimum globale
22: finsi

```

3.Critère d'arrêt de l'algorithme de Franck-Wolfe :

Le critère d'arrêt de l'algorithme itératif sera la proximité entre deux solutions successives :

$$\frac{\sqrt{\sum_u (\varphi_u^{k+1} - \varphi_u^k)^2}}{\sum_u \varphi_u^k} \leq \epsilon \quad (4.7)$$

L'algorithme peut être résumé comme suit.

L'algorithme [3] :

Algorithme 4 L'algorithme de calcul d'un Equilibre de l'Utilisateur

- 1: **Pas0. Initialisations.** Chargement tout ou rien du réseau sur base des plus courts chemins calculés avec les temps :

$$t_u^0 = t_u(0), \forall u$$

Ceci fournit φ_u^1 . Soit $k = 1$.

- 2: **Pas 1. Détermination de la direction.** Chargement tout ou rien sur base des plus courts chemins calculés avec les t_u^k . Ceci fournit les flux y_u^k
- 3: **Pas3. Recherche unidimensionnelle.** Déterminer α_k qui minimise :

$$\min_{0 \leq \alpha \leq 1} \sum_u \int_0^{\varphi_u^k + \alpha(y_u^k - \varphi_u^k)} t_u(w) dw$$

- 4: **Pas4. Effectuer le pas.** Poser

$$\varphi_u^{k+1} = \varphi_u^k + \alpha(y_u^k - \varphi_u^k)$$

- 5: **Pas5. Test de convergence.**

6: **si** $\frac{\sqrt{\sum_u (\varphi_u^{k+1} - \varphi_u^k)^2}}{\sum_u \varphi_u^k} \leq \epsilon$ **alors**

7: stop.

8: **sinon**

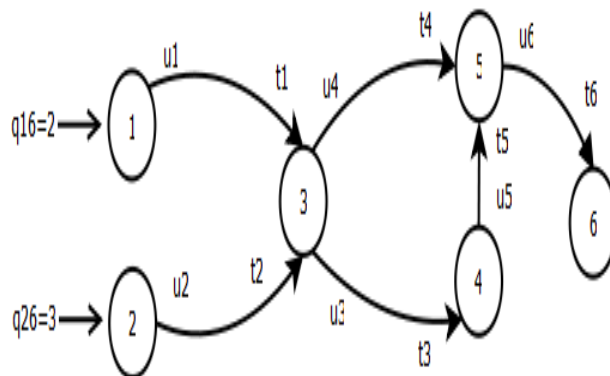
9: poser $k \leftarrow k + 1$

10: et retour au pas 1.

11: **fini**

4.1.2 Exemple illustratif

Soit $G = (X, U, T)$ le réseau de trafic urbain suivant :



Réseau du trafic $G = (X, U, T)$.

Remarque : On garde toujours le même exemple précédent afin de pouvoir comparer les résultats analytiques et numériques.

Dans cet exemple nous allons appliquer l'algorithme de Franck-Wolfe.

Les données :

1-L'offre :

- $X = \{1, 2, 3, 4, 5, 6\}$ $n = 6$
- $U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ $m = 6$
- $OD = \{(16), (26)\}$ c'est-à-dire nous avons deux paires origine-destination.
- Les temps des parcours des arcs sont données par :

$$\begin{cases} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2\varphi_3 \\ t_4 = 2 + \varphi_4 \\ t_5 = 3 \\ t_6 = 2 \end{cases} \dots\dots\dots(1)$$

Avec $\varphi_{u_j} = \varphi_j$ est le flux de l'arc u_j $j = \overline{1, m}$.

2-La demande :

La demande origine-destination est donnée par le vecteur :

$$q_{od} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

C'est-à-dire : $O = \{1, 2\}$, $D = \{6\}$ et $q_{16} = 2$, $q_{26} = 3$

Nous avons quatre chemins :

– Pour le couple $od = (16)$ nous avons deux chemins :

$$\begin{cases} p_1 = (u_1 - u_4 - u_6) \\ p_2 = (u_1 - u_3 - u_5 - u_6) \end{cases}$$

– Pour le couple $od = (26)$ nous avons deux chemins :

$$\begin{cases} p_3 = (u_2 - u_4 - u_6) \\ p_4 = (u_2 - u_3 - u_5 - u_6) \end{cases}$$

On cherche la fonction objective :

$$\begin{aligned} f(\varphi) &= \sum_{u \in U} \int_0^{\varphi_u} t_u(w) dw \\ &= \int_0^{\varphi_1} t(w_1).dw_1 + \int_0^{\varphi_2} t(w_2).dw_2 + \int_0^{\varphi_3} t(w_3).dw_3 + \int_0^{\varphi_4} t(w_4).dw_4 + \int_0^{\varphi_5} t(w_5).dw_5 + \\ &\quad \int_0^{\varphi_6} t(w_6).dw_6 \\ &= \int_0^{\varphi_1} 1.dw_1 + \int_0^{\varphi_2} 2.dw_2 + \int_0^{\varphi_3} (1 + 2w_3).dw_3 + \int_0^{\varphi_4} (2 + w_4).dw_4 + \int_0^{\varphi_5} 3.dw_5 + \int_0^{\varphi_6} 2.dw_6 \\ &= [w_1]_0^{\varphi_1} + [2w_2]_0^{\varphi_2} + [w_3]_0^{\varphi_3} + [w_3^2]_0^{\varphi_3} + [2w_4]_0^{\varphi_4} + [\frac{1}{2}w_4^2]_0^{\varphi_4} + [3w_5]_0^{\varphi_5} + [2w_6]_0^{\varphi_6} \\ &= \varphi_1 + 2\varphi_2 + \varphi_3 + \varphi_3^2 + 2\varphi_4 + \frac{1}{2}\varphi_4^2 + 3\varphi_5 + 2\varphi_6 \\ &= \varphi_3^2 + \frac{1}{2}\varphi_4^2 + \varphi_1 + 2\varphi_2 + \varphi_3 + 2\varphi_4 + 3\varphi_5 + 2\varphi_6 \dots \dots \dots (\star) \end{aligned}$$

Déroulement de l'algorithme de Franck-Wolfe :

Itération 1 :

Pas 0. Initialisation : ($k = 0$ La solution de départ φ^0)

$$\varphi^0 = (\varphi_1^0, \varphi_2^0, \varphi_3^0, \varphi_4^0, \varphi_5^0, \varphi_6^0) = (0, 0, 0, 0, 0, 0)$$

Par remplacement de valeur φ^0 dans (1), nous obtenons :

$$\begin{cases} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2 * 0 = 1 \\ t_4 = 2 + 0 = 2 \\ t_5 = 3 \\ t_6 = 2 \end{cases} \implies t(\varphi^0) = (1, 2, 1, 2, 3, 2)$$

La valeur de la fonction objectif pour φ^0 est :

$$f(\varphi^0) = f(0, 0, 0, 0, 0) = 0^2 + \frac{1}{2} \times 0^2 + 0 + 2 \times 0 + 0 + 2 \times 0 + 3 \times 0 + 2 \times 0 = 0.$$

a Cherchons le plus court chemin par l'algorithme de Dijkstra :

On applique l'algorithme de Dijkstra-moore pour la première couple origine-destination $od = (1, 6)$.

Déroulement

Nous avons le graphe suivant :

Nous avons : $X = \{1, 3, 4, 5, 6\}$

$U = \{u_1, u_2, u_3, u_4, u_5, u_6\}$

$T = \{1, 1, 2, 3, 2\}$

$s = 1$ (la racine)

d : tableau de marquage de dimension 5 (nombre des sommets existe).

$d[1] = \infty;$

$d[3] = \infty;$

$d[4] = \infty;$

$d[5] = \infty;$

$d[6] = \infty;$

$A = X = \{1, 3, 4, 5\}$

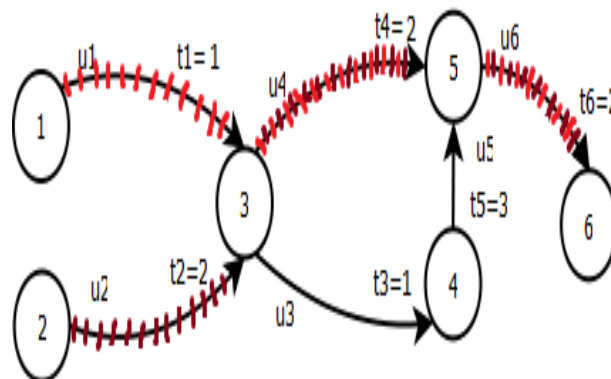
$V = \emptyset$

$d[1] = 0;$

A chaque étape on met dans une liste des sommets traités les sommets dont la distance a été correctement calculée. Les sommets arrivent dans cette liste par distance croissante.

Pour les sommets x de la liste A , le tableau d indique le plus court chemin de la source s à x , dont tous les sommets sauf x sont déjà traités.

