

Université de Montréal

Problème d'affectation dynamique sur un réseau de transport

par

Jose M. Rubio-Ardanaz

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures

en vue de l'obtention du grade de

Philosophiæ Doctor (Ph. D.)

en informatique

Juillet, 2000

© JOSE M. RUBIO-ARDANAZ, 2000



QA

76

U54

2002

v. 007

Université de Montréal

Faculté des études supérieures

Cette thèse intitulée

Problème d'affectation dynamique sur un réseau de transport

présentée par

Jose M. Rubio-Ardanaz

a été évaluée par un jury composé des personnes suivantes :

Président-rapporteur: L'Ecuyer, Pierre
Directeur de recherche: Florian, Michael
Codirecteur : Wu, Sia Hao
Membre du jury: Marcotte, Patrice
Examinateur externe: Chabeni, Ismail
*Représentant du doyen
de la FES: Doffow, Michel*

Thèse acceptée le : 15 mars 2002

À mes enfants Jose et Arantza.

Table des matières

Liste des tableaux	viii
Liste des figures	ix
1 Introduction	2
2 Revue de la littérature	10
2.1 Notation et notions de base	10
2.2 Formulation du modèle d'équilibre dynamique	12
2.2.1 Temps de parcours instantané d'un chemin ("Modèle réactif").	13
2.2.2 Temps de parcours effectif d'un chemin. ("Modèle préventif")	18
2.3 Exposé des contributions dans la littérature	22
2.4 Modèles basés sur la programmation mathématique	24
2.4.1 Merchant et Nemhauser : modèles discrets d'équilibre dynamique normatifs	24
2.4.2 Modèles discrets d'équilibre dynamique basés sur des réseaux espace-temps	28
2.5 Modèles de contrôle optimal	32
2.5.1 Modèles de Friesz et al. : avec fonctions du flot de sortie	32
2.5.2 Modèles de Ran et al. : avec fonctions de temps sur les arcs	35
2.6 Modèles d'inégalités variationnelles (PIV)	38
2.6.1 Formulation basée sur le coût instantané des chemins	39

2.6.2	Formulation basée sur le coût effectif de chemins	40
2.6.3	Chargement dynamique de réseau (network loading)	41
2.6.4	Modèle avec choix du temps de départ	45
2.6.5	Formulations continues	49
2.6.6	Formulations discrètes	51
3	Formulation mathématique : les contraintes du modèle considéré	55
3.1	Contraintes d'état	57
3.2	Contraintes de conservation du flot	58
3.3	Contraintes de propagation du flot	59
3.3.1	Contraintes de propagation du flot dans l'espace des chemins .	60
3.3.2	Contraintes de propagation du flot dans l'espace des arcs . . .	60
3.4	Contrainte FIFO : premier entré-premier sorti	61
3.5	Fonction du flot de sortie	64
4	Résolution du problème de chargement : deux nouveaux algorithmes	66
4.1	Algorithme-1 (variante de DYNALOAD)	69
4.2	Algorithme-2	76
4.3	Résultats numériques	81
5	Résolution du problème d'équilibre dynamique	87
5.1	Formulation du problème d'inégalité variationnelle	89
5.2	Méthode de projection	91
5.2.1	Direction de projection	93
5.3	Résultats numériques	97
5.4	Méthode de restriction	112
5.4.1	Plus court chemin dynamique	116
5.4.2	Algorithme de restriction	119
5.5	Résultats numériques	122
5.5.1	Algorithme des proportions successives	128

6	Modèle d'équilibre dynamique avec capacités sur les arcs	130
6.1	Problème de chargement dynamique avec capacités	131
6.2	Formulation mathématique	132
6.2.1	Notation	134
6.2.2	Condition FIFO	135
6.2.3	Propagation du flot	136
6.2.4	Conservation du flot	137
6.2.5	Noeud simple	138
6.2.6	Noeud général	142
6.2.7	La file d'attente	147
6.3	Algorithme de chargement dynamique avec capacités	148
6.4	Résultats numériques	151
	Conclusion	158
	Appendice	161
	Bibliographie	164

Liste des tableaux

4.1	Sommaire comparatif de Algorithme-1, Algorithme-2 et Algorithme-3.	82
4.2	Réseaux et temps CPU.	83
4.3	Paramètres utilisés dans l'exemple 1.	85
5.1	Sioux Falls : flot de départ initial pour tous les chemins.	98
5.2	Sioux Falls : temps de parcours initial ; paire O-D 1 ; chemins 1 – 12.	100
5.3	Sioux Falls : temps de parcours initial ; paire O-D 6 ; chemins 77 – 92.	101
5.4	Sioux Falls : taux de départ final ; paire O-D 1 ; chemins : 1 – 12.	102
5.5	Sioux Falls : temps de parcours final ; paire O-D 1 ; chemins 1 – 12.	102
5.6	Sioux Falls : taux de départ final ; paire O-D 6 ; chemins : 77 – 92.	104
5.7	Sioux Falls : temps de parcours final ; paire O-D 6 ; chemins 77 – 92.	105
5.8	Sioux Falls : ($T=120$).	107
5.9	Réseau de Hull : ($T=30$).	109
5.10	Réseau de Hull : ($T=60$).	110
5.11	Réseau de Hull : taux de départ final, paire O-D 17 ($T=60$).	111
5.12	Réseau de Hull : taux de départ final, paire O-D 17 ($T=60$).	111
5.13	Réseau de Hull, arcs qui forment les chemins minimaux.	112
5.14	Réseau de Hull : méthode de restriction.	122
5.15	Réseau de Hull : méthode de restriction.	123
5.16	Réseau de Hull : méthode de restriction.	124
5.17	Réseau de Hull : méthode de restriction.	124
5.18	Réseau de Hull : méthode de restriction.	125

5.19 Réseau de Hull : méthode de restriction.	125
5.20 Réseau de Hull.	125
5.21 Réseau de Hull : taux de départ final ($T=60$).	126
5.22 Réseau de Hull : temps de parcours final ($T=60$).	126
5.23 Réseau de Hull, arcs qui forment les chemins minimaux.	127
5.24 Réseau de Hull : méthode des proportions successives : flot sur les chemins.	128
5.25 Réseau de Hull : méthode des proportions successives : temps de parcours.	129
5.26 Réseau de Hull : méthode de proportions successives : flot sur les chemins.	129
5.27 Réseau de Hull : méthodes des proportions successives : temps de parcours.	129
6.1 Exemple 1 : paramètres utilisés.	152
6.2 Exemple 2 : paramètres utilisés.	156
6.3 Exemple 2 : taux de départ final.	156
6.4 Exemple 2 : temps de parcours final.	157
6.5 Exemple 2 : convergence de la méthode de projection.	157

Table des figures

2.1	Réseau élargi espace-temps.	29
4.1	Réseau de Hull : volume accumulé pour chaque arc du chemin 1.	86
5.1	Sioux Falls : flot initial pour tous les chemins.	99
5.2	Sioux Falls : temps de parcours initial ; paire O-D 1 ; chemins 1 – 12.	100
5.3	Sioux Falls : temps de parcours initial ; paire O-D 6 ; chemins 77 – 92	101
5.4	Sioux Falls : taux de départ final ; paire O-D 1 ; chemins : 2,7,9,12).	103
5.5	Sioux Falls : temps de parcours final ; paire O-D 1 ; chemins 1 – 12	103
5.6	Sioux Falls : taux de départ final ; paire O-D 6 ; chemins : 88, 91.	105
5.7	Sioux Falls : temps de parcours final ; paire O-D 6 ; chemins 77 – 92.	106
5.8	Sioux Falls : gap de Fukushima ($T=120$).	108
5.9	Résolution du problème d'équilibre dynamique.	121
6.1	Noeud simple.	138
6.2	Noeud général.	142
6.3	Petit réseau.	151
6.4	Exemple 1 : volume total accumulé sur les arcs du chemin 1.	152
6.5	Exemple 1 : volume total accumulé sur les arcs du chemin 2.	153
6.6	Réseau carré	155

Sommaire

Cette thèse porte sur le *problème d'affectation temporelle d'équilibre* sur un réseau de transport. Elle est motivée par le besoin de prévoir comment les véhicules peuvent utiliser les infrastructures de transport en vue de déterminer les déplacements temporels sur les différentes routes d'un réseau de transport existant ou futur. Ceci est important pour estimer la durée des voyages, la congestion, la pollution, etc, pour les plans de développement des infrastructures de transport, les mesures de contrôle et leur exploitation efficace. Dans ce but, il est nécessaire de construire des modèles qui déterminent comment les voyageurs utilisent les différents chemins disponibles entre une origine et une destination données.

Dans un premier temps, ce problème a été étudié du point de vue *statique* sans tenir compte de la variation temporelle du trafic. Nous disposons aujourd'hui d'une grande variété d'algorithmes qui ont été incorporés ou adaptés à des logiciels de planification de transport. Cependant, il y a plusieurs limitations dans l'application du modèle d'affectation statique alors que de plus en plus de gestionnaires de transport l'utilisent pour des études opérationnelles telles que le système de gestion de trafic, l'estimation de la matrice origine-destination, etc. Pourtant, lors de la planification et du contrôle d'un système de transport il est nécessaire de tenir compte de la variation temporelle du flot sur le réseau. Ce sujet est très important pour la modélisation des réseaux de transport routier qui sont assujettis à une grande congestion.

Nous pouvons définir le *problème d'affectation dynamique* comme la détermination

du *flot d'équilibre* sur un réseau de transport en tenant compte de la variation à chaque moment, étant donnée une *demande temporelle* de déplacements. Nous trouvons différentes descriptions d'équilibre ; Wardrop (1952) énonce deux principes pour définir la répartition des flots : le principe d'équilibre dit *descriptif* et le principe *normatif*. Le modèle descriptif considère qu'un usager choisit son chemin de façon à minimiser son coût de parcours (comportement égoïste). Par contre, le deuxième principe cherche à minimiser les coûts totaux des voyages. Nous allons énoncer une généralisation de ces deux principes pour le cas dynamique.

Le présent travail porte sur la résolution du problème d'équilibre descriptif dynamique. Nous allons formuler ce problème comme une *inégalité variationnelle* que nous allons résoudre par un *algorithme de projection* inspiré du travail de Fukushima (1989). À chaque itération de l'algorithme de projection, nous allons générer une nouvelle solution qui déterminera les flots temporels d'entrée sur les chemins. Pour trouver le *temps effectif* de parcours des chemins, nous devons résoudre le *problème de chargement du réseau*. La condition *FIFO*, *premier entré - premier sorti*, est nécessaire et imposée ainsi qu'une formulation pour décrire la *propagation* et la *conservation des flots*.

Nous allons formuler les modèles temporels d'affectation dans l'espace des chemins et nous allons utiliser une méthode de *décomposition simpliciale* dans laquelle nous résoudrons le *problème d'inégalité variationnelle* sur un domaine restreint. Pour générer la restriction nous devons résoudre le *problème des plus courts chemins dynamiques*. Nous utiliserons la *méthode de Dijkstra temporelle* adaptée pour le cas dynamique, qui est valide quand la condition FIFO est respectée.

Aussi, nous allons formuler un modèle pour le *problème d'équilibre temporel avec capacités limitées sur les arcs*. Nous utiliserons les concepts de *demande locale* et de *disponibilité locale* pour déterminer la quantité du flot prêt à sortir et à entrer dans chaque arc. Nous allons diviser les temps de parcours sur les arcs en deux parties : la première tient compte du temps passé à parcourir l'arc, basé sur une fonction de parcours ; la deuxième partie tient compte du temps passé dans une file d'attente ; ce temps dépend des disponibilités et des demandes locales des arcs, basé sur un *modèle de noeud*. De cette façon, nous pouvons considérer le *phénomène de débordement vers l'arrière* résultat d'un goulot d'étranglement.

Mots clefs :

Transport

Affectation dynamique d'équilibre

Optimisation

Problème d'inégalité variationnelle

Chargement dynamique de réseau

Chapitre 1

Introduction

Le développement des systèmes intelligents de transport a motivé l'exploration des *modèles d'affectation dynamique* afin d'améliorer les stratégies pour choisir les routes, les moments de départ et les modes de transport. Il est nécessaire de faciliter la planification ainsi que l'exploitation des réseaux de transport pour assurer une gestion efficace en temps réel, toujours en tenant compte de l'incertitude du caractère dynamique des phénomènes en jeu.

Nous allons considérer un réseau de transport urbain représenté par un *graphe* qui consiste en un ensemble non vide de *noeuds* et d'*arcs* orientés. Chaque arc correspond à une paire ordonnée de noeuds qui représentent une extrémité initiale, la queue de l'arc, et une extrémité terminale, la tête de l'arc. Les noeuds représentent les origines, les destinations ou les intersections des arcs ; les arcs représentent les rues d'une zone urbaine, les routes d'une région, etc.

Nous allons considérer l'*ensemble des origines* composé par tous les noeuds où le flot est généré, c'est-à-dire les noeuds de départ, et également l'*ensemble des destinations* formé de

tous les noeuds qui absorbent du flot, c'est-à-dire les noeuds de destination. Nous tiendrons compte de l'ensemble de toutes les paires formées par une origine et une destination. Nous définirons un *chemin* comme une route qui rejoint une origine donnée et une destination ; il peut exister beaucoup de chemins pour chaque paire origine-destination. À chaque arc est associée une *fonction de coût* qui représente le coût du parcours ou le prix à payer pour le traverser. Généralement, ce coût est défini par le temps de parcours, mais nous pourrions aussi tenir compte d'autres facteurs comme le temps et le prix du stationnement, l'essence consommée, l'usure du véhicule, les frais de voyages, etc. Nous allons considérer la fonction de coût comme étant le temps de parcours pour chaque arc, qui dépend de la quantité de véhicules qu'il y a sur l'arc et sur le réseau.

Le problème d'affectation, ou d'équilibre, sur les réseaux consiste à déterminer le nombre de déplacements ayant lieu sur les routes d'un réseau de transport. Pendant plusieurs années, le problème d'équilibre a été étudié du point de vue *statique*, c'est-à-dire sans tenir compte de la variation de l'état des routes à chaque intervalle de temps. Nous pouvons différencier deux modèles basés sur les principes de Wardrop (1956) : le *modèle d'affectation* (ou d'équilibre) dit *descriptif* et le *modèle d'affectation normatif*.

Le modèle d'affectation normatif a pour but de minimiser les coûts totaux du parcours du système lorsque les voyageurs doivent se déplacer de leur origine à leur destination. Le modèle d'affectation descriptif tient compte du comportement des usagers lors de leurs choix de routes pour leurs déplacements. Un usager choisit son chemin de façon à minimiser son coût de parcours. Ainsi, selon Wardrop, nous dirons que les flots d'un réseau satisfont les conditions d'équilibre descriptif si les coûts de parcours (par unité) des chemins utilisés sont

égaux et si ceux des chemins non utilisés leur sont supérieurs ou égaux. Mais l'affectation du trafic selon l'équilibre descriptif n'est pas nécessairement optimale, du point de vue de coût total, pour le système. Par contre, l'équilibre normatif correspond à une utilisation optimale du réseau.

Nous nous intéressons principalement au modèle d'équilibre descriptif. Beaucoup d'auteurs ont proposé différentes méthodes pour résoudre la variante statique de ce problème comme, par exemple, Smith (1979), Dafermos (1980), Florian et Spiess (1982), Guélat (1983), Bertsekas et Gafni (1982), Lawphongpanich et Hearn (1984), Nguyen et Dupuis (1984), Wu, Florian et Marcotte (1993), etc. Mais, au cours d'une journée, la demande n'est pas constante et ces modèles ne reflètent pas les changements. Ces dernières années, les limites des modèles statiques ont motivé l'étude du problème d'affectation en tenant compte des changements des flots produits dans le temps. Nous appelons ceci le *problème d'affectation ou d'équilibre dynamique* sur un réseau de transport.

Le problème d'affectation dynamique consiste à trouver les taux temporels du flot sur les chemins étant donné les taux temporels de départ, de l'origine vers la destination, pour une période de temps fixe.

Nous allons supposer que les voyageurs sont familiers avec le réseau et qu'ils ont de l'information sur son état, ce qui leur permet de prendre des décisions rationnelles sur le parcours à suivre. Ainsi, nous allons analyser les variations survenues d'une journée à l'autre. Plusieurs auteurs parlent de journée moyenne pour décrire l'état d'un réseau. Mais, ici aussi les difficultés sont grandes, parce que de la même façon que les conditions de circulation

changent selon l'heure, l'observation pratique nous montre que les journées ne sont pas non plus homogènes, et que la définition de journée moyenne serait également "assez imparfaite". Par exemple, le trafic n'est pas le même un lundi matin qu'un vendredi matin, et ce, sans parler de samedi ou de dimanche. De ce fait, pourrions-nous parler d'un lundi moyen ? Ou d'un vendredi moyen ? Mais le trafic n'est pas le même en hiver qu'en été, et non plus pendant la période des fêtes de Noël ou des vacances de juillet par exemple. Ainsi, nous pourrions définir beaucoup d'événements qui mettent en question l'utilisation de la journée moyenne : quels sont les effets s'il y a un match de hockey ? Ou de baseball ? Est-ce qu'on devrait parler d'un mardi après-midi, avec match de hockey, durant le mois de janvier avec une tempête de neige de 15 cm, MOYEN ? Et si le match est retransmis à la télévision ?

La journée moyenne est utilisée pour faire référence à une période de temps cyclique. La durée de cette période n'est pas nécessairement de 24 heures, elle pourrait couvrir quatre heures, ou cinq, ou une période déterminée pendant la journée, ou même une semaine (semaine moyenne).

Une autre difficulté pour modéliser le trafic est le comportement des usagers face aux différentes informations sur l'état de la route. Comment les usagers réagissent devant un panneau qui leur indique qu'une artère donnée est congestionnée ? Il y en a qui vont s'y engager quand même, parce qu'ils n'aiment pas changer de route sans préavis, il y en a qui sortiront dès que possible, il y en a qui chercheront une route alternative. Le tempérament diffère selon les usagers, il y en a qui sont stressés, d'autres qui sont plus calmes, il y a ceux qui aiment courir et ceux qui sont pressés par manque de temps. Il y en a pour qui épargner du temps est une priorité, et pour ce faire, ils prendront le chemin le plus court peu

importe si la route est à péage car, le temps, c'est de l'argent. Il y a aussi des usagers qui quittent leur origine à une heure précise, d'autres qui doivent être à leur destination à une heure déterminée et ceux qui n'ont pas de contrainte sur l'heure d'arrivée ou de sortie. Nous pouvons faire beaucoup de classifications des usagers qui vont réagir de façons différentes devant le même fait et la même information. Devrait-on alors considérer ces différences ?

De nouveaux modèles ont été proposés ces dernières années basés sur des concepts et des descriptions différents du flot dynamique. Ainsi, nous trouvons des modèles en temps discret ou continu avec différentes conditions d'équilibre, différentes définitions de temps de parcours d'un chemin, différentes modélisations de la propagation du flot, etc.

Dans cette thèse, nous allons étudier le problème d'équilibre dynamique basé sur une généralisation temporelle du principe descriptif de Wardrop. Ainsi le problème réside dans la recherche des taux temporels du flot sur les chemins qui vérifient les conditions temporelles de Wardrop, que nous allons énoncer plus tard. L'une des plus grandes difficultés pour résoudre ce problème consiste à déterminer les volumes temporels des arcs ainsi que les temps de déplacement sur eux et sur les chemins, étant donné les taux de départs temporels du trafic sur les chemins du réseau. Ceci est connu comme le *problème de chargement dynamique* du réseau ; c'est un sous-problème fondamental que nous devons résoudre plusieurs fois avant de trouver une solution d'équilibre.

Les principales contributions de cette thèse sont les suivantes :

- Nous avons développé et implanté deux nouveaux algorithmes pour résoudre le problème de chargement du réseau dynamique, basés sur un algorithme précédent

appelé DYNALOAD (Wu, Florian, Rubio-Ardanaz 1998c). Ces algorithmes sont formulés en utilisant une fonction de flot de sortie qui calcule la quantité de flot qui sort de chaque arc à chaque moment. Nous allons analyser comment le flot se propage sur les arcs et sur les chemins. Nous devons tenir compte des contraintes de propagation du flot.

Nous avons comparé ces deux nouveaux algorithmes avec des algorithmes existants. Nous avons constaté, sur des exemples numériques, que les nouveaux algorithmes sont beaucoup plus performants que les précédents.

- Nous avons résolu le problème d’affectation dynamique sur un domaine où les chemins sont fixés à l’avance. Nous avons utilisé une méthode itérative de projection. Pour calculer la “direction” de l’algorithme de projection nous avons utilisé, à chaque itération, une méthode, que nous appelons, d’*interpolation unidimensionnelle* basée sur une approche utilisée par Wu (1991) (thèse doctorale), qui est inspirée d’une contribution de Kennington (1980). Les solutions que nous avons trouvées vérifient les conditions d’équilibre que nous allons énoncer (adaptation dynamique de Wardrop).
- Nous avons utilisé une méthode de restriction pour résoudre le problème d’affectation dynamique sur les réseaux sans limiter le domaine réalisable dans l’espace des chemins. Ainsi, nous avons trouvé une solution d’équilibre globale à ce problème. Les chemins ne sont pas fixés à l’avance, mais plutôt générés au fur et à mesure.

- Nous avons développé des formulations mathématiques pour le problème de chargement de réseau avec capacité limitée sur les arcs. Nous avons considéré le phénomène de débordement vers l'arrière ("spill back") produit par un goulot d'étranglement. En utilisant ces formulations et la méthode de restriction nous avons trouvé des solutions d'équilibre pour le problème d'affectation dynamique avec capacité limitée sur les arcs.

Cette thèse est organisée de la façon suivante. Au chapitre deux, nous allons présenter certaines définitions et descriptions de base que nous allons utiliser par la suite et nous ferons une révision de différentes approches analytiques pour résoudre le problème d'équilibre dynamique. Nous allons diviser les modèles proposés en trois grandes catégories : les modèles basés sur des modèles d'optimisation, les modèles basés sur la théorie du contrôle optimal et les modèles basés sur les inégalités variationnelles. Nous allons présenter notre modèle et ses contraintes au chapitre trois. Nous allons ainsi exposer les contraintes d'état, les contraintes de conservation et de propagation du flot ainsi que des contraintes premier entré - premier sorti (FIFO). Au chapitre quatre, nous allons présenter les algorithmes que nous avons développés pour la résolution du problème de chargement. Nous présenterons la formulation mathématique, les détails de l'implantation et des résultats numériques. La résolution du problème d'équilibre dynamique sera exposée au chapitre cinq. Nous allons utiliser une méthode de projection pour arriver à l'équilibre, d'abord sur un domaine fixé à l'avance et puis sur un cas général. Nous allons exposer aussi une méthode pour résoudre le problème des plus courts chemins dynamiques. Dans ce chapitre, nous allons présenter un algorithme simple pour affecter le flot sur un réseau, algorithme de moyennes successives, afin de comparer les résultats obtenus par cet algorithme avec les résultats d'équilibre obtenus par

notre algorithme. Au chapitre six, nous allons présenter un modèle d'équilibre dynamique avec limite de capacité sur les arcs. Pour résoudre ce problème, nous devons reformuler le problème de chargement pour tenir compte de ces limites de capacité. Nous allons introduire dans nos modèles des files d'attente ainsi que le phénomène de débordement vers l'arrière produit par la congestion. Nous allons présenter un algorithme de résolution ainsi qu'un exemple numérique.

Chapitre 2

Revue de la littérature

Dans ce chapitre, nous allons présenter une revue des travaux réalisés pour résoudre le problème d'équilibre dynamique en utilisant des approches analytiques. Avant d'exposer cette revue, nous allons introduire certaines notions et descriptions que nous allons utiliser au long de cette thèse.

2.1 Notation et notions de base

Dans cette section, nous présentons la notation utilisée dans cette thèse suivie d'un énoncé du principe d'équilibre de Wardrop généralisé pour le cas dynamique.

Nous considérons un réseau de transport représenté par un graphe $G = (N, A)$ constitué d'un ensemble fini N de noeuds et d'un ensemble fini A d'arcs reliant certaines paires de noeuds. Nous considérons aussi l'ensemble de toutes les paires formées par une origine et une destination, ensemble des paires (OD) , que nous représentons par l'ensemble I . Chaque paire origine-destination est reliée par un ensemble des chemins K_i . Un chemin k , $k \in K_i$, est une suite d'arcs tels que le noeud terminal de chacun est le noeud initial du suivant. Le noeud initial du premier arc d'un chemin est l'origine, et le noeud terminal du dernier arc

est la destination. L'ensemble $K = \{k : k \in K_i, \forall i \in I\}$ représente l'ensemble de tous les chemins du réseau. Les déplacements sur le réseau se font d'une origine à une destination en utilisant les chemins choisis.

Nous allons considérer le flot de véhicules qui entrent dans le réseau dans l'intervalle de temps $[0, T]$, *période de départ*, où T représente le dernier temps de départ. La période de temps à étudier, *période du système*, sera l'intervalle $[0, T']$, où $T' > T$ doit être assez grand pour que tous les flots qui ont commencé à parcourir leur chemin dans la période de départ arrivent à leur destination au plus tard au temps T' . Nous pouvons considérer T' comme le dernier temps d'arrivée, ou le moment d'arrivée du dernier flot.

À chaque paire origine-destination i est associée une *demande temporelle* non négative $g_i(t)$. Ainsi, les $g_i(t)$ sont les fonctions qui définissent le taux de départ, c'est-à-dire la quantité de flot qui entre dans le réseau par unité de temps correspondant à la paire origine-destination i . Les demandes $g_i(t)$ sont distribuées entre les chemins qui appartiennent à l'ensemble K_i . Nous définissons $h_k(t)$ comme le taux ou flot de départ sur un chemin k au temps t . Le flot de départ $h_k(t)$ satisfait la *conservation de flot* et la *non-négativité*

$$\sum_{k \in K_i} h_k(t) = g_i(t), \forall i \in I, \text{ pour presque tout } t \in [0, T] \quad (2.1)$$

$$h_k(t) \geq 0, \forall k \in K, \text{ pour presque tout } t \in [0, T] \quad (2.2)$$

(presque tout t dans le sens de Lebesgue, Rudin (1976)).

Nous représentons par $h(t)$ le vecteur des taux de départ sur les chemins au moment t .

$$h(t) = \{h_k(t) : h_k(t) \geq 0, \forall k \in K_i, \forall i \in I\}, \text{ pour presque tout } t \in [0, T], \quad (2.3)$$

Nous allons définir la région réalisable $\Omega(t)$ comme suit :

$$\Omega(t) = \left\{ h(t) : \sum_{k \in K_i} h_k(t) = g_i(t), h_k(t) \geq 0, \forall k \in K_i, \forall i \in I \right\}, \text{ pour presque tout } t \in [0, T]. \quad (2.4)$$

Nous définissons également $b_a(t)$ comme le flot de véhicules qui entrent dans l'arc a au temps t . À chaque arc, a , est associé un coût (unitaire) temporel de parcours dépendant de la quantité de flots qui se trouvent dans le réseau à chaque instant. Nous allons représenter par $s_a(v(t))$ le coût pour traverser l'arc a au moment t et par $v(t)$ le vecteur des quantités de flot traversant les arcs du réseau au moment t ; $v(t) = \{v_a(t), \forall a \in A\}$ où $v_a(t)$ est la quantité de flot sur l'arc a au moment t .

Nous considérons qu'un chemin k est utilisé au temps t si le flot d'entrée sur le chemin, au temps t , est plus grand que zéro, $h_k(t) > 0$. De la même façon, nous considérons qu'un arc a est utilisé au temps t si le flot qui entre dans l'arc au temps t , $b_a(t)$, est plus grand que zéro, $b_a(t) > 0$. Le flot d'entrée dans les arcs est restreint et doit vérifier certaines conditions que nous allons décrire au chapitre suivant.

2.2 Formulation du modèle d'équilibre dynamique

Le problème d'affectation d'équilibre dynamique consiste à trouver les taux temporels des flots sur les chemins étant donné les taux temporels de départ de l'origine vers la destination pour une période de temps fixe.

Nous pouvons généraliser les principes de Wardrop afin de les adapter à ce problème d'affectation dynamique. Mais il faut tenir compte du fait que cette généralisation n'est

pas unique. Nous ferons référence au modèle d'*affectation dynamique normatif* (ADN) si le coût total de transport de tout le système, sur toute la période, est minimisé. Nous parlerons du modèle d'*affectation dynamique descriptif* (ADD) si à chaque instant, pour chaque origine-destination, le coût, par unité du flot de chaque chemin utilisé, est le même et est égal ou plus petit que celui qui pourrait être expérimenté sur n'importe quelle autre route non utilisée. C'est-à-dire, en considérant une période de temps $[0, T]$, nous chercherons un équilibre à chaque instant t où le coût pour chaque usager est minimisé. Selon la définition du "coût du chemin", on peut formuler des modèles différents.

Pour le modèle statique, le temps de parcours (coût) d'un chemin a été calculé comme étant la somme des temps (coût) qu'un usager utiliserait pour traverser tous les arcs qui appartiennent au chemin. Dans le modèle dynamique, par contre, nous trouvons deux définitions différentes pour le coût d'un chemin : le *temps instantané*, qui fait référence au temps de parcours des arcs au même moment, et le *temps effectif*, qui fait référence aux moments (futurs) estimés d'arrivée à chaque arc.

2.2.1 Temps de parcours instantané d'un chemin ("Modèle réactif").

Le temps de parcours instantané d'un chemin, pour simplifier l'exposition, nous dirons *temps instantané*, est le coût d'un chemin calculé à l'instant t , lequel est calculé comme la somme des temps de parcours de tous les arcs qui font partie du chemin selon les conditions dans lesquelles ils se trouvent à l'instant t .

Soit $\Phi_k(t)$, le temps de parcours instantané sur le chemin k du flot qui entre sur le chemin à l'instant t , et $s_a(v(t))$ le coût de parcours d'un arc, dépendant de la quantité de

flot qui se trouve dans le réseau à l'instant t , $v(t)$. Alors

$$\Phi_k(t) = \sum_{a \in k} s_a(v(t)), \quad \forall k \in K_i, i \in I. \quad (2.5)$$

Nous allons définir les conditions d'équilibre dynamique basées sur le temps instantané et sur l'espace des chemins. Ce modèle est aussi appelé comme étant *réactif*.

Définition 1 (équilibre dynamique descriptif instantané sur les chemins) *Une affectation dynamique du flot sur un réseau de transport est dite en équilibre descriptif instantané sur les chemins si, pour chaque paire origine-destination, à chaque instant, le temps de voyage instantané de toutes les routes utilisées est le même et est minimal.*

Nous allons développer une formulation mathématique de cette définition. Soit $\nu_i(t)$ le *temps instantané de voyage minimal* des chemins qui relient la paire origine-destination i , au moment t . Par définition $\Phi_k(t)$ est plus grand ou égal à $\nu_i(t)$ et $h_k(t) \geq 0$, $\forall k \in K_i$, $i \in I$, pour presque tout $t \in [0, T]$.

Une route k est ou bien utilisée ou bien elle ne l'est pas. Considérons les deux cas selon la définition 1 :

- si une route k est utilisée à l'instant t , c'est-à-dire $h_k(t) > 0$, le temps de voyage $\Phi_k(t)$ est égal au temps instantané de voyage minimal $\nu_i(t)$, donc $\Phi_k(t) = \nu_i(t)$.

Alors :

$$(\Phi_k(t) - \nu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]; \quad (2.6)$$

- si une route k n'est pas utilisée à l'instant t , nous avons $h_k(t) = 0$, donc :

$$(\Phi_k(t) - \nu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]. \quad (2.7)$$

Ainsi, la définition d'équilibre dynamique instantané sur les chemins peut être formulée dans l'espace du flot sur les chemins comme suit :

$$(\Phi_k(t) - \nu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.8)$$

$$\Phi_k(t) - \nu_i(t) \geq 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.9)$$

$$h_k(t) \geq 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.10)$$

Ces conditions d'équilibre sont équivalentes à :

$$\Phi_k(t) > \nu_i(t) \implies h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.11)$$

$$h_k(t) > 0 \implies \Phi_k(t) = \nu_i(t), \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]. \quad (2.12)$$

Nous supposons que chaque usager a toutes les informations sur l'état du réseau, qu'il pourrait obtenir, par exemple, par la radio. Les informations que celle-ci nous donne sont sur l'état du réseau au moment même où nous les écoutons, donc nous disposons d'une information "instantanée". Mais un désavantage du temps instantané est que dans la réalité la situation change continuellement, de telle façon qu'à notre arrivée à un point déterminé de la route, nous pouvons trouver que les conditions sont tout à fait différentes. Il peut arriver que beaucoup d'utilisateurs partent au même moment d'une même origine vers une même destination, donc avec le même temps instantané, et qu'ils aient des temps de voyage réels totalement différents à cause du changement continu des conditions sur le réseau.

Les conditions d'équilibre descriptif instantané peuvent être définies aussi dans l'espace du flot sur les arcs comme suit :

Définition 2 (équilibre dynamique descriptif instantané sur les arcs) *Une affectation dynamique du flot sur un réseau de transport est dite en équilibre descriptif instantané sur les arcs si, pour chaque paire origine-destination, à chaque noeud et à chaque instant, le temps de voyage instantané (du noeud à la destination) pour toutes les routes utilisées est le même et est minimal.*

Il faut noter que cette définition se différencie de la définition 1 car nous considérons le temps de voyage instantané de chaque noeud à la destination, tandis que dans la définition 1 nous considérons le temps de voyage instantané des chemins entiers, c'est-à-dire à partir de l'origine jusqu'à la destination. Nous allons donner une formulation mathématique de cette définition.

La définition 2 définit ce que nous appelons "équilibre sur l'espace du flot sur les arcs", même si nous n'avons pas fait mention explicite des arcs, mais plutôt des noeuds, dans cette définition. Nous allons voir dans la formulation mathématique que cette définition est bien basée sur les arcs. Nous avons défini un arc comme une paire ordonnée de noeuds. Nous allons voir dans le développement mathématique que, selon la définition 2, le coût pour traverser un arc qui est utilisé est égal à la différence entre les coûts instantanés minimaux, au temps t , des noeuds qui lui sont adjacents. Et les coûts des arcs qui ne sont pas utilisés sont plus grands ou égaux à cette différence. Remarquez aussi que dans une définition de flot sur les arcs nous devrions dire plutôt espace arc-commodité, puisque nous devons tenir compte de chaque paire origine-destination. Nous avons préféré laisser l'expression "espace du flot sur les arcs" par uniformité avec les autres auteurs.

Soit i une paire origine-destination (origine o_i , destination d_i), et G_i le sous-graphe de G formé par tous les chemins qui relient o_i à d_i . Supposons que p et q sont deux noeuds reliés par l'arc a et $a = (p, q) \in G_i$. Soit $b_a^i(t)$ le taux du flot qui entre dans l'arc a au temps t en se dirigeant vers la destination d_i ; soient $\nu_p^i(t)$ et $\nu_q^i(t)$ les coûts instantanés de voyage minimaux, au temps t , qui relient respectivement p et q à la destination d_i dans le réseau en équilibre, alors :

$$\nu_q^i(t) + s_a(v(t)) \geq \nu_p^i(t), \quad \forall a \in G_i, \quad i \in I. \quad (2.13)$$

S'il y a un flot qui part du noeud p vers la destination d_i et qui entre dans l'arc a au temps t , c'est-à-dire si $b_a^i(t) > 0$, la définition de l'équilibre dans l'espace des arcs implique que le lien a est dans une route qui relie p à d_i , laquelle a un temps de voyage instantané minimal; c'est-à-dire que le temps de voyage instantané minimal $\nu_p^i(t)$ pour tous les flots qui partent du noeud p vers la destination d_i au temps t doit être égal à la somme de $\nu_q^i(t) + s_a(v(t))$:

$$\nu_q^i(t) + s_a(v(t)) = \nu_p^i(t), \quad \text{si } b_a^i(t) > 0, \quad t \in [0, T'] \quad (2.14)$$

pour tout $a = (p, q) \in G_i$, et pour tout $i \in I$. Nous pouvons conclure que :

$$(\nu_q^i(t) + s_a(v(t)) - \nu_p^i(t))b_a^i(t) = 0, \quad (2.15)$$

$$\nu_q^i(t) + s_a(v(t)) - \nu_p^i(t) \geq 0, \quad (2.16)$$

$$b_a^i(t) \geq 0. \quad (2.17)$$

$\forall a = (p, q) \in G_i, \quad i \in I$, pour presque tout $t \in [0, T']$.

Ces conditions peuvent s'écrire :

$$s_a(v(t)) > \nu_p^i(t) - \nu_q^i(t) \implies b_a^i(t) = 0, \quad (2.18)$$

$$b_a^i(t) > 0 \implies s_a(v(t)) = \nu_p^i(t) - \nu_q^i(t), \quad (2.19)$$

$\forall a = (p, q) \in G_i, i \in I$, pour presque tout $t \in [0, T']$.

Il faut noter que, s'il existe une autre paire origine-destination i' (ou plusieurs) avec la même destination que la paire i , $d_i = d_{i'}$, alors il s'ensuit que $\nu_p^i(t) = \nu_p^{i'}(t)$, $\nu_q^i(t) = \nu_q^{i'}(t)$ et $b_a^i(t) = b_a^{i'}(t)$ pour tout $t \in [0, T]$. Nous pouvons écrire les équations (2.13) - (2.19) en regroupant les paires origine-destination qui ont la même destination.

Un autre inconvénient du temps instantané est que les usagers qui ont un moment de départ t , assez près de T , n'auront pas le temps de finir leur voyage dans l'intervalle du temps à étudier, $[0, T]$. Nous signalons une incohérence entre la définition que nous avons donnée de chemin et arc "occupé" pour le cas instantané, puisqu'un chemin pourrait être occupé au moment t et avoir un ou plusieurs de ses arcs qui ne le sont pas, et vice versa, un arc peut être occupé à un moment donné pendant que le chemin suivi par le flot qui le traverse ne l'est pas.

2.2.2 Temps de parcours effectif d'un chemin. ("Modèle préventif")

Une alternative plus réaliste que le modèle réactif et de l'utilisation du temps instantané est le temps de parcours effectif d'un chemin, pour simplifier, *temps effectif*. Il est déterminé selon le temps de parcours de chaque arc calculé au moment d'arrivée du flot. Le temps de parcours effectif d'un chemin est calculé comme la somme de temps de parcours de tous les arcs qui font partie du chemin selon les conditions dans lesquelles ils se trouvent au moment où le flot arrive à chacun d'eux. Nous définissons le temps effectif pour un flot qui entre sur le chemin k au moment t , $S_k(t)$, de la façon suivante :

$$S_k(t) = \sum_{a \in k} s_a(v(t + S_{ka}(t))), \quad \forall k \in K_i, i \in I, \quad (2.20)$$

où $S_{ka}(t)$ est le temps effectif de voyage passé par un voyageur qui se déplace par le chemin k de l'origine jusqu'à la fin de l'arc a , avec un temps de départ t .

Nous pouvons calculer $S_k(t)$ récursivement :

$$\begin{cases} t_0 & = t, \\ t_j & = \tau_{a_j}(t_{j-1}) = t_{j-1} + s_{a_j}(v(t_{j-1})), \quad j = 1, 2, \dots, n_k, \\ S_k(t) & = t_{n_k} - t = \tau_{a_{n_k}}(\dots\tau_{a_2}(\tau_{a_1}(t))\dots) - t. \end{cases} \quad (2.21)$$

pour les moments de départ $t \in [0, T]$, pour tous les chemins $k \in K_i$, $i \in I$, où n_k indique le dernier arc du chemin k et $\tau_a(t)$ le moment de sortie de l'arc a du flot qui entre dans a au moment t . Il faut remarquer que le flot qui entre dans l'arc a au moment t sort de a au moment $\tau_a(t) = t + s_a(v(t))$, et que le flot qui entre dans le chemin k au moment t sort de a au moment $\Pi_{ka}(t) = t + S_{ka}(t)$.

Comme avant, nous allons supposer que chaque usager possède l'information complète sur l'état du réseau durant toute la période d'analyse. Nous définissons aussi les conditions d'équilibre dynamique descriptif effectif basé sur l'espace du flot sur les chemins. Le modèle est appelé comme étant *préventif*.

Définition 3 (équilibre dynamique descriptif effectif sur les chemins) *Une affectation dynamique du flot sur un réseau de transport est dite en équilibre descriptif effectif sur les chemins si, pour chaque paire origine-destination, à chaque instant, le temps de voyage effectif de toutes les routes utilisées est le même et est minimal.*

Comme pour le modèle réactif nous allons développer une formulation mathématique équivalente à cette définition.

Soit $\mu_i(t)$, le *temps effectif de voyage minimal* des chemins qui relient la paire origine-destination i , au moment t . Par définition $S_k(t)$ est plus grand ou égal que $\mu_i(t)$ et $h_k(t) \geq 0$, $\forall k \in K_i, i \in I$, pour presque tout $t \in [0, T]$.

