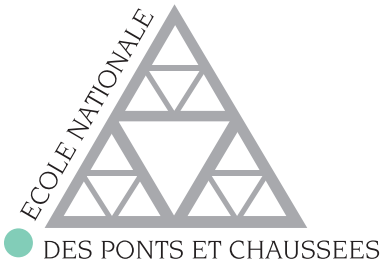


# THÈSE

présentée pour l'obtention du titre de

Docteur de l'École Nationale des Ponts et Chaussées

**Spécialité : Mathématiques, Informatique**

par

**Eugénie LIORIS**

# Évaluation et optimisation de systèmes de taxis collectifs en simulation

Soutenue le 17 décembre 2010
devant le jury composé de :

| | | |
|---|---|---|
| Rapporteurs : | André De PALMA | ENS Cachan |
| | Jean-Pierre QUADRAT | INRIA Rocquencourt |
| Examinateurs : | Richard DARBÉRA | École des Ponts-ParisTech |
| | Arnaud de LA FORTELLE | École des Mines-ParisTech |
| | Frédéric MEUNIER | École des Ponts-ParisTech |
| | Michel PARENT | INRIA Rocquencourt |
| | Felisa VÁZQUEZ-ABAD | City University of New-York |
| Directeur de thèse : | Guy COHEN | École des Ponts-ParisTech |

Cette thèse a été effectuée conjointement au CERMICS de l'École des Ponts-ParisTech et dans l'équipe IMARA de l'INRIA-Rocquencourt.

*A Guy Cohen*

*To see with one's own eyes, to feel and judge without succumbing to the suggestive power of the fashion of the day, to be able to express what one has seen and felt in a trim sentence or even a cunningly wrought word - is that not glorious ? Is it not a proper subject for congratulation ?*

*Regarder avec ses propres yeux, sentir et juger sans succomber au pouvoir suggestif de la mode du jour, être capable d'exprimer ce qu'on a vu ou ressenti en une phrase concise ou même en un mot habilement choisi — n'est-ce pas splendide ? N'est-ce pas un vrai motif de compliment ?*

*Albert Einstein, 1934*

# Remerciements

Voilà que le temps est arrivé de vous présenter cette étude sur les taxis collectifs, laquelle m'a donné l'opportunité et le grand honneur de pouvoir travailler avec Monsieur Guy Cohen et partager autant que possible son énorme richesse de connaissances, sa grande expérience, et son savoir-faire. Monsieur Cohen m'a montré une autre façon de réfléchir et d'appréhender chaque chose en me permettant de me transformer intégralement dans ma vie professionnelle et personnelle. C'est un des plus grands cadeaux que la vie ait pu me faire. Mon respect et admiration pour cette personne n'ont fait qu'augmenter. Ceux parmi vous qui ont eu la chance de le connaître peuvent me comprendre ; quant aux autres, vous n'avez que la sincérité de ma parole.

Je tiens à remercier particulièrement le Directeur de l'École Nationale des Ponts et Chaussées, Monsieur Philippe Courtier, d'avoir accepté ma candidature à l'École.

L'accueil du CERMICS de l'École des Ponts, le soutien de Messieurs Serge Piperno, Jean-François Delmas, Madame Catherine Baccaert ainsi que de tout le secrétariat ne peuvent que recevoir toute ma gratitude. Merci aussi à Alice Tran pour sa diligence et son assistance dans toutes les opérations administratives et à Jacques Daniel pour sa bienveillance et son aide dans tous les problèmes informatiques.

La participation de Messieurs Pierre Carpentier et Jean-Philippe Chancelier a été très précieuse par leur offre généreuse de conseils scientifiques... mais pas seulement.

Le rôle de mon co-directeur, Monsieur Arnaud de La Fortelle, a été tout-à-fait *décisif* pour la réalisation de ce travail. Je lui en suis extrêmement reconnaissante, les mots étant très pauvres pour pouvoir exprimer mes sentiments. Je me souviendrai de son soutien tout au long de ma vie.

En m'accueillant dans son équipe, il m'a donné l'occasion de faire la connaissance du directeur du projet, Monsieur Michel Parent chez qui j'ai admiré dès la première rencontre l'intelligence, la simplicité et le grand humour. J'ai pu observer de près ses ambitions révolutionnaires et sa façon de les réaliser, et j'ai suivi son exemple pour mettre en place mon projet qui est considéré encore de nos jours comme très futuriste...

Je dois remercier toute son équipe à l'INRIA, Monsieur Fawzi Nashashibi, Jean-Marc Lasgouttes dont j'ai apprécié la sincérité, Monsieur Armand Yvet qui a toujours fait de son mieux pour satisfaire mes souhaits, Madame Chantal Chazelas, les ressources humaines, le centre de documentation, Monsieur Gérard Finet pour son soutien pendant des moments très difficiles, le service informatique MYRIAD qui m'a offert les meilleures conditions possibles pour réaliser ce travail, sans oublier l'amitié de Messieurs Yves Sorel et Daniel de Rauglaudre.

Ce fut un grand honneur pour moi de pouvoir discuter tout au long de ce travail avec Monsieur Jean-Pierre Quadrat lequel a pris tout le temps qu'il fallait pour dialoguer avec moi chaque fois que je me suis m'adressée à lui. Je remercie également les deux rapporteurs, lui-même ainsi que Monsieur André de Palma, d'avoir bien voulu examiner ce travail en détail.

Je remercie particulièrement Frédéric Meunier qui a toujours montré une grande disponibilité pour offrir sa collaboration et nous aider de ses précieuses connaissances.

En plus des membres du jury déjà cités, je souhaite exprimer ma gratitude envers Madame Felisa Vázquez-Abad et Monsieur Richard Darbéra pour leur intérêt pour ce projet et pour avoir accepté de participer à ce jury.

À ce stade, je souhaite remercier l'INRIA et l'École des Ponts de leur soutien dans des moments très délicats que j'ai du traverser un peu avant la fin de mon travail en les assurant de mon profond regret pour les soucis que j'ai pu involontairement créer.

Je souhaite par ailleurs exprimer ma reconnaissance à ma famille, en particulier mes parents, mon oncle Jean et ma tante Lilo (qui n'est plus parmi nous), lesquels dès mon premier âge m'ont éduquée pour m'investir dans les vraies valeurs de la vie. Leur passion pour l'apprentissage et

le courage avec lequel ils se battent pour atteindre leurs buts a été le meilleur encouragement pour moi.

Pendant que je traversais des moments très difficiles, j'ai eu la chance de connaître Madame Claudette Wallerand qui a pu remettre le sourire dans ma vie. Les moments que j'ai passés avec elle ont été un vrai bonheur et je suis émerveillée par ses connaissances et sa joie de vivre.

Je remercie aussi mes amis (en respectant leur souhait de rester discrète) qui me permettent de me concentrer sur mon travail en me prouvant la vraie valeur de l'amitié.

J'ai pris un énorme plaisir en faisant ce travail et je l'ai réalisé avec beaucoup d'amour et de passion, pas seulement pour obtenir un diplôme supplémentaire, mais parce que j'ai cru à ce projet réalisé sous la direction de Monsieur Cohen. Pendant toute sa durée, de multiples aléas difficiles à gérer sont advenus, mais j'ai été si bien entourée qu'aucun de ces obstacles n'a pu mettre en question la réalisation du projet. Aujourd'hui, si on me demandait de tout recommencer, sans aucune hésitation, la réponse serait positive. J'espère que le résultat sera au niveau des espérances et des efforts de tous les gens qui ont contribué à cette entreprise, et des attentes de Monsieur Guy Cohen ainsi qu'aux vôtres. Je vous en laisse juge...

Paris, le 15 novembre 2010
J.L.

# Résumé

Le développement économique d'une région urbaine est étroitement lié à son accessibilité. Le rôle des taxis comme moyen de transport est reconnu mondialement comme le mode offrant la meilleure qualité de service à ses usagers dans des conditions normales ou d'urgence. Malheureusement c'est un moyen très coûteux qui ne peut pas être utilisé quotidiennement par toute la population. Pour abaisser les coûts, il faudrait faire partager le service par plusieurs utilisateurs tout en préservant ses qualités essentielles (trajet presque direct, service porte à porte) en accroissant en même temps la productivité de ces véhicules devenus "collectifs". Cette idée n'est pas nouvelle : P.H. Fargier et G. Cohen l'avaient déjà étudiée en 1971, même si elle peut encore aujourd'hui être considérée comme révolutionnaire et un peu prématurée pour un marché strictement réglementé. Avec une révision de la réglementation, cette extension du service des taxis, si on lui donnait l'opportunité de se mettre en place, pourrait permettre aux taxis de prendre leur part du transport public en s'adressant à la majorité de la population et pas seulement à une minorité de privilégiés pouvant assumer le prix d'un transfert individuel.

Le **chapitre I** de ce mémoire présente une revue des "transports à la demande" dans le monde, des modèles développés à ce sujet, et des algorithmes en rapport avec ces problèmes. Il développe ensuite le projet de "taxis collectifs" étudié dans ce travail et se termine par un aperçu de l'ensemble du mémoire.

Le **chapitre II** évoque les différents problèmes et questions qu'il faut aborder pour l'étude d'un système de taxis collectifs en ville, argumente sur la nécessité de construire un simulateur, décrit globalement sa structure avec une partie "mécanique" et une partie "algorithmique", les choix informatiques, les différents modes d'exploitation de ce simulateur, les données d'entrée requises et les sorties brutes.

Le **chapitre III** développe la gestion décentralisée correspondant à un système qui fonctionne sans réservation préalable auprès d'un dispatching central : les clients rencontrent les taxis directement au bord du trottoir. Ce chapitre décrit en détail le modèle de "simulation par événements" adapté à ce mode de gestion, puis les problèmes de décision qui concernent la gestion d'un tel système et les algorithmes utilisés pour résoudre ces problèmes.

Le **chapitre IV** rend compte d'une longue campagne de simulations numériques sur un cas d'étude fictif mais réaliste. Il illustre d'abord comment une simulation peut être analysée en détail et il montre les nombreuses sorties qui peuvent être produites à partir des sorties brutes du simulateur pour évaluer la qualité de service offerte et les coûts de fonctionnement associés. Il se tourne ensuite vers l'exploitation de séries de simulations où on fait varier certains paramètres relatifs à la gestion temps réel et au dimensionnement du système afin de réaliser les meilleurs compromis possibles entre plusieurs critères contradictoires. À l'aide de cette démarche, il montre aussi l'influence sur les résultats de données exogènes comme la demande (intensité et géométrie).

Le **chapitre V** traite de la gestion à partir d'un dispatching central, les clients appelant ce dispatching pour réserver un trajet. La gestion mixte est la superposition dans un même système des deux modes de gestion, centralisée et décentralisée. Le simulateur existant est réputé capable, pour sa partie "mécanique" qui gère et enchaîne les événements, de s'accommoder de tous les modes de gestion. Par contre, nous avons manqué de temps pour travailler suffisamment sur la partie algorithmique (la gestion temps réel du cas centralisé est bien plus difficile que le cas décentralisé) et il n'y a donc pas non plus d'expériences numériques relatives à cette situation.

Le chapitre se contente donc de décrire le modèle par événements pour les gestions centralisée et mixte et évoque la question algorithmique jusqu'à un certain point.

Le **chapitre VI** présente quelques conclusions de ce travail.

# Abstract

The economical development of an urban region depends greatly on ease of access. The taxi is recognised throughout the world as a convenient means of transport providing a high quality personal service for both normal and emergency purposes. Despite the advantages however, the taxi is also too expensive for most people to use routinely for their daily transport. If a taxi and hence the expense could be shared between passengers, there would be the potential to preserve almost the original standard of service whilst increasing vehicle productivity by becoming "collective". This is not a new concept, it was studied in 1971 by P.H. Fargier and G. Cohen who suggested and studied such a system. Despite this, even today it may be regarded as a somewhat premature and revolutionary idea for a strictly regulated market. Reviewing the form of operation applied to this mode of transport could allow a collective taxi system to form an acceptable part of the public transport system by offering this service to the majority of the population without being limited to the privileged on the basis of cost.

**Chapter I** of this document presents a review of existing "Transport on Demand" systems at various locations in the world, describing the various models developed on this subject as well as the corresponding algorithms treating the problems encountered. This is followed by an introduction into the project of "Collective Taxis" studied in this work, and finally we give an overview of the rest of this document.

**Chapter II** brings up the multiple problems and questions that have to be dealt with during the study of a collective taxi system covering an entire urban area. At this point, the need to construct a simulator is justified and its architecture is explained : its design is comprised of separate mechanical and decision-making components connected to each other. The technical choices concerning the chosen programming language and software are discussed together with the different exploitation modes of the simulator, the input data and the information produced by the simulation runs.

**Chapter III** provides an insight into the decentralised approach consisting of a system based entirely on casual business obtained from clients at the roadside with no facility for advanced booking available. This chapter extensively describes the modelling of a discrete event simulator specifically constructed for this management approach followed by the decision making issues concerning this approach and the algorithms developed to provide solutions and allow optimised control of the system

**Chapter IV** is a thorough debrief of a long campaign of numerical simulations covering a fictitious but realistic example. The principal idea of this study is to provide a firm methodology from which one should be able to evaluate the applied strategy and measure the system performance.

Initially, methods of analysing a single simulation are being illustrated whilst showing the various outcomes and conclusions that can be produced by dealing with the simulation results. Thus the quality of service provided by the given policy can be judged and quantified as well as the associated functioning costs.

Further on, a series of simulations, where the value of multiple parameters vary, are being explored with the aim of realising all possible compromises between contradictory criteria for optimising real-time management and system dimensioning. This process also points out the

influence of the exogenous data like the demand (geometry and intensity) to the performance of the system.

**Chapter V** handles system management from the central dispatching point of view, where prospective clients call a control centre to make a pre-booking (this is called the centralised mode of operation). A mixed mode can also be considered by superimposing the centralised mode onto the decentralised one.

The mechanical part of the present simulator, which is responsible for processing the system events, is designed to enable the management of the three possible approaches (decentralised, centralised and mixed). Nevertheless the decisional part of the mixed and centralised modes has not been completely developed because of the lack of time (the real time management of the centralised approach is much more complex than the decentralised one).

Consequently numerical results are not presented concerning the centralised and mixed modes. For the moment, this chapter is making do with the description of the model and events of the centralised and mixed scenarios, evoking the algorithmic part up to a certain point.

**Chapter VI** develops some conclusions regarding this work and future possible developments.

# Table des matières

# Table des figures

CHAPITRE I

# Introduction

*L'imagination est la meilleure compagnie de transport au monde.*

*Roger Fournier*

## I.1. Arguments pour un système de taxis collectifs en ville

Les effets de la spectaculaire augmentation de l'urbanisation mondiale pendant le dernier siècle a transformé les villes en centres de richesse économique et de services administratifs. À la recherche des opportunités économiques et afin d'améliorer leur qualité de vie, une grande partie de la population s'y déplace quotidiennement. Par conséquent la productivité urbaine est étroitement liée à l'efficacité de moyens de transport aussi bien de personnes que de marchandises d'un endroit à l'autre. Ainsi avec le temps, cette expansion urbaine aboutit à la nécessité de faire face à un accroissement du volume de personnes et de biens transportés mais aussi à une augmentation de la complexité des problèmes associés.

Les services de transport proposés sont parfois inadaptés et nous sommes très familiers des phénomènes comme :

- les embouteillages,
- des transports publics retardés,
- des places de parking insuffisantes,
- une progressive diminution de l'espace public attribué aux véhicules particuliers, etc.

Afin de faire face à la congestion, la solution la plus répandue est la création de nouvelles structures routières ce qui résout le problème localement dans le temps, puisque cette mesure ne fait qu'encourager les automobilistes à circuler plus souvent et éventuellement suscite l'achat de véhicules supplémentaires. Par conséquent bientôt on se retrouve dans un même contexte où on a besoin de plus d'espace routier, de nouvelles places de stationnement, de l'énergie supplémentaire etc. Ces problèmes ont des conséquences aussi sur l'environnement avec une efficience de l'énergie réduite associée à une augmentation de la pollution (dont on voit les conséquences en termes de changements climatiques). D'autre part, l'accroissement du trafic a inévitablement conduit à un accroissement des accidents routiers.

Le plus souvent, et surtout pour des raisons de confort, de commodité et d'indépendance, les gens préfèrent se déplacer en véhicule personnel, ce qui induit un déséquilibre entre le transport public et les véhicules privés. Cependant, on peut mentionner un certain nombre d'inconvénients de la voiture individuelle :

- il faut être capable de conduire dans les conditions difficiles de la circulation urbaine, ce qui pose un problème avec une population dont la moyenne d'âge tend à augmenter ;
- les voitures individuelles ont un ratio "passager/véhicule" souvent proche de 1 ;
- il faut parvenir à se débarrasser de son véhicule une fois arrivé à destination, ce qui est de plus en plus difficile avec la raréfaction des places de parking.

Plusieurs tentatives ont été menées afin de réduire cette dépendance par rapport à l'automobile :

- des limitations pour traverser le centre des villes,
- des limitations de parking,
- création de structures du type autopartage, de véhicules en libre service (autolib),

– des schémas qui associent des véhicules privés avec des transports publics, etc.

Malgré tout l'encouragement apporté à ce type de mobilité, ces suggestions ne restent que des solutions partielles et ne peuvent pas être considérées comme des modes de transport réalistes et autonomes. Pour les systèmes du type autopartage (covoiturage),

- il faut qu'au moins un des passagers puisse conduire ;
- pour chaque trajet effectué, les passagers devront :
  - pouvoir trouver d'autres passagers pour partager l'itinéraire ou au moins une partie de l'itinéraire ;
  - choisir un itinéraire plus ou moins convenable ;
- une modification imprévue (annulation) dans l'itinéraire sera difficilement supportée sans modification du coût ; par conséquent un tel système nécessite une planification préalable de la part des voyageurs, une tâche lourde en temps et effort.

Les structures du type autolib où des véhicules en libre service sont mis à la disposition des utilisateurs ne constitue pas non plus une solution optimale puisque :

- le ratio "nombre passagers par véhicule" n'est pas amélioré dans ces systèmes par rapport à l'usage de véhicules personnels ;
- généralement, on demande ou encourage les utilisateurs à amener le véhicule dans des stations bien précises ; par conséquent la qualité de service ne peut pas être comparable à celle d'une voiture individuelle offrant la possibilité d'un trajet porte à porte ; on doit s'assurer par ailleurs qu'il y a des places disponibles de stationnement ; si on autorisait l'utilisateur à abandonner le véhicule où il veut, il faudrait prévoir la technologie de localisation des véhicules et de leur transport et redistribution dans les points de parking ;
- l'utilisateur d'autolib doit être apte à la conduite.

Prenant acte du fait que la voiture particulière est le choix préféré de la population, au moins pour les déplacements en dehors des grandes migrations du matin et du soir, on envisage que les taxis peuvent offrir un service au moins équivalent sinon même avec quelques avantages supplémentaires (on se fait conduire par un professionnel qui connaît les itinéraires mieux que les particuliers, on n'a pas le souci du stationnement en fin de parcours, etc.). Malheureusement, c'est un moyen très coûteux, et ce constat n'est pas étonnant lorsqu'on considère le temps de conduite effective d'un chauffeur de taxi traditionnel qui passe beaucoup de temps en stationnement. Cette faible productivité a inévitablement une conséquence sur les coûts. Pour abaisser ces coûts, il faudrait donc faire partager le service par plusieurs utilisateurs tout en préservant ses qualités essentielles (service porte à porte, trajet presque direct), et accroître du même coup le temps de déplacement effectif de ces taxis devenus "collectifs".

Ce concept de "taxis collectifs" peut être rattaché à celui de "Transport à la Demande" (TAD) qui commence à devenir de plus en plus populaire [**42**],[**9, 13**]. Dans la section suivante, nous faisons une rapide revue de ce domaine afin de mieux situer notre idée par rapport à la variété des systèmes envisagés.

### I.2. Systèmes de transport en commun à la demande

Les systèmes de TAD ont des caractéristiques variées selon, par exemple, les restrictions imposées (itinéraires fixes ou semi-fixes [**20**], réservations préalables [**33**], etc.). D'autres systèmes qui associent une inter-mobilité entre des véhicules privés et des transports en commun ont été également proposés afin d'encourager l'utilisation du transport public. Un encouragement particulier est donné à tous ces systèmes afin de pouvoir lutter contre la congestion selon [**39**]. De tels systèmes existent déjà en Europe et aux États-Unis. Ils opèrent surtout dans les régions péri-urbaines, et souvent pour servir les populations âgées ou handicapées. Ces systèmes se rencontrent aussi dans les pays du Moyen-Orient, d'Asie, et d'Amérique Centrale et du Sud. Pourtant, les rapports entre les conditions d'exploitation et la praticabilité d'un système ne sont pas toujours clairs, et l'expérience d'un système est difficilement généralisable aux autres

systèmes à cause de différences entre les régions, par exemple dans les topologies et les utilisations des sols, ainsi que dans les habitudes de voyage des populations.

Par rapport à tous ces systèmes, notre concept de "taxis collectifs" vise à lever la plupart des restrictions pour un système de transport *collectif* (ce dernier mot impliquant des *prix bas*), opérant en ville, sans itinéraires ou points de rencontre fixes, et éventuellement sans réservations préalables.

Il existe en effet divers concepts de taxis collectifs, mais nous nous intéressons en particulier aux systèmes qui sont les plus proches des systèmes de taxis traditionnels. C'est-à-dire que nous voulons étudier les systèmes avec itinéraire complètement ajustable à une demande dynamique (service porte-à-porte). Les taxis sont collectifs dans le sens que les clients peuvent partager une partie ou tout le trajet tout en partageant le prix du trajet.

Un concept similaire a déjà été proposé par [**11**] pour constituer une alternative aux véhicules privés dans les villes et les banlieues. Le présent travail peut être considéré comme une suite à cette étude, mais aussi une extension : au delà du mode de fonctionnement "décentralisé" de [**11**] qui opère sans dispatching central, uniquement sur la base des rencontres taxi-client, nous envisageons aussi le mode "centralisé" avec appel des clients à un dispatching central, ce qui se justifie par le développement actuel des moyens de télécommunications. On peut espérer faire fonctionner efficacement de tels systèmes à condition, d'une part, de bien en optimiser la gestion en temps réel, et d'autre part, d'en ajuster avec précision le dimensionnement (nombre de taxis en service pour chaque niveau de demande, capacité des taxis en nombre de passagers, etc.). Avant de mieux expliquer la structure à laquelle on s'intéresse, on va présenter une courte revue des systèmes et des modèles existants.

**I.2.1. Brève revue des systèmes de TAD existants en France.** Les systèmes de transport en commun à la demande sont composés de véhicules qui servent plusieurs clients à la fois, mais sont pourtant plus souples que les systèmes de transport en commun traditionnels dans le sens que les itinéraires et les horaires des véhicules sont variés selon les origines, destinations et heures de départ des clients. Il existe de nombreuses variations des systèmes de transport en commun à la demande dans différentes régions du monde. Dans cette section, on décrit quelques uns des systèmes qui existent en France, en particulier dans la région Île-de-France, en considérant leur zone d'opération, les services offerts, la tarification des services, et la stratégie de routage des véhicules. La description des modes de transport à la demande qui suit doit beaucoup au travail d'Elaine Chang [**5**].

I.2.1.1. *ATA France.* Créé en février 2000, le siège social se trouve dans la cité de banlieue Mantes la Jolie-Val Fourré de l'Île de France. L'organisme compte 31 salariés dont 26 chauffeurs et 26 véhicules dont 3 adaptés aux personnes handicapées [**31**]. Le fondateur s'est établi à Mantes au départ sachant qu'à Mantes et ses environs, un ménage sur trois n'a pas de voiture et 30% de la population est au chômage, donc le tarif des taxis traditionnels n'est pas adapté à cette population. Les services des taxis collectifs, plus rapides que les autobus et moins chers que les taxis traditionnels, sont donc bien adaptés aux particuliers, mais la clientèle comprend également des entreprises, des VIP et des personnes à mobilité réduite. Le centrale de réservation accepte des appels du lundi au samedi de 8h à 20h, et les services de transport sont offerts 24 heures sur 24, 7 jours sur 7. Le chiffre d'affaires s'est accru de 106 000€ en 2000, jusqu'à 900 000€ en 2003, avec un portefeuille de grands comptes comprenant la SNCF, Matra, Dunlop, Sony, BMG comme clients. Au début, ATA France se trouvait en concurrence directe avec les taxis traditionnels, mais il travaille maintenant avec eux. Quand ATA ne peut pas assurer la demande, il fait appel aux taxis. En contrepartie, quand les taxis ne peuvent pas honorer certaines courses, ils renvoient la demande à ATA [**31**]. De plus, en cas de besoin de véhicules de grande capacité ou de prestige, des cars ou des limousines sont loués auprès des compagnies dans les alentours. Pour démarrer le concept dans d'autres villes en France, ATA France vend des franchises. Les franchisés peuvent utiliser le nom et bénéficier d'une formation pour reproduire le concept d'ATA [**1**]. Des franchises existent actuellement à Montpellier, Caen, Rouen, Toulouse et Annemasse. Aucun renseignement

n'a été trouvé sur la méthode de routage des véhicules, et la société n'a pas voulu partager cette information ni par email ni en personne. Par conséquent, nous ignorons si la société se sert d'un algorithme sur ordinateur pour le routage des véhicules.

I.2.1.2. *Diapason, Val de Bièvre.* Subventionné par les communes du Val de Bièvre, Diapason propose depuis juin 2003 des transports de porte à porte aux personnes à mobilité réduite, y compris les personnes âgées et handicapées. Le tarif est de 2€ par trajet, et les trajets sont limités au communes du Val de Bièvre, en banlieue de Paris. Les réservations doivent être faites la veille avant midi, et les services de transport sont proposés du lundi au vendredi entre 8h et 18h. Diapason se sert de deux véhicules du type "cab anglais" avec une capacité de cinq clients chacun et adapté à l'accueil des fauteuils roulants. Le routage des véhicules est effectué selon les connaissances et les estimations du personnel de la centrale de réservations, et non pas avec un algorithme sur ordinateur.

I.2.1.3. *Les navettes d'aéroport.* Les navettes d'aéroport sont une variation du concept de transport en commun à la demande et elles existent dans de nombreuses villes dans plusieurs pays. Les navettes d'aéroports transportent plusieurs clients de chez eux à l'aéroport ou de l'aéroport à la maison. Les itinéraires et les horaires sont adaptés aux demandes des clients, mais le problème de routage est simplifié car un bout du trajet est partagé entre tous les clients. En particulier, soit les clients partagent tous l'aéroport comme destination, soit les clients commencent tous leurs trajets à l'aéroport. Quelques systèmes de navettes d'aéroport qui existent dans la région parisienne sont le Yellow Van Shuttle, le Parishuttle et le Airport Connection. Ces organismes se servent de véhicules avec une capacité de cinq à huit clients, et fonctionnent selon les connaissances et les estimations du personnel à la centrale des réservations ou des chauffeurs pour effectuer le routage. Autrement dit, aucun n'utilise d'algorithmes de routage sur ordinateur.

**I.2.2. Systèmes TAD en Europe-Asie-États-Unis.** Des structures similaires à celles précédemment présentés pour la France existent déjà en Europe.

I.2.2.1. *Royaume-Uni.* On peut citer les services suivants.

**London Dial-a-Ride [21]:** Ce système offre des services de transport porte-à-porte à Londres à des personnes âgées ou handicapées, toujours avec réservations. Ils assurent des trajets de moins 5 miles. Un service équivalent est le Bristol Dial-a-Ride [**3**] lequel couvre la région de Bristol. Les clients sont informés si le retard va dépasser les 15mn et peuvent accepter ou refuser le service. D'autre part, les demandes peuvent être refusées si elles ne peuvent être satisfaites, et les réservations déjà attribuées peuvent être annulées dans le cas de pannes de véhicules, de mauvais temps etc.

**London Taxi sharing [22]:** Ce système offre des services porte-à porte aux clients ayant des destinations communes dans le but de diminuer le nombre de véhicules qui circulent dans la ville et, d'autre part, d'offrir un coût moins élevé pour les utilisateurs. Encore une fois, les réservations sont nécessaires. La tarification est basée sur la distance, la vitesse du véhicule et l'heure du service. Il y a toujours un prix minimal et le passager doit payer le prix total indiqué au taximètre sauf si le client et le chauffeur se mettent d'accord sur un prix au début du trajet. De plus, il y a une limite sur la distance que ce service couvre (12 miles ou un trajet durant au plus une heure). Des trajets plus longs sont acceptés seulement si le point de départ est l'aéroport de Heathrow.

I.2.2.2. *Belgique.* À Bruxelles, on peut citer, parmi d'autres, "Collecto" [**7**] qui est un service de taxis collectifs disponible 7 jours sur 7, de 6h à 23h. Il y a un nombre fixe de points de départ (plus de 200 points de rencontre) couvrant la région bruxelloise. Le prix est fixé à environ 6€.

I.2.2.3. *Pays-Bas.* On y propose le service "Regiotaxi" [**29**] entre des stations fixées pour les provinces de Overijssel, Salland, Vechtdal. Le service est disponible tous les jours entre 6h et 1h30. Il y a une prise en charge initiale de 1,65€ ou 2,65€ (selon le mode d'appel de l'utilisateur). Chaque zone supplémentaire coûte 1,65€.

I.2.2.4. *Autriche.* Un système de taxis collectifs a été lancé à Linz afin d'offrir une meilleure qualité de service en particulier la nuit [**36**], avec le double but de réduire le coût du transport public (qui fonctionne pendant la nuit) tout en offrant une bonne qualité de transport. Ce système dessert la totalité de l'agglomération avec 180 arrêts.

I.2.2.5. *Allemagne.* Dans ce pays où le prix du taxi est assez élevé, des services de taxis collectifs existent par exemple à Hanovre [**35**]. Il y a deux projets offrant ces services, "TaxiBus GmbH" et "Hallo Taxi 3811". Encore une fois, les réservations sont obligatoires au moins une heure avant l'heure de départ souhaitée. On remarque que l'Allemagne est un des pays offrant des services de transport spécialement pour des femmes qui peuvent éventuellement accompagner des enfants [**40**]. Ce service "üstra-FrauenNachtTaxi" est construit en collaboration avec l'association des femmes de la ville d'Hanovre. Ce système fonctionne entre octobre et mars de 19h à 6h, et du 1 avril jusque au 30 septembre entre 21h et 5h. Cependant lorsque un système de taxis collectifs sera mis en place par des professionnels, on pourrait assurer la sécurité des passagers (hommes et femmes de tout âge, enfants) tout en obtenant un prix compétitif.

I.2.2.6. *Suisse.* On y trouve le service Proxibus [**27**] (le même service existe aussi en France [**26**] surtout pour les personnes âgées). Le but est de transporter des gens qui vivent à la campagne, sur des liaisons transversales, c'est-à-dire à des endroits pour lesquels il n'existe pas de lignes régulières. Les origines des passagers peuvent être une adresse quelconque ou un arrêt de bus. Les destinations peuvent être soit une autre adresse soit un point de liaison pour une ligne régulière.

Le service Proxibus, employant des véhicules de petite capacité est divisé en trois secteurs, chacun ayant son propre numéro de réservation : Genève-Est, Genève-Sud et Genève-Ouest-Mandement.

Pour les clients disposant une carte "Billet Tout Genève" ainsi que pour ceux qui ont un abonnement ou une carte journalière, un supplément de 3CHF est requis. Le paiement s'effectue seulement en espèces auprès du conducteur Proxibus.

I.2.2.7. *Corée du Sud.* Les "Bullet Taxis" ou "Chongal Taekshi" [**6**] sont des véhicules pour des longues distances. Souvent ils cherchent à voyager avec 4 passagers à bord. Par conséquent les premiers passagers embarqués doivent attendre jusqu'à ce que le nombre requis de passagers soit atteint ou bien jusqu'à ce que le chauffeur se décide à partir.

À Séoul, on trouve les Hapseung [**16**] : il s'agit de taxis où le chauffeur demande aux passagers si ils acceptent de partager le véhicule avec d'autres clients afin de gagner le double du prix de la course. On rencontre facilement ce service du vendredi au lundi. Ainsi on est face à des véhicules qui s'arrêtent très souvent afin d'interroger des clients en attente tout au long du voyage. C'est le chauffeur qui décide de l'itinéraire ; si la destination du client candidat se trouve dans un endroit où on ne peut pas rencontrer facilement un grand nombre de nouveaux clients potentiels, le nouveau client est refusé. Avec ce critère même des personnes très âgées sont refusées si leur destination n'est pas de forte demande.

I.2.2.8. *Chine.* À Hong Kong, on trouve des minibus ou maxicabs qui passent dans des endroits où le transport public est rare ou inexistant [**34**]. Il s'agit de véhicules de 16 places qui sont très populaires en raison de la grande densité de population. Les minibus de couleur verte proposent des services continus. Ceux de couleur rouge offrent un service plus proche des taxis collectifs. Dans ce cas, le chauffeur attend jusqu'au moment où il estime le nombre de passagers suffisant pour couvrir ses coûts. Les minibus rouges peuvent opérer partout, sauf interdictions spéciales, sans contrôle particulier sur les itinéraires et les tarifs.

I.2.2.9. *États-Unis.* New York a mis en place un service de taxis collectifs le 24/02/2010 [**37**]. Le prix est fixé et les clients embarquent et débarquent à des endroits bien déterminés. On espère de cette façon diminuer les coûts de transport ainsi que le nombre de véhicules qui circulent dans la ville (surtout pendant les heures de pointe), les émissions, ainsi que mieux contrôler les congestions. Pour le moment, ce système couvre 3 avenues et il est envisagé d'élargir ce programme.

**I.2.3. Modèles de transport en commun à la demande.** En dehors des systèmes directement exploités sur le terrain, il existe plusieurs modèles de simulation de transport en commun à la demande dans la littérature.

Kunaka [**19**] a proposé un modèle de simulation comprenant un module de GIS (Geographic Information System) pour l'analyse du réseau et un module de mouvement de véhicules. Ce dernier calcule les positions des véhicules à intervalles réguliers (typiquement à chaque seconde) selon leur vitesse, ainsi que leur décélération et accélération aux arrêts. Le modèle est donc capable d'estimer les coûts de carburant au niveau de détail de l'intervalle temporel choisi. De plus, le modèle fournit les revenus, les temps d'accès, d'attente à bord et de descente, ainsi que le nombre d'arrêts et de trajets effectués. Enfin, un graphique de distance et temps est présenté pour chaque véhicule. L'approche ne comprend pas de centrale de réservation et donc modélise seulement les clients qui se présentent sur le trottoir sans réservation.

Fu [**13**] à proposé SimParatransit, un modèle de simulation qui considère les réservations faites à l'avance (au moins 24 heures) et les clients qui se présentent sans réservation. Au début de la simulation, les itinéraires des véhicules sont optimisés par le dispatcher selon les réservations faites à l'avance. Au cours de la simulation, les trajets des clients sans réservation sont insérés au moment de la rencontre entre le client et le véhicule. Les véhicules maintiennent une communication avec le dispatcher à travers un système de localisation automatique des véhicules ("automatic vehicle location" ou "AVL") et les itinéraires peuvent donc être ré-optimisés à intervalles réguliers. Les véhicules progressent sur les arcs du réseau avec un temps de parcours aléatoire sur chaque arc. Les temps de parcours sont tirés au hasard à partir d'une distribution log-normale lors de l'arrivée des véhicules à l'origine des tronçons. Le temps de parcours aléatoires modélisent donc les effets des embouteillages, des files d'attente et des feux de carrefour. Les durées des arrêts sont également aléatoires, et de plus le modèle considère les annulations et les clients qui ne se présentent pas. Le modèle donne comme sorties le nombre de clients servis, le nombre d'heures de service des véhicules, le nombre de clients servis par heure de service, la durée moyenne des trajets, le temps de retard des rencontres et le temps d'attente subi par les clients. L'auteur a présenté l'application de la simulation sur la ville d'Edmonton au Canada, et Fu et Xu [**15**], ont utilisé le modèle pour analyser les avantages de AVL dans le système à Edmonton. Fu [**14**] a continué ses études des systèmes de transport en commun à la demande en développant un modèle analytique pour le planning de la capacité (nombre de véhicules) et la qualité de service d'un système Paratransit. Les paramètres du modèle analytique ont été calibrés selon les résultats du modèle de simulation proposé dans Fu [**12**]. Un exemple d'application du modèle analytique a été présenté pour la ville d'Edmonton.

Horn [**17**] a proposé LITRES-2, un modèle de simulation de trajets multimodaux. C'est à dire que LITRES-2 modélise les mouvements des personnes dans un réseau qui comprend les transports en commun traditionnels ainsi que les transports en commun à la demande. Les choix des voyageurs sont donc estimés selon la qualité de service offerte par les différentes modes et une fonction généralisée de coût qui reflète les préférences du voyageur. La modélisation du système de transport en commun à la demande considère les réservations faites à l'avance, ainsi que les clients qui se présentent sur le trottoir sans réservation. Cependant, toute insertion de trajet dans un itinéraire doit être approuvée par le dispatcher qui vérifie que l'insertion respecte les contraintes d'affectation. Le modèle considère aussi les annulations, les clients qui ne se présentent pas, et la possibilité que les véhicules tombent en panne. Le routage des taxis collectifs est effectué avec un algorithme d'insertion des trajets lors de chaque réservation ou rencontre avec les clients. Les itinéraires peuvent être ré-optimisés par le dispatcher à intervalles réguliers. Le modèle permet de plus que les trajets soient complétés par des portions de marche à pied aux deux bouts. Des résultats numériques du modèle ont été présentés pour la région Gold Coast de Queensland en Australie.

Dessouky, Rahimi et Weidner [**8**] ont proposé un modèle de transport en commun à la demande qui optimise les coûts d'exploitation du système, la qualité de service et les conséquences

sur l'environnement. Le modèle d'optimisation comprend un module de simulation des mouvements des clients et des véhicules dans le système. La simulation modélise les réservations faites à l'avance (avant le début de la période de simulation) et les demandes des clients qui se présentent sur le trottoir sans réservation. Les temps de parcours des véhicules sont déterministes, mais les durées des arrêts sont aléatoires. Les positions des véhicules sont connues à tout moment par le dispatcher et le routage est effectué par un algorithme d'insertion qui considère les fenêtres qu'il est permis d'appliquer aux heures de rencontre.

Xu et Huang [**41**] proposent un modèle de simulation multi-agents pour un système TAD dont le but est de minimiser le nombre de véhicules utilisés, la durée des voyages des passagers, et les temps d'attente des clients.

Niles et Toliver [**18**] proposent une extension de type "car pooling" avec le IHOV (Intelligent High Occupancy Vehicle) en incitant les propriétaires des véhicules à transporter plus d'un passager dans leur véhicule, soit des gens de leur environnement familial et/ou amical, soit des inconnus, en utilisant les moyens offerts par les nouvelles technologies de la communication. Pour cela il faut pouvoir motiver les propriétaires d'accepter de partager leur propre véhicule avec d'autres voyageurs, s'assurer que pendant les allers-retours chaque véhicule transporte plusieurs personnes, mettre à jour des systèmes de télécommunication. En parallèle des questions de sécurité se présentent et une solution sécurisante semble difficile à proposer. D'autre part ce système rencontre des difficultés supplémentaires à cause de la résistance des acteurs des modes de transport "classiques" à la réalisation de tout système concurrent qui pourrait s'avérer opérationnel et remettre leur position acquise en question.

**I.2.4. Algorithmes de routage des véhicules.** La performance d'un système de transport en commun à la demande dépend en particulier de l'efficacité du routage des véhicules, et le routage des véhicules est donc un module essentiel dans le modèle de simulation. En premier lieu, on considère les modèles de simulation discutés dans la section précédente. Kunaka [**19**] qui avait proposé un modèle où les clients étaient acceptés uniquement sans réservation, indique que les trajets sont insérés dans les itinéraires, mais n'explique pas l'algorithme de routage suivi. Dans le modèle de Fu [**13**], les demandes avec et sans réservation sont acceptées, et le modèle permet l'insertion des trajets dans les itinéraires, ainsi que la modification des itinéraires, mais les détails des algorithmes ou des heuristiques de routage ne sont pas donnés.

Dans le modèle Horn [**17**], les demandes avec et sans réservation sont possibles, et Horn propose des algorithmes de routage pour optimiser l'insertion des demandes individuelles, ainsi que la ré-optimisation de tous les itinéraires à intervalles réguliers. Pour l'insertion des trajets individuels, Horn propose une énumération des insertions possibles de l'origine et de la destination du trajet. Comme alternative, il propose une heuristique qui optimise l'insertion de l'origine du trajet, et ensuite optimise l'insertion de la destination. L'heuristique ne garantit pas un routage optimal, mais les exemples numériques donnent de bons résultats. Horn propose également deux algorithmes pour améliorer les itinéraires après chaque insertion. Le premier cherche les possibilités de transférer des trajets d'un véhicule à un autre, et le deuxième cherche les possibilités de changer l'ordre des trajets dans un itinéraire. De plus, un algorithme pour ré-ordonner et échanger les trajets est proposé dans le but de ré-optimiser les itinéraires à intervalles réguliers.

Diana et Dessouky [**10**] ont proposé un heuristique pour chercher le routage qui minimise une fonction généralisée de regret qui considère la distance parcourue par les véhicules, le temps supplémentaire de trajet au-delà du trajet direct subi par les clients, et le temps inoccupé des véhicules. L'heuristique commence en insérant les trajets avec les heures de départ les plus proches et les points d'origine les plus éloignés du dépôt. Ensuite, pour insérer le restant des demandes, l'heuristique cherche la meilleure insertion (celle qui minimise l'augmentation du coût) dans chaque itinéraire. Une matrice d'augmentation du coût est donc construite où chaque ligne correspond à une demande et chaque colonne correspond à un véhicule. Si une demande ne peut pas être servie par un véhicule, une augmentation du coût infiniment grande est retenue. Ensuite, le regret est calculé pour chaque demande en prenant la somme des différences entre

chaque élément de la ligne et le minimum de la ligne. La demande avec la plus grande valeur de regret est insérée à la position calculée. Ces étapes sont répétées jusqu'à ce que toutes les demandes soient insérées.

Dial [**9**] a proposé une approche basée sur le concept d'une vente aux enchères pour un système décentralisé. C'est-à-dire que le système envisagé ne comprend pas de dispatching central, et par contre chaque véhicule est équipé avec son propre ordinateur. Les véhicules sont groupés par région, et quand un véhicule dans le groupe rencontre un client, chaque véhicule dans le groupe cherche à ajouter le trajet dans son itinéraire. Chaque véhicule calcule l'augmentation de coût résultant de chaque insertion possible du trajet dans son itinéraire. Le trajet est affecté au véhicule pour lequel l'augmentation est minimale. Cette approche est en fait un algorithme d'énumération exécuté sur des machines en parallèle.

Malucelli, Nonato et Pallotino [**23**] ont étudié trois versions du problème de routage des véhicules de transport en commun à la demande. La première version du problème exige que les clients soient pris et déposés exactement à leurs points d'origine et de destination. La deuxième version exige que les clients soient pris exactement à leur point d'origine, mais ils peuvent être déposés près de leur point de destination avec une pénalité à la mesure du service. Dans la troisième version, les clients peuvent être pris et déposés près de leurs points d'origine et de destination, toujours avec une pénalité à la mesure du service. Ces trois problèmes ont été formulés comme programmes linéaires en variables mixtes (entières et continues). Pour résoudre ces problèmes, les auteurs proposent des algorithmes d'énumération, relaxation lagrangienne et de génération de colonnes.

### I.3. Nos objectifs

Comme on l'a indiqué plus haut, notre projet de recherche s'intéresse aux systèmes les plus souples possible en matière d'itinéraires, d'origines et de destinations et d'horaires. Dans la suite, on va proposer une alternative *prometteuse* aux modes de transport existants appelée "taxis collectifs" par laquelle on cherche à offrir une qualité de service comparable à celle des taxis conventionnels ou des voitures particulières à des prix accessibles pour la majorité de la population.

Cet objectif sera atteint par la construction d'itinéraires intelligents et bien adaptés pour tous les passagers, en assurant que la presque totalité de la capacité choisie pour le véhicule est utilisée. On cherche donc, en optimisant le fonctionnement du système, à obtenir :

- une qualité de service équivalente à celle des taxis classiques,
- un service porte-à-porte,
- une attente réduite pour les usagers,
- des itinéraires bien adaptés, pas trop éloignés de leurs trajets directs,
- à des coûts comparables à ceux du transport public,

ce qui devrait contribuer :

- à une amélioration du trafic urbain,
- à une amélioration de le sécurité routière,
- au respect de l'environnement par l'utilisation de véhicules gérés par des professionnels.

**I.3.1. Modes de gestion du système.** On envisage trois modes de fonctionnement pour ce systèmes, à savoir :

**la gestion décentralisée :** dans cette approche les clients cherchent un véhicule dès leur apparition au nœud d'origine. Les taxis rencontrent aléatoirement les clients dans la rue et il n'y a aucun système de réservation préalable.

**la gestion centralisée :** dans ce mode les clients effectuent des réservations pour un départ d'un nœud donné dans une plage de temps donnée. Les itinéraires des taxis sont redéfinis dynamiquement par le dispatching central et sont composés des nœuds de rendez-vous avec les clients et des nœuds de destination de ces clients.

**La gestion mixte :** il s'agit du cas le plus complet et par conséquent du plus difficile à gérer. On considère les deux types de clients précédemment définis. Les véhicules peuvent transporter des clients ayant réservé un trajet à l'avance ainsi que ceux qui n'ont pas prévu de réservation, souhaitant partir dès que possible.