A	T	d(1)	d(3)	d(4)	d(5)	d(6)
1,3,4,5,6	\emptyset	0	∞	∞	∞	∞
3,4,5,6	1	0	1	∞	∞	∞
4,5,6	1,3	0	1	2	3	∞
5,6	1,3,4	0	1	2	3	∞
6	1,3,4,5	0	1	2	3	5
\emptyset	1,3,4,5,6	0	1	2	3	5



Donc :

Pour le couple $od = (16)$ le plus court chemin est le chemin $p_1 = (u_1 - u_4 - u_6)$

Pour le couple $od = (26)$ le plus court chemin est le chemin $p_3 = (u_2 - u_4 - u_6)$

b Chargement tout ou rien de plus court chemin trouvé :

$$\left\{ \begin{array}{l} \varphi_1^1 \leftarrow q_{16} = 2 \\ \varphi_2^1 \leftarrow q_{26} = 3 \\ \varphi_3^1 \leftarrow 0 \\ \varphi_4^1 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \\ \varphi_5^1 \leftarrow 0 \\ \varphi_6^1 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \end{array} \right. \implies \varphi^1 = (2, 3, 0, 5, 0, 5)$$

La valeur de la fonction objectif pour φ^1 est :

$$f(\varphi^1) = f(2, 3, 0, 5, 0, 5) = 0^2 + \frac{1}{2} \times 5^2 + 2 + 2 \times 3 + 0 + 2 \times 5 + 3 \times 0 + 2 \times 5 = 12.5 + 28 = 40.5.$$

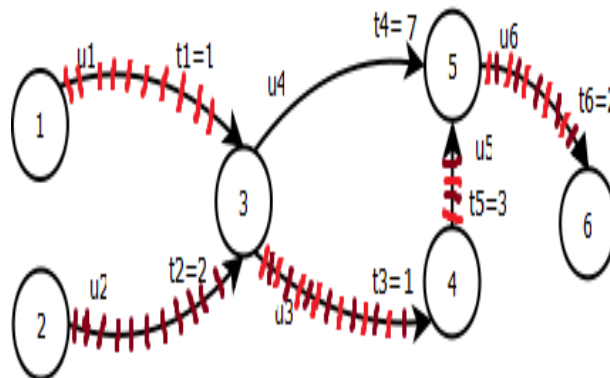
Pas 1. Détermination de la direction :

Par remplacement la solution φ^1 dans (1), on obtient :

$$\left\{ \begin{array}{l} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2 \times 0 = 1 \\ t_4 = 2 + 5 = 7 \\ t_5 = 3 \\ t_6 = 2 \end{array} \right.$$

Donc : $t(\varphi^1) = (1, 2, 1, 7, 3, 2)$

a Cherchons le plus court chemin par l'algorithme de Dijkstra :



Donc :

Pour le couple $od = (16)$ le plus court chemin est le chemin $p_2 = (u_1 - u_3 - u_5 - u_6)$

Pour le couple $od = (26)$ le plus court chemin est le chemin $p_4 = (u_2 - u_3 - u_5 - u_6)$

b Chargement tout ou rien de plus court chemin trouvé :

$$\left\{ \begin{array}{l} y_1^1 \leftarrow q_{16} = 2 \\ y_2^1 \leftarrow q_{26} = 3 \\ y_3^1 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \\ y_4^1 \leftarrow 0 \\ y_5^1 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \\ y_6^1 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \end{array} \right. \implies y^1 = (2, 3, 5, 0, 5, 5)$$

La direction :

$$d = y_u^k - \varphi_u^k = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 0 \\ 5 \\ 5 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \\ 0 \\ 5 \\ 0 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 5 \\ -5 \\ 5 \\ 0 \end{pmatrix}$$

Pas 2. Recherche unidimensionnelle :

Comment trouver α qui minimise $f(x_u^k + \alpha d)$?

1. Calculer $f(x_u^k + \alpha d)$.
2. Calculer le minimum par la méthode de la section dorée.

Calcule de $f(x_u^k + \alpha d)$:

$$\varphi_u^{k+1} = \varphi_u^2 = \varphi_u^1 + \alpha d = \begin{pmatrix} 2 \\ 3 \\ 0 \\ 5 \\ 0 \\ 5 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 0 \\ 5 \\ -5 \\ 5 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 5\alpha \\ 5 - 5\alpha \\ 5\alpha \\ 5 \end{pmatrix}$$

On a :

$$f(\varphi) = \varphi_3^2 + \frac{1}{2}\varphi_4^2 + \varphi_1 + 2\varphi_2 + \varphi_3 + 2\varphi_4 + 3\varphi_5 + 2\varphi_6$$

$$f(\varphi_u^2) = f(\alpha) = f(2, 3, 5\alpha, 5 - 5\alpha, 5\alpha, 5)$$

$$= (5\alpha)^2 + \frac{1}{2} + (5(1 - \alpha))^2 + 2 + 2 \times 3 + 5\alpha + 10(1 - \alpha) + 3 \times 5\alpha + 2 \times 5$$

$$\begin{aligned}
&= (5\alpha)^2 + \frac{1}{2} + (5(1-\alpha))^2 + 2 + 6 + 5\alpha + 10(1-\alpha) + 15\alpha + 10 \\
&= 25\alpha^2 + 12.5(1-\alpha)^2 + 8 + 5\alpha + 10 - 10\alpha + 15\alpha + 10 \\
&= 37.5\alpha^2 - 15\alpha + 40.5
\end{aligned}$$

Trouver le minimum par la méthode de la section dorée :

$$\alpha \in [a, b] = [0, 1], \text{ tolerance} = 10^{-2}$$

$$f(\alpha) = 37.5\alpha^2 - 30\alpha + 43$$

$$m = \frac{\sqrt{5}-1}{2} \simeq 0.618$$

$$\left\{ \begin{array}{l} a = 0 \\ b = 1 \\ c = \alpha \times a + (1-\alpha)b = 0.618 \times 0 + (1-0.618) \times 1 = 0.382 \\ d = a + b - c = 0 + 1 - 0.382 = 0.618 \end{array} \right.$$

Itération 01 : $b - a = 1 > 0.02$

$$\left\{ \begin{array}{l} f(c = 0.382) = 37.0121 \\ f(d = 0.618) = 38.7822 \end{array} \right. \implies f(d) > f(c) \longrightarrow \text{élimination de } b$$

$b = 0.6180$
 $d = 0.3820$
 $c = 0 + 0.618 - 0.382 = 0.2360$
 L'intervale devient : $[0, 0.6180]$

Itération 02 : $b - a = 0.6180 - 0 = 0.6180 > 0.02$

$$\begin{cases} f(c = 0.236) = 38.0086 \\ f(d = 0.382) = 37.0121 \end{cases} \implies f(c) > f(d) \longrightarrow \text{élimination de } a$$

$a = 0.2360$
 $c = 0.3820$
 $d = 0.2360 + 0.6180 - 0.3820 = 0.4720$
 L'intervale devient : $[0.236, 0.618]$

Itération 03 : $b - a = 0.618 - 0.236 = 0.3820 > 0.02$

$$\begin{cases} f(c = 0.382) = 37.0121 \\ f(d = 0.472) = 37.1944 \end{cases} \implies f(d) > f(c) \longrightarrow \text{élimination de } b$$

$b = 0.4720$
 $d = 0.3820$
 $c = 0.2360 + 0.4720 - 0.3820 = 0.3260$
 L'intervale devient : $[0.236, 0.472]$

Itération 04 : $b - a = 0.4720 - 0.2360 = 0.2360 > 0.02$

$$\begin{cases} f(c = 0.3260) = 37.2053 \\ f(d = 0.3820) = 37.0121 \end{cases} \implies f(c) > f(d) \longrightarrow \text{élimination de } a$$

$a = 0.3260$
 $c = 0.3820$
 $d = 0.3260 + 0.0472 - 0.3820 = 0.4160$
 L'intervale devient : $[0.3260, 0.4720]$

Itération 05 $b - a = 0.4720 - 0.3260 = 0.1460 > 0.02$

$$\begin{cases} f(c = 0.3260) = 37.0121 \\ f(d = 0.4160) = 37.0096 \end{cases} \implies f(c) > f(d) \longrightarrow \text{élimination de } a$$

$a = 0.3820$
 $c = 0.4160$
 $d = 0.472 + 0.3820 - 0.4160 = 0.4380$
 L'intervale devient : $[0.3820, 0.4720]$

Itération 06 : $b - a = 0.3820 - 0.4720 = 0.0900 > 0.02$

$$\begin{cases} f(c = 0.4160) = 37.0096 \\ f(d = 0.4380) = 37.0542 \end{cases} \implies f(d) > f(c) \longrightarrow \text{élimination de } b$$

$b = 0.4380$
 $d = 0.4160$
 $c = 0.3820 + 0.4380 - 0.4160 = 0.4040$
 L'intervale devient : $[0.3820, 0.4380]$

Itération 07 : $b - a = 0.4380 - 0.3820 = 0.0560 > 0.02$

$$\begin{cases} f(c = 0.4040) = 37.0006 \\ f(d = 0.4380) = 37.0096 \end{cases} \implies f(d) > f(c) \longrightarrow \text{élimination de } b$$

$b = 0.4160$
 $d = 0.4040$
 $c = 0.3820 + 0.4160 - 0.4040 = 0.3940$
 L'intervale devient : $[0.3820, 0.4160]$

Itération 08 : $b - a = 0.4160 - 0.3820 = 0.0340 > 0.02$

$$\begin{cases} f(c = 0.3940) = 37.0014 \\ f(d = 0.4040) = 37.0006 \end{cases} \implies f(c) > f(d) \longrightarrow \text{élimination de } a$$

$a = 0.3940$
 $c = 0.4040$
 $d = 0.3940 + 0.4160 - 0.4040 = 0.4060$
 L'intervale devient : $[0.3940, 0.4160]$

Itération 9 : $b - a = 0.4160 - 0.3940 = 0.0220 < 0.02$

$$\begin{cases} f(c = 0.4040) = 37.0006 \\ f(d = 0.4380) = 37.0014 \end{cases} \implies f(d) > f(c) \longrightarrow \text{élimination de } b$$

$b = 0.4060$
 $d = 0.4040$
 $c = 0.3940 + 0.4060 - 0.4040 = 0.3960$
 L'intervale devient : $[0.3820, 0.4160]$

Itération 10 : $b - a = 0.4160 - 0.3820 = 0.0120 < 0.02$

On s'arrête l'algorithme de section dorée avec : $\alpha = \frac{a+b}{2} = \frac{0.3820+0.4160}{2} = 0.1980$.