Selon la définition 3 :

- si une route k est utilisée à l'instant t , c'est-à-dire $h_k(t) > 0$ le temps de voyage effectif $S_k(t)$ est égal au temps effectif de voyage minimal $\mu_i(t)$, donc $S_k(t) = \mu_i(t)$, alors :

$$(S_k(t) - \mu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]; \quad (2.22)$$

- si une route k n'est pas utilisée à l'instant t , c'est-à-dire $h_k(t) = 0$ nous pouvons affirmer aussi que :

$$(S_k(t) - \mu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.23)$$

Alors, nous pouvons formuler les conditions d'équilibre dynamique descriptif effectif sur les chemins comme suit :

$$(S_k(t) - \mu_i(t))h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.24)$$

$$S_k(t) - \mu_i(t) \geq 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]. \quad (2.25)$$

$$h_k(t) \geq 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (2.26)$$

Ces conditions d'équilibre sont équivalentes à :

$$S_k(t) > \mu_i(t) \implies h_k(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]. \quad (2.27)$$

$$h_k(t) > 0 \implies S_k(t) = \mu_i(t), \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T]. \quad (2.28)$$

Les conditions d'équilibre descriptif effectif peuvent être interprétées aussi dans l'espace du flot sur les arcs. Nous pouvons alors redéfinir l'équilibre comme suit :

Définition 4 (équilibre dynamique descriptif effectif sur les arcs) *Une affectation dynamique du flot sur un réseau de transport est dite en équilibre descriptif effectif sur les arcs si, pour chaque paire origine-destination, à chaque noeud et à chaque instant, le temps de voyage effectif expérimenté par les voyageurs qui ont le même temps de départ est le même et il est minimal.*

Nous allons présenter maintenant la formulation mathématique équivalente à cette définition en utilisant les arcs. Supposons, comme dans le cas réactif, que i est la paire origine-destination qui relie o_i à d_i , et G_i le sous-graphe généré par tous les chemins qui les relient ; $a = (p, q) \in G_i$. Soit $\mu_p^i(t)$ et $\mu_q^i(t)$, les coûts effectifs de voyage minimal entre l'origine o_i et p et q respectivement du flot qui part de o_i au moment t . (Notez que μ_p^i représente le coût effectif de voyage minimal de l'origine o_i au sommet p ; par contre, ν_p^i (défini et utilisé à la page 17) fait référence au coût instantané de voyage minimal du sommet p à la destination d_i .)

$$\mu_p^i(t) + s_a[v(t + \mu_p^i(t))] \geq \mu_q^i(t), \quad \forall a \in G_i, \quad i \in I. \quad (2.29)$$

Nous pouvons faire un raisonnement semblable à celui fait pour le coût instantané, ce qui nous amène à :

$$\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) = \mu_q^i(t), \quad \text{si } b_a^i(t + \mu_p^i(t)) > 0, \quad t \in [0, T'], \quad (2.30)$$

pour tous les arcs $a = (p, q) \in G_i$, et pour toutes les paires origine-destination $i \in I$. Nous pouvons conclure que :

$$[\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) - \mu_q^i(t)]b_a^i(t + \mu_p^i(t)) = 0, \quad (2.31)$$

$$\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) - \mu_q^i(t) \geq 0, \quad (2.32)$$

$$b_a^i(t + \mu_p^i(t)) \geq 0, \quad (2.33)$$

pour presque tout $t \in [0, T']$, $a = (p, q) \in G_i$, et $i \in I$. Ces conditions peuvent s'écrire :

$$s_a(v(t + \mu_p^i(t))) > \mu_q^i(t) - \mu_p^i(t) \implies b_a^i(t + \mu_p^i(t)) = 0, \quad (2.34)$$

$$b_a^i(t + \mu_p^i(t)) > 0 \implies s_a(v(t + \mu_p^i(t))) = \mu_q^i(t) - \mu_p^i(t), \quad (2.35)$$

pour $a \in G_i, i \in I$, pour presque tout $t \in [0, T']$. Parallèlement au cas de coût instantané, nous pouvons noter que s'il existe une autre paire $(OD)_{i'}$ (ou plusieurs) ayant la même origine que la paire i , $o_i = o_{i'}$, alors nous avons $\mu_p^i(t) = \mu_p^{i'}(t)$, $\mu_q^i(t) = \mu_q^{i'}(t)$ et $b_a^i(t) = b_a^{i'}(t)$. Nous pouvons écrire les équations (2.29) - (2.35) par rapport à chaque origine.

2.3 Exposé des contributions dans la littérature

Les deux définitions d'équilibre temporel nous permettent de présenter les différents modèles qui ont été proposés pour le problème d'affectation dynamique. Afin de le faire, nous allons les diviser en trois grandes familles : les modèles de programmation mathématique basés sur la théorie de l'optimisation, les modèles basés sur la théorie du contrôle optimal et les modèles formulés comme des inégalités variationnelles.

La première formulation d'un modèle d'optimisation, pour le problème dynamique normatif, à temps discret, est due à Merchant et Nemhauser (1978), qui proposent un modèle non linéaire et non convexe qui ne peut pas être résolu par les techniques conventionnelles d'optimisation. La propagation du flot entre les arcs est formulée en termes d'une fonction du flot de sortie pour chaque arc. D'autres auteurs ont révisé ce modèle. Ho (1980) et Carey

(1986) analysent les conditions d'optimalité, Carey (1987) et Drissi-Kaïtouni (1990a) transforment le modèle en un problème convexe similaire. Zawack et Thompson (1987), Janson (1991), Drissi-Kaïtouni et Hamed-Benchekroun (1992), Fernández et de Cea (1994) proposent des formulations basées sur des descriptions différentes du flot dynamique.

Plusieurs auteurs ont proposé d'utiliser la théorie du contrôle optimal pour formuler le modèle réactif du problème d'équilibre dynamique en temps continu. Ces méthodes normalement tiennent compte des définitions d'équilibre dynamique instantané basées sur des généralisations du principe de Wardrop. L'objectif de l'application de la théorie de contrôle optimal consiste à déterminer des stratégies de contrôle qui font qu'un processus satisfait aux contraintes physiques et, en même temps, minimise un certain critère de performance. Friesz et al. (1989), Wie et al. (1990), Ran et al. (1993), Boyce et al. (1995) ont proposé des modèles basés sur cette théorie.

Le troisième grand groupe englobe les méthodes basées sur les inégalités variationnelles. Cette approche analytique consiste à énoncer les conditions de flot en équilibre dynamique comme un problème équivalent d'inégalité variationnelle. La structure de ces modèles nous permet de les diviser en deux parties : une partie d'optimisation, qui représente les variations de parcours que les usagers d'un réseau réalisent et qui convergent vers l'équilibre (modèles au jour le jour), et une autre partie qui consiste à déterminer les volumes des flots temporels des arcs une fois que la stratégie est déterminée (chargement de réseau). Cette modélisation nous permet aussi certaines généralisations, telles que l'utilisation de fonctions de coût asymétriques ou le choix d'un moment de départ plus intéressant. Friesz et al. (1993), Wu (1994), Wu et al. (1995), Wie et al. (1995), sont des exemples de l'utilisation de ces

méthodes. Dans les sections suivantes nous présentons ces contributions avec plus de détails.

2.4 Modèles basés sur la programmation mathématique

2.4.1 Merchant et Nemhauser : modèles discrets d'équilibre dynamique normatifs

Merchant et Nemhauser (1978 a,b) proposent un modèle discret pour représenter le problème d'affectation normatif instantané formulé comme un problème d'optimisation non linéaire et non convexe, et pour un réseau où plusieurs origines sont permises, mais avec une seule destination. Pour décrire la propagation du flot, ils utilisent une fonction du *flot de sortie* pour chaque arc, $e_a(v_a)$, telle que décrite à la section 3.5. L'intervalle de temps est fini $[0, T']$ et discrétisé en M petits sous-intervalles.

Les fonctions de coût et du flot de sortie doivent vérifier les hypothèses suivantes :

- La fonction du flot de sortie sur l'arc a dépend de la quantité de flot qui se trouve sur l'arc au début du sous-intervalle, $e_a(v_a(t))$.
- La fonction $e_a(v_a)$ est non décroissante, continue, concave et $0 \leq e_a(v_a) \leq v_a$.
- La fonction $e_a(v_a)$ vérifie une propriété de *saturation* : $\lim_{\omega \rightarrow \infty} \frac{de_a(\omega)}{d\omega} = 0$.
- La fonction de coût s_a dépend seulement du nombre de flots sur l'arc au début du sous-intervalle, $v_a(m)$. Elle est non négative, non décroissante, convexe et continue.
- La dérivée $\frac{ds_a(v_a(m))}{dv_a}$ existe pour tout v_a non négatif; et pour toute paire d'arcs (a_i, a_j) qui font partie d'un même chemin qui relie un noeud donné à la destination d de tel sorte que a_j est plus proche de d que a_i . La fonction de coût doit vérifier l'inégalité suivante :

$$\sup_{v_a} \left\{ \frac{ds_{a_j}(v_a(m))}{dv_a} \right\} \leq \inf_{v_a} \left\{ \frac{ds_{a_i}(v_a(m))}{dv_a} \right\}.$$

Cette dernière condition est très restrictive et irréaliste. Le coût sur les arcs doit croître avec la distance à la destination.

Le modèle proposé par Merchant et Nemhauser est le suivant :

$$\min \sum_{m=1}^M \sum_{a \in A} s_a(v_a(m)) \quad (2.36)$$

s. à.

$$v_a(m+1) = v_a(m) - e_a(v_a(m)) + b_a(m), \quad \forall a \in A, \forall m \quad (2.37)$$

$$\sum_{a \in n^+} b_a(m) = h_n(m) + \sum_{a \in n^-} e_a(v_a(m)), \quad \forall n \neq d \in N, \forall m \quad (2.38)$$

$$b_a(m) \geq 0, \quad \forall a \in A, \forall m \quad (2.39)$$

$$v_a(m) \geq 0, \quad \forall a \in A, \forall m \quad (2.40)$$

($v_a(0)$ est donné), d représente la destination et h_n le flot généré au sommet n et n^+ l'ensemble des arcs qui sortent du noeud n , arcs avec queue égale à n . Le premier groupe de contraintes (2.37) sont les contraintes d'état, lesquelles incluent la propagation du flot (fonction du flot de sortie). Le deuxième (2.38) représente les contraintes de conservation du flot.

Si la fonction $e_a(v_a)$ est non linéaire, l'ensemble de solutions réalisables, en général, est non convexe. La contrainte (2.40) est redondante, puisque $0 \leq e_a(v_a) \leq v_a$ et $b_a(m) \geq 0$ impliquent, par la contrainte, (2.38), $v_a(m+1) \geq 0$.

Ils proposent un algorithme basé sur une linéarisation par morceau, mais la non-convexité ajoute encore des difficultés. Ils montrent que les solutions de l'algorithme vérifient les conditions d'optimalité de Kuhn-Tucker, mais ces conditions restent seulement

nécessaires. Les conditions d'optimalité de Kuhn-Tucker sont suffisantes si au moins une, parmi certaines qualifications de contraintes, est vérifiée (Bazaraa et al. 1993). Mais cette vérification n'a pas été démontrée par Merchant et Nemhauser.

Le modèle de Merchant et Nemhauser, comme nous l'avons déjà dit, a été la première formulation de programmation mathématique pour le problème d'équilibre dynamique et est une référence incontournable et une source d'inspiration pour beaucoup d'autres travaux, mais nous pouvons signaler plusieurs limitations :

- Ils formulent seulement le problème normatif, pas le descriptif.
- Ils considèrent seulement une destination.
- Les conditions imposées sont très restrictives et ne sont pas très réalistes.
- Ils utilisent une fonction du flot de sortie, cette façon de décrire la propagation du flot génère certaines déficiences. (voir section 3.5).
- Le modèle est non convexe, difficile à résoudre et il est aussi difficile de démontrer les conditions d'optimalité.

Ho (1980) et Carey (1986) analysent aussi cette méthode. Carey ajoute les conditions suivantes :

- $e_a(v_a)$ continûment différentiable pour $v_a > 0$ et
- $s_a > 0$ à la place de $s_a \geq 0$. Cette condition pourrait être vérifiée avec une petite perturbation sur S_k .

Ils montrent que, sous ces conditions, une des qualifications de contraintes requises, l'indépendance linéaire des contraintes saturées, est satisfaite pour le modèle de Merchant

et Nemhauser, ce qui donne plus d'intérêt aux résultats obtenus.

À cause des difficultés dues à la non-convexité du modèle, Carey (1987) reformule un modèle similaire, mais convexe, et qui, d'une part, peut être résolu plus facilement avec l'utilisation d'un algorithme développé pour la programmation convexe ; et, d'autre part, les conditions d'optimalité de Kuhn-Tucker sont nécessaires et suffisantes pour caractériser la solution optimale. Il généralise le modèle pour plusieurs destinations et commodités.

Les deux différences par rapport au modèle de Merchant et Nemhauser sont :

- Le flot de sortie des arcs $e_a(t)$ apparaît de façon explicite dans la contrainte (2.37).

Il remplace la fonction du flot de sortie, comme suit :

$$v_a(m+1) = v_a(m) - e_a(m) + b_a(m), \quad a \in A, \quad m = 0, \dots, M-1.$$

- Le flot de sortie des arcs est non négatif et est limité par la fonction de sortie,

$$0 \leq e_a(m) \leq e_a(v_a(m)) (\leq v_a(m)), \quad a \in A, \quad m = 0, \dots, M-1.$$

Carey appelle $\sigma_a(m) = e_a(v_a(m)) - e_a(m)$ contrôle du flot. Il montre qu'à l'optimum $\sigma_a(m)$ est égal à zéro pour tout a et pour tout m . Mais quand le modèle est généralisé pour un réseau ayant plusieurs destinations ou plusieurs commodités, la propriété FIFO peut être violée. Carey (1992) formule plusieurs classes de contraintes, non convexes, pour assurer le FIFO.

Drissi-Kaïtouni (1992) présente un modèle semblable à celui de Carey (1986), mais pour l'équilibre descriptif avec des conditions plus réalistes. Il propose un algorithme heuristique basé sur l'hypothèse que les usagers connaissent à chaque instant les temps effectifs

de voyage sur les liens du réseau et s'engagent sur les chemins les plus courts les conduisant à leur destination. Ainsi, à chaque noeud du réseau, les usagers ont la possibilité de modifier leur chemin. C'est un algorithme plus facile à mettre en oeuvre, mais rien ne nous assure que les résultats seront acceptables, parce que les chemins les plus courts sont calculés selon le coût instantané avant qu'une décision soit prise, alors que le coût effectif du chemin choisi pourrait bien être différent. La condition de FIFO n'est pas assurée non plus.

2.4.2 Modèles discrets d'équilibre dynamique basés sur des réseaux espace-temps

Les modèles que nous allons présenter dans cette sous-section ont été conçus pour résoudre le problème d'affectation dynamique en temps discret et sont basés sur des modèles proposés pour le problème du flot maximal dynamique par Ford et Fulkerson (1962) et Fulkerson (1975). Dans ces modèles, chaque arc a a une capacité d'entrée limitée et il faut un temps fixe pour le traverser. À chaque période et à chaque noeud, les véhicules peuvent sortir de l'arc et continuer leur voyage ou attendre la prochaine période dans une file d'attente. Le temps d'attente est en fonction des véhicules sur l'arc.

Ils construisent un réseau élargi *espace-temps* et résolvent le problème comme un modèle statique.

Le temps du système, $[0, T']$, est divisé en M intervalles de la même longueur. Nous déterminons Δt comme la longueur de chaque intervalle.

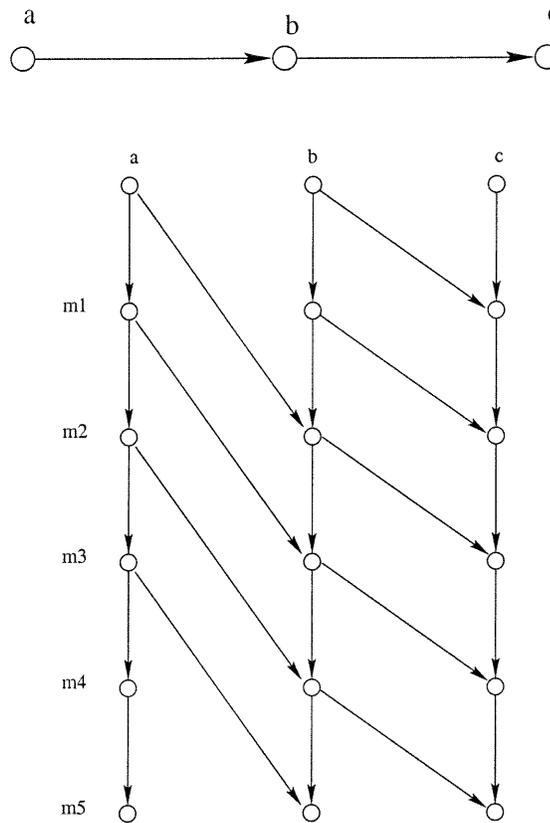


FIG. 2.1 – Réseau élargi espace-temps.

Le réseau élargi espace-temps est construit comme suit :

- chaque noeud $n \in N$ est étendu en $(M + 1)$ noeuds, n_m , $m = 0, 1, \dots, M$;
- arcs pour le flot de sortie de chaque noeud : chaque arc $a = (i, j) \in A$ pour chaque intervalle m , est étendu en arc $(i_m, j_{m+|s_a|})$ où $|s_a|$ est le nombre de périodes nécessaires pour traverser l'arc a , $m = 0, 1, \dots, M - |s_a|$;
- arcs pour les véhicules qui attendent à l'intervalle suivant, file d'attente, pour chaque noeud : arcs (n_m, n_{m+1}) qui représentent le temps d'attente où $n \in N$ et $m = 0, 1, \dots, M - 1$;

Le temps fixe pour traverser un arc de sortie est multiple de Δt , $s_a = |s_a| * |\Delta t| \forall a =$

$(i_m, j_{m+|s_a|})$, le temps pour traverser un arc d'attente est égal à Δt , $m = 0, 1, \dots, M - |s_a|$.

Les arcs pour le flot de sortie ont une capacité limitée, mais il n'y a pas de limite de capacité pour les arcs qui représentent les files d'attente.

Zawack et Thompson (1987) ont adapté ce modèle pour le problème d'affectation dynamique normatif avec une seule destination ; ils ont identifié des arcs et des sommets du réseau espace-temps qui ne sont pas nécessaires ; ils ajoutent une contrainte de capacité sur les files d'attente, mais puisque les files d'attente sont supportées sur les noeuds, tous les arcs qui ont la même tête vont partager la même file ; donc il continue d'y avoir une incohérence par rapport aux capacités sur les arcs. Ils minimisent le coût (temps) total de transport de tout le système en formulant le problème d'affectation dynamique normatif comme un problème de réseau à coût minimum avec capacité sur les arcs et ils utilisent deux méthodes (heuristiques) de résolution, une basée sur le flot à coût minimum sur un réseau et l'autre sur le plus court chemin.

Drissi-Kaïtouni (1990b) et Drissi-Kaïtouni et Hamed-Benchekroun (1992) présentent un modèle pour le problème d'affectation dynamique descriptif à destinations multiples, formulé comme un problème d'inégalité variationnelle à une seule dimension.

Ils utilisent un réseau espace-temps dans le même esprit que les précédents où ils incorporent la file d'attente sur les arcs. Ainsi chaque noeud $n \in N$ est étendu aussi en $(M + 1)$ noeuds et pour chaque arc $a = (i, j) \in A$ ils définissent :

- un noeud l_m^a ;

- un arc (i_{m-s_a}, l_m^a) ;
- un arc (l_m^a, j_m) ;
- un arc (l_m^a, l_{m+1}^a) ;

Ces trois arcs correspondent aux variables $b_a(m - s_a)$, $v_a(m)$ et à la file d'attente $q_a(m + 1)$, respectivement.

Ainsi les files d'attente sont incorporées sur les arcs. Alors, à chaque période, le nombre de véhicules qui peuvent entrer dans un arc ajouté aux véhicules qui se trouvent déjà dans la file d'attente de cet arc, de la période antérieure, est borné par la capacité totale de l'arc en question.

Drissi-Kaïtouni et Drissi-Kaïtouni et Hamed-Bencheikroun transforment donc le problème d'affectation dynamique en un problème d'affectation statique sur le réseau élargi espace-temps et ils utilisent l'algorithme, classique pour ce problème, d'approximation linéaire de Frank et Wolfe (1956) pour résoudre ce modèle.

Un grand inconvénient de ce modèle est que toutes les périodes doivent être de la même longueur. La condition de FIFO n'est pas nécessairement vérifiée. Ils tiennent compte explicitement de la limite de capacité d'entrée sur les arcs, mais la capacité de sortie est bornée implicitement. Pour les arcs qui représentent les déplacements du flot, cette capacité de sortie est égale à la capacité d'entrée, mais pour les arcs qui représentent les files d'attente, la limite de capacité de sortie est égale à la somme de capacités d'entrées des arcs suivants, arcs a^+ . Il faudrait considérer aussi que le réseau espace-temps est très grand et qu'il requiert beaucoup de mémoire.

2.5 Modèles de contrôle optimal

Nous allons décrire les modèles d'affectation dynamique formulés comme un problème continu de contrôle optimal avec demandes fixes. Ces modèles peuvent être considérés comme une extension continue du modèle de Merchant et Nemhauser. La formulation de la théorie de contrôle optimal est associée à des processus temporels. Le système dépend de certaines variables : variables de contrôle, qui vont "contrôler" l'état dans lequel se trouve le système, et des variables d'état qui varient en fonction des variables de contrôle. Les modèles proposés visent à vérifier les conditions d'équilibre avec le temps de parcours instantané sur les chemins, donc ils utilisent le principe réactif.

La période à étudier, $[0, T]$, est fixe et continue. Les contraintes pourront être regroupées par paires origine-destination. Les variables d'état sont les $v_{ka}(t)$ et les variables de contrôle les $b_{ka}(t)$, parfois aussi les $e_{ka}(t)$ et $f_k(t)$. La variable $f_k(t)$ représente le flot d'arrivée à la destination. L'équation d'état est de la forme suivante :

$$\frac{dv_{ka}(t)}{dt} = b_{ka}(t) - e_{ka}(t). \quad (2.41)$$

Le coût de parcours d'un arc est limité à être fonction des variables de l'arc même (flot de sortie ou d'entrée ou de véhicules sur l'arc) sans aucune influence des arcs voisins. Alors les modèles considérés sont à demande fixe, séparables, diagonaux et à un seul mode.

2.5.1 Modèles de Friesz et al. : avec fonctions du flot de sortie

La première formulation d'un modèle continu de contrôle optimal est due à Friesz et al. (1989) et est une extension du celui de Merchant et Nemhauser. Ils proposent deux modèles, un normatif et un autre descriptif avec des coûts de chemins instantanés et une seule desti-

nation (connu comme modèle d'équilibre de Boston). Wie et al. (1990) étendent le modèle pour des destinations multiples. Les deux modèles sont semblables. Ils utilisent une fonction du flot de sortie linéaire pour éviter des problèmes de non-convexité, $e_a(v_a(t)) = \xi_a v_a(t)$, où ξ_a est une constante.

Le modèle de contrôle optimal pour le cas descriptif avec destinations multiples est le suivant :

$$\min \sum_{a \in A} \int_0^T \int_0^{v_a} m_a(w_a) dw_a dt \quad (2.42)$$

s. à.

$$\frac{dv_a^d(t)}{dt} = b_a^d(t) - \xi_a v_a^d, \quad \forall a \in A, \quad (2.43)$$

$$g_n^d(t) + \sum_{a \in n^-} \xi_a v_a^d(t) - \sum_{a \in n^-} b_a^d(t) = 0, \quad \forall n \in N, \quad (2.44)$$

$$b_a^d(t) \geq 0, \quad \forall a \in A, \quad (2.45)$$

$\forall d \in D, t \in [0, T]$, où D est l'ensemble des destinations, $g_n^d(t)$ est le flot engendré au noeud n avec destination d au temps t , qui est une fonction continue et non négative sur $t \in [0, T]$; v_a^d est le nombre de véhicules sur a qui se dirigent vers la destination d , b_a^d le flot d'entrée dans a avec destination d ; $m_a(v_a(t))$ doit être intégrable et croissant. Pour le modèle descriptif de Friesz et al. (modèle de Boston) la fonction objectif est : $m_a(v_a(t)) = s_a(v_a(t))e'_a(v_a(t))$, où $e'_a(v_a(t))$ est la dérivée de la fonction du flot de sortie par rapport à la variable d'état $v_a(t)$. Pour le modèle de Wie et al., $m_a(v_a(t)) = \xi_a s_a(v_a(t))$.

Il n'y a pas de contraintes de non-négativité, $v_a(t) \geq 0, a \in A$, parce qu'elles sont redondantes. Les fonctions de coût $s_a(v_a(t)), a \in A$, doivent être positives, croissantes et différentiables. Les variables de contrôle sont les b_a . Friesz et al. montrent que les condi-

tions d'optimalité nécessaires et suffisantes de ce problème de contrôle optimal vérifient la définition d'équilibre de ADD où le coût instantané sur une route, k , est défini comme suit :

$$\Phi_k(t) = \sum_{a \in k} [s_a(v_a(t)) + \frac{\dot{\lambda}(t)}{e'_a(v_a(t))}] \quad (2.46)$$

où $\dot{\lambda}(t)$ est le taux de changement par rapport à t du multiplicateur de Lagrange, $\lambda(t)$, de la contrainte (2.43) .

Les principales limitations que nous avons remarquées pour ces modèles sont :

- Le coût optimisé n'est pas le vrai coût, mais (2.46).
- La fonction de coût sur les chemins est instantanée. Les décisions sont prises selon l'état courant, pas futur, du réseau.
- La fonction de coût sur les arcs doit être symétrique, c'est-à-dire que la fonction de coût sur chaque arc, s_a , dépend du nombre de véhicules qui se trouvent sur celui-ci, v_a .
- Ils utilisent une fonction du flot de sortie, ce qui provoque des déficiences.

Fernández et de Cea (1994) présentent un modèle du même style, mais avec les coûts effectifs sur les chemins, non instantanés, où ils proposent une nouvelle formulation de la fonction du flot de sortie pour corriger le fait que selon cette fonction, une fois que le flot entre sur un arc, il est immédiatement distribué sur celui-ci d'une façon homogène. Pour ce faire, ils définissent une densité moyenne pour les arcs. Ils proposent aussi la condition (3.24) sur les fonctions de coût pour que le FIFO soit respecté (voir section 3.4).

$$\frac{s_a(v(t + \Delta t)) - s_a(v(t))}{\Delta t} + 1 > 0.$$

2.5.2 Modèles de Ran et al. : avec fonctions de temps sur les arcs

Ran, Boyce et LeBlanc (1993) évitent l'utilisation des fonctions du flot de sortie et ils formulent un nouveau modèle de contrôle optimal pour résoudre le ADD avec fonction de coût aussi instantanée, formulé dans l'espace des arcs. La fonction de coût de chaque arc, s_a , dépend du nombre de véhicules qui se trouvent sur celui-ci ainsi que du taux d'entrée et de sortie, on la note $s_a[v_a(t), b_a(t), e_a(t)]$. Elle est égale à la somme de deux composantes : $g_{1a}[v_a(t), b_a(t)]$ qui fait référence au temps de parcours, et $g_{2a}[v_a(t), e_a(t)]$ qui fait référence à un temps d'attente de sortie dans une file d'attente. Le temps instantané de parcours d'un chemin $\Phi_k(t)$ est déterminé comme suit :

$$\Phi_k(t) = \sum_{a \in k} s_a[v_a(t), b_a(t), e_a(t)], \quad \forall k, \quad t \in [0, T]. \quad (2.47)$$

s_a, g_{1a}, g_{2a} sont supposées être non négatives, croissantes et différentiables par rapport à $v_a(t), b_a(t), e_a(t)$ respectivement.

Ils ajoutent les variables $f_k(t)$ et $F_k(t)$ qui sont le taux du flot et la quantité accumulée de véhicules qui sont arrivés à la destination par le chemin k à l'instant t . Les contraintes d'état deviennent

$$\frac{dv_{ka}}{dt} = b_{ka}(t) - e_{ka}(t), \quad \forall a, k, \quad t \in [0, T'] \quad (2.48)$$

et

$$\frac{dF_k(t)}{dt} = f_k(t), \quad \forall k, \quad t \in [0, T']. \quad (2.49)$$

Ils ajoutent aussi une contrainte de conservation du flot à la destination qui requiert que le flot qui arrive à chaque destination d soit égal à la somme de tous les flots qui sortent des arcs entrant dans d , au moment t :

$$\sum_{a \in d^-} e_{ka}(t) = f_k(t), \quad \forall k, d, \quad t \in [0, T']. \quad (2.50)$$

Les variables de contrôle sont b_{ka} , e_{ka} et f_{ka} , et les variables d'état sont v_{ka} et F_k .

Ils proposent une formulation différente pour la contrainte de propagation du flot. Pour la décrire, nous allons d'abord définir le sous-chemin \tilde{k} comme la section du chemin k qui relie q à la destination (q est un noeud intermédiaire sur le chemin k). Soit q^- , l'ensemble des arcs qui entrent dans le noeud q , arcs avec tête égale à q , et soit $a \in q^-$, les véhicules qui se trouvent sur a au temps t , qui ont emprunté le chemin k , vont engendrer au temps $t + s_a(v(t))$ une augmentation du nombre de véhicules sur la route \tilde{k} ou/et une augmentation du nombre de véhicules qui sont arrivés à la destination. Il faut tenir compte aussi que, bien que le modèle soit formulé en fonction du coût instantané, la propagation du flot est faite en tenant compte du coût effectif sur les arcs, puisqu'une propagation du flot en temps instantané n'aurait aucun sens. Soit \tilde{s} , une estimation de $s_a(v(t))$ qui sera mise à jour à chaque itération de l'algorithme de résolution. Ils définissent le nombre de véhicules sur l'arc a qui correspond au chemin k au moment t comme la différence entre le nombre de véhicules qui se trouvent sur \tilde{k} au moment $t + \tilde{s}_a(t)$ et ceux qui se trouvaient sur \tilde{k} au moment t , plus la quantité de véhicules qui sont sortis entre t et $t + \tilde{s}_a(t)$. Nous avons

$$v_{ka}(t) = \sum_{\tilde{a} \in \tilde{k}} \{v_{k\tilde{a}}([t + \tilde{s}_a(t)] - v_{k\tilde{a}}(t))\} + \{F_k[t + \tilde{s}_a(v(t))] - F_k(t)\}. \quad (2.51)$$

Le nombre de véhicules sur l'arc est associé à des variables d'état seulement (pas de variables de contrôle, parce que ceci est nécessaire pour la preuve d'équivalence avec les conditions d'équilibre). Cette contrainte assure la condition de FIFO. Elle a l'inconvénient d'introduire de nouvelles variables qui vont nécessiter de nouvelles contraintes.

En utilisant la théorie de contrôle optimal, ils formulent le modèle d'optimisation suivant équivalent au ADD :

$$\min_{b,e,v,f,F} I = \int_0^T \sum_a \left\{ \int_0^{b_a(t)} g_{1a}[v_a(t), w] dw + \int_0^{e_a(t)} g_{2a}[v_a(t), w] dw \right\} dt \quad (2.52)$$

s. à :

$$\frac{dv_{ka}}{dt} = b_{ka}(t) - e_{ka}(t), \quad \forall a, k, \quad t \in [0, T'], \quad (2.53)$$

$$\frac{dF_k(t)}{dt} = f_k(t), \quad \forall k, \quad t \in [0, T'], \quad (2.54)$$

$$\sum_{a \in o^+} b_{ka}(t) = h_k(t), \quad \forall o \text{ (origine)}, k \quad t \in [0, T], \quad (2.55)$$

$$\sum_{a \in d^-} e_{ka}(t) = f_k(t), \quad \forall d, k \quad t \in [0, T], \quad (2.56)$$

$$\sum_{a \in j^+} b_{ka}(t) = \sum_{a \in j^-} e_{ka}(t), \quad \forall j \neq o, d, \quad \forall k, \quad t \in [0, T'], \quad (2.57)$$

$$v_{ka}(t) = \sum_{\bar{a} \in \bar{k}} \{v_{k\bar{a}}([t + \tilde{\tau}_a(t)] - v_{k\bar{a}}(t))\} + \{F_k[t + \tilde{\tau}_a(t)] - F_k(t)\}, \quad \forall a, k, \quad t \in [0, T']. \quad (2.58)$$

Puisque la fonction objectif est convexe par rapport aux variables de contrôle, il existe une solution. Ran et al. montrent que les conditions d'optimalité de premier ordre de ce problème de contrôle optimal impliquent les conditions de ADD instantané. Ils proposent aussi un autre modèle où la seule différence est que la fonction de coût est en fonction de $v_a(t)$ et $e_a(t)$, $s_a(v(t)) = s_a[v_a(t), e_a(t)]$.

Boyce, Ran et LeBlanc (1995) proposent une méthode de résolution pour ce modèle. La formulation de contrôle optimal continue est discrétisée et devient un problème de programmation non linéaire avec des contraintes linéaires. Ils utilisent une technique de diagonalisation où, à chaque itération, ils mettent à jour le coût effectif estimé et ils résolvent le problème avec ce coût estimé sur un réseau élargi espace-temps en utilisant la méthode de Frank et Wolfe. Il n'y a aucune preuve que la mise à jour du coût effectif converge vers la

solution du problème.

Nous pouvons résumer les principales limitations trouvées :

- L'ajout des variables et des contraintes rend plus difficile la résolution du modèle.
- Le coût sur les chemins est instantané.
- La propagation du flot doit être faite en fonction du coût effectif, bien que le modèle soit basé sur le coût instantané, donc il est nécessaire de réaliser une estimation du coût effectif qui est mise à jour, mais sans garantie de convergence.
- La fonction de coût est restreinte à être symétrique et dépend de $v_a(t)$, $b_a(t)$ et $e_a(t)$.
- Le réseau élargi espace-temps requiert beaucoup de mémoire.

2.6 Modèles d'inégalités variationnelles (PIV)

Une manière logique de formuler le problème d'équilibre dynamique est d'essayer de modéliser le problème en utilisant les inégalités variationnelles, puisque cette approche a été employée pour modéliser le problème d'équilibre statique. Dans un premier temps, elles étaient appliquées aux problèmes de fonction de coût symétrique (Nguyen (1975), Leblanc et al (1975), Dafermos (1980), Florian (1982a), etc.) où le PIV représente la condition d'optimalité d'un problème d'optimisation à coûts convexes. Plus tard, plusieurs auteurs ont trouvé des méthodes pour vaincre cette limitation (Bertsekas et Gafni (1982), Florian et Spiess (1982), Lawphongpanich et Hearn (1984), Wu, Florian et Marcotte (1993), etc). Une référence importante pour la résolution de problèmes d'inégalité variationnelle est le livre de Patriksson (1999).

Nous allons formuler le ADD comme une inégalité variationnelle de dimension infinie. Nous pouvons diviser le problème en deux parties : une partie d'optimisation, qui décrit le processus par lequel les usagers modifient les décisions relatives à leur déplacement d'une journée à l'autre. Ce processus cherche une condition d'équilibre dynamique (modèles au jour le jour). La deuxième partie consiste à déterminer les volumes du flot temporel des arcs une fois qu'une stratégie est déterminée (chargement de réseau).

Cette modélisation nous permet aussi de formuler facilement d'autres généralisations, comme le choix de l'heure de départ ou l'existence de différentes classes d'usagers.

2.6.1 Formulation basée sur le coût instantané des chemins

Nous allons formuler une inégalité variationnelle de dimension infinie basée sur le coût instantané des chemins dans l'espace des arcs et dans l'espace des chemins.

Dans l'espace des chemins, le PIV est formulé comme suit : trouver $h^*(t) \in \Omega(t)$, pour presque tout $t \in (0, T)$ tel que :

$$\sum_{i \in I} \sum_{k \in K_i} \int_0^T \Phi_k^*(t) [h_k(t) - h_k^*(t)] dt \geq 0, \quad \forall h(t) \in \Omega(t). \quad (2.59)$$

Sur l'espace des arcs, le PIV se formule comme : trouver des $b_a^{i*}(t) \in \Omega_b(t)$ tel que

$$\sum_{i \in I} \sum_{a \in G_i} \int_0^T [s_a(v(t)) + \nu_q^i(t) - \nu_p^i(t)] [b_a^i(t) - b_a^{i*}(t)] dt \geq 0, \quad \forall b \in \Omega_b(t), \quad (2.60)$$

où l'arc $a = (p, q) \in G_i$. L'ensemble $\Omega_b(t)$ est défini par la formulation du problème de chargement dynamique de réseau (section 2.6.3).

Ran et Boyce (1994) montrent l'équivalence de ces deux formulations avec les conditions d'équilibre instantané sur le réseau dans l'espace des chemins et des arcs respectivement.

2.6.2 Formulation basée sur le coût effectif de chemins

Les formulations de contrôle optimal sont basées sur le modèle réactif dans lequel les décisions du choix de route sont basées sur des connaissances instantanées. Par contre, la formulation d'inégalité variationnelle propose aussi des décisions basées sur l'évolution de la route, c'est à dire le modèle préventif.

Nous formulons le PIV basé sur le coût effectif des chemins dans l'espace des chemins : trouver $h^*(t) \in \Omega(t)$, pour presque tout, $t \in (0, T)$ tel que :

$$\sum_{i \in I} \sum_{k \in K_i} \int_0^T S_k^*(t) [h_k(t) - h_k^*(t)] dt \geq 0, \quad \forall h(t) \in \Omega(t); \quad (2.61)$$

et dans l'espace des arcs :

$$\sum_{i \in I} \sum_{a \in G_i} \int_0^T [\mu_q^i(t) + s_a(v(t + \mu_q^i(t))) - \mu_p^i(t)] [b_a^i(t + \mu_p^i(t)) - b_a^{i*}(t + \mu_p^i(t))] dt \geq 0 \quad (2.62)$$

où l'arc $a = (p, q)$.

Ran et Boyce (1994) ainsi que Xu et al. (1999) montrent l'équivalence de ces deux formulations avec les conditions d'équilibre effectif (section 2.2.2).

2.6.3 Chargement dynamique de réseau (network loading)

Le problème de chargement dynamique de réseau, qui a été identifié par Cascetta et Cantarella (1991), consiste à déterminer les volumes temporels des arcs ainsi que les temps de déplacement sur eux et sur les chemins, étant donné les taux de départs temporels du trafic sur les chemins du réseau. Nous présentons ce problème en détail au chapitre 4.

Drissi-Kaïtouni et Gendreau (1992) proposent une formulation discrète avec des files d'attente et des fonctions de sortie. Ces dernières sont en fonction des véhicules qui sont dans la file et de certains autres arguments. Les véhicules sont forcés de suivre un ordre de sortie des arcs pour respecter le FIFO. Le temps est divisé en intervalles de même longueur, et la méthode consiste à "pousser" le flot de l'origine jusqu'à la destination, à travers le réseau et à chaque intervalle de temps.

Friesz et al. (1993) présentent une formulation pour le problème de chargement défini comme un système d'équations fonctionnelles, ils utilisent la fonction du temps de sortie pour décrire la propagation du flot, mais ils ne proposent aucune méthode pour le résoudre :

$$S_k(t) = \Pi_{ka_{n_k}}(t) - t, \quad (2.63)$$

$$\Pi_{ka}(t) = \Pi_{ka^-}(t) + s_a(v_a(\Pi_{ka^-})), \quad (2.64)$$

$$v_{ka}(t) = \int_0^{\Pi_{ka^-}^{-1}(t)} h_k(w)dw - \int_0^{\Pi_{ka}^{-1}(t)} h_k(w)dw, \quad (2.65)$$

$\forall a \in k, k \in K_i, i \in I, t \in [0, T]$; a^- représente l'arc qui précède l'arc a sur le chemin k , a_{n_k} le dernier arc du chemin k et $\Pi_{ka}(t)$ le moment de sortie de l'arc a du flot qui entre dans le chemin k au temps t . L'équation (2.63) exprime le temps effectif de parcours du chemin k , d'un flot qui part de l'origine au moment t ; l'équation (2.64) détermine le moment de sortie

de chaque arc a pour un flot qui part de l'origine au moment t ; l'équation (2.65) exprime le flot de véhicules sur chaque arc a correspondant au chemin k au moment t . Il faut remarquer que l'équation (2.65) représente la contrainte de propagation et de conservation du flot.

Nous rappelons que :

$$S_k(t) = \sum_{a \in k} s_a(v(\Pi_{ka}^-(t))),$$

$$\Pi_{ka}(t) = t + S_{ka}(t).$$

Nous pouvons écrire une formulation équivalente à (2.63) - (2.65) de la forme suivante :

$$\Pi_{ka}(t) = t + \sum_{b \in k_a} s_b(v(\Pi_{kb}(t))), \quad \forall a \in A, k \in K_i, i \in I, t \in [0, T], \quad (2.66)$$

$$v_{ka}(t) = \int_{\Pi_{ka}^{-1}(t)}^{\Pi_{ka}^{-1}(t)} h(w) dw, \quad \forall a \in A, k \in K_i, i \in I, t \in [0, T'], \quad (2.67)$$

$$v_a(t) = \sum_{i \in I} \sum_{k \in K_i} v_{ka}(t), \quad \forall a \in A, t \in [0, T'], \quad (2.68)$$

où k_a est l'ensemble des arcs du chemin k compris entre le premier arc et l'arc a . Wu (1994) et Wu, Chen et Florian (1998) présentent la même formulation pour le problème de chargement sous une condition de "stabilité" qu'implique la condition de FIFO.

Wu (1994) propose la discrétisation suivante en M intervalles de longueur Δt :

$$S_k(m) = \sum_{a \in k} s_a(v(m + \frac{S_{ka}(m)}{\Delta t})), \quad \forall k \in K_i, i \in I, m = 1, \dots, M, \quad (2.69)$$

$$v_{ka}(m) = \sum_{r=\Pi_{ka}^{-1}(m)}^{\Pi_{ka}^{-1}(m)} h_k(r) \Delta t, \quad \forall a \in k, k \in K_i, i \in I, m = 1, \dots, M', \quad (2.70)$$

où $M' \geq M$, $v(0) = 0$, $v(M') = 0$. Cette discrétisation est très compliquée, le temps de parcours est normalement réel (pas entier), spécialement $S_{ka}(m)$, $\Pi_{ka}(m)$ et $\Pi_{ka}^{-1}(m)$ peuvent

ne pas être entiers. Si l'intervalle $m + \frac{S_{ka}(m)}{\Delta t}$ et $\Pi_{ka}^{-1}(m)$ ne sont pas entiers, il propose de les arrondir au plus grand entier inférieur à eux. Ce système n'a pas de solution analytique.

Wu, Chen et Florian (1998) cherchent $\theta_{ka} = \Pi_{ka}^{-1}$, tel que $\theta_{ka}(\Pi_{ka}(t)) = t, \forall a \in A, k \in K_i, i \in I$. Ils visent à minimiser :

$$\| \theta_{ka}(\Pi_{ka}(t)) - t \| . \quad (2.71)$$

Ils proposent une discrétisation dans laquelle ils divisent le temps T en M petits intervalles.

$\theta_{ka}(t)$ est approximée par une forme polynômiale :

$$\theta_{ka}(t) = \sum_{j=1}^{j_0} x_{akj} t^{j-1}, t \in T' \quad (2.72)$$

où j_0 est un entier positif et x_{akj} est un réel.