**I.3.2. La démarche : un outil de simulation et optimisation en face de données exogènes.** Ayant défini un système extrêmement souple et libéré du maximum de contraintes (pas d'itinéraires fixes, pas de réservations de la veille pour le lendemain, etc.), on est en présence d'un système offrant un maximum de potentialités, mais aussi en face d'un système extrêmement difficile à évaluer et à gérer. Or la compétitivité de ce système en face d'autres systèmes de transport collectif existants d'une part, de la voiture individuelle et du taxi classique d'autre part, dépend évidemment de son positionnement en termes de coût et de qualité de service, et on se doit de tirer le meilleur parti de ces potentialités.

Ceci oblige à recourir à toutes les ressources offertes par la Recherche Opérationnelle pour optimiser son fonctionnement et son dimensionnement. Ces techniques s'appuient généralement sur une modélisation mathématique qui s'avère ici pratiquement impossible : c'est un système multi-agents (taxis, clients, opérateurs du dispatching éventuellement) soumis à des conditions aléatoires (demande, temps de parcours, etc.). Il nous a semblé qu'un modèle de "simulation par événements" était le plus adapté à fournir un outil réaliste d'évaluation du système, des algorithmes et procédures de gestion employés (par exemple, on peut chercher à comparer les modes de fonctionnement décentralisé et centralisé évoqués plus haut) et du dimensionnement (nombre de véhicules en service, capacité de ces véhicules, etc.).

Bien sûr, la part de demande de déplacements qu'un tel système peut attirer dans une agglomération donnée dépend de son rapport coût/qualité de service. Mais en même temps, l'efficacité du système peut précisément dépendre du niveau de la demande à desservir : c'est la classique confrontation entre l'offre et la demande présente dans tout système économique. Nous considérons que la prise en compte des moyens déjà existants dans une agglomération pour servir les déplacements doit permettre aux économistes des transports de tracer la "courbe de la demande", c'est-à-dire le niveau de demande que chaque mode de transport peut capturer en fonction de son ratio coût/qualité de service. Notre ambition est de fournir l'outil qui permette de tracer la "courbe de l'offre taxis collectifs". Autrement dit, pour chaque niveau (et géométrie) de demande que l'utilisateur placera en entrée du modèle, il s'agit d'évaluer la meilleure réponse possible du système en termes de coût et de qualité de service. Il n'y a donc dans notre approche aucune boucle de feedback destinée à évaluer la part de marché que les taxis collectifs pourraient capturer.

De la même façon, on ne préjugera pas d'une politique tarifaire particulière, par exemple fixe (à la course), ou proportionnelle. On cherchera à fournir en sortie du modèle tous les éléments permettant de calculer le chiffre d'affaires en fonction d'une structure tarifaire donnée : nombre de clients transportés en moyenne par taxi, nombre de kilomètres servis par clients et toutes les données statistiques correspondantes (histogrammes, etc.).

En plus de ces éléments de coût, on attend du modèle qu'il fournisse toutes les statistiques utiles à l'évaluation de la qualité de service : durées des parcours rapportées aux durées des trajets directs (évaluation des détours), attente initiale, etc.

Tous ces indicateurs dépendront bien sûr des algorithmes de gestion temps réel (politique d'acceptation/refus des clients, affectation des taxis aux clients dans le cas de la gestion centralisée, construction des itinéraires, etc.). Ils dépendront aussi des ressources mises en œuvre : nombre de taxis en service, capacité de ces véhicules, nombre d'opérateurs au dispatching dans le cas de la gestion centralisée, etc. On se doute que certains indicateurs iront en s'améliorant lorsqu'on augmente par exemple le nombre de taxis en service, tandis que d'autres se détérioreront (nombre moyen de clients transportés par taxi, ce qui obligera à augmenter les tarifs). L'outil doit fournir au décideur tous les indicateurs permettant de réaliser les compromis qu'il juge les plus appropriés (optimisation de Pareto).

**I.3.3. Aperçu du mémoire.** Dans le chapitre II, on discute d'abord de la nécessité de disposer d'un modèle de simulation pour les systèmes de taxis collectifs et des avantages et inconvénients de ces modèles de simulation. On s'oriente vers une simulation de type "à événements discrets" et on décrit les principes généraux de ce type de simulation. Ce chapitre donne alors un aperçu de l'architecture du simulateur et l'on justifie le choix qui a été fait de bien séparer la simulation "mécanique" des phénomènes élémentaires de la partie dite "décisionnelle" qui recouvre tous les algorithmes de gestion temps réel. Ces algorithmes doivent être optimisés, mais il existe un autre niveau d'optimisation qui concerne les ressources mises en œuvre (dimensionnement du système). On décrit les données d'entrée requises, les sorties obtenues, et les divers modes d'exploitation du simulateur. Enfin, on discute des choix informatiques que nous avons faits pour réaliser ce simulateur.

Les deux chapitres suivants sont consacrés à la gestion décentralisée, c'est-à-dire un système de taxis collectifs sans réservation préalable, basé uniquement sur les rencontres taxi-client. Dans le chapitre III, on décrit d'abord avec plus de détails qu'au chapitre précédent les diverses entités impliquées (clients, taxis, files d'attente aux nœuds du réseau) et on donne une description de tous les types d'événements qui interviendront dans ce type de simulation et qui concernent, pour chacun d'eux, une ou plusieurs de ces entités.

On se tourne ensuite vers la partie décisionnelle de ce mode de gestion décentralisée. On formule le principal problème de décision qui concerne la question de l'acceptation ou du refus d'un client lorsque celui-ci rencontre un taxi déjà partiellement occupé par d'autres passagers. Cette question de l'acceptation ou du refus est indissociable de la reconstruction de l'itinéraire du taxi dans l'hypothèse où ce nouveau client est accepté. La formulation de ce problème de décision prend la forme d'un problème d'optimisation pour lequel plusieurs algorithmes de résolution plus ou moins exacts sont envisagés, et deux d'entre eux seront comparés au chapitre suivant.

La formulation de ce problème d'acceptation/refus fera intervenir un "seuil", c'est-à-dire un paramètre qui limite les détours subis par tous les clients (passagers déjà à bord comme nouveau candidat). C'est un objet de l'étude en simulation de déterminer, par des simulations répétées, une valeur raisonnable pour ce seuil. Un seuil restrictif permet d'obtenir une bonne qualité moyenne des trajets effectués par les clients (acceptés), mais s'il est trop sévère, il conduira à des refus trop fréquents de nouveaux candidats, ce qui entraîne l'allongement des temps d'attente et éventuellement un taux d'abandon trop élevé des clients potentiels (lassés d'attendre un taxi et trop souvent refusés).

Dans le chapitre IV, on montre comment se servir du simulateur pour calibrer un système de taxis collectifs dans la gestion décentralisée. Pour cela, on utilise une étude de cas basée sur des données fictives mais qu'on espère réalistes. On commence donc par décrire toutes ces données quantitatives. On décrit ensuite la façon dont les résultats des simulations peuvent être exploités.

Les simulations elles-mêmes se contentent de fournir un catalogue complet de tous les événements qui ont eu lieu pendant l'exécution du programme. De cette énorme masse de données, il faut alors extraire les données relatives à chaque entité (clients, taxis, files d'attente aux nœuds du réseau), puis les traiter pour en extraire toutes les statistiques souhaitables. On peut commencer par vérifier que les lois de probabilité sur la demande où les temps de trajet des véhicules sur les arcs du réseau qui ont été introduites dans les données sont bien vérifiées statistiquement. Il y a ensuite deux modes d'exploitation des résultats.

**Analyse détaillée d'une simulation:** il s'agit d'examiner les statistiques moyennes, mais aussi *extrêmes*, pour toutes les entités (chaque client, chaque taxi, chaque nœud) afin de repérer les anomalies ou dysfonctionnements du système. Cette exploitation est interactive : par exemple, pour les taxis, on examinera combien de passagers ont été amenés à destination par chacun des taxis au cours de la simulation. On pourra par exemple alors repérer le taxi ayant transporté le moins de clients, puis examiner de plus près l'histoire de ce taxi et se demander pourquoi il a subi ce comportement en examinant en détail son historique. Ceci nous a permis par exemple de comprendre

pourquoi certains taxis passaient presque tout le temps de la simulation sans transporter de passagers à la suite d'une mauvaise conception de la politique de choix du nœud de stationnement des taxis à vide.

On peut aussi repérer par cette technique les zones géographiques du réseau qui sont mal desservies (allongement excessif des files d'attente ou des temps moyens d'attente) pour essayer d'y remédier.

**Analyse statistique d'une série de simulations:** comme on l'a vu au chapitre précédent, l'algorithme d'acceptation/refus comporte un paramètre de seuil que l'on peut chercher à fixer à une valeur raisonnable en exécutant une série de simulations qui, *toutes choses égales par ailleurs*, parcoure une série de valeurs de ce paramètre. En examinant la variation de quelques indicateurs pertinents, on peut ainsi déterminer une, ou une plage de, valeur(s) appropriée(s). Le même procédé sera aussi utilisé pour déterminer un compromis raisonnable dans le choix de deux paramètres simultanément, à savoir ce seuil d'acceptation et aussi le nombre de taxis à mettre en service. On étudiera de la même façon l'influence de la capacité des taxis en nombre de passagers.

Ce procédé permettra également de montrer l'influence sur les performances du système du niveau de la demande et de sa géométrie (demande de type centripète ou centrifuge dans une ville, ou demande équilibrée).

Avant le chapitre de conclusion, l'avant-dernier chapitre sera consacré à la gestion centralisée et à la gestion mixte. Ce sujet n'a pas pu être, faute de temps, étudié de façon très approfondie et il reste une piste pour de futures recherches. En particulier, il serait intéressant de pouvoir comparer, pour une ville donnée, les performances d'une gestion décentralisée et d'une gestion centralisée pour voir si l'existence d'un dispatching central se justifie. Dans l'état actuel de notre travail, la partie "mécanique" du simulateur est prête pour simuler la gestion centralisée et les événements correspondant à ce mode de gestion seront décrits en détail (ainsi que pour la gestion mixte). Pour ce qui concerne la partie "décisionnelle", on abordera la formulation du principal problème, à savoir l'affectation d'un client appelant le dispatching à un taxi en service, et ce problème suppose aussi la reconstruction de l'itinéraire d'un taxi en incorporant l'origine et les destination du candidat. On discutera d'algorithmes possibles pour résoudre ce problème en prenant en compte leur complexité.

Le mémoire se termine par un chapitre de conclusions.

CHAPITRE II

# Simulation pour les taxis collectifs

*Une station de métro c'est un endroit où les métros s'arrêtent, une station de taxis, c'est un endroit où les taxis s'arrêtent ; sur mon bureau j'ai une station de travail...*

*Anonyme*

## II.1. Taxis collectifs : un système complexe qui nécessite la simulation

**II.1.1. Une multitude de questions.** Plus un système est flexible et soumis à un minimum de contraintes a priori, plus il est potentiellement efficace, mais plus il est difficile à modéliser, à faire fonctionner, et par conséquent plus il est difficile de prédire ses performances dans des conditions variées.

Lorsqu'on considère notre système de taxis collectifs, on commence par se demander si un tel système pourrait être réalisable et il en résulte une multitude de questions du type :

- Quand et comment un tel système peut-il être efficace ?
- Quel est le nombre de véhicules à mettre en service ?
- Est-ce que ce nombre doit être constant ou doit-il varier pendant la journée ?
- Quelle doit être la capacité de ces taxis (nombre de passagers maximum) ?
- Quand doit-on accepter un client à bord d'un véhicule et dans ce cas comment reconstruire un itinéraire *optimal* pour tous les passagers et le véhicule en question ?
- Que doit-on faire d'un véhicule vide ?
- Comment évaluer les performances du système et caractériser la qualité de service ?
- Comment évaluer la rentabilité du système en fonction d'une politique tarifaire donnée ?
- Comment comparer diverses stratégies ou algorithmes de gestion ?
- etc.

Afin de pouvoir étudier un tel système, on a besoin d'un modèle mathématique auquel on peut appliquer des méthodes d'Optimisation et de la Recherche Opérationnelle. Mais il est impossible de pouvoir décrire avec précision le système de taxis collectifs par une formulation mathématique à cause de sa grande complexité.

**II.1.2. Expérimentations sur le terrain : risquées et coûteuses.** Souvent afin de pouvoir tester le comportement de nouveaux systèmes de transport, on procède brutalement par des applications directement sur le terrain. Ou bien on commence par des modélisations très simplifiées et l'extrapolation de ces résultats à des situations plus réalistes sont hasardeuses. Une fois transposé sur le terrain, le comportement du système peut s'avérer très différent : les conséquences financières seront généralement catastrophiques pour ne pas parler du mécontentement des clients. Dans le meilleur des cas, il faudra éventuellement beaucoup de temps et de tâtonnements pour arriver à reconfigurer le système et le ramener vers un fonctionnement plus satisfaisant.

La nécessité d'un moyen fiable d'évaluation du système et la possibilité de concevoir et d'expérimenter diverses stratégies de gestion sans coût excessif et en face de scénarios variés nous paraissent évidentes. Dans d'autres domaines comme l'aéronautique par exemple, il n'est

pas concevable de construire réellement un système sans l'avoir modélisé et étudié en détail par des moyens virtuels comme ceux offerts aujourd'hui par la technologie.

**II.1.3. Une solution offerte par la technologie : la simulation.** On est confronté à un problème spatio-temporel complexe et un apprentissage approfondi de son comportement est indispensable pour pouvoir atteindre des performances souhaitables. Une modélisation mathématique du problème pourrait être une réponse à notre problème si seulement elle était réalisable. Le progrès des nouvelles technologies conduit de plus en plus à la construction de systèmes dans des environnements virtuels reproduisant le comportement de systèmes réels par des logiciels spécialement adaptés. Un simulateur n'est rien d'autre qu'une architecture modélisant une structure réelle, et donc souvent très complexe, permettant d'appréhender son comportement, lequel serait difficile à prévoir par toute autre méthode. Chaque politique planifiée peut alors être expérimentée efficacement sans risque et en général à peu de frais et en temps accéléré dans cet environnement virtuel.

II.1.3.1. *Avantages et inconvénients des simulations.* La construction d'un modèle de simulation pour un système éventuellement très complexe peut être cependant relativement simple lorsque la complexité du système provient du très grand nombre d'agents ou d'entités qui interagissent dans le système, mais que le comportement individuel de ces agents est relativement simple à reproduire. C'est en particulier le cas de notre système où chaque client ou taxi a un mode opératoire relativement simple, mais ces agents sont nombreux et ont de nombreuses interactions. La construction d'un simulateur peut être alors relativement modulaire.

Une autre source de complexité provient souvent d'un environnement aléatoire. Mais la simulation permet de représenter ces conditions. Même si les lois de probabilité sous-jacentes ne sont pas bien connues, on peut expérimenter avec des lois variées, ce qui revient juste à modifier les données d'entrée du programme de simulation, et évaluer ainsi l'importance ou la sensibilité à la représentation exacte de ces phénomènes aléatoires.

D'une manière générale, l'intérêt d'un simulateur est qu'il permet de tester le système et d'en étudier le comportement dans des conditions très diverses juste en modifiant les données d'entrée du programme afin de faire face à toutes les hypothèses que l'on peut rencontrer en situation réelle. En particulier, la simulation est le seul moyen qui permet de maîtriser totalement l'environnement et de ne changer qu'un seul paramètre à la fois, ce qui est une situation très favorable pour optimiser certains paramètres par exemple.

Même si la construction du simulateur peut être simple dans son principe, elle peut s'avérer cependant longue et fastidieuse et il peut en être de même pour son utilisation (en dépit du progrès en vitesse des machines modernes). Une autre tâche complexe est l'exploitation des résultats. Un simulateur peut potentiellement produire une très grande quantité de sorties qui peuvent être exploitées pour produire des statistiques et répondre à toutes sortes de questions. Mais il peut être difficile de dégager les indicateurs les plus pertinents. De plus, ces modèles permettent de constater un certain nombre de phénomènes dont certains peuvent s'avérer surprenants ou inattendus, mais ils ne fournissent pas d'explications sur ces constats. Contrairement aux modèles mathématiques qui sont plutôt de nature synthétique, les modèles de simulations opèrent à un niveau plutôt microscopique et c'est parfois à ce niveau qu'il faut redescendre pour comprendre les observations qu'ils permettent de faire.

II.1.3.2. *Diverses techniques de simulation.* Selon les techniques employées, on peut classer les simulateurs en trois grandes catégories :

- les simulations *temporelles*,
- les simulations *à événements discrets*,
- les simulations *par agents*.

Chacune de ces techniques est plus ou moins adaptée à certaines applications en particulier. Le choix dépend surtout du système qu'on souhaite analyser et du type d'étude qu'on souhaite faire.

II.1.3.3. *Les simulations temporelles.* Elles sont surtout adaptées à l'analyse du comportement d'un système lorsque l'évolution temporelle de celui-ci peut être représentée par une série d'équations différentielles ou récurrentes en temps discret. Ainsi on peut suivre cette évolution en déroulant le temps. Ce type de simulation suppose donc qu'on dispose d'un modèle constitué par un certain nombre d'équations, généralement un nombre relativement limité, sinon les temps d'exécution risquent de devenir prohibitifs.

II.1.3.4. *Les simulations à événements discrets.* Elles sont basées sur le fait que l'état du système se modifie chaque fois que certains événements se produisent. Ce type de simulation est donc adapté lorsque on traite de systèmes dynamiques dont l'intérêt principal n'est pas de suivre l'évolution continûment, mais plutôt de se concentrer sur certains changements discontinus. Ces systèmes peuvent être de grande taille mais leur comportement résulte de l'application de principes simples. Ce type de simulation est considéré en général comme assez rapide et fournit des résultats relativement précis.

Dans une simulation par événements discrets, on traite les événements un par un et l'avancement du temps se fait lorsque l'on passe d'un événement au suivant. Dans une simulation temporelle, c'est l'avancement de l'horloge qui provoque le déroulement des phénomènes.

II.1.3.5. *Les simulations par agents.* Dans ces simulations, le système est composé d'entités autonomes qui interagissent entre elles pour constituer le système global Ce type de simulation est plus adapté pour des systèmes qui se trouvent déjà en état d'équilibre ou qui en sont proches. Quand un modèle analytique permet aux utilisateurs de caractériser l'équilibre du système, les modèles par agents favorisent la possibilité de générer cet équilibre. Ce type de simulation est souvent utilisé pour analyser les congestions du trafic, représenter des modèles économiques ou sociaux, dans les applications biomédicales, etc.

## II.2. Vers un simulateur de systèmes de taxis collectifs

Le discussion précédente nous a conduits à la conclusion que pour espérer mener une expérience d'un système de taxis collectifs ayant un comportement optimal, la meilleure voie est de construire d'abord un outil capable de reproduire l'évolution du système par le moyen de simulations numériques.

Pour cela il faut :
- choisir la technique de simulation qu'on va adopter,
- créer le modèle représentant le système de taxis collectifs,
- valider le modèle,
- apprendre à gérer les résultats obtenus,
- optimiser les performances du système.

**II.2.1. Choix du type de simulation.** On choisit d'approcher le système de taxis collectifs par la technique de la simulation à événements discrets dans laquelle on reproduit tous les phénomènes élémentaires. Par le mot *système*, on désigne un ensemble d'entités qui interagissent entres elles dans le temps. Dans une simulation à événements discrets, le fonctionnement du système est représenté comme une suite chronologique d'événements. Chaque événement se produit à un instant donné en modifiant l'état du système. Plus précisément l'évolution du système est représentée comme une suite de la forme :

$$\{\ldots, \{s_i, t_i, e_i\}, \{s_{i+1}, t_{i+1}, e_{i+1}\}, \ldots\}, \quad i = 0, 1, \ldots,$$

où $s_i$ est l'état du système à l'instant $t_i$ et $e_i$ est l'événement qui se produit au même instant et qui fait passer le système de l'état $s_i$ à l'état $s_{i+1}$ et ainsi de suite. Bien sûr, $t_i \leq t_{i+1}$. Autrement dit, la pile des événements est rangée dans l'ordre chronologique. En fait, du point de vue informatique, il n'est pas nécessaire de ranger vraiment toute la pile mais seulement d'être capable de repérer le premier événement à traiter, ce que permet de faire une instruction du langage Python avec lequel le simulateur a été écrit.

La particularité, et en même temps la difficulté, de cette technique de modélisation est qu'il faut déterminer d'une part quels seront les types d'événements décrivant l'évolution du système, et d'autre part, pour chaque type d'événement, on doit définir une procédure qui sera activée lors de l'avènement de ce type d'événement. Cette procédure de traitement engendrera elle même de nouveaux événements qui seront placés dans leur position chronologique dans la pile des événements à traiter dans le futur.

Ainsi, si $\{E_i\}_{i=1,\dots,N}$ sont les $N$ types d'événements définis, on doit concevoir les $\{P_i\}_{i=1,\dots,N}$, où $P_i$ est la procédure appelant éventuellement d'autres procédures décisionnelles (la partie algorithmique ou "contrôle" du simulateur), puis définissant d'autres événements qui seront éventuellement des conséquences de ce traitement, et finalement remettant à jour l'état du système.

**II.2.2. Hypothèses et précautions.** La durée de chaque procédure $P_i$ est supposée instantanée ; autrement dit, pendant le déroulement d'une telle procédure, on suppose que le temps réel *n'avance pas*. La raison de cette hypothèse est la suivante : puisque le temps réel n'avance pas, l'état du système ne peut pas changer pendant que la procédure se déroule et on ne risque pas d'avoir deux procédures concomitantes qui voudraient modifier l'état du système de deux façons éventuellement contradictoires ou incompatibles.

Alors comment modéliser un événement qui a une durée ? Dans la modélisation du système par un réseau de Petri [**25**] notre hypothèse revient à l'hypothèse que toutes les activations de transitions sont instantanées. Pour modéliser une transition qui a une durée, il faut dédoubler cette transition en deux transitions instantanées, un *début* et une *fin*, séparées par une place qui portera la durée de la transition originelle (voir Figure II.1).

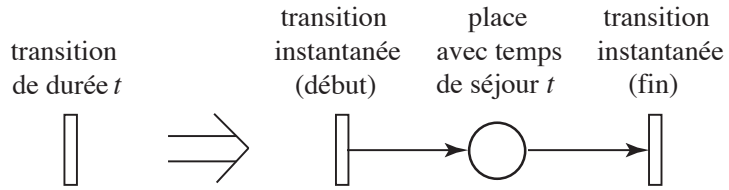


FIGURE II.1. Retour à des transitions instantanées dans un réseau de Petri

De la même façon, pour tout type d'événement ayant une durée, il nous faudra créer deux types d'événements représentant le début et la fin. Le traitement du début de l'événement conduira à créer l'événement de fin. Ainsi par exemple, comme on le verra plus en détail ultérieurement, lors de la mise en attente d'un client au nœud d'apparition, il y aura la création d'un événement de fin de l'attente de ce client. Cet événement pourra être annulé par le traitement d'autres événements comme par exemple la rencontre avec un taxi et l'embarquement du client dans le taxi avant l'apparition de l'événement "fin d'attente".

D'une manière générale, pour toute action ayant une durée, une heure limite et la création d'un événement "fin" sont systématiquement prévues de sorte que l'on soit certain qu'aucune entité ne soit "oubliée" éternellement dans le système.

**II.2.3. Aperçu du fonctionnement d'une simulation.**

II.2.3.1. *Horloge, temps et simultanéité.* L'horloge du simulateur indique l'heure actuelle dans la simulation. À chaque événement instantané est associée une heure. L'horloge du simulateur avance dès qu'on passe d'un événement au suivant à condition que les deux événements ne soient pas simultanés. *L'unité de temps utilisée dans le simulateur est la seconde.*

En théorie, deux événements peuvent avoir exactement la même heure. En pratique, ils seront traités l'un après l'autre. Si ces deux événements ne concernent pas les mêmes entités ou des entités potentiellement en interaction dans un même événement, l'ordre de traitement de ces deux événements simultanés n'a pas d'influence sur le déroulement de la simulation (par

exemple la montée simultanée de deux clients différents en des nœuds différents et évidemment dans des taxis différents).

Mais dans le cas contraire, cette affirmation est fausse. Par exemple, si un client attend un taxi et que son heure limite d'attente est atteinte, le client va quitter le système. On peut imaginer qu'un taxi arrive précisément à l'heure limite. Si l'événement "le client abandonne" est traité avant celui de l'arrivée du taxi, le résultat peut être différent du cas où l'arrivée du taxi au nœud est traité *avant* celle du départ du client de ce nœud. Cela n'a cependant rien de choquant puisque c'est aussi ce qui peut se passer dans la "vraie vie".

II.2.3.2. *Pile des événements.* Comme on l'a dit, un élément central de la simulation est la pile d'événements qui stocke tous les événements futurs prévisibles avec leur date prévue de réalisation. La simulation avance en traitant à chaque fois l'événement qui se trouve en haut de la pile dans l'ordre chronologique, et ce traitement peut entraîner la création d'autres événements prévisibles dans le futur.

Tous les événements ainsi créés ne seront pas forcément effectivement réalisés. On a donné l'exemple d'une entité (client, taxi, etc.) qui, lorsqu'elle est mise en attente, se voit attribuer un événement "fin d'attente" à l'heure limite décidée pour cette attente. Mais si d'autres événements viennent sortir cette entité de cette attente avant l'heure limite, cet événement "fin d'attente par atteinte de l'heure limite" ne sera pas traité.

Lorsqu'un client apparaît à un nœud, on verra que cette apparition est modélisée par un processus de Poisson : ceci signifie que l'intervalle de temps qui sépare cette arrivée de celle du client suivant au même nœud est une variable aléatoire de loi exponentielle. Par conséquent, dès l'apparition d'un client, on est en mesure de programmer l'apparition du prochain client dans le futur en tirant aléatoirement une réalisation de cette variable aléatoire.

De la même façon, lorsqu'un taxi quitte un nœud du réseau en direction d'un autre nœud, son temps de parcours sur l'arc correspondant est une variable aléatoire selon une certaine loi. On tire donc, dès son départ, le temps du parcours, ce qui permet de programmer l'événement de son arrivée au prochain nœud.

II.2.3.3. *Déroulement d'une simulation.* Lors de l'exécution de la simulation, les étapes suivants sont réalisées :

(1) l'événement ayant la plus petite date est récupéré dans la pile ;

(2) on compare la date de cet événement à l'heure limite de fin de simulation choisie par l'utilisateur :
   - si la date de l'événement en question est inférieure à cette heure de fin de simulation :
     - l'heure actuelle de la simulation est mise à jour à l'heure de l'événement correspondant (on rappelle que tout événement est supposé instantané) ;
     - l'événement est activé et son traitement commence ;
     - à la fin de ce traitement, l'événement sera supprimé de la pile d'événements ;
   - si l'heure de cet événement est supérieure ou égale à l'heure de fin de simulation, on arrête la simulation.

Dans ces conditions, la pile n'est jamais vide en fin de simulation : elle contient les événements non traités parce que devant survenir au delà de l'heure de fin. Par exemple, l'apparition d'un client à un nœud entraînant systématiquement la programmation de l'apparition du prochain client au même nœud, il reste toujours des événements à traiter dans la pile. Ces événements sont conservés afin de pouvoir reprendre une simulation là où elle a été interrompue si l'utilisateur le souhaite. Ce n'est que lors du "démarrage à froid" d'une simulation que la pile est vide et il faut commencer par l'initialiser par la création des premiers événements (faire apparaître un premier client à chaque nœud, positionner les taxis aux nœuds avec une heure limite de stationnement, etc.).

## II.3. Architecture du Simulateur

La discussion précédente nous a conduits à adopter l'architecture du simulateur illustrée par la figure II.2.

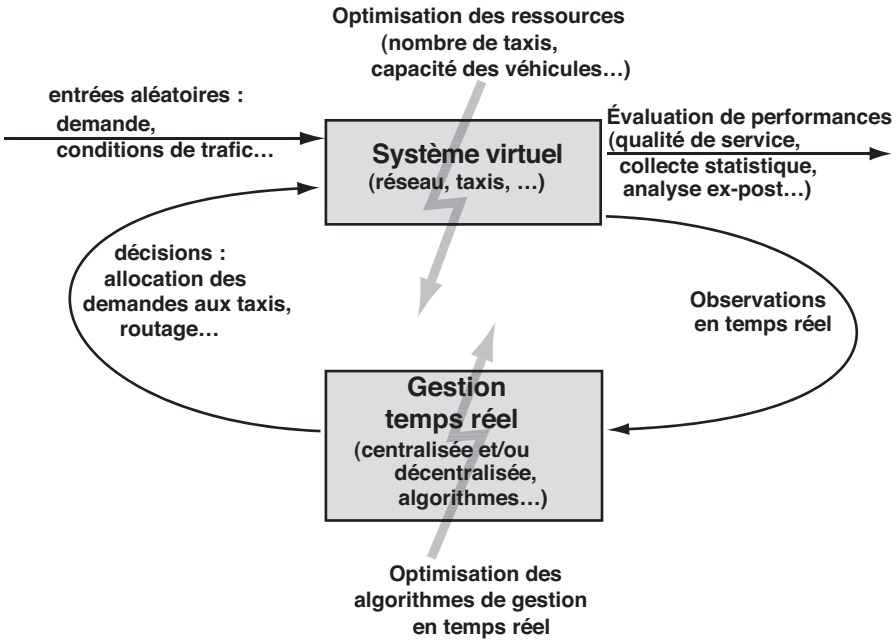


FIGURE II.2. Architecture du Simulateur

**II.3.1. La partie mécanique.** C'est une représentation virtuelle du système réel. Elle reçoit deux types d'entrées :

- des entrées exogènes, données déterministes relatives aux ressources utilisées, et lois de probabilité décrivant les phénomènes stochastiques comme les conditions de trafic, la demande, etc. ;
- des décisions ou "contrôles" qui sortent de la "boîte de feedback" qui constitue l'autre partie du simulateur.

La partie mécanique est chargée de faire évoluer l'état de l'ensemble des agents présents dans le système et d'interroger si nécessaire la partie décisionnelle. Plus précisément, cette partie du simulateur est chargée :

- de détecter l'arrivée d'un véhicule à un nœud,
- de faire débarquer les clients dont la destination est ce nœud,
- de traiter les clients en attente à ce nœud pour les faire dialoguer avec le taxi,
- de traiter les rendez-vous avec des clients ayant réservé au départ de ce nœud,
- de gérer les taxis vides,
- de gérer les appels des clients désirant faire une réservation,
- de faire sortir du système des clients en attente lorsque l'heure limite de leur attente est atteinte,
- de faire partir un véhicule d'un nœud en déterminant l'arc qu'il doit emprunter,
- de mettre hors service les ressources (taxis, dispatchers) ayant atteint leur heure limite et ayant achevé le service en cours, ou au contraire faire entrer en service une nouvelle ressource (sur décision exogène),
- etc.

**II.3.2. La partie décisionnelle.** Elle est responsable de toutes les décisions de gestion temps réel du système. La partie décisionnelle est en étroite interaction avec la partie mécanique : chaque fois que celle-ci a besoin d'une décision, elle formule sa question à la partie décisionnelle en fournissant les éléments de contexte et les informations requises. C'est cette interaction entre les deux parties qui fait finalement évoluer le système. On donne maintenant quelques exemples de telles décisions.

- Lors de la rencontre entre un client sans réservation et un taxi, on doit déterminer si le client en question peut être accepté à bord du véhicule. La réponse, positive ou négative, passe par la détermination d'un itinéraire du taxi incorporant la nouvelle destination.
- Quand un véhicule se retrouve à vide, il faut déterminer un nœud vers lequel le taxi doit se diriger pour stationner ainsi que le temps maximal d'attente à ce nœud.
- Au moment où un véhicule est prêt de partir d'un nœud, il faut déterminer le nœud suivant vers lequel il doit se diriger en fonction des points de passage prévus dans son itinéraire.

**II.3.3. Modes d'exploitation du simulateur.** Comme on le verra ultérieurement, les algorithmes de gestion temps réel peuvent comporter un ou plusieurs paramètres dont une valeur raisonnable ne peut pas, en général, être déterminée a priori. Il faudra donc réaliser des séries de simulations en gardant autant que possible les mêmes conditions aléatoires (par exemple le même historique d'apparitions des clients aux nœuds avec les mêmes destinations) afin d'explorer une plage de variation de la valeur de ces paramètres.

En plus de cette optimisation paramétrique de la gestion temps réel, une autre raison de réaliser des séries de simulations "toutes choses égales par ailleurs" concerne le dimensionnement des ressources du système, comme par exemple le nombre de véhicules mis en service, leur capacité en nombre de passagers, le nombre de dispatchers employés etc.

On verra donc plus tard qu'on doit chercher le meilleur compromis entre plusieurs objectifs divergents comme la qualité de service offerte d'une part, le coût de fonctionnement d'autre part.

Nous avons en conséquence prévu plusieurs modes d'utilisation du simulateur. Une simulation peut démarrer "à froid", c'est-à-dire qu'initialement le système est vide de clients. Il faut d'abord répartir les taxis sur les nœuds du réseau et faire apparaître les premiers clients pour initialiser la pile d'événements. Les données, comme les lois de probabilité décrivant la demande ou celles relatives aux temps de parcours sur les arcs des véhicules, peuvent varier au cours de la simulation ou bien être maintenues fixes. Dans le second cas, on peut espérer, si la simulation est assez longue, que les statistiques des grandeurs observées se stabilisent dans un régime stationnaire qui aura "oublié" l'état initial de la simulation. C'est pourquoi il a été prévu de pouvoir *continuer* un simulation, c'est-à-dire pouvoir relancer une simulation à partir d'un état initial hérité de la fin d'une simulation précédente, ce qui permet ainsi de prolonger celle-ci au cas où on estimerait que l'état stationnaire n'a pas encore été atteint.

Par ailleurs, pour réaliser les séries de simulation permettant d'optimiser un paramètre en essayant une série de valeurs, il est souhaitable de maintenir aussi égales que possible toutes les autres conditions. Ceci n'est pas rigoureusement possible. En effet, changer un paramètre dans un algorithme de gestion temps réel, c'est changer parfois la décision — accepter ou refuser un client par exemple — et ceci entraînera une évolution différente du système. Ainsi, un taxi pourra être amené à suivre des parcours différents d'une simulation à l'autre et les temps de parcours aléatoires des arcs ne pelistoffigures mêmes. Mais ceci est au moins possible pour toutes les entrées exogènes (ne dépendant pas des décisions temps réel) comme par exemple la création des clients en leur nœud d'origine ainsi que leur destination. On a donc prévu de pouvoir relancer des simulations sans créer de nouveaux clients mais en utilisant plutôt un historique enregistré d'apparition de ces clients (à condition bien sûr que l'historique couvre une période au moins aussi longue que la nouvelle simulation).

Finalement, on peut donc croiser deux caractéristiques dans les modes d'exploitation du simulateur :

(1) démarrage "à froid" ou reprise d'une simulation,

(2) création de nouveaux clients ou reprise d'un historique de ces clients.

### II.4. Choix informatiques pour la construction du simulateur

Il existe un certain nombre de logiciels ou de librairies spécialisés pour la création de simulateurs. On peut citer par exemple Scicos [**30**], [**4**] dans l'environnement Scilab, équivalents dans le monde du logiciel libre de Simulink et Matlab respectivement, aujourd'hui fusionnés dans ScicosLab. Ce logiciel permet la construction par bloc-diagrammes de simulateurs de systèmes dynamiques à la fois continus et discrets.

Elaine Chang [**5**] a développé son simulateur de taxis collectifs en utilisant la bibliothèque C$^{++}$Sim, librairie écrite en C$^{++}$ qui modélise chaque agent de la simulation par un "thread". Cette librairie n'est semble-t-il plus maintenue à l'heure actuelle et la technique des threads a semblé un peu lourde pour absorber le nombre d'agents dont on a besoin pour traiter des cas de taille réaliste pour notre modèle.

Au début de notre travail, nous avons fait quelques essais avec SimPy [**32**], [**24**] outil de simulation orienté objet écrit en Python, mais on s'est rendu compte que ce logiciel n'offrait pas tous les outils requis pour notre modélisation. Il fallait s'adresser aux concepteurs et leur demander d'enrichir leur librairie en ajoutant des fonctionnalités supplémentaires.

Finalement, pour être en mesure de maîtriser complètement notre entreprise, nous avons décidé de développer notre simulateur depuis le début à partir d'un langage informatique généraliste. Il nous fallait donc commencer par définir le modèle, et ensuite choisir le langage offrant des qualités suffisantes de modularité, d'extensibilité et de portabilité.

La modélisation "objet" permet de représenter informatiquement des éléments complexes du monde réel [**2**]. Ses avantages sont la robustesse, l'efficacité, mais surtout la possibilité de pouvoir développer des composants réutilisables. Les notions d'*héritage* et d'*héritage multiple* donnent la possibilité ce créer des structures de plus en plus complexes de façon précise et progressive. On s'est donc dirigé vers cette philosophie de la "programmation orientée objet" (P.O.O) plutôt que vers celle d'un langage classique.

Il fallait ensuite choisir le langage utilisé. C$^{++}$ est un langage compilé, très puissant, offrant tous les outils nécessaires pour un langage de P.O.O. Par ailleurs, Python [**28**], [**43**] est un langage orienté objet interprété. Il est très connu pour sa capacité à pouvoir traduire rapidement des architectures complexes. Ce langage figure en bonne place dans plusieurs domaines scientifiques, en offrant la possibilité d'avoir une syntaxe *claire et lisible*, une modularité complète, des librairies extensibles, etc. On peut développer *rapidement* et efficacement avec ce langage des maquettes d'applications.

Mais, comme il s'agit d'un langage interprété, il y a le risque d'une perte de rapidité à l'exécution. On peut déjà s'efforcer de pallier ces inconvénients en utilisant de bons principes de programmation et il reste ensuite la possibilité de remplacer progressivement certaines parties du code qui nécessiteraient d'être accélérées (comme par exemple des algorithmes complexes mis en œuvre dans la partie décisionnelle) par des procédures en C$^{++}$ qui seraient interfacées avec le reste du code.

Nous avons donc choisi, dans un souci d'arriver rapidement à un premier programme opérationnel, d'écrire à partir du début notre simulateur en Python, quitte à identifier ensuite les parties qui nécessiteraient un code compilé (mais nous n'avons pas eu finalement à en arriver à ce stade dans le cadre de ce travail).

### II.5. Les données d'entrée du programme de simulation

Dans cette section, on décrit d'abord les données d'entrée du programme de simulation que l'utilisateur doit réunir. Comme nous n'avons pas pu travailler sur un cas réel, nous avons dû fabriquer des données artificielles, mais que nous espérons réalistes, pour pouvoir mener les expérimentations décrites au chapitre IV. On décrira donc la façon dont ces données ont été fabriquées. Les valeurs numériques seront précisées au chapitre IV.

**II.5.1. Aperçu des données de simulation.** Ces données concernent essentiellement

**le réseau :** topologie du réseau (nœuds, arcs) sur lequel va opérer le système de taxis collectifs, lois de probabilité des temps de parcours des arcs par les véhicules (qui peuvent dépendre des périodes de la journée), etc. ;

**la demande :** c'est d'une part les lois de probabilité d'apparition des clients (avec ou sans réservation) à leur nœud d'origine, d'autre part les lois de probabilité qui déterminent leur destination, et enfin les lois de probabilité des temps d'attente (attente maximale avant abandon) ;

**le dispatching :** (dans le cas de la gestion centralisée — avec réservation — ou mixte) ce sont les lois des temps de service et de réponse des dispatchers.

**les taxis :** en plus des temps de parcours des arcs mentionnés plus haut, ce sont essentiellement les lois des temps de dialogue avec les clients, et des temps de montée et de descente des passagers.

En dehors de ces données, il faut bien sûr, pour un "démarrage à froid" d'une simulation, fixer les ressources mises en jeu (nombre de taxis mis en service, capacité en nombre de passagers, temps de service, nombre de dispatchers, etc.) et initialiser le système (répartition des taxis sur les nœuds du réseau).

La partie décisionnelle (§II.3.2) et l'ensemble des algorithmes qui la constituent font aussi en quelque sorte partie des "données" que l'utilisateur doit fournir, mais nous ne la mentionnons ici que pour mémoire.

II.5.1.1. *Réseau.* Le réseau est un graphe composé de nœuds et d'arcs *orientés* (chaque arc n'est parcouru que dans un sens). La modélisation habituelle des temps de parcours aléatoires est faite à l'aide de loi log-normales "avec shift". Le "shift" est une constante $s$ rajoutée à la variable aléatoire distribuée selon la loi log-normale afin que son support soit $[s, +\infty[$ au lieu de $[0, +\infty[$ (donc $s$ est une durée *minimale* du temps de parcours). La loi log-normale est une loi de densité

$$f(t) = \frac{e^{-(\log t - \mu)^2/2\sigma^2}}{t\sigma\sqrt{2\pi}} \tag{II.1}$$

où $\mu$ et $\sigma$ sont respectivement la moyenne et l'écart-type du logarithme de la variable aléatoire (logarithme qui est distribué selon une loi normale). La moyenne et la variance de $t$ sont données par :

$$\text{moyenne} = e^{\mu+\sigma^2/2}\,, \quad \text{variance} = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}\,. \tag{II.2}$$

Il faut donc se donner un triplet de paramètres $(\mu, \sigma, s)$ par arc du réseau.

Lors des calculs de plus courts chemins (exprimés en unités de *temps*) pour construire les itinéraires prévisionnels des taxis, on ne se sert que des *temps moyens* de parcours des arcs (voir (II.2) en rajoutant $s$ à la moyenne). En fait, nous avons *précaculé*, avant le démarrage de la simulation, tous les plus courts chemins entre toutes les paires de nœuds, et cette matrice des temps moyens des plus courts chemins a été stockée une fois pour toutes. De même, nous avons stocké la *matrice des stratégies* : pour toute paire ordonnée $(i, j)$ de nœuds (origine et destination d'un trajet), l'élément $(i, j)$ de cette matrice donne le nœud $k$ vers lequel il faut se diriger à partir de $i$ pour rejoindre $j$ par le plus court chemin [1].

II.5.1.2. *Demande.* Le processus d'apparition des clients au nœud d'origine est généralement modélisé par un processus de Poisson, c'est-à-dire que le temps qui sépare l'arrivée de deux clients successifs est une variable aléatoire distribuée selon une loi exponentielle de densité

$$g(t) = \lambda e^{-\lambda t}\,. \tag{II.3}$$

1. Nous sommes reconnaissante à Stéphane Gaubert (INRIA) pour nous avoir fourni un programme basé sur l'algorithme d'Howrd pour le calcul de ces deux matrices de plus courts chemins et de stratégies.

La paramètre $\lambda$ est le nombre moyen de clients arrivant par unité de temps (donc $1/\lambda$ est la moyenne de $t$, mais c'est aussi l'écart-type). Là encore, il y a bien sûr un paramètre $\lambda$ par nœud du réseau. Nous noterons désormais $\lambda$ le *vecteur ligne* de tous ces paramètres pour l'ensemble des nœuds du réseau.

Par ailleurs, lorsqu'un client apparaît à un nœud, il faut déterminer sa destination. Ceci est généralement fait en tirant au hasard selon la probabilité de chaque destination pour une origine donnée. La matrice dont chaque ligne, correspondant à une origine, représente la loi de probabilité de chaque destination (en colonne) est appelée *matrice Origine-Destination* (matrice O-D) et elle sera notée $M$. Puisque chaque ligne représente une loi de probabilité discrète, la somme de ses éléments sur chaque ligne vaut 1. Si on désigne par $\mathbf{1}$ le *vecteur colonne* de dimension égale aux nombre de nœuds et composé uniquement de 1, on a donc :

$$M\mathbf{1} = \mathbf{1}\,. \tag{II.4}$$

**II.5.2. Fabrication d'un jeu de données pour nos expériences.**

II.5.2.1. *Réseau.* Le réseau qui a été utilisé sera décrit au chapitre IV. Nous décrivons ici comment ont été obtenues les temps de parcours. Comme on l'a vu ci-dessus, il faut déterminer pour chaque arc du réseau trois paramètres $\mu, \sigma, s$. Pour cela, il nous faut trois équations. Voici celles qui ont été utilisées :

(1) la moyenne du temps de parcours sur un arc est proportionnelle à sa longueur ; on peut donc convertir ces longueurs en unités de temps à l'aide d'une vitesse commerciale nominale ; ceci donne une valeur numérique pour la moyenne (voir (II.2) en ajoutant $s$) ;

(2) pour fixer le shift $s$, c'est-à dire le temps minimum de parcours d'un arc, on a supposé que ce temps de parcours minimum est égal à 80% de la moyenne ;

(3) la troisième relation est obtenue en supposant que le quantile à 99% est situé à deux fois la moyenne, c'est-à-dire que la probabilité que le temps de parcours dépasse le double du temps moyen est de 1%. D'après *Mathematica*, le quantile à la probabilité $q$ (= 0,99 dans notre cas) est donné par

$$e^{\mu+\sqrt{2}\,\mathrm{InverseErf}(2q-1)} + s\,.$$

C'est avec *Mathematica* que nous avons résolu ces trois équations pour obtenir les valeurs numériques de $(\mu, \sigma, s)$ pour chaque arc. Curieusement, il s'avère qu'il y a deux solutions à ce système d'équations et nous avons retenu celle qui correspond à la plus petite variance. Nous donnerons au chapitre IV une idée des temps de parcours sur le réseau utilisé.