Pas 4. Effectuer le pas :

$$\varphi_u^{k+1} = \varphi_u^k + \alpha(y_u^k - \varphi_u^k)$$

$$\varphi_u^2 = \begin{pmatrix} 2 \\ 3 \\ 5\alpha \\ 5 - 5\alpha \\ 5\alpha \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 5 \times 0.1980 \\ 5 - 5 \times 0.1980 \\ 5 \times 0.1980 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0.99 \\ 4.01 \\ 0.99 \\ 5 \end{pmatrix}$$

La valeur de la fonction objectif pour φ^2 est :

$$\begin{aligned} f(\varphi_u^2) &= f(2, 3, 0.99, 4.01, 0.99, 5) \\ &= 0.99^2 + \frac{1}{2}(4.01)^2 + 2 + 2 \times 3 + 0.99 + 2 \times 4.01 + 3 \times 0.99 + 2 \times 5 \\ &= 0.9802 + 8.0401 + 2 + 6 + 0.99 + 8.02 + 2.97 + 10 = 39.0003 \end{aligned}$$

Pas 5. Test de convergence :

$$\begin{aligned} t &= \frac{\sqrt{\sum_u (\varphi_u^2 - \varphi_u^1)^2}}{\sum_u \varphi_u^1} \\ &= \frac{\sqrt{(2-2)^2 + (3-3)^2 + (0.99-0)^2 + (4.01-5)^2 + (0.99-0)^2 + (5-5)^2}}{2+3+0+5+0+5} \\ &= \frac{\sqrt{2.9403}}{15} = 0.19602 > 0.02 \end{aligned}$$

Donc $k=k+1=2$

Itération 2 de l'algorithme de Franck-Wolfe :

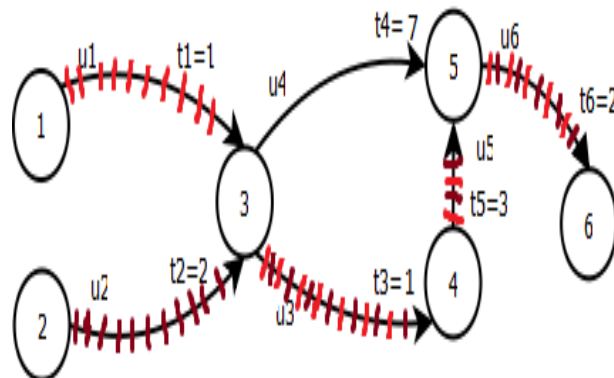
Pas 1. Détermination de la direction :

Par remplacement de φ^2 dans (1), on obtient :

$$\left\{ \begin{array}{l} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2 \times 0.99 = 2.98 \\ t_4 = 2 + 4.01 = 6.01 \\ t_5 = 3 \\ t_6 = 2 \end{array} \right.$$

Donc : $t(\varphi^2) = (1, 2, 2.98, 6.01, 3, 2)$.

a Cherchons le plus court chemin par l'algorithme de Dijkstra :



Donc :

Pour le couple $od = \{16\}$ le plus court chemin est le chemin $p_2 = (u_1 - u_3 - u_5 - u_6)$.

Pour le couple $od = \{26\}$ le plus court chemin est le chemin $p_4 = (u_2 - u_3 - u_5 - u_6)$.

b Chargement tout ou rien de plus court chemin trouvé :

$$\begin{cases} y_1^2 \leftarrow q_{16} = 2 \\ y_2^2 \leftarrow q_{26} = 3 \\ y_3^2 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \\ y_4^2 \leftarrow 0 \\ y_5^2 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \\ y_6^2 \leftarrow q_{16} + q_{26} = 2 + 3 = 5 \end{cases}$$

La direction :

$$d = y_u^k - \varphi_u^k = \begin{pmatrix} 2 \\ 3 \\ 5 \\ 0 \\ 5 \\ 5 \end{pmatrix} - \begin{pmatrix} 2 \\ 3 \\ 0.99 \\ 4.01 \\ 0.99 \\ 5 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 4.01 \\ -4.01 \\ 4.01 \\ 0 \end{pmatrix}$$

Pas 2. Recherche unidimensionnelle :

Comment trouver α qui minimise $f(x_u^k + \alpha d)$?

1. Calculer $f(x_u^k + \alpha d)$.
2. Calculer le minimum par la méthode de section dorée.

Calcul de $f(x_u^k + \alpha d)$:

$$\varphi_u^k + \alpha d = \begin{pmatrix} 2 \\ 3 \\ 0.99 \\ 4.01 \\ 0.99 \\ 5 \end{pmatrix} + \alpha \begin{pmatrix} 0 \\ 0 \\ 4.01 \\ -4.01 \\ 4.01 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0.99 + 4.01\alpha \\ 4.01(1 - \alpha) \\ 0.99 + 4.01\alpha \\ 5 \end{pmatrix}$$

On a :

$$f(\varphi) = \varphi_3^2 + \frac{1}{2}\varphi_4^2 + \varphi_1 + 2\varphi_2 + \varphi_3 + 2\varphi_4 + 3\varphi_5 + 2\varphi_6$$

$$f(\varphi_u^k + \alpha d) = f(2, 3, 0.99 + 4.01\alpha, 4.01\alpha, 0.99 + 4.01\alpha, 5)$$

$$= (0.99 + 4.01\alpha)^2 + \frac{1}{2}(4.01(1 - \alpha))^2 + 2 + 2 \times 3 + 0.99 + 4.01\alpha + 2 \times (4.01(1 - \alpha)) + 3 \times (0.99 + 4.01\alpha) + 2 \times 5$$

$$= 0.9801 + 16.0801\alpha^2 + 7.9398\alpha + 8.04005 + 8.04005\alpha^2 - 16.0801\alpha + 2 + 6 + 0.99 + 4.01\alpha + 8.02 - 8.02\alpha + 2.97 + 12.03\alpha + 10$$

$$= 24.1202\alpha^2 - 0.1203\alpha + 39.00015$$

Trouver le minimum par la méthode de section dorée :

$$\alpha \in [a, b] = [0, 1], \text{ tolerance} = 10^{-2}$$

$$f(\alpha) = 24.1202\alpha^2 - 0.1203\alpha + 39.00015$$

$$m = \frac{\sqrt{5}-1}{2} \simeq 0.618$$

Après l'application de l'algorithme de section dorée on trouve : $\alpha = 0.006$ tq :

$$\alpha \in [a, b] = [0, 0.012]$$

Pas 4. Effectuer le pas :

$$\varphi_u^3 = \varphi_u^2 + \alpha(y_u^2 - \varphi_u^2)$$

$$\varphi_u^3 = \begin{pmatrix} 2 \\ 3 \\ 0.99 + 4.01\alpha \\ 4.01(1 - \alpha) \\ 0.99 + 4.01\alpha \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 0.99 + 4.01(0.006) \\ 4.01(1 - 0.006) \\ 0.99 + 4.01(0.006) \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 1.0141 \\ 3.9859 \\ 1.0141 \\ 5 \end{pmatrix}$$

$$\begin{aligned}
f(\varphi_u^3) &= f(2, 3, 1.0141, 3.9859, 1.0141, 5) \\
&= (1.0141)^2 + \frac{1}{2}(3.9859)^2 + 2 + 2 \times 3 + 1.0141 + 2 \times 3.9859 + 2 \times 5 \\
&= 1.0282 + 7.9437 + 2 + 6 + 1.0141 + 7.9718 + 3.0423 + 10 \\
&= 39.0001
\end{aligned}$$

Pas 5. Test de convergence :

$$\begin{aligned}
t &= \frac{\sqrt{\sum_u (\varphi_u^3 - \varphi_u^2)^2}}{\sum_u \varphi_u^2} \\
&= \frac{\sqrt{(2-2)^2 + (3-3)^2 + (1.0141-0.99)^2 + (3.9859-4.01)^2 + (1.0141-0.99)^2 + (5-5)^2}}{1+2+5+5+3+2} \\
&= \frac{\sqrt{(0.0006)^2 + (0.0006)^2 + (0.0006)^2}}{18} \\
&= \frac{\sqrt{0.0018}}{18} = 0.002 < 0.02
\end{aligned}$$

L'algorithme s'arrête avec :

1. $\varphi_u = \varphi_u^3 = (2, 3, 1.0141, 3.9859, 1.0141, 5)$.
