



Titre: Algorithme de jumelage multimodal pour le covoiturage
Title:

Auteur: Mathieu Gagnon
Author:

Date: 2016

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Gagnon, M. (2016). Algorithme de jumelage multimodal pour le covoiturage
Citation: [Master's thesis, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/2251/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/2251/>
PolyPublie URL:

**Directeurs de
recherche:** Louis-Martin Rousseau
Advisors:

Programme: Maîtrise recherche en génie industriel
Program:

UNIVERSITÉ DE MONTRÉAL

ALGORITHME DE JUMELAGE MULTIMODAL POUR LE COVOITURAGE

MATHIEU GAGNON

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

AOÛT 2016

© Mathieu Gagnon, 2016.

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ALGORITHME DE JUMELAGE MULTIMODAL POUR LE COVOITURAGE

présenté par : GAGNON Mathieu

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. TRÉPANIÉ Martin, Ph. D., président

M. ROUSSEAU Louis-Martin, Ph. D., membre et directeur de recherche

Mme MORENCY Catherine, Ph. D., membre

REMERCIEMENTS

J'adresse mes remerciements à mon directeur de recherche M. Louis-Martin Rousseau pour m'avoir aidé à mener à bien le projet présenté dans ce mémoire.

Je remercie également M. Thibaut Jaurou et ses directeur et co-directeur de recherche, collaborateurs du projet, et l'entreprise Netlift, en particulier MM. Marc-Antoine Ducas et Guillaume Boudreau, partenaires du projet.

Enfin je remercie ma famille et mes amis pour m'avoir soutenu dans la réalisation de ce projet et dans l'écriture de ce mémoire.

RÉSUMÉ

Le covoiturage multimodal urbain est une solution économique pour réduire les émissions de gaz à effet de serre dans les villes. Le but du projet présenté dans ce mémoire est de modéliser et d'implémenter un algorithme de jumelage multimodal permettant de mettre en relation conducteurs et passagers pour effectuer des trajets quotidiens en zone urbaine. Cet algorithme a pour vocation d'être rapide et d'offrir des jumelages de qualité, en termes de détour acceptable et de respect des horaires. Il a également pour ambition de coupler le covoiturage avec les transports en commun.

Ce projet est en partenariat avec Netlift, startup Montréalaise, ainsi qu'avec un autre étudiant en maîtrise recherche au département de Génie Civil de l'Ecole Polytechnique de Montréal, qui travaille principalement sur les données utilisées.

Les objectifs de ce mémoire sont multiples. Le premier consiste à construire une structure de données permettant de modéliser la ville de Montréal et de calculer des temps de parcours. Ceci permettrait de comparer les différents trajets des utilisateurs. Aussi, cette structure de données doit permettre le calcul d'itinéraires multimodaux, auto et transports en commun combinés. Le second objectif est de modéliser et d'implémenter en JAVA un algorithme de jumelage passagers/conducteurs pour le covoiturage dit « classique » (auto uniquement) et pour le covoiturage multimodal.

Une revue de littérature a permis de diriger les travaux à mener. Ce travail est présenté dans le premier chapitre. Après une brève synthèse des concepts relatifs au covoiturage, une classification des systèmes et algorithmes existants permet d'amener différentes conclusions quant à la structure de données à implémenter, sur laquelle s'appuie l'algorithme envisagé. Elle doit permettre d'accéder rapidement aux données nécessaires à l'obtention de jumelages pour un passager donné. La structure du reste du mémoire est influencée par la chronologie du projet : la définition du besoin et les objectifs à atteindre ont été définis au fur et à mesure avec Netlift et les différents collaborateurs. Le second chapitre du corps du mémoire concerne les premières avancées menées en parallèle de la définition du besoin, tandis que le troisième chapitre décrit l'algorithme et la structure de données retenus pour satisfaire les objectifs fixés. Le quatrième et dernier chapitre présente les conclusions et les perspectives de recherche.

Dans le second chapitre, on essaye d'établir des indicateurs de potentiel de covoiturage au moyen d'un score et de différentes régressions linéaires. Ce sont ces recherches préalables qui ont conduit à l'élaboration d'une structure de données plus complexe, présenté dans le troisième chapitre, qui fait appel aux concepts de la théorie des graphes. L'algorithme développé dans cette partie fait notamment appel à des calculs de plus courts chemins. Il permet de trier une liste de conducteurs pour un passager donné en fonction de leur potentiel de covoiturage – notion qui sera expliquée en détail. Son évaluation est réalisée à l'aide de différentes métriques relatives aux données fournies par Netlift (jumelages trouvés par leur algorithme) et aux données de l'enquête Origine-Destination de Montréal pour l'année 2008.

Les résultats sont satisfaisants pour le covoiturage classique (sans transports en commun) puisque l'algorithme implémenté réussit à fournir rapidement des covoitureurs de bonne qualité pour une grande partie des utilisateurs. Parmi les passagers des données de l'enquête Origine-Destination, plus d'un passager sur deux possède un conducteur qui peut covoiturer avec lui pour des détours de 10min maximum. Le potentiel du covoiturage multimodal pour la période de pointe du matin est évalué grâce à une étude des trajets de l'enquête OD de 2008. Les jumelages obtenus sont moins bons que pour le covoiturage classique, mais la méthode employée présente une marge d'amélioration et une perspective de recherche future.

Ce projet permet à Netlift de gagner en pertinence et en rapidité par rapport aux jumelages proposés dans leur application actuelle.

ABSTRACT

Urban multimodal ridesharing is an economical way to reduce greenhouse gases emissions in cities. The goal of the project presented in this thesis is to modelise and implement a multimodal matching algorithm able to match drivers and passengers for everyday short ridesharing. This algorithm aims to be fast and to offer precise matches regarding acceptable detour and schedule respect. It also tries to mix ridesharing with public transportation.

This project is led in partnership with Netlift, a Montreal startup, an another master's student linked to the Civil Engineering department of Polytechnique Montreal, working especially on data.

Multiple objectives are targeted in this thesis. The first one consists of making a data structure representing Montreal and enabling travelling time calculation. This could lead to compare user's paths. Multimodal paths need also to be calculated thanks to this data structure. The second objective is to modelise and implement in JAVA a matching algorithm between riders and drivers for « classic » ridesharing (only car) and multimodal ridesharing (car and public transportation).

In the first part of this thesis, a literature review has been conducted in order to guide the goals to achieve. After a short synthesis of ridesharing concepts, a classification of existing articles about ridesharing leads to conclusions related to the data structure to implement. A need of speed is necessary to propose matched drivers to a given passenger.

The structure of this thesis is affected by the chronology of the project : the requirements definition and goals to achieve have been precised all along with Netlift and the other partners. The second chapter of the thesis deals with the initial steps conducted at the same time than the requirements definition. The third chapter describes then the data structure and algorithm selected to achieve goals.