II.5.2.2. *Demande.* Nous indiquerons au chapitre IV comment a été obtenu le vecteur $\lambda$ des paramètres des lois exponentielles pour le réseau considéré, ainsi que la matrice O-D $M$.

Il est intéressant de faire le bilan, par unité de temps et pour chaque nœud $i$, entre le nombre de clients apparaissant à ce nœud, à savoir $\lambda_i$, et ceux qui apparaissent aux autres nœuds du réseau et qui désirent arriver en ce nœud $i$, à savoir $\sum_j \lambda_j M_{ji} = (\lambda M)_i$. Le bilan de à tous les nœuds s'écrit donc vectoriellement :

$$b = \lambda M - \lambda\,. \tag{II.5}$$

Les coordonnées positives de ce vecteur indiquent les nœuds plutôt "récepteurs" et les négatives indiquent les nœuds plutôt "émetteurs". Par exemple, dans un graphe représentant une ville, si la demande est *centripète*, les nœuds émetteurs sont à la périphérie et les nœuds récepteurs sont au centre de la ville.

Une demande *parfaitement équilibrée* est une demande où le bilan (II.5) de tous les nœuds est nul. Pour $M$ donnée, une telle demande équilibrée correspond à un vecteur propre à gauche associé à la valeur propre 1 de $M$ :

$$\nu M = \nu\,. \tag{II.6}$$

La valeur propre 1 existe bien d'après (II.4). Le vecteur propre est défini à une constante multiplicative près. Si on veut que le nombre total de clients apparaissant sur le réseau par unité de temps soit égal à $a$, on choisira la constante telle que $\nu\mathbf{1} = a$. C'est ainsi que nous construirons, pour nos expériences numériques au chapitre IV, une demande équilibrée à partir d'une demande centripète, en conservant la matrice $M$ mais en changeant $\lambda$ en $\nu$, solution de (II.6) et de $\nu\mathbf{1} = \lambda\mathbf{1}$.

Enfin, on peut s'intéresser à la demande *réciproque* d'une demande donnée, c'est-à-dire un flux de déplacements correspondant aux déplacements inverses ou "retour" d'un flux original. Le vecteur "bilan" (II.5) associé à cette demande réciproque aura le signe opposé de celui associé à la demande originale (une demande *centripète* sera transformée en demande *centrifuge*). Examinons comment on peut construire cette demande réciproque, caractérisée par le couple $(\mu, N)$ d'une demande caractérisée par le couple $(\lambda, M)$.

L'équation fondamentale est celle qui exprime que le nombre de clients par unité de temps qui veulent se déplacer de $i$ à $j$ dans les trajets directs est le même que le nombre de clients par unité de temps qui veulent se déplacer de $j$ à $i$ dans les trajets retour. Ceci donne

$$\lambda_i M_{ij} = \mu_j N_{ji} \,. \tag{II.7}$$

En sommant ces équations par rapport à l'indice $i$, on obtient

$$\sum_i \lambda_i M_{ij} = \mu_j \sum_i N_{ji} = \mu_j \,. \tag{II.8}$$

On obtient donc les équations qui permettent de calculer les $\mu_j$. Ensuite, avec (II.7), on calcule les $N_{ji}$.

## II.6. Sorties brutes du programme de simulation

Pendant la simulation, on se contente simplement d'enregistrer tous les événements dans une base de données sans chercher à exploiter d'une manière ou d'une autre les résultats. Cette base de données (très volumineuse pour 8 heures de simulation temps réel) permet de retracer exactement l'histoire de n'importe quel entité, ce qui nous a été très utile dans la phase de débogage du programme de simulation.

Au delà de cette phase de mise au point, on s'est empressé d'écrire des procédures d'extraction permettant de fabriquer des fichiers plus concis destinés à l'analyse et à l'exploitation des résultats. Ces analyses seront décrites au chapitre IV pour le cas de la *gestion décentralisée*. Disons simplement ici que l'on extrait de la base de données :

- un fichier clients,
- un fichier par taxi en service.

**II.6.1. Fichier clients.** C'est un fichier à 10 colonnes et autant de lignes que de clients apparus pendant la durée de la simulation. Les 10 colonnes contiennent les informations suivantes (les champs éventuellement inutilisés contiennent $-1$) :

(1) numéro d'identification du client (chaque "objet" reçoit un numéro au moment de sa création) ;

(2) heure d'apparition du client ;

(3) heure limite d'attente au nœud origine ;

(4) heure à laquelle le client a quitté effectivement ce nœud origine ;

(5) numéro d'identification du taxi ayant pris en charge le client ;

(6) heure de montée du client dans le taxi (cette heure est égale à l'item (4) ci-dessus augmentée du temps d'embarquement si le client a effectivement embarqué dans un taxi) ;

(7) heure de sortie du client de son taxi ;

(8) numéro d'identification du nœud origine ;

(9) numéro d'identification du nœud destination ;

(10) durée moyenne du trajet direct du client.

Le dernier item est une information redondante car elle est une conséquence directe des items (8) et (9) : cette information est stockée, comme nous l'avons dit au §II.5.1.1, une fois pour toute dans une matrice gardée en mémoire. Elle a été rajoutée dans le fichier pour accélérer le traitement post-simulation.

**II.6.2. Fichiers taxi.** Chaque fichier taxi est identifié par le numéro d'identification du taxi. Ce fichier contient autant de lignes que de nombre d'événements qui ont concerné ce taxi au cours de la simulation, et $n + 5$ colonnes où $n$ est la capacité du taxi. Là encore, les champs inutilisés contiennent $-1$. Les colonnes contiennent :

(1) l'heure de l'événement ;

(2) le type d'événement (identifié par un numéro) ;

(3) le numéro d'identification du client (si un client est aussi concerné par l'événement) ;

(4) le numéro d'identification du nœud où se trouve le taxi ;

(5) le nombre de passagers présents dans le taxi ;

les colonnes suivantes contiennent chacune le numéro d'identification d'un passager présent dans le taxi au moment de cet événement.

CHAPITRE III

# Gestion décentralisée

*Vous connaissez le chemin le plus court entre deux points ?*
*— La ligne droite. Et le plus long ? — Le taxi !*

*Patrick Timsit*

## III.1. Introduction

Dans ce chapitre, on présente plus en détail certains aspects du système de taxis collectifs dans la gestion décentralisée exclusivement. Il s'agit d'un mode de fonctionnement où les clients se présentent au bord du trottoir, plus précisément dans notre modèle aux nœuds du réseau, à la recherche d'un taxi susceptible de les prendre en charge. Par conséquent, aucune réservation préalable n'est requise.

On est amené à considérer dans notre modèle trois types d'agents :
- les clients,
- les véhicules,
- les nœuds du réseau, et plus précisément les files d'attente (clients et taxis) qui y sont rattachées.

On décrira alors tous les types d'événements pouvant impliquer un ou plusieurs types d'agents simultanément. Le traitement de ces événements d'une part modifiera l'état des agents impliqués, et d'autre part engendrera d'autres événements du même type ou d'un autre type positionnés à des dates ultérieures dans la pile des événements.

La seconde partie du chapitre sera consacrée au principal problème qu'il faut résoudre pour gérer le système, à savoir la question de l'acceptation/refus d'un client lorsque celui-ci rencontre un taxi, question indissociable de celle de la reconstruction d'un itinéraire optimisé pour le taxi en cas d'acceptation du client. Après avoir montré comment on peut formuler le problème, on discutera de quelques méthodes exactes ou approchées de résolution.

On verra que la formulation du problème fait intervenir un paramètre dont on propose de choisir la valeur par des expérimentations en simulation ce qui seront décrites au chapitre suivant. Ce choix sera adapté aux diverses conditions de demande (intensité, géométrie) ainsi qu'aux ressources mises en œuvre (nombre et capacité des taxis).

## III.2. Aperçu du fonctionnement des agents

**III.2.1. Clients.** Les clients apparaissent aux nœuds du réseau selon des processus de Poisson ; ils sont également affectés d'une destination choisie à l'aide d'une matrice O-D comme cela a été décrit au §II.5.1.2. Dès son apparition au nœud d'origine, un client souhaite trouver le plus rapidement possible un taxi pour l'amener à sa destination. Le client ne peut entrer en contact qu'avec les véhicules qui se trouveraient en attente (à vide) à ce nœud, ou bien avec ceux qui passeraient par ce nœud pendant toute la durée de son attente [1].

1. Il n'y a pas a priori dans notre modèle de distinction entre les taxis arrivant au nœud en provenance d'une direction ou d'une autre, en fait d'un arc ou d'un autre aboutissant à ce nœud, en particulier pas de distinction "entre le trottoir et le trottoir d'en face". Mais ce n'est pas une restriction du modèle : les arcs étant des voies à sens unique de circulation, un nœud placé au milieu d'un arc à sens unique ne représentera que le trottoir correspondant à ce sens de circulation.

Le comportement de l'agent "client" est donc typiquement le suivant : dès son arrivée, le client commence à chercher si il y a des véhicules en stationnement au nœud. Dans ce cas, il examine s'il peut être accepté par un de ces véhicules (a priori oui si ces véhicules sont vides). S'il n'y a pas de véhicules en stationnement, le client se placera dans une "file d'attente clients" attachée à ce nœud et on doit décider combien de temps il est prêt à attendre avant d'abandonner et de quitter le système. Cette durée d'attente maximale qui modélise sa "patience" peut être représentée par un tirage selon une loi de probabilité. La file d'attente est FIFO (first in, first out). Pendant la durée de cette attente, il est susceptible de voir passer un ou plusieurs taxis avec qui il entamera éventuellement un dialogue (du moins pour les taxis qui ne sont pas déjà à leur capacité maximale de passagers) après que tous les clients qui le précédent aient eux-mêmes discuté avec ce taxi, pour voir s'il peut être pris en charge. Si l'heure d'attente maximale est atteinte (ou dépassée après la fin d'un dialogue négatif), le client abandonne le système et il sera comptabilisé dans cette catégorie de clients non servis.

**III.2.2. Véhicules.** Un taxi ne peut donc interagir avec des clients que lorsqu'il est arrivé à un nœud. Pendant ses trajets sur les arcs, le véhicule "n'est plus sous contrôle" jusqu'à son arrivée à l'extrémité de l'arc, événement dont l'heure a été programmée au moment de son départ de l'autre extrémité de l'arc en fonction d'un temps de parcours déterminé comme cela a été indiqué au §II.5.2.1.

Le comportement typique d'un agent "taxi" est donc le suivant : lorsqu'il arrive à un nœud,

- il commence par faire débarquer les passagers arrivés à destination ;
- il examine s'il peut prendre en charge de nouveaux clients (dans la mesure ou il n'est pas complet et si son heure de fin de service n'est pas atteinte — voir plus loin) ;
- si, à l'issue de toutes ces opérations, il a un itinéraire à poursuivre, il détermine vers quel nœud suivant il doit alors se diriger ;
- s'il est vide (et si son heure de fin de service n'est pas atteinte), il doit déterminer à quel nœud il ira se mettre en stationnement ; si c'est au nœud où il se trouve, il doit déterminer l'heure limite de son attente, attente qui prendra fin si un client vient le solliciter ou bien si cette heure limite est atteinte, auquel cas la question d'un nouveau nœud de stationnement est reposée.

Chaque véhicule commence et termine son service à des heures spécifiées. L'heure de fin de service peut de ne pas être rigoureusement respectée : lorsque cette heure est atteinte mais que le véhicule n'est pas vide, il n'accepte plus de nouveaux clients mais il doit mener à destination ses passagers avant de pouvoir sortir du service.

**III.2.3. Nœuds du réseau.** Comme on l'a vu, à chaque nœud du réseau, il faut prévoir deux files d'attente FIFO : une file "clients" en attente d'un taxi pouvant les prendre en charge et une file "taxis" en attente de clients.

### III.3. Modélisation de la gestion décentralisée par la construction des événements

On va maintenant détailler les types d'événements qui affectent les clients ou les taxis, ou les deux simultanément, et dont le traitement modifie aussi l'état de ces agents et éventuellement celui des files d'attente attachées aux nœuds du réseau. On va en particulier expliciter quels nouveaux événements sont engendrés à la suite du traitement de chaque type d'événement. Comme on l'a vu, c'est l'enchaînement de ces traitements, orchestré par la gestion de la pile d'événements, qui produit finalement l'historique d'une simulation à événements discrets.

On rappelle que chaque traitement d'événement est supposé instantané, de sorte que l'état est bien défini avant et après l'activation de chaque événement, et donc aucun autre événement ne risque de se produire pendant le même laps de temps où cet événement ferait "muter" l'état du système.

Ceci dit, il n'y a pas de façon unique d'aboutir à la liste de types d'événements que nous donnons ci-dessous. Il faut essayer autant que possible d'aboutir à la liste la plus restreinte

possible, mais sans sacrifier au réalisme de la description des phénomènes élémentaires du système réel à travers leur représentation par le modèle.

Dans cette section, on décrit donc les 9 types d'événements auxquels nous avons finalement abouti pour la gestion décentralisée. La numérotation des types pourra paraître étrange mais il ne faut pas oublier que notre simulateur a aussi été prévu pour la gestion centralisée, voire mixte. Il s'agit donc ici d'un sous-ensemble des types d'événements relatifs à la gestion décentralisée. On décrit ensuite comment le traitement de chaque type engendre de nouveaux événements dans la même liste de 9 types.

Les 9 types d'événements ci-dessous sont regroupés en deux catégories même si certains concernent à la fois un client et un taxi. L'idée est de considérer l'agent déclencheur de chaque type d'événement.

### III.3.1. Événements déclenchés par les clients.

III.3.1.1. *Type 17 : apparition client.* C'est l'événement qui décrit le comportement du client lors de son apparition à son nœud d'origine. Le client cherche immédiatement à trouver un véhicule qui l'amènera à sa destination. S'il en trouve un, le dialogue commence entre le client et le véhicule (ce dialogue n'est pas instantané, il a une durée et donc l'événement "apparition" s'arrête là ; il passe la main à un événement de type 18 décrit au §III.3.1.3). Si aucun véhicule n'est trouvé immédiatement, le client décide s'il doit attendre à son nœud d'origine et pour combien de temps (ce qui conduit à la création d'un événement de type 16 — cf. §III.3.1.2).

III.3.1.2. *Type 16 : client abandonne.* C'est l'événement qui survient lorsque le client a atteint l'heure limite jusqu'à laquelle il était prêt à attendre. Cet événement peut être créé dans la pile mais ne jamais être exécuté : ce sera la cas si le client est finalement pris en charge par un taxi avant l'heure limite d'attente ; un autre cas est celui où le client est occupé dans un dialogue avec un taxi au moment où cette heure limite est atteinte. Dans ce dernier cas, le traitement de cet événement consiste seulement à modifier un indicateur d'état, ce qui permettra, au moment du traitement de fin de dialogue (cf §III.3.2.5), et si le client est finalement refusé, de le faire effectivement sortir du système.

III.3.1.3. *Type 18 : client entre dans un taxi.* À la fin d'un dialogue entre un client et un taxi, si le client est accepté, le client va entrer dans le véhicule, ce qui prend un temps non nul dans notre modèle. Cet événement marque la *fin* de l'embarquement.

### III.3.2. Événements déclenchés par les taxis.

III.3.2.1. *Type 14 : véhicule se met en service.* Le véhicule entrant en service se met à la disposition des clients en un nœud particulier (sa position initiale) ; il cherche immédiatement s'il y a des clients en attente à ce nœud. Dans ce cas, un dialogue commence entre le premier client et le taxi. Dans le cas contraire, on doit décider si le taxi va stationner à ce nœud et pour combien de temps au maximum (il entre alors dans la file d'attente des taxis à ce nœud), ou bien se diriger vers un autre nœud.

III.3.2.2. *Type 15 : véhicule termine son service.* À l'heure fixée pour la fin de son service, le taxi sort du système à condition d'être vide. Dans le cas contraire, il est "marqué" comme ne pouvant plus prendre en charge de nouveaux clients et il sortira du service dès qu'il sera vide.

III.3.2.3. *Type 11 : véhicule arrive à un nœud.* Cet événement décrit le comportement d'un taxi lorsqu'il arrive à un nœud. La première opération à faire est de chercher si il y a des passagers qui débarquent à ce nœud. Dans ce cas, un événement de type 13 (voir §III.3.2.4) est programmé (car l'opération a une durée). Sinon, le taxi cherche d'éventuels clients en attente au nœud (à condition que son heure de fin de service ne soit pas atteinte). Si c'est le cas, un dialogue commence entre le premier client de la file et le taxi ; ce dialogue ayant une durée, un événement de type 12 (voir §III.3.2.5) est programmé. Dans le cas où il n'y a pas de clients en attente, soit le taxi a encore des passagers à bord et donc il a un itinéraire planifié, et il détermine alors vers quel nœud suivant il doit se diriger, soit il est vide et il doit déterminer à quel nœud stationner sauf dans le cas où la date de fin de service est déjà atteinte ou dépassée.

III.3.2.4. *Type 13 : véhicule fait sortir des passagers.* Si le véhicule a détecté qu'il y a des clients à débarquer, il les fait sortir ; l'événement marque la fin de cette opération.

III.3.2.5. *Type 12 : véhicule termine un dialogue avec un client.* Cet événement marque la fin d'un dialogue entre un client et un taxi avec une réponse négative ou positive d'acceptation et un nouvel itinéraire défini dans le cas positif. Dans le cas d'un refus du client, si celui-ci a déjà dépassé son temps limite d'attente, on le fait sortir du système. Si le véhicule est vide et a dépassé son heure de fin de service, on le fait également sortir du système.

III.3.2.6. *Type 10 : véhicule vide quitte son stationnement.* Lorsqu'un taxi est mis en stationnement à vide, il l'est jusqu'à une heure limite. Il peut quitter ce stationnement avant cette heure limite s'il est sollicité par un client (événement de type 12 — cf. III.3.2.5), mais, dans le cas contraire, cet événement de type 10 l'oblige à se reposer la question d'un nouveau point de stationnement, sauf si son heure limite de service est atteinte ou dépassée.

**III.3.3. Enchaînement des événements.** Chaque fois qu'un événement est traité, cela modifie l'état du système, mais cela peut éventuellement engendrer la programmation d'autres événements dans la pile d'événements. La figure III.1 représente les types d'événements sous forme de nœuds d'un graphe dont les arcs sont de deux types :

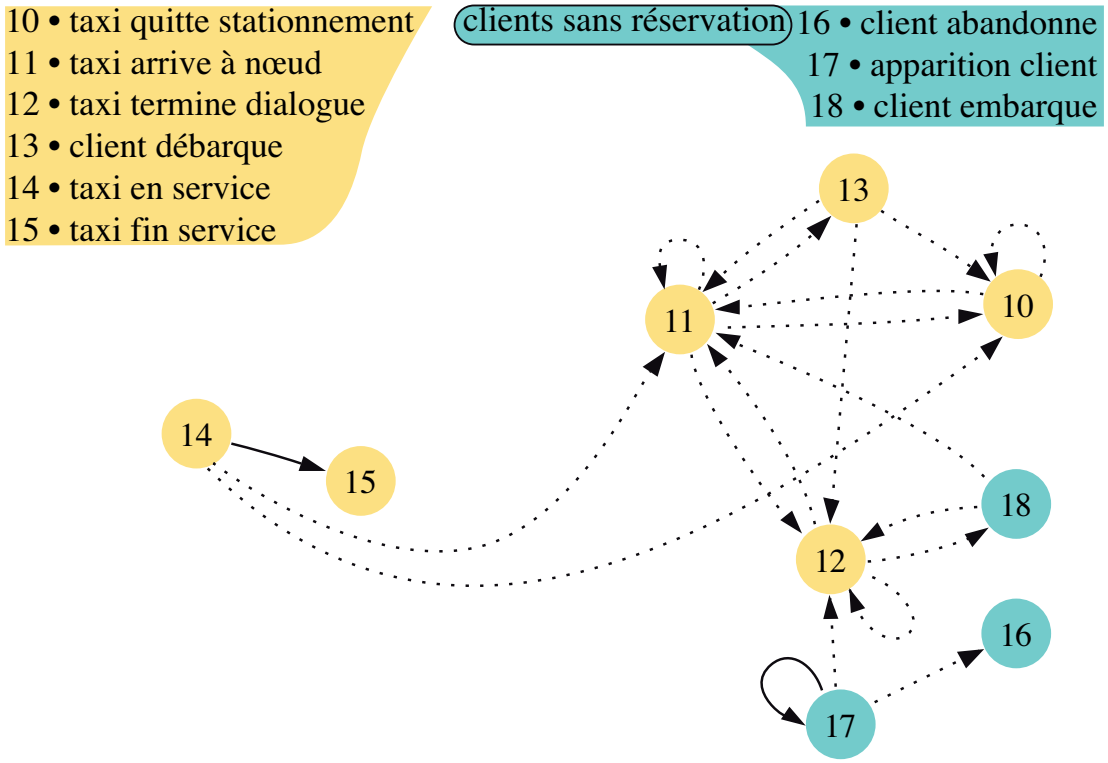


FIGURE III.1. Génération d'événements par d'autres événements (gestion décentralisée)

- les arcs en trait continu indiquent la création d'événements qui suivent *nécessairement* l'événement d'origine ;
- les arcs en pointillé indiquent une éventuelle génération d'événements : cela dépend de l'état du système et des décisions prises au moment du traitement de l'événement d'origine (par exemple l'acceptation ou le refus d'un client par un taxi).

On va maintenant détailler les éléments de cette figure.

III.3.3.1. *Type 17 : apparition client.* L'arc de 17 vers lui-même indique qu'un événement "apparition client à un nœud du réseau" engendre l'événement d'apparition du prochain client au même nœud.

– Si le client qui vient d'apparaître trouve tout de suite un taxi, comme il s'instaure un dialogue entre ce client et ce taxi, il y aura la création d'un événement de type 12 (cf. §III.3.2.5) correspondant à la fin du dialogue entre ces deux agents.
– Si aucun véhicule n'est trouvé, le client décide combien de temps au maximum il va attendre au nœud et dans ce cas un événement de type 16 (cf. §III.3.1.2) doit être généré marquant la fin d'attente du client.

III.3.3.2. *Type 16 : client abandonne.* Si cet événement se réalise effectivement, il n'engendre aucun autre événement aval. Il suffit de mettre à jour l'état du système (en l'occurrence la file d'attente au nœud correspondant du réseau). L'événement ne se réalise pas si le client a déjà été pris en charge par un taxi (il n'est plus dans la file d'attente) ou s'il est occupé dans un dialogue. Dans ce cas, un indicateur d'état signale qu'il doit sortir du système en cas de refus en fin de dialogue (traitement de l'événement 12 — voir §III.3.2.5).

III.3.3.3. *Type 18 : client entre dans un taxi.* À la fin de l'embarquement du client,
– si le véhicule peut dialoguer avec un autre client, il y aura la génération d'un événement de type 12 (cf. §III.3.2.5) pour programmer la fin de ce nouveau dialogue ;
– si aucun nouveau dialogue n'est possible, le taxi part du nœud et un événement de type 11 (cf. §III.3.2.3) sera créé pour indiquer l'arrivée du véhicule au prochain nœud de son trajet.

Notons que dans le second cas, le taxi quitte effectivement toujours le nœud car il n'est pas vide puisqu'il sort d'un événement de type 18.

III.3.3.4. *Type 14 : véhicule se met en service.* Dès qu'un véhicule entre en service, on a déjà décidé de sa durée de service, et on génère donc un événement de type 15 (cf. §III.3.2.2) afin de prévoir la sortie du véhicule du système.
– Si le véhicule trouve un client en attente à sa position initiale, il y aura un dialogue et la création d'un événement de type 12 (cf. §III.3.2.5) indiquant la fin de ce dialogue.
– Dans le cas contraire, le véhicule se demande à quel nœud il va stationner.
  – Si ce nœud est sa position actuelle, on doit créer un événement marquant la fin du stationnement du véhicule à ce nœud (événement de type 10 — cf. §III.3.2.6).
  – Si un autre nœud de stationnement que la position actuelle du véhicule a été décidé, alors le véhicule quitte le nœud immédiatement et un événement d'arrivée du véhicule au nœud suivant de son trajet sera généré (événement de type 11 — cf. §III.3.2.3).

III.3.3.5. *Type 15 : véhicule termine son service.* Si cet événement se produit, le véhicule n'est plus à la disposition de nouveaux clients. Cet événement ne crée aucun autre événement. L'événement ne peut pas se produire si le taxi est encore occupé à une tâche. Dans ce cas, un indicateur d'état signalera qu'il ne peut plus accepter de nouvelles tâches et qu'il doit sortir du système dès qu'il est libre.

III.3.3.6. *Type 11 : véhicule arrive à un nœud.* Lorsque le véhicule arrive à un nœud, il effectue les actions suivantes.

(1) Il détecte d'abord s'il a des passagers qui descendent à ce nœud. Dans ce cas, un événement de type 13 (cf. §III.3.2.4) sera créé (pour marquer la fin du débarquement de l'ensemble des passagers avec une durée de l'opération proportionnelle au nombre de passagers concernés). Sinon il passe à l'étape suivante.

(2) Pour qu'un dialogue puisse s'établir avec un client, il faut réunir plusieurs conditions :
  – il faut qu'il y ait au moins un client présent dans la file d'attente à ce nœud ; que ce client n'ait pas déjà été examiné par ce taxi ; qu'il ne soit pas occupé dans un dialogue avec un autre taxi ;
  – il faut qu'il reste au moins une place libre à bord du taxi ;
  – il faut que l'heure limite de son temps de service ne soit pas dépassée.

  Si toutes ces conditions sont réunies, un événement de type 12 (cf. §III.3.2.5) est créé pour marquer la fin du dialogue entre le client candidat et le taxi.

(3) Si l'une de ces conditions n'est pas satisfaite, le taxi peut quitter le nœud si l'*une* des conditions suivantes est satisfaite :
   - l'itinéraire du taxi n'est pas vide (il a des passagers à bord) ;
   - ou bien, le taxi est vide *et* son heure limite de service n'est pas atteinte *et* il a choisi de stationner à un autre nœud que celui de sa position actuelle.

   Dans ces deux cas, un événement de type 11 (cf. §III.3.2.3) est créé pour marquer l'arrivée au prochain nœud de son trajet.

(4) Si le taxi est vide *et* s'il n'y a pas de client dans la file d'attente *et* si son heure limite de service n'est pas atteinte *et* s'il décide de stationner dans sa position actuelle, un événement de type 10 (cf. §III.3.2.6) est créé pour marquer l'heure à laquelle le taxi devra reconsidérer ce stationnement.

(5) Si le taxi est vide *et* si son heure limite est atteinte ou dépassée, le taxi est mis hors service.

III.3.3.7. *Type 13 : véhicule fait sortir des passagers.* Cet événement marque la fin du débarquement de tous les passagers arrivés à destination. Une fois la mise à jour de l'état du système effectuée, il suffit de reprendre le traitement ci-dessus de l'événement de type 11 à partir du point (2).

III.3.3.8. *Type 12 : véhicule termine un dialogue avec un client.* À l'issue de ce dialogue,

- si le client a été accepté, un événement de type 18 (cf. §III.3.1.3) sera créé pour marquer la fin de l'entrée du client dans le taxi ;
- si le client a été refusé, et si son heure limite d'attente est dépassée, on le fait sortir du système. S'il a été refusé, c'est donc que le taxi n'est *pas vide*, et qu'il n'est *pas plein* non plus (sinon le dialogue n'aurait pas eu lieu) :
  - si le taxi est en mesure de dialoguer avec un autre client, il y aura création d'un nouvel événement de type 12 (cf. §III.3.2.5) pour marquer la nouvelle fin de dialogue ;
  - sinon, le véhicule doit partir (puisqu'il n'est pas vide) et donc un événement de type 11 (cf. §III.3.2.3) sera créé pour marquer l'arrivée du véhicule au prochain nœud.

III.3.3.9. *Type 10 : véhicule vide quitte son stationnement.* Lorsque l'heure limite de stationnement est atteinte et que le véhicule est toujours là (donc vide), si son heure limite de service est atteinte ou dépassée, le taxi sort du système. Dans le cas contraire, il faut se demander si le véhicule va se remettre en stationnement au même nœud pour une nouvelle durée, ou bien aller chercher un stationnement ailleurs.

- Dans le premier cas, il faut recréer un événement de type 10.
- Si on choisit un autre nœud de stationnement, le véhicule part et un événement de type 11 (cf. §III.3.2.3) est créé pour programmer l'arrivée du véhicule à son premier nœud de passage.

## III.4. Gestion temps réel dans le mode décentralisé

L'analyse précédente des événements nous a montré qu'il y a deux décisions essentielles à considérer dans la gestion temps réel du système.

- L'une concerne la stratégie de choix d'un nœud de stationnement pour les taxis vides, ainsi que la durée limite de ce stationnement avant de devoir éventuellement se reposer la même question.
- L'autre concerne l'acceptation ou le refus d'un client à bord d'un taxi lors de leur rencontre.

Dans cette section, nous nous pencherons essentiellement sur le second problème qui est de loin le plus important. Nous ne dirons que quelques mots du premier en évoquant quelques idées simples qui ont été testées au chapitre IV, ce qui a montré cependant que des stratégies trop simplistes peuvent conduire à des dysfonctionnements importants.

Sur la question de l'acceptation/refus d'un nouveau client, il paraît assez évident que cette décision binaire ne peut être prise sans essayer de reconstruire un itinéraire "optimal" du taxi

en incluant la destination du candidat dans la liste des nœuds à desservir du fait de la présence d'autres passagers déjà à bord. Une décision d'acceptation aura de toutes façons une conséquence en termes de détours supplémentaires pour certains de ces passagers et pour le candidat lui-même. Ces détours doivent être appréciés par rapport à la durée du trajet direct que tout passager aurait effectué s'il était seul à bord du taxi. On proposera donc une formulation du problème en termes d'optimisation d'un critère naturel sous des contraintes de détours qui ne devront pas dépasser un certain seuil, au moins de façon prévisionnelle, car la décision ne peut être basée que sur la connaissance du passé et sur une anticipation statistique de l'avenir.

Ce seuil est un paramètre de la formulation proposée. Un seuil très sévère devrait aboutir à des trajets assez directs en moyenne pour tous les clients pris en charge, mais il risque par contre de provoquer de nombreux rejets de candidats et donc des attentes plus importantes avant le départ, voire un taux d'abandon élevé après une attente excessive. Ce sera l'objet de la méthodologie développée au chapitre IV de montrer, par des simulations répétées, comment ce seuil peut être fixé à des valeurs raisonnables, et en tout cas de quantifier les conséquences de ce choix.

Pour en revenir à la formulation de ce problème d'acceptation/rejet et à sa résolution algorithmique, on envisagera des méthodes exactes et approchées et on discutera de leur complexité. Des comparaisons seront effectuées au chapitre IV.

**III.4.1. Choix d'un nœud de stationnement.** Le problème se pose pour un taxi qui se trouverait à vide après avoir débarqué ses passagers à un nœud sans trouver d'autres clients en attente à ce nœud. Le taxi peut décider de stationner à ce même nœud (c'est-à-dire se placer dans la file d'attente des taxis attachée à ce nœud pour un certain temps maximum) ou bien décider de se diriger vers un autre nœud de stationnement (avec la possibilité de rencontrer des clients en chemin). Dans la gestion décentralisée (donc sans dispatching central susceptible d'avoir une vision globale de l'état du système), on ne peut s'appuyer pour prendre cette décision que sur les informations observables par le taxi au nœud où il se trouve (notamment la longueur de la file d'attente des taxis à ce nœud) et/ou sur des informations d'ordre statistique comme la connaissance des nœuds statistiquement à forte demande.

Pour les expériences rapportées au chapitre IV, on s'est limité à la seconde option, mais on doit aussi prendre en compte, en plus de l'attractivité de chaque nœud du réseau du fait de son niveau moyen de demande, sa distance à la position actuelle du taxi. On ne veut pas effectivement, pour rejoindre un nœud de stationnement de forte demande, avoir à parcourir à vide une distance trop grande.

On construit donc une matrice de type "matrice O-D" dont chaque ligne $i$ est associée à un nœud $i$ du réseau représentant la position actuelle du taxi vide et chaque colonne $j$ représente l'attractivité d'un nœud $j$ pour aller y stationner (son "poids" $p_{ij}$) lorsqu'on se trouve en $i$. La somme des poids dans chaque ligne peut accessoirement être normalisée à 1 comme dans une matrice O-D. Le poids $p_{ij}$ doit prendre en compte, de façon *croissante*, le coefficient du processus de Poisson $\lambda_j$ représentant l'intensité de la demande au nœud $j$ (voir §II.5.1.2) et, de façon *décroissante*, la distance $d_{ij}$ (en temps de trajet direct moyen) séparant $i$ de $j$. Nous avons utilisé la formule suivante :

$$p_{ij} = \lambda_j e^{-\alpha d_{ij}} \tag{III.1}$$

où $\alpha$ est un coefficient positif que l'on peut ajuster pour donner plus ou moins d'importance relative à l'un des deux critères (intensité de la demande et distance). On peut par exemple ajuster $\alpha$ itérativement pour obtenir que l'élément diagonal dans chaque ligne (après normalisation de la la somme en ligne, cela représente en quelque sorte la probabilité de rester sur place) soit supérieur à $0,3$ dans au moins la moitié des lignes.

Une fois cette matrice construite, on peut l'utiliser de diverses façons, par exemple :

– en tirant au hasard le nœud $j$ de stationnement lorsqu'on se trouve en $i$ avec une probabilité proportionnelle à $p_{ij}$ : c'est finalement ce que nous avons fait ;

– en choisissant systématiquement le nœud $j$ qui a le poids $p_{ij}$ maximum : c'est ce que nous avions fait dans un premier temps, mais qui a conduit à des dysfonctionnements qui seront rapportés au chapitre IV et que les simulations ont permis de détecter, et donc de corriger.

Quant à la durée maximale du stationnement, nous avons choisi une durée fixe comme ce sera rapporté au chapitre IV.

**III.4.2. Acceptation/refus de clients.** On va formuler ce problème sous la forme d'un problème d'optimisation sous contraintes. S'il n'existe aucune solution admissible à ce problème, le client sera refusé. Sinon, la solution du problème d'optimisation définira le nouvel itinéraire prévisionnel du taxi une fois le client accepté. Tous les calculs de parcours *prévisionnels* sont faits sur la base des temps moyens de parcours par le chemin le plus court (matrice $\delta$ définie ci-dessous). Par contre, jusqu'à la date présente, on prend en compte les durées *réelles* des parcours effectués (les temps de parcours étant aléatoires comme on l'a indiqué au §II.5.2.1).

III.4.2.1. *Notations.* On définit les notations suivantes :

$t_0$ **:** date actuelle (date de la rencontre) ;

$n_0$ **:** nœud de la position actuelle du taxi ;

$n_i^o$ **:** nœud d'origine du passager $i$ ;

$n_i^d$ **:** nœud de destination du passager $i$ ;

$c$ **:** indice du client candidat ;

$n_c^d$ **:** nœud de destination du candidat (son nœud d'origine est évidemment $n_0$) ;

$L = \{n_1, n_2, \ldots, n_{M'}\}$ **:** liste *ordonnée* des nœuds de passage (destinations à desservir) de l'itinéraire *futur* (après $n_0$) du véhicule *avant acceptation du candidat* ;

$\ell = L \cup \{n_c^d\}$ **:** liste *non ordonnée* des destinations à desservir (après $n_0$) si le candidat est accepté ; l'un des objets de l'algorithme est d'*ordonner* $\ell$ si le candidat est accepté ; [2]

$J$ **:** ensemble des indices des éléments de $\ell$ ; autrement dit, $J = \{1, \ldots, M\}$ où $M$ est le cardinal de $\ell$ (selon que $n_c^d$ appartient ou n'appartient pas à $L$, $M = M'$ ou $M' + 1$) ;

$I$**:** ensemble des indices des passagers du véhicule *et* du candidat $c$ ($I = I \cup \{c\}$) ;

$d : I \to J$ **:** application qui sélectionne la destination des clients, c'est-à-dire que $\forall i \in I, n_i^d = n_{d(i)} \in \ell$ ;

$p(j), j \in J$ **:** nombre de passagers qui descendent au nœud $j$ ; autrement dit, $p(j)$ est le cardinal de l'ensemble $d^{-1}(j)$ ;

$\delta(a, b)$ **:** durée du trajet direct pour aller du nœud $a$ au nœud $b$ dans le graphe avec les temps de parcours moyens ; cette matrice des plus courtes distances en *temps* entre paires de nœuds du graphe évaluées avec les temps de parcours moyens des arcs est précalculée et entre comme une donnée de la simulation comme indiqué au §II.5.2.1 ;

$t_i^o$ **:** date à laquelle le passager $i$ est entré dans le taxi (pour $i = c$, on a évidemment $t_c^o = t_0$) ;

$t_j^p$ **:** date prévisionnelle d'arrivée à la destination $n_j \in L$ *avant* que le candidat ne soit accepté :

$$t_j^p = t_0 + \delta(n_0, n_1) + \sum_{\substack{n_k \in L \\ k=1,\ldots,j-1}} \delta(n_k, n_{k+1}) \ ;$$

$s$ **:** seuil de détour acceptable par rapport au trajet direct ; l'un des buts des simulations est d'optimiser ensuite ce paramètre $s$ pour obtenir le meilleur compromis entre temps d'attente initial des candidats — ou taux de rejet de ceux-ci — et la durée moyenne de leur trajet effectif rapportée au trajet direct une fois qu'ils sont acceptés ;

2. Étant donné que les nœuds à desservir sont tous des destinations de passagers ou du candidat, il n'y a aucun intérêt à ne pas débarquer ces passagers dès le premier passage par leur destination, et donc de faire figurer plusieurs fois une même destination dans la liste $\ell$, ce qui justifie l'opération d'union.

$t_j^{\lim}$ **:** heure limite d'arrivée à la destination $n_j \in \ell$ : cette heure limite doit traduire le seuil de détour acceptable pour *tous* les passagers qui descendent au nœud $j$ ; mais elle ne peut être inférieure à l'heure prévisionnelle d'arrivée à $n_j$ compte tenu de la situation actuelle, *même si le candidat n'est pas accepté dans le taxi* : en effet, les temps de parcours étant aléatoires, il est possible que la contrainte de détour pour certains passagers soit *déjà* compromise ; il ne faut pas que l'acceptation d'un nouveau client aggrave cette situation, mais il ne faut pas non plus que le nouveau client soit rejeté pour une violation de contrainte qui ne lui est pas imputable. On adopte donc la formule suivante :

$$t^{\lim}(j) = \begin{cases} \max\left(t_j^p, \min\limits_{i \in d^{-1}(j)} \left(t_i^o + s \times \delta(n_i^o, n_i^d)\right)\right) & \text{si } n_j \in L, \\ t_0 + s \times \delta(n_0, n_c^d) & \text{si } n_j = n_c^d \text{ et si } n_c^d \notin L\ ; \end{cases} \tag{III.2}$$

Les notations précédentes permettent de définir les données du problème. On va maintenant introduire les notations permettant d'écrire une formulation du problème sous forme d'un programme dynamique. Le problème consiste essentiellement à trouver un ordre de parcours de la liste $\ell$ tel que la contrainte d'heure limite soit respectée à chaque étape de l'itinéraire, et que le critère constitué par la somme de ces dates de passage à toutes les étapes du parcours (pondérées par le nombre de clients descendant à chacune de ces étapes) soit minimisée. Si aucune solution admissible n'est trouvée, le candidat doit être rejeté. Sinon, on obtient une liste ordonnée de l'ensemble $\ell$.

Le parcours de $\ell$ nécessite $M$ mouvements entre les $M+1$ nœuds de l'ensemble $\ell \cup \{n_0\}$ : ces étapes seront indexées par un indice $k$ allant de 0 à $M$. On note

$x(k), k = 1, \ldots, M$ **:** la suite ordonnée des nœuds de $\ell$ successivement visités ; on complète cette définition par $x(0) = n_0$ ;

$u(k), k = 0, \ldots, M-1$ **:** le choix du prochain nœud à visiter une fois qu'on a atteint $x(k)$ ; par conséquent, $x(k+1) = u(k)$ ;

$\mathcal{E}(k)$ **:** l'ensemble des nœuds déjà parcourus à l'étape $k$ ; on considère que $\mathcal{E}(0) = \emptyset$ ; évidemment, $\mathcal{E}(k+1) = \mathcal{E}(k) \cup \{u(k)\}$ ; pour que la totalité de la liste ait été parcourue à la dernière étape, il faut et il suffit que, $\forall k$, $u(k) \in \overline{\mathcal{E}(k)}$ (complémentaire de l'ensemble $\mathcal{E}(k)$ dans $\ell$) ;

$t(k)$ **:** l'heure à laquelle on atteint $x(k)$ ; $t(0) = t_0$ et $t(k+1) = t(k) + \delta(x(k), u(k))$.

III.4.2.2. *Formulation du problème.* On cherche alors à résoudre le problème d'optimisation suivant :

$$\min_{u(\cdot)} \sum_{k=1}^{M} p\big(x(k)\big)\, t(k) \tag{III.3}$$

$$\text{sous} \quad x(k+1) = u(k), \quad k = 0, \ldots, M-1, \quad x(0) = n_0, \tag{III.4}$$

$$\mathcal{E}(k+1) = \mathcal{E}(k) \cup \big\{u(k)\big\}, \quad k = 0, \ldots, M-1, \quad \mathcal{E}(0) = \emptyset, \tag{III.5}$$

$$t(k+1) = t(k) + \delta(x(k), u(k)), \quad k = 0, \ldots, M-1, \quad t(0) = t_0, \tag{III.6}$$

$$u(k) \in \overline{\mathcal{E}(k)}, \quad k = 0, \ldots, M-1, \tag{III.7}$$

$$t(k) \le t^{\lim}(x(k)), \quad k = 1, \ldots, M, \tag{III.8}$$

où $t^{\lim}$ est précalculable indépendamment de la solution du problème par la formule (III.2).

Dans cette formulation, l'évaluation des temps de passage à chaque étape de l'itinéraire (III.6) néglige la durée des opérations de débarquement des passagers (durée non nulle dans la simulation), ainsi d'ailleurs que la durée du dialogue initial avec le candidat et la durée de l'embarquement (éventuel) du candidat (durées également non nulles dans la simulation). Cependant, ces durées sont réputées relativement petites (et possiblement aléatoires) et bien d'autres

facteurs viennent encore perturber l'évaluation théorique des dates de passage prévisionnelles aux étapes futures du parcours. En effet, les temps de parcours des arcs sont eux-mêmes aléatoires dans la simulation, et les futurs candidats rencontrés sur le parcours (induisant autant de durées de dialogue supplémentaires au minimum) ne peuvent évidemment pas être pris en compte. Pour cette raison, la formulation du problème de décision ci-dessus avec une contrainte de non dépassement du seuil $s$ pour le détour est juste une façon d'aboutir à une décision, mais il faut s'attendre au dépassement aléatoire de ce seuil $s$ qui n'est qu'un paramètre à optimiser. C'est le rôle des simulations de nous aider à ajuster $s$ en tenant compte de ces aléas.

On discute maintenant de plusieurs méthodes de résolution exacte ou approchée de ce problème.

III.4.2.3. *Résolution par la programmation dynamique.* On s'inspire ici de l'article [**38**] qui discute de problèmes analogues [3]. Cependant, contrairement à cette référence, on adopte ici le point de vue en temps rétrograde plus classique dans le raisonnement de programmation dynamique.

La formulation (III.3)–(III.8) est celle d'un programme dynamique ayant pour état le triplet $(x, \mathcal{E}, t)$. On introduit la fonction de Bellman $V_k(x, \mathcal{E}, t)$ représentative du coût optimal partant de l'étape $k$ jusqu'à $M$ avec l'état initial $(x, \mathcal{E}, t)$.

Pour $k = M$,

$$V_M(x, \mathcal{E}, t) = \begin{cases} 0 & \text{si } \mathcal{E} = \ell, \\ +\infty & \text{sinon.} \end{cases}$$

La récurrence rétrograde classique de la programmation dynamique s'écrit alors :

$$V_k(x, \mathcal{E}, t) = \min_{\substack{u \in \overline{\mathcal{E}} \\ t+\delta(x,u) \le t^{\lim}(u)}} \Big( p(u)\,\big(t + \delta(x,u)\big) + V_{k+1}\big(u, \mathcal{E} \cup \{u\}, t + \delta(x,u)\big) \Big), \quad k = 0, \dots, M-1.$$

Si $V(n_0, \emptyset, t_0) = +\infty$, c'est qu'il est impossible de satisfaire toutes les contraintes (III.8) et le candidat doit être refusé. Sinon, le candidat est accepté et la suite des $\arg\min_u$ (c'est-à-dire la trajectoire optimale de $x$) détermine la solution, c'est-à-dire l'ordre dans lequel il faut parcourir la liste $\ell$.

Étudions rapidement la complexité de mise en œuvre de cette méthode. La résolution de cette équation de la programmation dynamique doit se faire dans l'espace produit des valeurs prises par $x, \mathcal{E}$ et $t$. En ce qui concerne $t$, c'est une quantité certainement comprise dans le segment $[t_0, \max_{x \in \ell} t^{\lim}(x)]$. La variable $x$ prend ses valeurs dans $\ell$ (de cardinal $M$) sauf pour le calcul de $V_0$ pour lequel seule la valeur $x = n_0 \notin \ell$ nous intéresse. Quant à $\mathcal{E}$, les valeurs prises sont a priori tous les sous-ensembles de $\ell$ donc un ensemble de cardinal $2^M$. Cependant, il y a moyen de réduire considérablement l'exploration de cette partie de l'espace d'état. En effet, à l'étape $k$, les $\mathcal{E}(k)$ atteignables sont forcément de cardinal $k$ : il y a donc seulement $\mathsf{C}_M^k$ valeurs possibles (nombre de combinaisons de $k$ éléments parmi $M$).