Ils proposent l'algorithme récursif suivant :

$$\min \sum_{a \in A} \sum_{i \in I} \sum_{k \in K_i} \sum_m \left(\sum_j x_{akj} (\Pi_{ka}(m))^{j-1} - t(m) \right)^2 \quad (2.73)$$

s.à.

$$\Pi_{ka}(m) = t(m) + \sum_{b \in k_a} s_b(v(\Pi_{kb^-}(m))), \quad \forall a, k, i, m, \quad (2.74)$$

$$v_b(t) = \sum_{i \in I} \sum_{k' \in K_i} \int_{\theta_{kb}(\Pi_{kb^-}(m))}^{\theta_{k'b}(\Pi_{kb^-}(m))} h_{k'}(y) dy \quad \forall b, k, m, \quad (2.75)$$

$$\theta_{k'b}(\Pi_{kb^-}(m)) = \sum_j x_{b-k'j} (\Pi_{kb^-}(m))^{j-1}, \quad \forall b, k', k, m, \quad (2.76)$$

$$\theta_{k'b}(\Pi_{kb^-}(m)) = \sum_j x_{bk'j} (\Pi_{kb^-}(m))^{j-1}, \quad \forall b, k', k, m. \quad (2.77)$$

L'équation (2.73) est la fonction objectif à minimiser ; l'équation (2.75) calcule le flot sur l'arc b , calculé en fonction du moment de départ sur les chemins des flots qui arrivent à l'arc au temps t . La condition FIFO doit être vérifiée. La sommation $\sum_{b \in k} s_b(v(\Pi_{kb^-}(m)))$ représente le temps effectif de parcours d'un chemin k , et l'équation (2.74) calcule le moment

de sortie de l'arc a du flot qui est entré dans le chemin k dans l'intervalle m , qui est égal au temps de départ plus le temps effectif du voyage.

Xu et al. (1999) proposent, pour les cas avec fonction de coût séparable, la formulation itérative suivante :

$$v_a(t) = \int_0^t (b_a(s) - e_a(s)) ds, \quad a \in A, \quad t \in (0, \tau_a(T_a)), \quad (2.78)$$

$$\tau_a(t) = t + s_a(v_a(t)), \quad a \in A, \quad t \in (0, T_a), \quad (2.79)$$

$$\tau'_a(t) = 1 + s'_a(v_a(t)), \quad a \in A, \quad t \in (0, T_a), \quad (2.80)$$

$$b_{ka}(t) = \begin{cases} h_k(t) & \text{si } a \text{ est le premier arc du chemin } k, \\ e_{ka} & \text{sinon,} \\ & a \in A, \quad k \in K, \quad t \in (0, T_{ka}), \end{cases} \quad (2.81)$$

$$e_{ka}(t) = \begin{cases} 0 & \text{si } 0 \leq t \leq s_a(0), \\ b_{ka}(\tau_a^{-1}(t)(\tau_a^{-1})'(t)) & \text{sinon,} \\ & a \in A, \quad k \in K, \quad t \in (s_a(0), \tau_a(T_{ka})), \end{cases} \quad (2.82)$$

$$b_a(t) = \sum_{k \in K_a} b_{ka}(t), \quad a \in A, \quad t \in (0, T_a), \quad (2.83)$$

$$e_a(t) = \sum_{k \in K_a} e_{ka}(t), \quad a \in A, \quad t \in (0, \tau(T_a)), \quad (2.84)$$

où T_a représente le moment d'arrivée à la queue de l'arc a du dernier flot, et T_{ka} le moment d'arrivée à la queue de l'arc a du dernier flot correspondant au chemin k . Notez que $s'_a(v_a(t))$ représente la dérivée de $s_a(v_a(t))$ par rapport à t . Le flot est aussi "poussé" de l'origine jusqu'à la destination, de façon similaire à Drissi-Kaïtouni et Gendreau, mais à chaque itération les calculs seront faits seulement pour un arc a et un intervalle de temps $[t, \tau_a(t)]$.

2.6.4 Modèle avec choix du temps de départ

Jusqu'à ici, nous avons présenté des modèles où la demande temporelle est fixe et où nous devons trouver un choix de route selon laquelle nous pouvions vérifier certaines conditions d'équilibre dynamique sur un réseau de transport. Dans cette sous-section, nous allons aborder une nouvelle classe de modèles qui sont aussi une généralisation du principe de Wardrop, tout en étant plus générale que la précédente, puisque les usagers vont chercher, non seulement, à optimiser leur temps de voyage, mais aussi à choisir l'heure de départ la plus appropriée. Ces modèles sont plus réalistes dans certains cas ; par exemple, si nous voulions représenter l'affectation du flot en équilibre pour le matin quand les usagers se déplacent de la maison vers leur travail ou occupation respective. Cette affectation serait un exemple typique où normalement les utilisateurs veulent optimiser leur temps de parcours en arrivant à leur destination, à une heure d'entrée qui se rapproche le plus possible de l'heure déterminée.

Nous parlerons du problème d'équilibre sur un réseau par rapport au choix de route et du moment de départ. Donc, nous allons élaborer une description d'équilibre plus appropriée où tous les usagers qui ont un même "but", (la même origine, la même destination et la même heure désirée d'arrivée) expérimentent le même coût de voyage, en regardant le temps de déplacement sur la route et le moment de l'arrivée ; celui-ci est plus petit ou égal au coût qui aurait été expérimenté par n'importe quelle autre combinaison "route-temps de départ" qu'on aurait pu utiliser.

Avec cette nouvelle définition d'équilibre, le coût de parcours d'un chemin doit être en équilibre pour chaque route-temps de départ.

Définition 5 *Une affectation du flot dynamique sur un réseau de transport est dite en équilibre par rapport au choix de route et au moment de départ si, pour chaque paire origine-destination, le coût effectif expérimenté par les voyageurs, incluant une pénalité pour arriver tôt ou tard, est le même et est minimal.*

Nous supposons que les voyageurs sont prêts à partir de leur origine à partir du moment zéro, mais ils préfèrent partir le plus tard possible, et ce “report” du moment de départ serait du temps gagné.

Mahmassani et Herman (1984) et Mahmassani et Chang (1987) utilisent un modèle de simulation du flot de trafic. L'équilibre est établi selon le temps de départ, et ils utilisent une relation entre la vitesse, la densité de véhicules, le flot et le taux d'arrivée au début d'une section de route déterminée. Ils résolvent des problèmes de temps de départ pour des réseaux simples.

Les inégalités variationnelles sont une bonne approche pour représenter cette nouvelle définition d'équilibre. De Palma et al. (1983) et Ben-Akiva et al. (1986) ont introduit une *fonction barème* pour décrire la pénalité qui sera “appliquée” si un usager arrive “tôt” ou “tard”. Soit t^* , l'heure d'arrivée visée, par exemple, l'heure d'entrée au travail. Nous allons définir un intervalle de temps d'arrivée “désiré” $[t^* - \delta, t^* + \delta]$ dans lequel il n'y aurait pas de pénalité.

Nous supposons, pour simplifier, mais sans perte de généralité, que tous les utilisateurs désirent arriver à la même heure et avec les mêmes valeurs de temps.

La fonction de pénalité est de la forme suivante :

$$\gamma_k(t, t^*) = \begin{cases} \alpha_1[(t^* - \delta) - (t + S_k(t))] & \text{si } t + S_k(t) < t^* - \delta; \\ 0 & \text{si } |t + S_k(t) - t^*| \leq \delta; \\ \alpha_2[(t + S_k(t)) - (t^* + \delta)] & \text{si } t + S_k(t) > t^* + \delta; \end{cases} \quad (2.85)$$

où α_1 et α_2 sont des paramètres positifs.

Friesz et al. (1993), Wie et al. (1995) et Bernstein et al. (1993), entre autres, l'utilisent et ils définissent une nouvelle fonction de "coût généralisé" :

$$C_k(t) = \eta S_k(t) + \gamma_k(t). \quad (2.86)$$

où η est une constante, Friesz et al. ainsi que Wie. et al. considèrent η égal à 1. Bernstein et al. considèrent qu'elle doit avoir une valeur réelle strictement positive.

Les conditions d'équilibre dynamique descriptif effectif sur les chemins, selon le choix de routes et le temps de départ deviennent celles-ci :

$$C_k(t) > C_k^{min}(t) \Rightarrow h_k(t) = 0, \quad (2.87)$$

$$h_k(t) > 0 \Rightarrow C_k(t) = C_k^{min}(t), \quad (2.88)$$

où $C_k^{min}(t)$ est le coût minimal des chemins/temps de départ qui relie la paire $(OD)_i$ au long de la période, pour $t \in [0, T]$, $k \in K_i$, $i \in I$.

Le PIV devient : trouver $h^*(t) \in \Omega(t)$ tel que :

$$\sum_{i \in I} \sum_{k \in K_i} \int_0^T C_k(t) [h_k(t) - h_k^*(t)] dt \geq 0, \quad \forall h(t) \in \Omega(t), \quad t \in (0, T). \quad (2.89)$$

Il est nécessaire de bien spécifier les valeurs des paramètres α_1 , α_2 , η de telle façon que

le temps de retard soit plus “ coûteux ” que le temps de parcours, et ce dernier plus “ coûteux ” que le temps d’attente, pour éviter des incohérences telles que prendre un chemin plus long afin de réduire le temps d’attente.

Ran, Hall et Boyce (1996) proposent une fonction de pénalité différente basée sur les paires origine-destination, et non pas sur les chemins :

$$\gamma_i(t, \mu_i) = \begin{cases} \alpha_1 [t + \mu_i(t) - t^* + \delta]^2 & \text{si } t + \mu_i(t) < t^* - \delta; \\ 0 & \text{si } |t + \mu_i(t) - t^*| \leq \delta; \\ \alpha_2 [t + \mu_i(t) - t^* - \delta]^2 & \text{si } t + \mu_i(t) > t^* + \delta; \end{cases} \quad (2.90)$$

dans ce cas α_1 est une constante négative parce qu’ils considèrent qu’arriver tôt doit être encouragé plutôt que découragé. Ils définissent

$$C_i(t) = \alpha t + \mu_i(t) + \gamma_i(t), \quad \forall i \in I, \quad (2.91)$$

et ils formulent la condition d’équilibre de la façon suivante :

$$(\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) - \mu_q^i(t)) b_a^i(t + \mu_p^i(t)) = 0, \quad \forall a = (p, q), \quad i \in I, \quad (2.92)$$

$$\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) - \mu_q^i(t) \geq 0, \quad \forall a = (p, q), \quad i \in I, \quad (2.93)$$

$$b_a^i(t + \mu_p^i(t)) \geq 0, \quad \forall a = (p, q), \quad i \in I, \quad (2.94)$$

$$g_i^*(t)(C_i^*(t) - C_i^{\min}) = 0, \quad \forall i \in I, \quad (2.95)$$

$$(C_i^*(t) - C_i^{\min}) \geq 0, \quad \forall i \in I, \quad (2.96)$$

$$g_i(t) \geq 0, \quad \forall i \in I, \quad (2.97)$$

C_i^{\min} est le minimum C_i sur toute la période.

Ils formulent l’inégalité variationnelle équivalente suivante :

$$\int_0^T \left\{ \sum_i \sum_a [\mu_p^i(t) + s_a(v(t + \mu_p^i(t))) - \mu_q^i(t)] [b_a^i(t + \mu_p^i(t)) - b_a^{i*}(t + \mu_p^i(t))] \right\}$$

$$+ \sum_i C_i^*(t)[g_i(t) - g_i^*(t)]dt \geq 0. \quad (2.98)$$

Cette formulation est très compliquée et $C_i(t)$ n'est pas continue. Si α_1 est négatif, tous les usagers voudront arriver le plus tôt possible, mais cela n'est pas réaliste.

Drissi-Kaïtouni et Gendreau (1992) ajoutent une pénalité sur le moment de départ, alors le coût total d'un voyage aurait trois composantes : le temps de parcours lui même, une pénalité relative au moment de départ et une autre relative au moment d'arrivée.

2.6.5 Formulations continues

Friesz et al. (1993) présentent une formulation d'inégalité variationnelle sur un espace de Hilbert, pour le problème d'équilibre dynamique portant sur le choix de routes et du temps de départ, basée sur le temps de parcours effectif. Ils démontrent que, si la fonction de parcours est linéaire, la condition de FIFO est respectée. Ils montrent l'équivalence entre ce modèle et les conditions d'équilibre.

Wu (1994) présente un modèle continu semblable, mais seulement pour le choix de routes, et il propose une méthode de discrétisation pour le résoudre où des inégalités variationnelles finies sont obtenues. Le problème discrétisé consiste à trouver $h^*(m) \in \Omega(m)$, $m = 1, 2, \dots, M$ (M intervalles) tel que

$$S(h^*(m)^t(h(m) - h^*(m))) \geq 0, \forall h(m) \in \Omega(m), \quad (2.99)$$

où $\Omega(m) = \{h(m) | \sum_{k \in K_i} h_k(m) = g_i(m), \forall i \in I; h_k(m) \geq 0, \forall k \in K_i, i \in I\}$.

Le problème de chargement sur le réseau aurait les mêmes discrétisations (fixées au départ), ce qui complique beaucoup la résolution.

Wu et al. (1995) proposent un algorithme de projection où, à chaque itération, l ils déterminent une solution réalisable h^{l+1} obtenue en résolvant le problème :

$$\min_{h^{l+1} \in \Omega} (S(h^l), h^{l+1} - h^l) + \frac{1}{2\alpha} (h^{l+1} - h^l, h^{l+1} - h^l), \quad (2.100)$$

où α est donné, fixe et strictement positif.

Ils montrent que, sous certaines conditions, le PIV a une solution unique, h^* , et que la séquence $\{h^l\}$ générée par l'algorithme de projection converge vers cette solution h^* . Les conditions nécessaires sont :

- Ω est compact et convexe.
- $S(h)$ satisfait des conditions de continuité de Lipschitz.
- $S(h)$ est fortement monotone par rapport à h .

Pour le résoudre, ils proposent une "discrétisation-continue" dans laquelle h_k est approximée par une fonction polynômiale :

$$h_k(t) = \sum_{r \in R} y_{kr} t^{r-1}, \quad t \in [0, T].$$

$R = \{1, 2, 3, \dots\}$ est un ensemble d'entiers positifs et y_{kr} sont des nombres réels. Cette fonction permet de faire des discrétisations différentes à chaque itération.

À chaque itération de l'algorithme de projection, on résout l'approximation quadratique discrète suivante :

$$\min \sum_{i \in I} \sum_{k \in K_i} \sum_{m \in M} [S_k^l(m)(h_k(m) - h_k^l(m)) + \frac{1}{2\alpha} (h_k(m) - h_k^l(m))^2] \quad (2.101)$$

s. à.

$$\sum_{k \in K_i} h_k(m) = g_i(m), \quad m \in M, \quad (2.102)$$

$$h_k(m) = \sum_{r \in R} y_{kr} (t(m))^{r-1}, \quad k \in K_i, \quad i \in I, \quad m \in M, \quad (2.103)$$

$$h_k(m) \geq 0, \quad k \in K_i, \quad i \in I, \quad m \in M. \quad (2.104)$$

Ran, Hall et Boyce (1996) formulent le PIV basé sur l'espace du flot sur les arcs ; ils proposent (2.62) sous les contraintes (2.53) - (2.58). Ils l'élargissent pour le choix du temps de départ tel que nous l'avons décrit dans la section précédente. Ils ne proposent pas d'algorithme pour le résoudre.

2.6.6 Formulations discrètes

Smith (1993) propose un modèle pour le choix de route où le temps est discrétisé en petites unités et le réseau en petites unités de parcours, arcs, ayant le même temps de parcours quand il n'y a pas de congestion, qui est égal à une unité de temps (c'est-à-dire, un arc est parcouru dans une unité de temps). Dans chaque arc, le flot de sortie est limité par une capacité maximale.

À chaque unité de temps, le flot prêt à sortir d'un arc continue son chemin si la capacité maximale de sortie n'est pas atteinte, sinon il reste sur l'arc au moins jusqu'à l'unité de temps suivante.

Les véhicules sont regroupés en paquets, "packets". Un "packet" est un ensemble de véhicules qui suivent la même route, qui ont le même moment de départ, qui sont sur le

même noeud au même moment et qui ont expérimenté les mêmes délais sur les noeuds précédents. Le volume d'un *packet* peut être non entier. Les *packets* peuvent être découpés, par exemple, à cause de la capacité, mais les véhicules qui étaient dans un même *packet* au moment de départ sont forcés de suivre le même chemin, même si leurs temps d'arrivée sont différents.

Pour respecter le FIFO, il introduit des priorités. Chaque *packet* est classé selon ces priorités de telle façon que les *packets* qui sont sur un même noeud au même moment sont placés dans un ordre approprié. À cause des capacités et de l'imposition de parcourir un même chemin à tous les véhicules d'un même *packet* de départ, le principe d'équilibre n'est pas vérifié. Il formule un principe qui est quand même dans l'esprit de Wardrop : pour chaque paire *OD* et chaque temps de départ t , les routes qui ont un coût plus élevé ne sont pas utilisées.

Il présente trois formulations équivalentes : une basée sur les inégalités variationnelles, une autre sur la condition de point fixe (projection) et une autre sur la formulation d'un problème de minimisation.

Wie et al. (1995) formulent un PIV pour le choix de routes et temps de départ sur un temps discret en termes de fonction de coût sur les chemins basé sur des fonctions de sortie du flot (exit flow) sur les arcs. Le volume de trafic sur chaque arc est décrit comme dans le modèle de Merchant et Nemhauser :

$$v_a(t+1) = v_a(t) + b_a(t) - e_a(v_a(t)), \quad \forall a \in A, t = 0, 1, \dots, T, \quad (2.105)$$

Ils évitent d'utiliser la fonction inverse du temps de sortie. Ils montrent qu'il y a une solution

au problème d'affectation sous les conditions suivantes : $e_a(v_a(t))$ non négative et continue $\forall v_a(t)$; $e_a(0) = 0$ et $e_a(v_a(t)) > 0$ quand $v_a(t) > 0 \forall a \in A$.

Ils proposent un algorithme heuristique (la convergence n'est pas établie) qui n'est pas basé sur des méthodes itératives pour résoudre le PIV. Ils procèdent en générant une séquence de solutions qui est supposée de converger vers la solution discrète de ADD. À chaque itération, l'algorithme génère une solution, résout le problème de chargement sur le réseau et fait un ajustement sur le taux de départ, δh , selon deux cas : si $C_k(t, h_k) > u_i(h_k) \Rightarrow \delta h_k$ est négative (on enlève du flot), si $u_i(h_k) \leq C_k(t, h_k) \leq u_i(h_k) + \epsilon \Rightarrow \delta h_p$ est positive (on ajoute du flot).

Boyce, Lee et Janson (1996) proposent un modèle discret pour ADD effectif formulé comme une inégalité variationnelle dans l'espace des arcs. Ils proposent un algorithme de diagonalisation. Mais ils considèrent qu'un arc a est occupé au temps t , si le nombre de véhicules qui se trouvent sur lui à ce moment t est positif, $v_a(t) > 0$ (il faut se rappeler que nous avons défini un arc comme occupé au temps t si le taux d'entrée est positif, $b_a(t) > 0$). Le résultats ne sont pas convaincants

Bernstein et al. (1993) proposent une formulation de *contrôle variationnel*, un modèle de contrôle optimal, mais ils substituent la fonction objectif à minimiser par une inégalité variationnelle. Cela est semblable au modèle de Ran, Hall et Boyce, mais en temps discret.

Chen et Smith (1998) proposent un modèle discret où ils formulent une inégalité variationnelle dans l'espace des arcs, où la condition de FIFO n'est pas vérifiée. Ils utilisent

un réseau élargi espace-temps avec des files d'attente où le temps de parcours des arcs doit être entier et où il est formulé en fonction du flot d'entrée. À chaque itération, ils trouvent de nouvelles valeurs des flots d'entrée sur les arcs et ils calculent la fonction de temps de parcours. Pour formuler la propagation du flot, ils utilisent les coûts de l'itération précédente dans la nouvelle solution trouvée, alors celle-ci n'est pas nécessairement réalisable. (La relation dynamique entre la propagation du flot, le temps de parcours et le flot sur les arcs est très difficile à établir avec une formulation sur l'espace des arcs).

Gendreau et Bouzaiene-Ayari (1996) proposent aussi une inégalité variationnelle pour le choix de routes et le temps de départ en utilisant une formulation du style de celle de Drissi-Kaïtouni et Gendreau (1992).

Chapitre 3

Formulation mathématique : les contraintes du modèle considéré

Dans ce chapitre, nous allons présenter le modèle et ses contraintes que nous utilisons pour résoudre le problème d'affectation dynamique. L'objectif est de trouver une solution du modèle d'affectation dynamique descriptif en le formulant comme une inégalité variationnelle sur l'espace des chemins basé sur le coût effectif :

$$\sum_{i \in I} \sum_{k \in K_i} \int_0^T S_k^*(t) [h_k(t) - h_k^*(t)] dt \geq 0, \quad \forall h(t) \in \Omega(t), t \in (0, T). \quad (3.1)$$

L'ensemble Ω vérifie les contraintes de conservation et de non-négativité des flots qui entrent dans les chemins. À partir d'une matrice de demandes temporelles et des fonctions de coût sur les arcs donnés, nous devons déterminer les chemins utilisés, k , à chaque moment de départ et leurs flots d'entrée, $h_k(t)$, modèle au jour le jour, qui sera exécuté d'une manière itérative. Mais il va falloir que, tout au long des chemins que ces flots vont emprunter, certaines conditions soient respectées. Ainsi, nous allons définir les contraintes d'état, les contraintes de conservations du flot, les contraintes de propagation du flot et les contraintes FIFO. Ces contraintes vont diriger les mouvements des usagers une fois que les h_k sont déterminés. Nous pourrions dire que ces contraintes vont définir le comportement des usagers

à l'intérieur d'une journée, ceci est définie comme le problème de chargement du réseau. (voir diagramme - figure ??). Nous parlerons aussi de la fonction du flot de sortie, (nous n'utilisons pas cette fonction, mais nous jugeons important de la présenter plus en détail, puisqu'elle a été très utilisée par plusieurs auteurs).

Soit $v_a(t)$ la quantité de flot (usagers) qui est sur l'arc a au moment t , et $v_{ka}(t)$, la quantité de flot qui est sur l'arc a et qui emprunte le chemin k au moment t :

$$v_a(t) = \sum_{k \in K} v_{ka}(t), \quad \forall a.$$

Nous dénotons $b_a(t)$, le taux du flot qui entre à l'arc a au moment t , et $b_{ka}(t)$, le taux du flot qui entre dans l'arc a et qui se déplace sur le chemin k au moment t . Également, nous dénotons $e_a(t)$, le taux du flot qui sort de l'arc a au moment t , et $e_{ka}(t)$, le taux du flot qui sort de l'arc a et qui voyage par le chemin k au moment t

$$\sum_{k \in K} b_{ka}(t) = b_a(t), \quad \sum_{k \in K} e_{ka}(t) = e_a(t), \quad \forall a.$$

Toutes ces variables doivent être non négatives :

$$v_{ka}(t) \geq 0, \quad b_{ka}(t) \geq 0, \quad e_{ka}(t) \geq 0.$$

Nous pouvons faire référence aussi à la quantité accumulée des flots qui sont entrés sur l'arc a en suivant le chemin k au moment t , $B_{ka}(t)$, ou qui en sont sortis, $E_{ka}(t)$.

$$B_{ka}(t) = \int_0^t b_{ka}(s) ds, \quad \forall a, k, t \in [0, T_{ka}],$$

$$E_{ka}(t) = \int_0^t e_{ka}(s) ds, \quad \forall a, k, t \in [0, \tau_a(T_{ka})].$$

où T_{ka} est le temps d'arrivée à la queue de l'arc a du dernier flot correspondant au chemin k , et $\tau(t)$ est la fonction du temps de sortie, c'est-à-dire le moment de sortie de l'arc a du

flot qui est entré à l'arc au moment t ; et

$$\sum_{k \in K} B_{ka}(t) = B_a(t), \quad \forall a, t \in [0, T_a],$$

$$\sum_{k \in K} E_{ka}(t) = E_a(t), \quad \forall a, t \in [0, \tau_a(T_a)],$$

où $B_a(t)$ et $E_a(t)$ sont la quantité accumulée des flots qui sont entrés et sortis respectivement de l'arc a , et T_a est le moment où le dernier flot entré dans l'arc a arrive à la queue de a .

La fonction de temps de parcours (ou de coût) de l'arc a est en fonction du flot qui est sur l'arc au moment t , $s_a(v_a(t))$. La quantité de flot accumulé qui est entré ou sorti de l'arc a au moment initial, $t = 0$, est considérée égale à 0,

$$B_{ka}(0) = 0, \quad E_{ka}(0) = 0, \quad \forall a, k,$$

et donc :

$$v_{ka}(0) \geq 0, \quad B_{ka}(t) \geq 0, \quad E_{ka}(t) \geq 0, \quad \forall a, k.$$

Nous allons présenter les contraintes nécessaires pour décrire le problème d'équilibre dynamique. En considérant les contraintes d'état, de conservation du flot, de propagation du flot et FIFO, premier entré premier sorti (*first in first out*).

3.1 Contraintes d'état

La quantité du flot sur l'arc a au moment t est déterminée par la différence entre la quantité du flot accumulé qui est entré dans l'arc a et la quantité du flot accumulé qui en est sorti.

$$v_a(t) = B_a(t) - E_a(t), \quad \forall a, t \in (0, \tau_a(T_a)). \quad (3.2)$$

La variation (dérivée) du nombre de véhicules sur l'arc a , flot sur l'arc, est définie par la différence entre le flot d'entrée et le flot de sortie :

$$\frac{dv_a(t)}{dt} = b_a(t) - e_a(t), \quad \forall a, t \in (0, \tau_a(T_a)). \quad (3.3)$$

Nous pouvons dire de même pour le flot sur l'arc a provenant du chemin k .

$$v_{ka}(t) = B_{ka}(t) - E_{ka}(t), \quad \forall a, k, t \in (0, \tau_a(T_{ka})), \quad (3.4)$$

$$\frac{dv_{ka}(t)}{dt} = b_{ka}(t) - e_{ka}(t), \quad \forall a, k, t \in (0, \tau_a(T_{ka})). \quad (3.5)$$

3.2 Contraintes de conservation du flot

Comme pour le problème statique, nous allons associer des contraintes de conservation du flot aux noeuds du réseau. La contrainte de conservation du flot associé à un noeud $n \in N$ indique que la quantité du flot entrant dans le noeud est égale à la quantité totale qui en sort. Nous allons la formuler d'abord pour chaque chemin. Nous allons déterminer le flot de véhicules qui entrent dans un arc $a = (p, q)$ au temps t provenant du chemin k selon un des deux cas suivants :

- a) si p est un noeud origine, c'est-à-dire a est le premier arc du chemin k , alors le flot de véhicules qui entrent dans a est égal au taux de départ sur le chemin au moment t :

$$b_{ka}(t) = h_k(t) \quad \forall k, t \in (0, T_{ka}). \quad (3.6)$$

La quantité accumulée du flot qui est entré au moment t sera :

$$B_{ka}(t) = \int_0^t h_k(s) ds \quad \forall k, t \in (0, T_{ka}); \quad (3.7)$$

- b) si p est un noeud intermédiaire, c'est-à-dire p n'est pas origine ni destination, le flot de véhicules qui entrent dans a correspondant au chemin k est égal au flot de

véhicules qui sortent de l'arc précédant a sur le chemin k .

$$b_{ka}(t) = e_{ka^-}(t) \quad \forall k \quad t \in (0, T_{ka}). \quad (3.8)$$

ka^- représente l'arc précédant a sur le chemin k .

Et donc, la quantité de flot qui est entré dans l'arc sera conservée aussi :

$$B_{ka}(t) = E_{ka^-}(t) \quad \forall k \quad t \in (0, T_{ka}). \quad (3.9)$$

Pour chaque noeud $n \in N$, nous définissons n^+ comme l'ensemble des arcs qui sortent du noeud n et n^- comme l'ensemble des arcs qui entrent dans le noeud n .

Dans une formulation arc-sommets, on obtient :

$$\sum_{a \in n^+} b_a(t) = \sum_{a \in n^-} e_a(t), \quad \forall n \neq o_i, d_i; \quad \forall i \in I, \quad (3.10)$$

$$\sum_{a \in o_i^+} b_a(t) = g_i(t), \quad \forall i \in I, \quad (3.11)$$

o_i et d_i représentent des origines et des destinations, respectivement.

3.3 Contraintes de propagation du flot

Il est nécessaire de s'assurer que le flot qui entre et sort ainsi que le flot de véhicules présents sur les arcs soient cohérents avec les temps de parcours de ceux-ci. Nous pouvons écrire cette condition dans l'espace des arcs ou dans celui des chemins. Nous avons besoin aussi d'une condition qui nous assure que le premier flot à entrer sera le premier à sortir, condition FIFO. Nous allons parler plus tard de cette condition.

3.3.1 Contraintes de propagation du flot dans l'espace des chemins

Soit $S_{ka}(t)$, le temps effectif de parcours sur le chemin k de l'origine jusqu'à la fin de l'arc a des usagers qui entrent dans le chemin au moment t . Nous savons que le flot qui s'engage sur le chemin k au moment t sort de l'arc $a = (p, q)$ au moment $t + S_{ka}(t) = \Pi_{ka}(t)$.

Nous avons

$$H_k(t) = E_{ka}(\Pi_{ka}(t)) \quad \forall a, k, \quad (3.12)$$

où $H_k(t)$ est la quantité de véhicules (flot accumulé) qui ont emprunté le chemin k entre 0 et l'instant t . En dérivant (3.12) par rapport à t , nous obtenons :

$$h_k(t) = e_{ka}(\Pi_{ka}(t))\Pi'_{ka}(t). \quad (3.13)$$

Nous supposons que S_{ka} est différentiable et que $\Pi_{ka}(t)$ est inversible. Par (3.13) :

$$h_k[\Pi_{ka}^{-1}(t)] = e_{ka}(t)\Pi'_{ka}(\Pi_{ka}^{-1}(t)) = \frac{e_{ka}(t)}{(\Pi_{ka}^{-1})'(t)}, \quad (3.14)$$

et nous pouvons conclure que :

$$e_{ka}(t) = \begin{cases} 0 & \text{si } t \in (0, S_{ka}(0)); \\ h_k(\Pi_{ka}^{-1}(t))(\Pi_{ka}^{-1})'(t) & \text{si } t \in (S_{ka}(0), \tau_a(T_{ka})). \end{cases} \quad (3.15)$$

3.3.2 Contraintes de propagation du flot dans l'espace des arcs

Nous pouvons dire que le flot de véhicules qui entre à l'arc a au moment t doit sortir au moment $t + s_a(v(t)) = \tau_a(t)$, et alors, pour chaque chemin k , la quantité du flot qui est entré dans l'arc a au moment t doit être égale à la quantité du flot qui est sorti de a au moment $\tau_a(t)$.

$$B_{ka}(t) = E_{ka}(\tau_a(t)) \quad \forall a, k, \quad t \in [0, T_{ka}]. \quad (3.16)$$

En dérivant (3.16) nous obtenons :

$$b_{ka}(t) = e_{ka}(\tau_a(t))\tau'_a(t). \quad (3.17)$$

Nous supposons que s_a est différentiable et $\tau_a(t)$ est inversible. Alors à partir de (3.17), nous pouvons déduire la formulation suivante :

$$b_{ka}(\tau_a^{-1}(t)) = e_{ka}(t)\tau'_a(\tau_a^{-1}(t)) = \frac{e_{ka}(t)}{(\tau_a^{-1})'(t)}, \quad (3.18)$$

$$e_{ka}(t) = \begin{cases} 0 & \text{si } t \in (0, s_a(0)); \\ b_{ka}(\tau_a(t)^{-1})(\tau_a^{-1})'(t) & \text{si } t \in (s_a(0), \tau_a(T_{ka})). \end{cases} \quad (3.19)$$

similaire à la contrainte de propagation du flot utilisée par Xu et al. (1999).

3.4 Contrainte FIFO : premier entré-premier sorti

La condition de FIFO peut être définie au niveau des arcs, des chemins ou des paires origine-destination.

La condition de FIFO sur les arcs est satisfaite si les usagers qui entrent plus tôt sur un arc donné en sortent aussi plus tôt.

$$t' > t'' \implies t' + s_a(v(t')) > t'' + s_a(v(t'')), \quad \forall a \in A, \quad t', t'' \in [0, T_{ka}]. \quad (3.20)$$

La condition de FIFO sur les chemins est satisfaite si les usagers qui entrent plus tôt sur un chemin en sortent aussi plus tôt.

$$t' > t'' \implies t' + S_k(t') > t'' + S_k(t''), \quad \forall k \in K_i, \quad i \in I, \quad t', t'' \in [0, T]. \quad (3.21)$$

La condition de FIFO sur les paires origine-destination est satisfaite si les usagers qui partent plus tôt de l'origine arrivent aussi plus tôt à la destination si le réseau est dans un état d'équilibre descriptif :

$$t' > t'' \implies t' + \mu_i(t') > t'' + \mu_i(t''), \quad \forall i \in I, \quad t', t'' \in T. \quad (3.22)$$

Beaucoup d'efforts ont été faits pour trouver des conditions, les plus générales possibles, pour assurer la condition de FIFO.

Supposons que la condition de FIFO sur les arcs soit vérifiée, et qu'un usager entre dans l'arc a au moment t . Il sortira donc à l'instant $t + s_a(v(t))$. Supposons aussi qu'un autre usager entre plus tard, c'est-à-dire au moment $t + \Delta t$, il sortira donc à $t + \Delta t + s_a(v(t + \Delta t))$. Alors par FIFO nous avons :

$$\begin{aligned} t + s_a(v(t)) &< t + \Delta t + s_a(v(t + \Delta t)), \\ \tau_a(t) &< \tau_a(t + \Delta t). \end{aligned} \tag{3.23}$$

En divisant (3.23) par Δt nous obtenons :

$$\frac{s_a(v(t + \Delta t)) - s_a(v(t))}{\Delta t} + 1 > 0. \tag{3.24}$$

Nous prenons la limite de (3.24) ($\Delta t \rightarrow 0$),

$$s'_a(v(t)) + 1 > 0 \implies s'_a(v(t)) > -1. \tag{3.25}$$

Wu, Chen et Florian (1998) appellent (3.24) condition de stabilité ou régularité ou cohérence. Ils montrent que cette condition est nécessaire et suffisante pour vérifier la condition de FIFO sur les arcs. Si la condition de FIFO est vérifiée sur les arcs, cela implique aussi qu'elle est vérifiée sur les chemins. Wu, Chen, Florian (1998) montrent que si la condition de FIFO est vérifiée sur les chemins ainsi que les conditions d'équilibre (2.27)-(2.28) sur le réseau sont vérifiées, alors la condition de FIFO sur les paires origine-destination est aussi vérifiée.

Friesz et al. (1993) montrent que si la fonction $s(v)$ est linéaire, la condition de FIFO

sur les arcs est toujours vérifiée. Ils énoncent le théorème suivant :

Théorème 1 (Friesz et al. 1993) *Soit s , une fonction linéaire, alors la fonction du temps de sortie τ est strictement croissante par rapport à t , et la fonction inverse τ^{-1} existe.*

Notez que, si τ est strictement croissante, la condition de FIFO sur les arcs est toujours vérifiée.

Xu et al. (1999) arrivent à la même conclusion par leur théorème 4.2 et ils définissent aussi certaines conditions pour assurer la condition FIFO quand la fonction de coût, s , est non linéaire et séparable : b_a non négatif et borné par B_a et intégrable, $\frac{ds_a}{dv} < 1/B_a$ pour tout v sur l'intervalle $[0, V_a]$ où $V_a = \int_0^{T'} b_a(t) dt$.

Astarita (1996) trouve que la condition suivante est nécessaire et suffisante pour que la condition de FIFO soit vérifiée :

$$s'_a(v(t)) = -1 + \frac{b_a(t)}{e_a(t + s(v(t)))}. \quad (3.26)$$

L'équation (3.26) implique (3.25) si $b(t)$ et $e(t + s(t))$ sont > 0 , condition très courante et normale puisqu'un flot d'entrée ou de sortie négatif aurait du sens seulement si le flot se déplace vers l'arrière.

À partir de (3.26) nous pouvons développer :

$$s'_a(t) + 1 = \frac{b_a(t)}{e_a(t + s(t))}, \quad (3.27)$$

$$\tau'_a(t) = \frac{b_a(t)}{e_a(t + s(t))}, \quad (3.28)$$

$$b_a(t) = e_a(\tau(t))\tau'_a(t), \quad (3.29)$$

cette dernière équation est la contrainte de propagation du flot (3.17). Donc, si $b_a(t) > 0$ cette contrainte vérifie la condition de FIFO.

3.5 Fonction du flot de sortie

Nous trouvons plusieurs auteurs qui utilisent pour décrire la propagation du flot une fonction qui définit la relation entre le flot qui sort d'un arc à un moment donné et le flot de véhicules qui sont sur l'arc, $e_a(v_a)$. Par exemple, Merchant et Nemhauser (1978a) l'utilisent pour un modèle discret et Friesz et al (1989) pour un modèle continu. La fonction $e_a(v_a)$ est dépendante du flot de véhicules qui se trouve sur l'arc et détermine le flot qui sort de l'arc a au moment t , c'est-à-dire, la quantité de véhicules par unité de temps. Ils remplacent la contrainte d'état et la contrainte de propagation du flot par cette formule :

$$\frac{dv_a(t)}{dt} = b_a(t) - e_a(v_a(t)). \quad (3.30)$$

En général, la vitesse du flot est déterminée d'une façon implicite par la fonction du flot de sortie. Elle permet de décrire l'évolution et la propagation du flot. Cependant, plusieurs déficiences ont été détectées lors de son utilisation. Quand on applique l'équation (3.30), le flot qui entre dans l'arc a au moment t est ajouté immédiatement au flot qui est déjà dans l'arc, $v_a(t)$, donc, il intervient aussi dans le calcul du flot sortant, $e_a(v_a(t))$. Ceci n'est pas réaliste, puisque le flot qui sort devrait dépendre uniquement de celui qui est "près" de la tête de l'arc, prêt à sortir. Quand de flot entre sur un arc vide, par exemple, la fonction du flot de sortie devient immédiatement positive même si le flot n'a pas eu le temps de traverser l'arc. Alors il y a de flot qui sort de l'arc au même temps t qu'il entre, leur vitesse de propagation est infinie. Une propagation instantanée est produite parce que quand un flot entre

sur un arc, il est immédiatement distribué sur tout l'arc. La condition de FIFO ne serait plus respectée, puisque des flots qui entrent au même moment ne sortent pas nécessairement en même temps, et il pourrait arriver que des flots entrés à des moments différents sortent ensemble.

D'autres auteurs ont essayé d'éviter l'utilisation de la fonction du flot de sortie en la substituant par une variable de taux de sortie de l'arc a au temps t , $e_a(t)$, qui est relié au taux d'entrée par la contrainte de propagation du flot.

Chapitre 4

Résolution du problème de chargement : deux nouveaux algorithmes

Comme nous l'avons déjà dit, nous nous intéressons au problème d'équilibre dynamique effectif sur les chemins d'un réseau de transport urbain. Nous avons formulé ce modèle d'équilibre au chapitre deux en utilisant les flots et les coûts temporels des chemins. Ainsi, nous voulons trouver les flots temporels sur les chemins, étant donné les taux de départ de chaque origine vers chaque destination, pour une période de temps donnée. Évidemment, les flots sur les chemins sont dépendants des coûts de déplacement encourus et ces coûts vont varier selon les flots. Les algorithmes que nous allons utiliser pour résoudre ce problème sont basés sur des méthodes itératives où à chaque itération nous allons calculer de nouvelles valeurs pour les flots et ces nouveaux flots seront affectés sur leurs chemins ; donc, puisque les flots vont changer, nous devons calculer à nouveau les coûts temporels pour parcourir les chemins du réseau avec des nouvelles conditions de trafic dues au changement des flots. Dans ce chapitre, nous allons développer la façon de résoudre ce problème en utilisant la solution d'un sous-problème, le problème de chargement dynamique de réseau.

Le problème de chargement dynamique consiste à déterminer les flots temporels et le temps de déplacement sur les arcs ainsi que sur les chemins étant donné les taux de départ temporels du trafic sur les chemins du réseau. Nous pouvons le considérer comme un problème d'affectation quotidien ("within-day") pour lequel les usagers utilisent des connaissances ou expériences acquises antérieurement sur le réseau pour choisir leurs parcours à réaliser dans la journée. Mais les parcours peuvent varier d'un jour à l'autre afin de tendre vers un équilibre, et ces variations sont étudiées par les modèles au jour le jour ("day-to-day").

Nous trouvons principalement deux approches, ou classes de méthodes différentes, pour résoudre le problème de chargement de réseau. La première approche est basée sur la simulation (DYNASMART, Jayakrishman et al (1994), CONTRAM, Taylor (1990)). Un des avantages de cette approche de simulation est la description relativement fidèle du trafic, mais, par contre, les calculs sont intensifs. La deuxième approche est basée sur une étude analytique du flot temporel. Friesz et al (1993) présentent une formulation définie comme un système d'équations en fonction du temps de sortie ("exit function") pour définir la propagation du flot sur chaque arc selon les taux de départ, mais ils ne proposent pas une procédure de résolution. Wu (1994) et Wu et al (1998a) proposent une formulation avec une fonction de "stabilité" qui implique que la condition, nécessaire, de FIFO soit vérifiée. Xu et al (1998,1999) fournissent une formulation qui peut être résolue et ils proposent un algorithme DYNALOAD basé aussi sur la fonction de temps de sortie de l'arc et son inverse. Si la condition FIFO est satisfaite, la fonction de sortie est inversible, et si la fonction de sortie et son inverse sont toutes les deux différentiables, alors le problème admet une solution, et l'algorithme atteint cette solution dans un nombre fini d'itérations. Des résul-

tats numériques avec un algorithme DYNALOAD modifié sont reportés par Wu et al (1998c).