2. Les temps de parcourt des arcs sont :

$$t(\varphi^3) = (1, 2, 3.0282, 5.9859, 3, 2)$$

Par remplacement de φ_u^3 dans (1) nous obtenons :

$$\left\{ \begin{array}{l} t_1 = 1 \\ t_2 = 2 \\ t_3 = 1 + 2 \times 1.0141 = 3.0282 \\ t_4 = 2 + 3.9859 = 5.9859 \\ t_5 = 3 \\ t_6 = 2 \end{array} \right.$$

Récapitulatif :

L'application de la méthode de Franck-Wolfe nécessite une suite d'affectation tout ou rien du réseau. Ces affectations consistent à mettre tous les voyageurs d'une même paire sur le plus court chemin de cette paire donc le problème de plus court chemin est central dans cette méthode.

le tableau suivant illustre Les itérations successives de l'algorithme de Franck-Wolfe :

k	$u = 1$	$u = 2$	$u = 3$	$u = 4$	$u = 5$	$u = 6$	$f(\varphi_1)$	α
0	$\varphi_1^0 = 0$ $t_1^0 = 1$ $\varphi_1^1 = 2$	$\varphi_2^0 = 0$ $t_2^0 = 2$ $\varphi_2^1 = 3$	$\varphi_3^0 = 0$ $t_3^0 = 1$ $\varphi_3^1 = 0$	$\varphi_4^0 = 0$ $t_4^0 = 2$ $\varphi_4^1 = 5$	$\varphi_5^0 = 0$ $t_5^0 = 3$ $\varphi_5^1 = 0$	$\varphi_6^0 = 0$ $t_6^0 = 2$ $\varphi_6^1 = 5$	0	
1	$t_1^1 = 1$ $y_1^1 = 2$ $\varphi_1^2 = 2$	$t_2^1 = 2$ $y_2^1 = 3$ $\varphi_2^2 = 3$	$t_3^1 = 1$ $y_3^1 = 5$ $\varphi_3^2 = 0.99$	$t_4^1 = 7$ $y_4^1 = 0$ $\varphi_4^2 = 4.01$	$t_5^1 = 3$ $y_5^1 = 5$ $\varphi_5^2 = 0.99$	$t_6^1 = 2$ $y_6^1 = 5$ $\varphi_6^2 = 5$	40.5	0.1980
3	$t_1^2 = 1$ $y_1^2 = 2$ $\varphi_1^3 = 2$ $t_1^3 = 1$	$t_2^2 = 2$ $y_2^2 = 3$ $\varphi_2^3 = 3$ $t_2^3 = 2$	$t_3^2 = 2.98$ $y_3^2 = 5$ $\varphi_3^3 = 1.0141$ $t_3^3 = 3.0282$	$t_4^2 = 6.01$ $y_4^2 = 0$ $\varphi_4^3 = 3.9859$ $t_4^3 = 5.9859$	$t_5^2 = 3$ $y_5^2 = 5$ $\varphi_5^3 = 1.0141$ $t_5^3 = 3$	$t_6^2 = 2$ $y_6^2 = 5$ $\varphi_6^3 = 5$ $t_6^3 = 2$	39.0003	0.006
							39.0001	

Illustration graphique de la solution :

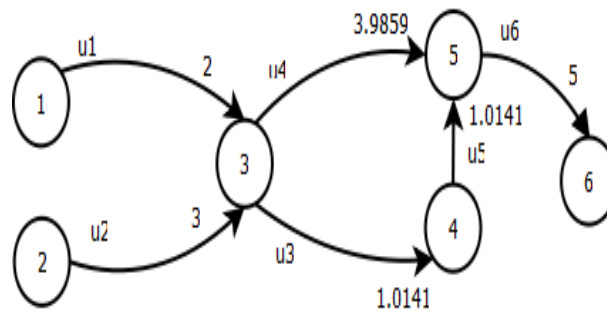


Illustration graphique de solution.

Interprétation du résultat :

L'algorithme de Franck-Wolfe sert à trouver une meilleure solution (meilleure distribution des voyageurs sur les arcs) qui rapprochent les temps de traversées de tous les chemins reliant l'origine o avec la destination d utilisés (1^{er} principe de wardrop).

En effet, dans cet exemple :

Au début de l'algorithme $k = 1 \rightarrow \varphi^1 = (2, 3, 0, 5, 0, 5)$

Les coûts de parcourt des chemins :

Pour $od = (16)$:

$$\begin{cases} c_{p1} = t(\varphi_1) + t(\varphi_4) + t(\varphi_6) = 1 + 2 + 2 = 5 \\ c_{p2} = t(\varphi_1) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 1 + 1 + 3 + 2 = 7 \end{cases}$$

Pour $od = (26)$:

$$\begin{cases} c_{p3} = t(\varphi_2) + t(\varphi_4) + t(\varphi_6) = 2 + 2 + 2 = 6 \\ c_{p4} = t(\varphi_2) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 2 + 1 + 3 + 2 = 8 \end{cases}$$

A la fin de l'algorithme $k = 3 \rightarrow \varphi^3 = (2, 3, 1.0141, 3.9859, 1.01406, 5)$

Les coûts de parcourt des chemins devient :

Pour $od = (16)$:

$$\begin{cases} c_{p1} = t(\varphi_1) + t(\varphi_4) + t(\varphi_6) = 1 + 5.9859 + 2 = 8.9859 \\ c_{p2} = t(\varphi_1) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 1 + 3.0282 + 3 + 2 = 9.0282 \end{cases} \implies c_{p1} \simeq c_{p2}$$

Pour $od = (26)$:

$$\begin{cases} c_{p3} = t(\varphi_2) + t(\varphi_4) + t(\varphi_6) = 2 + 5.9859 + 2 = 9.9859 \\ c_{p4} = t(\varphi_2) + t(\varphi_3) + t(\varphi_5) + t(\varphi_6) = 2 + 3.0282 + 3 + 2 = 10.0282 \end{cases} \implies c_{p3} \simeq c_{p4}$$

Remarque importante :

Ce résultat est atteint lorsque la $\text{tolerance} = 10^{-2}$

Si $\text{latolérance} < 10^{-2}$ alors l'algorithme fournit une solution plus proche de l'équilibre entre les coûts de parcours de tout les chemins reliant l'origine o avec la destination d .

Conclusion

A la fin de ce chapitre nous avons pu appliquer un exemple sur l'algorithme de Franck-Wolf dans le chapitre suivant nous allons modéliser et simuler cet exemple sur le logiciel Matlab afin de bien voir le résultat de cet algorithme.

5

Implémentation

Introduction

Dans ce chapitre, nous implémentons la méthode étudiée "Frank-Wolfe" sur le logiciel de programmation mathématique "MATLAB" afin de voir l'efficacité de notre méthode, nous testons le programme s'il améliore la fonction objectif.

5.1 Choix du langage

Le choix s'est porté sur l'emploi du langage du logiciel **MATLAB**, MATLAB ("matrix laboratory") est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, MATLAB permet de manipuler des matrices, d'afficher des courbes et des données, de mettre en œuvre les algorithmes, de créer des interfaces utilisateurs.

Matlab est un langage simple et très efficace, optimisé pour le traitement des matrices, d'où son nom. Pour le calcul numérique, Matlab est beaucoup plus concis que les "vieux" langages (C, Pascal, Fortran, Basic).

Un exemple : plus besoin de programmer des boucles modifier pour un à un les éléments d'une matrice. On peut traiter la matrice comme une simple variable.

Matlab contient également une interface graphique puissante, ainsi qu'une grande variété d'algorithmes scientifiques. On peut enrichir Matlab en ajoutant des "boîtes à outils" (toolbox) qui sont des ensembles de fonctions supplémentaires, profilées pour des applications particulières (traitement de signaux, analyses statistiques, optimisation, etc.).

Les fonctions du Matlab permettent de manipuler directement et interactivement

ces données matricielles, le rendant ainsi particulièrement efficace en calcul numérique, analyse et visualisation de données en particulier. Il existe deux modes de fonctionnement sur Matlab :

1. **Le mode interactif** : Les instructions sont exécutées au fur et à mesure qu'elles sont données par l'usage.
2. **Le mode exécutif** : Dans ce cas, l'utilisateur utilise un fichier " M-file " contenant toutes les instructions à exécuter.

Les fonctions prédéfinis utilisées :

- La fonction **graphshortestpath** cherche le plus court chemin entre deux sommets origine-destination (L) et sa valeur (dis) ainsi que l'ensemble de prédécesseurs ($pred$) dans un graphe orientée. Pour utiliser cette fonction nous utilisons le code suivant :

$$[dist, L, pred] = graphshortestpath(DG, o, d)$$

- Les fonctions mathématiques **sqrt** trouve la racine carrée d'un nombre positif.

La structure sparse :

On appelle matrice creuse (" sparse matrix ") une matrice comportant une forte proportion de coefficients nuls. De nombreux problèmes issus de la physique conduisent à l'analyse de systèmes linéaires à matrice creuse. L'intérêt de telles matrices résulte non seulement de la réduction de la place mémoire (on ne stocke pas les zéros) mais aussi de la réduction du nombre d'opérations (on n'effectuera pas les opérations portant sur les zéros). Par défaut dans matlab une matrice est considérée comme pleine (ou " full" en anglais), c'est-à-dire que tous ses coefficients sont mémorisés. Si A est une matrice, la commande **sparse (M)** permet d'obtenir la même matrice mais stockée sous la forme sparse.