Finalement, la variable d'état la plus pénalisante est la variable $t$ dont le domaine de variation, a priori seulement borné par $[t_0, \max_{x \in \ell} t^{\lim}(x)]$, doit être discrétisé. Notre unité de temps dans la simulation est la seconde mais les parcours dans le graphe sont de l'ordre de plusieurs minutes, dizaines de minutes, voire de l'ordre de l'heure. Nous ne sommes pas obligés de discrétiser le domaine de variation de $t$ par pas de temps d'une seconde bien sûr, mais tout pas de discrétisation plus grossier introduit une approximation numérique dans le calcul de la fonction de Bellman et dans le respect des contraintes (III.8).

Dans [**38**], il apparaît que l'état peut être réduit au couple $(x, \mathcal{E})$ si le critère qui est retenu à la place de (III.3) est simplement $t(M)$, c'est-à-dire la date du *dernier* client desservi. Dans ce cas, c'est la fonction de Bellman $V$ elle-même qui donne les temps de passage aux nœuds intermédiaires (si ces temps respectent les contraintes (III.8)) à condition de procéder, pour

3. Nous sommes reconnaissante à Frédéric Meunier (LVMT-ENPC) de nous voir signalé cette référence et de ses discussions éclairantes sur ce sujet.

l'équation de la programmation dynamique, dans le sens direct plutôt que rétrograde, et en initialisant $V_0(n_0, \emptyset)$ à $t_0$. Cependant, ce critère $t(M)$ semble moins satisfaisant que celui que nous avons retenu en (III.3).

III.4.2.4. *Résolution par énumération exhaustive.* Puisqu'il s'agit en définitive d'ordonner la liste $\ell$ en vérifiant les contraintes (III.8), ce qui implique, pour chaque ordre envisagé, de propager les dates prévisionnelles selon (III.6) et d'évaluer le critère (III.3), on peut se demander si une brutale exploration de tous les ordonnancements possibles des $M$ destinations à desservir n'est pas envisageable, et plus facile à programmer. Il y a $M!$ ordres possibles : pour des capacités de véhicules de l'ordre de 5 passagers, $M!$ est majoré (mais souvent plus petit) que $5! = 120$. Pour chaque ordre envisagé, on peut commencer le calcul de (III.3) et (III.6) récursivement pour $k$ croissant, mais l'interrompre à la valeur de $k$ pour laquelle la contrainte (III.8) n'est plus satisfaite ou lorsque le coût est devenu plus grand qu'un coût déjà évalué pour un autre ordre, ce qui réduit encore le temps de calcul.

**Algorithme.**

(1) Commencer par calculer les $t^{\lim}(j)$ selon la formule (III.2).

(2) Créer les listes ordonnées $\ell_m$ pour $m \in \{1, \ldots, M!\}$ avec tous les ordres possibles pour les nœuds de l'itinéraire. Poser $c^* = +\infty$ et $m^* = 0$.

(3) Pour chaque valeur de $m$,

  (a) poser $x(0) = n_0$ (le nœud actuel) et $c(0) = 0$.

  (b) pour $k = 0, \ldots, M-1$,

    (i) prendre pour $x(k+1)$ le $(k+1)^{\text{ième}}$ élément de $\ell_m$ ;

    (ii) calculer (III.6) ;

    (iii) si (III.8) est vérifiée, calculer aussi $c(k+1) = c(k) + p\big(x(k+1)\big)\, t(k+1)$ ;

      (A) si $c(k+1) < c^*$, passer à la prochaine valeur de $k$ ;

      (B) sinon, passer à la prochaine valeur de $m$ ;

    (iv) si (III.8) n'est pas vérifiée, passer à la prochaine valeur de $m$.

  (c) Fin de la boucle sur $k$ : si $c(M) < c^*$, remplacer $c^*$ par $c(M)$ et $m^*$ par la valeur courante de $m$.

(4) Fin de la boucle sur $m$ : si $m^* = 0$, le client est rejeté. Sinon, le client est accepté et l'ordre optimal est celui de la liste $\ell_{m^*}$.

On observe que, pour chaque valeur de $m$ (chaque permutation), la boucle sur $k$ est interrompue soit en (B) (cas $c(k+1) \geq c^*$) soit en (iv) (cas où (III.8) n'est pas vérifiée). La première interruption en (B) à d'autant plus de chances de se produire que l'estimation $c^*$ du coût optimal est bonne (plus près de l'optimum). Ceci dépend du fait de savoir si, en explorant toutes les permutations de la liste $\ell$, on a plutôt commencer par des "bonnes" ou des "mauvaises" permutations. On aurait donc intérêt, pour accélérer l'algorithme, à essayer de commencer par les "bonnes" permutations. Voici donc une stratégie pour essayer d'atteindre cet objectif.

On considère les $M$ distances $\delta(n_0, n_i)$, $i = 1, \ldots, M$, et on classe les éléments de $\ell$ selon l'ordre *croissant* de ces distances. On appelle cet ordre initial, l'ordre *primitif*. On considère alors les $M!$ permutations possibles de $\ell$ (numérotées par l'indice $m$) dans un ordre où les nœuds classés en premier dans l'ordre primitif restent classés le plus longtemps possible dans les premiers rangs. Voici un exemple pour $M = 4$, l'idée étant de permuter d'abord les deux premiers éléments, puis les trois premiers, etc. On espère ainsi trouver la permutation optimale

dans les premières visitées.

| $m$ | $\ell_m$ | $m$ | $\ell_m$ | $m$ | $\ell_m$ | $m$ | $\ell_m$ |
|---|---|---|---|---|---|---|---|
| 1 | $(1,2,3,4)$ | 7 | $(1,2,4,3)$ | 13 | $(1,3,4,2)$ | 19 | $(2,3,4,1)$ |
| 2 | $(2,1,3,4)$ | 8 | $(2,1,4,3)$ | 14 | $(3,1,4,2)$ | 20 | $(3,2,4,1)$ |
| 3 | $(1,3,2,4)$ | 9 | $(1,4,2,3)$ | 15 | $(1,4,3,2)$ | 21 | $(2,4,3,1)$ |
| 4 | $(3,1,2,4)$ | 10 | $(4,1,2,3)$ | 16 | $(4,1,3,2)$ | 22 | $(4,2,3,1)$ |
| 5 | $(2,3,1,4)$ | 11 | $(2,4,1,3)$ | 17 | $(3,4,1,2)$ | 23 | $(3,4,2,1)$ |
| 6 | $(3,2,1,4)$ | 12 | $(4,2,1,3)$ | 18 | $(4,3,1,2)$ | 24 | $(4,3,2,1)$ |

Le surcoût de cette heuristique est de classer d'abord la liste $\ell$ en fonction des distances de ces nœuds aux nœud $n_0$.

Cette résolution par énumération exhaustive est celle que nous avons utilisée lorsque la capacité des taxis était de 5 passagers. Il a fallu y renoncer lorsqu'on a expérimenté avec des taxis de capacité 7 en raison de temps de calcul trop longs ($5! = 120$ mais $7! = 5040$). On peut alors envisager une exploration limitée de l'espace des solutions qui consiste à ne pas modifier l'ordre de la liste $L$, celle qui représente l'itinéraire du taxi *avant* que le nouveau client ne soit accepté, mais à chercher seulement à insérer le nœud $n_c^d$ de destination de ce candidat entre deux étapes successives de $L$. C'est cet algorithme sous-optimal que nous décrivons maintenant.

III.4.2.5. *Algorithme sous-optimal sans changement de l'ordre des passagers.* Lorsque le taxi rencontre un nouveau candidat au nœud $n_0$ avec pour destination $n_c^d$, s'il n'a pas de passager à bord, il doit naturellement accepter ce client et son itinéraire planifié devient le plus court chemin pour aller de $n_0$ à $n_c^d$. S'il a déjà des passagers à bord, il a alors un itinéraire planifié représenté par une liste ordonnée $L$ à $M'$ éléments. Il s'agit ici de chercher à insérer $n_c^d$ soit entre $n_0$ et le premier nœud $n_1$ de $L$, soit entre deux nœuds successifs de $L$, soit après le dernier nœud $n_{M'}$ de $L$, et de retenir la meilleure solution au sens du critère (III.3) parmi les solutions satisfaisant les contraintes (III.8), ou de rejeter le candidat si aucune solution ne satisfait ces contraintes.

Le temps de calcul sera a priori proportionnel à $M' + 1$. Mais on peut faire plusieurs observations conduisant à des économies supplémentaires de temps de calcul.

(1) Si $n_c^d$ appartient déjà à $L$, il n'y a qu'une seule position possible d'insertion dans l'itinéraire. Les calculs à faire dans ce cas sont décrits ci-dessous.

(2) Si $u \in \{1, \ldots, M' + 1\}$ est la position d'insertion de $n_c^d$ dans $L$ et $t(u)$ le temps prévisionnel correspondant de passage à ce nœud (calculé selon (III.6)), cette fonction $t(u)$ est évidemment monotone croissante. Cette remarque va permettre d'interrompre prématurément l'exploration des solutions dans certains cas.

(3) Tout passager $i$, ayant selon les notations précédemment définies, un nœud de destination $n_i^d \in L$, avec $n_i^d = n_{d(i)}$, et tel que $d(i) < u$, c'est-à-dire dont la destination sera desservie *avant* celle du candidat, ne subira aucun détour supplémentaire du fait de l'acceptation du candidat puisque la date prévisionnelle d'arrivée au nœud $n_i^d$ reste la même dans ce cas (l'itinéraire n'est pas modifié *avant* l'étape insérée).

Dans le cas (1), la seule chose à décider est l'acceptation ou le refus du candidat et donc l'admissibilité de l'acceptation. Cette admissibilité ne dépend que de la satisfaction de la contrainte (III.8) pour l'indice $k$ correspondant au nœud $n_c^d$ qui fait déjà partie de la liste $L$. En effet, les contraintes (III.8) pour des autres $k$ sont certainement satisfaites même en cas d'acceptation du candidat car l'itinéraire du taxi est inchangé par l'acceptation du candidat. Pour vérifier (III.8) pour l'indice $k$ en question, il faut juste calculer $t^{\lim}(x(k))$ et $t(k)$. En cas d'acceptation du candidat (contrainte satisfaite), il faut alors compléter l'ensemble $I$ (le client est inséré dans $I$ après le ou les autres clients qui descendent au même nœud $n_c^d$) et la définition de l'application $d$ (qui désigne les nœuds de destination des clients de $I$ dans $J$) doit aussi être mise à jour avec $(d(c) = k)$.

Si on n'est pas dans le cas où $n_c^d \in L$, et si $u$ désigne, comme à l'observation (2) ci-dessus, la position d'insertion du nœud $n_c^d$ par rapport à la liste $L$, on a par définition $x(u) = n_c^d$ et la valeur de $t^{\lim}(x(u))$ est donnée par la deuxième ligne dans (III.2). La conséquence de l'observation (2) ci-dessus est que, si la contrainte (III.8) est violée pour une valeur de $k = u$ (c'est-à-dire lorsque le candidat est placé en position $u$), elle le sera aussi pour toutes les valeurs de $u$ plus grandes. Si donc on a commencé à examiner ces valeurs de $u$ dans l'ordre *croissant*, dès que (III.8) est violée pour une certaine valeur, on peut interrompre l'algorithme et ne pas examiner les valeurs suivantes de $u$.

La conséquence de l'observation (3) ci-dessus est que, pour chaque valeur de $u$ examinée, seules les contraintes (III.8) pour $k \geq u$ ont besoin d'être vérifiées.

Toutes ces remarques conduisent à dérouler l'algorithme ci-dessous en examinant les valeurs de $u$ dans l'ordre croissant.

**Algorithme.** On suppose $L$ non vide (sinon la solution est triviale). Commencer par tester si $n_c^d \in L$.

($\alpha$) **Si oui** :

(1) calculer $t(k)$ pour le $k$ correspondant au nœud candidat ainsi que $t^{\lim}\big(x(k)\big)$ ;

(2) vérifier la contrainte (III.8) :

(a) si elle est satisfaite, on accepte le candidat dans cette position $k$ telle que $n_k = n_c^d$ ; STOP.

(b) sinon, on refuse le candidat ; STOP.

($\beta$) **Sinon** :

(1) Calculer les $t^{\lim}(j)$ selon la formule (III.2) ;

(2) poser $c^* = +\infty$ et $u^* = 0$ ;

(3) pour $u = 1, \ldots, M' + 1$ : constituer les listes ordonnées $\ell_u$ en insérant $n_c^d$ dans la liste ordonnée $L$ *avant* le nœud $n_u$ de la liste $L$ (pour $u = M = M' + 1$, on obtient $\ell_M = \{L, n_c^d\}$) ;

(4) pour chaque liste $\ell_u$ en commençant par $u = 1$ : propager la récurrence (III.6) jusqu'à $t(u)$ et tester si (III.8) est satisfaite pour $t(u)$ ;

(a) si elle n'est pas satisfaite :

(i) si $u^* = 0$, le client est rejeté ; STOP ;

(ii) sinon le client est accepté et le nouvel itinéraire est donné par $\ell_{u^*}$ ; STOP ;

(b) si elle est satisfaite, calculer la somme $c(u) = \sum_{k=1}^{u} p\big(x(k)\big)t(k)$ :

(i) si $c(u) > c^*$, passer à la valeur suivante de $\ell_u$ ;

(ii) sinon, pour $k = u, \ldots, M - 1$,

- calculer (III.6) ; si (III.8) est satisfaite, calculer aussi $c(k+1) = c(k) + p\big(x(k+1)\big)t(k+1)$ ;
  - si $c(k+1) < c^*$, passer à la valeur suivante de $k$ ;
  - sinon, passer à la valeur suivante de $\ell_u$ ;
- si (III.8) n'est pas satisfaite, passer à la valeur suivante de $\ell_u$ ;

(iii) Fin de la boucle sur $k$ ; si $c(M) < c^*$, remplacer $c^*$ par $c(M)$ et $u^*$ par la valeur courante de $u$.

(5) Fin de la boucle sur $\ell_u$ ; si $u^* = 0$, le client est rejeté ; sinon le client est accepté et le nouvel itinéraire est donné par $\ell_{u^*}$.

CHAPITRE IV

# Une méthodologie d'exploitation du simulateur pour la gestion décentralisée

*Il n'y a pas moyen de savoir que la solution à un problème est la bonne tant que l'on n'a pas, au moins, fait l'effort d'en rechercher de meilleures.*

*Edward de Bono*

## IV.1. Introduction

Au cours de ce travail, nous n'avons pas cherché à étudier un cas réel car nous n'avions pas en face de nous un utilisateur potentiel susceptible de nous fournir les données correspondantes. Nous avons plutôt cherché à développer une méthodologie d'exploitation du simulateur présenté au chapitre II en nous concentrant, dans le présent chapitre, sur le mode de gestion décentralisée qui a été discuté plus en détail au chapitre III.

Il nous a donc fallu d'abord construire des données fictives, mais suffisamment réalistes, du moins c'est ce que nous espérons. Cependant, ces données n'étant pas réelles, nous ne prétendons pas tirer de conclusions définitives sur le système proposé. Nous sommes persuadés que l'intérêt plus ou moins grand d'un système de taxis collectifs dépend évidemment du contexte réel dans lequel on cherchera à l'appliquer : étendue de la région, densité du réseau, niveau de la demande potentielle et géométrie de celle-ci, et même des systèmes concurrents qui peuvent déjà exister sur le site.

Comme nous l'avons indiqué au §I.3.2, nous ne cherchons même pas à savoir la part de demande que ce système peut attirer, mais nous voulons simplement fournir un outil qui permette de faire fonctionner "au mieux" ce système en face d'une demande donnée. Il s'agit de fournir finalement à l'utilisateur des éléments *quantitatifs* sur la qualité de service d'une part, et sur les coûts correspondants d'autre part, que l'on peut espérer en face de la situation envisagée telle qu'elle est reflétée par les données qui ont été rapidement énumérées au §II.5. Il appartient ensuite à cet utilisateur d'estimer si le rapport performances/coûts obtenu est effectivement susceptible d'attirer la demande supposée, et éventuellement d'itérer en modifiant les données d'entrée de la simulation.

Il s'avère que, si la simulation est un outil extrêmement puissant qui fournit une quantité impressionnante de sorties, la démarche d'analyse et de synthèse de cette masse de résultats est en revanche loin d'être triviale. C'est l'objet de ce chapitre de montrer une façon de procéder, et de montrer aussi comment le système peut être diagnostiqué, et donc corrigé, dans ses dysfonctionnements, puis optimiser et calibrer. Cette démarche ne conduit pas à une solution unique, et l'utilisateur garde une grande part d'appréciation pour tirer le système vers plus de qualité de service ou vers de moindres coûts.

On va donc d'abord détailler les données quantitatives qui ont servi aux expériences numériques. Puis on va illustrer divers modes d'analyse des simulations. Pour une simulation donnée, on peut

- soit s'intéresser à des statistiques globales sur le réseau, les clients, les taxis ;
- soit se focaliser de façon interactive sur certains éléments ou agents au comportement extrême :

– repérer les nœuds mal desservis (où l'attente moyenne ou la file d'attente moyenne est longue) ;
– repérer les taxis peu actifs (en nombre de passagers transportés pendant la simulation ou en proportion du temps passé en stationnement à vide) ;
– repérer les passagers ayant subi des détours anormalement longs ;

et essayer de comprendre pourquoi.

Mais on peut ensuite considérer l'analyse d'une série de simulations (exploitant éventuellement la même séquence d'apparition de clients) en faisant varier un ou plusieurs paramètres afin d'optimiser la gestion temps réel et/ou le dimensionnement des ressources mises en œuvre.

On illustrera enfin l'influence sur les performances du système de paramètres liés à la demande (son niveau, sa géométrie).

## IV.2. Les données numériques

**IV.2.1. Choix du réseau.** Dans Scilab, il y a une démonstration de la partie Metanet (boîte à outils dédiée aux réseaux) qui utilise un plan du Métro parisien. Nous sommes partis de ces données pour construire notre réseau expérimental. Nous avons fait subir aux données quelques transformations qu'il est inutile de détailler ici. Nous avons abouti à un réseau de 288 nœuds et 674 arcs qui est représenté sur la figure IV.1. Sur ce plan, on a simplifié la

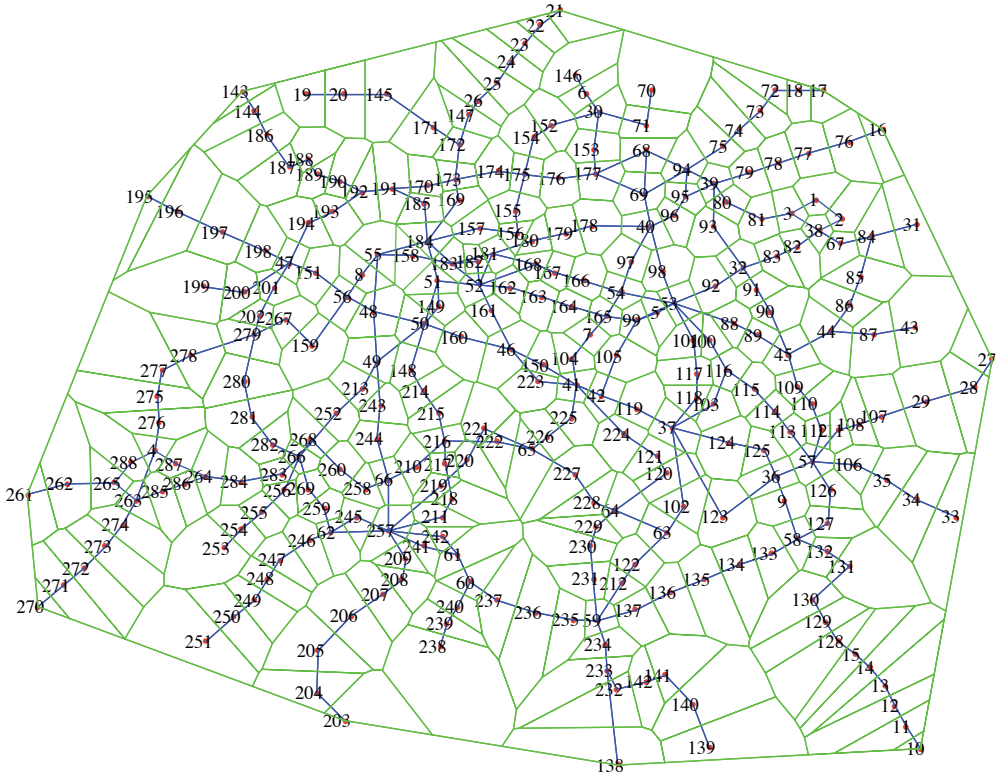

FIGURE IV.1. Plan du réseau

représentation en ne faisant pas figurer les arcs aller et retour entre paires de nœuds contigus (on rappelle que dans le graphe utilisé en simulation, les arcs sont à *sens unique*). Sur le dessin figurent aussi les cellules de Voronoï autour de chaque nœud du graphe : pour chaque cellule attachée à un nœud, ce sont les points du plan (en fait de l'enveloppe convexe du graphe) qui sont plus proches de ce nœud que de tout autre nœud : on peut donc les considérer comme les zones d'attraction de chaque nœud. *Mathematica* permet de les tracer et d'en calculer la surface : ceci nous servira plus loin pour fabriquer les données relatives à la demande.

Le nombre de nœuds et d'arcs de ce graphe peut paraître très modeste et il l'est probablement pour représenter une ville moyenne. Mais du point de vue de la durée d'exécution des simulations, ce qui compte vraiment c'est le nombre d'événements traités, et donc essentiellement du niveau de la demande (dont nous parlerons plus loin), même s'il est vrai que le nombre de nœuds rencontrés sur le parcours des taxis entre aussi en jeu dans ce décompte.

**IV.2.2. Temps de parcours.** On a décrit au §II.5.2.1 les lois de probabilité qui régissent les temps de parcours sur les arcs, à savoir des lois log-normales avec shift. On a également décrit à l'aide de quelles hypothèses on a pu fixer les trois paramètres associés à ces lois pour chaque arc : on rappelle simplement ici que 99% de la densité de probabilité est comprise entre le temps de parcours minimum (qui a été fixé à 80% du temps moyen de parcours) et 200% de ce temps moyen. Quant au temps moyen de parcours d'un arc, il est proportionnel à la distance géographique entre les extrémités de l'arc (avec une vitesse commerciale d'environ 30 km/h). Il est plus parlant de préciser que, pour l'ensemble du réseau, le temps moyen de parcours entre deux nœuds quelconques (non nécessairement contigus sur le graphe) est d'environ 18 minutes, et que le temps du parcours le plus long (par l'itinéraire le plus court basé sur les temps moyens) est d'approximativement 50 minutes.

Bien sûr, dans une étude de cas réel, on peut faire varier ces lois de temps de parcours au cours de la simulation pour représenter des conditions variables de circulation dans la journée.

**IV.2.3. Demande.** Comme on l'a expliqué au §II.5.1.2, les clients apparaissent aux nœuds du réseau selon des processus de Poisson décrits par un vecteur $\lambda$ de paramètres (un par nœud d'origine, paramètre ayant pour signification le nombre moyen d'apparitions par seconde), et leur destination est ensuite choisie par tirage selon la loi de probabilité contenue dans la ligne correspondant à leur origine dans une matrice O-D notée $M$.

Un scénario de demande est donc décrit par cette paire $(\lambda, M)$. Si l'on veut faire varier le niveau de la demande, il suffit de multiplier le vecteur $\lambda$ par un scalaire positif plus grand ou plus petit que 1. Pour faire varier la géométrie de la demande que l'on peut caractériser par le bilan "émission-réception" de tous les nœuds (voir (II.5)), on peut agir sur $M$ ou sur $\lambda$ (de façon *non* proportionnelle), voire même sur ces deux éléments.

On va donc dans un premier temps construire un scénario de demande dit "de référence" et qui correspondra à une demande plutôt *centripète*. On fera ensuite varier cette demande en intensité en multipliant $\lambda$ par un coefficient allant de 0,8 à 1,2, soit une variation de 50% entre la demande "faible" et la demande "forte". Par ailleurs, on conduira des expériences avec un scénario

- *équilibré* : on a expliqué au §II.5.2.2 comment on l'obtient en gardant $M$ inchangée mais en prenant pour $\lambda$ un vecteur propre à gauche de $M$ tout en respectant un certain volume global de la demande ;
- *centrifuge* : il s'agira de ce que nous avons appelé la demande "réciproque" de la demande de référence centripète, et cela oblige à changer à la fois le vecteur $\lambda$ et la matrice $M$ d'une façon qui a été décrite au §II.5.2.2 (voir les explications autour des équations (II.7)–(II.8)). On rappelle que dans la demande réciproque, l'équation de bilan aux nœuds (II.5) est changée de signe.

Pour construire la demande de référence, on a procédé selon les principes suivants :

**vecteur $\lambda$ :** on a tenu compte pour chaque nœud $i$ de la surface $S_i$ de la cellule de Voronoï qui entoure ce nœud et qui est censée représenter la zone d'attraction des habitants qui se dirigeront (à pied) vers ce nœud ; sommairement, $\lambda_i$ est proportionnel à $(S_i)^\alpha$ où $\alpha$ a été réglé pour que le ratio entre le plus grand et le plus petit $\lambda_i$ soit égal à 3 ; sur la figure IV.1, on note que les cellules les plus grandes se situent plutôt à la périphérie, et donc ce sera aussi le cas des nœuds les plus émissifs ;

**matrice $M$ :** on ne rentrera pas dans les détails mais on indique que l'on a tenu compte de 3 facteurs dans la construction de chaque élément $M_{ij}$ où $i$ est le nœud d'origine et $j$ est le nœud de destination ;

(1) on a tenu compte de la surface $S_j$ : plus le nœud de destination dessert une grande surface, plus il a de chances d'être choisi comme destination ; l'impact de ce facteur a été réglé à 1,5 lorsqu'on passe de la plus petite à la plus grande surface ;

(2) on a tenu compte de la distance $\delta(i, j)$ (en temps moyen de trajet direct) entre $i$ et $j$ : les clients sont supposés privilégier les courses en taxi les plus longues ; l'impact de ce facteur a été réglé à 2 lorsqu'on passe de la plus petite à la plus grande distance ;

(3) enfin, on a tenu compte du caractère centripète de la destination $j$ en tenant compte de la distance du nœud $j$ par rapport au centre du graphe ; l'impact de ce facteur a été réglé à 3 lorsqu'on passe de la plus petite à la plus grande distance.

Finalement, l'intensité de la demande de référence peut être résumée par un chiffre : il y a environ 15 400 clients à l'heure qui apparaissent sur l'ensemble du réseau. Pour se représenter la géométrie centripète de la demande, on a coloré dans des nuances plus ou moins foncées les cellules de Voronoï en fonction du bilan "émission-réception" du nœud centroïde de la cellule (cf. équation (II.5)). Sur la figure IV.2, les nuances les plus foncées correspondent aux nœuds avec le bilan (II.5) le plus négatif, c'est-à-dire les nœuds qui émettent plus de clients qu'ils n'en reçoivent.

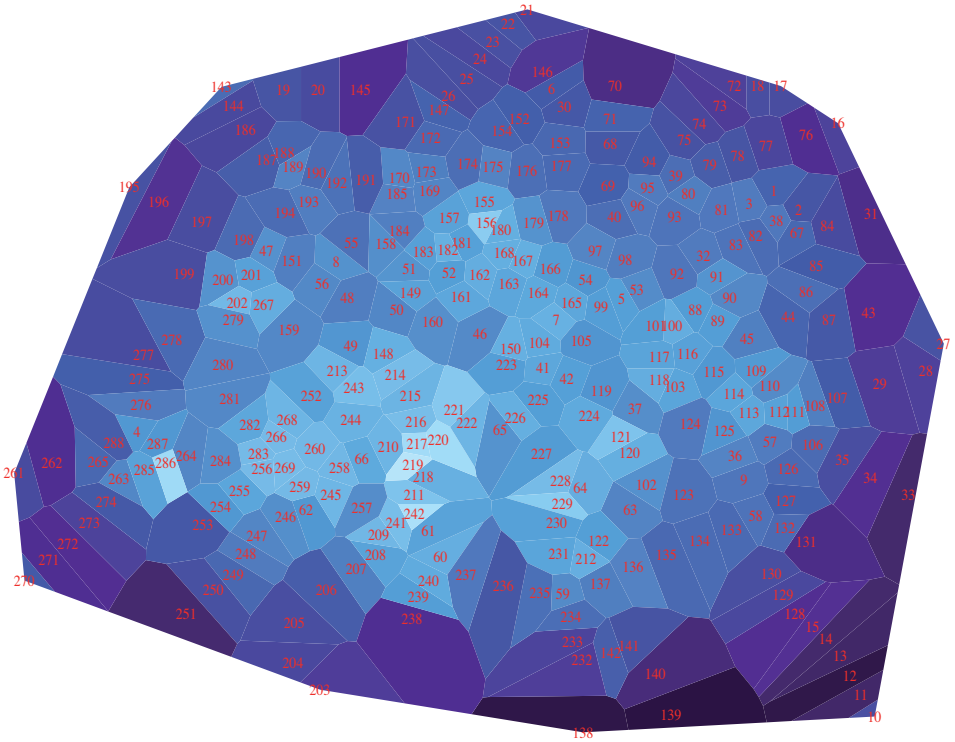

FIGURE IV.2. Demande centripète

Évidemment, si on considère la demande réciproque, cela donne, avec la même convention de coloration, la figure (IV.3). Quant à la demande équilibrée, il est inutile de la représenter de la sorte puisque le bilan (II.5) est par définition nul pour tous les nœuds.

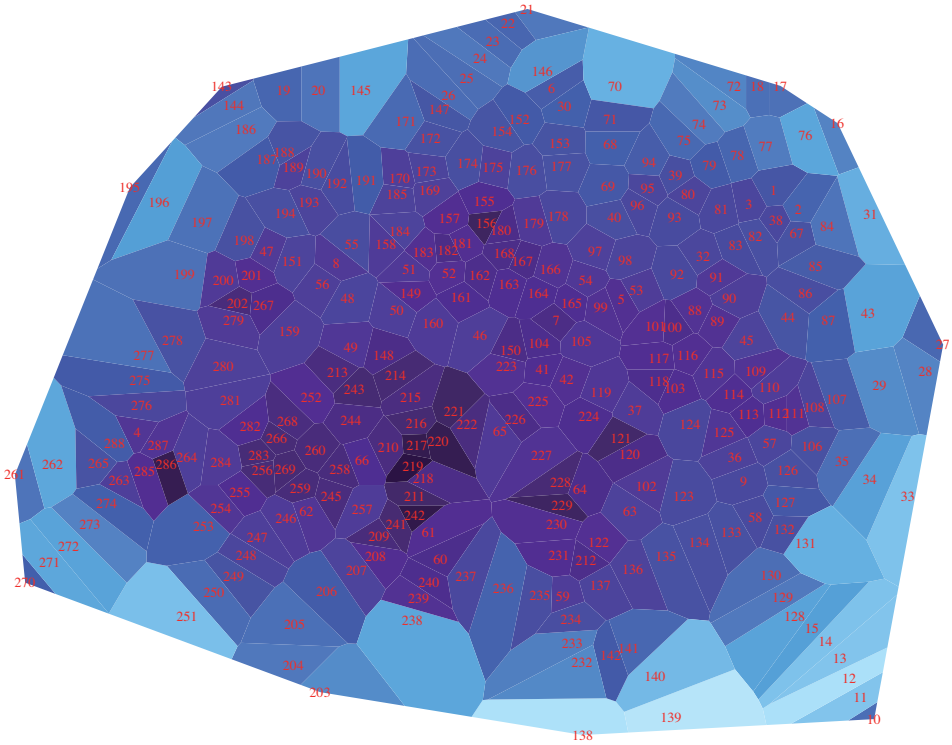

FIGURE IV.3. Demande centrifuge

**IV.2.4. Durée des opérations.** On rappelle que l'unité de temps élémentaire dans la simulation est la seconde. Un certain nombre d'opérations dans la simulation ne sont pas considérées comme instantanées mais ont une durée. C'est le cas des opérations d'embarquement et de débarquement des clients dans les taxis, et des opérations de dialogue entre clients et chauffeurs. Par ailleurs, lorsqu'un agent (client ou taxi) est mis en attente, c'est toujours pour une durée maximale au bout de laquelle, si son état d'attente n'a pas changé, sa situation sera systématiquement reconsidérée.

Toutes ces durées d'opérations ou d'attente maximale pourraient sans inconvénient être aléatoires selon des lois à préciser par l'utilisateur. Dans les expériences numériques rapportées dans ce chapitre, nous avons adopté des durées fixes qui sont précisées ci-après.

**Embarquement et débarquement de passagers :** chacune de ces opérations prend un temps unitaire de 10 secondes.

**Dialogue client/taxi :** Chaque dialogue, se terminant par une acceptation ou un refus, prend 30 secondes.

**Durée maximale d'attente des clients :** cette durée a été fixée uniformément à 10 minutes, que le client ait pu rencontrer un ou plusieurs taxis mais ait été refusé, ou qu'il n'ait pu rencontrer aucun taxi. Par conséquent, dans un laps de temps de 10 minutes (voire 10 minutes et 30 secondes) après son apparition à un nœud, le client est accepté par un taxi ou bien il quitte le système car on considère qu'il a renoncé. On comptabilise évidemment la proportion de clients qui abandonnent ainsi le système.

**Durée maximale de stationnement des taxis :** comme on l'a déjà dit, un taxi vide doit stationner au nœud où il a débarqué son dernier passager, ou bien il peut se diriger vers un autre nœud dans le but d'y stationner (à moins qu'il ne rencontre de nouveaux clients sur son chemin). Nous allons revenir ci-dessous sur le choix de ce nœud de stationnement, question qui a déjà été évoquée au §III.4.1. On pourrait lier cette décision à celle de la durée maximale de stationnement. Pour les expériences

rapportées ici, on a fixé cette durée maximale de stationnement à 15 minutes. À l'issue de ces 15 minutes, si le taxi est toujours en stationnement parce qu'aucun client ne s'est présenté à lui (ou parce que la longueur de la file d'attente des taxis à ce nœud est telle que ce taxi n'a pas pu avoir accès à un client), la question de son lieu de stationnement est reposée.

**IV.2.5. Questions diverses.** On regroupe ici un certain nombre de précisions supplémentaires qui ne sont pas stricto sensu des données numériques, mais qui sont nécessaires à la marche des expériences rapportées dans la suite de ce chapitre.

IV.2.5.1. *Retour sur le choix du nœud de stationnement des taxis vides.* Comme on vient de le préciser, on a dissocié ce choix de celui de la durée maximale de stationnement fixée uniformément à 15 minutes (on aurait pu évidemment lier ces deux choix).

La question du choix du nœud de stationnement a déjà été évoquée au §III.4.1. On y a indiqué que ce choix est basé sur la construction d'une sorte de matrice O-D où la ligne représente la position actuelle du taxi vide et la colonne représente le nœud qui sera choisi (à cette différence que dans une "vraie" matrice O-D, les éléments sur la diagonale sont généralement nuls car il y a une probabilité nulle de vouloir rester sur place, alors que pour le stationnement du taxi, ce choix est loin d'être absurde). On a vu que cette matrice a été construite en tenant compte de l'attractivité des nœuds en fonction de leur paramètre $\lambda_i$ représentatif du niveau de demande issue de ces nœuds, et aussi de la distance par rapport à la position actuelle pour éviter de trop longs parcours à vide.

Une fois cette matrice construite, on peut l'utiliser d'au moins deux façons :

- soit on choisit systématiquement la colonne qui, pour une ligne donnée, contient l'élément maximal, ce qui conduit donc à un choix *déterministe* ;
- soit on tire au hasard le nœud de stationnement selon une loi de probabilité proportionnelle à la ligne considérée de la matrice, ce qui correspond à une stratégie *randomisée*.

Dans un premier temps, nous avons adopté la première solution. L'intérêt de l'analyse détaillée de simulations qui sera présentée plus loin est que cette analyse a permis de détecter rapidement un fonctionnement catastrophique. On s'est aperçu, en analysant l'activité des taxis en moyenne et individuellement, que certains taxis ne transportaient aucun client pendant toute la durée de la simulation. En fait, ces taxis, mis en service à certains nœuds, se retrouvaient dans une file d'attente de taxis qui ne se vidait pratiquement jamais. Ils atteignaient donc la durée maximale d'attente avant d'avoir pu accéder à un client. Se reposant alors la question du nœud de stationnement, mais selon la stratégie déterministe, ils étaient amenés à choisir le même nœud et donc à se replacer systématiquement en fin de file, ce qui perpétuait le phénomène pendant toute la durée de la simulation. Le fait de passer à une stratégie randomisée a permis de surmonter ce dysfonctionnement. Évidemment, une autre stratégie (en feedback sur la longueur de la file des taxis au nœud de la position actuelle) aurait pu aussi éviter cette difficulté.

IV.2.5.2. *Algorithme d'acceptation/refus.* Le §III.4.2 a été consacré à la formulation de ce problème et à la présentation de quelques algorithmes exacts ou approchés pour le résoudre.

Dans la suite, on a utilisé l'algorithme exact par énumération exhaustive décrit au §III.4.2.4 lorsque cela a été possible en termes de temps de calcul, c'est-à-dire en fait lorsque les simulations utilisaient des véhicules de capacité limitée à 5 passagers au maximum. Avec cette capacité, on a mené quelques expériences préalables en traitant les cas d'acceptation/refus de clients à bord des taxis par les deux algorithmes exact et sous-optimal présentés aux §III.4.2.4 et III.4.2.5 respectivement, ceci afin d'évaluer la fréquence à laquelle ces deux méthodes ne donnaient pas le même résultat.

Il faut d'abord noter que lorsque le dialogue a lieu entre un client et un taxi vide ou comportant un seul passager à bord, il ne peut y avoir de différence entre "changer l'ordre de l'itinéraire prévu du taxi" ou "ne pas le changer" lorsqu'on insère le nouveau client dans cet itinéraire. La fréquence de cette situation (dialogue avec un taxi vide ou avec un seul passager à bord)

sera mesurée au cours d'une expérience rapportée plus loin et elle est de l'ordre de 30% (voir figure IV.14 ci-après).

Pour les autres cas d'une fréquence de l'ordre de 70%, on a mesuré la fréquence des cas de divergence entre les décisions prises par l'algorithme exact et l'algorithme sous-optimal à seulement 0,5%. On voit donc que, du moins pour des véhicules de capacité 5, on n'aurait pas eu de grandes différences dans les résultats en recourant à l'algorithme sous-optimal. Mais puisque l'algorithme exact du §III.4.2.4 était practicable, on a retenu cette méthode.

Malheureusement, lorsqu'on passe de la capacité 5 à la capacité 7 comme ce sera le cas au §IV.4.3, les temps de calcul deviennent prohibitifs avec cet algorithme ($5! = 120$ mais $7! = 5\,040$). Dans ce cas, on a dû se limiter à l'utilisation de l'algorithme sous-optimal du §III.4.2.5. Au vu de la fréquence très faible des cas de divergence entre les deux algorithmes constatée ci-dessus, on peut espérer que cela n'aura pas affecté grandement la qualité des résultats présentés. Cependant, la conclusion précédente est peut être à remettre en question lorsqu'on passe de la capacité 5 à la capacité 7.

C'est sans doute pour surmonter cette difficulté qu'il aurait fallu passer à une programmation en C ou C$^{++}$ pour cet algorithme, ce que nous n'avons pas eu le temps de faire.

IV.2.5.3. *Durée des simulations.* Les simulations délivrant des résultats basés sur la génération de nombres pseudo-aléatoires, il faudrait en toute rigueur répéter ces simulations un certain nombre de fois pour pouvoir estimer des moyennes et des variances de toutes les grandeurs auxquelles on s'intéresse avant de pouvoir tirer des conclusions à peu près sures. Dans les expériences que nous avons conduites, les données statistiques (modèles de la demande et des temps de parcours) sont stables (alors qu'on pourrait les faire varier au cours du temps pour la simulation d'un cas réel). On peut donc compter sur une certaine "ergodicité" des résultats, c'est-à-dire remplacer la répétition de simulations plus courtes par une simulation longue avec des données statistiquement stables afin de pouvoir "oublier" les conditions initiales et d'obtenir des estimateurs statistiques de variance suffisamment petite pour être représentatifs.

Après quelques essais, on a pu vérifier que des simulations d'une durée de 8 heures de temps réel étaient largement assez longues pour qu'il ne soit pas nécessaire de répéter ces simulations. Une simulation de 8 heures de temps réel prend environ 5 à 6 minutes sur un MacBook Pro (portable Apple) avec un processeur Intel Core i7 à 2,66 Ghz, 8 Go de RAM et un disque SSD (mémoire flash). Le fichier historique produit (voir §II.6) fait environ 600 à 700 Mo et la production des fichiers "clients" et "taxis" requiert environ 3 minutes [1].

## IV.3. Analyse détaillée d'une simulation

Dans cette section, on illustre la façon dont on peut analyser les résultats d'une seule simulation. Celle qui est utilisée ici correspond aux données suivantes :

- la simulation correspond à 8 heures de temps réel avec un démarrage "à froid" (système vide initialement) ;
- on place 13 taxis à chaque nœud du réseau soit 3 744 taxis au total ;
- la capacité des taxis est de 5 passagers ;
- la demande correspond à la demande dite "de référence", demande centripète générant en moyenne 15 400 clients à l'heure (soit 123 200 pour 8 heures) ;
- le seuil $s$ de l'algorithme d'acceptation/refus (voir §III.4.2 et équation (III.2)) est fixé à 1,9.

Au §II.6, on a expliqué que la simulation elle-même produit un historique complet des événements qui se sont déroulés pendant toute sa durée. Puis on extrait de cet historique un fichier "clients" (§II.6.1) et des fichiers "taxi" (§II.6.2). Ce sont ces fichiers qui sont ensuite traités par un script ScicosLab. Ce script est interactif : des pauses sont prévues dans son déroulement avec des possibilités de traitements optionnels. Par exemple, l'utilisateur peut choisir de vérifier ou

1. Sur une machine munie d'un disque SATA, ce temps était pratiquement 3 fois plus long, ce qui doit donc être probablement imputé à la durée de l'écriture sur disque de ces fichiers nombreux — un par taxi en particulier, soit de l'ordre de quelques milliers — et volumineux — environ 123 000 lignes dans le fichier clients.

de ne pas vérifier l'adéquation de la génération des clients ou des temps de parcours des taxis pendant la simulation avec les données probabilistes fournies en entrée du programme.

Après avoir examiné les résultats statistiques moyens et extrêmes comme par exemple :

- attente (ou file d'attente) des clients en moyenne sur tous les nœuds du réseau, puis attente moyenne minimale et maximale calculée individuellement nœud par nœud ;
- statistiques moyennes sur tous les taxis (passagers transportés, pourcentage du temps passé en circulation, au stationnement, etc.) puis mêmes statistiques minimales et maximales lorsqu'elles sont calculées individuellement par taxi ;

l'utilisateur peut examiner individuellement les statistiques des agents (clients ou taxis) de son choix, chaque agent ayant un numéro d'identification. En général on se penchera justement sur les agents ayant des statistiques extrêmes.

La suite de cette section donne un aperçu des informations que l'utilisateur peut obtenir dans une analyse détaillée de ce type. Certains des graphiques sont produits directement par ScicosLab. D'autres ont été obtenus avec *Mathematica* en exportant les résultats depuis ScicosLab.

**IV.3.1. Vérifications statistiques.** Il s'agit de vérifier si les statistiques que l'on peut faire sur 8 heures de simulation sont conformes aux lois de probabilité qui ont été introduites comme données. Ceci concerne essentiellement la demande et les temps de parcours.

IV.3.1.1. *Vecteur* $\lambda$. La coordonnée $i$ de $\lambda$ représente le nombre moyen de clients apparaissant à ce nœud par seconde. Il est donc facile d'estimer cette quantité à partir de l'historique de la simulation. Sur la figure IV.4, on a placé la valeur théorique de chaque $\lambda_i$ en abscisse et la valeur estimée en ordonnée, ce qui donne 288 points qui devraient, idéalement, être alignés sur la première bissectrice.