In the second chapter, ridesharing potential is represented by two differents indicators : a score and different linear regressions. These preliminary searches led to the development of a data structure more complex, presented in the third chapter. Graph theory is central in this chapter. The final algorithm particularly uses shortest path calculation. It sorts a list of drivers for a given rider according to their ridesharing potential. A dedicated section of the thesis details this notion. The

algorithm is evaluated thanks to different metrics related to Netlift data (found matches by Netlift algorithm) and the Origin-Destination Survey of Montreal conducted in 2008.

The results are satisfying for classic ridesharing (without public transportation) since the implemented algorithm succeeds to give good drivers for a big amount of passengers fastly. More than one out of two among riders from the OD Survey has a driver to share the ride with for detours less than 10 minutes. The multimodal ridesharing potential for the morning peak period is evaluated by a study of rides from the Montreal 2008 OD Survey. Obtained matches lack of quality compared to classic ridesharing, but the used method deserves improvements and a perspective of future research.

This project enables Netlift to gain in relevance and computation speed against the matches proposed by the current algorithm.

TABLE DES MATIÈRES

REMERCIEMENTS	III
RÉSUMÉ.....	IV
ABSTRACT	VI
TABLE DES MATIÈRES	VIII
LISTE DES TABLEAUX.....	XI
LISTE DES FIGURES.....	XIII
LISTE DES SIGLES ET ABRÉVIATIONS	XV
LISTE DES ANNEXES.....	XVI
CHAPITRE 1 INTRODUCTION.....	1
1.1 Mise en contexte.....	1
1.2 Description du projet.....	2
1.3 Revue de littérature	3
1.3.1 Synthèse des concepts clés	3
1.3.2 Classification.....	5
1.3.3 Efficacité d'un système de covoiturage intra-urbain.....	11
CHAPITRE 2 STRUCTURE DE DONNÉES ET ALGORITHME INITIAUX	14
2.1 Données disponibles.....	14
2.1.1 Base de données Netlift.....	14
2.1.2 Algorithme de Netlift	15
2.1.3 Données de l'enquête Origine-Destination de 2008	15
2.2 Structure de données	16
2.2.1 Découpage de Montréal en zones et patterns	16
2.2.2 Pattern associé à un trajet	19

2.2.3	Tri de la base de données	19
2.3	Algorithme initial	19
2.4	Tests de l'algorithme initial.....	21
2.4.1	Un score comme critère de tri	21
2.4.2	Une estimation du détour comme critère de tri	29
2.4.3	Synthèse des tests	35
2.5	Conclusion.....	35
CHAPITRE 3 STRUCTURE DE DONNÉES ET ALGORITHME FINAUX		36
3.1	Structure de données	36
3.1.1	Graphe multimodal.....	36
3.1.2	Représentation des trajets OD	41
3.1.3	Algorithme de recherche de chemins multimodaux.....	41
3.2	Algorithme de jumelage	48
3.2.1	Sélection des meilleurs conducteurs	49
3.2.2	Sélection des jumelages	56
3.2.3	Processus de jumelage dynamique	56
3.2.4	Avantages et principe de fonctionnement	58
3.3	Tests et résultats sur les données de l'enquête OD	58
3.3.1	Validation du graphe multimodal.....	58
3.3.2	Validation de l'algorithme de jumelage multimodal.....	64
CHAPITRE 4 RECOMMANDATIONS ET CONCLUSION		78
4.1	Recommandations	78
4.1.1	Calculs hors-ligne.....	78
4.1.2	Calibrage du tri ajustable.....	78

4.1.3 Pistes d'amélioration	78
4.2 Contributions	79
4.3 Conclusion.....	80
BIBLIOGRAPHIE	82
ANNEXES	85

LISTE DES TABLEAUX

Tableau 1.1 : Classification d'articles scientifiques relatifs au covoiturage intra-urbain	7
Tableau 2.1 : Classement de trajets actifs - Exemple.....	22
Tableau 2.2 : Score simple - Exemple.....	22
Tableau 2.3 : Score complexe - Exemple.....	23
Tableau 2.4 : Couverture des jumelages Netlift - Tous.....	24
Tableau 2.5 : Couverture des jumelages Netlift - Au moins un.....	25
Tableau 2.6 : Classement via le score	27
Tableau 2.7 : Classement via le détour	27
Tableau 2.8 : Méthodologie d'évaluation- Exemple.....	28
Tableau 2.9 : Évaluation du score par le détour - Résultat	28
Tableau 2.10 : Classement via le détour estimé	30
Tableau 2.11 : Classement via le détour	30
Tableau 2.12 : Statistiques descriptives pour le premier échantillon.....	31
Tableau 2.13 : Modèle de régression linéaire avec x_1, x_2, x_3, x_4, x_5	32
Tableau 2.14 : Modèle de régression linéaire avec x_3, x_4, x_5	32
Tableau 2.15 : Modèle de régression linéaire avec x_1, x_2, x_5	33
Tableau 2.16 : Statistiques descriptives pour le deuxième échantillon.....	33
Tableau 2.17 : Modèle de régression linéaire avec x_1, x_2, x_5	34
Tableau 2.18 : Modèle de régression linéaire avec x_1, x_2, x_5	34
Tableau 3.1 : Temps de parcours nœuds artériels - Statistiques descriptives	61
Tableau 3.2 : Temps de parcours nœuds centroïdes - Statistiques descriptives.....	61
Tableau 3.3 : Détours - Critères 1 et 2	63
Tableau 3.4 : Détours - Critère 3.....	63

Tableau 3.5 : Algorithme de recherche de chemin Routier-TC - Résultats	65
Tableau 3.6 : Algorithme de recherche de chemin TC-Routier - Résultats	65
Tableau 3.7 : Tri ajustable pour le covoiturage classique – Tests.....	68
Tableau 3.8 : Tri ajustable pour le covoiturage Routier-TC - Tests	69
Tableau 3.9 : Tri ajustable pour le covoiturage TC-Routier - Tests	70

LISTE DES FIGURES

Figure 1.1 : Covoiturage multimodal	2
Figure 2.1 : Exemple d'une entrée de l'enquête OD 2008	15
Figure 2.2 : Zones de l'enquête OD 2008.....	16
Figure 2.3 : Différents points dans les zones	17
Figure 2.4 : Rectangle "circonsrit" à une zone	17
Figure 2.5 : Première séquence de zones	18
Figure 2.6 : Seconde séquence de zones	18
Figure 2.7 : Algorithme initial.....	20
Figure 3.1 : Quadrillage de la ville de Montréal	37
Figure 3.2 : Graphe routier.....	38
Figure 3.3 : Graphe routier et graphe TC	39
Figure 3.4 : Graphe GM	40
Figure 3.5 : Algorithme de recherche de chemins multimodaux	42
Figure 3.6 : Légende du graphe GM'	43
Figure 3.7 : GM' dans son état initial	44
Figure 3.8 : TC-Routier – GM' modifié.....	44
Figure 3.9 : TC-Routier – Plus court chemin p	45
Figure 3.10 : TC-Routier – GM' modifié après ajout de p dans L	45
Figure 3.11 : TC-Routier – Chemin renvoyé p routier uniquement	46
Figure 3.12 : Routier-TC – GM' modifié	46
Figure 3.13 : Routier-TC – Plus court chemin p	47
Figure 3.14 : Routier-TC – Plus court chemin p	47
Figure 3.15 : Routier-TC – Chemin renvoyé p routier uniquement	48