Si l'on a une matrice stockée sous la forme **sparse**, on peut obtenir la même matrice stockée sous la forme ordinaire par la commande **full**.

Exemple :

```
>> A = [1 2 3;0 1 0;1 0 0]
```

```
A =
```

```
1 2 3
0 1 0
1 0 0
```

```
>> sparse(A)
```

```
ans =
```

(1, 1)	1
(3, 1)	1
(1, 2)	2
(2, 2)	1
(1, 3)	3

5.2 La Fonction BPR (Bureau of Public Roads (1964))

La Fonction BPR est une fonction de coût (fonction de performance des arcs) largement utilisé est suggéré par le Bureau de la voie publique (BPR) 1964 de la forme suivante :

$$t(\varphi) = t(0)[1 + a(\varphi/cap)^b]$$

Où :

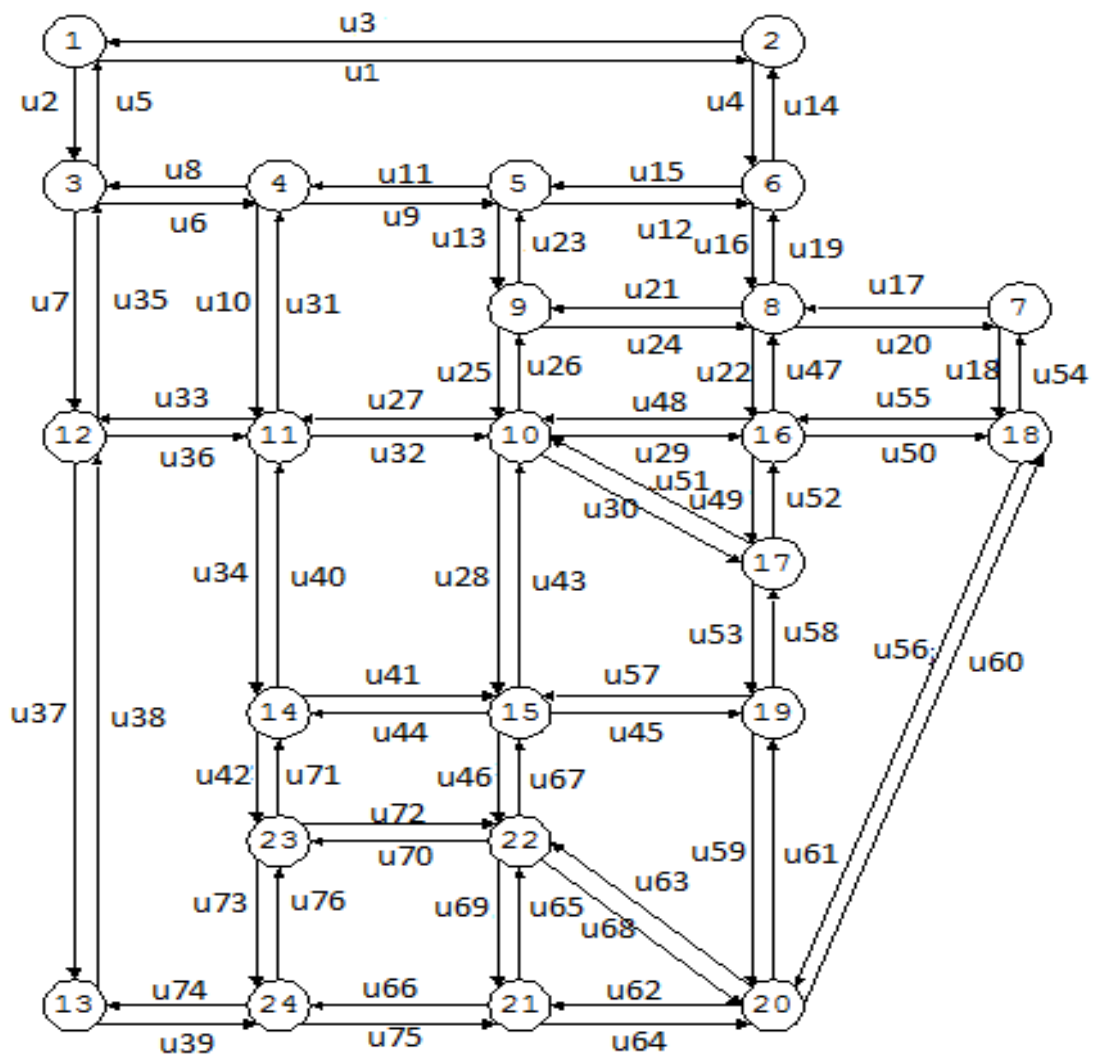
- φ : est le flux sur l'arc u (nombre de véhicules).
- $t(\varphi)$: est le temps de voyage lorsque le flux est φ .
- $t(0)$: est le temps de voyage à écoulement libre, lorsque φ est proche de zéro.
- cap :est la capacité pratique de la liaison routière.
- a, b :sont des paramètres de réglage pour les différents types de routes.

Les valeurs les plus courantes actuellement utilisées sont $a = 0,15$ et $b = 4$ (d'où l'abus de langage souvent utilisé "loi de puissance quatrième").

5.3 Exemple d'application (Instance de Benchmark)

Le Benchmarking est un aspect important de la recherche opérationnelle. Dès lors qu'on veut évaluer en pratique des algorithmes, il est nécessaire de les exécuter sur de nombreuses données.

Dans cette section, nous fournissons des informations détaillées sur les instances de Benchmark existantes, En particulier, l'instance de sioux falls (une ville des États-Unis, siège du comté de Minnehaha, dans l'est du Dakota du Sud, à quelques kilomètres des États du Minnesota et de l'Iowa.) :

FIGURE 5.1 – Réseau de sioux falls $G = (X, U, T)$ **Les données [29] :**

Le réseau de sioux falls comporte 24 sommets ($|X| = 24$) et 76 arcs ($|U| = 76$).

1. L'offre :

l'offre est représenté par un tableau dans l'annexe 1 (76 lignes et 4 colonnes), où les arcs sont représentés par l'existence de deux extrémités sommet initiale (*colonne1*) et sommet finale (*colonne2*), la capacité de ces arcs est représenté dans la colonne 3, la quatrième colonne représente le temps de parcourt à vide (T_0).

2. La demande :

La demande est représentée dans une matrice carrée q_{od} (nombre de véhicules /heure) dans l'annexe 1 (24 lignes et 24 colonnes) où les indices des lignes comportent l'ensemble des sommets origine (O), et les indices des colonnes comportent l'ensemble des sommets destination (D).

5.4 Application sous MATLAB

1.Implémentation des données :

Dans la commande Windows nous allons saisir les données de Banchmark par des matrices sous forme de matrice d'adjacence (Voir l'annexe 2)

- La matrice T0 des temps de parcourt :

$$T0 = \begin{cases} t0(k) & \text{si } (x_i, x_j) \in U ; \\ 0 & \text{Sinon.} \end{cases}$$

Tel que $1 \leq i, j \leq n, 1 \leq k \leq m$

- La matrice de capacité :

$$cap = \begin{cases} cap(k) & \text{si } (x_i, x_j) \in U ; \\ 0 & \text{Sinon.} \end{cases}$$

Tel que $1 \leq i, j \leq n, 1 \leq k \leq m$

- La matrice de demande q_{od} .

2.Exécution :

L'exécution sera atteinte lorsque fait entrer le nom donné a notre programme :

```
>> [ $\varphi$ ,f]=Franck-Wolfe(q,T0,cap)
```

3.Résultat :

Le tableau ci-dessous illustre les résultats du l'implémentation de l'algorithme de Franck Wolfe, tel que, au niveau de la précision (10^{-2}) l'algorithme fournit 50 itérations en temps d'exécution rapide (*5second*), en remarquant que la valeur de la fonction objectif diminue dans chaque itération K , par conséquent la fonction objectif s'améliore.

K	$f(\varphi) \times 10^6$	Pas de déplacement
1	6.6945	0.3290
2	5.6695	0.2110
3	5.3441	0.1830
4	4.8766	0.4690
5	4.7186	0.1310
6	4.5776	0.2730
7	4.5184	0.1130
8	4.3936	0.4230
9	4.3421	0.1030
10	4.2789	0.2610
11	4.2468	0.1650
12	4.2054	0.2770
13	4.1901	0.0970
14	4.1660	0.2670
15	4.1506	0.1130
16	4.1413	0.1230
17	4.1336	0.1030
18	4.1147	0.3110
19	4.1055	0.0810
20	4.1018	0.0910
21	4.0959	0.1390
22	4.0932	0.0590
23	4.0894	0.1310
24	4.0874	0.0490
25	4.0811	0.2670

K	$f(\varphi) \times 10^6$	Pas de déplacement
26	4.0770	0.0770
27	4.0755	0.0750
28	4.0743	0.0490
29	4.0724	0.1030
30	4.0713	0.0370
31	4.0697	0.1010
32	4.0686	0.0450
33	4.0676	0.0710
34	4.0670	0.0370
35	4.0658	0.0930
36	4.0651	0.0430
37	4.0638	0.1050
38	4.0629	0.0450
39	4.0627	0.0250
40	4.0618	0.0970
41	4.0613	0.0410
42	4.0610	0.0370
43	4.0606	0.0530
44	4.0601	0.0650
45	4.0599	0.0150
46	4.0597	0.0310
47	4.0590	0.0870
48	4.0588	0.0150
49	4.0586	0.0250
50	4.0585	0.0150

Illustration graphique des résultats :

La solution finale φ (nombre de véhicule/heure) est représenté en termes des flux sur les arcs du réseau de sioux- falls .