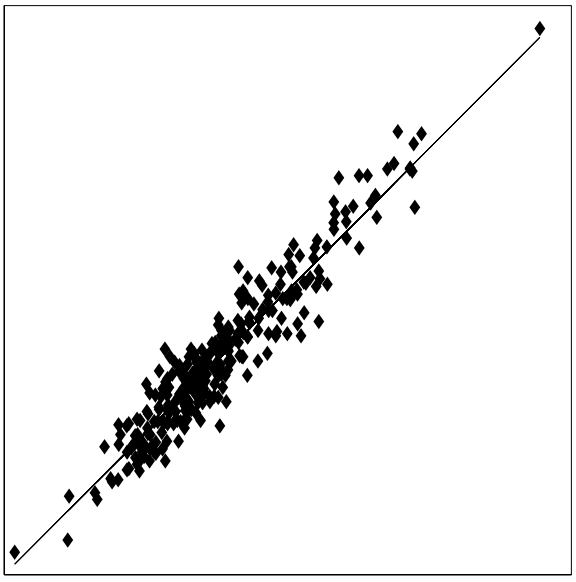

FIGURE IV.4. Vérification de $\lambda$

IV.3.1.2. *Vérification de M (matrice O-D).* Pour les clients qui apparaissent au nœud $i$, la probabilité de chaque destination $j$ est théoriquement $M_{ij}$. On peut par ailleurs estimer cette probabilité à partir de la simulation. Sur la figure IV.5, on a plutôt représenté cette loi de probabilité discrète par sa *fonction de répartition* : pour chaque valeur de $i = 1, \dots 288$, on a 289 valeurs allant de 0 à 1, la valeur $j + 1$ étant égale à $\sum_{k \leq j} M_{ik}$. Si on reporte ces 289 valeurs théoriques en abscisse et les valeurs estimées correspondantes en ordonnée, on obtient 288 "courbes" (en reliant les 289 points correspondant à chaque valeur de $i$), et ces courbes devraient toutes idéalement coïncider avec la première diagonale du carré $[0, 1] \times [0, 1]$.

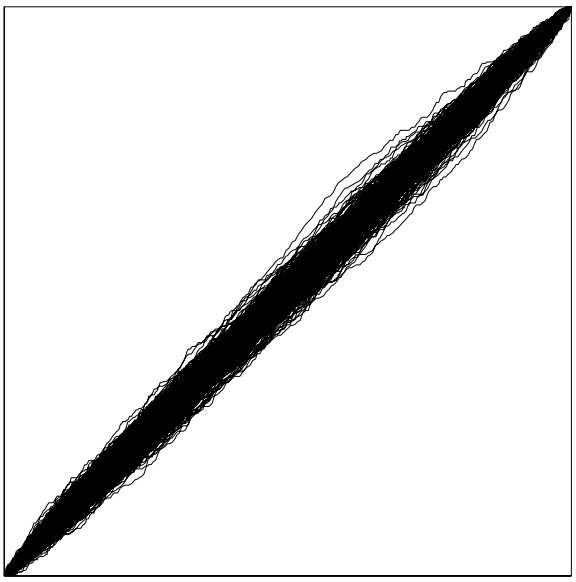

FIGURE IV.5. Vérification de la matrice OD

IV.3.1.3. *Vérification de la moyenne des temps de parcours des arcs.* Pour chaque arc du réseau, on vu que ces temps de parcours sont donnés par des lois log-normales avec shift (cf. §II.5.2.1 et IV.2.2). On a une moyenne théorique, et par ailleurs une estimation peut être faite pour chaque arc en observant tous les taxis qui ont traversé cet arc. Sur la figure IV.6, on a 674 points avec la valeur théorique en abscisse et la valeur estimée en ordonnée.

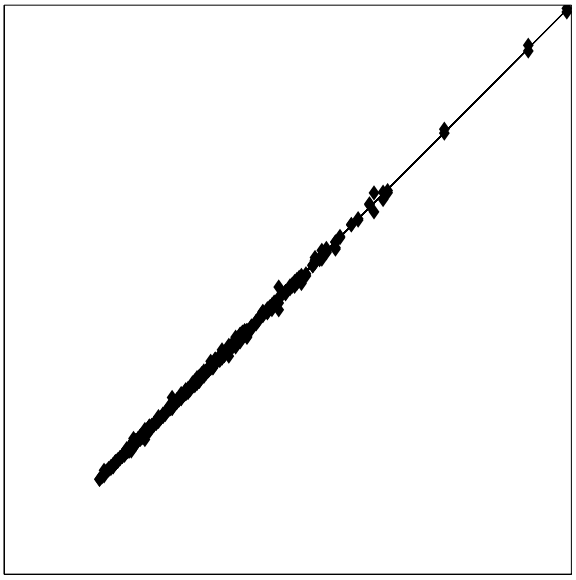

FIGURE IV.6. Vérification des moyennes des temps de parcours

### IV.3.2. Résultats concernant les clients et la qualité de service.

IV.3.2.1. *Taux d'abandon.* On a vu qu'un client apparu à un nœud attend au maximum 10 minutes (voire 30 secondes de plus s'il est en dialogue avec un taxi juste avant la fin de cette attente maximale) avant d'abandonner et de sortir du système. On doit évidemment surveiller le *taux d'abandon*, c'est-à-dire le pourcentage de clients ayant abandonné pendant la simulation, rapportée au nombre total de clients apparus (à l'exception de ceux qui sont encore dans une

file d'attente à la fin de la simulation). Ce taux d'abandon peut être calculé sur l'ensemble du réseau, et dans le cas de la simulation considérée, il est de 1,33%.

Mais on peut également calculer ce taux d'abandon nœud par nœud. La figure IV.7 permet de repérer rapidement les nœuds les plus critiques de ce point de vue. On voit qu'au nœud 10,

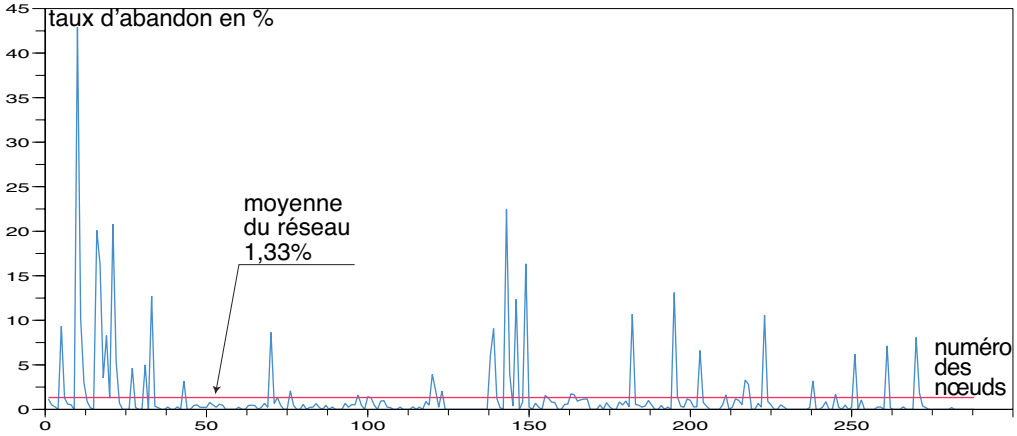


FIGURE IV.7. Taux abandon par nœud du réseau

le taux a dépassé 40%. Afin de mieux localiser géographiquement les nœuds critiques, on peut plutôt regarder la figure IV.8 : les nuances foncées indiquent les nœuds à fort taux d'abandon : le nœud 10 est situé à l'extrême Sud-Est et on voit que la plupart des nœuds critiques se situent en périphérie, ce qui n'est pas surprenant avec une demande centripète qui tend à envoyer les taxis vers le centre. Cependant, on localise quelques nœuds défavorisés qui se situent aussi vers le centre-ville. Le plus foncé est le nœud 149. On reviendra sur ce cas plus loin.

IV.3.2.2. *Temps d'attente des clients avant acceptation par un taxi.* On analyse le temps d'attente des clients ayant finalement embarqué dans un taxi (ce qui exclut ceux qui ont abandonné et ceux qui se trouvent encore dans les files d'attente en fin de simulation). Ils sont au nombre de 121 242. Le temps d'attente moyen pour tout le réseau est de 97 secondes. Cette valeur inclut le temps de dialogue entre le client et le taxi qui est de 30 secondes. L'écart-type est de 99 secondes. La figure IV.9 montre un histogramme de cette attente.

Là encore on peut procéder à une analyse par nœud du réseau plutôt que de considérer une statistique globale. La figure IV.10 indique le temps d'attente des clients à chaque nœud du réseau (numéro des nœuds en abscisse) : on a représenté pour chaque nœud la valeur moyenne plus et moins l'écart-type (ainsi que la moyenne et la moyenne plus ou moins l'écart-type pour l'ensemble du réseau).

Pour situer géographiquement les nœuds critiques, il faut regarder la figure IV.11 où la coloration des cellules de Voronoï est faite en fonction de la moyenne de l'attente au nœud correspondant (les couleurs foncées correspondent aux moyennes fortes). À nouveau, il apparaît que la périphérie du réseau est généralement plus défavorisée que le centre, mais il y a cependant des parties centrales qui sont également critiques. En particulier, le nœud 149 se signale à nouveau.

IV.3.2.3. *Longueur des files d'attente des clients aux nœuds du réseau.* À partir du fichier clients qui donnent la date d'apparition de chaque client au nœud d'origine ainsi que la date de son départ (à bord d'un taxi ou par abandon), on peut reconstituer l'historique de la file d'attente à n'importe quel nœud du réseau à la demande. La figure IV.12 montre l'évolution de la longueur de la file au nœud 149. On voit que la file a culminé à 12 clients à un certain moment. On peut aussi représenter l'histogramme de cette longueur de file, c'est-à-dire le pourcentage du temps passé avec 0, 1, 2, ... clients (figure IV.13). La longueur a dépassé 5 clients pendant 28% du temps. On pourrait enfin tracer une carte du même type que celle de la figure IV.11

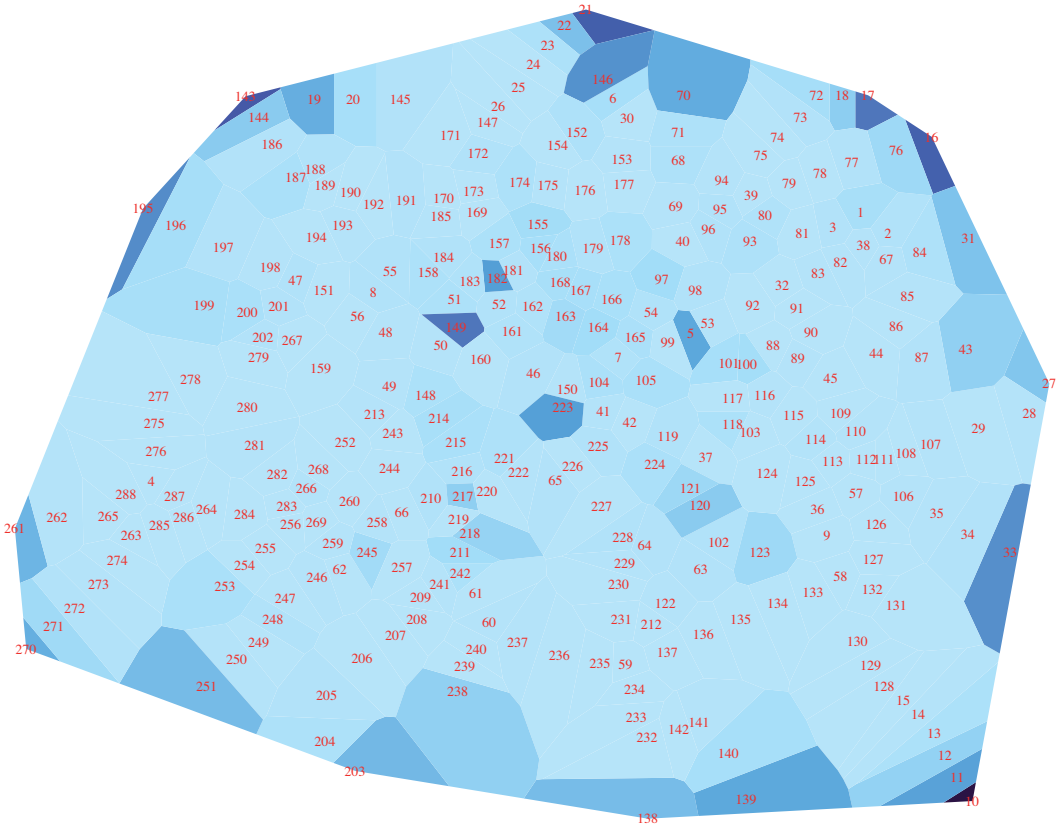

FIGURE IV.8. Carte du taux d'abandon aux nœuds du réseau

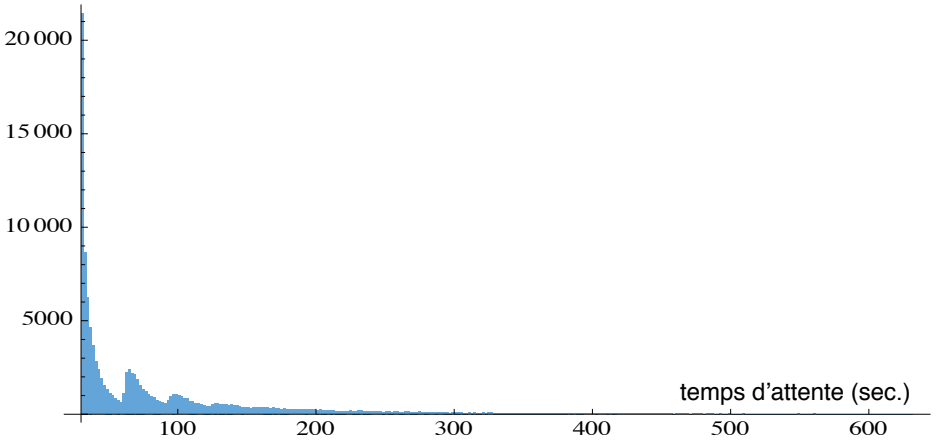


FIGURE IV.9. Histogramme du temps d'attente des clients pour l'ensemble du réseau

en considérant les valeurs moyennes de ces files d'attente. On constate en fait, ce qui n'est pas surprenant, une forte corrélation entre attente moyenne et longueur moyenne de la file à chaque nœud.

Le même type d'histogramme que celui de la figure IV.13 peut aussi être obtenu en regroupant l'ensemble des nœuds du réseau. Disons seulement ici que la moyenne de cette longueur de file globalement pour tout le réseau est de 1,54 clients.

IV.3.2.4. *Probabilité d'acceptation des clients par les taxis.* On peut se demander si les clients sont fréquemment acceptés par les taxis avec lesquels ils entrent en dialogue. Dans la simulation

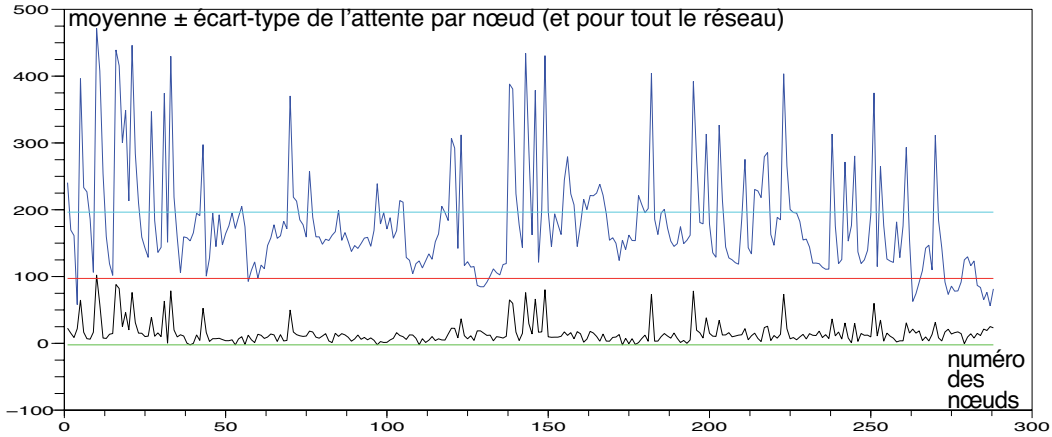


FIGURE IV.10. Moyenne plus ou moins écart-type de l'attente par nœud et pour l'ensemble du réseau

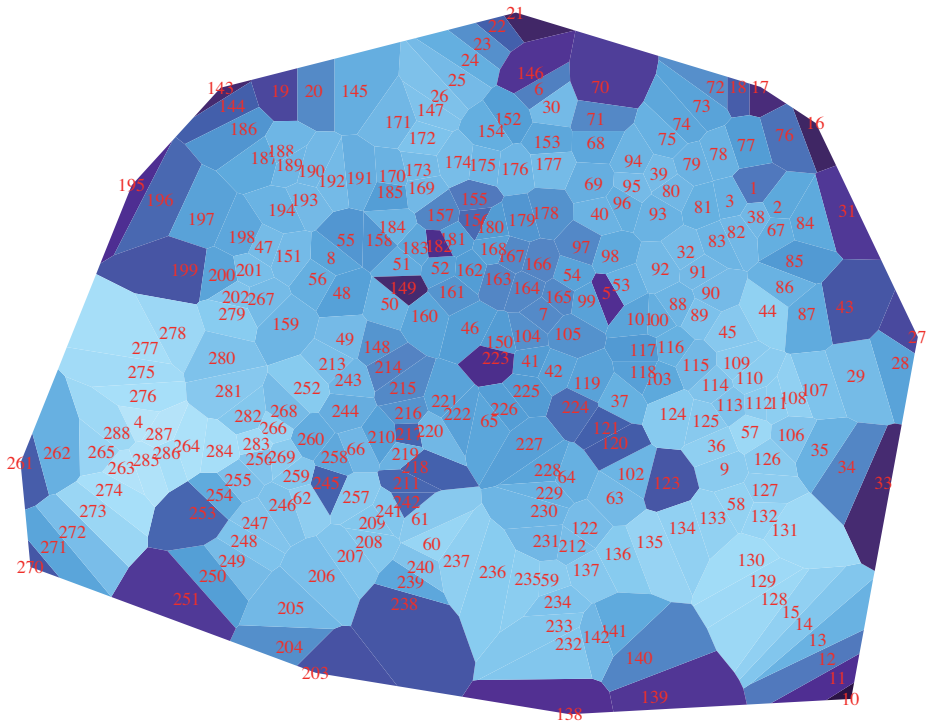

FIGURE IV.11. Carte du temps d'attente moyen des clients aux nœuds du réseau

analysée ici, la probabilité globale qu'un client soit accepté par un taxi est de 45%. Mais on peut pousser cette analyse plus en détail en mesurant la fréquence à laquelle un client rencontre des taxis vides, avec 1 passager, 2 passagers, etc. (on a écarté le cas des taxis pleins — 5 passagers ici — car on suppose que ces taxis ne s'arrêtent pas pour entamer un dialogue). On peut également évaluer la probabilité *conditionnelle* d'acceptation connaissant le nombre de passagers déjà présents dans le taxi. C'est un exemple de la finesse des résultats qu'une simulation permet d'obtenir.

Sur la figure IV.14 son représentées simultanément ces deux probabilités : celle de rencontrer un taxi avec $x$ passagers ($x = 0, \ldots, 4$) et celle d'être accepté lorsqu'il y a $x$ passagers à bord. Par exemple, on peut lire que la probabilité de rencontre avec un taxi ayant 1 passager à bord

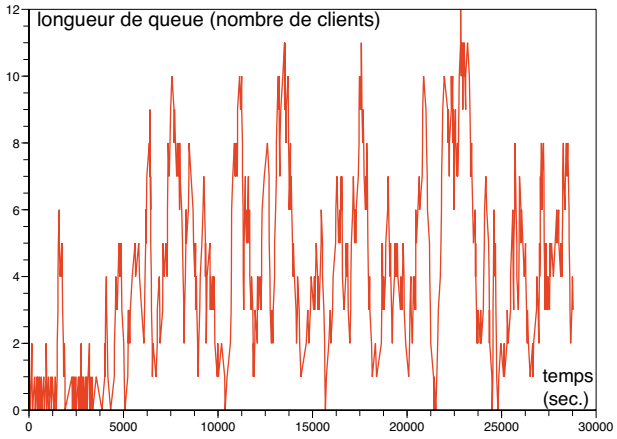


FIGURE IV.12. Évolution de la longueur de la file d'attente des clients au nœud 149

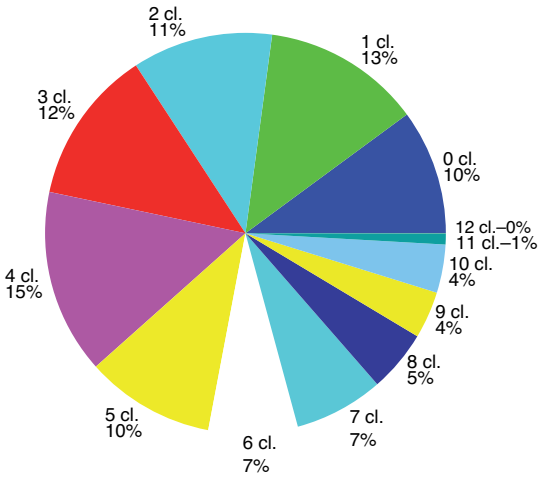


FIGURE IV.13. Histogramme de la file d'attente des clients au nœud 149 (pourcentage du temps passé avec $x$ clients dans la file)

est de 23% et que la probabilité d'être accepté dans cette circonstance est de 67%. Évidemment, avec un taxi vide, la probabilité d'acceptation est de 100%.

IV.3.2.5. *Détours.* Deux facteurs très importants de la qualité de service offerte aux clients sont, d'une part, l'attente initiale avant d'être accepté dans un taxi (voir §IV.3.2.2), et, d'autre part, la durée du trajet qu'ils subissent pour arriver à destination, durée qu'il faut évidemment comparer au trajet direct moyen pour aller de leur origine à leur destination. On introduit les deux notions suivantes :

**le ratio de détour :** il est défini comme le rapport entre, au numérateur, la durée du trajet effectué pour aller de l'origine à la destination, et au dénominateur, le trajet direct par le plus court chemin évalué avec la durée moyenne de parcours des arcs ;

**le ratio de détour *total* :** on reprend la définition précédente mais on rajoute au numérateur du rapport la durée de l'attente initiale avant embarquement.

Lors de cette simulation, on a mesuré le ratio de détour moyen pour tous les clients parvenus à destination avant la fin de la simulation (soit 112 883 clients au total) à 1,54 avec un écart-type de $0,39$. On rappelle que le seuil $s$ qui a été introduit dans l'algorithme d'acceptation/refus pour limiter ce ratio de détour a été fixé, pour cette simulation, à 1,9. La valeur moyenne obtenue est donc très inférieure à ce seuil. Par contre, certains clients peuvent subir des détours bien plus

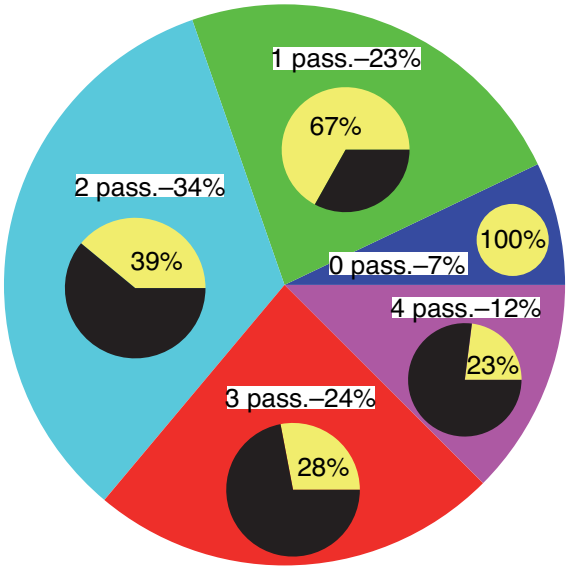


FIGURE IV.14. Probabilité conditionnelle d'acceptation des clients

importants : le maximum détecté dans cette simulation est de 7,24 ! La figure IV.15 présente

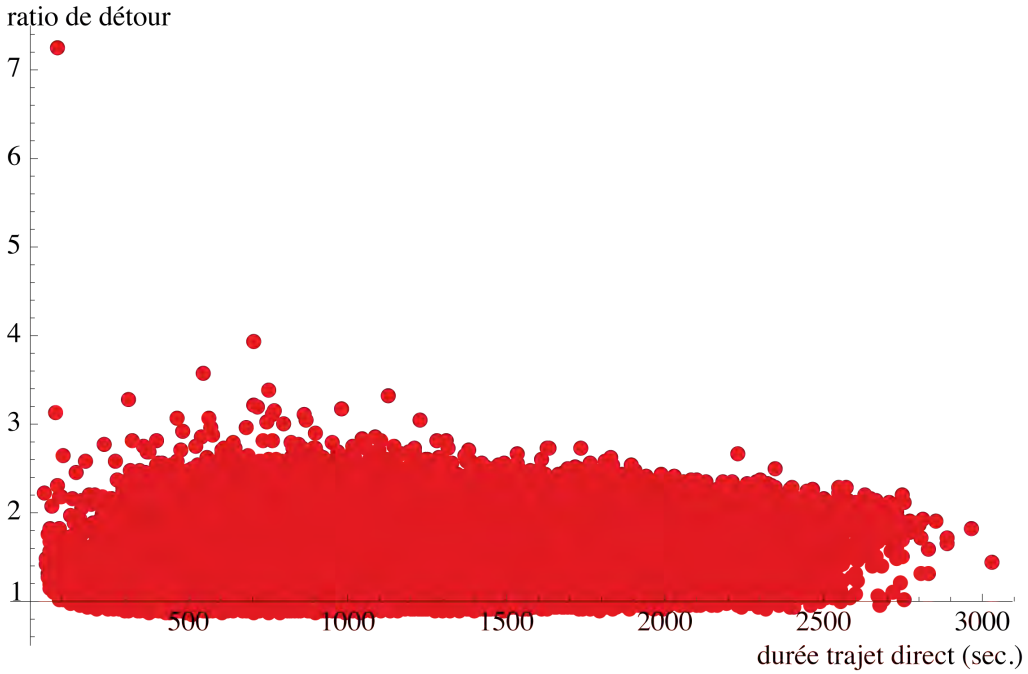


FIGURE IV.15. Corrélation entre durée du trajet direct et ratio de détour

pour chaque client un point représentatif du couple (durée du trajet direct, ratio de détour). Elle ne montre pas une corrélation très forte entre ces deux quantités, mais elle révèle que les valeurs du ratio de détour exceptionnellement grandes correspondent à des trajets directs relativement courts (inférieurs à 12 minutes). En particulier, la valeur extrême de 7,24 s'est produite pour un trajet direct de 83 secondes.

Sur la figure IV.15, on note aussi que le ratio de détour peut prendre des valeurs *inférieures* à 1 pour certains clients. Ceci s'explique par le fait que la durée du trajet direct placée au dénominateur du rapport est évaluée avec des temps *moyens* de parcours des arcs sur le réseau

alors que le numérateur correspond à une durée de trajet *effective* : puisque les temps de parcours sont aléatoires, il est possible de parvenir à destination plus vite que par le trajet direct moyen.

En ce qui concerne le détour *total*, la moyenne est de 1,64 avec un écart-type de 0,40.

La figure IV.16 présente les histogrammes du ratio de détour et du ratio de détour total (histogrammes limités aux valeurs inférieures à 3,5). On note que ces histogrammes présentent

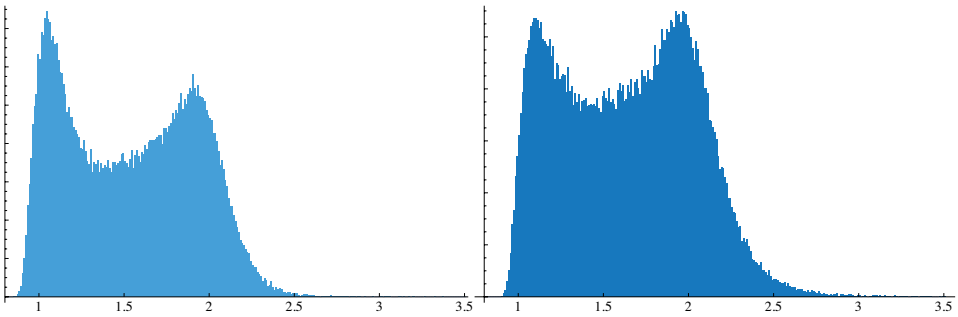


FIGURE IV.16. Histogrammes du ratio de détour (à gauche) et du ratio de détour total (à droite) tronqués à 3,5

deux maxima locaux correspondant apparemment à deux catégories de clients.

On s'est longtemps demandé à quoi correspondent ces deux catégories de clients. C'est l'illustration du fait que les simulations permettent de révéler certains phénomènes mais ne fournissent pas directement une explication à ces constats. Il faut alors faire preuve d'imagination pour émettre des hypothèses qu'on peut ensuite chercher à vérifier. L'hypothèse la plus convaincante nous a été proposée par Frédéric Meunier (LVMT-ENPC) qui a suggéré que les ratio de détours faibles correspondent à des clients montés "en premier" dans un taxi, c'est-à-dire dans un taxi vide.

Pour vérifier cette hypothèse, on a divisé la population des 112 883 clients amenés à destination en 5 catégories en fonction du nombre de passagers déjà présents dans le taxi au moment de leur embarquement :

(1) clients montés dans un taxi vide (18 261 clients),

(2) clients montés dans un taxi ayant déjà 1 passager (38 990 clients),

(3) clients montés dans un taxi ayant déjà 2 passagers (32 410 clients),

(4) clients montés dans un taxi ayant déjà 2 passagers (16 285 clients),

(5) clients montés dans un taxi ayant déjà 4 passagers (6 937 clients).

On a ensuite tracé (figure IV.17) les histogrammes du ratio de détour et du ratio de détour total pour chaque catégorie séparément. Il apparaît effectivement que l'importance relative des deux maxima dans ces histogrammes varie au fur et à mesure que l'on passe du taxi vide au taxi presque plein. Le phénomène est encore plus net avec le ratio de détour total.

Cependant, les deux maxima existent clairement dans tous ces histogrammes et nous avons cherché un autre facteur qui pourrait également caractériser, au moins en partie, les populations de clients correspondant à ces deux "modes". L'idée est de séparer la population des clients en deux catégories en fonction de leur ratio de détour individuel. Sur l'histogramme de gauche de la figure IV.16, on situe le creux entre les deux maxima à environ 1,35 en abscisse. C'est donc ce critère qui a été adopté. Cela conduit aux deux catégories suivantes :

(1) clients dont le ratio de détour est inférieur ou égal à 1,35 (42 580 clients) ;

(2) clients dont le ratio de détour est supérieur à 1,35 (70 303 clients).

On étudie ensuite un certain nombre de caractéristiques statistiques pour chacune de ces deux catégories séparément. Celle qui semble offrir l'explication la plus nette est la fréquence de la

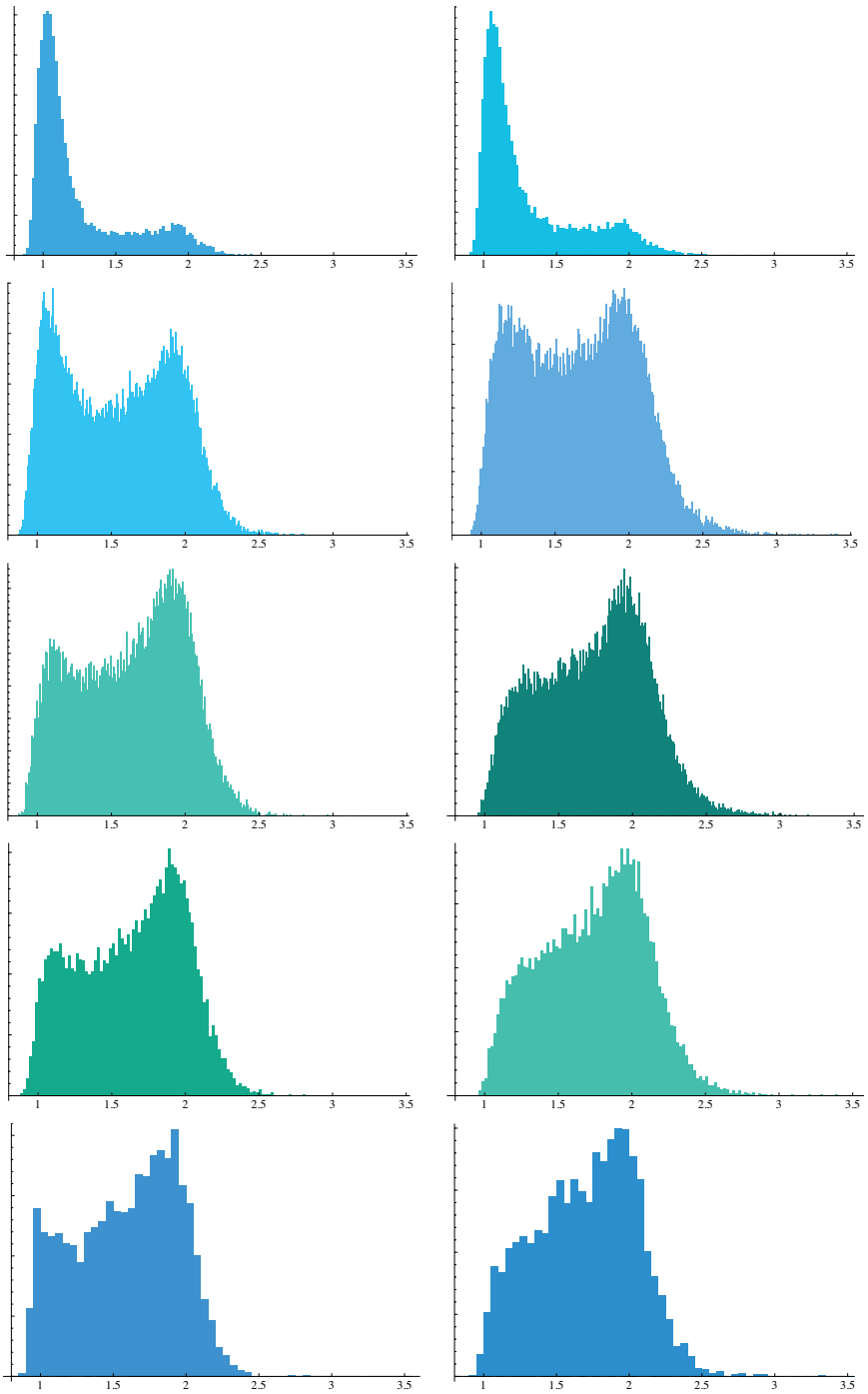


FIGURE IV.17. Histogrammes du ratio de détour (à gauche) et du ratio de détour total (à droite) pour les clients montés dans un taxi vide, puis avec 1 passager, 2 passagers, etc. (de haut en bas)

destination des clients dans les deux catégories. La figure IV.18 représente cette fréquence des destinations, les valeurs les plus foncées étant les plus faibles. La carte de gauche semble indiquer que les clients ayant un faible ratio de détour sont ceux qui cherchent à se diriger vers le centre.

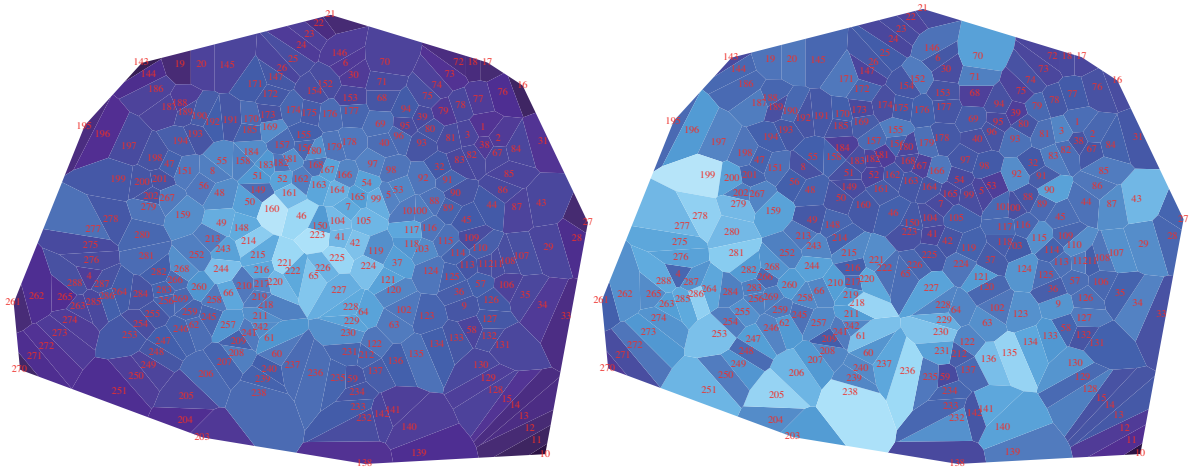

FIGURE IV.18. Fréquence des destinations des clients ayant un ratio de détour inférieur ou égal à 1,35 (à gauche) et supérieur à 1,35 (à droite)

### IV.3.3. Résultats concernant l'activité des taxis.

IV.3.3.1. *Fréquence de passage des véhicules.* Pendant une simulation, on peut mesurer la fréquence de passage des taxis par les nœuds et par les arcs du réseau. On se limite ici au passage par les nœuds. La figure IV.19 indique le nombre total de passages des véhicules par chaque nœud sur toute la durée de la simulation. On voit que ce nombre varie énormément d'un

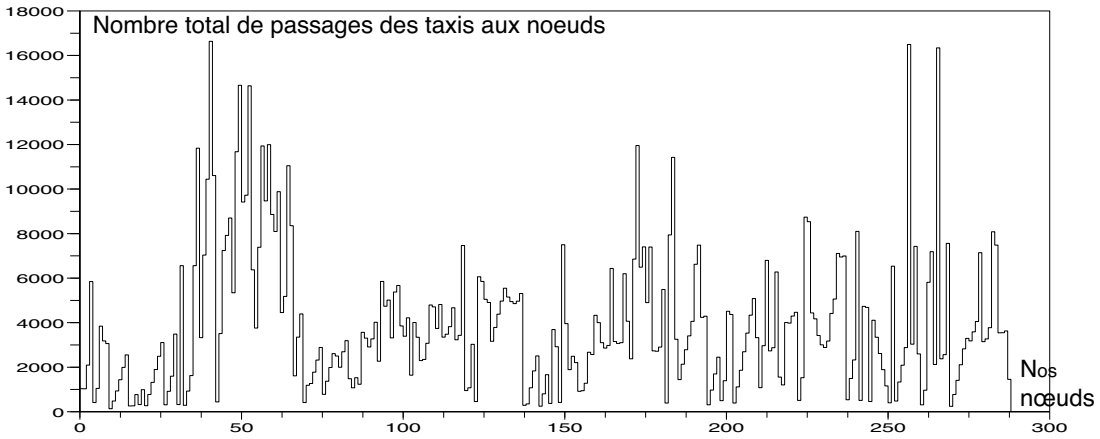


FIGURE IV.19. Nombre de passage des véhicules par chaque nœud

nœud à l'autre. On pourrait à nouveau cartographier ces nombres de passages pour avoir une vision géographique de cet indicateur. On observe déjà sur la figure IV.19 que le nœud 149, bien que situé vers le centre-ville, est relativement défavorisé. Nous allons donc chercher à expliquer ce phénomène.

La figure IV.20 représente une vue partielle du réseau dans le voisinage du nœud 149. Le premier nombre à côté de chaque nœud est son numéro et le nombre entre parenthèses en dessous du numéro représente le nombre de passages moyen *par minute* des taxis à ce nœud. On voit en effet que le nœud 149 est nettement moins fréquenté que les nœuds au voisinage et en particulier que les nœuds 50 et 51 qui sont adjacents. L'explication est sans doute que les taxis empruntent de façon privilégiée les arcs aller-retour directs entre 50 et 51 en "by-passant" le nœud 149. On voit donc comment une configuration défavorable du réseau peut être détectée en observant en détail les résultats de simulation.

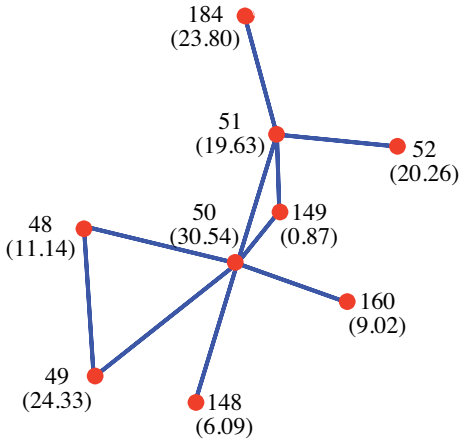


FIGURE IV.20. Fréquence de passage (par minute) des taxis dans le voisinage du nœud 149

IV.3.3.2. *Taux d'occupation moyen en nombre de passagers.* En moyenne sur la durée de la simulation (et plus précisément pendant tout le temps de leur trajet sur les arcs où la notion de nombre de passagers est bien définie), les taxis ont à bord 2,16 passagers sur une capacité de 5. On peut examiner cette statistique taxi par taxi et les valeurs extrêmes observées sont de 1,29 pour la plus petite, et 3,15 pour la plus grande.

C'est en effectuant ces observations qu'on s'était rendu compte que certains taxis restaient vides tout le long de la simulation à la suite d'une mauvaise politique du choix du nœud de stationnement, politique qui a été ensuite rectifiée comme cela a été expliqué au §IV.2.5.1.

La figure IV.21 représente le pourcentage du temps des trajets effectués en moyenne par tous les taxis avec $x$ passagers à bord, pour $x = 0, \ldots, 5$.

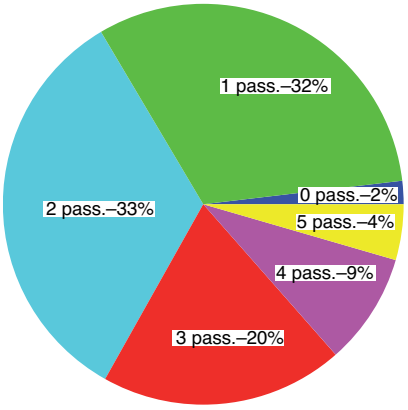


FIGURE IV.21. Histogramme du temps passé avec $x$ passagers à bord (moyenne sur tous les taxis)

Ce genre d'histogramme peut évidemment être obtenu à la demande pour un taxi particulier quelconque.

IV.3.3.3. *À quoi les taxis passent leur temps ?* La figure IV.22 donne le pourcentage du temps de la simulation pendant lequel les taxis sont :

(1) en mouvement sur les arcs du réseau ;

(2) à l'arrêt aux nœuds, occupés à diverses tâches (embarquement et débarquement de passagers, dialogues avec des clients, etc.) ;

(3) en stationnement à vide.

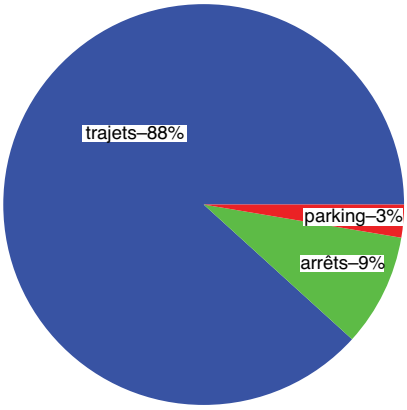


FIGURE IV.22. À quoi les taxis passent leur temps ?

On voit que ce temps de stationnement à vide est très réduit (3%) contrairement à ce qui se passe apparemment pour les taxis classiques.

Là encore, le même type d'histogramme peut être obtenu pour un taxi particulier à la demande. Ces statistiques d'activité des taxis permettraient d'évaluer les *dépenses de fonctionnement* des véhicules.

IV.3.3.4. *Vers l'estimation du chiffre d'affaires.* Comme nous l'avons dit dès l'introduction (cf. §I.3.2), nous n'avons pas introduit d'a priori ni sur la part de demande que ce système de taxis collectifs pourrait enlever aux autres modes de transport, ni sur la politique tarifaire qu'il y aurait lieu de mettre en œuvre. Tous les résultats examinés jusqu'à maintenant montrent comment on peut évaluer la qualité de service offerte aux clients en termes d'attente, de qualité du trajet, etc., et les résultats que nous venons d'examiner pour les taxis permettent d'évaluer les dépenses encourues pendant la durée simulée. Il reste à donner des indications sur ce que pourrait être le chiffre d'affaires en fonction d'une structure tarifaire choisie. On imagine deux modèles de tarifs possibles :

- une tarification fixe à la course,
- une tarification proportionnelle mais dans ce cas, il paraît logique de se baser non pas sur le trajet effectif subi par le client, mais sur le trajet direct entre son origine et sa destination.

D'autres modes de tarification plus compliqués pourraient être également considérés, notamment en faisant intervenir le nombre de passagers simultanément à bord du taxi, l'idée étant que plus le taxi est partagé, moins le service doit coûter cher aux clients. Mais on voit que cette notion n'est pas facile à mettre en œuvre pratiquement puisqu'au cours du trajet d'un client particulier, le nombre de passagers présents dans le taxi n'est pas fixe. On pourrait aussi imaginer, dans une optique de tarification proportionnelle, une forme de *décote* pour les détours subis par le client par rapport à son trajet direct. Mais là encore, la mise en œuvre ne serait pas immédiate quoique réalisable.

Pour chacun de ces modes de tarification, on peut demander aux simulations de produire les indicateurs permettant d'évaluer le chiffre d'affaires moyen des taxis. En nous limitant aux deux modes les plus simples envisagés ci-dessus (fixe — à la course, ou proportionnelle au trajet direct), on présente ici ces indicateurs à savoir, dans le cas du tarif à la course, le nombre de clients transportés par taxi en moyenne, et pour la tarification proportionnelle, l'histogramme des trajets directs des clients transportés.

Pour cette simulation d'une durée de 8 heures, les taxis ont amené à destination entre 21 clients (pour le taxi le moins actif) et 41 clients (pour le taxi le plus actif) avec une moyenne de 30,15 sur l'ensemble des 3 744 taxis. Rapporté à l'heure de simulation, ceci donne une moyenne de 3,77 clients/heure avec des valeurs extrêmes de 2,63 et 5,13.