Figure 3.16 : Algorithme simplifié.....	49
Figure 3.17 : Algorithme – Sélection des meilleurs conducteurs	50
Figure 3.18 : Fenêtres de temps de départ - Exemple	51
Figure 3.19 : Fenêtres de temps d’arrivée - Exemple	52
Figure 3.20: Direction des trajets – Exemple.....	52
Figure 3.21 : Zones spatio-temporelles – Exemple.....	53
Figure 3.22 : Exception pour la direction des trajets - Exemple	54
Figure 3.23 : Processus de jumelage dynamique	57
Figure 3.24 : Temps de parcours des trajets de l’enquête OD – Nœuds artériels	59
Figure 3.25 : Temps de parcours des trajets de l'enquête OD - Nœuds centroïdes.....	60
Figure 3.26 : Temps de parcours des trajets de l'enquête OD - Nœuds TC	60
Figure 3.27 : Exemple de trajet Routier-TC 1.....	66
Figure 3.28 : Exemple de trajet Routier-TC 2.....	66
Figure 3.29 : Exemple de trajet TC-Routier 1.....	67
Figure 3.30 : Exemple de trajet TC-Routier 2.....	67
Figure 3.31 : Covoiturage classique – Pourcentage de passagers dont le 1er jumelage satisfait les horaires	71
Figure 3.32 : Covoiturage classique – Pourcentage de passagers dont au moins un des trois premiers jumelages satisfait les horaires	72
Figure 3.33 : Exemple 1 de jumelage classique	73
Figure 3.34 : Exemple 2 de jumelage classique	73
Figure 3.35 : Covoiturage multimodal Routier-TC – Pourcentage de passager dont au moins un des trois 1ers jumelages satisfait les horaires	74
Figure 3.36 : Exemple 1 de jumelage multimodal Routier-TC.....	76
Figure 3.37 : Exemple 2 de jumelage multimodal Routier-TC.....	77

LISTE DES SIGLES ET ABRÉVIATIONS

AMT	Agence Métropolitaine de Transport
BD	Base de données
CMM	Communauté Métropolitaine de Montréal
GTFS	General Transit Feed Specification
OSM	OpenStreetMap
OTP	OpenTripPlanner
RTL	Réseau de Transport de Longueuil
STL	Société de Transport de Laval
STM	Société de Transport de Montréal
TC	Transports en commun
Trajet OD	Trajet dont l'origine est O et la destination D

LISTE DES ANNEXES

ANNEXE A – IMPLÉMENTATION.....	85
ANNEXE B – EXEMPLE DE RÉSULTAT DE RÉGRESSION LINÉAIRE.....	86

CHAPITRE 1 INTRODUCTION

1.1 Mise en contexte

La recherche de solutions visant à lutter contre le réchauffement climatique est un enjeu majeur du 21^{ème} siècle. Pour ce faire, un cadre réglementaire contraignant pour les états a été façonné lors de la conférence des Nations Unies à Paris en décembre 2015. L'accord signé par 55 pays prévoit de cantonner la hausse des températures « bien en dessous de 2°C par rapport aux niveaux préindustriels », et éventuellement de viser à « poursuivre les efforts pour [la limiter] à 1,5°C ».

Cette montée des températures est sans doute corrélée à la société de consommation répandue dans les pays développés. La remise en cause de ce modèle dépensier permettrait aux sociétés occidentales de servir d'exemple aux pays en voie de développement.

C'est dans cette optique et pour satisfaire les objectifs de l'accord de Paris qu'il est nécessaire de réduire notre consommation d'énergie.

Le secteur des transports regroupe à lui seul près de 28% de la consommation d'énergie finale dans le monde. Il est donc urgent de diminuer les émissions de gaz à effet de serre qui lui sont dues. La voie de l'innovation technologique permet d'atteindre ces objectifs : voitures propres, aménagement urbain, villes intelligentes. Le covoiturage est plutôt du côté de l'innovation sociale qu'est le partage des ressources. La technologie et les moyens de communication modernes permettent cependant de contribuer à l'expansion de ce phénomène.

Pour les longs trajets, le covoiturage interurbain est une solution qui fonctionne déjà efficacement, soutenu par des plateformes de communication et des modèles d'échange et de sécurité de haut niveau : Blablacar en Europe, Amigo Express au Québec. Le covoiturage intra-urbain tarde quant à lui à se développer, bien que de multiples applications y soient consacrées : Karos et Wayz'Up en France, Flinc en Allemagne ou encore Netlift au Québec en sont quelques-unes.

Ceci peut éventuellement s'expliquer par deux facteurs : le peu d'utilisateurs présents sur ces plateformes, et la faible complexité des algorithmes de jumelage en place.

Selon Furuhata et al., il existe trois clés nécessaires à l'expansion d'un système de covoiturage (Furuhata et al., 2013) :

- L'attractivité du service, en termes d'ergonomie et de sécurité, et en termes d'incitatifs à la participation.
- La construction d'un système basé sur la confiance et la participation d'utilisateurs qui ne se connaissent pas initialement.
- La qualité des jumelages proposés.

La première est relative au modèle économique déployé et à la technologie utilisée pour construire l'interface utilisateur. La seconde concerne plus directement chaque usager, qui participe au bon fonctionnement du service. Pour la dernière, la complexité des algorithmes et modèles employés est cruciale à l'obtention de jumelages intéressants.

Un algorithme efficace de jumelage multimodal (cf. 1.2 pour la définition de covoiturage multimodal) entre passagers et conducteurs couplé aux technologies actuelles et à un modèle socio-économique pertinent permettrait de développer le covoiturage urbain. C'est une solution économique pour ses utilisateurs, qui réduisent ainsi leurs coûts de transport quotidiens monétaires et environnementaux, tout en participant à un vaste réseau d'échanges intercommunautaires.

1.2 Description du projet

Le covoiturage multimodal est illustré sur la figure ci-dessous.