Prévision sur l'état du trafic au point d'équilibre (φ) :

Le tableau ci-dessous présente une prévision sur l'état de trafic du réseau de Sioux-Falls en un temps précisé (le matin), où on va représenter les arcs par deux sommets, initial (SI) et final (SF) auquel associons à chaque arc une capacité, un flux, un coût(temps de parcourt), un pourcentage de congestion(PC) qui représenté le taux de flux par rapport à la capacité et l'état de trafic par rapport au pourcentage de congestion .

SI	SF	Capacité	Flux (φ)	Coût	PC	Etat de trafic
1	2	25900.20064	3694	6.0004	14	Fluide
1	3	23403.47319	7304	4.0057	31	Fluide
2	1	25900.20064	3708	6.0004	14	Fluide
2	6	4958.180928	5438	6.0861	110	Forte congestion
3	1	23403.47319	7291	4.0057	31	Fluide
3	4	17110.52372	13873	4.2604	81	Congestion gênante
3	12	23403.47319	9369	4.0149	40	Fluide
4	3	17110.52372	13919	4.2563	81	Congestion gênante
4	5	17782.7941	17064	2.2550	96	Congestion gênante
4	11	4908.82673	5783	7.7384	118	Forte congestion
5	4	17782.7941	17037	2.2473	96	Congestion gênante
5	6	4947.995469	8463	9.1253	171	Forte congestion
5	9	100005	15279	5.0004	15	Fluide
6	2	4958.180928	5452	6.0971	110	Forte congestion
6	5	4947.995469	8456	9.0278	171	Forte congestion
6	8	4898.587646	11738	11.8825	240	Forte congestion
7	8	7841.81131	11897	5.3888	152	Forte congestion
7	18	23403.47319	15448	2.0570	66	Forte congestion
8	6	4898.587646	11745	11.7927	240	Forte congestion
8	7	7841.81131	11853	5.3531	151	Forte congestion
8	9	5050.193156	7216	16.3173	143	Forte congestion
8	16	5045.822583	8161	10.2105	162	Forte congestion
9	5	100005	1.5258	5.0004	15	Fuide
9	8	5050.193156	7171	16.2305	142	Forte congestion
9	10	13915.78842	21238	5.4319	153	Forte congestion
10	9	13915.78842	21272	5.4255	153	Forte congestion
10	11	100005	19906	5.0012	20	Fuide
10	15	13512.00155	23133	13.7374	171	Forte congestion
10	16	4854.917717	11169	20.3980	230	Forte congestion
10	17	4993.510694	8102	16.3183	162	Forte congestion
11	4	4908.82673	5955	7.9565	121	Forte congestion
11	10	100005	19736	5.0011	20	Fuide
11	12	4908.82673	8872	15.9880	181	Forte congestion
11	14	4876.508287	9952	14.5389	204	Forte congestion
12	3	23403.47319	9310	4.0151	39	Fluide
12	11	4908.82673	8933	15.8116	181	Forte congestion
12	13	25900.20064	12371	3.0234	48	Fluide

SI	SF	Capacité	Flux (φ)	Coût	PC	Etat de trafic
13	12	25900.20064	12473	3.0242	48	Fluide
13	24	5091.256152	11256	18.3261	221	Forte congestion
14	11	4876.508287	9994	14.5442	205	Forte congestion
14	15	5127.526119	5173	5.7857	101	Forte congestion
14	23	4924.790605	8142	8.4830	165	Forte congestion
15	10	13512.00155	23195	13.8110	452	Forte congestion
15	14	5127.526119	5144	5.7438	100	Forte congestion
15	19	14564.75315	19585	4.4824	134	Forte congestion
15	22	9599.180565	18719	9.5185	195	Forte congestion
16	8	5045.822583	8169	10.0932	162	Forte congestion
16	10	4854.917717	11193	21.1434	231	Forte congestion
16	17	5229.910063	11661	9.4019	223	Forte congestion
16	18	19679.89671	14553	3.1331	74	Fluide
17	10	4993.510694	8121	16.3974	163	Forte congestion
17	16	5229.910063	11676	9.4284	223	Forte congestion
17	19	4823.950831	9938	7.4228	209	Forte congestion
18	7	23403.47319	15492	2.0576	66	Congestion moyen
18	16	19679.89671	14569	3.1355	74	Congestion gênante
18	20	23403.47319	19054	4.2612	81	Congestion gênante
19	15	14564.75315	19607	4.4761	135	Forte congestion
19	17	4823.950831	9972	7.4876	207	Forte congestion
19	20	5002.607563	8775	9.7711	175	Forte congestion
20	18	23403.47319	19015	4.2619	81	Fluide
20	19	5002.607563	8830	9.7942	177	Forte congestion
20	21	5059.91234	6368	8.2442	126	Forte congestion
20	22	5075.697193	7179	8.0036	142	Forte congestion
21	20	5059.91234	6312	8.1736	125	Forte congestion
21	22	5229.910063	8022	3.6866	153	Forte congestion
21	24	4885.357564	7045	4.8726	144	Forte congestion
22	15	9599.180565	18630	9.3876	194	Forte congestion
22	20	5075.697193	7151	7.9558	141	Forte congestion
22	21	5229.910063	8086	3.6655	155	Forte congestion
22	23	50004	1.6458	40071	33	Fluide
23	14	4924.790605	8212	8.6429	167	Forte congestion
23	22	50004	1.6404	40069	33	Fluide
23	24	5078.508436	8591	4.5343	169	Forte congestion
24	13	5091.256152	11259	18.3384	221	Forte congestion
24	21	4885.357564	6926	4.8456	142	Forte congestion
24	23	5078.508436	8608	4.4451	169	Forte congestion

Interprétation :

Le réseau ci-dessus représente l'état de trafic de Sioux-Falls au point d'équilibre (la meilleure distribution de voyageurs selon Wardrop).

A partir de ce résultat on peut envisager les axes congestionnés pour faire intervenir les partisans du transport afin de minimiser ces derniers.

Analyse de l'algorithme Frank Wolf :

	Temps d'exécution	Espace mémoire occupé	Niveau de précision
Réseau de SiouxFalls	Petit ($< 6s$)	Petit ($< 4MO$)	Petit (10^{-2})

Interpretation :**1.L'avantage :**

- L'algorithme de Franck-Wolfe fournit rapidement une bonne solution lorsque le niveau de précision est petit (10^{-2}).
- L'algorithme de Franc-Wolfe occupe un espace mémoire petite (inferieur à 4MO).

2.L'inconvénient :

- L'algorithme de Franck-Wolfe nécessite un long temps d'exécution si le niveau de précision est très petit.

Conclusion

L'utilisation de Matlab pour le calcul numérique, sur l'exemple de Benchmark montre une nette amélioration de la fonction objectif (diminution de la valeur de la fonction objectif), ceci a permis de faire des prévisions sur l'état de trafic au point d'équilibre pour la représentation des axes fluides , des axes à congestion gênante et des axes à forte congestion .

Conclusion générale

Dans ce travail, la démarche adoptée pour l'étude de Problème d'Affectation de Trafic Urbain (PATU) est établi à travers le passage par les deux aspects, théorique et numérique.

Dans la partie théorique le travail consiste en l'étude du modèle mathématique, tel que l'existence, les caractéristiques, les conditions d'optimalité, l'unicité de la solution qui désigne un équilibre. Un exemple d'application illustratif sur un petit réseau est présenté afin de mieux éclaircir les démarches de la modélisation et les caractéristiques de cet équilibre.

Dans la partie numérique nous avons observer comment l'algorithme de Franck Wolfe s'adapte pour résoudre PATU, un détail de l'algorithme ainsi qu'une application sur le même exemple sont effectués afin de prouver la nécessité de l'aspect numérique pour PATU.

Par la suite, nous avons élaborer une application à travers le langage de programmation Matlab afin de voir de près la solution d'un exemple d'application, nous avons exécuter notre application sur l'instance de Benchmark.

Enfin, compte tenu de la diversité des notions et des bases nécessaires, nous n'avons pu étudier qu'un petit noyau de cette problématique, pour cela nous proposons aux lecteurs des perspectives qui semblent pour nous ouvrir des nouvelles voies de recherche tel que :

- L'étude PATU en modèle stochastique et dynamique,
- L'analyse comparative entre les méthodes de résolution de PATU,
- Proposition des autres modèles mathématiques,
- ⋮

Bibliographie

- [1] A.Rondepierre and P.Weiss, *Méthodes Standards en Optimisation Linéaire Déterministe*, 1^{er} Edition, Année 2013.
- [2] C.Solnon, *Théorie des graphes et optimisation dans les graphes*, 2^{ème} Edition, Edition Hermann, Année2000.
- [3] D.Dewolf, *Transport et Environnement*, 1^{er} Edition, Edition Dunkerque, Année 2003.
- [4] D.Müller, *Introduction à la théorie des graphes*, Edition : Cahiers n°6 de la crm.
- [5] D.Gleich, *Matlab*, 1^{er} Edition, Edition : BGL v1.0, Année 2002.