Si on s'intéresse maintenant à la tarification proportionnelle (au trajet direct), il est intéressant de considérer l'histogramme des trajets directs des clients amenés à destination. La figure IV.23 présente cet histogramme qui suggère une distribution approximativement log-

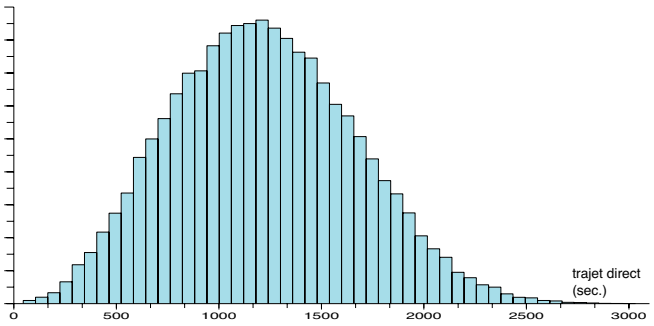


FIGURE IV.23. Histogramme des trajets directs des clients amenés à destination

normale. La moyenne de la durée des trajets directs est d'environ 20 minutes et 15 secondes avec un écart-type d'environ 7 minutes et 30 secondes.

Une fois de plus, ces statistiques peuvent être produites individuellement par taxi.

## IV.4. Analyse d'une série de simulations et optimisation

### IV.4.1. Une méthodologie pour l'optimisation du système.

IV.4.1.1. *Problématique.* Jusqu'à maintenant, on a essayé de montrer la diversité des renseignements qui peuvent être tirés de l'analyse détaillée d'une simulation (assez longue pour fournir des indicateurs statistiques fiables). Ces renseignements permettent de cerner la qualité de service offerte aux clients de ce système et les éléments permettant d'évaluer les coûts et les revenus potentiels pour les opérateurs du système.

Mais bien sûr, tous ces éléments dépendent de divers facteurs exogènes comme la topologie du réseau et la demande (en intensité et en géométrie). Ils dépendent aussi de décisions concernant d'une part la gestion temps réel et d'autre part le dimensionnement du système.

Au titre de la gestion temps réel, on a vu qu'il faut en particulier prévoir des stratégies pour affecter un stationnement aux taxis momentanément vides, et une politique d'acceptation/refus des clients à bord des taxis, ce qui passe aussi par la reconstruction permanente de leur itinéraire. On a proposé pour ce dernier problème une formulation et un algorithme faisant intervenir un paramètre $s$ (seuil de ratio de détour) dont il importe d'ajuster correctement la valeur. Au titre du dimensionnement du système, deux paramètres sont importants : le nombre $n$ de taxis mis en service et leur capacité en nombre de passagers.

La principale difficulté à laquelle on est confronté pour ajuster tous ces paramètres est paradoxalement l'abondance des sorties de simulations dont nous avons essayé de donner une idée au §IV.3. Comment parvenir à une synthèse de cette prolifération de résultats pour faire des choix quantitatifs à propos de ce paramètre $s$, du nombre $n$ de taxis en service et de leur capacité ?

Il importe que cette vision synthétique n'"oublie" aucun des aspects de la performance du système ni aucun des points de vue, à savoir ceux des clients et ceux des opérateurs, qui sont multiples et contradictoires dans chacune de ces deux catégories. Par exemple, comme on l'a déjà signalé, si on choisit une valeur de $s$ assez proche de 1, on obtiendra probablement une bonne qualité des trajets effectués par les passagers en termes de détours par rapport à leur trajet direct, mais on risque d'avoir un taux de rejet des clients en quête de taxi très élevé, ce qui conduira à une attente très longue, voire à un taux d'abandon très élevé. Ces deux ou trois indicateurs contradictoires (ratio de détour, attente initiale et/ou taux d'abandon) concernent principalement les clients, mais aussi indirectement les taxis (un fort taux d'abandon, ce sont

des clients en moins ; des détours importants, ce sont des courses inutiles si le taxi est rémunéré à la course ou proportionnellement au trajet *direct*). Les paramètres $s$ et $n$ ont un impact sur presque tous les indicateurs de qualité de service qu'on peut observer pendant les simulations.

Si on met en service beaucoup de taxis, ce sera probablement un facteur d'amélioration de la qualité de service offerte aux clients de tous les points de vue à la fois (car on se rapproche finalement du taxi individuel), mais il faudra alors certainement augmenter les tarifs pour équilibrer les coûts. Un accroissement de la capacité des taxis nous rapproche au contraire plus du transport collectif, donc d'un abaissement des coûts et des tarifs, mais probablement avec une qualité dégradée des trajets, du moins si le système est réglé pour accroître le taux effectif d'occupation des véhicules.

Ceci donne une idée des nombreux dilemmes qu'il faut tenter de résoudre en se basant sur *quelques indicateurs quantitatifs pertinents*, car s'ils étaient trop nombreux, on aurait certainement du mal à dégager les bons choix. Ceci dit, il est évident que dans ce genre de problème avec des tendances contradictoires, la réponse n'est pas unique et on pourra "tirer" le système vers plus de qualité ou bien vers des coûts plus bas. La décision finale appartient à l'opérateur du système. On cherche ici à lui donner quelques outils pour éclairer cette décision.

On va donc d'abord essayer de sélectionner un minimum d'indicateurs statistiques permettant de cerner les différents aspects du système sans en oublier de fondamental. On va ensuite élaborer une méthodologie pour se servir de ces indicateurs afin de fixer les principaux paramètres de réglage du système :

– le paramètre $s$ de l'algorithme d'acceptation/refus ;
– le nombre $n$ de taxis en service ;
– la capacité de ces taxis ;

sachant que ces choix sont probablement liés les uns aux autres et ne peuvent pas être faits indépendamment.

On va ensuite étudier à l'aide de cette méthodologie l'influence des données exogènes et particulièrement la demande (intensité, géométrie) sur les performances ainsi optimisées du système.

IV.4.1.2. *Choix des indicateurs.* Parmi tous les résultats qui ont été examinés au §IV.3, on a vu que certains d'entre eux sont relativement bien corrélés positivement. C'est par exemple le cas de l'attente moyenne et de la longueur moyenne de file d'attente. Dans ce cas, il suffit de retenir l'un des deux.

D'autres contiennent au contraire une information unique que l'on ne pourra pas retrouver, même indirectement, au travers d'autres indicateurs. C'est en particulier le cas du taux d'abandon (cf. IV.3.2.1). En effet, les clients ayant abandonné prématurément le système ne se retrouvent dans aucune autre statistique : il faut donc explicitement garder une trace de cet indicateur.

Enfin, certains indicateurs sont en conflit les uns avec les autres et la quantification du compromis à opérer dans cette circonstance nécessite d'observer leur variation simultanée.

En fonction de ces réflexions, on a finalement retenu les 3 indicateurs suivants :

(1) le taux moyen d'abandon des clients pour l'ensemble du réseau : on le placera sur l'axe des $x$ dans les graphiques (voir §IV.3.2.1) ;
(2) le nombre moyen de clients transportés par taxi pendant une simulation de 8 heures (voir §IV.3.3.4) ; on a vu que cet indicateur est pertinent pour estimer le chiffre d'affaires dans le cas d'une tarification fixe à la course ; avec d'autres types de tarification, il y aurait lieu d'adapter l'indicateur ; on placera cet indicateur *changé de signe* sur l'axe des $y$ dans les graphiques ; la raison du changement de signe sera expliquée plus loin ;
(3) le ratio de détour *total* moyen : il a été défini au §IV.3.2.5 ; l'avantage de cet indicateur est qu'il reflète à la fois la qualité des trajets (en termes de détours par rapport au trajet direct) mais aussi l'attente initiale ; on le placera sur les graphiques le long de l'axe des $z$.

Il est intuitif que les indicateurs $x$ et $z$ s'amélioreront (c'est-à-dire diminueront) lorsque le nombre $n$ de taxis en service augmentera, alors que l'indicateur $y$ (en valeur absolue) ira plutôt en se détériorant (c'est-à-dire qu'il diminuera lui aussi). Pour mieux appréhender le compromis à réaliser de ce point de vue, on a décidé de changer le signe sur l'axe des $y$ de sorte que, pour les trois indicateurs représentés graphiquement, *meilleur* soit synonyme de *plus petit*.

IV.4.1.3. *Représentation graphique.* Ayant choisi les 3 indicateurs ci-dessus, on va considérer une série de simulations dans laquelle la capacité est maintenue fixe mais dans laquelle on fait varier les paramètres $s$ et $n$ de façon incrémentale.

Par "série de simulations", on entend des simulations répétées pour lesquelles on maintient au maximum toutes les données exogènes constantes lorsqu'on fait varier ces deux paramètres. Comme on l'a déjà expliqué, les données exogènes aléatoires concernent la demande (génération des clients) et les temps de parcours des arcs (un temps de parcours doit être généré à chaque fois qu'un taxi emprunte un arc). Cette dernière génération aléatoire dépend de décisions endogènes : si on fait varier $s$, les décisions d'acceptation/refus de clients s'en trouvent modifiées, les itinéraires suivis par les taxis ne sont plus les mêmes, et par conséquent les temps de parcours générés ne concernent pas toujours les mêmes arcs d'une simulation à l'autre. Par contre, l'apparition des clients aux nœuds, la date de cette apparition, et leur destination sont des entrées qui ne dépendent de rien d'autre que de la séquence pseudo-aléatoire utilisée. Par conséquent, on peut réutiliser les mêmes historiques de clients pour toute une série de simulations après les avoir stockés dans un fichier, pourvu cependant que la série de simulations corresponde entièrement à une loi de demande unique (ce mode d'exploitation a été décrit au §II.3.3).

Chaque simulation d'une série produit donc un triplet de valeurs $(x, y, z)$ qui ont été définies ci-dessus et qui peuvent être représentées par un point dans un espace 3D. Une série de simulations produit un nuage de points et ces points sont le résultat du choix de 2 paramètres $s$ et $n$. On a donc obtenu la discrétisation d'une variété de dimension 2 dans un espace de dimension 3. Ce qui nous intéresse, c'est de choisir un point sur cette "surface" qui ait les coordonnées les plus petites possible comme expliqué plus haut, c'est-à-dire, dans la configuration du repère 3D de la figure IV.24, de nous situer le plus possible "en bas et à gauche".

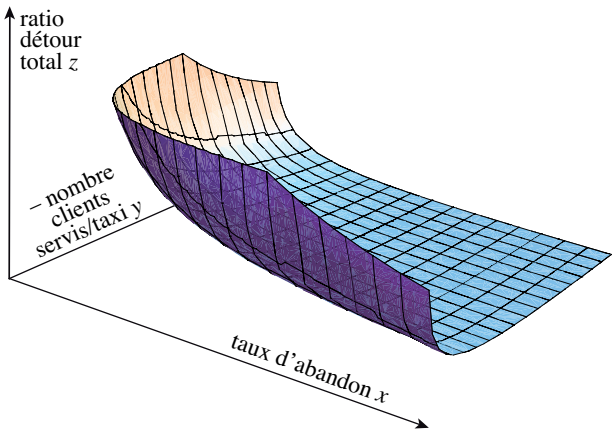


FIGURE IV.24. Représentation s'une série de simulations

Dans un premier temps, pour améliorer le tracé de la surface, on peut éliminer tous les points qui ne sont pas sur la *frontière de Pareto*, c'est-à-dire tous les points qui sont strictement dominés par d'autres points du nuage, autrement dit, tous les points pour lesquels on peut trouver un autre point avec des coordonnées $(x, y, z)$ toutes strictement plus petites. Il reste ensuite à trianguler la surface à partir de son "squelette" du nuage de points de Pareto reflétant la série de simulations.

Pratiquement, on a écrit un script ScicosLab qui traite (cette fois-ci en "batch" et non pas de façon interactive) la série de simulations pour extraire le tableau des 3 indicateurs choisis (en fait

on en profite pour en évaluer d'autres également) en fonction des plages de valeurs balayées par les paramètres $s$ et $n$. On traite ensuite ce tableau (toujours en ScicosLab) pour en extraire les points non dominés. On transfère enfin ce tableau "nettoyé" à *Mathematica* qui trace la surface en 3D par triangulation. On utilisera aussi une représentation 2D dans le plan $(x, y)$ en traçant aussi les lignes de niveau de la coordonnée $z$.

Tous les points de la surface obtenue étant des points de Pareto, aucun n'est meilleur que les autres du point de vue des trois critères simultanément. Mais on peut identifier des zones "raisonnables" et d'autres "moins raisonnables" : typiquement, dans une zone où l'on peut, en se déplaçant sur la surface, gagner beaucoup sur un ou deux critères sans trop dégrader le ou les autres, on a en général intérêt à faire ce déplacement.

Ce qui est certain, c'est que si pour deux séries de simulations correspondant par exemple à deux configurations de demande différentes, les surfaces sont "bien ordonnées" l'une par rapport à l'autre (c'est-à-dire ne se chevauchent pas), alors, indépendamment du compromis choisi sur ces surfaces, on pourra affirmer qu'une configuration est plus favorable que l'autre.

Nous allons voir plusieurs exemples de ces comparaisons en faisant varier
- le niveau de demande à géométrie constante,
- la géométrie de la demande (centripète, centrifuge, équilibrée) à niveau constant,
- la capacité des véhicules que nous avons supposée fixée pour une série de simulations dans la représentation graphique que nous avons décrite.

**IV.4.2. Influence de la demande sur les performances.**

IV.4.2.1. *Influence du niveau de la demande.* On va comparer ici les performances du système en face de deux hypothèses de demande de même géométrie mais d'intensités différentes. On considère la demande dite "de référence" qui correspond à un scénario centripète et on multiplie le vecteur $\lambda$ correspondant par 0,8 (12 320 clients à l'heure) et par 1,2 (18 480 clients à l'heure) soit un écart de 50% entre le scénario "faible" et le scénario "fort".

En suivant la procédure décrite précédemment, on obtient les deux surfaces de la figure IV.25 : la surface "du dessous", donc la meilleure, correspond évidemment à la demande forte. La figure IV.26 donne une représentation dans le plan $(x, y)$ avec tracé des lignes de niveaux montrant la valeur de $z$. Les points choisis sur les deux surfaces sont comparables en ce sens qu'ils ont les mêmes valeurs pour les coordonnées $y = -31$ et $z = 1,7$. Par contre celui qui est sur la surface de demande forte à sa coordonnée $x$ égale à $1,33$ alors que celui correspondant à la demande faible se situe à $x = 2,71$. Autrement dit, pour le même nombre de clients transportés en moyenne par taxi et pour le même ratio de détour, on obtient un taux d'abandon deux fois meilleur pour la demande forte.

Bien sûr, ces fonctionnements ne sont pas obtenus en utilisant les mêmes valeurs des paramètres $s$ et $n$. La figure IV.26 donne les valeurs de ces paramètres qui ont été estimées, pour chacun des points, par interpolation sur les trois points effectivement obtenus par des simulations et qui définissent le triangle auquel le point de fonctionnement choisi appartient. Il y aurait lieu, idéalement, de relancer une simulation avec ces valeurs pour vérifier les résultats interpolés. Observons ici simplement que le nombre de taxis $n$ correspondant à la demande faible (soit 2 852) et à la demande forte (soit 4 320) sont dans un rapport de 51%, ce qui est conforme à l'augmentation de 50% du niveau de la demande et au fait que le nombre moyen de clients transportés par taxi est le même dans les deux cas.

IV.4.2.2. *Influence de la géométrie de la demande.*

**Comparaison d'une demande équilibrée avec une demande centripète.** On a vu au §II.5.2.2 comment on peut construire une demande *équilibrée*, c'est-à-dire une demande où tous les nœuds sont statistiquement autant émissifs que réceptifs, à partir du matrice O-D donnée. On considère donc ici la demande équilibrée associée à la matrice O-D de notre demande de référence et on compare les performances atteignables avec ces deux types de demande. Évidemment, les deux demandes sont calibrées en intensité pour émettre toutes les deux le même nombre de clients à l'heure sur tout le réseau, en l'occurrence 15 400. Elles ne diffèrent donc que par leur

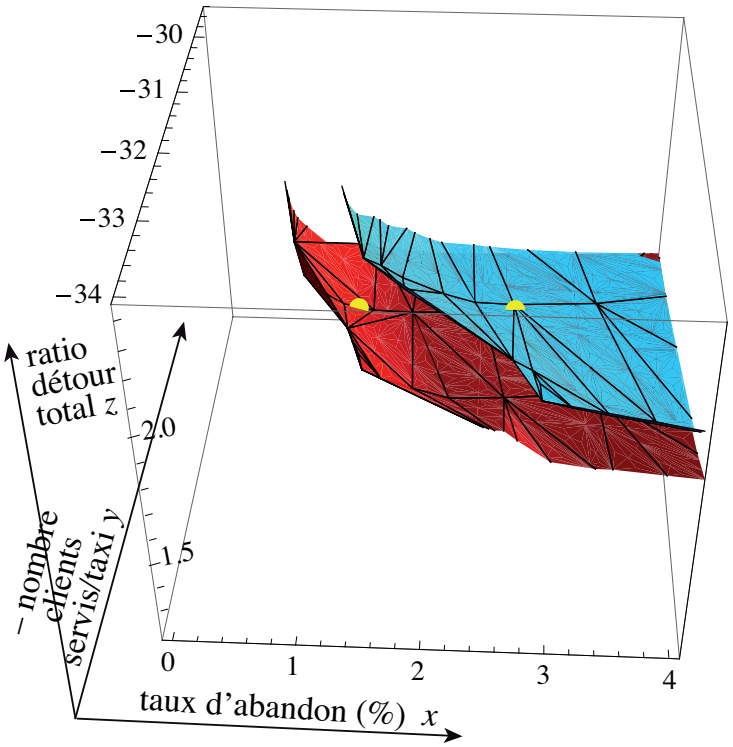


FIGURE IV.25. Comparaison de deux intensités de demande (cas d'une demande centripète) — vision 3D

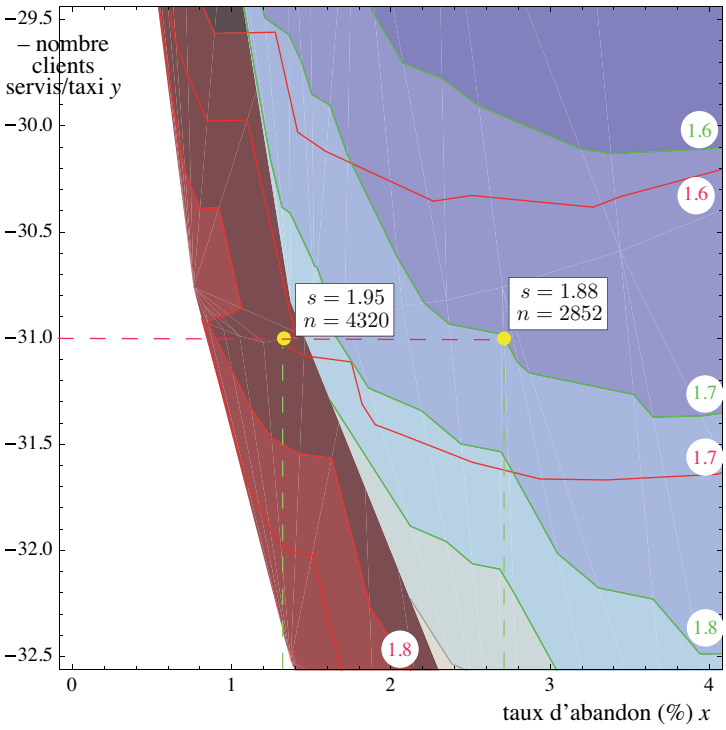


FIGURE IV.26. Comparaison de deux intensités de demande (cas d'une demande centripète) — vision 2D

géométrie puisque la demande de référence est centripète. Pour donner une idée du déséquilibre de cette demande, disons que le nœud avec le bilan (défini par (II.5)) le plus négatif "perd" environ 33 clients à l'heure et celui avec le bilan le plus positif "gagne" à peu près 21 clients à l'heure.

La figure IV.27, montre les surfaces obtenues pour ces deux demandes, sachant que la surface

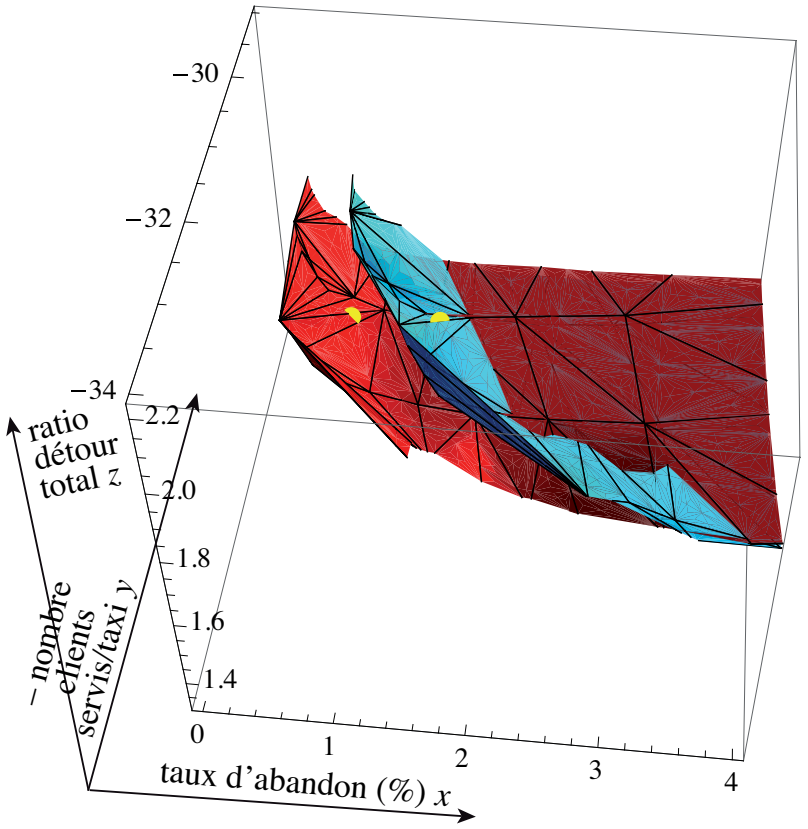


FIGURE IV.27. Comparaison d'une demande équilibrée et d'une demande centripète de même intensité — vision 3D

"du dessous" (donc la meilleure) est associée à la demande équilibrée. En fait, les deux surfaces s'intersectent, mais dans la zone intéressante, c'est bien la demande équilibrée qui est la plus favorable, ce qui n'est pas étonnant.

Sur la vision 2D de la figure IV.28, les points de fonctionnement choisis pour les deux demandes correspondent à nouveau à la même valeur de $y = -31$ et à la même valeur de $z = 1,8$. Le point correspondant à la demande équilibrée à une abscisse $x = 0,85$ alors qu'on obtient $x = 1,54$ pour la demande centripète, autrement dit une dégradation du taux d'abandon de plus de 80%. Ces points sont obtenus pour des valeurs sensiblement égales des paramètres $s$ et $n$ (valeurs estimées dans les deux cas par interpolation sur 3 sommets d'un triangle correspondant à des points effectivement simulés). Ces valeurs sont indiquées sur la figure IV.28.

**Comparaison d'une demande centripète avec une demande centrifuge.** On considère à nouveau la demande dite "de référence" centripète et on la confronte à la demande réciproque dont nous avons montré comment elle peut être définie au §II.5.2.2. Cette demande correspond aux mêmes flux statistiques entre toutes les paires de nœuds en inversant le rôle des origines et les destinations, ce qui donne ici une demande centrifuge, aussi déséquilibrée que la demande centripète et avec la même intensité (15 400 clients à l'heure).

Sur la figure IV.29, on voit que les surfaces obtenues correspondant à ces deux demandes sont

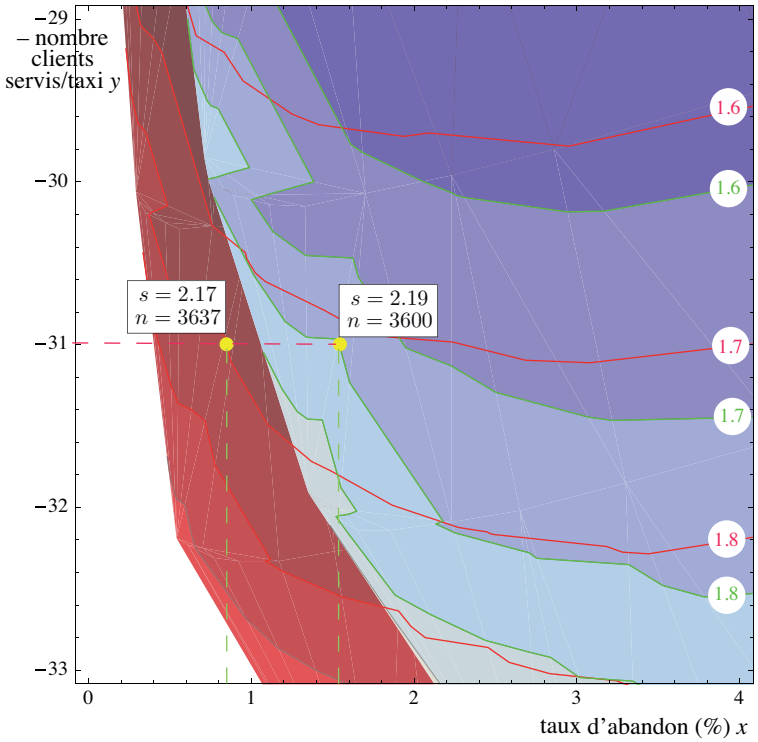


FIGURE IV.28. Comparaison d'une demande équilibrée et d'une demande centripète de même intensité — vision 2D

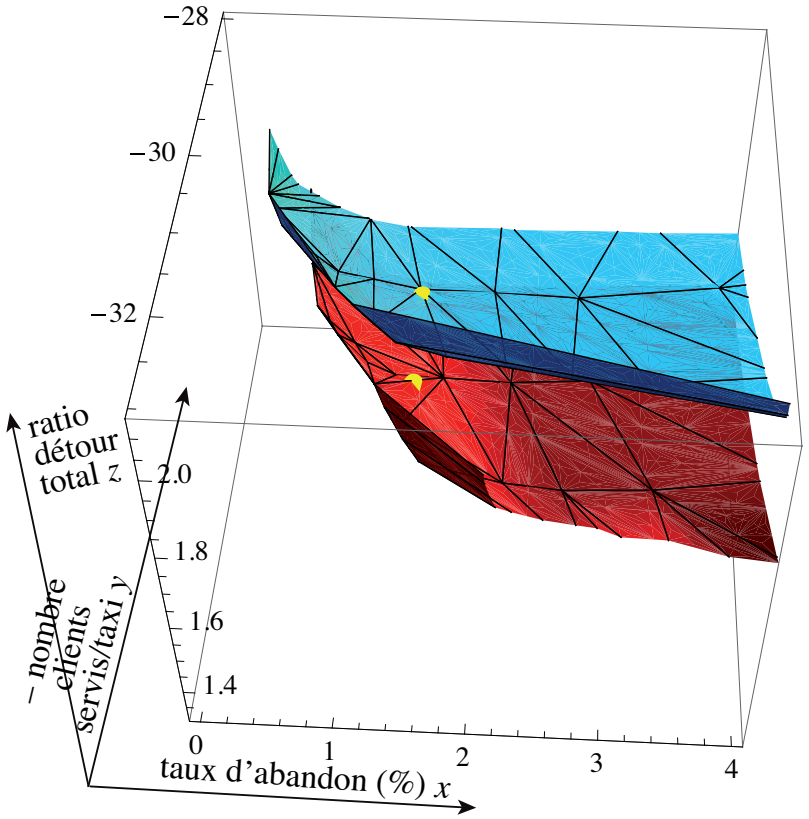


FIGURE IV.29. Comparaison d'une demande centripète et de sa demande réciproque (centrifuge) — vision 3D

bien distinctes, la surface “du dessous” (la meilleure) correspondant à la géométrie centripète qui apparaît donc être plus favorable.

Sur la figure IV.30 montrant la représentation 2D, on a choisi le même point de fonc-

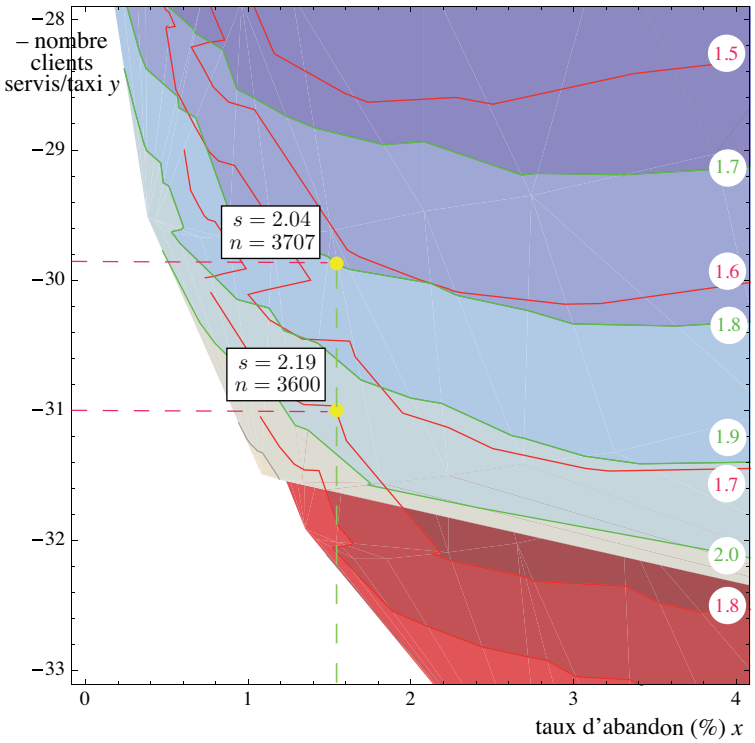


FIGURE IV.30. Comparaison d'une demande centripète et de sa demande réciproque (centrifuge) — vision 2D

tionnement que précédemment pour la demande centripète, correspondant aux coordonnées $\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} 1,54 & -31 & 1,8 \end{pmatrix}$. On peut le comparer au point de la surface correspondant à la demande centrifuge de coordonnées $\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} 1,54 & -29,87 & 1,8 \end{pmatrix}$ offrant le même taux d'abandon et le même ratio de détour total, mais un nombre moyen de clients transportés par taxi (pour 8 heures de simulation) en recul de 3 à 4%. De fait, ce point correspond à un nombre véhicules en service en augmentation d'environ 3%.

On peut interpréter intuitivement ce résultat en disant que dans la demande centrifuge, les véhicules tendent à se disperser vers la périphérie du réseau ce qui nécessite, pour maintenir le même taux d'abandon et le même ratio de détour total, un peu plus de véhicules en service.

**IV.4.3. Variation de la capacité des véhicules.** Jusqu'à maintenant, toutes les simulations présentées correspondaient à des véhicules de capacité égale à 5 passagers. Dans cette section, nous allons étudier l'effet d'un recours à des véhicules de capacité 7 en comparant deux séries de simulations pour la même demande, à savoir la demande de référence (centripète) avec $\lambda$ multiplié par 1,2 (environ 18 480 clients à l'heure).

On rappelle que lorsque l'on passe à la capacité 7, l'algorithme exact du §III.4.2.4 devient impraticable pour une question de temps de calcul et on a dû se contenter de l'algorithme sous-optimal du §III.4.2.5. Comme cela a été discuté au §IV.2.5.2, on espère, mais sans pouvoir le garantir, que cela n'a pas affecté grandement la validité des comparaisons présentées ci-dessous.

La Figure IV.31 représente les surfaces correspondant à des simulations avec des véhicules de capacité 5 et 7 : la surface “du dessous” correspond aux véhicules de capacité 7. En réalité, dans le domaine où les deux surfaces existent, elles sont assez proches et elles s'intersectent comme on peut le voir en observant les lignes de niveaux sur la représentation 2D de la figure IV.32 On

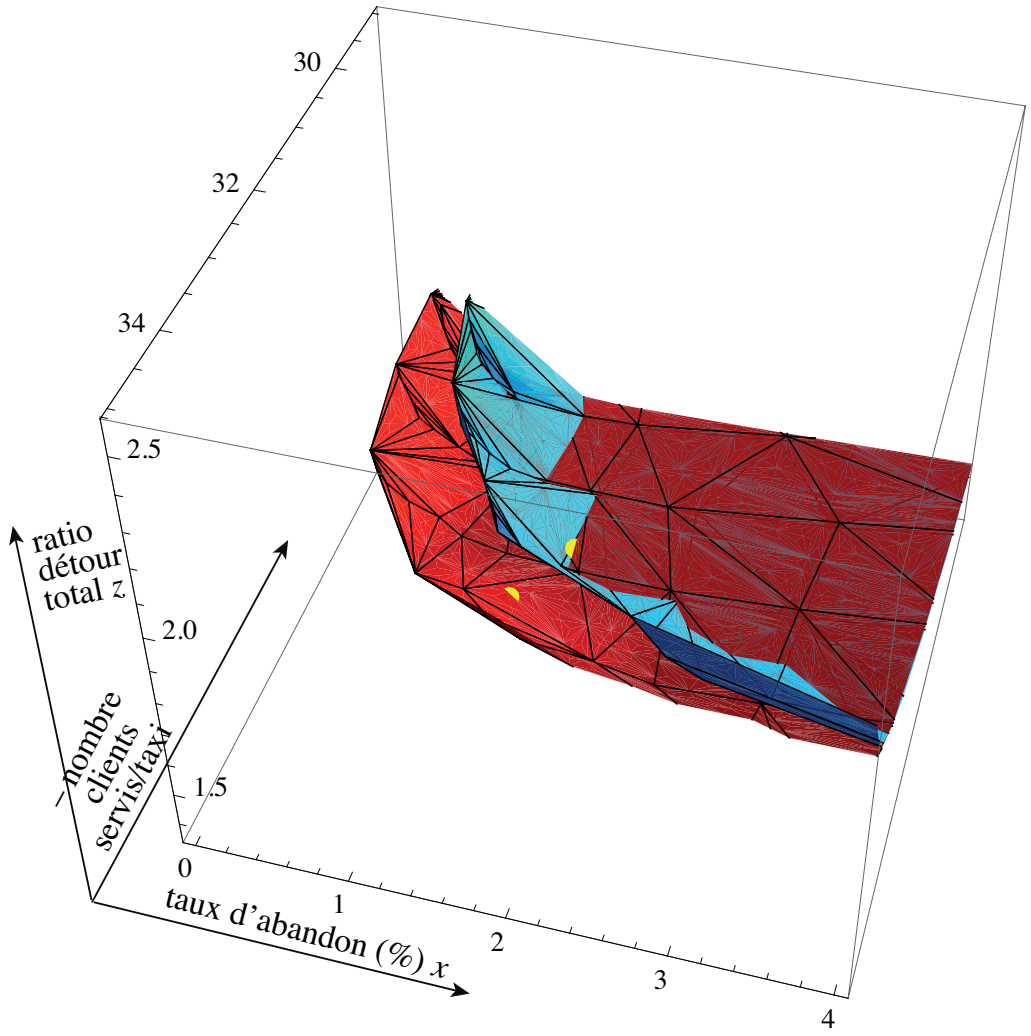


FIGURE IV.31. Comparaison d'une exploitation avec des véhicules de capacités 5 et 7 — vision 3D

peut même dire que la surface correspondant aux taxis de capacité 5 devient meilleure lorsqu'on se déplace vers les $x$ ou les $y$ croissants (c'est-à-dire en réalité pour un taux d'abandon plus grand ou pour un nombre moyen de clients transportés par taxi plus petit).

Plaçons nous en un point qui se trouve à l'intersection de ces deux surfaces : comme d'habitude, les coordonnées $\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} 1,89 & -32,29 & 1,8 \end{pmatrix}$ de ce point sont calculées, pour chacune des deux surfaces, par interpolation sur les 3 sommets d'un triangle contenant le point et correspondant à des simulations effectivement faites. Comme indiqué sur la figure IV.32, ce point est obtenu pour pratiquement le même nombre $n$ de véhicules en service (4104 pour le cas de la capacité 5 et 4105 pour le cas de la capacité 7) et pour des valeurs de $s$ légèrement différentes (seuil plus sévère pour le cas de la capacité 7). Dans cette configuration, on peut dire que la capacité 7 n'est pas vraiment utilisée et le système fonctionne pratiquement de la même façon qu'avec des taxis de capacité 5.

La véritable utilité de la capacité 7 se fait sentir si on veut pousser plus loin le système vers un transport collectif. En effet, si on considère par exemple le point cerclé de noir de la figure IV.32 de coordonnées $\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} 1,89 & -34,20 & 2,1 \end{pmatrix}$, ce point correspond, pour le cas de la capacité 7 aux paramètres $s = 2,43$ et $n = 3802$. Ce point n'est par contre pas atteignable avec des véhicules de capacité 5. On voit donc que pour le même taux d'abandon

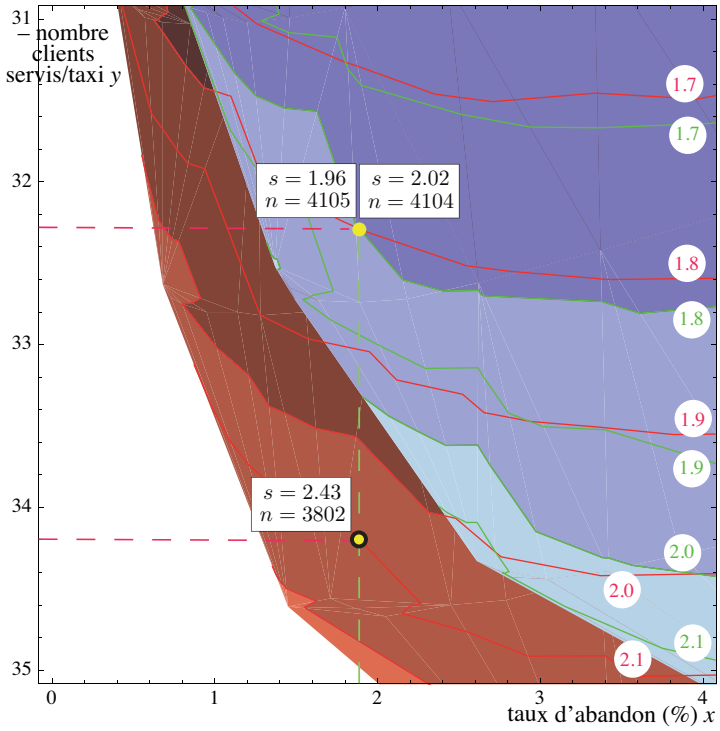


FIGURE IV.32. Comparaison d'une exploitation avec des véhicules de capacités 5 et 7 — vision 2D

que précédemment (1,89), on parvient maintenant à augmenter le nombre moyen de clients transportés par taxi de près de 6% (en mettant évidemment en service de 7 à 8% de taxis en moins). Cependant cet accroissement du chiffre d'affaires s'accompagne d'une baisse de qualité de service puisque le ratio de détour total passe de 1,8 à 2,1, soit une augmentation de près de17%.

En conclusion, l'augmentation de la capacité des véhicules de 5 à 7 places permet d'obtenir, pour certains réglages des paramètres $s$ et $n$, un mode de fonctionnement pratiquement identique et ce changement n'est donc pas intéressant dans ces plages de valeurs. Par contre, cette augmentation de capacité permet de pousser plus loin le nombre de clients transportés en moyenne par taxi (en diminuant notamment le nombre $n$ de taxis en service, mais en augmentant aussi parallèlement le seuil $s$ de l'algorithme d'acceptation) : ceci permettrait donc d'abaisser le tarif pour les clients, mais au prix d'une dégradation sensible de la qualité de service représentée par le ratio de détour total. On s'écarte alors un peu plus du service offert par les taxis individuels pour se rapprocher d'un système de transport collectif.

CHAPITRE V

# Gestions centralisée et mixte

*Les gens superstitieux vous recommandent instamment de ne jamais passer sous une échelle, mais ils ne vous empêchent pas de passer sous un taxi.*

*Pierre Dac*

## V.1. Introduction

Dans ce chapitre, on présente le modèle de simulation relatif au fonctionnement du système de taxis collectifs dans le mode de gestion centralisé puis dans le mode mixte.

Le mode centralisé correspond au cas où les clients appellent un dispatching central pour réserver un trajet d'un nœud d'origine vers un nœud de destination, ce trajet débutant à une heure souhaitée. Le rôle du dispatching est de trouver le taxi qui, compte tenu des informations connues au moment de l'appel, sera le mieux adapté à répondre à cette demande, et d'informer le client de l'heure probable de rendez-vous qui doit se situer dans une fourchette commençant à l'heure souhaitée du départ et limitée en durée après cette heure idéale. De plus, le taxi affecté doit être susceptible d'amener le client à destination dans un délai décompté à partir de l'heure souhaitée de départ et n'excédant une limite calculée à partir d'un certain pourcentage du trajet direct de l'origine à la destination. Si aucun taxi n'est susceptible de satisfaire ces contraintes, le client est refusé.

Accessoirement, on peut imaginer qu'un autre taxi soit finalement désigné pour acheminer le client si, entre le moment de l'appel et le moment du rendez-vous prévu, cet autre taxi s'avère finalement plus apte à satisfaire cette demande compte tenu de l'évolution de la situation dont on suppose qu'elle est connue du dispatching à tout instant et pour tous les taxis et clients.

La gestion mixte est celle où les clients avec réservation coexistent avec les clients sans réservation de la gestion décentralisée telle que décrite au chapitre III.

Le simulateur a été conçu, dans sa partie dite "mécanique", pour modéliser les agents intervenant dans tous ces modes de gestion, à savoir :

– les clients,
– les dispatcheurs ou serveurs,
– les nœuds du réseau (files d'attente diverses),
– les véhicules.

Comme cela a été fait au chapitre III pour la gestion décentralisée, on va décrire les types d'événements à traiter et l'enchaînement de ces types événements dans le mode de fonctionnement centralisé puis mixte.

Pour fonctionner, cette partie mécanique du simulateur doit être interfacée avec des algorithmes de gestion temps réel. On se limitera ici à formuler le problème de l'affectation d'un taxi à un appel de client au niveau du dispatching, puis à discuter rapidement de quelques méthodes de résolution.

## V.2. Les agents dans la gestion centralisée

**V.2.1. Clients.** Le comportement typique d'un client est le suivant. Son apparition dans la simulation est aussi celui de son appel au dispatching. Cet appel doit être pris en charge par

un dispatcheur humain ou un serveur télématique. Si aucun dispatcher n'est libre, un temps d'attente maximal doit être généré au bout duquel le client abandonnera si son appel n'est pas pris en charge entre temps.

Le client souhaite réserver pour lui, et éventuellement pour d'autres personnes l'accompagnant sur le même trajet, un taxi le prenant à bord à un nœud d'origine à une heure souhaitée dans le futur (on imagine des délais d'au moins un quart d'heure ou une demi-heure jusqu'à quelques heures entre l'appel et l'heure souhaitée de rendez-vous).

Le dispatcheur doit donc trouver le taxi en service susceptible de répondre à cette demande à l'heure dite, ou dans une fourchette de temps limitée après cette heure idéale. Un certain nombre d'autres contraintes sur lesquelles nous reviendrons lors de la formulation précise de ce problème doivent aussi être satisfaites. Tout ce processus de décision est basé sur la connaissance actuelle de l'état du système, connaissance supposée complète au niveau du dispatching, et sur les projections dans le futur qui peuvent être faites.

Si la demande de réservation peut finalement être satisfaite, une heure prévisionnelle de rendez-vous est annoncée au client. Sinon le client est refusé (ou bien il est supposé refuser lui-même une offre ne satisfaisant pas toutes les contraintes, ce qui revient au même du point de vue de la simulation).

Dès son arrivée au nœud de rendez-vous, le client cherche si son taxi est présent au même nœud. On peut évidemment simuler un retard aléatoire du client par rapport à l'heure de rendez-vous prévue, et le taxi lui-même peut arriver en retard. Une heure de dépassement limite de cette heure de rendez-vous prévue est supposée connue du client et du taxi. Le premier arrivé (client ou taxi) attendra l'autre jusqu'à cette heure limite. Au delà, le client abandonnera l'attente et quittera le système. Dans le cas du taxi, il quittera de la même façon le nœud pour poursuivre son itinéraire, à supposer que cet itinéraire n'est pas vide.

**V.2.2. Véhicules.** Les taxis ont pour mission de suivre l'itinéraire défini et redéfini constamment par le dispatching pour déposer les passagers déjà à bord et prendre en charge les clients ayant un rendez-vous. Aux nœuds de rendez-vous, les taxis attendent éventuellement les clients à prendre en charge jusqu'à l'heure limite connue d'eux et des clients concernés. Au delà de cette heure limite, si le client n'est pas trouvé, le taxi poursuit son itinéraire et le dispatching doit alors redéfinir celui-ci (puisque la destination de ce client manqué doit être éliminée de l'itinéraire du taxi).

Si un véhicule se trouve momentanément sans passager ni rendez-vous, le dispatching doit lui assigner un nœud de stationnement avec une heure limite d'attente. Ce véhicule sera réactivé dès qu'il se verra attribuer un nouveau rendez-vous, ou bien de toutes façons après l'heure limite de stationnement.

Chaque taxi commence et termine son service à des instants déterminés. Ces heures de service sont prises en compte dans l'attribution des rendez-vous aux taxis. Bien évidement, l'heure de sortie du service ne pourra être respectée que si le taxi est vide à cette heure.

**V.2.3. Nœuds du réseau.** À chaque nœud du réseau, on doit gérer deux files d'attente : celle des clients en attente de leur taxi et celle des taxis en attente de leurs rendez-vous.