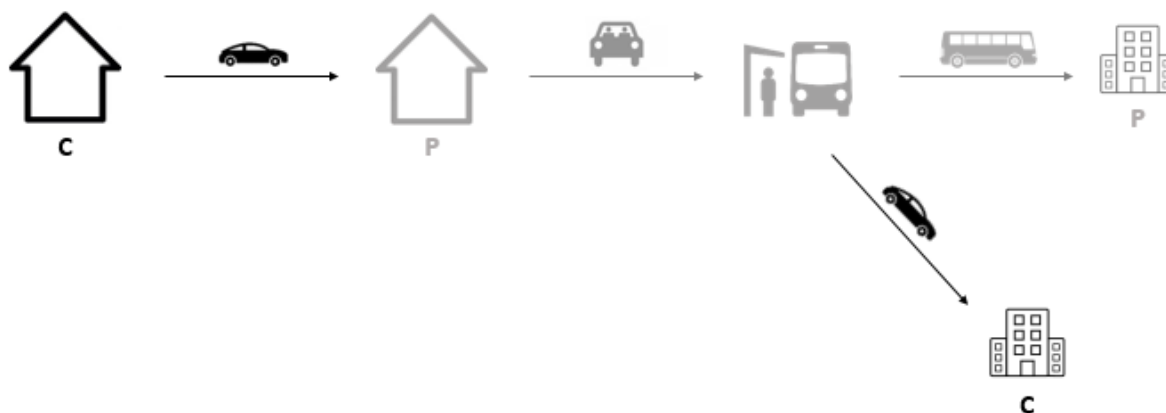


Figure 1.1 : Covoiturage multimodal

Dans la figure 1.1, un conducteur C effectue son trajet domicile-travail quotidiennement en auto, représenté en noir, et un passager P désire se rendre à son travail depuis son domicile, en gris. Le covoiturage multimodal consiste pour le passager à partager une partie de son trajet avec le

conducteur et à utiliser les transports en commun sur la partie restante de son trajet. Ici, le conducteur récupère le passager à son domicile et le dépose à une station de transports en commun. Il continue sa route en auto pour arriver à son travail, tandis que le passager atteint sa destination en transports en commun, tels que le train, le métro ou le bus.

Le projet relatif à ce mémoire consiste à modéliser et implémenter un algorithme de jumelage multimodal pour le covoiturage, et à l'évaluer au regard de métriques bien choisies. Simplement, pour un passager donné, le(s) meilleur(s) conducteur(s) sont recherchés parmi une liste de conducteurs donnée. Le(s) jumelage(s) obtenu(s) répond(ent) à des critères d'optimisation en termes de préférences utilisateurs, de détour effectué et de temps gagné.

Ce projet est en partenariat avec l'entreprise Netlift, qui dispose d'une plateforme internet et mobile proposant un service de covoiturage urbain. Les données à disposition sont celles de l'entreprise Netlift, les demandes de trajets effectuées sur leur application, ainsi que les jumelages obtenus. Les données de l'enquête Origine-Destination (OD) 2008 de Montréal sont également utilisées.

1.3 Revue de littérature

1.3.1 Synthèse des concepts clés

Le thème du covoiturage a déjà été étudié dans divers articles, présentant des résumés assez complets des différents concepts associés. Le premier article – (Furuhata et al., 2013) – classe à plusieurs niveaux les systèmes de covoiturage : marché cible, type de covoiturage, critères d'appariement des conducteurs aux passagers, etc. Le second – (Agatz, Erera, Savelsbergh, & Wang, 2012) – s'attarde en particulier sur le covoiturage dynamique. Voici une brève synthèse des points intéressants dans ces deux articles.

Furuhata et al. distinguent deux types de covoiturage (Furuhata et al., 2013) :

- « *Unorganized ridesharing* », traduit par covoiturage non-planifié, qui concerne le covoiturage entre particuliers (amis, famille, collègues), ne faisant pas appel à des méthodes de communication et d'organisation poussées.
- « *Organized ridesharing* », traduit par covoiturage planifié à l'avance, dont le but est de répondre à l'offre des conducteurs et à la demande des passagers. Ce service peut être proposé par des « *service operators* », traduit par opérateurs de service, possédant leur

propre flotte de véhicules, ou par des « *matching agencies* », traduit par agences de jumelages, qui permettent de jumeler des passagers et des conducteurs, en fonction de leurs préférences.

Un des problèmes rencontrés par ce type d'agence, dont Netlift fait partie, est d'atteindre une masse critique d'utilisateurs, ce qui permettrait au service d'être efficace. En pratique, c'est assez complexe puisque pour attirer des conducteurs, l'agence doit proposer un nombre important de demandes de trajet par des passagers, mais pour attirer ces passagers, il faut aussi un nombre suffisamment important d'offres de conducteurs. L'une des composantes de la recherche sur le covoiturage consiste donc à étudier et proposer des schémas incitatifs à la participation pour les conducteurs et les passagers (Agatz et al., 2012).

Selon Furuhashi et al., quatre patrons de covoiturage sont détaillés (Furuhashi et al., 2013). Le premier concerne les covoitureurs qui partagent la même origine et la même destination (« *identical ridesharing* »). Dans le second, l'origine et la destination sont incluses dans le parcours initial du conducteur (« *inclusive ridesharing* »). Pour le troisième, le covoiturage ne concerne qu'une partie du trajet du passager : son origine ou sa destination n'est pas sur le parcours initial du conducteur. Le dernier ressemble au précédent, mais ici le conducteur effectue un détour pour aller chercher le passager et pour le déposer. Les points d'embarquement et de dépôt du passager ne sont donc pas sur le parcours initial du conducteur et ne coïncident pas obligatoirement avec l'origine et la destination du passager. Ce dernier patron est le plus complexe à prendre en compte pour les agences de jumelages, parce qu'il inclut des détours et d'autres moyens de transports, le passager n'étant embarqué à son origine et déposé à destination que dans certains cas. C'est celui qui est le plus important pour ce projet. Il est alors aisé de deviner la complexité de planifier à l'avance de tels trajets pour plusieurs utilisateurs.

La classification établie par Furuhashi et al. permet de classer différentes agences de jumelages selon deux critères : le critère de recherche principal pour former des jumelages conducteurs-passagers et le marché cible (Furuhashi et al., 2013). Parmi les types de covoiturage retenus, deux sont particulièrement intéressants :

- « *Dynamic real-time ridesharing* », le covoiturage dynamique en temps réel qui permet de jumeler des utilisateurs en temps réel ou quasiment.

- « *Carpooling* », le covoiturage où les passagers partagent la plupart du temps des origines et destinations similaires. C'est le cas pour des passagers habitant dans la même zone urbaine et se rendant tous les jours au travail dans la même zone d'affaires.