Le problème de chargement est un sous-problème important. Il en résulte que c'est la partie la plus coûteuse du problème d'affectation dynamique. Dans un algorithme itératif pour résoudre le problème d'équilibre, le chargement du réseau doit être répété à chaque itération. Il est donc très important d'obtenir des résultats théoriques et numériques qui nous assurent une solution pour le problème de chargement et un temps d'exécution le plus bref possible. Nous allons proposer deux algorithmes : le premier, appelé ici *Algorithme-1*, est une révision et une modification de DYNALOAD avec le but de le rendre plus efficace. Il est basé sur les taux d'entrée et de sortie sur les arcs et la fonction de temps de sortie. Le deuxième consiste en une nouvelle formulation qui utilise les quantités cumulatives de flots qui sont entrées et sorties des arcs (plutôt que les taux). Les fonctions de sortie et leurs inverses sont utilisées aussi, mais leurs dérivées ne sont pas nécessaires. La condition de FIFO est imposée dans les contraintes de propagation du flot. Nous appelons cette procédure l'*Algorithme-2*.

Nous allons exposer l'Algorithme-1 à la section 4.1. À la section 4.2 nous développons l'Algorithme-2. Des résultats numériques sont présentés à la section 4.3.

4.1 Algorithme-1 (variante de DYNALOAD)

Dans cette section, nous allons présenter l'Algorithme-1 pour résoudre le problème de chargement. La formulation mathématique que nous appelons *taux-formulation* est la suivante :

$$v_a(t) = \int_0^t (b_a(s) - e_a(s)) ds, \quad a \in A, \quad 0 \leq t \leq \tau_a(T_a), \quad (4.1)$$

$$\tau_a(t) = t + s_a(v_a(t)), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (4.2)$$

$$\tau'_a(t) = 1 + s'_a(v_a(t)) > 0, \quad a \in A, \quad 0 \leq t \leq T_a, \quad (4.3)$$

$$b_{ka}(t) = \begin{cases} h_k(t) & \text{si } a \text{ est le premier arc du chemin } k, \\ e_{ka-}(t) & \text{sinon,} \end{cases} \quad a \in A, \quad k \in K, \quad 0 \leq t \leq T_{ka}, \quad (4.4)$$

$$e_{ka}(t) = \begin{cases} 0 & 0 \leq t < s_a(0), \\ b_{ka}(\tau_a^{-1}(t))(\tau_a^{-1})'(t) & s_a(0) \leq t \leq \tau_a(T_{ka}), \end{cases} \quad a \in A, \quad k \in K, \quad (4.5)$$

$$b_a(t) = \sum_{k \in K_a} b_{ka}(t), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (4.6)$$

$$e_a(t) = \sum_{k \in K_a} e_{ka}(t), \quad a \in A, \quad 0 \leq t \leq \tau_a(T_a). \quad (4.7)$$

où (4.1) représente l'équation d'état, (4.2) est l'équation de la fonction de temps de sortie de l'arc, (4.3) est la dérivée de la fonction de temps de sortie de l'arc, (4.4) est la contrainte de conservation du flot et (4.5) la contrainte de propagation du flot.

Finalement, pour un chemin $k = a_1 - a_2 - \dots - a_{n_k}$, le temps de parcours effectif pour le flot qui emprunte le chemin au moment t , $S_k(t)$, peut être calculé par la récursion suivante :

$$\begin{cases} t_0 = t; \\ t_j = \tau_{a_j}(t_{j-1}) = t_{j-1} + s_{a_j}(v(t_{j-1})), \quad j = 1, 2, \dots, n_k, \quad t \in [0, T], \quad k \in K; \\ S_k(t) = t_{n_k} - t = \tau_{a_{n_k}}(\dots \tau_{a_2}(\tau_{a_1}(t)) \dots) - t. \end{cases} \quad (4.8)$$

Où t_j est l'instant dans lequel le flot arrive à la tête de l'arc a_j sur le chemin k , et n_k est le nombre d'arcs du chemin. Une manière alternative d'exprimer le temps de parcours effectif est :

$$S_k(t) = \sum_{j=1}^{n_k} s_{a_{k_j}}(v(t_{j-1})) \quad \text{ou} \quad S_k(t) = \tau_{a_{n_k}}(\dots \tau_{a_2}(\tau_{a_1}(t))\dots) - t. \quad (4.9)$$

Il est requis que $v_a(t)$, $b_{ka}(t)$ et $e_{ka}(t)$ soient non négatives.

Les variables $\Pi_{kq}(t)$ et $\Pi_{kp}(t)$, où $(p, q) = a$ et $t \in [0, T]$, sont définies comme le temps de sortie du noeud q du flot qui part de l'origine au moment t sur le chemin k et le temps d'arrivée (ou de sortie) au noeud p du flot qui part à l'instant t sur le chemin k , respectivement. Il est clair que :

$$\Pi_{kq}(t) = \Pi_{kp}(t) + s_a(v_a(\Pi_{kp}(t))) = \tau_a(\Pi_{kp}(t)). \quad (4.10)$$

et

$$v_{ka}(y) = \int_x^y b_{ka}(\xi) d\xi = \int_{\Pi_{kp}^{-1}(x)}^{\Pi_{kp}^{-1}(y)} h_k(t) dt \quad (4.11)$$

où $x = \tau_a^{-1}(y)$

Xu et al. (1999), dans leurs théorèmes 4.1 et 4.2, prouvent l'existence et l'unicité de la solution de cette formulation si les conditions suivantes sont vérifiées :

- À l'instant zéro, il n'y a pas de flot sur le réseau.
- Le flot est conservé, c'est-à-dire qu'il est généré seulement aux noeuds origines et absorbé aux noeuds destinations.
- Le temps pour traverser un arc a par un flot qui arrive à l'arc au moment t est déterminé seulement par le flot qui se trouve sur l'arc a à l'instant t . Ceci implique que les fonctions de temps de parcours s_a sont séparables.

- Les fonctions de temps de parcours des arcs sont positives, non décroissantes, et continûment différentiables par rapport au total de volume sur l'arc.
- Les fonctions de taux de départ (flot de départ) $h_k(t)$ sont non négatives, finies et Lebesgue intégrables sur leur domaine $[0, T]$.
- La condition FIFO sur les arcs doit être satisfaite. Les fonctions de temps de sortie du flot sur les arcs, $\tau(t)$, sont croissantes et inversibles $\tau_a^{-1}(\tau(t)) = t$.

Si une solution peut être calculée, elle a les propriétés suivantes :

- La condition FIFO est satisfaite pour tous les chemins et pour tous les sous-chemins correspondants à l'intérieur de chaque chemin.
- La fonction de temps de sortie d'un noeud d'un chemin, $\Pi_{kp}(t)$, est inversible.
- Il est possible d'établir une relation entre le flot d'entrée dans un arc et le flot de départ des chemins.

Dans l'algorithme DYNALOAD, à chaque itération i , un intervalle de temps est établi, $[l_a^-, l_a^+]$, pour chaque arc $a = (p, q)$ (nous allons appeler cet intervalle *arc-intervalle*) où l_a^- et l_a^+ sont les limites inférieure et supérieure de l'arc-intervalle. Alors $\Pi_{kp}^{-1}(l_a^-)$ et $\Pi_{kp}^{-1}(l_a^+)$ sont évalués afin de calculer un intervalle de temps de départ sur chaque chemin $k \in K^a$ (K^a représente l'ensemble de tous les chemins qui traversent l'arc a). Pour chaque intervalle de temps de départ, une discrétisation de quatre points est faite comme suit :

$$t_{ka}^1 = \Pi_{kp}^{-1}(l_a^-), t_{ka}^3 = \Pi_{kp}^{-1}(l_a^+), t_{ka}^1 = t_{ka}^0 + \frac{t_{ka}^3 - t_{ka}^0}{3}, t_{ka}^2 = t_{ka}^1 + \frac{t_{ka}^3 - t_{ka}^0}{3}. \quad (4.12)$$

Le temps d'entrée dans l'arc a pour le flot qui entre dans chaque chemin $k \in K^a$, à chaque instant t_{ka}^j ($j = 0, 1, 2, 3$), est calculé. Les valeurs de $v_{ka}\{\Pi_{kp}(t_{ka}^j)\}$ sont calculées en utilisant l'équation (4.11). Les $\Pi_{kp}(t_{ka}^j)$ sont des points discrets de l'arc-intervalle, différents

pour chaque $k \in K^a$. Évidemment $\Pi_{kp}(t_{ka}^0) = l_a^-$ et $\Pi_{kp}(t_{ka}^3) = l_a^+$. Les valeurs $\Pi_{kp}^{-1}(t_{ka}^j)$ et $\Pi_{kp}^{-1}(\tau_a^{-1}(t_{ka}^j))$ sont évaluées pour chacun des chemins $k \in K^a$. Elles sont approximées par une fonction polynômiale de degré trois en utilisant une technique de spline cubique. Ainsi, v_{ka} est calculée pour chaque instant $\Pi_{kp}(t_{ka}^0), \Pi_{kp}(t_{ka}^1), \Pi_{kp}(t_{ka}^2), \Pi_{kp}(t_{ka}^3)$, et conservée pour chaque $k \in K^a$. Tous ces calculs sont répétés pour chaque chemin qui traverse l'arc a (et pour chaque arc $a \in A$). Pour calculer v_a , les calculs sont exécutés $|K^a|$ fois pour chaque chemin $k \in K^a$.

Dans l'Algorithme-1, nous utilisons une nouvelle stratégie pour réaliser la discrétisation dans chaque arc-intervalle, avec le but de réduire la quantité de calculs et de les simplifier, c'est à dire avec l'objectif de réduire le temps d'exécution de l'algorithme. La discrétisation est faite de la manière suivante :

Soit l'arc-intervalle i_a , ($i_a = 1, 2, 3, \dots$), $[l_a^-, l_a^+]$, où $l_a^+ = \tau_a(l_a^-)$ et soient $t_{i_a}^j$,

($j = 0, 1, 2, 3$) des points dans l'intervalle i_a qui sont définis comme suit :

- Si l'arc a est rencontré pour la première fois, $i_a = 1$:

$$t_1^0 = l_a^-, t_1^3 = l_a^+, t_1^1 = t_1^0 + \frac{t_1^3 - t_1^0}{3}, t_1^2 = t_1^1 + \frac{t_1^3 - t_1^0}{3}; \quad (4.13)$$

- sinon, pour chaque arc-intervalle i_a :

$$t_{i_a}^j = \tau_a(t_{i_a-1}^j); \quad (4.14)$$

Les points $t_{i_a}^j$ sont définis en fonction de l'arc a et sont les mêmes pour tous les chemins qui le traversent, $k \in K^a$.

Puisque les points $t_{i_a}^j$ sont les mêmes pour chaque chemin $k \in K^a$, l'équation (4.11) est utilisée une seule fois, pour chaque chemin, pour calculer $v_{ka}(t_{i_a}^j)$. Les variables v_a, Π_{kp}^{-1}

et $\Pi_{kp}^{-1}(\tau_a^{-1}(t_{i_a}^j))$ sont calculées pour chaque point $t_{i_a}^j$. Puisque $t_{i_{a+1}}^j = \tau_a(t_{i_a}^j)$, alors :

$$\Pi_{kq}^{-1}(t_{i_{a+1}}^j) = \Pi_{kp}^{-1}(t_{i_a}^j); \quad (4.15)$$

$$\Pi_{kp}^{-1}(\tau_a(t_{i_{a-1}}^j)) = \Pi_{kp}^{-1}(\tau_a^{-1}(\tau_a(t_{i_{a-1}}^j))). \quad (4.16)$$

Les arc-intervalles sont différents pour chaque arc, donc, l'ensemble de points discrets pour l'arc a^+ , $\{t_{i_{a+}}^j\}$, est différent de celui de l'arc a , $\{t_{i_a}^j\}$. Donc, le temps discret $t_{i_{a+1}}^j$ n'appartient nécessairement pas à l'ensemble $\{t_{i_a}^j\}$ et, par conséquent, les valeurs $\Pi_{kq}^{-1}(\tau_a(t_{i_{a-1}}^j))$ sont approximées à partir de $\Pi_{kp}^{-1}(t_{i_a}^j)$. Pour ce faire, nous utilisons un polynôme de Lagrange d'ordre 3.

$$\Pi_{kq}^{-1}(t) = \sum_{j=0}^3 \Pi_{kp}^{-1}(t_{i_a}^j) L_j(t), \quad t \in [t_{i_a}^0, t_{i_a}^3]$$

$$L_j(t) = \frac{\prod_{m=0, m \neq j}^3 (t - t_{i_a}^m)}{\prod_{m=0, m \neq j}^3 (t_{i_a}^j - t_{i_a}^m)}, \quad j = 0, 1, 2, 3.$$

Pour réaliser les approximations nous utilisons des techniques d'interpolation en utilisant la méthode du polynôme de Lagrange; des méthodes du type spline cubique ont été utilisées pour des interpolations semblables par Xu et al. (1999). Nous avons remarqué que l'utilisation du polynôme de Lagrange donne des approximations plus précises.

Une fois que v_a a été calculée, nous calculerons s_a et τ_a , et l'arc-intervalle courant sera mis à jour de la façon suivante : $l_a^+ := l_a^-$ et $l_a^+ := \tau_a(l_a^+)$; L'ensemble des arcs \bar{A} contiendra les arcs $a \in A$ qui sont "actifs" à chaque itération, c'est-à-dire les arcs qui portent du flot. Le temps du système T_i sera incrémenté à la plus petite valeur entre les limites supérieures des arc-intervalles courants de tous les arcs de l'ensemble \bar{A} , $T_i = \min\{l_a^+ : a \in \bar{A}\}$, et nous sélectionnerons les arcs tels que la valeur de l_a^+ courante sera égale à T_i , $A_i = \arg \min\{l_a^+ : a \in \bar{A}\}$.

Les détails de l'Algorithme-1 sont énoncés dans le suivant :

Algorithme-1

1. INITIALISATION

$i := 0, T_0 := 0;$

pour $a \in A$ **faire**

$l_a^- := 0;$

$l_a^+ := s_a(0);$

$i_a := 0;$

$\rho_a :=$ le nombre de chemins $k \in K^a$ tel que a est l'arc initial

fin de pour

$\bar{A} := \{a \in A : \rho_a > 0\}$

2. BOUCLE PRINCIPALE

tant que $T_i < T$ **ou** $\bar{A} \neq \emptyset$ **faire**

$i := i + 1$

$T_i := \min\{l_a^+ : a \in \bar{A}\}$

$A_i := \arg \min\{l_a^+ : a \in \bar{A}\}$

pour $a \in A_i$ **faire**

pour $k \in K^a$ **faire**

si k est " active" dans l'arc-intervalle $[l_a^-, l_a^+]$ **alors**

ARC DYNAMICS(a, k) sur l'arc-intervalle $[l_a^-, l_a^+]$

fin de si

fin de pour

mise à jour $l_a^- := T_i$ et $l_a^+ := \tau_a(T_i)$

fin de pour

si $\rho_a = 0$ **alors**

$\bar{A} := \bar{A} - \{a\}$

fin de si

fin de tant que

3. **ARC DYNAMICS**(a, k) sur l'arc-intervalle $[l_a^-, l_a^+]$

déterminer($t_{i_a}^0, t_{i_a}^1, t_{i_a}^2, t_{i_a}^3$)

pour $j = 0, 1, 2, 3$ **faire**

calculer $\Pi_{kp}^{-1}(t_{i_a}^j)$.

calculer $v_a(t_{i_a}^j)$ en utilisant (4.11)

calculer $t_{i_a+1}^j = \tau_a(t_{i_a}^j) = t_{i_a}^j + s_a(v_a(t_{i_a}^j))$

fin de pour

si $a \notin \bar{A}$ **et** a n'est pas le dernier arc du chemin k **alors**

$\bar{A} := \bar{A} + \{\text{l'arc suivant de } a \text{ au chemin } k\}$

fin de si

si $l_a^+ = T_{k_a}$ **alors**

$\rho_a := \rho_a - 1$

fin de si

4.2 Algorithme-2

Dans cette section, nous allons présenter une autre formulation mathématique pour le problème de chargement. Celle-ci est une reformulation du problème en utilisant les quantités cumulatives de flots, plutôt que le flot même, qui sont entrés et sortis des arcs. La dérivée de la fonction de temps de sortie des arcs et de son inverse ne sont plus nécessaires. Avec ces quantités cumulatives, le délai de déplacement et les flots sur les arcs peuvent être facilement calculés.

La formulation est la suivante :

$$v_a(t) = B_a(t) - E_a(t), \quad a \in A, \quad 0 \leq t \leq \tau_a(T_a), \quad (4.17)$$

$$\tau_a(t) = t + s_a(v_a(t)), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (4.18)$$

$$B_{ka}(t) = \begin{cases} \int_0^t h_k(y) dy, & \text{si } a \text{ est le premier arc du chemin } k, \\ E_{ka^-}(t), & \text{sinon,} \end{cases} \quad a \in A, \quad k \in K, \quad 0 \leq t \leq T_{ka}, \quad (4.19)$$

$$E_{ka}(t) = \begin{cases} 0, & 0 \leq t \leq s_a(0), \\ B_{ka}(\tau_a^{-1}(t)), & s_a(0) \leq t \leq \tau_a(T_{ka}), \end{cases} \quad a \in A, \quad k \in K, \quad (4.20)$$

$$B_a(t) = \sum_{k \in K^a} B_{ka}(t), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (4.21)$$

$$E_a(t) = \sum_{k \in K^a} E_{ka}(t), \quad a \in A, \quad s_a(0) \leq t \leq \tau_a(T_a), \quad (4.22)$$

où (4.17) représente l'équation d'état, (4.18) est l'équation de temps de sortie de l'arc, (4.19) est la contrainte de conservation, (4.20) la contrainte de propagation. Il est requis que $v_a(t)$, $B_{ka}(t)$ et $E_{ka}(t)$ soient non négatives.

Dans l'algorithme-2 nous allons établir des arc-intervalles de temps, $[l_a^-, l_a^+]$, comme dans l'algorithme précédent. Nous allons discrétiser chacun des ces arc-intervalles en 4 points de la même façon décrite pour l'Algorithme-1, c'est-à-dire : si l'arc a est rencontré pour la première fois :

$$t_1^0 = l_a^-, t_1^3 = l_a^+, t_1^1 = t_1^0 + \frac{t_1^3 - t_1^0}{3}, t_1^2 = t_1^1 + \frac{t_1^3 - t_1^0}{3}; \quad (4.23)$$

sinon, pour chaque arc-intervalle $i_a : t_{i_a}^j = \tau_a(t_{i_a-1}^j)$, les points $t_{i_a}^j$ sont les mêmes pour tous les chemins qui traversent l'arc a , c'est-à-dire $\forall k \in K^a$. Il faut remarquer que

$$E_a(t_{i_a}^j) = B_a(t_{i_a-1}^j) \quad (4.24)$$

et

$$v_a(t_{i_a}^j) = B_a(t_{i_a}^j) - E_a(t_{i_a}^j) = B_a(t_{i_a}^j) - B_a(t_{i_a-1}^j), \quad (4.25)$$

donc la formulation discrétisée devient :

$$v_a(t_{i_a}^j) = \begin{cases} B_a(t_{i_a}^j) & \text{si } i_a = 1, \\ B_a(t_{i_a}^j) - B_a(t_{i_a-1}^j) & \text{sinon,} \end{cases} \quad \forall a \in A, 0 \leq t_{i_a-1}^j \leq \tau_a(T_a), \quad (4.26)$$

$$t_{i_a+1}^j = t_{i_a}^j + s_a(v_a(t_{i_a}^j)), \quad a \in A, 0 \leq t_{i_a}^j \leq \tau_a(T_a), \quad (4.27)$$

$$\forall a \in A, k \in K, 0 \leq t \leq T_{ka},$$

$$B_{ka}(t_{i_a}^j) = \begin{cases} \int_0^{t_{i_a}^j} h_k(y) dy, & \text{si } a \text{ est le premier arc du chemin } k, \\ B_{ka^-}(\tau_{a^-}^{-1}(t_{i_a}^j)), & \text{sinon,} \end{cases} \quad (4.28)$$

$$B_a(t_{i_a}^j) = \sum_{k \in K^a} B_{ka}(t_{i_a}^j), \quad a \in A, 0 \leq t_{i_a}^j \leq T_a, \quad (4.29)$$

Donc, l'Algorithme-2 détermine d'abord $B_{ka}(t)$ sur les points discrétisés de l'arc-intervalle courant pour tous les chemins $k \in K^a$. Si a est le premier arc du chemin k , $B_{ka}(t)$ sera calculé directement ; sinon, comme les arc-intervalles correspondants aux arcs a et a^- sont différents, le temps discret $t_{i_a}^j$ n'appartient pas nécessairement à l'ensemble $\{t_{i_a^-}^j\}$, et par conséquent, la valeur $B_{ka^-}(\tau_{a^-}^{-1}(t_{i_a}^j))$ doit être approximée. Pour le faire, nous allons chercher l'intervalle

i_{a-} tel que $t_{i_{a-}}^0 \leq t_{i_{a-}}^j \leq t_{i_{a-}}^3$ alors la valeur que nous voulons calculer, $B_{ka-}(\tau_{a-}^{-1}(t_{i_a}^j))$ est comprise entre $B_{ka-}(t_{i_{a-}}^0)$ et $B_{ka-}(t_{i_{a-}}^3)$. Ensuite nous utiliserons une technique d'approximation polynômiale de Lagrange (polynôme de Lagrange d'ordre 3) :

$$B_{ka-}(t) = \sum_{j=0}^3 B_{ka-}(t_{i_{a-}}^j) L_j(t), \quad t \in [t_{i_{a-}}^0, t_{i_{a-}}^3]$$

$$L_j(t) = \frac{\prod_{m=0, m \neq j}^3 (t - t_{i_{a-}}^m)}{\prod_{m=0, m \neq j}^3 (t_{i_{a-}}^j - t_{i_{a-}}^m)}, \quad j = 0, 1, 2, 3.$$

Après, B_a , v_a seront calculés par (4.29) et (4.26) respectivement et ensuite nous pouvons calculer s_a .

Dans le but de prouver l'existence et l'unicité de la solution pour cette formulation avec flot accumulé, nous avons besoin de démontrer que (4.17)-(4.22) sont équivalents à la formulation avec taux (4.1)-(4.7). Ceci peut être prouvé par induction sur les arcs de chaque chemin. Ensuite nous pourrons utiliser les théorèmes 4.1 et 4.2 de Xu et al. (1999) afin d'obtenir le résultat désiré. D'abord, nous pouvons voir que $B_{ka}(t)$ et $E_{ka}(t)$, dans la formulation avec le flot accumulé, sont égales à $\int_0^t b_{ka}(y) dy$ et $\int_0^t e_{ka}(y) dy$ respectivement dans la formulation avec taux .

Proposition 1 *La formulation avec taux cumulatif (4.17)-(4.22) pour résoudre le problème de chargement sur un réseau est équivalente à la formulation avec taux (4.1)-(4.7).*

Preuve.

- Soit a , le premier arc du chemin k . Par (4.4) nous avons :

$$b_{ka}(t) = h_k(t) \implies \int_0^t b_{ka}(y) dy = \int_0^t h_k(y) dy; \quad (4.30)$$

par (4.19) et (4.30)

$$B_{ka}(t) = \int_0^t b_{ka}(y) dy. \quad (4.31)$$

Également par (4.5) nous avons :

$$\int_0^t e_{ka}(y)dy = \int_0^t b_{ka}(\tau_a^{-1}(y))(\tau_a^{-1})'(y)dy = B_{ka}(\tau_a^{-1}(y)); \quad (4.32)$$

et par (4.20) et (4.32) il suit que :

$$E_{ka}(t) = \int_0^t e_{ka}(y)dy. \quad (4.33)$$

- Si a n'est pas le premier arc du chemin k , et les équations $B_{k^-}(t) = \int_0^t b_{k^-}(y)dy$ et $E_{k^-}(t) = \int_0^t e_{k^-}(y)dy$ sont vérifiées pour l'arc a^- , prédécesseur de a , sur le chemin k , alors, comme avant, par (4.4) nous avons :

$$\int_0^t b_{ka}(y)dy = \int_0^t e_{k^-}(y)dy = E_{k^-}(t); \quad (4.34)$$

Par (4.19) et (4.34), il suit que

$$B_{ka}(t) = \int_0^t b_{ka}(y)dy, \quad (4.35)$$

et par (4.5), (4.20) et (4.32), il en résulte que

$$E_{ka}(t) = \int_0^t e_{ka}(y)dy. \quad (4.36)$$

Alors (4.19) et (4.20) sont équivalents à (4.4) et (4.5) respectivement, et (4.17) est équivalent à (4.1). □

Il est important de remarquer que les contraintes de propagation du flot nous assurent la condition FIFO, puisque $B_a(t) = E_a(\tau_a(t))$ implique que $b_a(t) = e_a(\tau_a(t))(\tau_a(t))' \implies (\tau_a(t))' = \frac{b_a(t)}{e_a(\tau_a(t))} > 0$, c'est-à-dire que $\tau_a(t)$ est croissante.

L'Algorithme-2 s'énonce comme suit :

Algorithme-2
1. INITIALISATION
 $i := 0, T_0 := 0;$
pour $a \in A$ **faire**
 $l_a^- := 0;$
 $l_a^+ := s_a(0);$
 $i_a := 0;$
 $\rho_a :=$ le nombre de chemins $k \in K^a$ tel que a est l'arc initial

fin de pour
 $\bar{A} := \{a \in A : \rho_a > 0\}$
2. BOUCLE PRINCIPALE
tant que $T_i < T$ **ou** $\bar{A} \neq \emptyset$ **faire**
 $i := i + 1$
 $T_i := \min\{l_a^+ : a \in \bar{A}\}$
 $A_i := \arg \min\{l_a^+ : a \in \bar{A}\}$
pour $a \in A_i$ **faire**
pour $k \in K^a$ **faire**
si k est "active" dans l'arc-intervalle $[l_a^-, l_a^+]$ **alors**
 $i_a = i_a + 1$

 ARC DYNAMICS(a, k) sur "l'arc-active" $[l_a^-, l_a^+]$
fin de si
fin de pour

 mise à jour $l_a^- := T_i$ et $l_a^+ := \tau_a(T_i)$

fin de pour

si $\rho_a = 0$ **alors**

$$\bar{A} := \bar{A} - \{a\}$$

fin de si

fin de tant que

3. **ARC DYNAMICS**(a, k) sur l'“ arc-active” $[l_a^-, l_a^+]$

déterminer $(t_{i_a}^0, t_{i_a}^1, t_{i_a}^2, t_{i_a}^3)$

pour $j = 0, 1, 2, 3$ **faire**

calculer $B_{ka}(t_{i_a}^j)$ et $B_a(t_{i_a}^j)$ en utilisant (4.28) et (4.29)

calculer $v_a(t_{i_a}^j)$ en utilisant (4.26)

calculer $t_{i_a+1}^j = \tau_a(t_{i_a}^j) = t_{i_a}^j + s_a(v_a(t_{i_a}^j))$

fin de pour

si $a \notin \bar{A}$ **et** a n'est pas le dernier arc du chemin k **alors**

$$\bar{A} := \bar{A} + \{\text{l'arc suivant de } a \text{ au chemin } k\}$$

fin de si

si $l_a^+ = T_{ka}$ **alors**

$$\rho_a := \rho_a - 1$$

fin de si

4.3 Résultats numériques

Dans cette section, nous allons présenter trois exemples numériques de résolution du problème de chargement. Nous allons comparer le temps d'exécution entre les deux algo-

rithmes énoncés et la dernière modification de DYNALOAD présentée par Wu, Florian et Rubio-Ardanaz (1998), que nous allons appeler Algorithme-3.

	Algorithme-1	Algorithme-2	Algorithme-3
formulation	flot de véhicules	flot accumulé	flot de véhicules
fonction du temps de sortie	sur les noeuds des chemins : Π_{kp} ; et sur les arcs : τ_a	sur les arcs : τ_a	sur les noeuds des chemins : Π_{kp} ; et sur les arcs : τ_a
variables	$v_{ka}, v_a, \Pi_{kp}, \Pi_{kp}^{-1}, h_k$	v_a, τ_a, B_{ka}, B_a	$v_{ka}, v_a, \Pi_{kp}, \Pi_{kp}^{-1}, h_k$
discrétisation	sur le temps de sortie de chaque arc	sur le temps de sortie de chaque arc	sur le temps d'entrée de chaque chemin
	arc-intervalle	arc-intervalle	arc-chemin-intervalle
	sur chaque arc	sur chaque arc	sur chaque arc-chemin
calcul de variables	un fois sur chaque chemin $k \in K^a$	un fois sur chaque chemin $k \in K^a$	$ K^a $ fois sur chaque chemin $k \in K^a$

TAB. 4.1 – Sommaire comparatif de Algorithme-1, Algorithme-2 et Algorithme-3.

Les exemples ont été réalisés en utilisant deux réseaux différents, le réseau de la ville Sioux Falls de 76 arcs et le réseau de la ville de Hull de 798 arcs. Les essais numériques ont été réalisés sur une station de travail SUN Microsystems ULTRA SPARC 10, 300 Mhz, 128 M. ram, Unix Solaris. Les dimensions des ces deux réseaux ainsi que les temps de calcul sont

donnés au tableau (4.2).

exemple	nombre d'arcs	nombre de chemins	période de départ	temps CPU		
				Algorithmme-1	Algorithmme-2	Algorithmme-3
1	76	552	120	6.17	5.95	58.03
2	798	465	30	18.88	13.87	92.28
3	798	465	60	29.55	24.77	174.33

TAB. 4.2 – Réseaux et temps CPU.

Le premier exemple est le réseau de la ville de Sioux Falls (même réseau utilisé par Suwansirikul et al. 1987) constitué de 24 noeuds, 76 arcs et 552 chemins. Tous les chemins sont générés à l'avance pour réaliser les calculs. La période de départ est $[0, 120]$, $T = 120$ et le flot sur chaque chemin est défini par une fonction quadratique donnée par : $h(t) = 0.0005 \times (T - t) \times t$, $t = 0, \dots, 120$, $\forall k$. La fonction de temps de parcours choisie est $s_a(v_a(t)) = \beta_a^0 + \beta_a^1 \times v_a(t)$, $\forall a \in A$. Les paramètres β sont donnés au tableau (4.3). Les temps CPU totaux sont de 5.95 secondes pour l'Algorithmme-2, 6.17 secondes pour l'Algorithmme-1 et 58.03 pour l'Algorithmme-3.

Le deuxième exemple est le réseau de la ville de Hull (Canada) constitué de 501 noeuds, 798 arcs et 465 chemins. Deux périodes de temps de départ sont utilisées, d'abord $T = 30$, donc la période de départ sera $[0, 30]$ et deuxièmement $T = 60$, période de départ $[0, 60]$. La fonction de temps de parcours choisie est : $s_a(v_a(t)) = 1.2 + 0.01 \times v_a(t)$, $\forall a \in A$. Pour $T = 30$, la fonction de taux de départ est : $h(t) = 0.010 \times (T - t) \times t$, $t = 0, \dots, 30$, $\forall k$.

Les temps CPU totaux sont 13.87 secondes pour l'Algorithme-2, 18.88 pour l'Algorithme-1 et 92.28 pour l'Algorithme-3. Pour $T = 60$, la fonction du flot de départ est donnée par : $h(t) = 0.0020 \times (T - t) \times t, t = 0, \dots, 60, \forall k$. Les temps CPU sont de 24.77 secondes pour l'Algorithme-2, 29.55 pour l'Algorithme-1 et 174.33 pour l'Algorithme-3 respectivement.

Dans le tableau (4.2), nous pouvons voir que les algorithmes que nous avons énoncés ont grandement amélioré la performance du précédent (par rapport au temps d'exécution). La figure (4.1) montre les courbes du volume cumulatif de véhicules qui sont entrés dans chaque arc du chemin 1 par rapport au temps de départ du deuxième exemple. La dernière courbe représente le volume cumulatif qui est sorti du dernier arc du chemin. Avec l'utilisation du flot cumulatif, il est très simple de calculer le temps de parcours des arcs et le temps effectif de parcours des chemins ainsi que le volume sur les arcs. Sur le chemin 1, figure (4.1), le flot qui est entré dans le premier arc au temps t_1 , est entré dans le deuxième arc au temps t_2 ; c'est-à-dire il a quitté l'arc 1 au moment t_2 , et alors le temps pour parcourir le premier arc pour le flot qui entre au temps t_1 est égal à $t_2 - t_1, s_{a_1} = t_2 - t_1$. Le temps effectif du parcours pour le chemin, du flot qui part au temps t_1 , est égal à $t_3 - t_1$. Le volume, correspondant au chemin 1 qui se trouve dans le premier arc au temps t_1 , est égal à $B_2 - B_1$.

From-node	To-node	β^0	β^1	From-node	To-node	β^0	β^1
1	2	3.6	0.01	13	24	2.4	0.01
1	3	2.4	0.01	14	11	2.4	0.01
2	1	3.6	0.01	14	15	3.0	0.01
2	6	3.0	0.01	14	23	2.4	0.01
3	1	2.4	0.01	15	10	3.6	0.01
3	4	2.4	0.01	15	14	3.0	0.01
3	12	2.4	0.01	15	19	2.4	0.01
4	3	2.4	0.01	15	22	2.4	0.01
4	5	1.2	0.01	16	8	3.0	0.01
4	11	3.6	0.01	16	10	3.0	0.01
5	4	1.2	0.01	16	17	1.2	0.01
5	6	2.4	0.01	16	18	1.8	0.01
5	9	3.0	0.01	17	10	4.8	0.01
6	2	3.0	0.01	17	16	1.2	0.01
6	5	2.4	0.01	17	19	1.2	0.01
6	8	1.2	0.01	18	7	1.2	0.01
7	8	1.8	0.01	18	16	1.8	0.01
7	18	1.2	0.01	18	20	2.4	0.01
8	6	1.2	0.01	19	15	2.4	0.01
8	7	1.8	0.01	19	17	1.2	0.01
8	9	6.0	0.01	19	20	2.4	0.01
8	16	3.0	0.01	20	18	2.4	0.01
9	5	3.0	0.01	20	19	2.4	0.01
9	8	6.0	0.01	20	21	3.6	0.01
9	10	1.8	0.01	20	22	3.0	0.01
10	9	1.8	0.01	21	20	3.6	0.01
10	11	3.0	0.01	21	22	1.2	0.01
10	15	3.6	0.01	21	24	1.8	0.01
10	16	3.0	0.01	22	15	2.4	0.01
10	17	4.8	0.01	22	20	3.0	0.01
11	4	3.6	0.01	22	21	1.2	0.01
11	10	3.0	0.01	22	23	2.4	0.01
11	12	3.6	0.01	23	14	2.4	0.01
11	14	2.4	0.01	23	22	2.4	0.01
12	3	2.4	0.01	23	24	1.2	0.01
12	11	3.6	0.01	24	13	2.4	0.01
12	13	1.8	0.01	24	21	1.8	0.01
13	12	1.8	0.01	24	23	1.2	0.01

TAB. 4.3 – Paramètres utilisés dans l'exemple 1.

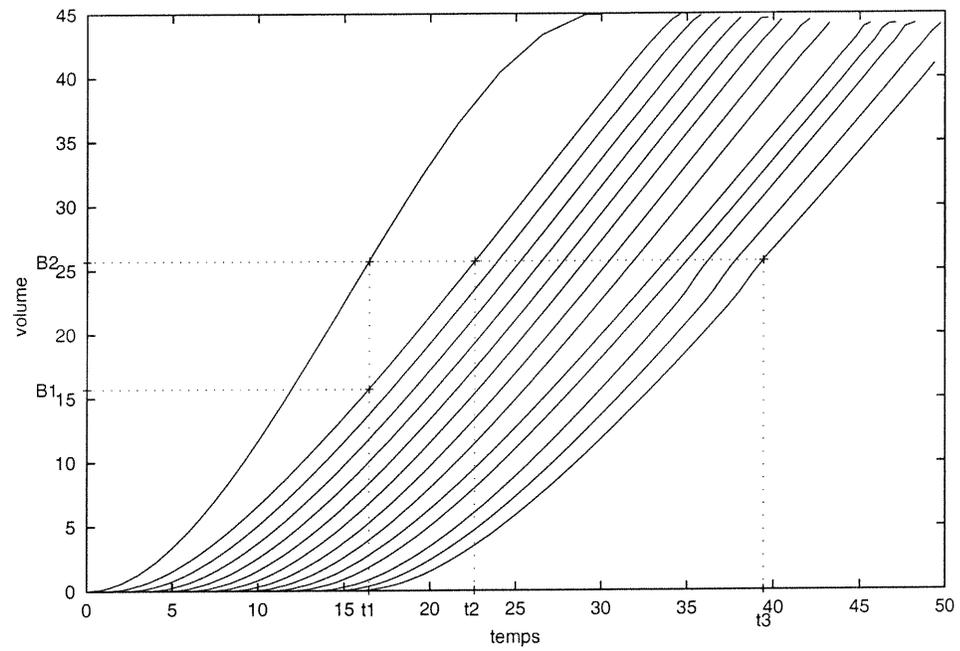


FIG. 4.1 – Réseau de Hull : volume accumulé pour chaque arc du chemin 1.

Chapitre 5

Résolution du problème d'équilibre dynamique

Nous allons exposer dans ce chapitre les méthodes que nous avons employées pour résoudre le problème d'affectation dynamique. Étant donné les demandes des taux temporels, nous allons distribuer les flots entre les chemins générés qui relient les origines aux destinations de telle sorte qu'un équilibre temporel soit vérifié. Au chapitre deux, nous avons décrit une généralisation des principes d'équilibre de Wardrop : le principe d'équilibre dynamique normatif et le principe d'équilibre dynamique descriptif. Nous nous intéressons au principe descriptif avec le temps de parcours effectif sur les chemins. Nous avons défini (définitions 3 et 4 du chapitre 2) cet équilibre sur le domaine des arcs et des chemins.

Nous utilisons un algorithme itératif de projection où, à chaque itération, nous allons calculer un "pas" de projection pour trouver une nouvelle solution réalisable à partir de la solution courante. Cette nouvelle solution représente une nouvelle distribution des flots qui respecte toutes les contraintes du problème. Une fois que les nouveaux flots des chemins sont obtenus, ils doivent être chargés sur le réseau afin de trouver les volumes et les coûts temporels pour la prochaine itération.

Le grand nombre de chemins qui existent entre une origine et une destination est une complication classique des algorithmes de solution dans l'espace des chemins. Le fait de travailler dans l'espace des arcs serait avantageux, mais la structure du problème dynamique ne permet pas un passage d'une solution réalisable à une autre dans l'espace des arcs et rend le problème difficile. Comment trouver une nouvelle solution sur les arcs telle que les équations qui définissent le problème de chargement soient satisfaites ? Le seul moyen d'aborder ce problème est de travailler dans l'espace des chemins.

Nous allons résoudre d'abord le problème d'équilibre dynamique en ne tenant compte que d'un certain nombre de chemins qui seront fixés a priori. Après, nous allons généraliser l'algorithme en résolvant le problème sur tout l'ensemble de solutions réalisables, c'est-à-dire sans se restreindre à certains chemins fixés à l'avance. Pour ce faire, nous allons utiliser une stratégie de restriction qui résout le problème sur un domaine réduit et qui sera élargi au fur et à mesure. Le domaine réalisable est élargi par la génération de nouveaux chemins en résolvant le problème du plus court chemin temporel (ou dynamique).

Nous allons définir et formuler le problème d'inégalité variationnelle, à la section 5.1, tel que nous allons le traiter ; à la section 5.2, nous allons exposer la méthode de projection et nous présenterons, à la section 5.3, certains résultats numériques de son application. Nous présenterons la méthode de restriction à la section 5.4 et certains résultats numériques obtenus avec cette méthode seront fournis à la section 5.5.

5.1 Formulation du problème d'inégalité variationnelle

Nous pouvons formuler mathématiquement les conditions d'équilibre dynamique normalif comme suit :

$$S_k(h^*(t)) \begin{cases} = \mu_i(t) & \text{si } h_k^*(t) > 0; \\ \geq \mu_i(t) & \text{sinon.} \end{cases} \quad (5.1)$$

$\forall k \in K_i, i \in I$, pour presque tout $t \in (0, T)$ et

$$\mu_i(t) = \min_{k \in K_i} \{S_k(h^*(t))\}, \text{ pour presque tout, } t \in (0, T). \quad (5.2)$$

Cette condition nous assure qu'aucun usager ne pourrait réduire son temps effectif de voyage.

Le problème correspondant d'inégalité variationnelle, PIV, est défini comme suit : trouver $h^*(t) \in \Omega(t)$ tel que

$$(S(h^*(t)), h(t) - h^*(t)) \geq 0, \forall h(t) \in \Omega(t), \text{ pour presque tout, } t \in (0, T). \quad (5.3)$$

Nous pouvons exprimer (5.3) comme suit :

$$\int_0^T \sum_{i \in I} \sum_{k \in K_i} S_k(h^*(t)) [h_k(t) - h_k^*(t)] dt \geq 0, \forall h(t) \in \Omega(t). \quad (5.4)$$

Nous allons utiliser (5.3) pour trouver les conditions d'équilibre (5.1), mais avant nous allons énoncer la proposition suivante :

Proposition 2 *Si le problème d'inégalité variationnelle (5.3) et les conditions d'équilibre dynamique descriptif effectif (5.1) ont des solutions, alors les deux problèmes sont équivalents.*

Preuve.

- Nous allons prouver d'abord que les conditions d'équilibre (5.1) impliquent le PIV (5.3) : par définition nous avons les deux relations suivantes :

$$[S_k(h^*(t)) - \mu_i^*(t)] h_k(t) \geq 0, \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (5.5)$$

$$[S_k(h^*(t)) - \mu_i^*(t)]h_k^*(t) = 0, \quad \forall k \in K_i, i \in I, \text{ pour presque tout } t \in [0, T], \quad (5.6)$$

où $h_k(t) \in \Omega(t)$. En soustrayant (5.6) de (5.5), nous obtenons :

$$[S_k(h^*(t)) - \mu_i^*(t)][h_k(t) - h_k^*(t)] \geq 0, \quad (5.7)$$

ceci implique que

$$\begin{aligned} 0 &\leq \int_0^T \sum_{i \in I} \sum_{k \in K_i} [S_k(h^*(t)) - \mu_i^*(t)][h_k(t) - h_k^*(t)] dt \\ &= \int_0^T \sum_{i \in I} \sum_{k \in K_i} S_k(h^*(t))[h_k(t) - h_k^*(t)] dt - \\ &\quad \int_0^T \sum_{i \in I} \mu_i^*(t) \sum_{k \in K_i} [h_k(t) - h_k^*(t)] dt \\ &= \int_0^T \sum_{i \in I} \sum_{k \in K_i} S_k(h^*(t))[h_k(t) - h_k^*(t)] dt. \end{aligned} \quad (5.8)$$

La deuxième égalité est due à la conservation du flot.