**V.2.4. Dispatching et serveurs.** Le dispatching central est constitué d'un certain nombre de dispatcheurs humains ou de serveurs informatiques permettant de gérer les appels (ou requêtes télématiques) des clients. Le nombre de dispatcheurs est évidemment, comme pour le nombre de taxis mis en service, un paramètre du dimensionnement du système qu'il convient d'optimiser par des simulations.

Le rôle des dispatcheurs est de répondre aux requêtes en trouvant le taxi le plus apte à répondre à chacune ou bien de rejeter la requête si certaines contraintes ne peuvent être satisfaites. Dans le cas où un taxi a pu être assigné à la demande, une heure de rendez-vous prévisible est annoncée au client et l'itinéraire du taxi est redéfini. On suppose que le dispatching a, à chaque instant, une vision complète et exacte de tous les taxis, leur position, leur planning, etc.

Le dispatching doit aussi intervenir pour modifier l'itinéraire d'un taxi au cas où celui-ci doit quitter un nœud sans avoir pu rencontrer le client avec lequel il avait un rendez-vous à ce nœud, ce qui se produit au delà d'une certaine heure limite d'attente connue du client et du taxi (il se peut que ce soit le taxi qui soit en retard et que le client ait déjà abandonné ou bien l'inverse).

Lorsqu'un appel arrive au dispatching, il est placé dans une file d'attente FIFO s'il ne peut être traité immédiatement. Cependant il ne restera dans la file que pour une durée limitée. La file d'attente des appels est donc une des ressources à gérer dans la simulation ainsi que la file d'attente des dispatcheurs disponibles à chaque instant.

## V.3. Modélisation de la gestion centralisée par la construction des événements

Comme on vient de le voir, par rapport à la gestion décentralisée du chapitre III, la gestion centralisée nécessite l'introduction de nouveaux agents et évidemment aussi de nouveaux types d'événements. L'objet de cette section est donc de décrire 15 types d'événements qui nous ont permis de modéliser ce mode de fonctionnement du système. Comme pour la gestion décentralisée, on les a regroupés en catégories en considérant l'agent déclencheur de chaque type d'événement.

### V.3.1. Événements déclenchés par les clients.

V.3.1.1. *Type 1 : apparition client.* Cet événement décrit le comportement du client lors de l'apparition de son appel à la centrale. Soit l'appel peut être traité immédiatement, auquel cas un dialogue ayant une certaine durée commence, soit cet appel est placé dans une file d'attente avec une heure limite.

V.3.1.2. *Type 2 : client quitte l'attente au dispatching.* Cet événement se produit quand le client en attente au dispatching a atteint une heure limite sans que son appel ait pu être traité avant. Dans ce cas le client quitte le système sans avoir pu déposer sa demande.

V.3.1.3. *Type 3 : client arrive au nœud de rendez-vous.* Lorsque le client arrive au nœud de départ, éventuellement avec un retard, il vérifie si son taxi est présent à ce nœud. Dans ce cas, le client indique qu'il doit embarquer et on aura la création d'un événement de type 19 (cf. §V.3.1.6). Si le taxi n'est pas là, le client se placera dans une file d'attente jusqu'à une heure limite, sauf si l'heure limite de rendez-vous est déjà atteinte, auquel cas il sortira immédiatement du système.

V.3.1.4. *Type 4 : client quitte l'attente au nœud de rendez-vous.* C'est l'événement qui se produit lorsque le client en attente de son taxi atteint l'heure limite d'attente sans avoir rencontré ce taxi.

V.3.1.5. *Type 23 : client annule rendez-vous.* On a envisagé le cas où un client ayant déjà appelé et obtenu un rendez-vous rappelle le dispatching pour annuler ce rendez-vous. Si son appel n'est pas traité immédiatement, il se placera dans la file d'attente des appels avec une heure limite, ce qui conduit à la création d'un événement de type 2 (cf. §V.3.1.2)

V.3.1.6. *Type 19 : un ou plusieurs clients embarque(nt) dans un taxi.* Cet événement marque la *fin* de l'embarquement d'un ou plusieurs clients dans un taxi. L'heure de cet événement a été calculée au moment de la création de cet événement en tenant compte du nombre de personnes à embarquer (cas d'un rendez-vous avec un groupe de clients).

**Remarque.** On peut se demander pourquoi on a attribué un type d'événement différent (ci-dessus type 19, mais type 18 pour les cas de la gestion décentralisée — voir §III.3.1.3) pour, en fait, le même événement de fin d'embarquement. Effectivement, le traitement est pratiquement le même, mais on a distingué, dans la simulation de la gestion mixte, la classe des clients sans réservation et celle des clients avec réservation.

Par contre, les taxis, dans la gestion mixte, sont supposés servir les deux types de clients et c'est pourquoi, dans la suite, on retrouvera des types d'événement déjà apparus dans la gestion décentralisée. Il faut cependant adapter leur traitement pour la gestion centralisée ou mixte. ♦

### V.3.2. Événements déclenchés par les taxis.

V.3.2.1. *Type 14 : véhicule se met en service.* Il s'agit d'adapter le traitement de ce type d'événement déjà décrit pour la gestion décentralisée (voir §III.3.2.1). Lorsque le véhicule se met en service à un nœud (sa position initiale), il cherche s'il a déjà des clients en rendez-vous à ce nœud. Sinon il part pour aller à l'endroit d'autres rendez-vous programmés. Si son itinéraire est vide, il doit recevoir une instruction sur son nœud de stationnement.

V.3.2.2. *Type 15 : véhicule termine son service.* Le traitement de cet événement est analogue à celui déjà décrit au §III.3.2.2.

V.3.2.3. *Type 11 : véhicule arrive à un nœud.* Il s'agit à nouveau d'adapter le traitement de ce type d'événement décrit au §III.3.2.4 au cas de la gestion centralisée. La première opération à faire, comme dans l'approche décentralisée, est de faire débarquer les passagers arrivés à destination à ce nœud, s'il y en a. Dans ce cas, un événement de type 13 (voir V.3.2.4) est créé. Sinon, et si le taxi a des rendez-vous à ce nœud, il vérifie si les clients concernés sont présents. Dans ce cas, un événement de type 19 (voir V.3.1.6) est généré. Si certains clients sont absents, mais si l'heure limite de rendez-vous n'est pas dépassée, le taxi se mettra en attente et un événement de type 8 (voir V.3.2.5) sera créé. Si l'heure limite est dépassée ou si le taxi n'a pas de rendez-vous à ce nœud, il part, sous la condition que son itinéraire ne soit pas vide, et un nouvel événement de type 11 sera créé pour marquer son arrivée au prochain nœud. Si l'itinéraire du taxi est vide, et si son heure de fin de service n'est pas atteinte ou dépassée (auquel cas il faut le faire sortir du système), on doit lui assigner un nœud de stationnement. Si ce nœud est sa position actuelle, il y aura la création d'un événement de type 10 (voir §V.3.2.6) pour marquer la fin de son stationnement. Sinon il quitte ce nœud pour aller stationner ailleurs et de nouveau un événement de type 11 est créé.

V.3.2.4. *Type 13 : véhicule fait sortir des passagers.* Le traitement est identique à celui de la gestion décentralisée (voir §III.3.2.4).

V.3.2.5. *Type 8 : véhicule interrompt l'attente de clients absents.* Lorsque le taxi ne trouve pas un client à son rendez-vous, il se met en attente jusqu'à l'heure limite de rendez-vous. Cet événement marque la fin de cette attente si le client ne s'est pas présenté avant l'heure limite.

V.3.2.6. *Type 10 : véhicule vide quitte son stationnement.* Le traitement est analogue à celui du §III.3.2.6. Cet événement oblige à se reposer la question du nœud de stationnement si un taxi vide en stationnement n'a pas reçu de nouvelle mission avant l'heure limite de stationnement et si son heure de fin de service n'est pas atteinte ou dépassée. Dans ce dernier cas, le taxi sort du système.

### V.3.3. Événements déclenchés par les dispatcheurs.

V.3.3.1. *Type 6 : dispatcheur se met en service.* Le dispatcheur devient opérationnel au dispatching.

V.3.3.2. *Type 7 : dispatcheur sort du service.* Le dispatcheur cesse d'être opérationnel mais doit terminer au préalable la tâche en cours. Ceci se traduit donc par un changement d'état qui sera pris en compte à la fin de la tâche en cours.

V.3.3.3. *Type 5 : dispatcheur termine le traitement d'un appel.* C'est l'événement de fin de traitement d'un appel. Si l'appel concerne une nouvelle demande, le dispatcheur donne la réponse au client, réponse positive ou négative selon qu'il a pu ou non satisfaire sa demande. Dans le cas d'une réponse positive, il met à jour le nouvel état des agents concernés (client, taxi affecté à cette demande). Dans le cas d'une réponse négative, le client sort du système. Si le client a appelé pour annuler un rendez-vous, le dispatcheur fait le nécessaire pour modifier l'itinéraire du véhicule correspondant. Finalement, le dispatcheur devient disponible pour traiter une nouvelle demande sauf si son heure de fin de service est atteinte ou dépassée.

**V.3.4. Enchaînement des événements.** Comme au §III.3.3, on décrit ici comment le traitement de chaque type d'événement engendre éventuellement d'autres événements à une date ultérieure. La figure V.1 représente les enchaînements possibles sous forme de graphe. Les

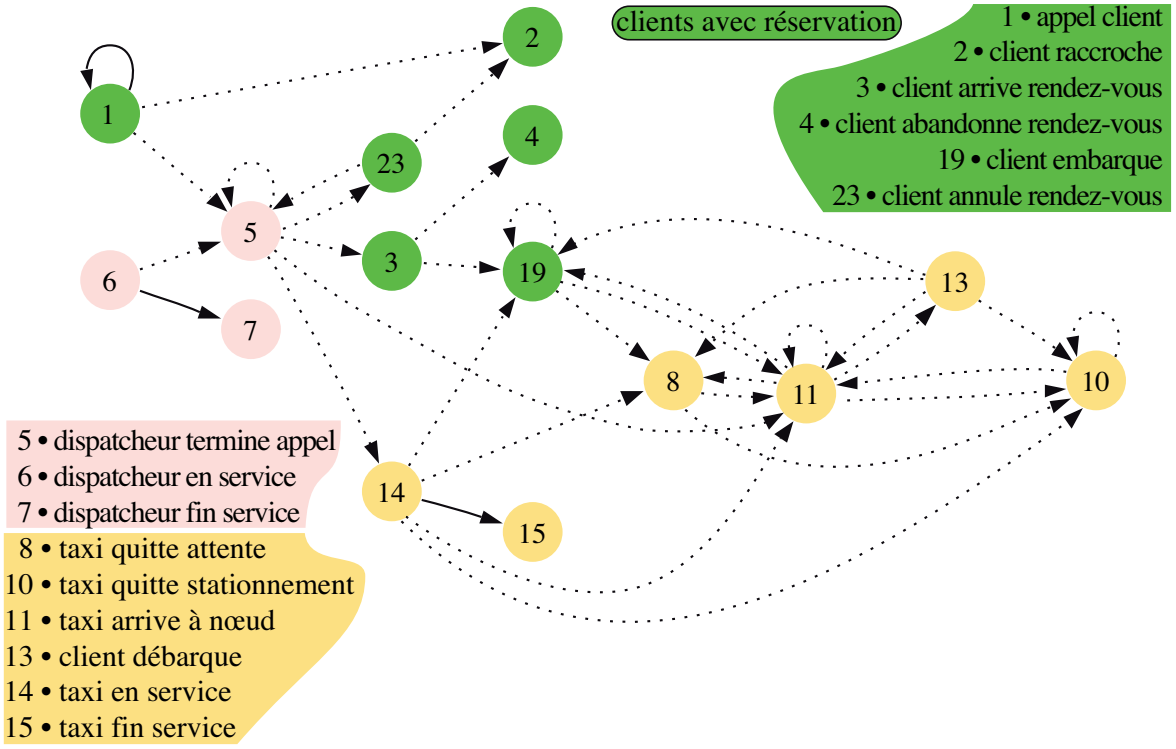


FIGURE V.1. Génération d'événements par d'autres événements (gestion centralisée)

arcs en trait continu indiquent la création d'événements qui suivent *nécessairement* l'événement d'origine et ceux en pointillés indiquent une *éventuelle* génération d'événements qui peut dépendre de l'état du système et des décisions prises.

V.3.4.1. *Type 1 : apparition client.* L'arc de 1 vers lui-même indique qu'un tel événement pour un client ayant comme origine un nœud du réseau engendre l'événement d'appel du client suivant pour le même nœud d'origine.

- Si l'appel du client peut tout de suite être traité par un dispatcheur, un dialogue commence, et donc il faut créer un événement de type 5 (cf. §V.3.3.3) correspondant à la fin du dialogue entre les deux agents. De plus le dispatcheur est marqué comme "occupé".
- Si aucun dispatcheur n'est disponible, le temps d'attente maximal du client est déterminé et un événement de type 2 (cf. §V.3.1.2) est généré pour marquer la fin d'attente du client. Le client est alors placé dans la file d'attente des appels à traiter.

V.3.4.2. *Type 2 : client quitte l'attente au dispatching.* Si cet événement a lieu, c'est-à-dire si le client est toujours dans la file à la date de l'événement et s'il n'est pas occupé dans un dialogue, le client sort du système. Par conséquent, aucun nouvel événement n'est engendré. Il faut seulement remettre à jour l'état du système (notamment celui de la file d'attente des appels). Si le client est en dialogue avec un dispatcheur, on modifie un indicateur d'état pour que le client sorte du système à la fin du dialogue si celui-ci se termine par un refus. Dans ce cas, c'est le traitement de l'événement de type 5 (cf. §V.3.3.3) qui fera effectivement sortir le client du système.

V.3.4.3. *Type 3 : client arrive au nœud de rendez-vous.* Dès son arrivée (même avec du retard), le client recherche son taxi.

- Si le véhicule est présent et sans être occupé, le client va embarquer, et par conséquent un événement de type 19 (cf. §V.3.1.6) est généré. Si le véhicule est en train de faire d'autres opérations, le client lui indique qu'il est prêt à entrer et se met dans la queue.

– Si le véhicule est absent, le client devra attendre au nœud au maximum jusqu'à l'heure limite du rendez-vous. Dans ce cas, un événement de type 4 (cf. §V.3.1.4) doit être créé pour marquer la fin de l'attente. On doit aussi mettre à jour l'état de la file d'attente.

V.3.4.4. *Type 4 : client quitte l'attente au nœud de rendez-vous.* Si cet événement se réalise, c'est-à-dire si le client est toujours en attente à l'heure de l'événement, le client sort du système et le traitement n'engendre aucun autre événement. Il suffit de mettre à jour l'état du système (en l'occurrence la file d'attente au nœud correspondant du réseau).

V.3.4.5. *Type 23 : client annule rendez-vous.* Lorsqu'un client rappelle pour annuler un rendez-vous précédemment attribué (on verra que cet événement de type 23 ne peut se rencontrer que dans ce cas),

– si cet appel est traité immédiatement, un événement de type 5 (cf. §V.3.3.3) est créé pour indiquer la fin de l'appel ;
– dans le cas contraire, le client est placé dans la file d'attente des appels avec une heure limite d'attente et un événement de type 2 (cf. §V.3.1.2) est créé pour indiquer la fin de son attente ; ou bien il quitte le système immédiatement si le temps d'attente décidé est nul. Dans ce dernier cas, aucun autre événement n'est créé mais le client ne se présentera pas de toutes les façons à son rendez-vous.

V.3.4.6. *Type 19 : un ou plusieurs clients embarque(nt) dans un taxi.* Cet événement marque la fin de l'embarquement d'un client ou d'un groupe de clients. On met à jour l'état du taxi après cette opération.

– Si d'autres clients sont également prêts à embarquer, un autre événement de type 19 est créé.
– Sinon, il faut vérifier si le taxi peut partir ou non.
  – Si tous les clients avec lesquels il avait un rendez-vous à ce nœud ont embarqués ou si l'heure limite de rendez-vous est atteinte ou dépassée pour tous les clients absents, le véhicule part (en effet, le taxi n'est sûrement pas vide car il vient d'embarquer des clients). Par conséquent, un événement de type 11 (cf. §V.3.2.3) est créé pour indiquer l'arrivée au prochain nœud.
  – Dans le cas où le taxi doit attendre encore d'autres clients, il ne peut pas partir et un événement de type 8 (cf. §V.3.2.5) est créé avec une heure limite afin d'indiquer la fin de l'attente du véhicule.

V.3.4.7. *Type 14 : véhicule se met en service.* À partir de cet instant, le véhicule devient disponible au nœud choisi. Un événement de type 15 (cf. §V.3.2.2) est créé pour indiquer la fin de service.

– S'il a des rendez-vous prévus à ce nœud, il vérifie si les clients sont là. S'il trouve au moins un client présent, on crée un événement de type 19 (cf. §V.3.1.6) pour indiquer la fin d'embarquement du ou des client(s).
– Si un des clients est absent, et si l'heure limite de rendez-vous n'est pas atteinte ou dépassée, le taxi se met en attente et un événement de type 8 (cf. §V.3.2.5) est créé pour marquer la fin de cette attente.
– Si les rendez-vous sont à un autre nœud, il part et un événement de type 11 (cf. §V.3.2.3) est créé pour indiquer l'arrivée du véhicule au prochain nœud.
– Si le taxi n'a pas de rendez-vous, donc son itinéraire est vide, on doit décider d'un nœud de stationnement. S'il doit stationner à sa position actuelle, on crée un événement de type 10 (cf. §V.3.2.6) pour indiquer la fin de son stationnement.
– S'il doit stationner ailleurs, il part et un événement de type 11 (cf. §V.3.2.3) est créé pour indiquer l'arrivée du véhicule au prochain nœud.

V.3.4.8. *Type 15 : véhicule termine son service.* Si le véhicule n'a plus de tâches à assurer et il se trouve à un nœud, il sort du service. Sinon, son état est modifié pour sortir du service dès que ces conditions sont vérifiées.

V.3.4.9. *Type 11 : véhicule arrive à un nœud.*

- Lorsque le véhicule arrive à un nœud, il commence par vérifier s'il y a des passagers parvenus à destination. Dans ce cas, il y aura la création d'un événement de type 13 (cf. §V.3.2.4) pour indiquer la fin de l'opération de débarquement.
- Si le véhicule n'a pas de sorties à effectuer, il vérifie s'il a des rendez-vous avec des clients à ce nœud. Dans ce cas, il cherche à trouver le(s) client(s) concerné(s). Si il y a au moins un client présent, un événement de type 19 (cf. §V.3.1.6) est créé pour marquer la fin de l'embarquement de ces clients.
- Si aucun client n'est présent, le taxi vérifie s'il doit attendre les clients absents. Dans ce cas, il se met en attente et un événement de type 8 (cf. §V.3.2.5) est créé pour mettre fin à cette attente. Des mises à jour du système (véhicule et nœud) sont effectuées.
- Si le taxi ne doit plus attendre, une mise à jour de l'itinéraire du taxi est requise (pour tenir compte de l'absence d'un ou de client(s)) puis, si son itinéraire n'est pas vide, le taxi part et un événement de type 11 est créé pour indiquer l'arrivée au prochain nœud.
- Si le taxi a un itinéraire vide et si son heure de fin de service n'est pas dépassée, il faut lui assigner un nœud de stationnement. Si ce nœud est sa position actuelle, on crée un événement de type 10 (cf. §III.3.2.6) pour indiquer la fin du stationnement. Des mises à jour au nœud sont effectuées.
- Si le véhicule doit stationner à un autre nœud, il part et un événement de type 11 est créé pour indiquer l'arrivée au prochain nœud de son itinéraire.
- Si son itinéraire est vide et si son heure de fin de service est dépassée, le taxi sort du système.

V.3.4.10. *Type 13 : véhicule fait sortir des passagers.* C'est l'événement de fin du débarquement des passagers. Après les mises à jour du système, il suffit de reprendre le traitement de l'événement de type 11 dans le cas où le véhicule n'a pas de sorties à effectuer.

V.3.4.11. *Type 8 : véhicule interrompt l'attente de clients absents.* Cet événement se réalise lorsque l'heure limite d'un rendez-vous est dépassée pour un taxi en attente d'un ou de client(s). Le(s) client(s) doi(ven)t être éliminé(s) de l'itinéraire du taxi (remise à jour).

- Si l'itinéraire du taxi n'est pas vide, le véhicule quitte le nœud et un événement de type 11 (cf. §V.3.2.3) est généré pour indiquer l'arrivée du véhicule au nœud suivant de son parcours.
- Si son itinéraire est vide et si son heure de fin de service n'est pas dépassée, il faut décider de son nœud de stationnement. Si le nœud choisi est sa position actuelle, un événement de type 10 (cf. §V.3.2.6) est créé.
- Si un autre nœud de stationnement est désigné, le taxi se met en route et un événement de type 11 est créé.
- Si son itinéraire est vide et si son heure de fin de service est dépassée, le taxi sort du système.

Une mise à jour est faite en particulier en ce qui concerne le nœud.

V.3.4.12. *Type 10 : véhicule vide quitte son stationnement.* Si le véhicule est encore en stationnement avec un itinéraire vide au moment de cet événement, on doit se reposer la question du nœud de stationnement, sauf si son heure de fin de service est atteinte ou dépassée, auquel cas le taxi sort du système.

- Si le nœud choisi est sa position actuelle, on crée à nouveau un événement de type 10 (cf. §III.3.2.6).
- Si on a choisi un autre nœud, le véhicule part et un événement de type 11 (cf. §V.3.2.3) est généré pour indiquer l'arrivée du véhicule à son premier nœud de passage. Il faut également mettre à jour l'état du nœud suite au départ du véhicule.

V.3.4.13. *Type 6 : dispatcheur se met en service.* Lorsque un dispatcheur prend son service, un événement de type 7 (cf. §V.3.3.2) est généré pour indiquer la fin de son service. D'autre part, des mises à jour sont faites au dispatching pour indiquer le nouvel état des serveurs.

– S'il y a des appels en attente, le dispatcheur prend le premier appel. Un événement de type 5 (cf. §V.3.3.3) est créé pour indiquer la fin du dialogue.
– Si aucun appel n'est en attente, le dispatcheur se met dans la file d'attente des serveurs disponibles.

V.3.4.14. *Type 7 : dispatcheur sort du service.* À la date de cet événement, le dispatcheur est marqué comme non disponible. Il sort immédiatement du service s'il n'est pas occupé et une mise à jour est faite au dispatching. Sinon un indicateur d'état signale qu'il doit sortir du système dès qu'il ne sera plus occupé.

V.3.4.15. *Type 5 : dispatcheur termine le traitement d'un appel.* Cet événement marque la fin d'un dialogue.

– Si l'appel concernait une demande pour un nouveau rendez-vous et si la réponse est positive, des mises à jour sont faites pour le client et le véhicule concerné. C'est à ce moment que l'on décide si le client fera un nouvel appel pour annuler ce rendez-vous. Si la réponse est positive, un événement de type 23 (cf. §V.3.1.5) est généré.
– Sinon, un événement de type 3 cf. §V.3.1.3) est créé pour indiquer l'arrivée du client au nœud de son rendez-vous.
– Si la réponse à la requête est négative, le client sort du système. Cela correspond à une simple remise à jour de l'état du système.
– Si le véhicule affecté au rendez-vous est en stationnement, on crée un événement de type 11 (cf. §V.3.2.3) car le véhicule quitte son stationnement pour se diriger vers le nœud du rendez-vous, à moins que le nœud de rendez-vous ne soit précisément le nœud de stationnement, auquel cas il suffit de faire sortir le taxi de la queue.
– Si on décide de mettre un nouveau véhicule en service, on crée un événement de type 14 (cf. §V.3.2.1).
– Si l'appel concerne l'annulation d'un rendez-vous, on doit remettre à jour l'état du véhicule. Si le véhicule était justement en attente de ce seul client, avec d'autres passagers à bord, on le fait partir et un événement de type 11 (cf. §V.3.2.3) est créé pour indiquer l'arrivée du véhicule au prochain nœud.
– Si le véhicule était en attente de ce seul client, sans autres passagers à bord, on doit décider de son nœud de stationnement : s'il doit stationner à sa position actuelle, un événement de type 10 (cf. §V.3.2.6) est créé pour indiquer la fin de son stationnement.
– S'il doit stationner à un autre nœud, il quitte sa position actuelle et un événement de type 11 est créé.
– Dans toutes les autres configurations, il suffit de mettre à jour l'itinéraire du véhicule.

Lorsque le dispatcheur termine le dialogue, il cherche s'il y a d'autres appels en attente, sauf si son heure limite de service est atteinte ou dépassée, auquel cas il sort du système (mise à jour de l'état du dispatching). Si un autre appel est à traiter, un nouvel événement de type 5 est créé. Sinon le dispatcheur se met dans la file d'attente des serveurs disponibles (mise à jour de cette file).

**Remarque.** À partir du moment où on a mis plusieurs dispatcheurs en service et que le temps de traitement d'un appel — qui comprend le temps d'exécution d'un algorithme pour choisir quel taxi répondra à cette demande, si c'est possible — n'est pas nul, il est possible que deux dispatcheurs ayant travaillé en parallèle aient choisi le même taxi et veuillent modifier son état de façon contradictoire. À cette difficulté, nous pouvons imaginer donner la solution suivante : à la fin du déroulement de l'algorithme, si un taxi a été choisi pour répondre à la demande en cours de traitement, le dispatcheur doit tester si l'état du taxi choisi a été modifié entre le début et la fin du traitement. Si c'est le cas, l'algorithme doit être relancé (ce qui donnera lieu à la création d'un nouvel événement de type 5) en prenant en compte l'état modifié du taxi, ce qui peut aboutir éventuellement à en choisir un autre. ♦

## V.4. Gestion mixte

Dans ce type de gestion, on suppose que les clients sans réservation de la gestion décentralisée (cf. chapitre III) et ceux avec réservation de la gestion centralisée décrite dans ce chapitre jusqu'à maintenant cohabitent. Il y a donc deux catégories de clients mais une seule catégorie de taxis qui servent les deux catégories de clients indistinctement.

**V.4.1. Clients.** Les clients des deux catégories doivent être distingués car ils ne sont pas traités de la même façon. Ceci nous oblige aussi à distinguer les types d'événements relatifs à ces deux catégories même quand ces événements sont similaires.

**V.4.2. Véhicules.** Les taxis sont pilotés par le dispatching comme dans la gestion centralisée et ils reçoivent des rendez-vous de clients ayant fait une réservation. Mais en chemin, ils peuvent rencontrer des clients sans réservation. Dans la mesure où les rendez-vous déjà assignés aux taxis et le sort des passagers déjà à bord le permettent, car les clients avec rendez-vous sont toujours prioritaires, ces clients peuvent être également acceptés. Le dispatching est supposé en permanence informé de l'état et de l'itinéraire prévu des taxis, pour ne pas dire qu'il gère aussi directement les décisions d'acceptation des clients sans réservation lorsqu'une rencontre taxi-client a lieu.

Le choix des points de stationnement des taxis vides est également géré par le dispatching qui a une vision globale de l'état du système et éventuellement des files d'attente des clients sans réservation aux nœuds du réseau.

**V.4.3. Nœuds du réseau.** Aux nœuds du réseau, on peut distinguer la file d'attente des clients sans réservation en attente d'un taxi de passage, et celle des clients ayant un rendez-vous et qui attendent le taxi qui leur a été assigné.

On peut aussi distinguer deux types de files d'attente pour les taxis : celle qui correspond à des taxis en attente d'un client avec rendez-vous, et celle des taxis vides en stationnement.

**V.4.4. Dispatching et serveurs.** Cette ressource fonctionne comme cela a été décrit au §V.2.4. Mais, comme on l'a déjà dit, le dispatching est aussi informé des conséquences des rencontres des taxis avec les clients sans réservation. On pourrait même modéliser la requête formulée au dispatching par le chauffeur de taxi concernant l'acceptation/refus d'un tel client au moment de cette rencontre de la même façon qu'on a modélisé l'appel des clients cherchant à faire une réservation. On peut aussi supposer que la décision est prise localement par le taxi (disposant d'un algorithme d'aide à la décision) qui tient compte néanmoins de ses rendez-vous connus et du sort de ses passagers, à condition que son nouvel itinéraire soit ensuite communiqué au dispatching.

Il faut noter ici aussi, comme à la remarque en fin de §V.3.4.15, que puisque le taxi et le dispatcheur peuvent éventellement être amenés à modifier en parallèle l'itinéraire de ce taxi (le taxi parce qu'il a rencontré un client au bord du trottoir, le dispatcheur en traitant un appel), il faut empêcher ce genre de conflit. On donnera la priorité au taxi dans ce cas et comme dans le traitement de l'événement de type 5 du §V.3.4.15, le dispatcheur doit relancer son algorithme s'il constate que l'état du taxi a été modifié entre le début et la fin de l'exécution de l'algorithme.

**V.4.5. Enchaînement des événements dans la gestion mixte.** Les types d'événements nécessaires à la modélisation de la gestion mixte ne sont que la *réunion* des types d'événements rencontrés au chapitre III pour la gestion décentralisée et ceux rencontrés dans les sections précédentes du présent chapitre pour la gestion centralisée. Il en est pratiquement de même pour l'enchaînement de ces types d'événements qui a été décrit au §III.3.3 pour la gestion décentralisée (voir figure III.1) et au §V.3.4 pour la gestion centralisée (voir figure V.1). En prenant l'union des graphes représentés sur ces deux figures, on obtient le graphe de la figure V.2 à quelques exceptions près que nous allons commenter maintenant.

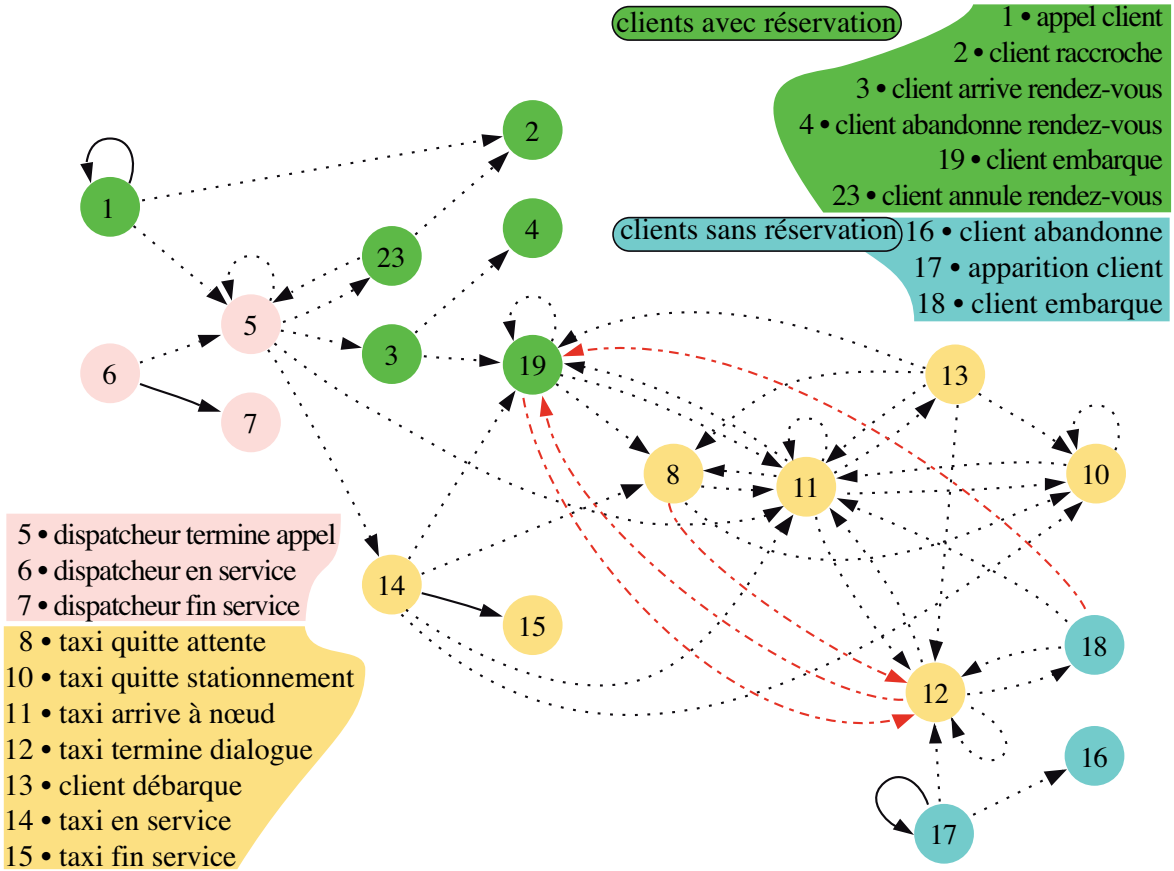


FIGURE V.2. Génération d'événements par d'autres événements (gestion mixte)

Ces exceptions sont au nombre de 4 et sont représentées sur la figure V.2 par les arcs de couleur rouge (si le document est en couleur) et avec un pointillé particulier (différent de la majorité des autres arcs en pointillé). Il s'agit plus précisément des arcs $8 \to 12, 12 \to 19, 18 \to 19, 19 \to 12$. Ces arcs ne pouvaient pas exister dans les deux graphes précédents car les extrémités de ces arcs ne figuraient pas toutes les deux simultanément dans un graphe ou dans l'autre.

V.4.5.1. *Type 8 : véhicule interrompt l'attente de clients absents.* Par rapport à la description de ce type d'événement telle qu'elle a été faite au §V.3.4.11, le taxi peut éventuellement maintenant, avant de se mettre en route (si son planning le permet et si son temps de service n'est pas épuisé) entrer en dialogue avec des clients sans réservation qui seraient en attente au nœud. Alors un événement de type 12 (cf. §III.3.2.5) est créé pour marquer la fin d'un tel dialogue.

V.4.5.2. *Type 12 : véhicule termine un dialogue avec un client (sans réservation).* Par rapport au traitement de cet événement qui a été décrit au §III.3.3.8, le taxi doit, à la fin du dialogue avec un client sans réservation, vérifier si un client avec rendez-vous n'est pas apparu pendant que ce dialogue se déroulait.

V.4.5.3. *Type 18 : client (sans réservation) entre dans un taxi.* De la même façon, par rapport au traitement de cet événement qui a été décrit au §III.3.3.3, le taxi doit, à la fin de l'embarquement d'un client sans réservation, vérifier si un client avec rendez-vous n'est pas apparu entre temps.

V.4.5.4. *Type 19 : client(s) (avec rendez-vous) entre(nt) dans un taxi.* C'est la situation symétrique de la liaison $12 \to 19$ traitée ci-dessus. Par rapport au traitement de l'événement 19 qui a été décrit au §V.3.4.6, le taxi doit, à la fin de l'embarquement d'un ou plusieurs client(s)

avec rendez-vous, vérifier si des clients sans réservation sont en attente d'un taxi. Dans ce cas (si l'état du taxi le permet), un dialogue commence avec le premier de ces clients, ce qui aboutit à la création d'un événement de type 12 (marquant la fin du dialogue).

### V.5. Sur le problème de décision au dispatching dans la gestion centralisée

**V.5.1. Position et difficulté du problème.** Dans cette section, on aborde le problème de la décision au dispatching dans le cadre de la gestion centralisée uniquement. Ce problème est beaucoup plus complexe que celui qui a été formulé au §III.4.2 pour les raisons suivantes.

- Dans le cas de la gestion décentralisée, un client rencontre un taxi avec un certain nombre de passagers à bord. Il n'y a donc pas de choix à opérer à propos des agents impliqués (client, taxi). Dans le cas de la gestion centralisée, l'appel du client parvient au dispatching qui doit choisir le taxi qui répondra à cette demande au mieux. Il ne s'agit donc pas seulement de reconstruire l'itinéraire le meilleur possible pour *un* taxi bien précis, mais celui de plusieurs taxis candidats à répondre a priori à cette demande puis de comparer les performances de ces itinéraires.
- Cette opération doit se faire en projetant dans le futur l'état du système jusqu'à l'heure du rendez-vous souhaitée par le client. Ceci ne peut évidemment se faire qu'à partir des informations connues au moment de l'appel du client.
- Pour un taxi donné, la construction d'un itinéraire est plus compliquée. En effet, dans la gestion décentralisée, les points de passage obligés de l'itinéraire futur du taxi ne sont que des étapes de *débarquement* des passagers à bord et du client candidat à la montée. Il n'y a donc pas de contraintes de précédence a priori entre ces étapes. Dans la gestion centralisée, les étapes futures de l'itinéraire sont à la fois des rendez-vous pour embarquer des clients, et des points de débarquement pour les passagers déjà à bord d'une part, pour les clients qui seront embarqués sur rendez-vous d'autre part. Pour ces derniers, il y a évidemment une contrainte de précédence : il faut d'abord les embarquer avant de songer à les débarquer, et donc l'ordre de parcours des étapes de l'itinéraire n'est pas entièrement libre.
- De plus, dans la gestion décentralisée, dès que le taxi passe par un nœud correspondant à la destination d'un client ou passager, celui-ci a intérêt à débarquer car le plus tôt est le mieux. Il n'y a donc pas lieu de repasser plusieurs fois par un même nœud (sauf si le chemin optimal l'exige), autrement dit, si deux passagers ont la même destination, on peut les fusionner en une seule étape obligatoire pour le taxi. Ceci n'est plus vrai pour la gestion centralisée puisque les étapes peuvent être des points de montée et de descente. Par exemple, si un passager monte en $A$ et descend en $B$ alors qu'un autre monte en $B$ et descend en $A$, le taxi devra faire le parcours $A \to B \to A$ ou $B \to A \to B$. Il est donc indispensable de conserver une *paire* d'étapes séparément pour chaque passager futur même si ces étapes correspondent aux mêmes nœuds du graphe pour plusieurs de ces futurs ou actuels passagers (pour ceux déjà à bord, seule l'étape de descente reste dans la liste des étapes futures du taxi). Ceci augmente sérieusement la dimension du problème de construction de l'itinéraire d'un taxi donné.
- Enfin, dans la gestion décentralisée, le nombre de passagers à bord d'un taxi sur l'itinéraire futur prévu de celui-ci (nous ne parlons pas ici des clients susceptibles d'être rencontrés dans le futur mais pour l'instant inconnus) ne peut que diminuer puisque toutes les étapes prévues sont des points de débarquement. Dans la gestion centralisée, certains points sont des points d'embarquement et donc le nombre de passagers du taxi fluctue. Il ne doit cependant pas dépasser, à tout instant, la capacité du taxi, et ceci dépend de l'ordre dans lequel les points d'embarquement et de débarquement seront visités. C'est une autre contrainte qui pèse sur la solution.

Dans la suite, on va se contenter d'aborder l'ensemble de la question en la divisant en trois étapes distinctes.

**Présélection des taxis à examiner.** Idéalement, tous les taxis en service devraient potentiellement être considérés, mais ce n'est pas réaliste. Il faut donc opérer une présélection des taxis qui seront d'abord examinés, quitte éventuellement à élargir ensuite cette présélection si aucun des taxis d'abord présélectionnés ne convient.

**Calcul de l'itinéraire pour chaque taxi présélectionné.** Il faut redéfinir l'itinéraire de chacun des taxis examinés pour servir ses points de passage obligatoires :

- points de dépôt des passagers déjà à bord ;
- points de rendez-vous des futurs passagers déjà affectés au taxi mais pas encore embarqués ;
- points de dépôt de ces futurs passagers ;
- origine et destination du nouveau candidat.

Les itinéraires considérés doivent satisfaire toutes les contraintes de dates imposées en chacun des points de l'itinéraire. Ils doivent de plus ne jamais conduire à dépasser la capacité maximale en passagers du taxi. Si plusieurs itinéraires satisfont toutes ces contraintes pour un même taxi, on doit choisir le meilleur au sens d'un critère à définir.

**Choix du meilleur taxi.** Si, pour plusieurs des taxis examinés, on a pu trouver un itinéraire acceptable, il faudra choisir le meilleur taxi au sens d'un critère à définir.

### V.5.2. Présélection de taxis.

V.5.2.1. *Critère de présélection.* Tout taxi en service a, à chaque instant, une position actuelle, et généralement un itinéraire prévu (sauf s'il est en stationnement parce que vide et sans client affecté). Pour un taxi $\theta$, ces informations peuvent être représentées par une liste de couples

$$\left\{(n_k^\theta, t_k^\theta)\right\}_{k\in K^\theta} \tag{V.1}$$

où $K^\theta$ indexe les étapes obligatoires prévues de l'itinéraire du taxi, $n_k^\theta$ est le nœud de la $k^{\text{ième}}$ étape, et $t_k^\theta$ est le temps de passage (plus précisément d'arrivée) prévu à cette étape (les étapes sont supposées rangées dans l'ordre des temps croissants).

- Si le taxi se trouve à l'arrêt à un nœud à l'instant présent, qu'il soit en stationnement (à vide) ou bien en train d'effectuer une opération, ce nœud figure en tête de la liste avec la date d'arrivée à ce nœud.
- Si le taxi est en stationnement à vide, la liste se limite à cet item.
- Si le taxi est en train de circuler entre deux nœuds à l'instant présent, son prochain nœud de passage [1] figure en tête de liste avec la date prévue d'arrivée à ce nœud.

Cette liste doit être constamment remise à jour, c'est-à-dire pratiquement à chaque fois qu'un événement d'un type quelconque, concernant le taxi en question, est traité dans la simulation.

À l'aide de ces listes, on peut envisager un principe de présélection des taxis pour répondre à un appel de client dont l'origine est le nœud $\nu$ et désirant partir à l'heure $h$ : on cherchera tous les taxis se trouvant dans un certain "voisinage espace-temps" de $(\nu, h)$. Pour définir la notion de "voisinage", on peut d'abord définir la notion de "distance" d'un taxi $\theta$ à un appel $(\nu, h)$. Pour cela, il faut rappeler la notation $\delta(m, n)$ qui donne, sur le réseau, la durée du trajet direct entre le nœud $m$ et le nœud $n$ (en utilisant le temps de parcours moyen des arcs). Alors, étant donné $(\nu, h)$, position de l'appel dans l'espace-temps, un taxi $\theta$ est à la "distance" $\Delta^\theta_{(\nu,h)}$ de cet appel si :

$$\Delta^\theta_{(\nu,h)} = \min_{k\in K^\theta} \left(\delta(n_k^\theta, \nu) + |t_k^\theta - h|\right). \tag{V.2}$$

L'idée de la présélection serait alors de retenir les taxis tels que leur distance à l'appel soit inférieure à un seuil $S$ (la valeur de ce seuil $S$ devrait ensuite être ajustée par des simulations).

1. c'est-à-dire celui qui se trouve à l'extrémité de l'arc sur lequel le taxi circule actuellement, mais pas nécessairement le premier nœud de passage *obligatoire*, c'est-à-dire le prochain nœud faisant partie de son itinéraire planifié ; ceci est nécessaire pour qu'il soit possible de redéfinir éventuellement un nouvel itinéraire pour aller chercher un nouveau client en insérant ce nouveau point de passage *avant* le premier nœud de l'itinéraire actuellement planifié. Autrement dit, le nœud en tête de la liste (V.1) est le premier nœud à partir duquel l'itinéraire du taxi peut être reconstruit.

V.5.2.2. *Discussion.* En dehors éventuellement du premier nœud de passage, seuls les nœuds représentant des points de passage *obligés* de l'itinéraire du taxi sont présents dans la liste. D'autres nœuds sont pourtant aussi visités par le taxi : ceux qui sont sur l'itinéraire le plus direct entre deux étapes successives dans l'itinéraire prévu. On peut espérer "récupérer" ces nœuds intermédiaires dans la mesure où on va chercher les taxis dans un "voisinage" de l'appel, pourvu que le seuil $S$ ne soit pas trop restrictif.

Un peu plus délicat est le cas d'un taxi en stationnement prolongé à un nœud parce que vide et sans affectation. Ce nœud de stationnement $n^\theta$ figure comme seul item dans la liste du taxi avec l'heure de mise en stationnement $t^\theta$. Le stationnement est assorti d'une heure limite $t^\theta_{\lim}$ : si aucun événement ne provoque la sortie du stationnement avant $t^\theta_{\lim}$, la situation du taxi ne sera systématiquement reconsidérée qu'à cette heure limite. Cependant, cela signifie que le couple $(n^\theta, t^\theta)$ ne sera pas remis à jour pendant toute la durée $t^\theta_{\lim} - t^\theta$. Ceci peut conduire à une évaluation peu réaliste de la distance à un appel $(\nu, h)$ avec $t^\theta \leq h \leq t^\theta_{\lim}$ : au lieu de $\delta(n^\theta, \nu) + |t^\theta - h|$ (que donnerait la formule (V.2)), on préférerait définir cette distance par $\delta(n^\theta, \nu)$ (le temps qu'il faut au taxi pour rejoindre le nœud de l'appel à partir de son nœud de stationnement en quittant immédiatement ce stationnement). Il faudra donc peut-être avoir un marquage spécial pour ces nœuds de stationnement.

Dans le même ordre d'idée, si $n^\theta$ est une étape obligée du taxi $\theta$ correspondant à un rendez-vous avec un client dont la demande a été affectée à ce taxi, on peut se demander s'il faut prendre en compte l'heure prévisionnelle $t^\theta$ d'*arrivée* à ce nœud $n^\theta$ ou bien s'il faut prendre en compte l'heure de *départ au plus tôt* de ce nœud compte tenu que l'heure de rendez-vous avec le client peut être *postérieure* à l'heure prévisionnelle d'arrivée. En fait, tout dépend de savoir si la nouvelle demande localisée au nœud $\nu$ sera servie *avant* ou *après* l'étape $n^\theta$, ce que l'on ne sait pas encore au stade de la présélection des taxis pour servir cette demande. On se contentera donc de la définition (V.2) de la "distance" pour cette présélection. La prise en compte de l'heure de départ au plus tôt du nœud $n^\theta$ sera effective lors du calcul d'itinéraire si ce taxi est examiné pour répondre à la nouvelle demande.