Netlift et l'algorithme visé par le projet se situent entre ces deux types de covoiturage. Les jumelages sont planifiés à l'avance pour des trajets quotidiens domicile-travail et pour des origines et destinations similaires, mais l'horizon de planification est assez étroit et pourrait s'apparenter à du temps quasi-réel. Il est à remarquer que les agences de jumelages récentes ont tendance, comme Netlift, à couvrir plusieurs types de covoiturage (Furuhata et al., 2013).

1.3.2 Classification

Fuhurata et al. et Agatz et al. cherchent à classer les différents systèmes et techniques employés pour le covoiturage dynamique en temps réel (Agatz et al., 2012; Furuhata et al., 2013).

L'une des clés d'un service de covoiturage réside dans la qualité des solutions proposées. Cette qualité dépend largement des méthodes et divers algorithmes employés pour réaliser des jumelages. Cette section s'intéresse à différents articles scientifiques utilisant divers outils pour établir des systèmes de covoiturage. Une centaine d'articles ont été trouvés sur plusieurs bases de données. Seuls les plus pertinents et présentant des modèles mathématiques avec un minimum de complexité ont été retenus (30 articles).

La classification réalisée a pour but d'identifier les différentes techniques utilisées, afin de permettre, si possible, à de futurs chercheurs/étudiants, de repérer les thématiques relatives à leurs projets. Ce travail aidera également à illustrer la multiplicité des modèles existants.

Plusieurs critères de classification sont retenus :

- Agatz et al. insistent sur la nécessité d'élaborer des systèmes décentralisés (Agatz et al., 2012). Ceux-ci utilisent des méthodes de décomposition – par exemple, le tri des offres et demandes en fonction des préférences ou de la zone géographique – permettant d'améliorer leur faisabilité, contrairement aux systèmes centralisés, qui cherchent une solution globale au problème de covoiturage, mais qui sont difficiles à mettre en pratique pour un nombre important d'utilisateurs. **C'est donc le premier critère retenu, D pour les articles détaillant des modèles s'apparentant plus aux systèmes décentralisés, C pour les systèmes centralisés.**

- **Le second critère** concerne les **structures de données** particulières employées. La majeure partie des articles fait appel aux graphes, que l'on symbolisera par la lettre G.
- **Le troisième critère** fait référence à la **modélisation** au sens large. Dans cette catégorie sont abordés les modèles mathématiques employés, et les outils au sens large permettant de modéliser les offres et demandes des utilisateurs du service de covoiturage.
- **Le quatrième critère** est en lien étroit avec le précédent. Il énonce le(s) technique(s) de **résolution** employé(es) pour résoudre les modèles développés.
- **Le cinquième critère** est lié aux variables de **décision** présentes dans les modèles. Au sens large, il réfère à la décision prise dans chaque article. Cela peut être les jumelages entre passagers et conducteurs, les « pools », i.e. les groupes d'utilisateurs formés (J), le plan de route, les chemins empruntés et les flux dans les graphes (P), les horaires des conducteurs et passagers (H), les points de transferts pour les passagers qui utilisent d'autres moyens de transports (T), le rôle des utilisateurs – conducteur, passager, transport en communs (R, quand l'article insiste sur ce point) ainsi que les détours acceptables (D).
- **Le sixième et dernier critère** présente la **fréquence de décision** du modèle présent dans l'article, i.e. le temps au bout duquel l'algorithme est relancé. Cela peut être effectué en temps réel, au bout d'une journée, à chaque nouvelle demande de trajet, ou au bout d'un certain temps Δt .

Les trois pages suivantes présentent le tableau obtenu.

Tableau 1.1 : Classification d'articles scientifiques relatifs au covoiturage intra-urbain

Référence	Type	Structure de données	Modélisation	Résolution	Décision	Fréquence de décision
(Ta Anh, Le Thi Hoai, Pham Dinh, & Khadraoui, 2012)	C	G	Programmation mixte en nombre entiers	Inspiré d'un algorithme de labellisation	P	N/A
(Guo, Goncalves, & Hsu, 2013)	C	G	Programmation mixte en nombre entiers	Heuristique, algorithme de colonies de fourmis, recherche locale	J, P, H, T	N/A
(Huang, S.-C., Jiau, & Lin, 2015)	C	G	Programmation mixte en nombre entiers	Algorithme génétique	J, P	Temps réel
(Armant & Brown, 2014)	C	N/A	Programmation mixte en nombre entiers	Cplex branch and cut	J, H	Un jour
(Naoum-Sawaya et al., 2015)	C	G	Programmation stochastique mixte en nombre entiers	Heuristique	J, P, H	N/A
(Baldacci, Maniezzo, & Mingozzi, 2004)	C	G	Programmation mixte en nombre entier et formulation en partition d'ensembles	Méthode exacte, heuristique, génération de colonnes	J, P	N/A
(Xu, Pang, Ordóñez, & Dessouky, 2015)	C	G	Modèle mixte de complémentarité linéaire	Méthode exacte, optimisation non linéaire via le solveur KNITRO	P, R	N/A
(Yan & Chen, 2011)	C	G espace-temps	Modèle de flot multi-commodités	Heuristique et relaxation lagrangienne	J, P	N/A
(Meng, Wang, Shao, & Ding, 2013)	C	G	Multi-hop ride sharing qui pourrait se traduire par covoiturage à sauts multiples	Algorithme de colonie de fourmis	P	N/A
(Coltin & Veloso, 2014)	C	G	Covoiturage à sauts multiples	Approche gloutonne, et algorithme d'enchères et de graphes	J, T	N/A
(Herbawi & Weber, 2011)	C	G	Plus courts chemins multi-objectifs	Algorithme génétique	P	N/A

Tableau 1.1 : Classification d'articles scientifiques relatifs au covoiturage intra-urbain (suite)

Référence	Type	Structure de données	Modélisation	Résolution	Décision	Fréquence de décision
(Hrnčič, Rovatsos, & Jakob, 2015)	C	G	Modèle multi-agents	Algorithme	P	Un jour
(Dakroub, Boukhatat, Lahoud, Awad, & Artail, 2013)	C	G biparti	Contraintes sur les arcs du graphe	Solution initiale basée sur les préférences des utilisateurs puis algorithme génétique	J	Un jour
(Vargas, Sefair, Walteros, Medaglia, & Rivera, 2008)	C	G biparti	Contraintes sur les arcs du graphe	Méthode exacte	J	Un jour
(Cao, Alarabi, Mokbel, & Basalamah, 2015)	C	N/A	Contraintes simples sur les préférences/distances	Algorithme de tri puis algorithme de jumelage	J	N/A
(Bruglieri, Ciccarelli, Colomi, & Luè, 2011)	C	Matrice de jumelages	Contraintes simples sur les préférences	Algorithme de Monte-Carlo	J, H	Un jour
(He, Hwang, & Li, 2014)	C	G, données GPS	Modèle assisté par GPS	Algorithme de forage de données	J	Temps réel
(Zhu, Feng, & Liu, 2013)	C	N/A	Etude des déplacements de voitures en provenance des mêmes parkings	Algorithme de jumelage	J	N/A
(Sghaier, Zgaya, Hammadi, & Tahon, 2011)	D	Partition du réseau, G	Modèle multi-agents	Algorithmes généraux	J	Temps réel
(Pelzer et al., 2015) ^{el}	D	Partition du réseau	Trajet = séquence de partitions du réseau	Algorithme de tri sur les séquences puis algorithme de jumelage	J	Demande de trajet
(Nourinejad & Roorda, 2016)	D	Partition du réseau, G	Modèle multi-agents	Algorithme de tri sur les partitions puis algorithme d'enchères pour les jumelages	J	Δt
(Geisberger, Luxen, Sanders, Neubauer, & Volker, 2010)	D	G, "buckets" stockés en avance	Plus courts chemins et contraintes sur les détours acceptables	Algorithme de jumelage	J, D	N/A