- Nous allons montrer que le PIV implique les conditions d'équilibre (5.1) :

supposons qu'il existe un chemin, $k_1 \in K_i$, tel qu'il existe un moment de départ $t' \in (0, T)$, pour lequel $S_{k_1}^*(t') > \mu_i(t')$ et $h_{k_1}(t') > 0$. Soit k_2 , un autre chemin, tel que $k_2 \in K_i$ et $S_{k_2}(h^*(t')) = \mu_i(t')$; prenons la solution $h(t)$ égale à $h^*(t)$ pour tout t sauf pour t' , et

$$h_{k_1}(t') = 0 \quad \text{et} \quad h_{k_2}(t') = h_{k_2}^*(t') + h_{k_1}^*(t').$$

Il est évident que $S_k(h^*(t))[h_k(t) - h_k^*(t)]$ est égal à zéro pour tout t sauf pour t' . À partir de (5.3), nous avons :

$$\begin{aligned} 0 &\leq \sum_{i \in I} \sum_{k \in K_i} S_k(h^*(t'))[h_k(t') - h_k^*(t')] \\ &= S_{k_1}(h^*(t'))[h_{k_1}(t') - h_{k_1}^*(t')] + S_{k_2}(h^*(t'))[h_{k_2}(t') - h_{k_2}^*(t')] \\ &= [S_{k_2}(h^*(t')) - S_{k_1}(h^*(t'))]h_{k_1}^* \\ &= [\mu_i(t') - S_{k_1}(h^*(t'))]h_{k_1}^*. \end{aligned}$$

Ceci implique que $\mu_i(t') = S_{k_1}(h^*(t'))$ si $h_{k_1}^* > 0$. Ceci complète la preuve. \square

Ran et Boyce (1994) ainsi que Xu et al. (1999) montrent aussi l'équivalence de formulations d'inégalité variationnelle et les conditions d'équilibre.

Nous procédons à une discrétisation du problème. La période de temps de départ $[0, T]$ est divisée en M petits intervalles. La longueur de chaque intervalle est $\frac{T}{M}$ et chacun d'entre eux sera indexé par m . Nous allons définir $h(m)$ comme les taux de départ sur les chemins dans l'intervalle m , $S(h(m))$ les temps effectifs de parcours des usagers qui partent dans l'intervalle m , $g(m)$ les taux des demandes dans l'intervalle m , $\Omega(m) = \{h(m) \mid \sum_{k \in K_i} h_k(m) = g_i(m), h_k(m) \geq 0, \forall k \in K_i, \forall i \in I\}$. Ainsi, nous pouvons reformuler (5.3) comme le problème de trouver $h^*(m) \in \Omega(m)$, $m = 1, 2, \dots, M$, tel que :

$$(S(h^*(m)), h(m) - h^*(m)) \geq 0, \forall h(m) \in \Omega(m). \quad (5.9)$$

5.2 Méthode de projection

Dans cette section, nous allons présenter la méthode utilisée pour résoudre le problème dynamique d'équilibre quand les chemins disponibles sont déterminés a priori. Nous présenterons une méthode de projection que nous avons utilisée pour résoudre le PIV. Cette méthode a été utilisée par Wu et al. (1998b). La fonction de *gap de Fukushima* (Fukushima 1989) est employée pour résoudre le problème quadratique suivant :

$$\min_{h^{l+1} \in \Omega} (S(h^l), h^{l+1} - h^l) + \frac{1}{2\alpha} (h^{l+1} - h^l, h^{l+1} - h^l) \quad (5.10)$$

ou l est l'indice d'itération, α est une constante positive, h^l est le taux de départ et $S(h^l)$ le temps de parcours sur les chemins courants; $S(h^l)$ est déterminé par le chargement dynamique du réseau. Nous pouvons réécrire (5.10) comme suit :

$$\min_{h^{l+1} \in \Omega} \sum_{i \in I} \sum_{k \in K_i} \int_0^T [S_k(h^l)(t)(h_k^{l+1}(t) - h_k^l(t)) + \frac{1}{2\alpha} (h_k^{l+1}(t) - h_k^l(t))^2]. \quad (5.11)$$

Nous allons discrétiser le temps en divisant la période en M intervalles de la même longueur, $m = 1, \dots, M$. Les temps discrets seront déterminés par le point qui est au milieu de l'intervalle. Ainsi nous aurons :

$$\min \sum_m \sum_{i \in I} \sum_{k \in K_i} [S_k(h^l(m))(h_k^{l+1}(m) - h_k^l(m)) + \frac{1}{2\alpha}(h_k^{l+1}(m) - h_k^l(m))^2]. \quad (5.12)$$

Le problème (5.12) peut être décomposé en un ensemble de plus petits programmes quadratiques indépendants, un pour chaque paire origine-destination et pour chaque temps discret.

(5.12) est équivalent à résoudre le problème suivant, pour chaque $i \in I$, $m = 1, \dots, M$,

$$\min \sum_{k \in K_i} [S_k(h^l(m))(h_k^{l+1}(m) - h_k^l(m)) + \frac{1}{2\alpha}(h_k^{l+1}(m) - h_k^l(m))^2]. \quad (5.13)$$

À chaque itération, nous disposons d'une solution courante et nous devons résoudre le problème de chargement pour obtenir les valeurs de $S_k(h^l(m))$ afin de calculer $h_k^{l+1}(m)$. La direction de projection est obtenue en résolvant le problème (5.13) pour chaque $i \in I$, $m = 1, \dots, M$,

$$h_k^{l+1}(m) = \arg \min_{h_k(m)} \sum_{k \in K_i} [S_k(h^l(m))(h_k(m) - h_k^l(m)) + \frac{1}{2\alpha}(h_k(m) - h_k^l(m))^2] \quad (5.14)$$

$$\sum_{k \in K_i} h_k(m) = g_i(m), \quad \forall i \in I, \quad (5.15)$$

$$h_k(m) \geq 0, \quad \forall k \in K_i, \quad i \in I. \quad (5.16)$$

La condition d'arrêt de l'algorithme est déterminée par la valeur de la fonction de gap de Fukushima ou par la différence entre deux solutions successives.

Les détails de l'algorithme de projection que nous avons implanté sont :

Algorithme de projection

- [initialisation] 1. Soit $g_i(t)$ la demande pour tout $i \in I$;
soit $|K_i|$ le nombre de chemins qui relie la paire $(OD)_i$;
 $h_k^0(t) = \frac{g_i(t)}{|K_i|} \forall k \in K_i, \forall i \in I, \forall 0 \leq t \leq T$;
 $l := 0$.
- [itération] 2. Si une des conditions d'arrêt est satisfaite : arrêter.
3. Résoudre le problème de chargement dynamique :
trouver $v_a(t), s_a(v(t)), S_a(t)$.
4. Résoudre (5.14) pour trouver $h_k^{l+1}, \forall k \in K_i, \forall i \in I$.
5. $l := l + 1$.

5.2.1 Direction de projection

Nous allons expliquer brièvement la méthode que nous avons utilisée pour résoudre (5.14) et trouver le nouveau flot, h_k^{l+1} . Cette méthode, que nous appelons méthode d'interpolation unidimensionnelle, est basée sur une approche utilisée par Wu dans sa thèse doctorale (1991), qui est inspirée d'une contribution de Kennington (1980). Soit le problème suivant à résoudre :

$$\min_{h_k} \sum_{k \in K_i} [S_k(h^l)(h_k - h_k^l) + \frac{1}{2\alpha}(h_k - h_k^l)^2] \quad (5.17)$$

$$\text{s. à.} \quad \sum_{k \in K_i} h_k = g_i, \quad (5.18)$$

$$h_k \geq 0, \forall k \in K_i. \quad (5.19)$$

Puisque le programme est strictement convexe, les conditions de Kuhn-Tucker sont des conditions nécessaires et suffisantes pour l'optimalité de la solution.

Soient μ, β_k , les variables duales associées aux contraintes (5.18) et (5.19) respectivement. Nous avons les conditions suivantes de Kuhn-Tucker :

$$S_k(h^l) + \frac{1}{\alpha}(h_k - h_k^l) - \mu - \beta_k = 0, \forall k \in K_i, \quad (5.20)$$

$$h_k \beta_k = 0, \forall k \in K_i, \quad (5.21)$$

$$\beta_k \geq 0, \forall k \in K_i, \quad (5.22)$$

$$\sum_{k \in K_i} h_k = g_i, \quad h_k \geq 0, \forall k \in K_i.$$

Alors nous pouvons établir :

$$h_k = \max\{\alpha(\mu - S_k(h^l)) + h_k^l, 0\} \quad (5.23)$$

$$\beta_k = \max\{S_k(h^l) - \frac{1}{\alpha}h_k^l - \mu, 0\} \quad (5.24)$$

Par définition (5.23) et (5.24) satisfont toujours (5.19) et (5.22) pour un μ donné.

Proposition 3 (5.23) et (5.24) satisfont (5.20)

Proposition 4 (5.23) et (5.24) satisfont (5.21)

Preuve. Voir Wu (1991).

Par conséquent, la seule contrainte dont nous devons nous préoccuper est (5.18).

Maintenant nous définissons la fonction suivante :

$$\Gamma(\mu) = \sum_{k \in K} h_k(\mu) = \sum_{k \in K} \max\{\alpha(\mu - S_k(h^l)) + h_k^l, 0\}. \quad (5.25)$$

Nous devons trouver μ^* tel que $\Gamma(\mu^*) = g_i$. Il est clair que la fonction $\Gamma(\mu)$ est non décroissante, monotone et linéaire par morceaux. Puisque $\Gamma(\mu)$ est la somme de $|K|$ composantes, les points de rupture de la fonction linéaire par morceaux $\Gamma(\mu)$ sont au nombre de $|K|$, c'est-à-dire, μ_k^0 pour $k = 1, \dots, |K|$ où

$$\mu_1^0 \leq \mu_2^0 \leq \dots \leq \mu_{|K|}^0,$$

sans perte de généralité. Nous avons :

$$\Gamma(\mu) = 0, \quad \text{si } \mu \leq \mu_1^0;$$

$$\Gamma(\mu) = \sum_{k \in K} \max\{\alpha(\mu_{|k|} - S_k(h^l)) + h_k^l, 0\}, \quad \text{si } \mu = \mu_{|k|}^0.$$

Nous présentons maintenant un algorithme pour calculer μ^* tel que :

$$\Gamma(\mu^*) = g_i.$$

Cette procédure consiste en une recherche binaire pour border μ^* entre deux points de rupture suivie par une interpolation linéaire.

Algorithme de recherche de μ^*

- [étape 0] [initialisation] $l = 1, r = |K|, g_{\max} = \infty, g_{\min} = 0;$
 $\mu_1^0 \leq \mu_2^0 \leq \dots \leq \mu_{|K|}^0$ sont les
 points de rupture (triés);
- [étape 1] [test d'intervalle] si $r - l = 1$, aller à [étape 4];
 sinon $m = \lfloor \frac{l+r}{2} \rfloor$
 où $\lfloor x \rfloor$ est le plus grand entier $\leq x$.
- [étape 2] [calcul $\Gamma(\mu^0)$] $\Gamma(\mu_m^0) = \sum_{k \in K} \max\{\alpha(\mu_m^0 - S_k(h^l)) + h_k^l, 0\}.$
- [étape 3] [mise à jour] Si $\Gamma(\mu_m^0) = g$
 $\mu^* = \mu_m^0$ et ARRÊT.
- Si $\Gamma(\mu_m^0) > g$
 $r = m, g_{\max} = \Gamma(\mu_m^0), \mu_{\max} = \mu_m^0;$
 aller à [étape 1].
- Si $\Gamma(\mu_m^0) < g$
 $l = m, g_{\min} = \Gamma(\mu_m^0), \mu_{\min} = \mu_m^0;$
 et aller à [étape 1].
- [étape 4] [interpolation] $\mu^* = \mu_{\min} + \frac{\mu_{\max} - \mu_{\min}}{g_{\max} - g_{\min}}(g - g_{\min})$ et ARRÊT.

Lorsque μ^* est déterminé, nous pouvons utiliser l'équation (5.23) pour calculer h .

Pour résoudre (5.10) le choix de α est important, car ce paramètre a une influence considérable sur le taux de convergence. D'un part, si α est "petit", le taux de convergence est "faible"; si α est plus élevé, le taux peut augmenter parce que le "pas" à chaque itération est plus élevé aussi. Mais, si α est "trop grand", il se peut que l'on perde la convergence.

5.3 Résultats numériques

Nous avons utilisé le réseau de Sioux Falls pour expérimenter avec l'algorithme de projection la résolution du problème d'équilibre dynamique. Nous avons défini six zones (origines et destinations) qui correspondent à quatre noeuds du réseau. Il y a un total de 36 paires O-D et 12 ou 16 chemins pour chaque paire. Le paramètre α de la méthode de projection est égal à 2. Les critères de convergence sont la valeur absolue de la fonction de gap de Fukushima (notez que la valeur de la fonction de gap de Fukushima est inférieure ou égale à zéro) et la valeur de la *norme de la différence entre deux solutions successives*, $\|h^{l+1} - h^l\|$. (Nous dirons *norme de la différence* pour abrégé)

L'algorithme se termine quand une de ces deux valeurs est plus petite ou égale à 1. La période de départ est $[0, 120]$, $T = 120$, comme dans l'exemple du problème de chargement. La fonction du temps de parcours et les paramètres β sont aussi les mêmes. Les taux de demandes sont donnés par la fonction $g_i(t) = |K_i| \times 0.0005 \times (T - t) \times t$, où i est un indice de la paire O-D, $i = 1, \dots, 36$, $t = 0, \dots, 120$ et $|K_i|$ est le nombre de chemins pour la paire i .

La période de départ est discrétisée en intervalles d'une minute. Dans les tableaux, nous allons montrer les résultats en utilisant des points discrets avec intervalles de 10 minutes.

La solution initiale est $h_k^0(t) = \frac{g_i(t)}{|K_i|}$, $\forall k \in K_i$, $i \in I$, c'est-à-dire le même taux de départ initial pour tous les chemins pour chaque paire origine-destination. Le tableau (5.1) et la figure (5.1) montrent les valeurs initiales des flots de départ sur les chemins par rapport au temps de départ.

temps	flot de départ
0.50	0.03
10.50	0.57
20.50	1.02
30.50	1.36
40.50	1.61
50.50	1.75
60.50	1.80
70.50	1.74
80.50	1.59
90.50	1.33
100.50	0.98
110.50	0.52

TAB. 5.1 – Sioux Falls : flot de départ initial pour tous les chemins.

Nous avons calculé les temps de parcours des chemins, $S_k(h^l)$, après la résolution du problème de chargement, en utilisant l'Algorithme-2 du chapitre précédent. Ainsi, l'algorithme commence avec le chargement du flot h_k^0 sur tous les arcs qui font partie des chemins $k \in K_i$, $i \in I$. La méthode de projection se termine après 34 itérations, et les résultats finaux s'approchent des conditions d'équilibre. Nous allons analyser les résultats obtenus pour la paire O-D 1 et 6.

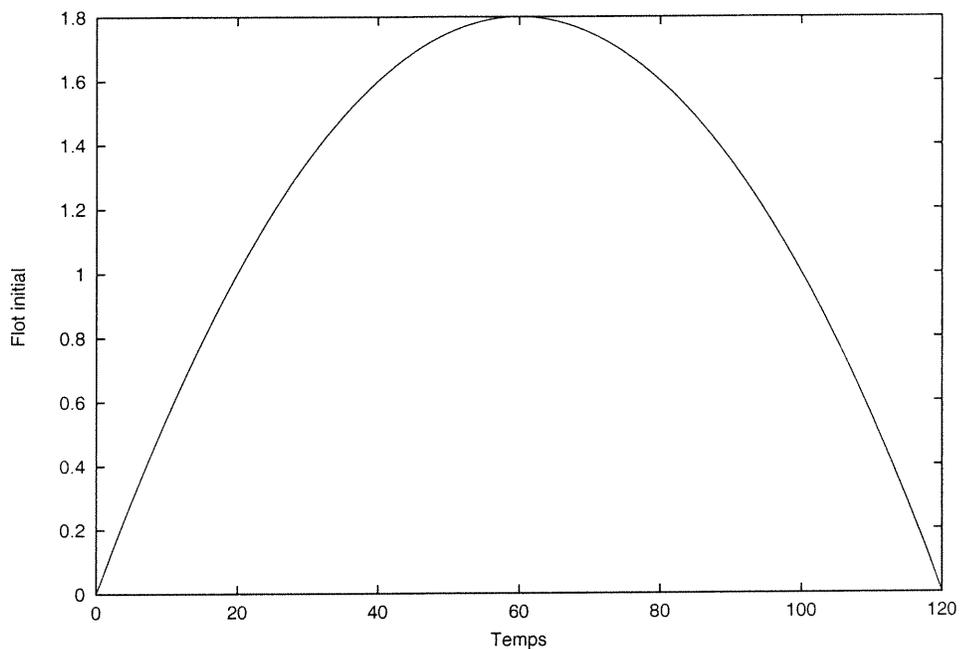


FIG. 5.1 – Sioux Falls : flot initial pour tous les chemins.

Le tableau (5.2) et la figure (5.2) contiennent les temps de parcours initiaux, au début de l'algorithme, par rapport au temps de départ, pour les chemins qui relient la première paire O-D (chemins 1 à 12). Le tableau (5.3) et la figure (5.3) montrent les temps de parcours initiaux pour les chemins qui relient la paire 6 (chemins 77 à 92).

Pair O-D 1 : temps de parcours des chemins												
temps	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
0.50	3.60	2.40	4.84	3.60	6.06	6.70	2.40	6.02	2.40	4.81	6.63	2.40
10.50	3.79	2.62	5.49	3.74	6.46	7.83	2.49	6.34	2.74	5.18	7.52	2.63
20.50	4.04	2.89	6.31	3.98	6.97	9.31	2.71	6.83	3.26	5.86	8.87	3.07
30.50	4.28	3.14	7.17	4.20	7.44	10.91	2.93	7.27	3.84	6.55	10.37	3.55
40.50	4.48	3.37	8.01	4.38	7.83	12.48	3.12	7.66	4.44	7.22	11.92	4.03
50.50	4.64	3.53	8.72	4.52	8.11	13.86	3.29	7.96	5.01	7.83	13.41	4.47
60.50	4.73	3.63	9.21	4.61	8.25	14.81	3.42	8.17	5.47	8.32	14.69	4.85
70.50	4.76	3.62	9.37	4.65	8.23	15.16	3.51	8.25	5.73	8.65	15.56	5.11
80.50	4.70	3.51	9.11	4.63	8.06	14.77	3.55	8.21	5.69	8.75	15.80	5.21
90.50	4.57	3.31	8.39	4.56	7.76	13.65	3.51	8.02	5.29	8.56	15.22	5.09
100.50	4.37	3.05	7.28	4.44	7.36	11.93	3.39	7.68	4.52	7.99	13.59	4.71
110.50	4.10	2.75	6.01	4.28	6.90	9.87	3.13	7.13	3.53	7.00	10.88	4.03

TAB. 5.2 – Sioux Falls : temps de parcours initial ; paire O-D 1 ; chemins 1 – 12.

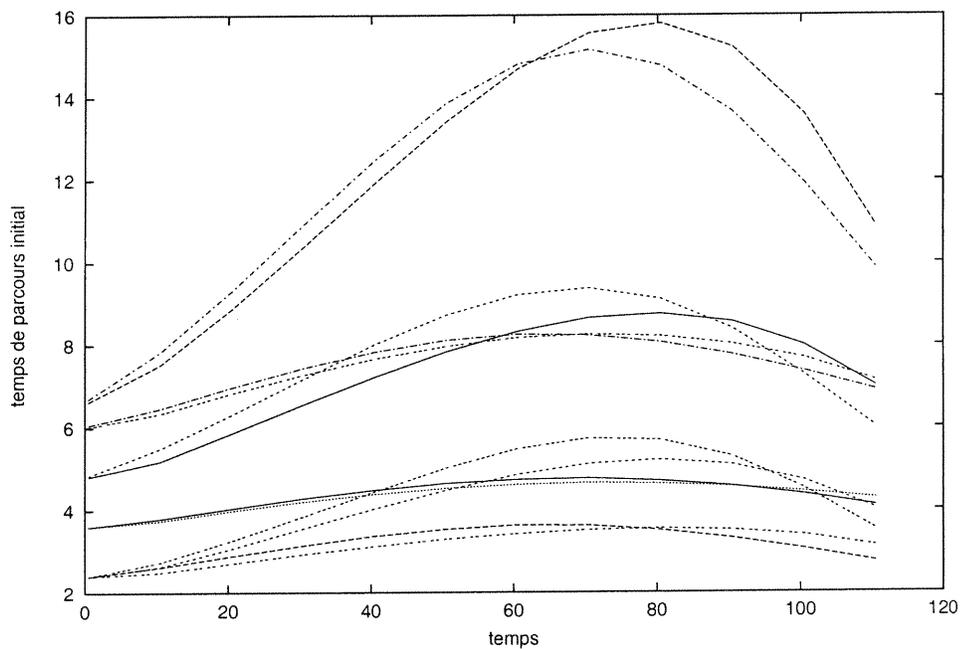


FIG. 5.2 – Sioux Falls : temps de parcours initial ; paire O-D 1 ; chemins 1 – 12.

Pair O-D 6 : temps de parcours																
temps	S_{77}	S_{78}	S_{79}	S_{80}	S_{81}	S_{82}	S_{83}	S_{84}	S_{85}	S_{86}	S_{87}	S_{88}	S_{89}	S_{90}	S_{91}	S_{92}
0.50	11.21	12.49	10.44	9.20	13.80	13.04	14.43	13.15	8.61	9.87	7.91	6.68	11.21	11.02	8.50	9.20
10.50	12.90	14.29	11.79	10.47	16.18	15.07	16.13	14.77	9.85	11.21	8.84	7.53	12.91	12.00	9.28	10.47
20.50	14.94	16.42	13.41	12.02	19.12	17.62	18.05	16.63	11.54	12.99	10.13	8.76	15.16	13.19	10.30	12.22
30.50	17.01	18.56	15.04	13.59	22.29	20.42	19.90	18.43	13.31	14.84	11.48	10.04	17.48	14.32	11.28	14.05
40.50	18.90	20.49	16.55	15.06	25.39	23.22	21.51	20.00	15.01	16.59	12.78	11.30	19.65	15.31	12.16	15.79
50.50	20.35	21.96	17.75	16.24	27.99	25.69	22.67	21.15	16.42	18.04	13.89	12.39	21.38	16.06	12.87	17.26
60.50	21.10	22.70	18.47	16.96	29.66	27.41	23.20	21.70	17.32	18.93	14.67	13.16	22.35	16.50	13.32	18.23
70.50	20.97	22.53	18.54	17.05	30.05	28.02	22.97	21.50	17.49	19.07	14.96	13.46	22.37	16.55	13.45	18.51
80.50	19.90	21.40	17.87	16.42	29.03	27.35	21.94	20.51	16.82	18.35	14.63	13.17	21.35	16.18	13.20	17.96
90.50	18.00	19.42	16.47	15.08	26.68	25.40	20.18	18.82	15.33	16.78	13.63	12.22	19.41	15.37	12.55	16.59
100.50	15.56	16.88	14.50	13.18	23.18	22.26	17.98	16.70	13.23	14.58	12.02	10.68	16.83	14.16	11.51	14.54
110.50	13.04	14.26	12.33	11.10	18.55	17.86	15.76	14.54	10.89	12.14	10.10	8.85	14.05	12.64	10.18	12.15

TAB. 5.3 – Sioux Falls : temps de parcours initial ; paire O-D 6 ; chemins 77 – 92.

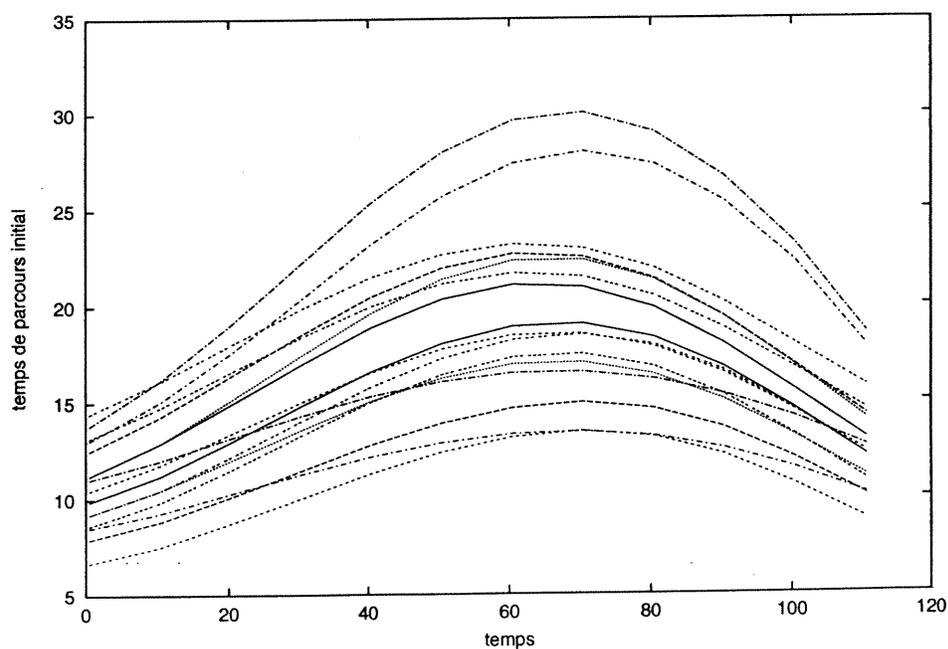


FIG. 5.3 – Sioux Falls : temps de parcours initial ; paire O-D 6 ; chemins 77 – 92

Le temps CPU total, après ces 34 itérations, est de 218 secondes sur une station de travail SUN Microsystems ULTRA SPARC 10, 300 Mhz, 128 M ram, où chaque itération requiert environ 6 secondes.

Paire O-D 1 : flot de départ sur les chemins												
temps	h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8	h_9	h_{10}	h_{11}	h_{12}
0.50	0.00	0.09	0.00	0.00	0.00	0.00	0.09	0.00	0.09	0.00	0.00	0.09
10.50	0.00	1.73	0.00	0.00	0.00	0.00	1.79	0.00	1.66	0.00	0.00	1.71
20.50	0.00	3.10	0.00	0.00	0.00	0.00	3.19	0.00	2.94	0.00	0.00	3.01
30.50	0.00	4.18	0.00	0.00	0.00	0.00	4.27	0.00	3.91	0.00	0.00	4.02
40.50	0.00	4.97	0.00	0.00	0.00	0.00	5.06	0.00	4.57	0.00	0.00	4.72
50.50	0.00	5.47	0.00	0.00	0.00	0.00	5.52	0.00	4.95	0.00	0.00	5.12
60.50	0.00	5.64	0.00	0.00	0.00	0.00	5.70	0.00	5.03	0.00	0.00	5.23
70.50	0.00	5.54	0.00	0.00	0.00	0.00	5.54	0.00	4.85	0.00	0.00	5.02
80.50	0.00	5.09	0.00	0.00	0.00	0.00	5.06	0.00	4.39	0.00	0.00	4.54
90.50	0.00	4.33	0.00	0.00	0.00	0.00	4.25	0.00	3.71	0.00	0.00	3.73
100.50	0.00	3.24	0.00	0.00	0.00	0.00	3.10	0.00	2.76	0.00	0.00	2.66
110.50	0.00	1.80	0.00	0.00	0.00	0.00	1.64	0.00	1.54	0.00	0.00	1.32

TAB. 5.4 – Sioux Falls : taux de départ final; paire O-D 1; chemins : 1 – 12.

Pair O-D 1 : temps de parcours												
temps	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}	S_{12}
0.50	3.60	2.40	4.81	3.60	6.01	6.66	2.40	6.00	2.40	4.81	6.60	2.40
10.50	3.60	2.44	4.88	3.60	6.05	7.03	2.44	6.04	2.44	4.89	6.92	2.44
20.50	3.60	2.47	4.95	3.60	6.08	7.41	2.47	6.07	2.47	4.95	7.27	2.47
30.50	3.60	2.50	5.00	3.60	6.11	7.75	2.50	6.10	2.49	5.01	7.57	2.50
40.50	3.60	2.52	5.04	3.60	6.13	8.01	2.53	6.13	2.51	5.05	7.84	2.52
50.50	3.60	2.54	5.06	3.60	6.14	8.16	2.54	6.14	2.52	5.07	8.09	2.53
60.50	3.60	2.55	5.07	3.60	6.14	8.24	2.54	6.14	2.53	5.08	8.23	2.53
70.50	3.60	2.54	5.06	3.60	6.14	8.22	2.54	6.14	2.52	5.07	8.23	2.53
80.50	3.60	2.53	5.05	3.60	6.13	8.08	2.53	6.13	2.51	5.04	8.12	2.52
90.50	3.60	2.51	5.00	3.60	6.10	7.83	2.51	6.11	2.49	5.00	7.90	2.50
100.50	3.60	2.48	4.95	3.60	6.07	7.45	2.48	6.08	2.47	4.94	7.59	2.47
110.50	3.60	2.45	4.88	3.60	6.03	6.99	2.45	6.05	2.44	4.87	7.21	2.44

TAB. 5.5 – Sioux Falls : temps de parcours final; paire O-D 1; chemins 1 – 12.

Les tableaux (5.4), (5.5) et les figures (5.4), (5.5) montrent le flot de départ et le

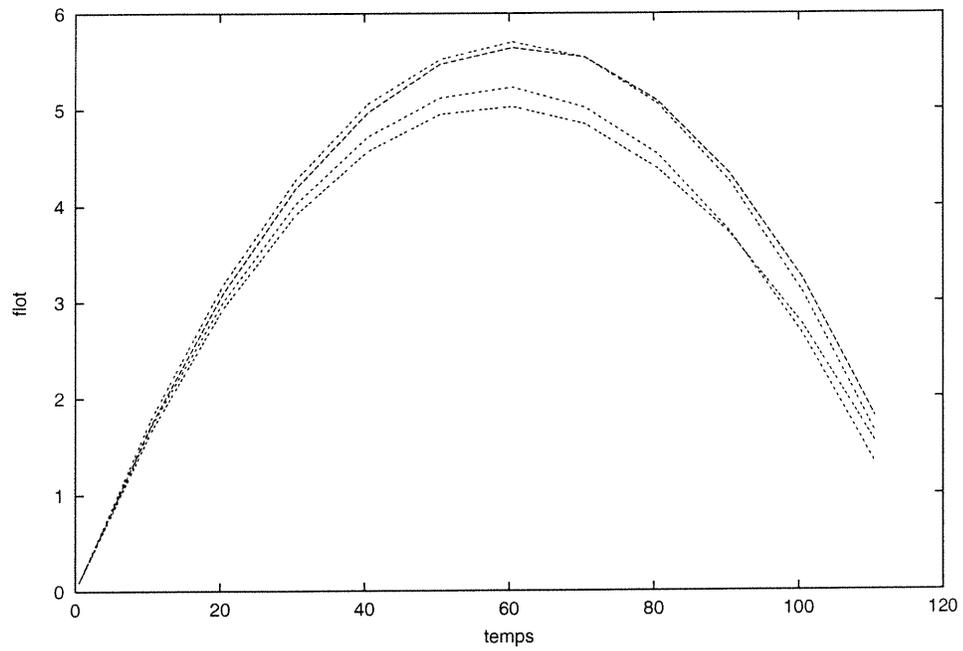


FIG. 5.4 – Sioux Falls : taux de départ final ; paire O-D 1 ; chemins : 2,7,9,12).

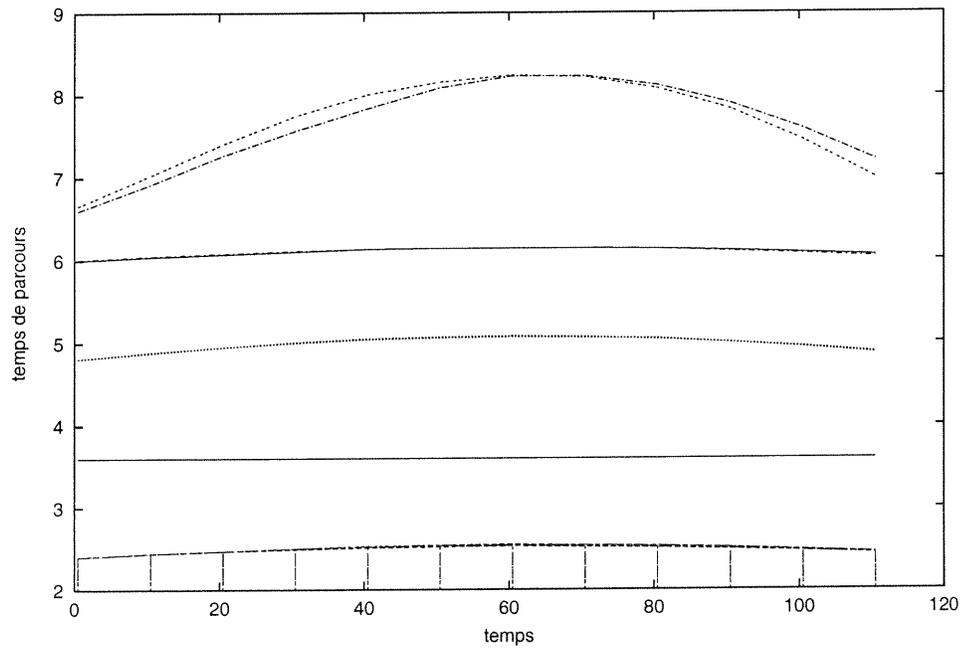


FIG. 5.5 – Sioux Falls : temps de parcours final ; paire O-D 1 ; chemins 1 – 12

temps de parcours pour les chemins qui connectent la paire O-D 1 par rapport au temps de départ à la fin de l'algorithme. Il faut noter que les conditions d'équilibre sont vérifiées. Les taux de départ sont positifs sur les chemins 2, 7, 9, 12 pour chaque intervalle de départ. Ceci implique que le coût pour traverser ces chemins, à chaque intervalle de départ, est minimal. Les autres chemins ont des coûts plus grands et leurs flots de départ sont égaux à 0.

Les tableaux (5.6), (5.7) et figures (5.6) (5.7) montrent que les conditions d'équilibre sont aussi vérifiées pour la paire O-D 6. Le flot positif implique que le temps de parcours est minimal. Alors seulement les chemins 88 et 91 ont des flots positifs. Les temps de parcours du chemin 91 pour le flot de départ au temps 0.5 et 10.5 sont plus grands que les temps minimaux, alors les flots sont nuls.

Paire O-D 6 : flot de départ sur les chemins																
temps	h_{77}	h_{78}	h_{79}	h_{80}	h_{81}	h_{82}	h_{83}	h_{84}	h_{85}	h_{86}	h_{87}	h_{88}	h_{89}	h_{90}	h_{91}	h_{92}
0.50	0	0	0	0	0	0	0	0	0	0	0	0.48	0	0	0	0
10.50	0	0	0	0	0	0	0	0	0	0	0	9.20	0	0	0	0
20.50	0	0	0	0	0	0	0	0	0	0	0	12.32	0	0	3.99	0
30.50	0	0	0	0	0	0	0	0	0	0	0	14.64	0	0	7.20	0
40.50	0	0	0	0	0	0	0	0	0	0	0	16.26	0	0	9.50	0
50.50	0	0	0	0	0	0	0	0	0	0	0	17.13	0	0	10.95	0
60.50	0	0	0	0	0	0	0	0	0	0	0	17.31	0	0	11.49	0
70.50	0	0	0	0	0	0	0	0	0	0	0	16.79	0	0	11.13	0
80.50	0	0	0	0	0	0	0	0	0	0	0	15.56	0	0	9.88	0
90.50	0	0	0	0	0	0	0	0	0	0	0	13.57	0	0	7.79	0
100.50	0	0	0	0	0	0	0	0	0	0	0	10.38	0	0	5.30	0
110.50	0	0	0	0	0	0	0	0	0	0	0	8.40	0	0	0	0

TAB. 5.6 – Sioux Falls : taux de départ final ; paire O-D 6 ; chemins : 77 – 92.

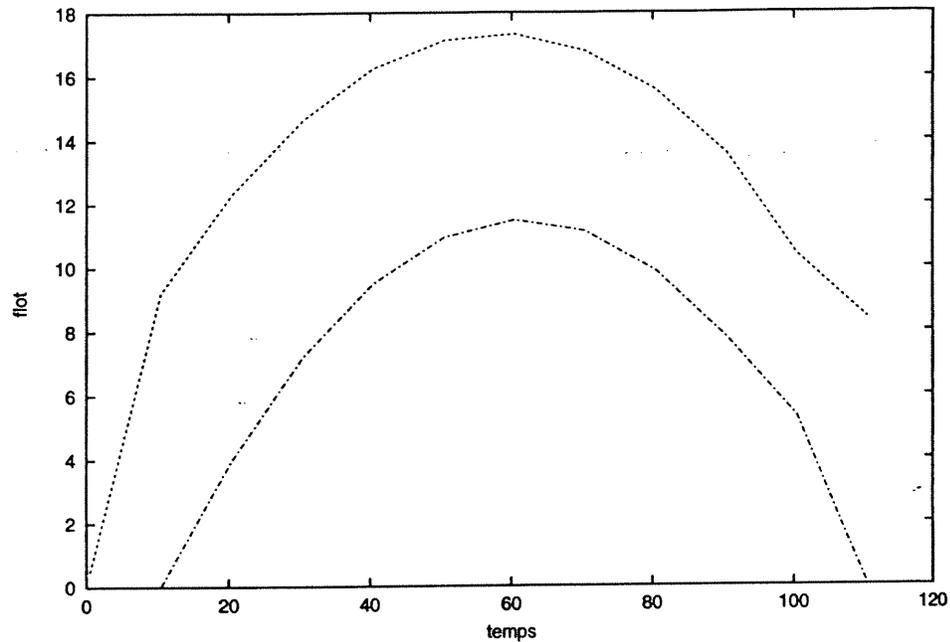


FIG. 5.6 – Sioux Falls : taux de départ final ; paire O-D 6 ; chemins : 88, 91.

Pair O-D 6 : temps de parcours sur les chemins																
temps	S_{77}	S_{78}	S_{79}	S_{80}	S_{81}	S_{82}	S_{83}	S_{84}	S_{85}	S_{86}	S_{87}	S_{88}	S_{89}	S_{90}	S_{91}	S_{92}
0.50	11.27	12.49	10.69	9.47	13.48	13.27	14.98	13.75	8.53	9.74	7.95	6.73	11.27	10.86	8.46	9.47
10.50	12.96	14.20	12.40	11.16	14.93	14.93	16.58	15.33	10.14	11.38	9.57	8.34	12.96	11.25	8.85	11.16
20.50	14.34	15.59	13.79	12.54	16.65	16.64	17.83	16.57	11.57	12.83	11.02	9.77	14.33	12.16	9.76	12.53
30.50	15.44	16.71	14.91	13.64	18.41	18.41	18.84	17.57	12.70	13.97	12.16	10.90	15.44	13.30	10.90	13.64
40.50	16.29	17.56	15.76	14.49	19.83	19.83	19.60	18.33	13.58	14.85	13.05	11.78	16.28	14.18	11.78	14.48
50.50	16.87	18.14	16.33	15.07	20.70	20.70	20.06	18.80	14.22	15.49	13.68	12.42	16.86	14.82	12.42	15.06
60.50	17.11	18.37	16.57	15.31	21.02	21.02	20.16	18.90	14.54	15.81	14.00	12.74	17.10	15.14	12.74	15.30
70.50	16.94	18.20	16.40	15.14	20.73	20.73	19.84	18.58	14.48	15.74	13.94	12.68	16.93	15.08	12.68	15.13
80.50	16.37	17.61	15.81	14.57	19.79	19.79	19.10	17.86	14.02	15.27	13.47	12.22	16.36	14.62	12.22	14.56
90.50	15.42	16.65	14.85	13.62	18.26	18.26	18.04	16.81	13.17	14.41	12.61	11.37	15.41	13.76	11.36	13.61
100.50	14.12	15.34	13.54	12.32	16.11	16.11	16.59	15.38	11.99	13.21	11.41	10.19	14.11	12.58	10.18	12.31
110.50	12.50	13.70	11.90	10.70	14.26	13.85	14.77	13.57	10.53	11.73	9.93	8.73	12.49	11.20	8.80	10.69

TAB. 5.7 – Sioux Falls : temps de parcours final ; paire O-D 6 ; chemins 77 – 92.

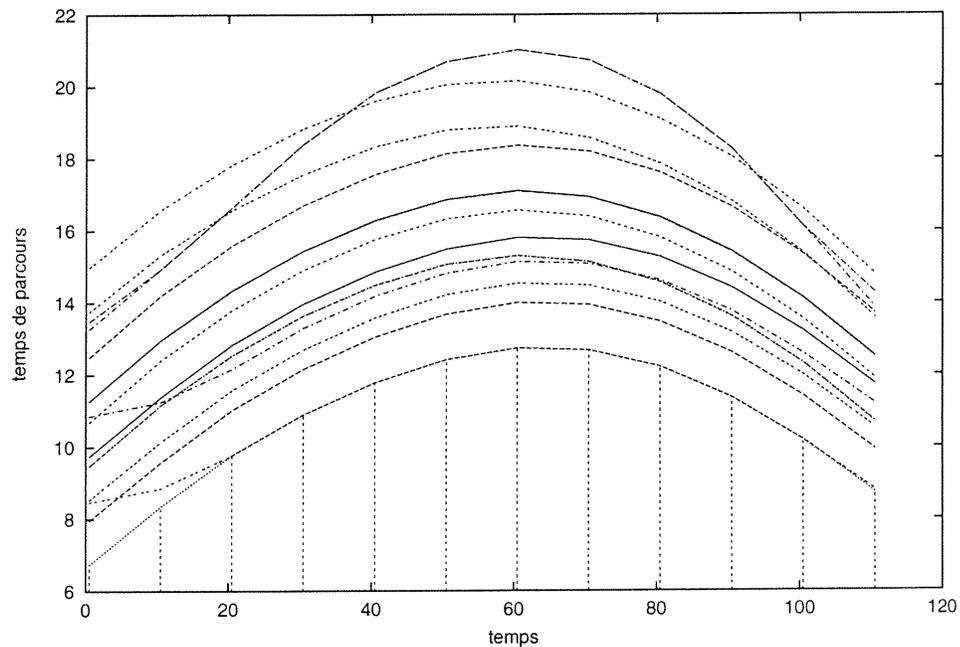


FIG. 5.7 – Sioux Falls : temps de parcours final ; paire O-D 6 ; chemins 77 – 92.