Comment réaliser pratiquement cette présélection d'une façon suffisamment rapide ? On peut réorganiser toutes ces informations concernant tous les taxis en service en les regroupant aussi par nœud. On aura alors, pour chaque nœud $\nu$, une liste

$$\left\{(\tau^\nu_j, t^\nu_j)\right\}_{j \in J^\nu} \tag{V.3}$$

où $\tau^\nu_j$ désigne l'ID $\theta$ du taxi dont le temps d'arrivée prévu au nœud $\nu$ est $t^\nu_j$. Alors, pour un appel $(\nu, h)$, on n'examinera que les taxis apparaissant dans les nœuds $\nu'$ tels que

$$\delta(\nu', \nu) \leq S \tag{V.4}$$

et on ne retiendra que les taxis $\tau^{\nu'}_j$ dont le temps d'arrivée prévu en $\nu'$ est tel que

$$j \in J^{\nu'} : \quad |t^{\nu'}_j - h| \leq S - \delta(\nu', \nu)\,. \tag{V.5}$$

Pour les taxis en stationnement à $\nu'$, la condition (V.4) devrait seule être prise en compte. Donc pour ces taxis, la paire $(\tau^{\nu'}_j, t^{\nu'}_l)$ pourrait par exemple être codée $(\tau^{\nu'}_j, -1)$, le $-1$ indiquant un taxi en stationnement.

**V.5.3. Calcul d'itinéraire pour un taxi.** Comme on l'a déjà au §III.4.2, le choix d'itinéraire nécessite de pouvoir

- d'une part, pour tout ordonnancement admissible des étapes obligatoires de cet itinéraire, propager les dates prévisionnelles d'arrivée aux étapes successives à partir de la position et de la date actuelle ;
- d'autre part, de calculer des dates limites de passage auxquelles sont soumises ces dates prévisionnelles, afin de valider ou de rejeter chacun des ordonnancements considérés ;

– et enfin, d'évaluer, par un critère ou fonction coût, tous les ordonnancements ayant satisfait à toutes les contraintes de dates limites de passage, afin de choisir le meilleur ordonnancement.

À la différence de la problématique du §III.4.2, il faut également ici surveiller l'évolution du nombre de passagers à bord du taxi le long de l'itinéraire planifié. On va donc détailler le déroulement de tous ces calculs en commençant par définir les notations dont certaines ont déjà été utilisées au §III.4.2.1.

V.5.3.1. *Notations.* On désigne par

$\mathcal{N}$**:** l'ensemble des nœuds du réseau ;

$\delta : \mathcal{N} \times \mathcal{N} \to \mathbb{N}$**:** (rappel) la matrice des durées des plus courts chemins en temps moyen de parcours sur le réseau ;

$t_a$**:** la date actuelle (date de traitement de l'appel d'un nouveau client) ;

$n_c^o \in \mathcal{N}$**:** le nœud du rendez-vous avec le nouveau client (d'indice $c$) ;

$n_c^d \in \mathcal{N}$**:** le nœud de destination du nouveau client ;

et pour un taxi $\theta$ considéré pour répondre à cet appel,

$L = \{n_0^\theta, n_1^\theta, \ldots, n_{M'}^\theta\} \subset \mathcal{N}$**:** la liste *ordonnée* des nœuds définissant l'itinéraire du taxi (avant prise en considération de la nouvelle demande) ; dans cette liste, le premier nœud $n_0^\theta$ est soit la position actuelle du taxi à $t_a$ (y compris si le taxi est en stationnement à vide en $n_0^\theta$, auquel cas $n_0^\theta$ est le seul élément de $L$), soit le premier nœud qu'atteindra le taxi s'il est, à $t_a$, en train de parcourir un arc ; autrement dit, $n_0^\theta$ est le premier nœud de l'itinéraire du taxi à partir duquel on peut envisager de modifier cet itinéraire ; les autres nœuds $n_1^\theta, \ldots, n_{M'}^\theta$ sont des points de passage obligés de cet itinéraire, c'est-à-dire soit des nœuds de destinations de passagers présents ou futurs, soit des nœuds de rendez-vous ; un même nœud du réseau peut figurer plusieurs fois dans la liste car, comme on l'a dit, on peut être amené à repasser plusieurs fois par le même nœud pour embarquer ou pour débarquer des clients ;

$t_0^\theta$**:** est la date d'arrivée à $n_0^\theta$ ; cette date est soit inférieure ou égale à la date actuelle $t_a$ si le taxi est déjà arrivé au nœud $n_0^\theta$, soit supérieur à $t_a$ si le taxi est en train de circuler sur l'arc qui aboutit à $n_0^\theta$ ;

$\ell \subset \mathcal{N}$**:** l'ensemble *non ordonné* des étapes à desservir (après $n_0^\theta$) si le candidat est accepté ; $\ell$ est donc constitué de toutes les destinations des passagers déjà à bord, de toutes les origines et destinations des rendez-vous déjà planifiés, et enfin de la paire origine-destination $(n_c^o, n_c^d)$ du candidat ; l'un des objets de l'algorithme est d'*ordonner* $\ell$ si le candidat est accepté ; le cardinal de $\ell$ est noté $M$ ;

$I$**:** l'ensemble des indices des clients du véhicule *et* du candidat $c$ ($I = I \cup \{c\}$) ; il s'agit de *tous* les clients concernant le taxi $\theta$, c'est-à-dire
  – les passagers déjà à bord : on désignera par $I_a$ l'ensemble des indices correspondants ;
  – les *futurs* passagers, c'est-à-dire les clients affectés au taxi mais pas encore embarqués : on désignera par $I_b$ l'ensemble des indices correspondants ;
  – et le passager potentiel d'indice $c$ ;

$n_i^o$**:** le nœud d'origine du client $i$ ;

$t_i^o$**:** la date à laquelle il a embarqué dans le taxi ;

$n_i^d$**:** le nœud de destination du client $i$ ;

$d : I \to \ell$**:** l'application qui sélectionne la destination des clients, c'est-à-dire que $\forall i \in I, n_i^d = d(i) \in \ell$ ;

$o : I \to \mathcal{N}$**:** l'application qui sélectionne l'origine des clients, c'est-à-dire que $\forall i \in I, n_i^o = o(i) \in \mathcal{N}$ ; si $i \in I_b \cup \{c\}$, alors $o(i) \in \ell$, mais ce n'est pas forcément le cas pour les passagers déjà à bord ($i \in I_a$) ;

$h_i^{\min}$**:** l'heure de rendez-vous du client $i$ : si le taxi arrive au nœud correspondant avant cette heure, il devra attendre jusqu'à cette heure au moins avant de pouvoir repartir avec ce client si celui-ci s'est présenté au rendez-vous ;

$h_i^{\max}$**:** l'heure *limite* de rendez-vous du client $i$ : si le taxi ne trouve pas le client au nœud correspondant avant cette heure, il devra attendre jusqu'à cette heure au plus avant de pouvoir repartir.

V.5.3.2. *Calcul des temps d'arrivée prévisionnels aux étapes d'un itinéraire.* Que ce soit pour la liste $L$ initiale avant acceptation du nouveau client (indice $c$) ou bien pour l'ensemble $\ell$ lui-même et *pour un ordonnancement donné et admissible* de ses étapes, on doit calculer les temps d'arrivée prévisionnels aux différents nœuds de ce parcours.

On considère donc ici une liste *ordonnée* $\mathcal{L} = \{n_0, \ldots, n_F\} \subset \mathcal{N}$ représentant soit la liste $L$, soit l'ensemble $\ell$ écrit avec un certain ordonnancement considéré. On démarre le calcul des temps prévisionnels d'arrivée aux différentes étapes de l'itinéraire avec la date $t(n_0)$ associée à $n_0$. Par rapport aux notations définies précédemment, on peut considérer que

$$t(n_0) = \max(t_a, t_0^\theta)\,. \tag{V.6}$$

De plus, si $\mathcal{L} = L$, il faut considérer que les domaines des applications $o$ et $d$ sont restreints à l'ensemble $I \setminus \{c\}$. L'équation récurrente est alors la suivante :

$$t(n_{j+1}) = \underbrace{\max\left(t(n_j), \max_{i \in o^{-1}(n_j)} h_i^{\min}\right)}_{t'(n_j)} + \delta(n_j, n_{j+1})\,, \quad j = 0, \ldots, F-1\,, \tag{V.7}$$

où $t'(n_j)$ est la date estimée de *départ* du nœud $n_j$. En effet, lorsque le taxi arrive au nœud $n_j$ à $t_j$, il ne peut repartir au mieux que lorsque l'heure de rendez-vous "au plus tôt" de tous les clients devant embarquer au nœud $n_j$ est atteinte.

V.5.3.3. *Suivi de l'occupation du taxi.* Comme on l'a dit plus haut, pour tout ordonnancement envisagé de l'itinéraire, il faut vérifier que le nombre de passagers à bord du taxi ne dépasse pas sa capacité. On doit donc suivre l'évolution de nombre de passagers à bord le long de l'itinéraire. Si $q(n_j)$ désigne le nombre de passagers du taxi lorsque celui-ci arrive au nœud $n_j$, $q(n_{j+1})$ sera le nombre de passagers à bord au départ du nœud $n_j$ comme à l'arrivée au nœud $n_{j+1}$. En notant $|\cdot|$ le cardinal d'un ensemble, on a

$$q(n_{j+1}) = q(n_j) - |d^{-1}(n_j)| + |o^{-1}(n_j)|\,, \quad j = 0, \ldots, F-1\,. \tag{V.8}$$

**Remarque.** Cette équation suppose que tous les passagers ayant un rendez-vous au nœud $n_j$ ont effectivement embarqué. Si le taxi est arrivé trop tard au nœud $n_j$ pour le client $i \in o^{-1}(n_j)$, c'est-à-dire si $t_j$, calculée par (V.7), est telle que $t_j > h_i^{\max}$, on pourrait considérer que le client $i$ a été "perdu" et modifier l'équation (V.8) en conséquence. Nous préférons cependant garder cette version car on s'interdit d'éliminer une étape de l'itinéraire pour cette raison du fait que le client concerné peut lui aussi arriver très en retard (au delà de $h_i^{\max}$) au nœud de rendez-vous. ♦

V.5.3.4. *Optimisation de l'itinéraire : formulation.* On définit maintenant le problème d'optimisation sous contraintes permettant de définir le meilleur itinéraire possible pour incorporer un nouveau candidat dans l'itinéraire d'un taxi ou bien pour prendre la décision que ce candidat ne peut pas être accepté par le taxi.

**Contraintes de précédence.** Un ordonnancement de $\ell$ est admissible si toutes les destinations des passagers sont desservies *après* leurs origines, ce qui est automatique pour les passagers $i \in I_a$ (passagers déjà à bord). On doit donc respecter les contraintes :

$$\forall i \in I_b \cup \{c\},\ \forall n_j \in o(i),\ \forall n_k \in d(i),\ n_j \text{ est rangé avant } n_k\,. \tag{V.9}$$

**Contraintes sur les dates prévisionnelles d'arrivée.** Pour un ordonnancement de $\ell$ satisfaisant toutes ces contraintes, on peut calculer les dates $t(n_j)$ prévisionnelles d'arrivée à chaque étape de ce parcours selon l'équation (V.7).

On peut utiliser le même algorithme pour calculer, si ce n'est déjà fait, les dates d'arrivée prévisionnelles au nœud de la liste $L$, c'est-à-dire *avant* d'avoir modifié l'itinéraire du taxi pour incorporer le candidat $c$. Ces dates seront notées $t^p(\cdot)$ à l'instar de la notation déjà utilisée au §III.4.2.1. Il faut rappeler les points suivants :

- l'ordre de parcours des nœuds dans le calcul des $t^p(\cdot)$ avec la liste $L$ n'est pas nécessairement le même que celui envisagé pour l'ensemble $\ell$, ce qui fait que la récurrence dans l'équation (V.7) ne se déroule pas nécessairement dans le même ordre que pour le calcul des $t(\cdot)$ concernant un ordonnancement envisagé pour $\ell$ ;
- dans la formule (V.7), les applications $o$ et $d$ sont restreintes au domaine $I \setminus \{c\}$ ;
- les dates $t^p(\cdot)$ ainsi obtenues n'étant définies que pour les nœuds apparaissant dans $L$, on complétera cette définition par

$$t^p(n) = +\infty \quad \text{si} \quad n \in \ell \quad \text{et} \quad n \notin L\,. \tag{V.10}$$

Les dates d'arrivée prévisionnelles sont soumises aux contraintes suivantes :

$$\forall n \in \ell, \quad t(n) \leq t^{\lim}(n)\,, \tag{V.11}$$

avec

$$t^{\lim}(n) = \max\left(t^p(n), \min\Big(\min_{i \in d^{-1}(n)} \big(t_i^o + s \times \delta(n_i^o, n_i^d)\big), \min_{i \in o^{-1}(n)} h_i^{\max}\Big)\right). \tag{V.12}$$

Dans cette formule, sont prises en compte :

- les dates limites $h_\cdot^{\max}$ de rendez-vous des clients à embarquer ;
- les dates prévisionnelles d'arrivée $t^p(\cdot)$ aux nœuds *avant* toute modification de l'itinéraire, car les contraintes imposées aux nouvelles dates prévisionnelles doivent pouvoir être au moins satisfaites si rien n'est changé dans l'itinéraire prévu ;
- les dates limites de dépose des clients à destination compte tenu d'un seuil $s$ de détour par rapport au trajet direct ; ces dates sont calculées en considérant l'heure $t_\cdot^o$ à laquelle les clients embarquent dans le taxi à leur nœud d'origine et la durée du trajet direct entre cette origine et la destination par le trajet direct, majorée par une coefficient $s > 1$ considéré comme le seuil de détour acceptable.

**Remarque.** À ce sujet se pose un nouveau problème. En effet, pour les indices $i \in I_a$ (passagers déjà à bord du taxi), la date $t_i^o$ est déjà connue. Par contre, pour les autres clients ($i \in I_b \cup \{c\}$), cette date (qui peut être assimilée en première approximation à la date prévisionnelle $t'(n_i^o)$ à laquelle le taxi quitte le nœud d'origine du client) dépend de l'ordonnancement de $\ell$. Autrement dit, la date limite $t^{\lim}(n_i^d)$ pour le nœud de destination d'un tel client ne peut être calculée que lorsque le calcul récursif (V.7) a déjà été amorcé et a atteint le nœud $n_i^o$ au moins. Comme ce nœud précède de toutes façons le nœud $n_i^d$, il n'y a pas de difficulté logique. Cependant, pour simplifier les calculs et pouvoir calculer les dates limites $t^{\lim}$ une fois pour toutes indépendamment de l'ordonnancement considéré de $\ell$, on peut éventuellement proposer de remplacer $t_i^o$ dans la formule (V.12) par $h_i^{\min}$ (heure minimale du rendez-vous pour le client $i$). On a alors la formule alternative :

$$t^{\lim}(n) = \max\left(t^p(n), \min\Big(\min_{i \in d^{-1}(n)} \big(h_i^{\min} + s \times \delta(n_i^o, n_i^d)\big), \min_{i \in o^{-1}(n)} h_i^{\max}\Big)\right). \tag{V.13}$$

En effet, $h_i^{\min}$ est connue à l'avance, indépendamment de l'ordonnancement de $\ell$ considéré. Ce faisant, l'interprétation de $s$ change : il s'agit d'une borne sur ce que nous avons appelé par ailleurs le *"détour total"* (voir §IV.3.2.5), c'est-à-dire le détour comptabilisé non pas à partir de l'entrée dans le véhicule, mais plutôt à partir de l'heure où le client est théoriquement prêt à partir.

– Dans le cas de la gestion *décentralisée*, l'écart entre l'heure d'apparition du client à son nœud origine (prêt à partir) et l'heure de montée dans le véhicule représente l'attente : le détour total incorpore donc cette attente initiale au numérateur du rapport avec le trajet direct.
– Dans le cas de la gestion *centralisée*, cette "attente" devient en fait, avec la formule (V.13), l'écart entre l'heure minimale de rendez-vous et l'heure d'embarquement (ou de départ du nœud origine) : en vérité, cette "attente" peut être aussi bien imputable à un retard du taxi qu'à un retard du client se présentant plus tard que cette heure minimale à son rendez-vous. Mais on peut retenir malgré tout cette définition du "détour total" indépendamment de la véritable raison de l'"attente" initiale.

♦

**Contraintes de capacité du taxi.** Pour tout ordonnancement considéré de la liste $\ell$, on peut calculer l'occupation du taxi à chaque étape en utilisant la récurrence (V.8). Cette occupation est soumise à la contrainte de ne jamais dépasser la capacité $C$ du taxi :

$$\forall n \in \ell \setminus \{n_0\}, \quad q(n) \leq C\,. \tag{V.14}$$

**Fonction objectif.** Sous les contraintes (V.9)–(V.11)–(V.14), on doit chercher un ordonnancement de $\ell$ qui minimise la fonction objectif suivante :

$$\sum_{n \in \ell \setminus \{n_0\}} |d^{-1}(n)| \times t(n)\,. \tag{V.15}$$

Comme pour le critère (III.3) du §III.4.2.2, on ne s'intéresse qu'aux dates d'arrivée prévisionnelles à des nœuds de l'itinéraire qui sont des nœuds de destination de clients et on pondère chacune de ces dates par le nombre de clients concernés. Ceci se justifie par le fait que les dates d'arrivée à des nœuds d'embarquement de clients sont déjà soumises à des contraintes et que le principal intérêt est de faire arriver les clients à destination le plus vite possible.

Si aucun ordonnancement ne permet de vérifier toutes les contraintes, alors on décide de ne pas accepter le candidat $c$, sachant que l'ordonnancement initial de $L$ doit satisfaire (pour cette liste) toutes les contraintes, y compris les contraintes (V.11) grâce à l'adjonction des $t^p$ dans la formule (V.12) (ou (V.13)).

**Le cas particulier des taxis vides.** Un taxi vide est soit en stationnement en un nœud $n_0$ depuis l'instant $t_0$ antérieur à l'heure $t_a$ de l'appel, soit en transit sur un arc aboutissant en $n_0$ qu'il atteindra à un instant $t_0$ postérieur à $t_a$ (mais $n_0$ n'est pas nécessairement le nœud qui avait été choisi pour le stationnement, c'est juste le prochain nœud de passage à partir duquel on peut éventuellement modifier son itinéraire). Dans tous les cas, on considère donc que le taxi est en $n_0$ à l'instant $t(n_0)$ donné par (V.6).

Pour ce taxi vide dans cette position à cet instant, les seules contraintes à vérifier sont
– qu'il peut atteindre le nœud $n_c^o$ avant $h_c^{\max}$, donc que

$$t(n_0) + \delta(n_0, n_c^o) \leq t^{\lim}(n_c^o) = h_c^{\max}\,; \tag{V.16}$$

la présélection de ce taxi vide par la règle (V.4) implique déjà que $\delta(n_0, n_c^o) \leq S$ ; donc si $S \leq h_c^{\max} - t(n_0)$, la contrainte (V.16) sera satisfaite ;
– qu'il peut atteindre le nœud $n_c^d$ avant

$$t^{\lim}(n_c^d) = \begin{cases} h_c^{\min} + s \times \delta(n_c^o, n_c^d) & \text{si on utilise la formule (V.13) ;} \\ t'(n_c^o) + s \times \delta(n_c^o, n_c^d) & \text{si on utilise la formule (V.12),} \end{cases}$$

avec

$$t'(n_c^o) = \max\left(t(n_0) + \delta(n_0, n_c^o), h_c^{\min}\right);$$

il faut donc vérifier que

$$t(n_c^d) = t'(n_c^o) + \delta(n_c^o, n_c^d) \leq t^{\lim}(n_c^d)\,,$$

ce qui est toujours vrai avec la deuxième option puisque $s \geq 1$, mais qui reste à vérifier avec la première option.

Si ces contraintes sont vérifiées, le candidat peut être accepté dans ce taxi vide et la valeur de la fonction objectif (V.15) est égale à $t(n_c^d)$.

**V.5.4. Choix du meilleur taxi.** Dans §V.5.3, nous avons donc essayé de formuler le problème du calcul de l'itinéraire "le meilleur" pour chacun des taxis qui ont été présélectionnés selon la technique du §V.5.2, calcul qui est indissociable de la décision de pouvoir ou non accepter le candidat à bord de ces taxis. Cependant, nous n'avons pas abordé la question de l'*algorithme* à employer pour résoudre le problème ainsi formulé. Cette question sera rediscutée plus loin (voir §V.5.5).

Dans l'éventualité où aucun des taxis présélectionnés ne peut accepter le candidat, faut-il élargir la présélection (en prenant une valeur de $S$ plus grande dans l'équation (V.5)) ou considérer que le candidat doit être définitivement refusé ?

Dans l'hypothèse où *plusieurs* taxis présélectionnés sont susceptibles d'accepter le candidat, comment choisir le "meilleur" taxi à lui affecter ? C'est l'objet de la discussion qui suit.

Intuitivement, le "meilleur" taxi à affecter à un nouveau client est celui dont l'itinéraire est le moins perturbé dans son planning initial par l'affectation de ce nouveau client. Dans la section V.5.3, on a comparé, pour un taxi $\theta$ donné, les ordonnancements possibles d'itinéraires en utilisant la fonction objectif (V.15). Cette fonction peut être évaluée

- *avant* l'affectation du nouveau client en utilisant les dates $t^p$ dont il a été question au §V.5.3.4 (et bien sûr l'ensemble $I \setminus \{c\}$ d'indices des clients) : notons $\alpha^\theta$ cette valeur ;
- et *après* le choix du meilleur itinéraire pour le même taxi si le nouveau client peut être accepté : notons $\beta^\theta$ cette valeur.

Le ratio de ces valeurs $\beta^\theta/\alpha^\theta$ serait une mesure possible de l'impact relatif de l'acceptation éventuelle du nouveau client dans le taxi $\theta$. Mais ce critère (que l'on chercherait à minimiser par le choix de $\theta$) tendrait à privilégier les taxis plutôt déjà bien remplis car le dénominateur du ratio serait plus élevé pour ceux-là.

On pourrait alors utiliser le même ratio mais en utilisant la fonction objectif (V.15) divisée (avant comme après acceptation) par le nombre de clients à déposer ($|I \setminus \{c\}| = |I| - 1$ avant, et $|I|$ après) : on considérerait donc le ratio précédent multiplié par $(|I| - 1)/|I|$, ce qui favoriserait maintenant les taxis plutôt peu chargés.

Il y a de toutes les façons une difficulté particulière avec les taxis vides pour lesquels $I \setminus \{c\}$ est vide et donc $\alpha^\theta = 0$. Faut-il privilégier ces taxis vides dès qu'il y en a un possible ? Et s'il y en a plusieurs, choisir celui qui peut arriver le plus tôt possible en $n_c^o$ ?

**V.5.5. Retour sur la question de la faisabilité.**

V.5.5.1. *Optimalité et faisabilité.* Le problème formulé au §V.5.3.4 pourrait sans doute être abordé par la programmation dynamique à l'instar de ce qui a été décrit au §III.4.2.3. Mais pour les raisons qui ont été discutées dans cette section là, et a fortiori ici où le problème est encore plus difficile et requiert un état de plus grande taille (il va falloir par exemple tenir compte de l'état supplémentaire introduit par l'équation (V.8)), il est peu probable que l'on parvienne à des temps de calcul acceptables par cette approche.

L'énumération exhaustive que nous avons envisagée au §III.4.2.4 risque elle aussi de poser des problèmes dans la mesure où le cardinal $M$ de l'ensemble $\ell$ à ordonner risque d'atteindre le double de la capacité $C$ des taxis dans le cas le plus défavorable où il faut inclure toutes les origines et les destinations de tous les futurs passagers et candidat dans cet ensemble.

Frédéric Meunier (LVMT-ENPC) a suggéré un problème plus simple que le problème formulé au §V.5.3.4 et qui pourrait plus facilement être abordé par la programmation dynamique. Ce problème consiste essentiellement à remplacer la fonction objectif (V.15) par une autre. Du fait que toutes les contraintes de la formulation précédente sont conservées, une solution de ce nouveau problème serait au moins une solution *admissible* du problème précédent, ou du moins

la "résolution" de ce nouveau problème permettrait de savoir s'il existe ou non une solution admissible (pour un taxi donné). L'optimisation plus ou moins exacte de (V.15) pourrait alors être conduite, dans cette hypothèse, par des méthodes heuristiques.

Nous donnons ici un aperçu de ce nouveau problème et de sa résolution par la programmation dynamique (développement dû essentiellement à F. Meunier).

V.5.5.2. *Faisabilité et programmation dynamique.* Comme on l'a dit, il s'agit simplement de changer la fonction objectif (V.15) en la remplaçant par

$$\max_{n\in\ell} t(n) \tag{V.17}$$

($t(n)$ obéit toujours à l'équation (V.7)), nouvelle fonction objectif qu'il s'agit de minimiser sous les mêmes contraintes que précédemment ; autrement dit, on cherche à terminer au plus tôt l'ensemble de la tournée du taxi représentée par l'ensemble $\ell$. Si le coût optimal est infini, c'est qu'il n'y a pas de solution admissible (c'est-à-dire pas d'ordonnancement de $\ell$ satisfaisant toutes les contraintes).

L'*état* requis pour écrire une équation de la programmation dynamique (en "temps" direct et non pas rétrograde) est le triplé $\big(n, j, \{k_l\}_{l\in\ell}\big)$ où

- $n$ est la position du taxi (numéro de nœud dans $\ell$),
- $j$ est le nombre de passagers dans le taxi à cette position,
- $k_l = 1$ si le nœud $l \in \ell$ a déjà été visité ; $k_l = 0$ sinon.

Les contraintes de précédence (V.9) se traduisent ici par un domaine admissible pour les états $\{k_l\}$ : si $l$ est un nœud de destination d'un futur passager ou du candidat, autrement dit si il existe $l' \in \ell$ tel que $l'$ soit l'origine de ce même futur passager ou candidat, alors si $k_l = 1$, nécessairement $k_{l'} = 1$ (la contrainte de précédence n'existe pas pour les passagers déjà à bord à la date $t(n_0)$ donnée par (V.6)).

La dynamique (rétrograde) de l'état est obtenue en définissant les *antécédents* d'un état $\big(n, j, \{k_l\}\big)$ donné : un état $\big(n', j', \{k'_l)\}$ précède $\big(n, j, \{k_l\}\big)$ si

- $j' = j + 1 \leq C$ et $n$ est un nœud de descente, ou bien $j' = j - 1 \geq 0$ et $n$ est un nœud de montée ;
- $k'_l = k_l$ pour $l \neq n$ et $k'_n = k_n - 1$.

On note $\mathcal{A}(x)$ l'ensemble des antécédents d'un état $x$ et $P$ le nombre de passagers présents à bord du taxi au moment $t(n_0)$ où le problème est considéré (le taxi se trouvant alors en $n_0$).

On appelle $V(x)$ la *fonction valeur* ou *fonction de Bellman*, c'est-à-dire ici le temps minimum dans lequel l'état $x$ peut être atteint. Cette fonction est infinie si l'état $x$ n'est pas atteignable en respectant les contraintes. La dynamique de l'état étant rétrograde, l'équation de la programmation dynamique sera écrite dans le sens direct. L'initialisation est la suivante :

- $V(n_0, P, \{0, \ldots, 0\}) = t(n_0)$ ;
- $V(n, P, \{k_l\}) = \begin{cases} t(n_0) + \delta(n_0, n) & \text{si } k_l = 0 \text{ pour } l \neq n \text{ et } k_n = 1, \\ \infty & \text{sinon ;} \end{cases}$
- $V(n, j, \{0, \ldots, 0\}) = \infty$ si $j \neq P$ ;

Avant d'écrire l'équation de la programmation dynamique, on redéfinit quelques notations. En chaque nœud $n$ de l'itinéraire, il y a un temps de passage *minimum* qui peut être nul ou supérieur ou égal à l'heure minimale $h_i^{\min}$ de rendez-vous de tout client $i$ tel que $o(i) = n$. Le temps minimum de passage en $n$ est donc $\max_{i\in o^{-1}(n)} h_i^{\min}$. Par ailleurs, il y a un temps de passage *maximum* qui est le temps $t^{\lim}(n)$ calculé de préférence par la formule (V.13). On notera plus simplement $\underline{h}(n)$ et $\overline{h}(n)$ ces bornes inférieure et supérieure requises pour l'heure de passage au nœud $n$.

L'équation de la programmation dynamique est écrite en deux étapes. On définit d'abord :

$$U(x) = \max\Big(\underline{h}(n), \min_{x'\in\mathcal{A}(x)} \big(V(x) + \delta(x', x)\big)\Big), \tag{V.18}$$

où $x = \big(n, j, \{k_l\}\big)$ et $x' = \big(n', j', \{k'_l\}\big)$. Puis

$$V(x) = \begin{cases} U(x) & \text{si } U(x) \leq \overline{h}(n), \\ \infty & \text{sinon.} \end{cases} \tag{V.19}$$

La réponse à la question de la faisabilité est positive s'il existe un état *terminal*, c'est-à-dire un état $x = \big(n, j, \{k_l\}\big)$ pour lequel $\{k_l\} = (1, \ldots, 1)$ et tel que $V(x) < \infty$.

CHAPITRE VI

# Conclusion

*Y pas d'grands, y a pas d'p'tits... la bonne longueur pour les jambes c'est quand les pieds touchent bien par terre.*

*Coluche*

Sans nécessairement chercher à les remplacer complètement, le système de taxis collectifs que nous avons cherché à étudier dans ce mémoire pourrait trouver sa place comme alternative à la voiture individuelle et au taxi traditionnel en milieu urbain et péri-urbain. Nous avons passé en revue dans le chapitre d'introduction tous les systèmes dits TAD (transport à la demande) qui existent ou ont été envisagés de par le monde, mais aucun ne ressemble exactement à ce que nous avons considéré ici.

Nous avons cherché à concilier la qualité de service du taxi ou de la voiture individuelle (levant même quelques inconvénients de cette dernière comme la nécessité de conduire soi-même ou de trouver des places de stationnement) en offrant un service porte-à-porte, tout en abaissant le coût par le partage du véhicule et l'accroissement de productivité du chauffeur par rapport à celui d'un taxi individuel qui passe beaucoup de temps à l'arrêt.

Pour autant, un tel système, libéré de la plupart des contraintes des transports collectifs (lignes, arrêts et horaires fixes), est difficile à évaluer, à gérer de façon optimale en temps réel, et à dimensionner correctement. De plus, il n'y a pas de référence de systèmes comparables déjà existants, et donc pas de retour d'expériences, notamment en ce qui concerne le mode de gestion décentralisé étudié aux chapitres III et IV. De toutes façons, devant la complexité du problème, il n'est pas souhaitable de se lancer dans des expérimentations de terrain sans étude préalable sérieuse. C'est ce qui nous a conduits à proposer la simulation comme outil d'évaluation et d'étude pour tester divers algorithmes de gestion et d'optimisation.

En fait, le système proposé peut s'imaginer sous deux ou trois formes différentes :

- le mode décentralisé où il n'existe aucune autorité centrale ayant une vue de l'ensemble du système ; comme pour le taxi individuel, le fonctionnement est basé sur la rencontre des clients et des taxis avec cette différence qu'un taxi n'est pas forcé d'accepter un client qui ne s'insère pas naturellement dans l'itinéraire prévu pour desservir les passagers déjà à bord ; néanmoins, une aide à la décision sous forme informatique peut évidemment être envisagée pour le chauffeur à bord de chaque véhicule ;
- le mode centralisé où toute la gestion (réservation des trajets, choix des taxis pour servir chaque demande, construction des itinéraires) passe par un dispatching central ; les technologies actuelles (télécommunications, GPS, etc.) permettent d'envisager ce mode ; cependant, il devient nécessaire, par rapport au mode décentralisé, d'implanter une infrastructure plus lourde à mettre en œuvre ; l'ambition de l'outil de simulation est de chercher à délimiter les cas où le surcoût de cette infrastructure se justifie par les gains de performances globales que l'on peut espérer d'une gestion mieux informée et plus intelligente ;
- le mode mixte ou les deux modes précédents cohabitent au sein du même système.

Malheureusement, nous n'avons pas eu le temps de mener à bien l'ensemble de ce projet. La conception du simulateur à événements discrets a effectivement été menée avec à l'esprit l'ensemble des modes de gestion ci-dessus. Ce que nous avons appelé la "partie mécanique" du simulateur (voir §II.3.1 et figure II.2) est réputée être capable de simuler le système dans tous

ses modes. Mais il faut admettre que le mode centralisé et a fortiori le mode mixte n'ont pas pu être testés aussi sérieusement que le mode décentralisé.

Mais la principale difficulté reste cependant la partie algorithmique qui est beaucoup plus complexe pour le mode de gestion centralisé que pour le mode décentralisé. Sans doute beaucoup d'heuristiques devront être testés et un énorme travail reste à faire dans cette direction. Mais c'est le mérite de disposer d'un simulateur que de savoir qu'on pourra étudier scientifiquement touts les idées qui pourront être envisagées.

Nous avons essayé de montrer dans ce travail comment la simulation peut être utilisée pour évaluer les performances d'un système (décentralisé) muni d'algorithmes de gestion. La simulation permet d'obtenir de nombreux résultats quantitatifs relatifs à la qualité de service offerte aux clients (attente, détours), aux coûts opérationnels (activité des taxis, distances parcourues, temps de parking) et au chiffre d'affaires (clients transportés, parcours "utiles" servis). Mais lorsqu'il s'agit de comparer des algorithmes, d'optimiser des paramètres, ou d'évaluer l'influence d'une configuration ou d'un niveau de demande, la profusion des résultats devient un handicap. Au chapitre IV, nous avons essayé de dégager une méthodologie permettant d'arriver à une vision synthétique et de faire des choix raisonnés, choix et compromis qui de toutes les façons restent de la responsabilité du gestionnaire du système, mais qui peuvent être quantitativement éclairés par l'outil de simulation.

Nous sommes conscients que notre projet reste à compléter sur de nombreux points.

- Il serait souhaitable d'améliorer l'ergonomie de l'outil pour l'utilisateur, avec notamment l'automatisation d'un certain nombre de tâches pour entrer les données d'une part, traiter les résultats d'autre part. Le travail de traitement notamment est resté pour nous relativement artisanal et la confection d'outils (graphiques en particulier) plus perfectionnés serait sans doute la bienvenue. Mais encore une fois, nous avons d'abord été préoccupés dans ce travail par la mise au point d'une méthodologie de traitement des résultats plus que par les outils informatiques.
- Il faudrait peut-être également améliorer l'outil informatique de simulation lui-même. Le choix du langage interprété Python a permis d'arriver relativement vite à un outil opérationnel qui a suffi à notre étude, mais le passage à un cas réel plus lourd à traiter exigerait éventuellement la reprogrammation du simulateur dans un langage compilé. Ceci risque d'être encore plus nécessaire pour la partie algorithmique dont nous avons senti les limites en termes de temps d'exécution comme cela été mentionné dans ce mémoire.
- Un travail important reste à faire sur le plan algorithmique pour les modes de gestion centralisé et mixte. Encore une fois, un tel système ne donnera sa mesure que s'il est bien géré.
- Enfin, l'étude méthodologique présentée dans ce mémoire ne devrait être que le préalable à l'étude d'un cas réel... en espérant que cela débouche finalement un jour sur une véritable expérimentation sur le terrain.

# Bibliographie


1. *ATA Plus rapide qu'un bus, moins cher qu'un taxi.*, `http://www.atafrance.com/`.
2. Hugues Bersini, *La programmation orientée objet*, Eyrolles, Paris, 2008.
3. *Bristol Dial-a-Ride.*, `http://www.bristoldialaride.org.uk/`.
4. Stephen L. Campbell, Jean-Philippe Chancelier, and Ramine Nikoukhah, *Modeling and simulation in SCILAB/SCICOS with ScicosLab 4.4*, Springer, 2010.
5. Elaine Chang, *Modèle de simulation d'un système de taxis collectifs*, Tech. report, CERMICS-ENPC, 2005.
6. *Taxis Chongal Taekshi*, `http://wiki.galbijim.com/Taxis`.
7. *Collecto Bruxelles Mobilité*, `http://www.bruxellesmobilite.irisnet.be/articles/taxi/comment-ca-marche`.
8. M. Dessouky, M. Rahimi, and M. Weidner, *Jointly optimizing cost, service, and environmental performance in demand-responsive transit scheduling*, Transportation Research Part D **8** (2003), 433–465.
9. R. Dial, *Autonomous dial-a-ride transit introductory overview*, Transportation Research Part C **5** (1995), 261–275.
10. M. Diana and M. Dessouky, *A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows*, Transportation Research Part B **38** (2004), 539–557.
11. P.H. Fargier and G. Cohen, *Study of a collective taxi system*, Proceedings of the Fifth International Symposium on the Theory of Traffic Flow and Transportation (University of California, Berkeley) (G.F. Newell, ed.), American Elsevier, June 16-18 1971, pp. 361–376.
12. L. Fu, *Simulation model for evaluating intelligent paratransit systems*, Transportation Research Record (2001), no. 1760, 93–99.
13. ———, *A simulation model for evaluating advanced dial-a-ride paratransit systems*, Transportation Research Part A **36** (2002), 291–307.
14. ———, *Analytical model for paratransit capacity and quality-of-service analysis*, Transportation Research Record (2003), no. 1841, 81–88.
15. L. Fu and Y. Xu, *Potential effects of automatic vehicle location and computer-aided dispatch technology on paratransit performance : a simulation study*, Transportation Research Board (2001), no. 1760, 107–113.
16. *Taxis Hapseung*, `http://wiki.galbijim.com/Taxis`.
17. M.E.T. Horn, *Multi-modal and demand-responsive passenger transport systems : a modelling framework with embedded control systems*, Journal of Software **36** (2002), no. 2, 167–188.
18. John S. Niles and Paul A. Toliver, *IVHS technology for improving ridesharing*, Annual Meeting of IVHS America (now ITS America) (Newport Beach, California), May 1992.
19. C. Kunaka, *Simulation modelling of paratransit services using GIS*, The Association for European Transport Conference, January 1996.
20. P. Lalos, A. Korres, C. K. Datsikas, G.S. Tombras, and K. Peppas, *A framework for dynamic car and taxi pools with the use of positioning systems*, 2009 Computation World (Athens, Greece), November 15-20 2009.
21. *London Dial-a-Ride*, `http://www.tfl.gov.uk/gettingaround/3222.aspx`.
22. *London taxi sharing scheme*, `http://www.tfl.gov.uk/modalpages/1144.aspx`.
23. Federico Malucelli, Maddalena Nonato, and Stefano Pallottino, *Demand adaptive systems : some proposals on flexible transit*, Operational Research in Industry (T.A. Ciriani et al., ed.), McMillan Press, London, 1999, pp. 157–182.
24. Klaus G. Müller, *SimPy book*, `http://simpy.sourceforge.net/simpy_book.htm`, 2008.
25. J. Peterson, *Petri net theory and the modeling of systems*, Prentice Hall, 1981.
26. *Proxibus*, `http://www.ville-chaville.fr/196/297/proxibus.html`.

27. *Bus sur appel-Proxibus*, http://www.tpg.ch/fr/horaires-et-reseau/bus-sur-appel/proxibus/index.php.
28. *Python programming language-official website*, http://www.python.org/.
29. *Regiotaxi Northwest part of province of Overijssel, Salland, Vechtdal*, http://www.ns.nl/cs/Satellite/travellers/about-trip/travelling-by-train/regional-taxi.
30. *SCICOS : Block diagram modeler/simulator*, http://www-rocq.inria.fr/scicos/.
31. *Aziz Senni, le taxi-brousse à la Française*, http://www.bladi.net/aziz-senni-le-taxi-brousse-a-la-francaise.html, 2004.
32. *SimPy simulation package homepage*, http://simpy.sourceforge.net/.
33. C.C. Tao, *Dynamic taxi-sharing service using intelligent transportation system technologies*, International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007) (Shanghai, China), September 21-25 2007, pp. 3209–3212.
34. *Public light bus*, http://en.wikipedia.org/wiki/Public_light_bus.
35. *Taxis collectifs Hanovre*, http://www.teiltaxi.de/.
36. *Linz*, http://www.inrets.fr/ur/ltn/publications/publis-kuhn/etude-cas/villes-europ/pdf/62-Linz.pdf.
37. *Transportation New York City : Share a cab with a stranger to save emissions*, http://www.inhabitat.com/2010/02/24/new-york-city-share-a-cab-with-a-stranger-to-save-emissions/.
38. J.N. Tsitsiklis, *Special cases of travelling salesman and repairman problems with time windows*, Networks **22** (1992), 263–282.
39. José Manuel Viegas, Joao de Abreu e Silva, and Rafaela Arriaga, *Innovation in transport modes and services in urban areas and their potential to fight congestion*, http://www.mitportugal.org/index.php?option=com_docman&task=doc_download&gid=229&Itemid=1, 2010.
40. *Taxi service : Women's night taxi*, http://www.gvh.de/sicherheit_taxi.html?&L=1.
41. Jin Xu and Zhe Huang, *An intelligent model for urban demand-responsive transport system control*, Journal of Software **4** (2009), no. 7, 766–776.
42. B. Zeddini, A. Yassine, M. Temani, and K. Ghedira, *Collective intelligence for demand-responsive transportation systems : a self organization model*, Proceedings of the 8th international conference on new technologies in distributed systems (Lyon, France), 2008.
43. Tarek Ziadé, *Programmation Python : Conception et optimisation*, Eyrolles, Paris, 2009, 2e ed.

# Résumé

Le développement économique d'une région urbaine est étroitement lié à son accessibilité. Le rôle des taxis comme moyen de transport est reconnu mondialement comme le mode offrant la meilleure qualité de service à ses usagers dans des conditions normales ou d'urgence. Malheureusement c'est un moyen très coûteux qui ne peut pas être utilisé quotidiennement par toute la population. Pour abaisser les coûts, il faudrait faire partager le service par plusieurs utilisateurs tout en préservant ses qualités essentielles (trajet presque direct, service porte à porte) en accroissant en même temps la productivité de ces véhicules devenus "collectifs". Cette idée n'est pas nouvelle : P.H. Fargier et G. Cohen l'avaient déjà étudiée en 1971, même si elle peut encore aujourd'hui être considérée comme révolutionnaire et un peu prématurée pour un marché strictement réglementé. Avec une révision de la réglementation, cette extension du service des taxis, si on lui donnait l'opportunité de se mettre en place, pourrait permettre aux taxis de prendre leur part du transport public en s'adressant à la majorité de la population et pas seulement à une minorité de privilégiés pouvant assumer le prix d'un transfert individuel.

Le **chapitre I** de ce mémoire présente une revue des "transports à la demande" dans le monde, des modèles développés à ce sujet, et des algorithmes en rapport avec ces problèmes. Il développe ensuite le projet de "taxis collectifs" étudié dans ce travail et se termine par un aperçu de l'ensemble du mémoire. Le **chapitre II** évoque les différents problèmes et questions qu'il faut aborder pour l'étude d'un système de taxis collectifs en ville, argumente sur la nécessité de construire un simulateur, décrit globalement sa structure avec une partie "mécanique" et une partie "algorithmique", les choix informatiques, les différents modes d'exploitation de ce simulateur, les données d'entrée requises et les sorties brutes. Le **chapitre III** développe la gestion décentralisée correspondant à un système qui fonctionne sans réservation préalable auprès d'un dispatching central : les clients rencontrent les taxis directement au bord du trottoir. Ce chapitre décrit en détail le modèle de "simulation par événements" adapté à ce mode de gestion, puis les problèmes de décision qui concernent la gestion d'un tel système et les algorithmes utilisés pour résoudre ces problèmes. Le **chapitre IV** rend compte d'une longue campagne de simulations numériques sur un cas d'étude fictif mais réaliste. Il illustre d'abord comment une simulation peut être analysée en détail et il montre les nombreuses sorties qui peuvent être produites à partir des sorties brutes du simulateur pour évaluer la qualité de service offerte et les coûts de fonctionnement associés. Il se tourne ensuite vers l'exploitation de séries de simulations où on fait varier certains paramètres relatifs à la gestion temps réel et au dimensionnement du système afin de réaliser les meilleurs compromis possibles entre plusieurs critères contradictoires. À l'aide de cette démarche, il montre aussi l'influence sur les résultats de données exogènes comme la demande (intensité et géométrie). Le **chapitre V** traite de la gestion à partir d'un dispatching central, les clients appelant ce dispatching pour réserver un trajet. La gestion mixte est la superposition dans un même système des deux modes de gestion, centralisée et décentralisée. Le simulateur existant est réputé capable, pour sa partie "mécanique" qui gère et enchaîne les événements, de s'accommoder de tous les modes de gestion. Par contre, nous avons manqué de temps pour travailler suffisamment sur la partie algorithmique (la gestion temps réel du cas centralisé est bien plus difficile que le cas décentralisé) et il n'y a donc pas non plus d'expériences numériques relatives à cette situation. Le chapitre se contente donc de décrire le modèle par événements pour les gestions centralisée et mixte et évoque la question algorithmique jusqu'à un certain point. Le **chapitre VI** présente quelques conclusions de ce travail.