Tableau 1.1 : Classification d'articles scientifiques relatifs au covoiturage intra-urbain (suite et fin)

Référence	Type	Structure de données	Modélisation	Résolution	Décision	Fréquence de décision
(Aissat & Oulamara, 2015)	D	G, "buckets" stockés en avance	Plus courts chemins	Algorithme de plus courts chemins puis algorithme de jumelage	J, P, T	Demande de trajet
(Huang, Y., Bastani, Jin, & Wang, 2014)	D	G	Plus courts chemins	Branch and bound	J	Temps réel
(Aissat & Varone, 2015)	D	G	Plus courts chemins, substitutions de trajets TC au covoiturage	Algorithme de tri puis algorithme de jumelage	J	Temps réel
(Yousaf et al., 2012)	D	G	Plus courts chemins, contraintes simples sur les préférences des utilisateurs	Algorithme de jumelage	J, P	N/A
(Zhao, Zhang, Gerding, Sakurai, & Yokoo, 2014)	D	G	Modèle mathématique et recherche d'incitatifs efficaces	N/A	J, R	N/A
(Drews & Luxen, 2013)	D	G espace-temps	Covoiturage à sauts multiples	Algorithme de jumelage	J, D	N/A
(Galland et al., 2014)	D	N/A	Modèle multi-agents	Algorithmes généraux	J	N/A
(Armant & Brown, 2014)	N/A	N/A	Programmation par contraintes	Solveur de programmation par contraintes	J, R	N/A

Près des deux tiers des articles étudiés proposent des modèles centralisés, pendant que le tiers d'entre eux décrivent des modèles décentralisés.

En ce qui concerne les structures de données utilisées, la plupart des systèmes présents dans les articles font appel aux graphes, qui représentent le réseau – graphe et graphe espace-temps – ou les relations entre conducteurs et passagers – graphe biparti. Parmi les modèles décentralisés, il existe néanmoins des structures différentes que pour les modèles centralisés : des partitions de réseau permettent de décomposer le problème de covoiturage en plusieurs parties, correspondant dans la plupart du temps à différentes zones géographiques. Il est également à souligner que certains auteurs utilisent des structures de données calculées à l'avance et accessibles rapidement, au moyen de « buckets ».

Pour ce qui est de la modélisation et de la résolution, il existe une multitude d'approches différentes. Les modèles décentralisés s'appuient sur la programmation mathématique – linéaire, non-linéaire, stochastique, en nombre entiers, binaire, mixte, etc. – et sur les modélisations relatives aux graphes – plus courts chemins, flots, contraintes sur les arcs. La programmation mathématique et l'optimisation globale n'est pas retrouvée dans les modélisations décentralisées, mais des modélisations relatives aux graphes, ou qui découlent de la structure de données choisie (notamment pour les réseaux partitionnés). Il y a aussi trois modèles multi-agents, qui permettent de considérer chaque utilisateur comme un agent autonome, et d'optimiser le covoiturage pour chacun d'entre eux.

Au niveau de la résolution, pour les modèles centralisés, tous types d'algorithmes sont présents – génétiques, algorithme de fourmis, etc. – ainsi que des méthodes de résolution exactes, faisant appel à des solveurs connus. Les modèles décentralisés sont majoritairement résolus de manière heuristique à l'aide d'algorithmes plus ou moins complexes, qui visent souvent à trier les demandes dans un premier temps puis à trouver des jumelages adéquats.

Au sujet des critères relatifs aux décisions prises dans les modèles, une très large majorité des systèmes étudiés vise à définir les jumelages entre passagers et conducteurs. Les systèmes centralisés donnent également assez fréquemment le plan de route pour un ensemble d'utilisateurs donnés (arcs empruntés et flux dans les graphes), et parfois les horaires de chacun, les points de transferts et le rôle de chaque utilisateur. Il est à constater que certains systèmes décentralisés ne donnent pas précisément les jumelages dans tous les cas. Au contraire, les systèmes décentralisés

s'attachent à bien définir les jumelages (le type de décision J est présent pour la totalité d'entre eux). Ils définissent parfois d'autres variables de décision, et certains d'entre eux évoquent la notion de détour acceptable pour les conducteurs.

Lorsque celle-ci est applicable, la fréquence de décision pour les systèmes centralisés est la journée et dans de rares cas en temps réel. Ils proposent en effet des plans de covoiturage globaux, assez incompatibles avec des systèmes flexibles et faciles à implémenter. Une corrélation opposée est à soulever sur le tiers restant d'articles présentant des modèles décentralisés, où, lorsque c'est applicable, ceux-ci présentent une fréquence de décision en temps réel ou quasi-réel (demande de trajet, Δt). Ceci conforte la praticité des modèles décentralisés.

1.3.3 Efficacité d'un système de covoiturage intra-urbain

Deux tendances semblent émerger au sein des articles de la classification effectuée à la partie précédente, notamment au niveau des systèmes décentralisés.

La première tendance concerne la complexité du modèle de jumelage à obtenir pour avoir un système de covoiturage efficace en pratique. Il est en effet assez trivial de constater que les appariements de conducteurs et passagers partageant une même origine et une même destination n'arrivent quasiment jamais en pratique. Selon Armant et al., les solutions actuelles ne prennent pas assez en compte la faisabilité des jumelages proposés, mais s'attardent plus sur le caractère dynamique du système optimal à atteindre (Armant & Brown, 2014).

Comme expliqué dans une partie précédente, c'est le quatrième patron (Furuhata et al., 2013) qui concerne directement le projet puisqu'il regroupe plus de possibilités de covoiturage, et donc un marché plus large, une offre et une demande plus large. C'est celui où le passager et le conducteur n'ont pas forcément la même origine, ni la même destination. Il faut alors réfléchir intelligemment afin de permettre au passager d'atteindre sa destination.