- [6] D.Werra, *Etude et illustration de méthodes itératives d'optimisation non linéaire*, Ecole Polytechnique Fédérale de Lausanne, Année 2006 .
- [7] G. Rodriguez-Guisantes, *Introduction á MATLAB*, Année 2008.
- [8] G.Legendre, *Méthodes numériques Introduction à l'analyse numérique et au calcul scientifique*, Année 2016.
- [9] H.Hocquard, *Optimisation des fonctions de plusieurs variables*, Université de Bordeaux, France, Année 2013.
- [10] I.Todinca , *Algorithmique des graphes*, 2^{ème} Edition, Année 2008.
- [11] I.Charon and O.Hudry, *Optimisation non linéaire*, École nationale supérieure des télécommunications.
- [12] J.Grenet, *Vade mecum : Optimisation statique.Lagrangien et conditions de Kuhn et Tucker*. École Normale Supérieure, Année 2008.
- [13] K.Benjamin, *Modélisation, Optimisation, Complexité et Algorithmes*, Année 2006.
- [14] M.Minoux, *Programmation mathématique, Théorie et algorithmes*, 2ème Edition, Edition TecDoc, Année 2008.
- [15] M.Guidère, *Les méthodes d'analyse*, 3^{ème} Edition, Edition Hermann, Année1998.
- [16] M.Patriksson, *The Traffic Assignment Problème-Models and Methods*.
- [17] M.Gondran and M.Minoux, *Graphes et algorithmes*. 4^eédition, Edition TEC DOC, Année 2009.
- [18] Ph. Roux, *Théorie des graphes*, 4^eédition, Année 2013.
- [19] R-K.Sundaram, *A Firste cours in optimization*, Cambridge University Press, Année 1999.
- [20] S.Fomine, A.Kolmogorove, *Elément de la théorie des fonctions et de l'analyse fonctionnelle*, Editions MIR.MOSCOU, Année 1977.

- [21] S.Yosef, *Urban Transportation Networks : Equilibrium Analysis with Mathematical Programming Methods*, Prentice-Hall, New Jersey, Année 1985 .
- [22] S.Mottelet, *Optimisation non-linéaire*, Université de Technologie de Compiègne, Année 2003.
- [23] S.Gueye and B.Ndiaye, *Optimisation des Réseaux de Transport du Sénégal : Analyse des flots et localisation d'activités*, Université du Havre, Année 2006.
- [24] S.Téphane Pelle, *Geomatique la théorie des graphes*, École nationale des sciences géographiques. Année 2006.
- [25] S. Laurent, *Analyse topologie générale et analyse fonctionnelle*, 2^{ème} Edition revue et corrigée : Hermann, Année1999.
- [26] S.Fomine and A.Kolmogorove, *Elément de la théorie des fonctions et de l'analyse fonctionnelle*, Editions Mir.Moscou, Année 1977.
- [27] T. Liebling and J-F.Hêche and D.Werra, *Recherche opérationnelle pour ingénieurs 1*.
- [28] Y.Correc, *Graphes. Recherche Opérationnelle*, Année 2007.
- [29] [http ://www.bgu.ac.il/ bargera/tntp/](http://www.bgu.ac.il/~bargera/tntp/)

Annexe

Annexe1 1.L'offre :

sommet initiale	sommet finale	Capacité	Temps de parcourt à vide(t(0))
1	2	25900.20064	6
1	3	23403.47319	4
2	1	25900.20064	6
2	6	4958.180928	5
3	1	23403.47319	4
3	4	17110.52372	4
3	12	23403.47319	4
4	3	17110.52372	4
4	5	17782.7941	2
4	11	4908.82673	6
5	4	17782.7941	2
5	6	4947.995469	4
5	9	100005	5
6	2	4958.180928	5
6	5	4947.995469	4
6	8	4898.587646	2
7	8	7841.81131	3
7	18	23403.47319	2
8	6	4898.587646	2
8	7	7841.81131	3
8	9	5050.193156	10
8	16	5045.822583	5
9	5	100005	5
9	8	5050.193156	10
9	10	13915.78842	3
10	9	13915.78842	3
10	11	100005	5
10	15	13512.00155	6
10	16	4854.917717	4
10	17	4993.510694	8
11	4	4908.82673	6
11	10	100005	5

Annexe

sommet initiale	sommet finale	Capacité	Temps de parcourt à vide($t(0)$)
11	12	4908.82673	6
11	14	4876.508287	4
12	3	23403.47319	4
12	11	4908.82673	6
12	13	25900.20064	3
13	12	25900.20064	3
13	24	5091.256152	4
14	11	4876.508287	4
14	15	5127.526119	5
14	23	4924.790605	4
15	10	13512.00155	6
15	14	5127.526119	5
15	19	14564.75315	3
15	22	9599.180565	3
16	8	5045.822583	5
16	10	4854.917717	4
16	17	5229.910063	2
16	18	19679.89671	3
17	10	4993.510694	8
17	16	5229.910063	2
17	19	4823.950831	2
18	7	23403.47319	2
18	16	19679.89671	3
18	20	23403.47319	4
19	15	14564.75315	3
19	17	4823.950831	2
19	20	5002.607563	4
20	18	23403.47319	4
20	19	5002.607563	4
20	21	5059.91234	6
20	22	5075.697193	5
21	20	5059.91234	6
21	22	5229.910063	2
21	24	4885.357564	3
22	15	9599.180565	3
22	20	5075.697193	5
22	21	5229.910063	2
22	23	50004	4
23	14	4924.790605	4
23	22	50004	4
23	24	5078.508436	2
24	13	5091.256152	4
24	21	4885.357564	3
24	23	5078.508436	2

2.la matrice de demande(q) :

 $q_{od} = 10^2*$

0	1	1	5	2	3	5	8	5	13	5	2	5	3	5	5	4	1	3	3	1	4	3	1
1	0	1	2	1	4	2	4	2	6	2	1	3	1	1	4	2	0	1	1	0	1	0	0
1	1	0	2	1	3	1	2	1	3	3	2	1	1	1	2	1	0	0	0	0	1	1	0
5	2	2	0	5	4	4	7	7	12	14	6	6	5	5	8	5	1	2	3	2	4	5	2
2	1	1	5	0	2	2	5	8	10	5	2	2	1	2	5	2	0	1	1	1	2	1	0
3	4	3	4	2	0	4	8	4	8	4	2	2	1	2	9	5	1	2	3	1	2	1	1
5	2	1	4	2	4	0	10	6	19	5	7	4	2	5	14	10	2	4	5	2	5	2	1
8	4	2	7	5	8	10	0	8	16	8	6	6	4	6	22	14	3	7	9	4	5	3	2
5	2	1	7	8	4	6	8	0	28	14	6	6	6	9	14	9	2	4	6	3	7	5	2
13	6	3	12	10	8	19	16	28	0	40	20	19	21	40	44	39	7	18	25	12	26	18	8
5	2	3	15	5	4	5	8	14	39	0	14	10	16	14	14	10	1	4	6	4	11	13	6
2	1	2	6	2	2	7	6	6	20	14	0	13	7	7	7	6	2	3	4	3	7	7	5
5	3	1	6	2	2	4	6	6	19	10	13	0	6	7	6	5	1	3	6	6	13	8	8
3	1	1	5	1	1	2	4	6	21	16	7	6	0	13	7	7	1	3	5	4	12	11	4
5	1	1	5	2	2	5	6	10	40	14	7	7	13	0	12	15	2	8	11	8	26	10	4
5	4	2	8	5	9	14	22	14	44	14	7	6	7	12	0	28	5	13	16	6	12	5	3
4	2	1	5	2	5	10	14	9	39	10	6	5	7	15	28	0	6	17	17	6	17	6	3
1	0	0	1	0	1	2	3	2	7	2	2	1	1	2	5	6	0	3	4	1	3	1	0