Le tableau (5.8) montre la valeur absolue de la fonction du gap de Fukushima et du *gap de Fukushima relatif* ainsi que la norme de la différence pour chaque itération. Nous avons défini le gap de Fukushima relatif comme suit :

$$\text{gap de Fukushima relatif} = \frac{(S(h^l), h^{l+1} - h^l) + \frac{1}{2\alpha}(h^{l+1} - h^l, h^{l+1} - h^l)}{(s(h^l), h^l)}. \quad (5.26)$$

La figure (5.8) montre la convergence de la valeur absolue du gap de Fukushima sur un échelle \log_{10} par rapport aux itérations. Le gap absolu à une valeur initiale de 334282 à la première itération et converge vers 0.7681 à l'itération finale. Le gap de Fukushima relatif final est égal à 0.18×10^{-7} . La valeur de la norme de la différence est 625 à la première itération et elle diminue à 1.8848 à l'itération 34.

Nous avons réalisé aussi des exemples numériques sur le réseau de la ville de Hull. Ce

itér	gap	gap relatif	$\ h^{l+1} - h^l\ $
1	334282.8438	0.3479604	625.9488
2	19122.1797	0.0403191	226.9684
3	6707.3315	0.0152745	146.8658
4	3296.4568	0.0076942	108.4380
5	2169.8633	0.0051388	88.0443
6	1215.3326	0.0029121	62.6681
7	481.3202	0.0011603	37.8925
8	215.4003	0.0005198	28.1984
9	142.3939	0.0003434	23.0797
10	100.9026	0.0002432	19.6409
11	74.8675	0.0001803	17.0962
12	58.3813	0.0001405	15.0988
13	47.4295	0.0001140	13.4687
14	37.3979	0.0000898	12.0482
15	30.5406	0.0000733	10.8091
16	23.3648	0.0000561	9.6910
17	20.1257	0.0000483	8.6889
18	15.8777	0.0000381	7.7935
19	11.7413	0.0000281	6.9577
20	9.5161	0.0000228	6.1389
21	8.8719	0.0000212	5.3979
22	5.0335	0.0000120	4.8853
23	5.3934	0.0000129	4.3791
24	4.0245	0.0000096	3.9086
25	2.6470	0.0000063	3.5584
26	1.9475	0.0000047	3.2917
27	1.6514	0.0000039	3.0240
28	1.5716	0.0000038	2.7998
29	2.1069	0.0000050	2.6119
30	2.3810	0.0000057	2.4446
31	1.8086	0.0000043	2.2673
32	1.6191	0.0000039	2.1269
33	1.2590	0.0000030	1.9910
34	0.7681	0.0000018	1.8848

TAB. 5.8 – Sioux Falls : ($T=120$).

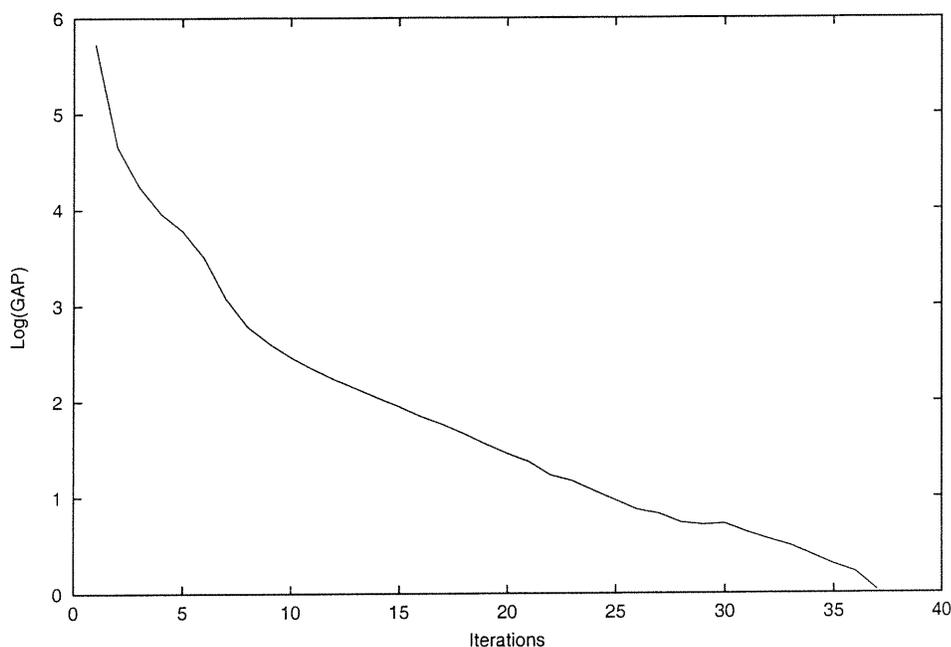


FIG. 5.8 – Sioux Falls : gap de Fukushima ($T=120$).

réseau comporte 92 paires O-D, et chaque paire est reliée par 5 chemins. Le tableau (5.9) montre la convergence du gap de Fukushima, du gap de Fukushima relatif et de la norme de la différence pour un exemple avec une période de départ de 30 minutes, $[0, 30]$, $T = 30$. Les taux des demandes sont donnés par la fonction $g_i(t) = |K_i| \times 0.010 \times (T - t) \times t$. L'algorithme termine après 17 itérations avec une valeur du gap égale à 0.7771 et du gap de Fukushima relatif égal à 0.15×10^{-7} ; remarquons que la valeur initiale du gap était égale à 132822.3750. La norme de la différence débute avec une valeur égale à 282.70 et termine avec une valeur égale à 2.2117. Le temps CPU total est de 242 secondes.

Le tableau (5.10) contient les valeurs obtenues pour un autre exemple sur le réseau de la ville de Hull avec une période de départ égale à 60 minutes. La fonction du taux de demandes dans cet exemple est la suivante : $g_i(t) = |K_i| \times 0.003 \times (T - t) \times t$. Dans ce cas,

itér	gap	gap relatif	$\ h^{l+1} - h^l\ $
1	132822.3750	0.2129651	282.7057
2	15464.7715	0.0287744	185.4984
3	9205.0527	0.0177502	122.8156
4	2005.8932	0.0039032	64.4413
5	233.0121	0.0004551	24.7806
6	44.7081	0.0000874	11.7757
7	16.2643	0.0000318	7.6202
8	9.3844	0.0000184	5.9340
9	6.6944	0.0000131	4.9972
10	4.6815	0.0000092	4.3040
11	4.2015	0.0000082	3.8169
12	3.5390	0.0000069	3.5311
13	2.8857	0.0000056	3.2530
14	2.2762	0.0000045	2.9825
15	1.7359	0.0000034	2.6564
16	1.3383	0.0000026	2.4024
17	0.7771	0.0000015	2.2117

TAB. 5.9 – Réseau de Hull : ($T=30$).

nous avons utilisé une valeur variable pour le paramètre α . La valeur initiale est égale à 2, et, après chaque itération, α est réduite en multipliant par 0.9 avec une borne inférieure de 1, $\alpha^{l+1} := \max\{1, \alpha^l \times 0.9\}$. La valeur initiale du gap de Fukushima est de 396451 et la valeur finale, à l'itération 27, est 0.9869. La valeur finale du gap de Fukushima relatif est 0.6×10^{-8} . La norme de la différence à la première itération est égale à 482.92 et elle diminue à 2.5052. Le temps CPU total d'exécution est de 829 secondes.

Comme nous pouvons le constater, le gap de Fukushima et la différence entre deux solutions convergent vers zéro. Les conditions d'équilibre sont aussi vérifiées par la solution finale. Pour ce dernier exemple, nous présentons, aux tableaux (5.11) et (5.12), le flot de départ et le coût des chemins minimaux pour les 10 premières paires origine-destination. Le tableau (5.13) contient les arcs qui forment ces chemins.

itér	gap	Fukushima	gap relatif	$\ h^{l+1} - h^l\ $
1		396451.4062	0.2104868	482.9207
2		150914.0781	0.0862236	465.2198
3		194297.2969	0.1139694	416.6104
4		92505.7969	0.0556498	346.6613
5		93512.8906	0.0570510	286.4566
6		26569.2969	0.0166609	194.3845
7		15118.1816	0.0095618	123.8362
8		1261.3599	0.0008047	44.9281
9		326.3672	0.0002084	23.0817
10		148.1146	0.0000946	16.2179
11		91.7404	0.0000586	12.8654
12		56.2043	0.0000359	10.3243
13		39.6237	0.0000253	8.5734
14		28.2750	0.0000181	7.2076
15		18.7286	0.0000120	6.2083
16		14.1859	0.0000091	5.4889
17		17.2564	0.0000110	5.0237
18		11.1740	0.0000071	4.5991
19		8.7930	0.0000056	4.2029
20		7.8462	0.0000050	3.9125
21		3.9008	0.0000025	3.5984
22		6.2453	0.0000040	3.3925
23		6.2162	0.0000040	3.1730
24		7.3097	0.0000047	2.9378
25		2.6213	0.0000017	2.7852
26		4.0800	0.0000026	2.6154
27		0.9869	0.0000006	2.5052

TAB. 5.10 – Réseau de Hull : ($T=60$).

Paire O-D	1	2	3	4	5	6		7	8		9	10
	h[1]	h[6]	h[11]	h[16]	h[24]	h[26]	h[29]	h[31]	h[36]	h[37]	h[41]	h[48]
0.50	0.45	0.45	0.45	0.45	0.45	0.45	0.00	0.45	0.45	0.00	0.45	0.45
10.50	7.80	7.80	7.80	7.80	7.80	7.80	0.00	7.80	7.80	0.00	7.80	7.80
20.50	12.15	12.15	12.15	12.15	12.15	10.06	2.09	12.15	12.15	0.00	12.15	12.15
30.50	13.50	13.50	13.50	13.50	13.50	13.50	0.00	13.50	8.63	4.87	13.50	13.50
40.50	11.85	11.85	11.85	11.85	11.85	11.85	0.00	11.85	11.85	0.00	11.85	11.85
50.50	7.20	7.20	7.20	7.20	7.20	7.20	0.00	7.20	7.20	0.00	7.20	7.20

TAB. 5.11 – Réseau de Hull : taux de départ final, paire O-D 17 ($T=60$).

Paire O-D	1	2	3	4	5	6		7	8		9	10
	S[1]	S[6]	S[11]	S[16]	S[24]	S[26]	S[29]	S[31]	S[36]	S[37]	S[41]	S[48]
0.50	17.44	12.06	6.05	10.87	9.65	14.48	15.74	9.65	16.73	20.93	8.46	21.05
10.50	22.61	15.81	9.03	14.41	12.17	18.24	19.05	12.68	22.30	24.41	11.70	26.58
20.50	30.44	22.19	14.62	20.20	16.30	24.14	24.14	17.50	29.32	29.51	17.27	33.64
30.50	40.60	30.12	20.82	25.92	21.40	29.90	30.20	23.57	36.07	36.07	23.29	41.20
40.50	45.96	36.53	25.65	30.08	25.56	33.92	34.68	28.04	37.75	39.87	28.15	43.24
50.50	42.58	36.02	26.70	30.84	26.15	34.74	35.38	28.77	36.95	39.69	28.93	37.81

TAB. 5.12 – Réseau de Hull : taux de départ final, paire O-D 17 ($T=60$).

chemin 1	1 400	262 315	738 320	759 323	566	565	563	561	403	399
chemin 6	1	262	738	759	566	565	563	561	403	398
chemin 11	1	261	116	117	119					
chemin 16	1	261	116	118	183	184	187	188	717	
chemin 24	1	262	738	758	560	511	244	245		
chemin 26	1 192	261 739	116	118	183	184	187	188	718	189
chemin 29	1 289	263 681	798 745	654	674	792	765	756	517	286
chemin 31	1	263	798	654	674	792	765	754		
chemin 36	1 79	261 488	116 489	117	120	121	73	74	76	313
chemin 37	1 265	263 267	798 269	654 586	674 689	792 380	765 763	756	516	264
chemin 41	1	261	116	117	120	122	115			
chemin 48	1 668	263 656	798 627	654 786	674 603	794 604	790	347	388	386

TAB. 5.13 – Réseau de Hull, arcs qui forment les chemins minimaux.

5.4 Méthode de restriction

Une des approches utilisées en optimisation pour résoudre des problèmes de grande taille est celle de la restriction (Lawphongpanich et Hearn, 1984). Cette approche consiste à résoudre le problème sur une région restreinte qui est élargie au fur et à mesure jusqu'à ce que la solution optimale soit trouvée.

À chaque itération, le problème d'équilibre est résolu sur un certain nombre de chemins déterminés d'avance et représentant le domaine restreint. Une fois la solution restreinte obtenue, un nouveau chemin est généré pour chaque paire origine-destination et pour chaque temps. Ces nouveaux chemins vont s'ajouter à l'ensemble des chemins dont nous disposons et ainsi agrandir le domaine réalisable. Nous allons chercher à nouveau une solution d'équilibre sur ce nouveau domaine. L'algorithme continue ainsi jusqu'à ce qu'une certaine condition

d'arrêt soit vérifiée.

Pour agrandir le domaine, c'est-à-dire, générer des nouveaux chemins, nous allons calculer, pour chaque paire origine-destination (et pour chaque intervalle) un chemin qui aura le plus petit temps de parcours selon les coûts que nous avons trouvés après le dernier chargement réalisé. C'est-à-dire, le problème suivant est résolu :

$$\min_{k \in K_i} (S(h^l(t), h_k(t)), \text{ pour presque tout } t \in [0, T], \forall i \in I. \quad (5.27)$$

Résoudre (5.27) est équivalent à chercher, pour chaque temps t de départ, les plus courts chemins dynamiques qui relient chaque paire origine-destination quand le réseau est chargé avec les flots $h^l(t)$.

La condition d'équilibre sur chaque domaine restreint $\Omega_l(t)$ est déterminée par la résolution du *problème d'inégalité variationnelle restreint*, RPIV, qui consiste à trouver $h^l(t)$ tel que :

$$(S(h^l(t)), h(t) - h^l(t)) \geq 0, \forall h(t) \in \Omega_l(t), \text{ pour presque tout } t \in [0, T]. \quad (5.28)$$

Il faut se rappeler que la condition d'équilibre sur $\Omega(t)$ est donnée par (5.3).

Puisque $h^*(t) \in \Omega(t)$ par (5.3) nous pouvons conclure que :

$$\min_h \{S(h^*(t)), (h(t) - h^*(t))\} = 0, \text{ pour presque tout } t \in [0, T]. \quad (5.29)$$

À l'itération l , soit $h_{pcc}^l(t)$ la solution du problème du plus court chemin dynamique, (5.27),

$$\arg \min (S(h^l(t)), h(t)) = h_{pcc}^l(t), \forall h(t) \in \Omega(t), \text{ pour presque tout } t \in [0, T], \quad (5.30)$$

et comme conséquence, à partir de (5.30), nous pouvons affirmer que :

$$(S(h^l(t)), h_{pcc}^l(t)) - (S(h^l(t)), h(t)) \leq 0, \forall h(t) \in \Omega(t), \text{ pour presque tout } t \in [0, T]; (5.31)$$

donc, si

$$(S(h^l(t)), h_{pcc}(t) - h^l(t)) = 0, \quad t \in [0, T],$$

par (5.30) par (5.31), $h^l(t)$ vérifie la condition d'optimalité sur $\Omega(t)$, définie par (5.3), puisque il n'existe pas $h(t) \in \Omega(t)$ tel que $(S(h^l(t)), h(t) - h^l(t)) < 0$.

$$\text{Notez que } (S(h^l), h_{pcc} - h^l) = \min\{(S(h^l), h - h^l)\} = -\max\{(S(h^l), h^l - h)\}, \quad \forall h \in \Omega.$$

Cette dernière fonction, qui est une fonction de *gap*, a été utilisée pour résoudre le problème d'équilibre statique (Hearn (1982), Nguyen et Dupuis (1984), Marcotte (1985), Marcotte et Dussault (1989), etc). Nous utilisons le $gap(h^l)$ pour mesurer la différence entre la solution actuelle et le coût minimal (plus court chemin avec fonction de coût déterminée par le flot de la solution actuelle). La valeur absolue du gap sera une des conditions d'arrêt de cet algorithme. Quand la valeur absolue du gap sera "assez" petit, les calculs vont s'arrêter. Il faut noter que cette fonction de gap (la valeur absolue) n'est pas nécessairement strictement décroissante.

Pour chaque paire origine-destination, nous allons générer une quantité indéterminée de nouveaux chemins, en théorie et au maximum, un pour chaque intervalle de temps. Mais il est évident, qu'en pratique, le nombre de chemins générés sera moindre. C'est souvent le cas que le chemin le plus court pour se déplacer d'un origine à une destination à un moment donné reste le plus court pendant un certain temps, même quelques fois pour tout l'intervalle à modéliser. Souvent le chemin le plus court est déjà utilisé, c'est-à-dire qu'il est un des chemins qui ont été générés dans une itération précédente. Nous allons voir, dans la section suivante, que ce phénomène a été observé dans nos exemples numériques. Ainsi, le nombre de nouveaux chemins générés sera normalement plus grand aux premières itérations

et il va décroître pour les itérations suivantes. Nous pouvons utiliser aussi la quantité de nouveaux chemins générés comme critère d'arrêt. Une troisième condition d'arrêt sera un nombre maximal d'itérations.

Nous allons chercher une solution approximative du RVIP. Nous utiliserons le nombre d'itérations de la méthode de projection ainsi que la valeur de la norme de la différence entre deux solutions de la méthode de projection comme critère d'arrêt pour le RPIV. À l'itération, $l + 1$, nous allons recommencer à résoudre le RPIV avec une solution initiale qui est égale à la solution trouvée à l'itération précédente, l , et avec un flot égal à zéro pour les nouveaux chemins générés. La valeur du $\text{gap}(h^l)$ est

$$\text{gap}(h^l) = (S(h^l), h_{pcc}^l - h^l) \quad (5.32)$$

Soit h^{l+1} , la première solution réalisable trouver avec l'algorithme de projection à l'itération $l + 1$; avec une valeur du gap de Fukushima égale à :

$$(S(h^l), h^{l+1} - h^l) + \frac{1}{2\alpha}(h^{l+1} - h^l, h^{l+1} - h^l) \quad (5.33)$$

puisque $0 \geq (S(h^l), h^{l+1} - h^l) \geq (S(h^l), h_{pcc}^l - h^l)$ et puisque $\frac{1}{2\alpha}(h^{l+1} - h^l, h^{l+1} - h^l)$ est plus grand ou égal à zéro, alors le $\text{gap}(h^l)$ est une borne inférieure de la première valeur de gap de Fukushima calculée à l'itération $l + 1$. Ainsi, au fur et à mesure que la différence entre la solution actuelle et le coût minimal décroît, la valeur absolue du gap de Fukushima initial sera normalement plus petit, et par conséquent, le nombre d'itérations nécessaires par l'algorithme de projection sera plus petit aussi.

5.4.1 Plus court chemin dynamique

Pour générer les nouveaux chemins, nous allons résoudre le *problème des plus courts chemins dynamique* (PCCD) de chaque origine à chaque destination et pour chaque $t \in T$, en fait, nous allons résoudre le PCCD dans des intervalles de 1 minute. Le problème des plus courts chemins a été largement étudié pour le cas statique donnant origine à une variété d'algorithmes pour le résoudre. Entre les plus "classiques", nous pourrions mentionner l'algorithme de Dijkstra (1959), l'algorithme de Ford-Moore-Bellman avec la variante d'Esopo-Pape (Pape 1974) et l'algorithme d'enchères de Bertsekas (1991). Nous allons utiliser, pour résoudre le PCCD, une adaptation dynamique de l'algorithme de Dijkstra.

Le problème consiste à trouver, pour chaque intervalle de départ, un arbre orienté T_{pcc} , qui est un graphe partiel du graphe G , tel que l'unique chemin qui relie l'origine O , au sommet j dans T_{pcc} est un plus court chemin de O à j dans G , pour tout j dans T_{pcc} .

L'intérêt porté, ces dernières années, aux problèmes dynamiques sur les systèmes de transport a mené aussi à l'étude de problèmes de plus courts chemins. Chabini (2000) a différencié plusieurs types de PCCD. Il analyse le PCCD selon le concept de coût sur les arcs, soit que le coût fasse référence au temps de parcours ou qu'il tienne compte d'autres facteurs. Il analyse aussi le PCCD selon si la condition de FIFO est vérifiée ou non, s'il y a un temps d'attente sur les noeuds, etc. Dans notre cas, où la fonction de coût tient compte seulement du temps de parcours, le problème est simplifié.

Nous avons implanté une adaptation de l'algorithme de Dijkstra. Kauffman et Smith (1993) ainsi que Ahn et Shin (1991) ont proposé des généralisations heuristiques de cet

algorithme. Cet algorithme est valide quand la condition de FIFO est vérifiée. Il faut se rappeler, que dans le modèle étudié ici, cette condition est assurée par la contrainte de propagation du flot.

Nous exécutons l'algorithme pour chaque intervalle de départ. Des étiquettes sont assignées aux sommets, mais à chaque révision, un sommet reçoit une *étiquette permanente* qui est la valeur du plus court chemin de l'origine vers ce sommet. On définit $l_i(t)$ comme la longueur du plus court chemin de l'origine au sommet i au moment de départ t ; $s_{ij}(t)$ comme le coût affecté à l'arc qui relie les sommets i et j au moment t ; V comme une liste de sommets candidats à ajouter au chemin; et O le sommet d'origine. L'algorithme est donné dans la suite :

Algorithme de Dijkstra temporel

répéter pour chaque temps de départ (t)

[Initialisation] $l_O = 0,$

$l_j := \infty, \forall j \in N, j \neq O,$

$V := 0.$

[Itération] **tant que** V n'est pas vide **faire** :

choisir $i \in V$ telle que $l_i = \min_{i \in V} l_j$

si $\exists j \in i^+$ et $j \neq O$ **faire**

si $l_j > l_i + s_{ij}(t + l_i)$ **faire**

$l_j := l_i + s_{ij}(t + l_i)$

si $j \notin V$ **faire**

ajouter j à V

fin de si

fin de si

fin de si

fin de tant que

Cet algorithme est valide si les longueurs des arcs sont toutes non négatives et si la condition de FIFO sur les arcs est vérifiée. On dit que c'est un algorithme d'*étiquetage permanent* parce que, pour chaque intervalle de départ, chaque sommet est introduit une et une seule fois dans la liste de candidats V . Lorsqu'un sommet est retiré de V , il lui est affecté une étiquette l qui demeure fixe par la suite. La valeur de cette étiquette est la longueur du plus court chemin de l'origine à ce sommet. Soit W , l'ensemble des sommets qui sont retirés de V , $W = \{i | l_i < \infty \text{ et } i \notin V\}$; nous pouvons remarquer les propriétés suivantes :

pour chaque intervalle de départ t :

- Lorsque i est retiré de V , i ne retourne jamais à V .
- À la fin de chaque itération, la valeur de l_i est :

Si $i \in W$: l_i est la longueur d'un plus court chemin du sommet origine à i .

Si $i \notin W$: l_i est la longueur d'un plus court chemin de l'origine vers i tel que tous les sommets intermédiaires du chemin sont dans W .

Pour chaque intervalle de départ, des étiquettes sont assignées aux sommets, mais à chaque révision un sommet reçoit une étiquette permanente qui est la valeur du plus court chemin effectif de l'origine vers ce sommet.

5.4.2 Algorithme de restriction

L'application de l'algorithme de restriction pour le problème d'affectation dynamique englobe les sous-problèmes suivants (voir figure 5.9) :

- Le problème de programmation quadratique pour résoudre le RPIV. Méthode de projection : pour résoudre la méthode de projection, nous avons également les sous-problèmes suivants :
 - Le problème de chargement du réseau : pour le résoudre nous utilisons l'Algorithme 2 présenté au chapitre 3, et
 - l'algorithme unidimensionnel pour trouver le "pas" de la méthode de projection, présenté à la section 6.2.1.
- Problème des plus courts chemins dynamiques : pour le résoudre nous utilisons l'algorithme de Dijkstra généralisé pour le cas dynamique.

Les calculs de l'algorithme de restriction sont initiés avec le réseau vide, sans flot, et pourtant avec les coûts des arcs avec flot nul. Par la suite, nous allons générer un seul chemin pour chaque paire origine-destination. Nous allons charger toutes les demandes temporelles sur ce chemin et nous calculerons les coûts pour chaque arc et chaque chemin à chaque intervalle de temps. Avec ces coûts ainsi trouvés, nous allons générer les nouveaux plus courts chemins et utiliser l'algorithme de projection pour équilibrer les flots sur le réseau en tenant compte seulement des chemins que nous avons générés.

Algorithme de restriction

- [initialisation] 1. $l := 0$; nb-chemins $:= 0$;
soit E^l l'ensemble de chemins disponibles. $E^0 := 0$.
- [itération] 2. Si une des conditions d'arrêt est satisfaite : arrêter.
3. $l := l + 1$.
4. Résoudre le problème de plus court chemin dynamique.
 $E^{l+1} = E^l +$ les chemins générés qui ne sont pas dans E^l .
5. Résoudre le RPIV sur l'ensemble des chemins E^l ;
 $h_{\text{initial}}^l = h_{\text{final}}^{l-1}$.

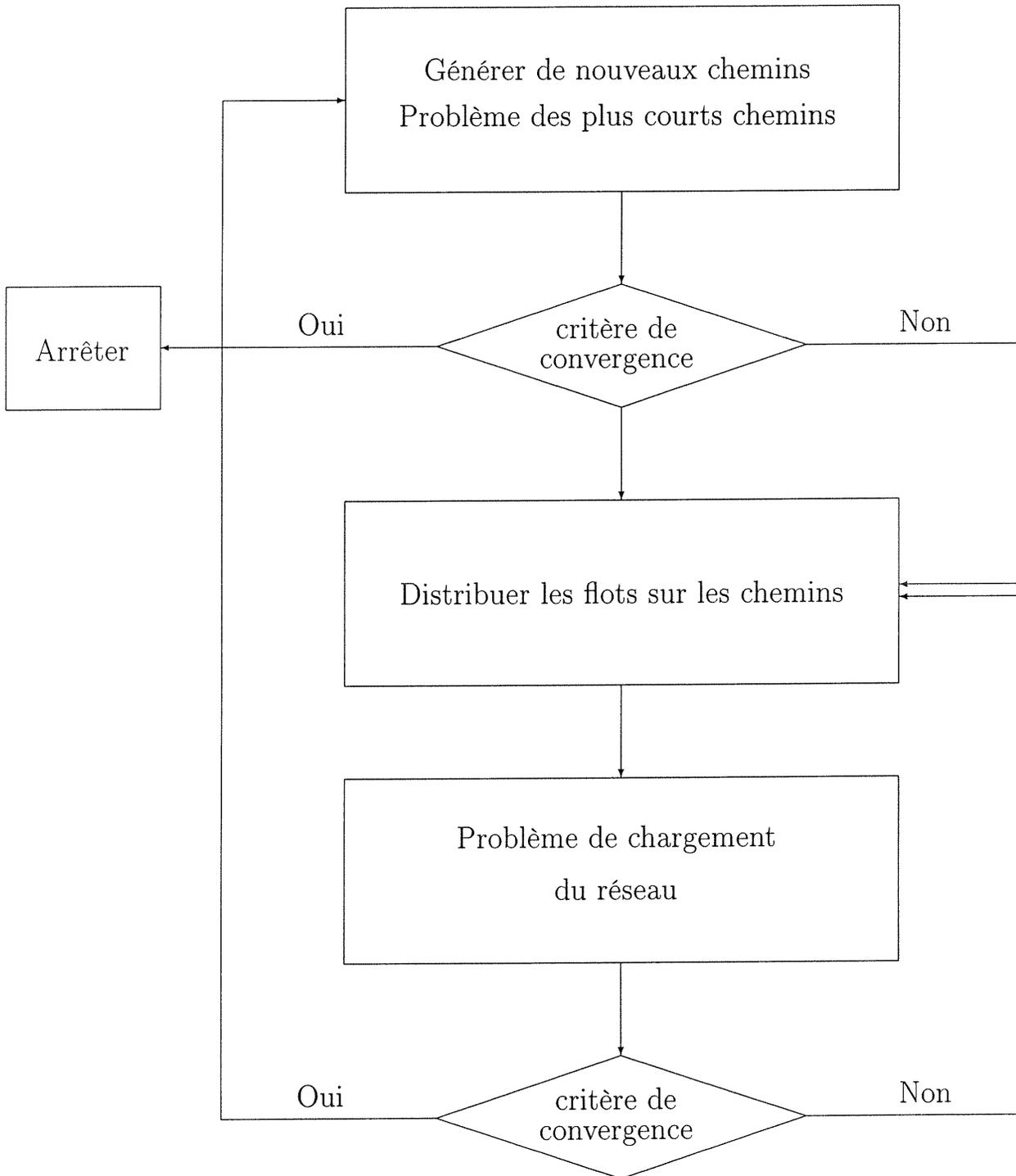


FIG. 5.9 – Résolution du problème d'équilibre dynamique.

5.5 Résultats numériques

Nous allons exposer dans cette section les résultats obtenus avec la méthode de restriction, en utilisant l'exemple du réseau de Hull, avec un intervalle de départ égal à 60 minutes.

À la première itération, nous générons un chemin pour chacune des 93 paires origine-destination. Nous résolvons ensuite le problème de chargement pour toutes les périodes de temps de départ à analyser sur le chemin que nous avons généré pour chaque paire origine-destination. Avec les nouveaux coûts des arcs, ainsi trouvés, nous allons résoudre, à nouveau, le problème des plus courts chemins dynamiques. Le nombre des nouveaux chemins générés est de 187, donc, nous avons au total 290 chemins. La valeur absolue de la fonction gap est 239314.25.

À la deuxième itération, nous allons faire 10 itérations de l'algorithme de projection. Nous présentons dans le tableau (5.14) la valeur de la norme de la différence entre deux

itération 2	
Itér	$\ h^{l+1} - h^l\ $
1	513.2651
2	401.5979
3	294.4817
4	225.0165
5	144.6669
6	83.7447
7	35.5476
8	26.2105
9	16.1793
10	12.0650

TAB. 5.14 – Réseau de Hull : méthode de restriction.

solutions de l'algorithme de projection.

Nous résolvons de nouveau le problème des plus courts chemins ; cette fois-ci, l'algorithme génère 108 nouveaux chemins, pour un total de 408. La valeur absolue de la fonction de gap est 9561. Puisque les flots des chemins sont sélectionnés tels qu'à chaque itération de l'algorithme de restriction nous commençons avec la solution de l'itération précédente, l'algorithme ne nécessite pas beaucoup d'itérations du sous-problème pour trouver une nouvelle solution d'équilibre. La valeur de la norme de la différence entre deux solutions de la méthode de projection est petite au début et elle décroît vers zéro. Le tableau (5.15) contient les résultats de la troisième itération. L'algorithme génère, cette fois-ci, 59 nouveaux chemins. La valeur absolue du gap décroît à 1628.

itération 3	
itér	$\ h^{l+1} - h^l\ $
1	56.9465
2	38.0240
3	29.2217
4	23.8473
5	16.2245
6	11.8766
7	9.1110
8	6.5304
9	5.8454
10	4.6546

TAB. 5.15 – Réseau de Hull : méthode de restriction.

La quatrième itération se réalise sur un domaine composé de 467 chemins et, pour les itérations successives, l'algorithme ne génère pas beaucoup de chemins. Les tableaux (5.16), (5.17), (5.18), (5.19) illustrent le comportement de la méthode de projection pour les itérations suivantes. Le tableau (5.20) contient le nombre de chemins par itération et la valeur absolue du gap.

itération 4	
itér	$\ h^{l+1} - h^l\ $
1	8.8776
2	7.1755
3	6.0716
4	4.9903
5	4.3334
6	3.6426
7	3.3393
8	2.7417
9	2.3950
10	2.0542

TAB. 5.16 – Réseau de Hull : méthode de restriction.

itération 5	
itér	$\ h^{l+1} - h^l\ $
1	3.6405
2	3.8564
3	3.7482
4	3.0331
5	2.2431
6	1.8737
7	1.5658
8	1.3671
9	1.2210
10	1.5163

TAB. 5.17 – Réseau de Hull : méthode de restriction.

itération 6	
itér	$\ h^{l+1} - h^l\ $
1	3.1280
2	2.4380
3	2.1020
4	1.7873
5	1.5604
6	1.3683
7	1.2592
8	1.0721
9	0.9526
10	0.8394

TAB. 5.18 – Réseau de Hull : méthode de restriction.

itération 7	
itér	$\ h^{l+1} - h^l\ $
1	2.1324
2	2.0156
3	1.6839
4	1.4893
5	1.3693
6	1.2028
7	1.0312
8	0.9191
9	0.8103
10	0.7191

TAB. 5.19 – Réseau de Hull : méthode de restriction.

itération	nombre de chemins	gap
1	93	239314.25
2	290	9561.00
3	408	1628.87
4	467	770.25
5	490	546.25
6	491	415.63
7	493	349.63
8	500	288.50
9	503	272.37

TAB. 5.20 – Réseau de Hull.

Le tableau (5.21) présente le nombre de chemins générés et le flot de chemins minimaux pour les dix premières paires origine-destination. Le tableau (5.22) contient les temps de parcours des chemins minimaux. Les arcs qui composent ces chemins sont présentés au tableau (5.23).

Paire O-D chemins	1		2	3	4	5	6			7	8			9	10	
	4	2	2	3	1	3			1	5			2	2		
	h[1]	h[4]	h[5]	h[7]	h[9]	h[12]	h[13]	h[14]	h[15]	h[16]	h[17]	h[18]	h[21]	h[22]	h[24]	h[25]
0.50	0.45	0.00	0.45	0.45	0.45	0.45	0.08	0.36	0.00	0.45	0.37	0.07	0.00	0.45	0.45	0.00
10.50	7.80	0.00	7.80	7.80	7.80	7.80	0.00	7.80	0.00	7.80	7.18	0.62	0.00	7.80	7.80	0.00
20.50	11.70	0.45	12.15	12.15	12.15	12.15	0.00	12.15	0.00	12.15	10.41	1.73	0.00	12.15	12.15	0.00
30.50	8.23	5.27	13.50	13.50	13.50	13.50	0.00	11.31	2.19	13.50	9.35	4.15	0.00	13.50	13.50	0.00
40.50	5.13	6.71	11.85	11.85	11.85	11.85	0.00	7.65	4.19	11.85	5.27	6.18	0.40	11.85	11.85	0.00
50.50	4.67	2.53	7.20	7.20	7.20	7.20	0.00	7.18	0.01	7.20	5.22	1.97	0.00	7.20	3.82	3.38

TAB. 5.21 - Réseau de Hull : taux de départ final ($T=60$).

Paire O-D	1		2	3	4	5	6			7	8			9	10	
	S[1]	S[4]	S[5]	S[7]	S[9]	S[12]	S[13]	S[14]	S[15]	S[16]	S[17]	S[18]	S[21]	S[22]	S[24]	S[25]
0.50	17.54	17.72	12.07	6.06	9.66	9.66	13.27	13.27	14.45	9.64	12.12	12.12	14.88	8.47	12.21	21.02
10.50	24.02	24.64	16.52	9.26	12.65	12.75	16.91	16.75	17.24	12.45	16.31	16.30	19.11	12.43	16.97	25.02
20.50	35.04	35.03	24.79	15.60	18.06	17.96	23.70	22.29	22.59	16.95	23.74	23.74	26.26	19.63	25.02	30.82
30.50	45.33	45.32	33.53	22.19	24.34	24.33	30.58	28.90	28.92	23.14	31.41	31.41	33.16	27.16	32.97	37.08
40.50	46.64	46.69	38.25	26.62	28.70	28.89	34.82	33.35	33.35	28.25	35.85	35.83	35.90	31.70	37.73	38.69
50.50	41.35	41.31	35.19	26.79	28.64	29.11	34.95	33.06	33.09	28.60	35.34	35.38	35.47	31.36	36.09	35.98

TAB. 5.22 - Réseau de Hull : temps de parcours final ($T=60$).

chemin 1	1 400	262 315	738 320	759 323	566	565	563	561	403	399
chemin 4	1 505	262 506	797 343	653 322	673	791	764	754	512	502
chemin 5	1	262	738	759	566	565	563	561	403	398
chemin 7	1	261	116	117	119					
chemin 9	1	261	116	118	183	185	410	713		
chemin 12	1	262	738	758	560	511	244	245		
chemin 13	1 741	261	116	117	119	17	447	451	454	199
chemin 14	1 743	262	738	758	560	511	244	246	249	252
chemin 15	1 192	261 739	116	118	183	184	187	188	718	189
chemin 16	1	263	798	654	674	792	765	754		
chemin 17	1	261	116	117	120	122	115	60	773	774
chemin 18	1	261	116	117	120	122	115	55	589	776
chemin 21	1 488	261 489	116	117	119	18	642	76	313	79
chemin 22	1	261	116	117	120	122	115			
chemin 24	1	261	116	117	120	122	115	59	771	766
chemin 25	1 668	263 656	798 627	654 786	674 603	794 604	790	347	388	386

TAB. 5.23 – Réseau de Hull, arcs qui forment les chemins minimaux.

5.5.1 Algorithme des proportions successives

Nous allons comparer les résultats numériques obtenus avec l'algorithme de restriction et les résultats obtenus avec l'algorithme de proportions successives. Cet algorithme consiste à distribuer le flot demandé en proportions égales entre les chemins qui relient chaque paire origine-destination. Nous allons générer des chemins comme dans l'algorithme de restriction précédent, mais au lieu de chercher l'équilibre avec l'algorithme de projection sur le nouveau domaine restreint, nous allons tout simplement distribuer en quantités égales le flot entre les chemins.

Nous exécutons cet algorithme avec l'exemple précédent. La condition d'arrêt est le nombre de chemins générés. Quand l'algorithme ne génère plus de nouveaux chemins, il s'arrête. L'exécution se termine après 6 itérations avec 360 chemins générés.

Paire O-D	1			2		3	
	h[1]	h[2]	h[3]	h[4]	h[5]	h[6]	h[7]
0.50	0.150	0.150	0.150	0.225	0.225	0.225	0.225
10.50	2.600	2.600	2.600	3.900	3.900	3.900	3.900
20.50	4.050	4.050	4.050	6.075	6.075	6.075	6.075
30.50	4.500	4.500	4.500	6.750	6.750	6.750	6.750
40.50	3.950	3.950	3.950	5.925	5.925	5.925	5.925
50.50	2.400	2.400	2.400	3.600	3.600	3.600	3.600

TAB. 5.24 – Réseau de Hull : méthode des proportions successives : flot sur les chemins.

Nous présentons, au tableau (5.24), le flot pour chaque chemin pour les trois premières paires origine-destination, qui est égal à la demande divisée par le nombre de chemins. Le tableau (5.25) montre les coûts de ces chemins et le coût minimal trouvé avec la méthode de restriction à la section précédente. Les tableaux (5.26) et (5.27) sont pour les paires origine-destination 4, 5, 6. Le nombre de chemins générés est plus petit que ceux générés avec la méthode de restriction à la section précédente. Les coûts de ces chemins sont différents,

il peut y avoir des chemins avec des coûts plus petits que le coût minimal trouvé avec la méthode de restriction, mais nous trouvons plusieurs chemins avec des coûts plus élevés. Les conditions d'équilibre sont loin d'être vérifiées. La valeur absolue du gap final est de 146161, à comparer à celle de 272 que nous avons trouvé à la section précédente.

Paire O-D	1				2			3		
	S[1]	S[2]	S[3]	coût min.	S[4]	S[5]	coût min.	S[6]	S[7]	coût min.
0.50	17.31	23.01	22.56	17.54	12.05	17.01	12.07	6.04	15.72	6.06
10.50	21.98	30.57	28.60	24.02	15.55	21.98	16.52	8.69	19.67	9.26
20.50	30.28	41.64	37.29	35.04	22.55	29.78	24.79	13.97	25.35	15.50
30.50	39.68	52.07	46.05	45.33	30.89	37.97	33.53	20.27	31.52	22.19
40.50	44.36	51.22	48.43	46.64	36.44	41.62	38.25	25.29	36.16	26.62
50.50	41.88	44.36	44.17	41.35	34.65	37.79	41.31	26.25	36.17	35.19

TAB. 5.25 – Réseau de Hull : méthode des proportions successives : temps de parcours.

Paire O-D	4			5			6		
	h[8]	h[9]	h[10]	h[11]	h[12]	h[13]	h[14]	h[15]	h[16]
0.50	0.150	0.150	0.150	0.150	0.150	0.150	0.150	0.150	0.150
10.50	2.600	2.600	2.600	2.600	2.600	2.600	2.600	2.600	2.600
20.50	4.050	4.050	4.050	4.050	4.050	4.050	4.050	4.050	4.050
30.50	4.500	4.500	4.500	4.500	4.500	4.500	4.500	4.500	4.500
40.50	3.950	3.950	3.950	3.950	3.950	3.950	3.950	3.950	3.950
50.50	2.400	2.400	2.400	2.400	2.400	2.400	2.400	2.400	2.400

TAB. 5.26 – Réseau de Hull : méthode de proportions successives : flot sur les chemins.

Paire O-D	4				5				6			
	S[8]	S[9]	S[10]	coût min.	S[11]	S[12]	S[13]	coût min.	S[14]	S[15]	S[16]	coût min.
0.50	9.65	12.09	12.17	9.66	9.66	12.18	12.12	9.66	13.32	13.35	14.50	13.27
10.50	12.43	15.35	15.76	12.65	12.76	15.27	15.21	12.75	16.70	17.04	17.29	16.75
20.50	17.21	20.37	21.16	18.06	17.78	20.18	20.16	17.96	22.57	22.49	22.01	22.29
30.50	22.97	26.12	27.11	24.34	23.59	25.83	25.88	24.33	29.09	28.39	27.68	28.90
40.50	27.55	30.54	31.62	28.70	28.06	30.27	30.29	28.89	34.01	32.74	32.21	33.35
50.50	28.12	30.74	31.93	28.64	28.37	30.72	30.55	29.11	34.98	33.01	32.87	33.06

TAB. 5.27 – Réseau de Hull : méthodes des proportions successives : temps de parcours.