La première idée consiste à amener le chauffeur sur le chemin du passager. Il faut alors contraindre le chauffeur à récupérer ou déposer son passager en effectuant un détour, si celui-ci n'est pas supérieur à un seuil défini par le chauffeur.

La seconde idée consiste à amener le passager sur le chemin du conducteur. Divers moyens existent : la marche, le vélo, l'auto, les transports en commun. C'est ici que la notion de covoiturage multimodal apparaît. Dans plusieurs articles, il est question de trouver le meilleur point de

rencontre et de dépose du passager, de manière à minimiser les détours des conducteurs (Drews & Luxen, 2013; Geisberger et al., 2010). Dans d'autres articles, le trajet d'un passager peut être composé de plusieurs tronçons en covoiturage avec différents conducteurs, alternant ainsi avec plusieurs conducteurs. Les réseaux espace-temps analogues à ceux utilisés dans les systèmes ferroviaires (Coltin & Veloso, 2014), et le covoiturage à saut multiples (Drews & Luxen, 2013) traitent de ce type de modélisation. Dans d'autre encore, il est évoqué la combinaison du covoiturage au transport en commun. C'est ce dont Agatz et al. discutent dans une partie consacrée, où ils expliquent que la combinaison de ces deux modes permettraient aux instances publiques d'obtenir une plus forte fréquentation des transports en commun (Agatz et al., 2012). Le covoiturage serait alors un moyen d'amener de nouveaux utilisateurs aux zones où le trafic est plus dense et où le transport en commun est plus rapide que l'auto. Il serait également le moyen de remplacer des lignes de transports en commun dont la fréquence de passage est faible. C'est une solution envisagée par Aissat et Varone qui substituent des tronçons en transport en commun à des tronçons en covoiturage (Aissat & Varone, 2015). Plus le nombre de modes empruntés par un passager et plus le nombre de véhicules empruntés au sein du même mode de transport sont importants, plus le trajet est sujet aux aléas, aux retards, et donc plus difficile à mettre en pratique.

La deuxième tendance concerne les structures de données à réaliser à l'avance, de manière à en avoir un accès rapide. Idéalement, un calculateur de chemin multimodal fonctionnant en temps réel et intégrant l'offre de covoiturage et du transport en commun permettrait de résoudre le problème de covoiturage multimodal. En pratique, un tel système poserait des problèmes d'ordre technique pour la réservation des voitures disponibles.

Un système dynamique en temps réel (« *dynamic real-time ridesharing* » (Furuhata et al., 2013)) a l'inconvénient d'être peu résilient, mais permet de toucher un maximum d'utilisateurs : celui qui se prépare la veille d'un déplacement comme celui qui souhaite covoiturer au dernier moment.

Un système non dynamique (« *carpooling* » (Furuhata et al., 2013)) reposant sur l'appariement passager conducteur planifié à l'avance est lui plus facile à implémenter en pratique mais concerne un public moins nombreux, donc une offre plus faible.

Comme expliqué précédemment, la différence entre les deux est floue, car la notion de « planification de trajet » concerne la fenêtre de temps de jumelage (« *matching time window* » (Agatz et al., 2012)).

Des capacités de calculs et de stockage de données importantes sont nécessaires pour réaliser un système de covoiturage dynamique. C'est pourquoi il faut essayer de contourner les algorithmes gloutons, qui considèrent toutes les possibilités une à une, ou la résolution d'un modèle mathématique au sens centralisé (optimisation une et une seule fois), comme préconisé par Agatz et al. (Agatz et al., 2012). Les algorithmes décentralisés semblent plus adaptés à un système robuste, peu cher en temps de calculs. C'est ce que plusieurs auteurs cherchent à faire, en stockant des chemins à l'avance (Aissat & Oulamara, 2015; Geisberger et al., 2010), ou en partitionnant le réseau routier en zones différentes de manière à effectuer un tri plus rapide parmi les demandes de covoiturage des utilisateurs (Pelzer et al., 2015).

Une solution hybride prenant en compte ces deux tendances, avec une structure de données préalablement déployée, et des calculs algorithmiques de jumelage suffisamment rapides et pertinents, pourrait permettre d'avoir la flexibilité d'une planification à l'avance, et la complexité d'un calculateur de chemins multimodal.

La démarche de la recherche qui a été menée dans ce mémoire a pour but d'obtenir ce type de solution hybride, tout en répondant aux besoins de l'entreprise partenaire Netlift. La définition de ces derniers au fur et à mesure de l'avancement de ce projet expliquent le découpage en chapitres de ce mémoire. Le second chapitre s'attarde notamment sur les réflexions et les avancées menées en début de projet notamment sur la structure de données à utiliser, mais qui n'ont pas été retenues par la suite. Le troisième chapitre concerne la structure de données et l'algorithme retenus qui répondent aux critères cités plus haut de solution hybride. Le quatrième et dernier chapitre s'attarde sur les recommandations permettant de calibrer l'algorithme et insiste sur le travail futur à réaliser.

CHAPITRE 2 STRUCTURE DE DONNÉES ET ALGORITHME INITIAUX

Ce chapitre concerne la première phase du projet qui a permis de mieux définir les contours de l'algorithme final à implémenter, présenté dans le chapitre 3. Les données de l'enquête Origine-Destination de 2008 sont les seules qui sont également utilisées dans le chapitre 3. Pour le reste, ce qui est dans chaque chapitre correspond à une structure de donnée différente testée pour un algorithme différent : la structure de données initiale pour l'algorithme initial et la structure de données finale pour l'algorithme final et retenu.

2.1 Données disponibles

Cette partie décrit les données utilisées pour le projet. Elles proviennent de deux sources différentes : l'entreprise partenaire Netlift et l'enquête Origine-Destination (enquête OD) 2008 réalisée par un consortium d'autorités de transport de la région Montréalaise.

2.1.1 Base de données Netlift

Le projet s'appuie sur la base de données historique des demandes de trajets effectuées sur l'application web de Netlift. Celle-ci couvre la période de janvier 2013 à juin 2015.