3	1	0	2	1	2	4	7	4	18	4	3	3	3	8	13	17	3	0	12	4	12	3	1
3	1	0	3	1	3	5	6	25	6	5	6	5	11	16	17	4	12	0	12	24	7	4	
1	0	0	2	1	1	2	4	3	12	4	3	6	4	8	6	6	1	4	12	0	18	7	5
4	1	1	4	2	2	5	5	7	26	11	7	13	12	26	12	17	3	12	24	18	0	21	11
3	0	1	5	1	1	2	3	5	18	13	7	8	11	10	5	6	1	3	7	7	21	0	7
1	0	0	2	0	1	1	2	2	8	6	5	7	4	4	3	3	0	1	4	5	11	7	0

3.La matrice q (q_{od}) :

```
>> q=[0 100 100 500 200 300 500 800 500 1300 500 200 500 300 500 500 400 100
300 300 100 400 300 100; 100 0 100 200 100 400 200 400 200 600 200 100 300 100 100
400 200 0 100 100 0 100 0 0; 100 100 0 200 100 300 100 200 100 300 300 200 100 100 100
200 100 0 0 0 0 100 100 0; 500 200 200 0 500 400 400 700 700 1200 1400 600 600 500 500
800 500 100 200 300 200 400 500 200; 200 100 100 500 0 200 200 500 800 1000 500 200
200 100 200 500 200 0 100 100 100 200 100 0; 300 400 300 400 200 0 400 800 400 800 400
200 200 100 200 900 500 100 200 300 100 200 100 100; 500 200 100 400 200 400 0 1000
600 1900 500 700 400 200 500 1400 1000 200 400 500 200 500 200 100; 800 400 200 700
500 800 1000 0 800 1600 800 600 600 400 600 2200 1400 300 700 900 400 500 300 200;
500 200 100 700 800 400 600 800 0 2800 1400 600 600 600 900 1400 900 200 400 600 300
700 500 200;1300 600 300 1200 1000 800 1900 1600 2800 0 4000 2000 1900 2100 4000
4400 3900 700 1800 2500 1200 2600 1800 800; 500 200 300 1500 500 400 500 800 1400
3900 0 1400 1000 1600 1400 1400 1000 100 400 600 400 1100 1300 600; 200 100 200 600
200 200 700 600 600 2000 1400 0 1300 700 700 700 600 200 300 400 300 700 700 500; 500
300 100 600 200 200 400 600 600 1900 1000 1300 0 600 700 600 500 100 300 600 600 1300
800 800; 300 100 100 500 100 100 200 400 600 2100 1600 700 600 0 1300 700 700 100 300
500 400 1200 1100 400; 500 100 100 500 200 200 500 600 1000 4000 1400 700 700 1300
0 1200 1500 200 800 1100 800 2600 1000 400; 500 400 200 800 500 900 1400 2200 1400
4400 1400 700 600 700 1200 0 2800 500 1300 1600 600 1200 500 300; 400 200 100 500 200
500 1000 1400 900 3900 1000 600 500 700 1500 2800 0 600 1700 1700 600 1700 600 300;
100 0 0 100 0 100 200 300 200 700 200 200 100 100 200 500 600 0 300 400 100 300 100
0; 300 100 0 200 100 200 400 700 400 1800 400 300 300 300 800 1300 1700 300 0 1200
400 1200 300 100; 300 100 0 300 100 300 500 900 600 2500 600 500 600 500 1100 1600
1700 400 1200 0 1200 2400 700 400; 100 0 0 200 100 100 200 400 300 1200 400 300 600
400 800 600 600 100 400 1200 0 1800 700 500; 400 100 100 400 200 200 500 500 700 2600
1100 700 1300 1200 2600 1200 1700 300 1200 2400 1800 0 2100 1100;300 0 100 500 100
100 200 300 500 1800 1300 700 800 1100 1000 500 600 100 300 700 700 2100 0 700; 100 0
0 200 0 100 100 200 200 800 600 500 700 400 400 300 300 0 100 400 500 1100 700 0]
```

Annexe3
Le résultat (la solution)

>> [φ ,f] = Franck-Wolfe(q,T0,cap)

les valeurs de la fonction objectif pour chaque itération :

f =

1.0e+06 *

Columns 1 through 8

0 6.6945 5.6695 5.3441 4.8766 4.7186 4.5776 4.5184

Columns 9 through 16

4.3936 4.3421 4.2789 4.2468 4.2054 4.1901 4.1660 4.1506

Columns 17 through 24

4.1413 4.1336 4.1147 4.1055 4.1018 4.0959 4.0932 4.0894

Columns 25 through 32

4.0874 4.0811 4.0770 4.0755 4.0743 4.0724 4.0713 4.0697

Columns 33 through 40

4.0686 4.0676 4.0670 4.0658 4.0651 4.0638 4.0629 4.0627

Columns 41 through 48

4.0618 4.0613 4.0610 4.0606 4.0601 4.0599 4.0597 4.0590

Columns 49 through 51

4.0588 4.0586 4.0585

La meilleure distribution des flux est :

φ =

1.0e+04 *

Annexe

Columns 1 through 8

0	0.3694	0.7304	0	0	0	0	0
0.3708	0	0	0	0	0.5438	0	0
0.7291	0	0	1.3873	0	0	0	0
0	0	1.3919	0	0	1.7064	0	0
0	0	0	1.7037	0	0	0	0
0	0.5452	0	0	0.8456	0	0	1.1738
0	0	0	0	0	0	0	1.1897
0	0	0	0	0	1.1745	1.1853	0
0	0	0	0	1.5258	0	0	0.7171
0	0	0	0	0	0	0	0
0	0	0	0.5955	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0.9310	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0.8169
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1.5492	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Annexe

Columns 9 through 16

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0.9369	0	0	0	0
0	0	0.5783	0	0	0	0	0
1.5279	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0.7216	0	0	0	0	0	0	0.8161
0	2.1238	0	0	0	0	0	0
2.1272	0	1.9906	0	0	0	2.3133	1.1169
0	1.9736	0	0.8872	0	0.9952	0	0
0	0	0.8933	0	1.2371	0	0	0
0	0	0	1.2473	0	0	0	0
0	0	0.9994	0	0	0	0.5173	0
0	2.3195	0	0	0	0.5144	0	0
0	1.1193	0	0	0	0	0	0
0	0.8121	0	0	0	0	0	1.1676
0	0	0	0	0	0	0	1.4569
0	0	0	0	0	0	1.9607	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	1.8630	0
0	0	0	0	0	0.8212	0	0
0	0	0	0	1.1259	0	0	0

Annexe

Columns 17 through 24

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1.5448	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0.8102	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1.1256
0	0	0	0	0	0	0.8142	0
0	0	1.9585	0	0	1.8719	0	0
1.1661	1.4553	0	0	0	0	0	0
0	0	0.9938	0	0	0	0	0
0	0	0	1.9054	0	0	0	0
0.9972	0	0	0.8775	0	0	0	0
0	1.9015	0.8830	0	0.6368	0.7179	0	0
0	0	0	0.6312	0	0.8022	0	0.7045
0	0	0	0.7151	0.8086	0	1.6458	0
0	0	0	0	0	1.6404	0	0.8591
0	0	0	0	0.6926	0	0.8608	0

Illustration graphique de résultat :
Biograph object with 24 nodes and 76 edges.

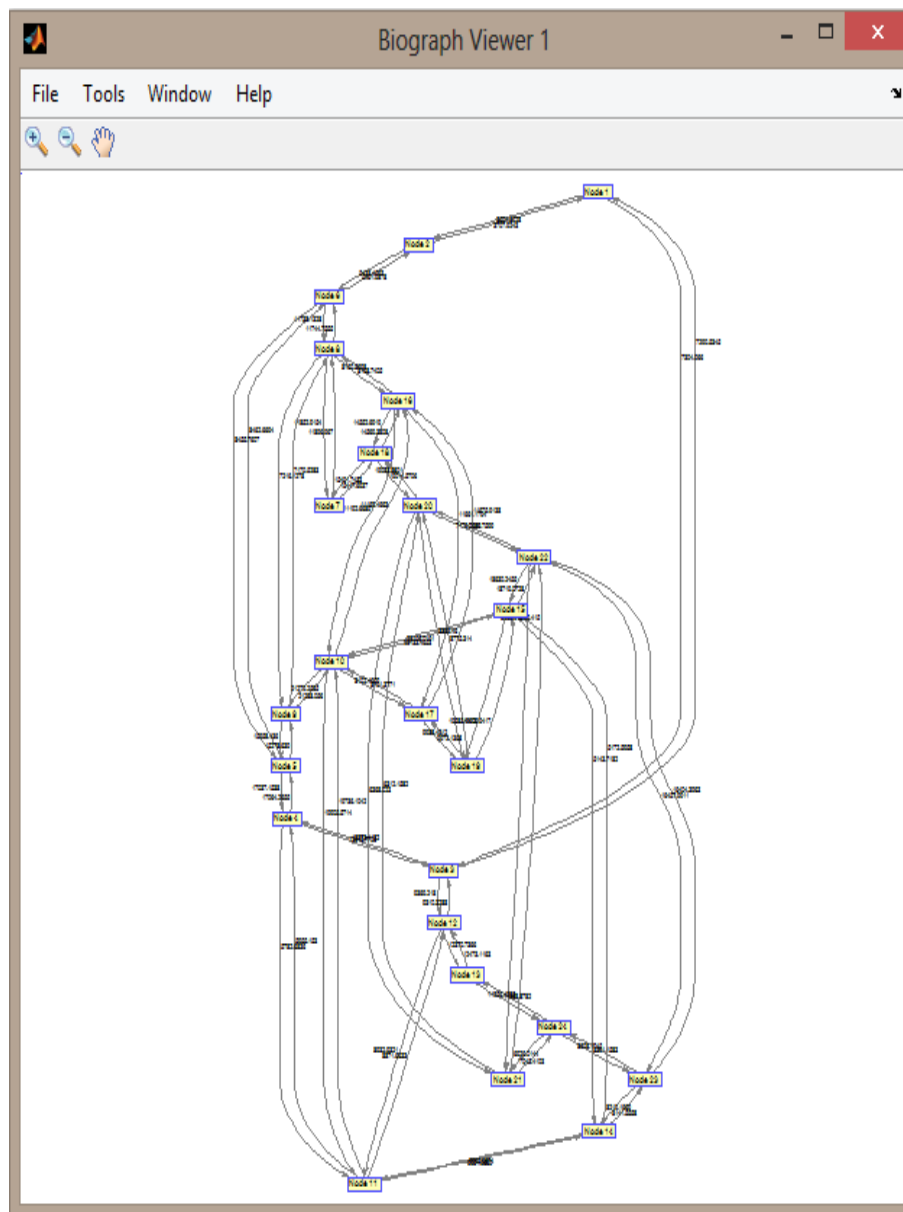


FIGURE 5.2 –