Chapitre 6

Modèle d'équilibre dynamique avec capacités sur les arcs

Dans ce chapitre, nous allons formuler et développer un modèle analytique pour le problème d'équilibre dynamique sur un réseau avec capacités sur les arcs. Nous considérons aussi le phénomène de *débordement vers l'arrière* (“*spill-back*”) produit par un goulot d'étranglement. Afin de bien représenter la dynamique du flot du trafic sur un réseau urbain, il est nécessaire de tenir compte des capacités finies sur les rues. Ces capacités ont des répercussions sur les conditions du réseau quand il est congestionné. Une rue pourrait être saturée jusqu'au point où il n'y aurait pas d'espace pour d'autres véhicules, ou il pourrait arriver aussi, qu'à un moment donné, la quantité de véhicules qui veulent entrer ou sortir d'une rue soit supérieure à l'espace disponible.

Pour modéliser des situations semblables, nous allons considérer une extension dans notre modèle avec des limites de capacité de stockage, d'entrée et de sortie sur les arcs. Si les capacités sont atteintes, il y aura du flot qui ne pourra pas entrer ou sortir de l'arc et, au même moment, une file pourrait se former et s'étendre, par le fait même, vers l'arrière, arc par arc.

La grande difficulté de ce problème réside dans la nécessité de concevoir un modèle et un algorithme pour résoudre le nouveau problème de chargement dynamique avec capacités limitées. Nous allons présenter le problème de chargement à la section 6.1, la formulation mathématique à la section 6.2, un algorithme de résolution à la section 6.3 et à la section 6.4, nous présenterons des résultats numériques. La méthode de projection sera utilisée avec le nouvel algorithme de chargement de réseau, avec capacités explicites sur les arcs, pour trouver une solution d'équilibre.

6.1 Problème de chargement dynamique avec capacités

Dans cette section, nous allons développer un modèle pour le problème de chargement. Nous nous inspirerons du modèle *transmission de cellule* (“*Cell Transmission Model*”) de Daganzo (1994) qui est une variante du modèle hydro-dynamique formulé par Lighthill et Whitham (1995) et Richards (1956) pour modéliser les flots de trafic. Dans le modèle de Daganzo, les arcs sont divisés en petites unités appelées cellules. Les longueurs de ces cellules ne sont pas choisies arbitrairement. Elles sont égales à la distance parcourue par un véhicule dans des conditions de trafic léger (sans congestion) par unité de temps. Le flot avance d'une cellule à l'autre selon la relation entre le flot présent dans la cellule à un moment donné, la capacité du flot d'entrée et la quantité d'espace disponible à la cellule suivante.

Daganzo suppose que chaque cellule a une valeur maximale du flot qui peut sortir et une valeur maximale du flot qui peut entrer à chaque instant. Ziliaskopoulos et Lee (1997) introduisent le concept de *cellule non-homogène*, c'est-à-dire que le temps pour parcourir

une cellule n'est pas nécessairement égal à l'unité de temps, le temps pour parcourir la cellule peut être un multiple de l'unité de temps. Adamo et al. (1999a) présentent un modèle analytique inspiré aussi du modèle de Daganzo, en tenant compte du comportement des véhicules noeud par noeud, selon les capacités et les distributions sur les arcs.

Les modèles que nous avons présentés antérieurement ainsi que les autres qui les ont précédés (Friesz et al, 1993, Wu et al, 1998, Xu et al, 1999) n'ont pas des capacités imposées sur les arcs. Le phénomène de débordement vers l'arrière ne peut pas être reflété dans ces modèles. Le modèle simplifié considéré dans la suite suppose que chaque arc a une seule voie de trafic et le changement de voie n'est pas considéré.

6.2 Formulation mathématique

Nous développons le modèle en considérant les limites des capacités d'entrée, de sortie et de stockage du flot sur les arcs. Le modèle divise le temps de déplacement sur un arc en deux parties qui correspondent au temps de déplacement même et au temps d'attente. La première partie représente le mouvement de flot à l'intérieur d'un arc, il sera représenté par un lien qui servira à modéliser les déplacements des usagers sur les arcs du réseau, la propagation du flot, selon certaines règles. Nous allons définir la formulation analytique pour décrire la propagation du flot de telle façon que le mouvement du flot à chaque instant soit cohérent avec le temps de parcours sur les arcs. Cette formulation assure que la condition FIFO est vérifiée. La deuxième partie du modèle représente le comportement du flot quand la fin de l'arc est atteinte. Ce comportement sera dépendant des limites de capacité de l'arc en question et des arcs adjacents rentrant au même sommet ainsi que des capacités des

arcs sortant de ce sommet. Donc, nous allons considérer le comportement du flot à chaque sommet et la conservation du flot. Également, une file d'attente accumulera tous les flots qui auront fini de parcourir un arc, mais qui n'auront pas pu continuer à cause des limites de capacité.

La quantité de flot qui est prêt à sortir d'un arc à un moment donné est appelée *demande locale* et la quantité de flot qui peut entrer dans l'arc est appelée *offre locale* ou *disponibilité*. Chaque sommet sera géré selon la demande des arcs qui entrent et la disponibilité des arcs sortants. Évidemment, il faut satisfaire aussi la condition FIFO : un flot ne pourra pas sortir si celui qui est devant lui n'est pas encore sorti, même si leurs chemins sont différents.

Nous allons utiliser le flot accumulé plutôt que le taux (ou flot) de véhicules. Le délai de parcours et le volume du flot entre deux sommets, par exemple, ainsi que l'évolution des files pourront être calculés plus facilement.

Dans ce modèle, nous déterminons plusieurs chemins pour chaque paire origine-destination. Chaque chemin est constitué par un arc initial qui sera une rampe d'entrée pour chaque origine, avec une capacité infinie, pour permettre à tous les flots d'entrer dans le réseau. Le temps de parcours de cet arc sera égal à zéro et sa capacité de sortie infinie. Il y aura aussi un arc fictif final, pour chaque destination, qui représentera une rampe de sortie, également, avec une capacité d'entrée et de stockage infinie. Cet arc sera utilisé seulement pour permettre la sortie du réseau de tous les flots qui ont fini leur parcours. Dans nos résultats numériques, nous ne tenons pas compte de cette rampe, c'est-à-dire que nous considérons

que le flot qui entre dans cette dernière rampe a fini son parcours.

6.2.1 Notation

Nous présentons une nouvelle notation, que nous utilisons pour le cas avec capacités limitées et qui s'ajoute à la notation que nous avons utilisée jusqu'à maintenant

Nous allons noter $C_a^{in}(l^-, l^+)$ et $C_a^{out}(l^-, l^+)$ ($C_a^{in}(l^-, l^+) = C_a^{out}(l^-, l^+)$) la capacité d'entrée et sortie d'un arc a dans l'intervalle de temps (l^-, l^+) . La notation Q_a^{st} représentera la capacité de stockage de l'arc a . La demande d'un arc a dans l'intervalle de temps (l^-, l^+) sera représentée par $\Delta_a(l^-, l^+)$; $\Delta_{ka}(l^-, l^+)$ sera la demande de l'arc a correspondant au chemin k dans l'intervalle de temps (l^-, l^+) et $\Delta_{a,b}(l^-, l^+)$ la demande de l'arc a vers l'arc b dans l'intervalle de temps (l^-, l^+) . L'offre de l'arc a dans l'intervalle de temps (l^-, l^+) sera notée $\Phi_a(l^-, l^+)$; l'offre de l'arc a correspondant au chemin k dans l'intervalle de temps (l^-, l^+) sera $\Phi_{ka}(l^-, l^+)$; et nous représenterons l'offre de l'arc b à l'arc a dans l'intervalle de temps (l^-, l^+) par $\Phi_{a,b}(l^-, l^+)$.

La notation $Br_{ka}(t)$ représentera la quantité des flots accumulés qui sont entrés dans l'arc a correspondant au chemin k au temps $t \in [0, T']$; et $Br_a(t)$ la quantité des flots accumulés qui sont entrés dans l'arc a au temps $t \in [0, T']$; $Br_a(t) = \sum_{k \in K^a} Br_{ka}(t)$. Nous utiliserons la notation $Bq_{ka}(t)$ pour représenter la quantité des flots accumulés qui sont entrés dans la file d'attente de l'arc a correspondant au chemin k au temps $t \in [0, T']$; et $Bq_a(t)$ la quantité des flots accumulés qui sont entrés dans la file d'attente de l'arc a au temps $t \in [0, T']$; $Bq_a = \sum_{k \in K^a} Bq_{ka}(t)$. De la même façon $Er_{ka}(t)$ sera la quantité des flots accumulés qui ont fini de parcourir l'arc a correspondant au chemin k au temps

$t \in [0, T']$; $Er_a(t)$ la quantité des flots accumulés qui ont fini de parcourir l'arc a au temps $t \in [0, T']$; $Er_a(t) = \sum_{k \in K^a} Er_{ka}(t)$; $Eq_{ka}(t)$ la quantité des flots accumulés qui sont sortis de la file d'attente de l'arc a correspondant au chemin k au temps $t \in [0, T']$ ainsi $Eq_a(t)$ sera la quantité des flots accumulés qui sont sortis de la file d'attente de l'arc a au temps $t \in [0, T']$; $Eq_a(t) = \sum_{k \in K^a} Eq_{ka}(t)$;

Nous allons représenter par $v_{ra}(t)$ le volume du flot qui est en train de traverser l'arc a à l'instant t ; de même, $v_{qa}(t)$ le volume du flot qui est dans la file de l'arc a à l'instant t ; $s_{ra}(v)$ représentera la fonction de temps de parcours de l'arc a (le temps d'attente n'est pas inclus); et $\tau_{ra}(t)$ le moment dans lequel le flot qui est entré dans l'arc a à l'instant t fini de parcourir l'arc (le temps d'attente n'est pas inclus); et son inverse est représenté par $\tau_{ra}^{-1}(t)$.

6.2.2 Condition FIFO

La condition du premier arrivé, premier servi, FIFO, doit être vérifiée pour chaque arc. Donc, l'ordre du flot sera toujours le même, selon l'ordre d'entrée, tout au long du parcours de l'arc. Ainsi, nous pourrions numéroter le flot par ordre d'entrée, et ce même ordre sera conservé tout au long du parcours de l'arc jusqu'à la sortie. Par exemple, si au moment t le dernier flot à entrer (ou sortir) dans l'arc est numéroté $B_a(t)$, cela implique qu'il y a eu un flot accumulé de $B_a(t)$ qui est déjà entré (ou sorti) dans l'arc. Nous pourrions le numéroter aussi selon le chemin qu'il est en train de parcourir; par exemple, si au moment t , le dernier flot à entrer dans l'arc a appartenant au chemin k est le flot $B_{ka}(t)$ et le dernier appartenant au chemin k' est le $B_{k'a}(t)$, cela implique qu'au moment t il y a eu $B_{ka}(t)$ et $B_{k'a}(t)$ flots, pour chaque chemin, qui sont entrés dans l'arc a , au total $B_{ka}(t) + B_{k'a}(t)$, mais le flot $(B_{ka}(t) + 1)$ ne pourra pas quitter l'arc si le $B_{k'a}(t)$ ne l'a pas fait avant, parce qu'il est

entré plus tard (Si $B_{ka}(t') = B_{ka}(t) + 1$ alors $t' > t$).

Nous pouvons numéroter le flot aussi selon l'arc où il se dirige, $B_{ab}(t)$. Ainsi, nous allons étiqueter, d'une façon fictive, trois fois le flot avec ces trois étiquettes $[B_a(t), B_{ka}(t), B_{ab}(t)]$, lesquelles indiquent respectivement l'ordre d'entrée dans l'arc a de flot, l'ordre d'entrée du flot qui suit le chemin k , et l'ordre d'entrée de celui que se dirige vers l'arc b , respectivement. La condition FIFO doit être vérifiée sur les arcs, sur les chemins et pour chaque paire d'arcs.

6.2.3 Propagation du flot

Dans cette section, nous allons définir les formulations pour décrire les déplacements du flot à l'intérieur d'un arc. Cette première partie du modèle (temps de déplacement) est basée sur la formulation présentée au chapitre 4 pour le cas sans limite de capacité.

$$v_{ra}(t) = Br_a(t) - Er_a(t), \quad a \in A, \quad 0 \leq t \leq \tau_{ra}(T_a), \quad (6.1)$$

$$\tau_{ra}(t) = t + s_{ra}(v_a(t)), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (6.2)$$

$$Er_a(\tau_{ra}(t)) = \begin{cases} 0 & 0 \leq t \leq s_{ra}(0), \\ Br_a(t) & s_{ra}(0) \leq t \leq T_a, \end{cases} \quad a \in A. \quad (6.3)$$

L'équation (6.1) définit le volume temporel de flot qui est en train de traverser l'arc a au moment t . La quantité de flot qui est en train de traverser l'arc est égale à la quantité cumulée de flot qui est entré dans l'arc moins la quantité de flot qui a déjà fini de le parcourir. L'équation (6.2) définit le temps de sortie, moment auquel, le flot qui est entré dans l'arc a au temps t , termine de le parcourir. L'équation (6.3) calcule la quantité de flot qui a fini de parcourir l'arc, et assure la condition FIFO sur la propagation du flot. Dérivons (6.3) par rapport à t :

$$e_{ra}(\tau_{ra}(t))\tau'_{ra}(t) = b_{ra}(t), \quad a \in A, \quad 0 \leq t \leq T_a, \quad (6.4)$$

$$\tau'_{ra}(t) = \frac{b_{ra}(t)}{e_{ra}(\tau_{ra}(t))}, \quad a \in A, \quad 0 \leq t \leq T_a. \quad (6.5)$$

Puisque $b_{ra}(t) > 0$ et $e_{ra}(\tau_{ra}(t)) > 0$ alors $\tau'_{ra}(t) > 0$ et ceci implique la vérification de la condition FIFO. ($\tau'_{ra}(t)$ dénote la dérivée de $\tau_{ra}(t)$ par rapport à t .)

6.2.4 Conservation du flot

Dans le cas où la limite de capacité n'est pas considérée, pour chaque chemin, le flot qui a fini de parcourir un arc va passer directement à l'arc suivant, puisqu'il n'y a pas de file d'attente. Puisque la contrainte de conservation du flot doit être vérifiée, la quantité de flot qui entre dans un arc est égale à la quantité de flot qui a fini de parcourir l'arc précédent. Si l'arc en question est le premier du chemin, alors la quantité de flot qui entre dans l'arc sera égale à la quantité qui entre dans le chemin, c'est-à-dire, l'intégrale du taux d'entrée.

Dans le cas présent, avec une limite maximale de capacité, le flot qui a fini de parcourir l'arc va se placer dans une file d'attente. Si la quantité de flot dans la file est plus grand que la capacité de sortie de l'arc, ou que la capacité d'entrée de l'arc suivant, ou s'il n'y a pas assez d'espace dans l'arc suivant pour la contenir, il y aura une certaine quantité de flot qui sera retenu dans la file, à la sortie de l'arc. Donc, la conservation du flot sera vérifiée si la quantité de flot accumulé qui a fini de parcourir l'arc est égale à la quantité de flot qui est entré à l'arc suivant plus la quantité de flot qui est retenu temporellement dans une file d'attente.

Nous commencerons par analyser le cas le plus simple, c'est-à-dire, un noeud qui a seulement un arc qui entre et un qui en sort, pour ensuite, considérer le cas plus général où

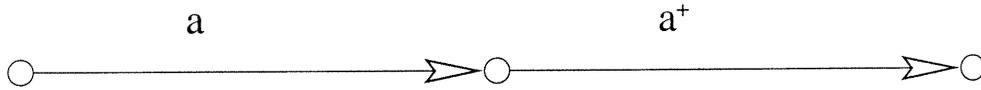


FIG. 6.1 – Noeud simple.

il peut y avoir plusieurs arcs adjacents à un noeud.

6.2.5 Noeud simple

Soit une intersection simple de deux arcs représentée par un noeud avec un seul arc qui entre, a , et un autre qui sort, a^+ . Figure (6.1).

Tous les flots qui ont fini de parcourir l'arc a au temps t sont entrés dans la file d'attente de l'arc a :

$$Bq_a(t) = Er_a(t), \quad a \in A, \quad s_{ra}(0) \leq t \leq \tau_{ra}(T_a), \quad (6.6)$$

et, ou bien ils sont sortis de la file, $Eq_a(t)$, pour entrer dans l'arc suivant $Br_{a^+}(t)$:

$$Br_{a^+}(t) = Eq_a(t), \quad a \in A, \quad s_{ra}(0) \leq t \leq T', \quad (6.7)$$

ou bien, à cause de la congestion, ils attendent encore dans la file d'attente de a . Nous avons, alors,

$$v_{qa}(t) = Bq_a(t) - Eq_a(t), \quad a \in A, \quad s_{ra}(0) \leq t \leq T', \quad (6.8)$$

et par (6.6), il suit que :

$$v_{qa}(t) = Er_a(t) - Eq_a(t), \quad a \in A, \quad s_{ra}(0) \leq t \leq T', \quad (6.9)$$

et finalement par (6.7) :

$$Bq_a(t) = Br_{a^+}(t) + v_{qa}(t), \quad a \in A, \quad s_{ra}(0) \leq t \leq T'. \quad (6.10)$$

Pour chaque intervalle de temps (l^-, l^+) , nous devons calculer la demande locale de flot qui est réellement prêt à sortir de l'arc ainsi que l'offre locale ou la quantité de flot qui

est admis à entrer dans chaque arc. C'est-à-dire, la demande sera la quantité de flot qui est dans la file, si celle-ci est inférieure à la capacité maximale de sortie ; sinon, la demande sera égale à la capacité de sortie :

$$\Delta_a(l^-, l^+) = \min\{Er_a(l^+) - Eq_a(l^-), C_a^{out}(l^-, l^+)\}. \quad (6.11)$$

Si la file est vide $Eq_a(l^-)$ est égal à $Er_a(l^-)$.

L'offre locale sera égale à la capacité maximale de flot qui pourra entrer dans l'arc (capacité d'entrée) s'il y a assez d'espace dans l'arc ; sinon, l'offre sera dépendante de l'espace disponible, c'est-à-dire la capacité maximale de stockage de l'arc moins la quantité du flot qui l'occupait déjà au début de l'intervalle plus la quantité du flot qui sort de l'arc pendant l'intervalle.

$$\Phi_a(l^-, l^+) = \min\{C_a^{in}(l^-, l^+), Q_a^{st} - v_a(l^-) + (Eq_a(l^+) - Eq_a(l^-))\}. \quad (6.12)$$

Évidemment, la quantité du flot qui va entrer dans l'arc a^+ est égale à la quantité qui sort de l'arc a , et celle-ci sera déterminée par la plus petite valeur entre la demande de a et l'offre de a^+ .

$$Eq_a(l^+) = Eq_a(l^-) + \min\{\Delta_a(l^-, l^+), \Phi_{a^+}(l^-, l^+)\}. \quad (6.13)$$

Mais, si l'arc a est traversé par plusieurs chemins, il faut distribuer le flot qui sort de l'arc entre les différents chemins. Nous devons alors calculer $Eq_{ka}(l^+)$ pour chaque chemin k qui traverse a de telle façon que la condition de FIFO soit vérifiée.

Nous allons chercher \bar{t} tel que $Eq_a(l^+) = Br_a(\bar{t})$, \bar{t} est le moment (temps) où le flot "numéroté" $Eq_a(l^+)$ entre dans l'arc a .

Nous allons définir $B_a^{-1}(x)$ comme l'inverse de $B_a(t)$, le moment où le flot "numéroté" x entre dans l'arc a . Alors,

$$\bar{t} = B_a^{-1}(Eq_a(l^+)) \quad (6.14)$$

et nous distribuerons le flot par chemin, $\forall k \in K^a$:

$$Eq_{ka}(l^+) = Br_{ka}(\bar{t}), \quad (6.15)$$

$$Br_{ka}(l^+) = \begin{cases} \int_0^{\bar{t}} h_k(s) ds, & \text{si } a \text{ est le premier arc du chemin } k, \\ Eq_{ka^-}(l^+), & \text{sinon,} \end{cases} \quad (6.16)$$

$$Br_a(l^+) = \sum_{k \in K_a} Br_{ka}(l^+), \quad (6.17)$$

$$Eq_a(l^+) = \sum_{k \in K_a} Eq_{ka}(l^+). \quad (6.18)$$

La quantité de flot qui sort de a est, par (6.13), la plus petite quantité entre $\{Eq_a(l^-) + \Delta_a(l^-, l^+)\}$ et $\{Eq_a(l^-) + \Phi_{a^+}(l^-, l^+)\}$. Puisque la condition de FIFO est vérifiée, cette quantité peut être déterminée par celle qui est entrée avant dans l'arc a , et nous pouvons remplacer (6.13) par :

$$Eq_a(l^+) = Br_a(\bar{t}), \quad (6.19)$$

où

$$\bar{t} = \min(\bar{t}_1, \bar{t}_2), \quad (6.20)$$

et

$$\bar{t}_1 = B_a^{-1}(Eq_a(l^-) + \Delta_a(l^-, l^+)), \quad (6.21)$$

$$\bar{t}_2 = B_a^{-1}(Eq_a(l^-) + \Phi_{a^+}(l^-, l^+)). \quad (6.22)$$

En résumé, le problème du chargement dynamique du réseau pour le cas simple est représenté par le modèle suivant :

Modèle pour le cas simple

Soit l'intervalle de temps (l^-, l^+) et $\forall a \in A$:

$$v_{ra}(l^+) = Br_a(l^+) - Er_a(l^+)$$

$$\tau_{ra}(l^+) = l^+ + s_{ra}(v_a(l^+))$$

$$Er_a(\tau_{ra}(l^+)) = \begin{cases} 0 & 0 \leq l^+ \leq s_{ra}(0) \\ Br_a(l^+) & s_{ra}(0) \leq l^+ \leq T_{ka} \end{cases} \quad a \in A$$

$$v_{qa}(l^+) = Bq_a(l^+) - Eq_a(l^+)$$

$$Eq_a(l^+) = Eq_a(l^-) + \min\{\Delta_a(l^-, l^+), \Phi_{a^+}(l^-, l^+)\} \quad (6.23)$$

$$Bq_a(l^+) = Er_a(l^+)$$

$$\Delta_a(l^-, l^+) = \min\{Er_a(l^+) - Eq_a(l^-), C_a^{out}(l^-, l^+)\}$$

$$\Phi_a(l^-, l^+) = \min\{Q_a^{in}(l^-, l^+), C_a^{st} - v_a(l^- + (Eq_a(l^+) - Eq_a(l^-)))\}$$

$$Eq_{ka}(l^+) = Br_{ka}(B_a^{-1}(Eq_a(l^+)))$$

$$Br_{ka}(l^+) = \begin{cases} \int_0^{l^+} h_k(s) ds & \text{si } a \text{ est le premier arc du chemin } k \\ Eq_{ka^-}(l^+) & \text{sinon} \end{cases}$$

$$Br_a(l^+) = \sum_{k \in K_a} Br_{ka}(l^+)$$

$$Er_a(l^+) = \sum_{k \in K_a} Er_{ka}(l^+)$$

$$l^+ = \min_a \{\tau_{ra}(l^-)\}$$

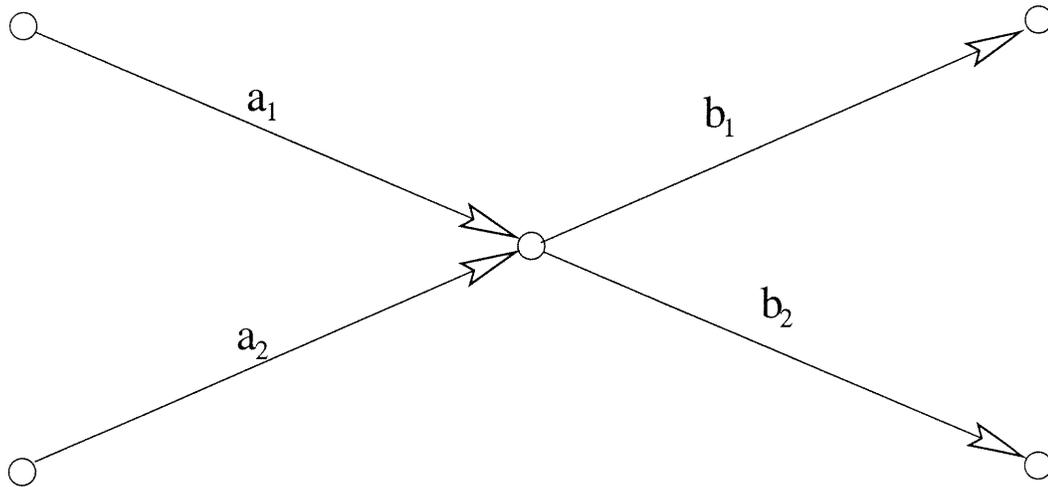


FIG. 6.2 – Noeud général.

La condition (6.23) peut être remplacée par (6.19), (6.20), (6.21), (6.22), c'est-à-dire trouver \bar{t}_1 et \bar{t}_2 tel que :

$$Bq_a(\bar{t}_1) = Eq_a(l^-) + \Delta_a(l^-, l^+) \quad (6.24)$$

$$Bq_a(\bar{t}_2) = Eq_a(l^-) + \Phi_{a^+}(l^-, l^+) \quad (6.25)$$

$$Eq_a(l^+) = Bq_a(\min(\bar{t}_1, \bar{t}_2)) \quad (6.26)$$

La capacité maximale de la file d'attente est égale à la capacité de l'arc.

6.2.6 Noeud général

Une intersection générale est représentée par un noeud avec plusieurs arcs qui sont incidents vers l'intérieur, les arcs a_1, a_2, \dots , et plusieurs autres qui sont incidents vers l'extérieur, les arcs b_1, b_2, \dots . Un exemple est illustré dans la figure (6.2).

Dans ce cas, il faut déterminer pour chaque demande locale Δ_a comment elle va se distribuer sur les arcs sortants. Nous allons calculer le moment \bar{t} où les Δ_a flots sont entrés

dans l'arc, et ainsi, les distribuer selon les chemins qu'ils sont en train de parcourir, pour calculer la demande de l'arc a correspondant au chemin k , Δ_{ka} . Nous allons calculer aussi la demande selon l'arc où les flots se dirigent, $\Delta_{a,b}$, la demande de l'arc a vers l'arc b . Une fois que nous avons calculé Δ_a selon (6.11), nous calculerons, Δ_{ka} , $\forall k \in K^a$ et $\Delta_{a,b}$, $\forall (a,b)$:

$$Bq_a(\bar{t}_1) = Eq_a(l^- + \Delta_a(l^-, l^+)), \quad (6.27)$$

$$\bar{t}_1 = B_a^{-1}[Eq_a(l^-) + \Delta_a(l^-, l^+)], \quad (6.28)$$

$$\Delta_{ka}(l^-, l^+) = B_{ka}(\bar{t}_1) - E_{ka}(l^-), \quad (6.29)$$

$$\Delta_{a,b}(l^-, l^+) = \sum_{k|(a,b) \in k} [B_{ka}(\bar{t}_1) - E_{ka}(l^-)]. \quad (6.30)$$

Nous avons obtenu ainsi les demandes $\Delta_{a,b}$ qui sont prêtes à sortir des arcs a et b dans l'intervalle (l^-, l^+) . Il faut regarder maintenant l'offre des arcs $b_1, b_2, \dots, \Phi_{b_1}, \Phi_{b_2}, \dots$. Si $\sum_a \Delta_{a,b}(l^-, l^+) \leq \Phi_b(l^-, l^+) \quad \forall b$, alors toute la demande est acceptée; sinon, il existe au moins un arc b' tel que $\sum_a \Delta_{a,b'}(l^-, l^+) > \Phi_{b'}(l^-, l^+)$, c'est-à-dire qu'il y a au moins un arc qui n'a pas assez d'offre pour laisser passer toute la demande. Dans ce cas, nous définirons $\Phi_{a,b'}$, comme l'offre de l'arc b' disponible pour chaque demande. $\sum_a \Phi_{a,b'}(l^-, l^+) = \Phi_{b'}(l^-, l^+)$. Différents modèles ont été présentés par Lebacque (1996) et Adamo et al. (1999) pour déterminer $\Phi_{a,b'}$. L'approche utilisé ici est de distribuer la disponibilité en quantités égales pour chaque arc a , puis réaffecter l'excédent aux arcs qui ont une demande non satisfaite.

Nous allons procéder de la façon suivante :

Calcul de $\Phi_{a,b'}$

Soit l'arc $b' = (q, r)$ et les arcs qui le précèdent $a_i = (p_i, q)$; soit $A_q^- := a_i$ et $|A_q^-|$ la cardinalité de A_q^- , c'est-à-dire le nombre d'arcs qui entrent au noeud q .

1. INITIALISATION

$$N := |A_q^-|, \text{ supp} := \Phi_b;$$

$$A_1 := A_q^-;$$

1. ITÉRATION

tant que $(\text{supp}/N) \geq \Delta_{a,b'}$ **faire**

pour $\forall a_i \in A$ **faire**

si $(\text{supp}/N) \geq \Delta_{a,b'}$ **alors**

$$\phi_{a,b'} = \Delta_{a,b'};$$

$$\text{supp} := \text{supp} - \Phi_{a,b'};$$

$$N := N - 1;$$

$$A_1 := A_1 - a_i;$$

fin de si

fin de pour

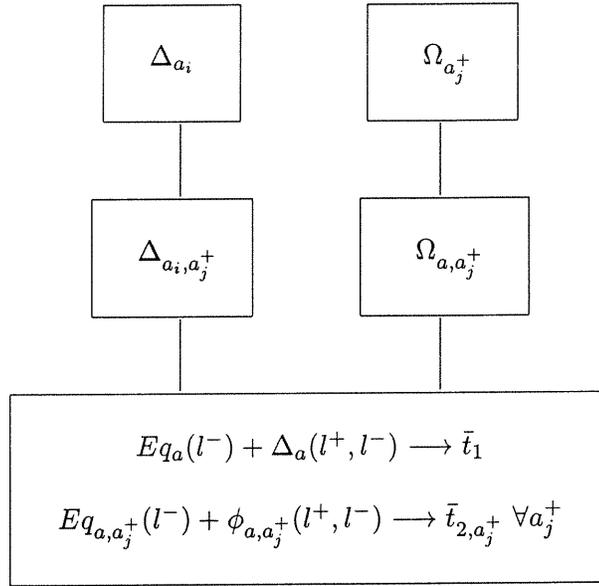
fin de tant que

pour $\forall a_i \in A_1$ **faire**

$$\Phi_{a,b'} := \text{supp}/N;$$

fin de pour

Puisque $\Phi_{b'}(l^-, l^+) = \sum_a \Phi_{a,b'}(l^-, l^+) < \sum_a \Delta_{a,b'}(l^-, l^+)$, il existe au moins un arc a' tel que $\phi_{a',b'}(l^-, l^+) < \Delta_{a',b'}(l^-, l^+)$. Cet arc (ou ces arcs) a' sera (seront) bloqué(s) par



les flots qui se dirigent vers l'arc b' . Seulement les premiers $\Phi_{a', b'}(l^-, l^+)$ flots parmi les $\Delta_{a', b'}(l^-, l^+)$ pourront sortir de a' pour entrer dans b' . Et, seulement les flots qui sont entrés dans la file de a' avant le flot $[Eq_{a', b'}(l^-) + \Phi_{a', b'}(l^-, l^+)]$ pourront sortir de a' . Ils seront déterminés par :

$$\bar{t}_2 = Bq_{a', b'}^{-1}[Eq_{a', b'}(l^-) + \Phi_{a', b'}(l^-, l^+)], \quad (6.31)$$

$$Eq_{ka'}(l^+) = Bq_{ka'}(\bar{t}_2), \quad (6.32)$$

$$Eq_{a', b'}(l^+) = Bq_{a', b'}(\bar{t}_2) = \sum_{k|(a', b') \in k} Eq_{ka}(\bar{t}_2). \quad (6.33)$$

Si nous avons deux ou plusieurs arcs b tels que $\sum_a \Delta_{a, b}(l^-, l^+) > \Phi_b(l^-, l^+)$, alors l'arc b' qui bloque l'arc a sera déterminé par : $\min_b(\bar{t}_{2, b})$ où $\bar{t}_{2, b}$ est défini par :

$$\bar{t}_{2, b} = Bq_{a', b}^{-1}[Eq_{a', b'}(l^-) + \Phi_{a', b'}(l^-, l^+)], \quad \forall a'. \quad (6.34)$$

Nous présentons, alors, le modèle de conservation du flot avec capacités pour le cas général :

Modèle pour le cas général

$$v_{ra}(l^+) = Br_a(l^+) - Er_a(l^+)$$

$$\tau_{ra}(l^+) = l^+ + s_{ra}(v_a(l^+))$$

$$Er_a(\tau_{ra}(l^+)) = \begin{cases} 0 & 0 \leq l^+ \leq s_{ra}(0) \\ Br_a(l^+) & s_{ra}(0) \leq l^+ \leq T_{ka} \end{cases} \quad a \in A$$

$$v_{qa}(l^+) = Bq_a(l^+) - Eq_a(l^+)$$

$$Bq_a(l^+) = Er_a(l^+)$$

$$\Delta_a(l^-, l^+) = \min\{Er_a(l^+) - Eq_a(l^-), C_a^{out}(l^-, l^+)\}$$

$$\Phi_a(l^-, l^+) = \min\{C_a^{in}(l^-, l^+), Q_a^{st} - v_a(l^-) + (Eq_a(l^+) - Eq_a(l^-))\}$$

$$Bq_a(\bar{t}_1) = Eq_a(l^-) + \Delta_a(l^+, l^-)$$

$$Bq_{a,a^+}(\bar{t}_{2,a^+}) = Eq_{a,a^+}(l^-) + \Phi_{a,a^+}(l^+, l^-)$$

$$Eq_a(l^+) = Bq_a(\min(\bar{t}_1, \bar{t}_{2,a^+}))$$

$$Eq_{ka}(l^+) = Bq_{ka}(\min(\bar{t}_1, \bar{t}_{2,a^+}))$$

$$Br_{ka}(l^+) = \begin{cases} \int_0^{l^+} h_k(s) ds, & \text{si } a \text{ est le premier arc du chemin } k, \\ Eq_{ka^-}(l^+), & \text{sinon,} \end{cases}$$

$$Br_a(l^+) = \sum_{k \in K_a} Br_{ka}(l^+)$$

$$Er_a(l^+) = \sum_{k \in K_a} Er_{ka}(l^+)$$

$$l^+ = \min_a \{\tau_{ra}(l^-)\}$$

Nous utilisons le total du flot accumulé et le temps d'entrée pour calculer le volume de flots sur les arcs, le temps effectif sur les arcs et le temps effectif sur les chemins. Le volume sur les arcs est donné par :

$$\begin{aligned} v_a(t) &= v_{ra}(t) + v_{qa}(t) \\ &= Br_a(t) - Er_a(t) + Bq_a(t) - Eq_a(t) \\ &= Br_a(t) - Eq_a(t) \\ &= Br_a(t) - Br_{a^+}(t) \end{aligned}$$

où la deuxième équation est donnée par (6.6) et (6.8), la troisième par (6.1) et la quatrième par (6.7). Le temps effectif sur les arcs est donné par : $s_a(v(t)) = \bar{t} - t$, où $\bar{t} = B_{a^+}^{-1}(B_a(t))$.

6.2.7 La file d'attente

Nous avons divisé les flots qui se trouvent sur l'arc a , $v_a(t)$, en deux parties, ceux qui sont en train de parcourir l'arc, $v_{ra}(t)$, et ceux qui sont dans la file d'attente, $v_{qa}(t)$. on a

$$v_a(t) = v_{ra}(t) + v_{qa}(t).$$

Il faut alors se poser la question, comment pourrions-nous modéliser la file d'attente ? D'une façon générale, nous utilisons les concepts de file verticale et file horizontale. Si la file n'occupe pas de place, elle est appelée verticale, sinon elle est appelée horizontale. Le fait d'utiliser une file verticale implique que les flots doivent parcourir l'arc entier même s'il y a des flots dans la file, puisqu'ils n'occupent pas de place sur l'arc. Mais cela n'est pas cohérent avec le fait d'avoir une capacité limitée. Si nous utilisons une file horizontale, les flots qui la constituent occupent de la place et la capacité disponible sur l'arc pourra varier

en conséquence. Mais les flots ont seulement une partie de l'arc à parcourir avant d'entrer dans la file qui sera la partie qui reste inoccupée de l'arc. Ceci est très difficile à modéliser, puisque la fonction de coût doit varier en conséquence. Cette variation serait considérée au moment de calculer la fonction de coût, c'est-à-dire au moment d'entrer sur l'arc. Mais la partie à parcourir, en fait, change aussi entre le moment d'entrer dans l'arc et le moment d'entrer dans la file, puisque la file est en train de changer aussi.

Nous allons donc choisir un modèle intermédiaire où la présence des flots dans la file n'a pas d'influence sur le temps de parcours d'un arc. Le temps de déplacement tient compte de l'arc entier. Mais les flots auront un effet sur la capacité; la capacité maximale de la file est égale à celle de l'arc, et l'espace disponible sur l'arc sera égal à sa capacité maximale moins l'espace occupé par la file.

6.3 Algorithme de chargement dynamique avec capacités

Nous allons définir $\bar{A}(t)$ comme l'ensemble des arcs actifs, c'est-à-dire arcs avec volume positif de flot au moment t , $hd(k)$ le premier arc du chemin k et HD l'ensemble des premiers arcs de chaque chemin $k \in K$, $\forall i \in I$. Nous appelons p l'extrémité initiale de a , et q l'extrémité terminale de a , $a = (p, q)$, $Pr(a)$ est l'ensemble des arcs avec extrémité terminale égale à p , c'est-à-dire ensemble des prédécesseurs de a , $Suc(a)$ est l'ensemble des arcs avec extrémité initiale égale à q , incluant a , et $In(p)$ est l'ensemble des arcs avec extrémité initiale égale à p , incluant a .

L'énoncé de l'algorithme de chargement dynamique avec capacité est le suivant :

Algorithme
1. INITIALISATION
 $i := 0, T_0 := 0;$
 $\bar{A}(t_0) := \{hd(k); \forall k \in K; \forall i \in I\};$
pour $a \in \bar{A}$ **faire**
 $l_a^- := 0;$
 $l_a^+ := s_{ra}(0);$
 $\rho_a :=$ le nombre de chemins $k \in K^a$ tel que a est l'arc de départ;

fin de pour
2. ITÉRATION
tant que $T_i < T$ ou $\bar{A} \neq 0$ **faire**
 $i := i + 1;$
 $T_i := \min\{l_a^+ : a \in \bar{A}\};$
pour $a = (p, q) \in \bar{A}$ **faire**

 ARC DYNAMICS($a = (p, q)$);

fin de pour
 $l_a^- := T_i;$ et $l_a^+ := \tau_a(T_i);$
si $\rho_a = 0$ **alors**
 $\bar{A} := \bar{A} - \{a\};$
fin de si
fin de tant que
3. ARC DYNAMICS ($a = (p, q)$) sur (T_{i-1}, T_i)

pour $n = 1, 2, 3, 4, 5$ **faire**

déterminer $t_n = T_{i-1} + \frac{T_i - T_{i-1}}{5} \times n$;

pour $a^- \in PR(a)$ **faire**

calculer $\Delta_{a^-}(t_{n-1}, t_n)$ selon (6.11);

calculer \bar{t}_{a^-} selon (6.28)

calculer $\Delta_{ka^-}(t_{n-1}, t_n)$ selon (6.29)

calculer $\Delta_{a^-,a}(t_{n-1}, t_n)$ selon (6.30)

pour $b \in In(p)$ **faire**

calculer $\Phi_b(t_{n-1}, t_n)$ selon (6.13)

fin de pour

si $\sum_{a^-} \Delta_{a^-,b}(t_{n-1}, t_n) \leq \Phi_b(t_{n-1}, t_n) \forall b \in In(p)$ **faire**

$Eq_{k,a^-}(t_n) := \Delta_{k,a^-}(t_{n-1}, t_n) + Eq_{ka^-}(t_n)$

fin de si

sinon

calculer $\Phi_{a^-,b}(t_{n-1}, t_n) \forall (a^-, b)$ selon l'algorithme de "calcul de $\Phi_{a,b}'$ "

calculer $\bar{t}_{2,b}$ selon (6.34)

calculer $Eq_{ka^-}(t_n)$ selon (6.32)

calculer $Eq_{a^-,b}(t_n)$ selon (6.33)

fin de sinon

calculer $Eq_{a^-}(t_n) = \sum_{k \in K^a} Eq_{ka^-}(t_n)$

calculer $v_{a^-} = Br_{a^-} - Eq_{a^-}$

fin de pour

calculer $Br_a(t_n) = Eq_{a^-}(t_n)$

fin de pour

$$\bar{A}^i := \bar{A}^i - In(p);$$

6.4 Résultats numériques

Dans cette section, nous allons présenter des exemples numériques. Nous avons implémenté l'algorithme précédant et nous avons résolu quelques problèmes pour le tester. Le premier exemple est un petit réseau montré dans la Figure 6.3 qui est constitué de 9 noeuds, 8 arcs et 4 chemins. Les noeuds a et b sont des origines et c et d des destinations. Chaque

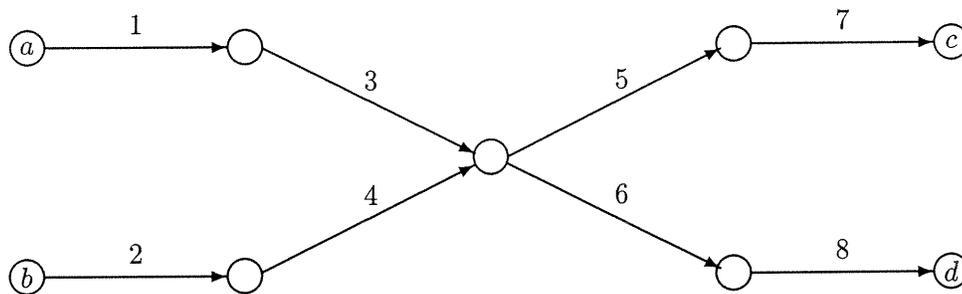


FIG. 6.3 – Petit réseau.

origine rejoint chaque destination par les chemins suivants :

chemin 1 : arcs : 1-3-5-7;

chemin 2 : arcs : 1-3-6-8;

chemin 3 : arcs : 2-4-5-7;

chemin 4 : arcs : 2-4-6-8.

La période de départ T est 10, et le taux de départ pour chaque chemin est fonction quadratique du temps donné par $h_k(t) = 0.15 \times (T - t) \times t$, $t = 0, 1, 2, \dots, 10$. La fonction de parcours est donnée par : $s_a(v_a(t)) = \beta_a^0 + \beta_a^1 \times v_a(t)$, $\forall a \in A$, où les paramètres β , les

arc	C^{out}	Q_{st}	β_a^0	β_a^1
1	∞	∞	1.6	0.10
2	∞	∞	1.4	0.10
3	2	10	2.3	0.20
4	6	15	2.3	0.20
5	4	15	2.2	0.40
6	3	15	2.3	0.20
7	∞	∞	1	0.10
8	∞	∞	1	0.10

TAB. 6.1 – Exemple 1 : paramètres utilisés.

limites de capacité d'entrée et de sortie par unité de temps $C^{out} = C^{in} = C$ et la capacité de stockage Q^{st} sont donnés au tableau (6.1). Les figures 6.4 et 6.5 montrent le flot total

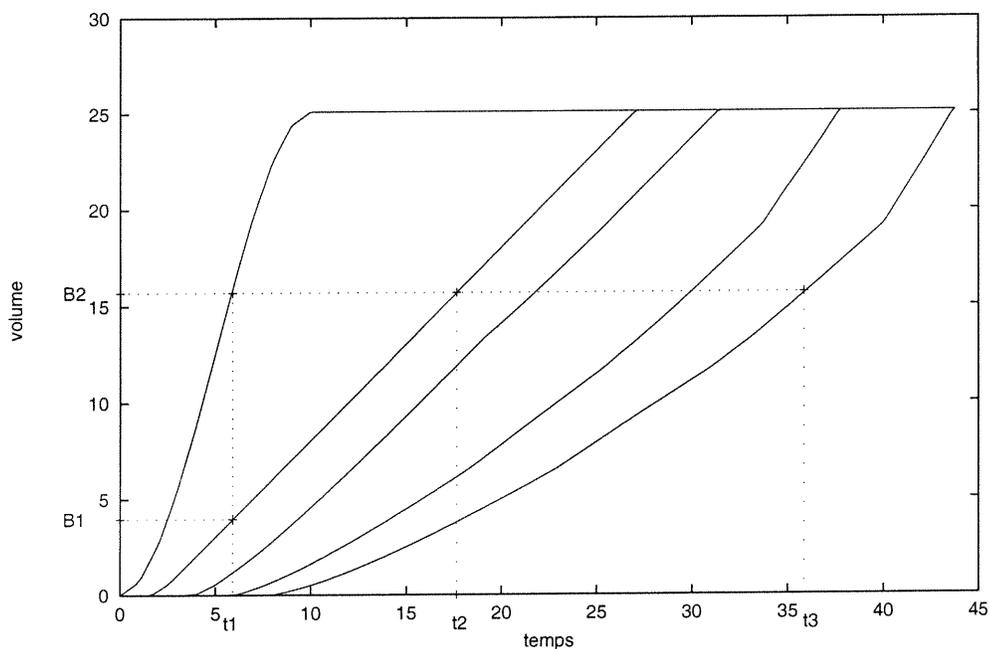


FIG. 6.4 – Exemple 1 : volume total accumulé sur les arcs du chemin 1.

cumulé qui est entré dans chacun des arcs qui appartiennent aux chemins 1 et 2, respectivement. C'est-à-dire, la figure 6.4 contient le graphe qui représente la quantité totale du flot qui est entré dans les arc 1, 3, 5, 7 (chemin 1), et la figure 6.5 le total du flot qui est

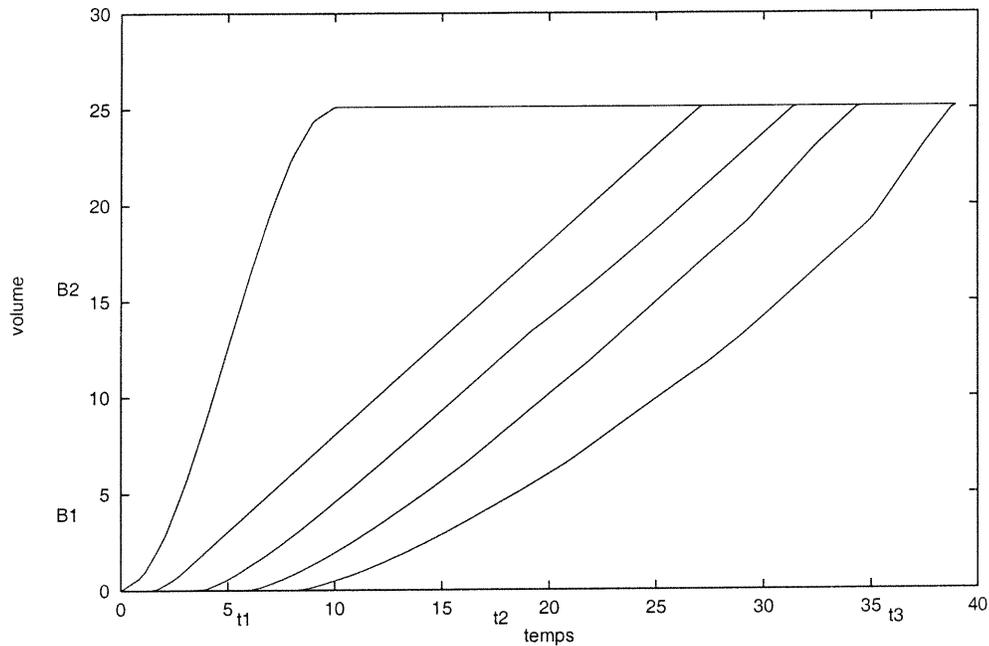


FIG. 6.5 – Exemple 1 : volume total accumulé sur les arcs du chemin 2.

entré dans les arcs 1, 3, 6, 8 (chemin 2). Nous pouvons voir que les courbes correspondant aux arcs 1 et 3 sont semblables sur les deux figures. La courbe de l'arc 5 sur la figure 6.4 et la courbe de l'arc 6 sur la figure 6.5 sont semblables aussi. Ceci est justifié, parce que si les conditions FIFO sont vérifiées la quantité du flot qui sort des arcs 3 et 4 vers l'arc 5 est la même que la quantité du flot qui sort des arcs 3 et 4 vers l'arc 6. Alors la quantité du flot qui entre dans les arcs 5 et 6 doit être la même. Sur le chemin 1, figure 6.4, le volume du flot qui entre au premier arc au temps t_1 quitte l'arc au temps t_2 , puisqu'au temps t_2 il entre au deuxième arc. Alors le temps de parcours de l'arc 1 au temps t_1 est égal à $t_2 - t_1$, $s_1(t_1) = t_2 - t_1$. Le temps de parcours du chemin pour le flot qui initie leur trajet au temps t_1 est égal à $t_3 - t_1$. Le flot sur le premier arc au temps t_1 est égal à $B_2 - B_1$.

Le deuxième exemple consiste à résoudre le problème d'équilibre dynamique avec limite de capacité sur les arcs. Nous utilisons la méthode de projection avec le *gap* de Fukushima et le problème de chargement du réseau avec capacité limitée sur les arcs. Pour déterminer les flots d'entrée nous résolvons le problème quadratique suivant :

$$\min_{h^{l+1} \in \Omega} (S(h^l), h^{l+1} - h^l) + \frac{1}{2\alpha} (h^{l+1} - h^l, h^{l+1} - h^l) \quad (6.35)$$

où α est une constante positive.

Le réseau utilisé est donné à la figure 6.6. Il est constitué de 11 noeuds, 13 arcs et 6 chemins. Le période de temps de départ est 15. Le taux de demande est donné par $d = |K| \times 0.15 \times (T - t) \times t$, où $|K|$ est le nombre de chemins, c'est-à-dire $|K| = 6$. La solution initiale est égale au même taux de départ pour tous les chemins. La fonction de temps de parcours est la même que dans l'exemple numérique précédent. Les paramètres β et les limites de capacité d'entrée, de sortie et de stockage sont donnés au tableau 6.2. Le critère de convergence est la valeur absolue du *gap* de Fukushima. La valeur absolue du *gap* de Fukushima à la première itération est 2382.80, la méthode de projection se termine après 20 itérations, avec une valeur absolue du *gap* de Fukushima de 1.59 à la dernière itération. Le tableau (6.5) montre les valeurs à chaque itération. Les tableaux (6.3) et (6.4) présentent le flot et le temps de parcours final (à la dernière itération) à intervalles d'une minute.

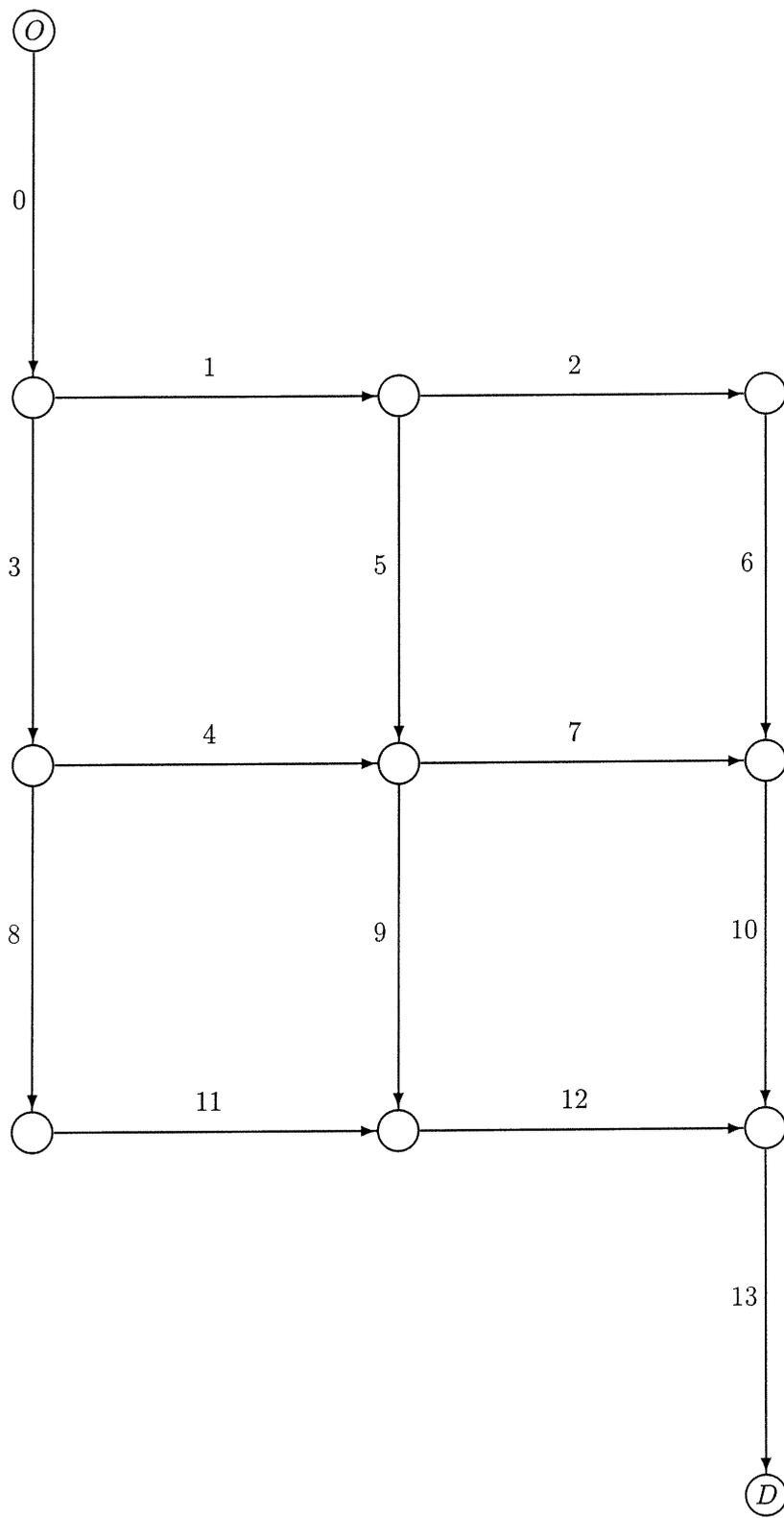


FIG. 6.6 - Réseau carré

arc	C^{out}	Q_{st}	β_a^0	β_a^1
0	∞	∞	1.6	0.10
1	15	36	1.4	0.10
2	8	24	2.3	0.20
3	8	48	2.3	0.20
4	6	40	2.2	0.40
5	4	24	2.3	0.20
6	5	30	1	0.10
7	4	24	1	0.10
8	4	24	1	0.10
9	4	24	1	0.10
10	4	9	2	0.10
11	4	24	1	0.10
12	2	24	1	0.10
13	∞	∞	1	0.10

TAB. 6.2 – Exemple 2 : paramètres utilisés.

temps	h_0	h_1	h_2	h_3	h_4	h_5
0.50	1.33	1.12	0.28	0.00	0.00	3.80
1.50	5.31	3.54	2.17	0.00	0.00	7.21
2.50	8.11	5.07	3.41	0.00	0.00	11.53
3.50	10.98	6.72	4.84	0.00	0.00	13.69
4.50	13.19	7.44	6.21	0.00	0.00	15.68
5.50	15.19	8.05	8.19	0.53	0.29	14.79
6.50	13.93	8.88	9.10	1.84	1.68	14.28
7.50	13.26	9.71	9.95	2.55	2.44	12.71
8.50	12.09	9.54	9.73	3.02	2.80	12.55
9.50	11.82	8.91	9.31	2.79	2.42	11.77
10.50	11.02	8.03	8.02	2.63	2.59	10.24
11.50	8.87	7.36	6.35	1.46	2.59	9.60
12.50	6.05	5.66	6.08	1.81	1.24	7.27
13.50	4.75	2.78	4.31	1.62	0.60	4.17
14.50	3.98	0.92	0.00	0.79	0.82	0.00

TAB. 6.3 – Exemple 2 : taux de départ final.

temps	$S[0]$	S_1	S_2	S_3	S_4	S_5
0.50	8.06	8.06	8.06	8.99	9.01	8.06
1.50	10.47	10.47	10.48	11.28	11.29	10.47
2.50	13.68	13.67	13.68	14.30	14.31	13.67
3.50	17.42	17.41	17.43	17.94	17.95	17.43
4.50	21.61	21.60	21.62	21.90	21.91	21.59
5.50	26.23	26.24	26.24	26.25	26.25	26.25
6.50	31.29	31.30	31.29	31.29	31.29	31.28
7.50	36.55	36.54	36.54	36.54	36.55	36.53
8.50	41.87	41.85	41.85	41.86	41.86	41.86
9.50	46.91	46.90	46.90	46.87	46.87	46.91
10.50	51.38	51.41	51.41	51.41	51.40	51.37
11.50	55.44	55.33	55.34	55.42	55.43	55.37
12.50	58.40	58.47	58.46	58.39	58.39	58.48
13.50	60.40	60.46	60.51	60.44	60.39	60.45
14.50	61.06	60.95	60.83	61.00	60.85	60.83

TAB. 6.4 – Exemple 2 : temps de parcours final.

iter	GAP
1	2382.80
2	1160.10
3	609.96
4	440.35
5	405.80
6	374.34
7	349.57
8	167.82
9	153.24
10	131.25
11	84.23
12	58.31
13	37.79
14	22.28
15	15.70
16	9.84
17	6.74
18	5.57
19	3.15
20	1.59

TAB. 6.5 – Exemple 2 : convergence de la méthode de projection.

Conclusion

Dans cette thèse, nous avons étudié et analysé le problème d'équilibre dynamique sur un réseau de transport en utilisant une approche analytique. Nous avons défini une région réalisable Ω , en fonction des flots temporels sur les chemins et de la demande de chaque paire origine-destination. Nous avons également défini l'équilibre dynamique comme une généralisation du principe descriptif de Wardrop, en fonction du temps effectif de parcours des chemins. Cette définition nous assure qu'aucun usager pourrait réduire son temps effectif de voyage.

Le temps de parcours effectif d'un chemin est calculé comme la somme des temps de parcours de tous les arcs qui font partie du chemin au moment où le flot arrive à chacun d'entre eux.

$$S_k(t) = \sum_{a \in k} s_a(v(t + S_{ka}(t))), \quad \forall k \in K_i, i \in I, t \in [0, T]. \quad (6.36)$$

Le temps de parcours effectif est donc une fonction dépendante du volume temporel, la quantité de véhicules qui sont présents sur les arcs à chaque moment. Pour le calculer, nous devons trouver d'abord le volume et le temps de parcours de chaque arc pour toute la période de temps à étudier. Ces quantités sont obtenues par la résolution du problème de chargement dynamique de réseau. Deux algorithmes continus pour résoudre le problème de chargement ont été présentés. La condition nécessaire de FIFO est vérifiée par la contrainte de conservation du flot pour des fonctions linéaires et non linéaires de temps de parcours des arcs. Les algorithmes exposés calculent une solution unique en un nombre fini d'itérations.

Nous avons comparé les deux méthodes avec une version précédente. Les résultats numériques indiquent que les deux nouveaux algorithmes sont plus efficaces et qu'ils peuvent être appliqués pour des réseaux de grande taille.

Les conditions d'équilibre sont équivalentes à la formulation d'un problème d'inégalité variationnelle, PIV. Nous avons présenté une méthode itérative de projection pour résoudre le PIV, basée sur la fonction de Fukushima (1989). Les chemins utilisés sont fixés à l'avance. Nous ne pouvons pas établir une preuve de convergence, en raison de la non monotonie de la fonction de coût. Nous avons résolu des exemples numériques sur des réseaux des villes de Sioux Falls et Hull et nous avons vérifié que les conditions d'équilibre sont atteintes.

Nous avons aussi présenté une méthode de restriction pour trouver une solution globale où les chemins ne sont pas fixés à l'avance, mais plutôt générés au fur et à mesure. Nous avons montré des exemples numériques de résolution du problème d'équilibre dynamique sur des réseaux de taille moyenne. Pour sélectionner les chemins, nous avons choisi ceux qui ont le plus petit coût selon les conditions de trafic trouvées à chaque itération de la méthode de restriction. Pour trouver ces chemins, nous avons implanté une adaptation dynamique de la méthode de Dijkstra pour résoudre le problème du plus court chemin dynamique. Dans les exemples examinés, le nombre de chemins sélectionnés est relativement petit. À chaque itération de l'algorithme de restriction, le problème d'inégalité variationnelle restreint est résolu de façon approximative. À l'itération suivante, la première solution est choisie de façon à être égale à la dernière solution de l'itération précédente. Ainsi, nous profitons de l'information déjà acquise, et la nouvelle solution initiale est *plus proche* de la solution d'équilibre optimale. Nous avons aussi implanté un algorithme de proportions successives qui consiste

à sélectionner de nouveaux chemins et à distribuer le flot demandé en proportions égales entre ces chemins qui relient chaque paire origine-destination. Les résultats obtenus avec cet algorithme ne sont pas convaincants

Nous pouvons conclure que les algorithmes que nous avons développés sont efficaces en pratique pour résoudre certaines instances du problème d'équilibre dynamique. Nous avons utilisé ces algorithmes sur des exemples numériques pour des réseaux de tailles considérables, et nous avons obtenu des résultats très satisfaisants. À notre connaissance, ces résultats sont les meilleurs obtenus jusqu'à présent.

Nous avons aussi présenté une formulation mathématique du problème de chargement de réseau avec capacité limitée sur les arcs. Nous avons établi une formulation pour que la condition FIFO soit vérifiée. Nous avons utilisé les concepts de demande et disponibilité locales. Le nouveau modèle est divisé en deux parties : temps de déplacement et temps d'attente. Nous avons modélisé une file d'attente correspondant à chaque arc. Nous avons également utilisé ce modèle pour formuler un algorithme de résolution du problème d'équilibre dynamique avec capacité finie sur les arcs. Nous avons résolu des exemples numériques sur de petits réseaux fictifs. Ces résultats ouvrent la voie à des recherches futures afin de développer des algorithmes plus efficaces et performants pour de grands réseaux.

Appendice

Notation

Nous présentons dans cette section la notation que nous utilisons dans ce document :

N : l'ensemble de noeuds ;

A : l'ensemble d'arcs ;

a^+ : l'ensemble d'arcs successeurs de a

a^- : l'ensemble d'arcs prédécesseurs de a

n^+ : ensemble des arcs qui sortent du noeud n , arcs avec queue égale à n ;

n^- : ensemble des arcs qui entrent dans le noeud n , arcs avec tête égale à n ;

I : l'ensemble de toutes les paires (OD) ;

k : indice d'un chemin ;

K_i : l'ensemble de chemins pour la paire $(OD)_i$;

K : l'ensemble de tous les chemins, $K = \cup_{i \in I} K_i$;

K^a : l'ensemble de tous les chemins qui traversent l'arc a ;

ka^- : l'arc qui précède l'arc a sur le chemin k ;

T : dernier temps de départ ;

T' : dernier temps d'arrivée (temps système) ;

T_{ka} : temps d'arrivée à la queue de l'arc a du dernier flot correspondant au chemin k

T_a : temps d'arrivée à la queue de l'arc a du dernier flot $T_a = \max_{k \in K^a} \{T_{ka}\}$;

$g(t)$: taux de départ demandé pour les paires (OD) au moment $t \in [0, T]$.

$g_i(t)$: taux de départ demandé pour la paire $(OD)_i$ au moment $t \in [0, T]$.

$h(t)$: taux de départ sur les chemins au moment $t \in [0, T]$;

$h_k(t)$: taux de départ sur le chemin k au moment $t \in [0, T]$;

$H_k(t)$: quantité des flots accumulés qui ont emprunté le chemin k entre 0 et l'instant $t \in [0, T]$;

$b_{ka}(t)$: taux d'entrée du flot dans l'arc a correspondant au chemin k au temps $t \in [0, T']$;

$B_{ka}(t)$: quantité des flots accumulés qui sont entrés dans l'arc a par le chemin k au temps $t \in [0, T']$;

$e_{ka}(t)$: taux de sortie du flot de l'arc a correspondant au chemin k à l'instant $t \in [0, T']$;

$E_{ka}(t)$: quantité des flots accumulés qui sont sortis de l'arc a correspondant au chemin k au temps $t \in [0, T']$;

$b_a(t)$: taux d'entrée du flot dans l'arc a au temps $t \in [0, T']$. $b_a(t) = \sum_{k \in K^a} b_{ka}(t)$;

$B_a(t)$: quantité des flots accumulés qui sont entrés dans l'arc a au temps $t \in [0, T']$;

$B_a(t) = \sum_{k \in K^a} B_{ka}(t)$;

$e_a(t)$: taux de sortie du flot de l'arc a au temps $t \in [0, T']$. $e_a(t) = \sum_{k \in K^a} e_{ka}(t)$;

$E_a(t)$: quantité des flots accumulés qui sont sortis de l'arc a au temps $t \in [0, T']$; $E_a(t) = \sum_{k \in K^a} E_{ka}(t)$;

$v_a(t)$: volume du flot sur l'arc a au moment $t \in [0, T']$;

$s_a(v)$: fonction de coût de l'arc a ;

$\tau_a(t)$: moment de sortie du flot qui entre dans l'arc a au moment $t \in [0, T']$;

$\tau_a^{-1}(t)$: fonction inverse de τ_a $t \in [0, T']$;

$\Phi_k(t)$: temps instantané de parcours sur le chemin k des usagers qui partent au moment $t \in [0, T]$;

$S_k(t)$: temps effectif de parcours sur le chemin k des usagers qui partent au moment

$t \in [0, T]$;

$S_{ka}(t)$: temps effectif de parcours sur le chemin k de l'origine jusqu'à la fin de l'arc a des usagers qui partent à l'instant $t \in [0, T]$;

$\Pi_{ka}(t)$: moment de sortie de l'arc a du flot qui entre au chemin k au temps $t \in [0, T]$;

$\Pi_{ka}^{-1}(t)$: moment de sortie de l'arc a du flot qui entre au chemin k au temps $t \in [0, T]$;

$\nu_i(t)$: le temps instantané de voyage minimal des chemins qui relient la paire $(OD)_i$, au moment $t \in [0, T]$;

$\mu_i(t)$: le temps effectif de voyage minimal des chemins qui relient la paire $(OD)_i$, au moment $t \in [0, T]$;

Bibliographie

- [1] Adamo, V., Astarita, V., Florian, M., Mahut, M. et Wu, J.H., *Modelling the spill-back of congestion in link based dynamic network loading models : a simulation model with application*, Proceedings of 14th. International Symposium on Transportation and Traffic Theory, (publié par Elsevier Science Ltd.) 555-573 (1999).
- [2] Ahn, B.H. et Shin, J.Y., *Vehicle Routeing with Time Windows and Time-varying Congestion*, Journal of the operational Research Society 42 393-400 (1991).
- [3] Astarita, V., *A Continuous Time Link Model For Dynamic Network Loading Based on Travel Time Function*, Proceedings 13th. International Symposium on Transportation and Traffic Theory, Lyon, France, (publié par Elsevier Science Ltd.) 77-102 (1996).
- [4] Bazaraa, M.S., Sherali, H.D. et Shetty, C.M., *Nonlinear programming, theory and algorithms*, John Wiley & Sons, Inc. (1993).
- [5] Ben-Akiva, M. et De Palma, A., *Some Circumstances in Which Vehicles Will Reach Their Destinations Earlier by Starting Later : Revisited*, Transportation Science 20 52-55 (1986).
- [6] Bernstein, D., Friesz, T.L., Tobin, R.L. et Wie, B.W., *A Variational Control Formulation of the Simultaneous Route and Departure-Time Choice Equilibrium Problem*, Transportation and Traffic Theory, C.F. Daganzo (Editor) 107-126 (1993).

- [7] Bertsekas, D.P., *Linear Network Optimization*, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts 194-197 (1991).
- [8] Bertsekas, D.P. et Gafni, E., *Projection methods for variational inequalities with application to the traffic assignment*, Mathematical Programming Study 17 139-159 (1982).
- [9] Boyce, D.E., Ran, B. et LeBlanc, L., *Solving an Instantaneous Dynamic User-Optimal Route Model*, Transportation Science 29 128-142 (1995).
- [10] Boyce, D.E., Lee, D.H. et Janson, B.N., *A Variational Inequality Model of an Ideal Dynamic User-Optimal Route Choice Problem*, EURO Working Group in Transportations and Urban Traffic, Newcastle (England) (1996).
- [11] Carey, M., *A Constraint Qualification for a Dynamic Traffic Assignment Model*, Transportation Science 20 55-58 (1986).
- [12] Carey, M., *Optimal time-varying flows on congested networks*, Operations Research 35 58-69 (1987).
- [13] Carey, M., *Nonconvexity of the dynamic traffic assignment problem*, Transportation Research B 26 127-133 (1992).
- [14] Cascetta, E. et Cantarella, G.E., *A day-to-day and within-day dynamic stochastic assignment model*, Transportation Research A 25 277-291 (1991).
- [15] Chabini, I., *Discrete dynamic shortest path problems in transportation applications : complexity and algorithms with optimal run time*, accepté pour publication dans Transportation Research Records.
- [16] Chen, H.K. et Smith, C.F., *A model and an algorithm for the dynamic user-optimal route choice problem*, Transportation Research B 32 219-234 (1998).

- [17] Codina, E. et Barceló, *Dynamic traffic assignment : Considerations on some deterministic modelling approaches*, Annals of Operations Research 60 1-58 (1995).
- [18] Dafermos, S., *Traffic equilibrium and variational inequalities*, Transportation Science 14 42-54 (1980).
- [19] Daganzo, C.F., *The cell transmission model : a dynamic representation of highway traffic consistent with the hydrodynamic theory*, Transportation Research B 28 269-287 (1994).
- [20] Daganzo, C.F., *The cell transmission model : network traffic*, Transportation Research B 29 79-93 (1995a).
- [21] Daganzo, C.F., *Properties of link travel time functions under dynamic loads*, Transportation Research B 29B 95-98 (1995b).
- [22] De Palma, A., Ben-Akiva, M., Lefèvre, C. et Litinas, N., *Stochastic Equilibrium Model of Peak Period Traffic Congestion*, Transportation Science 17 430-453 (1983).
- [23] Dijkstra, E., *A note on two problems in connexion with graphs*, Numerische Mathematik 1 269-271 (1959).
- [24] Drissi-Kaïtouni, O., *A model for the dynamic traffic assignment problem*, Publication CTR-702, Centre de recherche sur les transports, Université de Montréal, (1990a).
- [25] Drissi-Kaïtouni, O., *A variational inequality formulation of the dynamic traffic assignment*, Publication CTR-735, Centre de recherche sur les transports, Université de Montréal, (1990b).
- [26] Drissi-Kaïtouni, O. et Hamed-Benchekroun, A., *A Dynamic Traffic Assignment Model and a Solution Algorithm*, Transportation Science 26 119-128 (1992).

- [27] Drissi-Kaitouni, O. et Gendreau, M., *A New Dynamic Traffic Assignment Model*, Publication CTR-854, Centre de recherche sur les transports, Université de Montréal (1992).
- [28] Fernandez, J.E. et de Cea, J., *Flow propagation description in dynamic network assignment models*, Proceeding of Triennial Symposium on Transportation Analysis, Capri, Italy 517-532 (1994).
- [29] Florian, M., *An Introduction to Network Models Used in Transportation Planning*, Publication 262, Centre de recherche sur les transports, Université de Montréal, (1982a).
- [30] Florian, M. et Spiess, H., *The convergence of diagonalization algorithms for asymmetric network equilibrium problems*, Transportation Research B 16 477-483 (1982b).
- [31] Florian, M, Guélat, J. et Spiess, H., *An efficient Implementation of the "Partan" variant of the Linear Approximation Method for the Network Equilibrium Problem*, Networks 17 319-339 (1987).
- [32] Ford, L.R. et Fulkerson, D.R., *Flows in Networks*, Princeton University Press, Princeton NY. (1962).
- [33] Frank, M. et Wolfe, P., *An algorithm for quadratic programming*, Naval Research Logistic Quarterly 3 (1956) 95-110.
- [34] Friesz, T.L., Luque, J., Tobin, R. et Wie, B.W., *Dynamic Network Traffic Assignment Considered as a Continuous Time Optimal Control Problem*, Operations Research 37 893-901 (1989).
- [35] Friesz, T.L., Bernstein, D., Smith, T.E., Tobin, R.L. et Wie, B.W., *A Variational Inequality Formulation on the Dynamic Network User Equilibrium Problem*, Operations Research 41 179-191 (1993).

- [36] Friesz, T.L., Bernstein, D., Mehta, N.J., Tobin, R.L. et Ganjalizadeh, S., *Day-to-day dynamic network disequilibria and idealized traveler information systems*, Operations Research 42 1120-1136 (1994).
- [37] Friesz, T.L., Bernstein, D. et Stough, R., *Dynamic Systems, Variational Inequalities and Control Theoretic Models for Predicting Time-Varying Urban Network Flows*, Transportation Science 30 14-31 (1996).
- [38] Fukushima, M., *Equivalent differentiable optimization problems and descent methods for asymmetric variational inequality problems*, Technical Report 89007, Dept. of Applied Math. and Physics, Kyoto University (1989).
- [39] Fulkerson, D.R., *Studies in graph theory, Part I*, Studies in Mathematics 11 (1975).
- [40] Guélat, J., *Algorithmes pour le problème d'affectation du trafic d'équilibre avec demandes fixes*, Publication 299, Centre de recherche sur les transports, Université de Montréal (1983).
- [41] Gendreau, M. et Bouzaiene-Ayari, B., *A discrete-time macroscopic dynamic traffic assignment model*, Publication 96-31, Centre de recherche sur les transports, Université de Montréal (1996).
- [42] Hearn, D.W., *The gap function of a convex program*, Operations Research Letters 1 67-71 (1982).
- [43] Heydecker, B.G. et Addison, J.D., *Analysis of Traffic Models for Dynamic Equilibrium Traffic Assignment*, 13th. International Symposium on Transportation and Traffic Theory INRETS, Lyon, France (1996).
- [44] Ho, J.K., *A Successive Linear Optimization Approach to the Dynamic Traffic Assignment Problem*, Transportation Science 14 295-305 (1980).

- [45] Janson, B.N., *Dynamic traffic assignment for urban road networks*, Transportation Research B 25 143-161 (1991).
- [46] Jayakrishnan, R., Mahmassani, H.S. et Hu, T.Y., *An evaluation tool for advanced traffic information and management systems in urban networks*, Transportation Research C 3 129-147 (1994).
- [47] Kaufman, D.E. et Smith, R.L., *Fastest Paths in Time-Dependent Networks for Intelligent Vehicle-Highway Systems Application*, IVHS Journal 1 1-11 (1993)
- [48] Kaufman, D.E., Nonis, J., et Smith, R.L., *A mixed integer linear programming model for dynamic route guidance*, Transportation Research B 32 431-440 (1998).
- [49] Kennington, J.L., *Algorithms for network programming*, John Wiley and Sons, (1980).
- [50] Lawphongpanich, S. et Hearn, D.W., *Simplicial decomposition of asymmetric traffic assignment problem*, Transportation Research B 18 123-133 (1984).
- [51] Leblanc, L.J., Morlok, E.K. et Pierskalla, W.P., *An efficient approach to solving the road network equilibrium traffic assignment problem*, Transportation Research 9 309-318 (1975).
- [52] Lighthill, M.J. et Whitham, J.B., *On Kinematic waves I : Flow movement in long rivers ; II : A theory of traffic flow on long crowded roads*, Proc Royal Soc. A 229 281-345 (1955).
- [53] Mahmassani, H.S. et Herman, R., *Dynamic User Equilibrium Departure Time and Routes Choice on Idealized Traffic Arterials*, Transportation Science 18 362-384 (1984).
- [54] Mahmassani, H.S. et Chang, G.L., *On Boundedly Rational User Equilibrium in Transportation Systems*, Transportation Science 21 89-99 (1987).
- [55] Marcotte, P., *A new algorithm for solving variational inequalities with application to the traffic assignment problem*, Mathematical Programming 33 339-351 (1985).

- [56] Marcotte, P. et Guélat, J., *Adaptation of a modified Newton method for solving the asymmetric traffic equilibrium problem*, Transportation Science 22 112-124 (1988).
- [57] Marcotte, P. et Dussault, J.P., *A sequential linear programming algorithm for solving monotone variational inequalities*, SIAM Journal of Control and Optimization 27 1260-1278 (1989).
- [58] Merchant, D. et Nemhauser, G., *A Model and an Algorithm for the Dynamic Traffic Assignment Problems*, Transportation Science 12 183-199 (1978a).
- [59] Merchant, D. et Nemhauser, G., *Optimality Conditions for a Dynamic Traffic Assignment Model*, Transportation Science 12 200-207 (1978b).
- [60] Nagurney, A. et Zhang, D., *Projected dynamical systems in the formulation, stability analysis, and computation of fixed-demand traffic network equilibria*, Transportation Science 31 147-158 (1997).
- [61] Newell, G.F., *A simplified theory of kinematic waves in highway traffic*, Transportation Research B 27 281-313 (1993).
- [62] Nguyen, S., *A mathematical programming approach to equilibrium methods of traffic assignment*, Publication 138, Département d'Informatique et de Recherche Operationnelle, Université de Montréal (1975).
- [63] Nguyen, S. et Dupuis, C., *An efficient method for computing traffic equilibria in networks with asymmetric transportation costs*, Transportation Science 18 185-202 (1984).
- [64] Pape, U., *Implementation and Efficiency of Moore-algorithms for the shortest route problem*, Mathematical Programming 7 212-222 (1974).
- [65] Patriksson, M., *Nonlinear programming and variational inequality problems, A unified approach*, Kluwer Academic Publishers (1999).

- [66] Peeta, S. et Mahmassani, H., *System optimal and user equilibrium time-dependent traffic assignment in congested networks*, Annals of Operations Research 60 81-113 (1995).
- [67] Ran, B., Boyce, D.E. et LeBlanc, L., *A New Class of Instantaneous Dynamic User-Optimal Traffic Assignment Models*, Operation Research 41 192-202 (1993).
- [68] Ran, B. et Boyce, D.E., *Dynamic Urban Transportation Network Models*, Springer-Verlag (1994).
- [69] Ran, B., Hall, R.W. et Boyce, D.E., *A link-based variational inequality model for dynamic departure time/route choice*, Transportation Research B 30 31-46 (1996).
- [70] Richards, P.I., *Shockwaves on the highway*, Operations Research 4 42-51 (1956).
- [71] Rundin, W., *Principles of mathematical analysis*, McGraw-Hill (1976).
- [72] Newell, G.F., *A simplified theory of kinematic waves in highway traffic*, Transportation Research B 27 281-313 (1993).
- [73] Smith, M.J., *The existence, uniqueness and stability of traffic equilibria*, Transportation Research B 13 295-304 (1979).
- [74] Smith, M.J. et Ghali, M., *The dynamics of traffic assignment and traffic control : a theoretical study* Transportation Research B 24 409-422 (1990).
- [75] Smith, M.J., *A new dynamic traffic model and the existence and calculation of dynamic user equilibria on congested capacity-constrained road networks*, Transportation Research B 27 49-63 (1993).
- [76] Smith, M.J. et Wisten, M.B., *A continuous day-to-day traffic assignment model and the existence of a continuous dynamic user equilibrium*, Annals of Operations Research 60 59-79 (1995).

- [77] Suwansirikul, C., T.L. Friesz et R. Tobin, *Equilibrium Decomposed Optimization : A Heuristic for the Continuous Equilibrium Network Design Problem*, Transportation Science 21 254-263 (1987).
- [78] Taylor, N.B., *CONTRAM5 : An enhanced traffic assignment model*, Transportation Research Laboratory, Crowthorne (1990).
- [79] Wardrop, J.G., *Some theoretical aspects of road traffic research*, Proceedings of the Institute of Civil Engineers, Part II 325-378 (1952).
- [80] Wie, B.W., Friesz, T.L. et Tobin, R.L., *Dynamic user optimal traffic assignment on congested multideestination networks*, Transportation Research B 24 431-442 (1990).
- [81] Wie, B.W., Tobin, R.L., Friesz, T.L. et Bernstein, D., *A Discrete Time, Nested Cost Operator Approach to the Dynamic Network User Equilibrium Problem*, Transportation Science 29 79-92 (1995).
- [82] J.H. Wu, *A Study of Monotone Variational Inequalities and their application to network equilibrium problems*, PH.D. dissertation, Publication 801, Centre de recherche sur les transports, Université de Montréal (1991).
- [83] Wu, J.H., *On descent simplicial decomposition methods for the monotone variational inequality problem with its application to the network equilibrium problem*, Publication 806, Centre de recherche sur les transports, Université de Montréal (1992).
- [84] Wu, J.H., Florian et M., Marcotte, P., *A general descent framework for the monotone variational inequality problem*, Mathematical Programming 61 281-300 (1993).
- [85] Wu, J.H., *Dynamic network equilibrium problem formulated as an infinite dimensional variational inequality problem*, Proceeding of Triennial Symposium on Transportation Analysis, Capri, Italy 497-513 (1994).

- [86] Wu, J.H., Chen, Y., Florian, M. et Xu, Y.W., *A projection algorithm for the dynamic network equilibrium problem*, Publication CTR-95-64, Centre de recherche sur les transports, Université de Montréal (1995).
- [87] Wu, J.H., Chen, Y. et Florian, M., *The continuous dynamic network loading problem : a mathematical formulation and solution method*, Transportation Research B 32 173-187 (1998a).
- [88] Wu, J.H., Florian, M., Xu, Y.W. et Rubio-Ardanaz, J.M., *A projection algorithm for the dynamic network equilibrium problem*, Proceedings of 1998 International Conference on Traffic and Transportation Studies, Beijing, publié par American Society of Civil Engineers (1998b), pp. 379-390.
- [89] Wu, J.H., Florian, M. et Rubio-Ardanaz, J.M., *The Continuous Dynamic Network Loading Problem : Recent Computational Results*, Présenté a Triennial Symposium on Transportation Analysis, TRISTAN III, Vol. 2, San Juan, Puerto Rico (1998c).
- [90] Xu, Y.W., Wu, J.H. et Florian, M., *On Dynamic Traffic Equilibrium Problems*, à publier, (1997).
- [91] Xu, Y.W., Wu, J.H. et Florian, M., *An efficient algorithm for the continuous network loading problem : A DYNALOAD implementation*, Transportation Networks : Recent Methodological Advances, Elsevier Science Ltd. 51-66 (1998).
- [92] Xu, Y.W., Wu, J.H., Florian, M., Zhu, D.L. et Marcotte, P., *New advances in the continuous dynamic network loading problem*, Transportation Science 33 341-353 (1999).
- [93] Zawack, D. et Thompson, G., *A Dynamic Space-Time Network Flow Model for City Traffic Congestion*, Transportation Science 21 153-162 (1987).

- [94] Zhang, D. et Nagurney, A., *On the local and global stability of a travel route choice adjustment process*, Transportation Research B 30 245-262 (1996).
- [95] Zhang, D. et Nagurney, A., *Formulation, stability, and computation of traffic network equilibria as projected dynamical systems*, Journal of Optimization Theory and Applications 93 417-444 (1997).
- [96] Zhang, D., Nagurney, A. et Wu, J.H., *On the equivalence between stationary link flow patterns and traffic network equilibria*, Publication CTR-98-35, Centre de recherche sur les transports, Université de Montréal (1998).
- [97] Zhu, D. et Marcotte, P., *On the existence of solutions to the dynamic user equilibrium problem*, Publication CTR-97-42, Centre de recherche sur les transports, Université de Montréal (1997).
- [98] Ziliaskopoulos, A.K. et Lee, D.H., *A cell transmission based assignment-simulation model for integrated freeway/surface street systems*, à paraître dans in the Transportation Research Record (1997).