Chaque entrée de la base de données concerne un trajet, qui a différents attributs :

- Id, le numéro d'identification du trajet. *Ex. : Id = 1.*
- User_id, le numéro d'identification du titulaire du trajet. *Ex. : User_id = 12.*
- User_role, le rôle de titulaire du trajet – conducteur ou passager
- Origin_id, le numéro d'identification de l'origine du trajet
- Origin_name, le nom de l'origine du trajet. *Ex. : Origin_name = Longueuil.*
- Origin_lat, la latitude de l'origine du trajet. *Ex. : Origin_lat = 45.533483.*
- Origin_lon, la longitude de l'origine du trajet. *Ex. : Origin_lon = -73.506965.*
- Destination_id
- Destination_name. *Ex. : Destination_name = Polytechnique Montréal*
- Destination_lon
- When, l'horaire du trajet. *Ex. : When = 04/03/2015 08:00:00.*
- When_type, le type d'horaire – départ ou arrivée

- Recurrence, les jours de la semaine lors desquels le trajet est effectué. *Ex. : Mon, Tue, Wed*
- Matched_rider_routes, les ids des jumelages passagers trouvés par l'algorithme de Netlift.
Ex : Matched_rider_routes = 10, 324, 43209.
- Matched_driver_routes, les jumelages conducteurs trouvés par l'algorithme de Netlift

2.1.2 Algorithme de Netlift

L'algorithme actuel de Netlift se divise en plusieurs étapes : lorsqu'un utilisateur fait une demande de covoiturage (conducteur ou passager), la base de données (BD) de Netlift des demandes/offres de trajet est sondée pour récupérer celles qui sont dans une fenêtre de temps de +/-2h et qui sont des offres dans le cas d'un utilisateur passager, et des demandes ou des offres dans le cas d'un utilisateur conducteur. Des calculs complexes (liés aux portions de trajets en commun) sont ensuite effectués pour déterminer le potentiel de covoiturage de chacun de ces trajets (correspondant à une offre ou une demande), afin de sélectionner le meilleur jumelage possible. Les points de dépose et d'embarquement de passagers sont également sauvegardés pour présentation finale aux utilisateurs.

2.1.3 Données de l'enquête Origine-Destination de 2008

L'enquête Origine Destination a été effectuée en 2008 conjointement par l'Agence Métropolitaine de Transport (AMT), la Société de Transport de Montréal (STM), la Société de Transport de Laval (STL), le Réseau de Transport de Longueuil (RTL), et d'autres partenaires gouvernementaux et associatifs. Elle consiste à analyser les déplacements de milliers de montréalais au cours de plusieurs semaines. L'enquête met également à disposition une base de données de déplacements en auto pour un jour moyen de la semaine. Par souci de simplicité, celle-ci est convertie selon les mêmes attributs que la base de données Netlift, le travail consistant essentiellement à convertir les coordonnées des trajets d'une base de données à l'autre.

id	heure	orig_lon	orig_lat	dest_lon	dest_lat
1	830	-73,6752009056086	45,5370951262182	-73,5709506278200	45,4980553756262

Figure 2.1 : Exemple d'une entrée de l'enquête OD 2008

Les trajets utilisés pour les tests sont les trajets effectués en auto solo conducteur pour se rendre au travail le matin, puisque ce sont ces utilisateurs qui peuvent potentiellement covoiturer quotidiennement.

2.2 Structure de données

Comme expliqué dans la revue de littérature, il est nécessaire de disposer d'une structure de données permettant l'accès et le traitement des trajets des utilisateurs de manière rapide et efficace. La structure de données initialement imaginée prend en compte ces critères et essaie de représenter de manière simplifiée ces trajets.

2.2.1 Découpage de Montréal en zones et patterns

La ville de Montréal est divisée en différentes zones, tirées de l'enquête OD de 2008.

Dans cette enquête 108 secteurs municipaux sont distingués, qui correspondent donc aux 108 zones de notre base de données, représentées ci-dessous :



Figure 2.2 : Zones de l'enquête OD 2008

Un pattern représente une séquence de zones du découpage de la ville de Montréal. Il est identifié par une zone d'origine Z_O et une zone de destination Z_D , ainsi que les zones intermédiaires traversées successivement lors du trajet entre les deux.

Pour chaque paire de zones (Z_O, Z_D), il y a plusieurs patterns différents. En effet, certaines zones étant très grandes, de grosses différences parmi les zones traversées peuvent exister, si différents points de départ/arrivée sont considérés.

Pour ce faire, plusieurs points de départ/arrivée sont définis pour chaque zone différente, de manière à récupérer en mémoire le plus de patterns judicieux, une dizaine en pratique. Ceux-ci sont séparés d'une distance égale au quart de la diagonale du rectangle « circonscrit » à la forme qui représente la zone en question.



Figure 2.3 : Différents points dans les zones



Figure 2.4 : Rectangle "circonscrit" à une zone

Par exemple, pour la paire de zones (Z_1, Z_{18}) , l'un des patterns correspondants est représenté par :

- Une zone d'origine Z_1 et un point de départ dans celle-ci
- Une séquence de zones successives : $Z_1 - Z_9 - Z_{24} - Z_{12} - Z_3 - Z_{27} - Z_{13} - Z_{18}$
- Une zone de destination Z_{18} et un point d'arrivée dans celle-ci

Dans les deux figures suivantes, il y a respectivement deux points de départ différents dans la zone d'origine Z_1 , et donc deux séquences de zones différentes (correspondant à deux patterns différents). Les zones traversées sont en gris foncé, et le trajet est en noir.



Figure 2.5 : Première séquence de zones



Figure 2.6 : Seconde séquence de zones

Ainsi, la liste des patterns possibles peut être calculée au préalable sur le découpage en zones de Montréal. Chaque paire (Z_O, Z_D) est liée à un ou plusieurs patterns différents.

2.2.2 Pattern associé à un trajet

Le pattern associé à un trajet donné est celui dont les points de départ et d'arrivée sont les plus proches de l'origine et de la destination du trajet en question.

2.2.3 Tri de la base de données

Sur les 136692 entrées de la base de données Netlift, les trajets doublons sont supprimés (attributs Origin, Destination et When égaux). Ensuite, seuls sont conservés ceux qui sont susceptibles d'être dans le découpage de Montréal en zones : la latitude des trajets doit se situer entre 45° et $46,25^{\circ}$ et la longitude entre $-74,7^{\circ}$ et 73° . Après ce premier filtre, il reste 47208 trajets différents. Sur ces 47208 trajets restants, 1981 trajets ont :

- Une correspondance avec au moins un pattern parmi la liste des patterns
- Au moins un jumelage, i.e. l'un des deux attributs `matched_rider_routes` et `matched_driver_routes` n'est pas vide. Ceci permet de comparer les résultats des algorithmes développés et de l'algorithme de Netlift.

Ces 1981 trajets peuvent donc avoir 0 ou plusieurs jumelages en tant que passager et 0 ou plusieurs jumelages en tant que conducteur, sachant que, par observation, le nombre maximum de jumelages trouvé par l'algorithme actuel de Netlift (passager ou conducteur) pour un trajet donné de la base de données est de l'ordre de la dizaine.

2.3 Algorithme initial

L'algorithme développé permet de jumeler un trajet passager donné à un trajet conducteur tiré d'une liste de trajets conducteurs donnée. Différents critères sont envisagés dans ce premier chapitre pour classer les différents trajets conducteurs en fonction de leur potentiel de covoiturage. Voici le schéma de l'algorithme proposé dans ce premier chapitre.

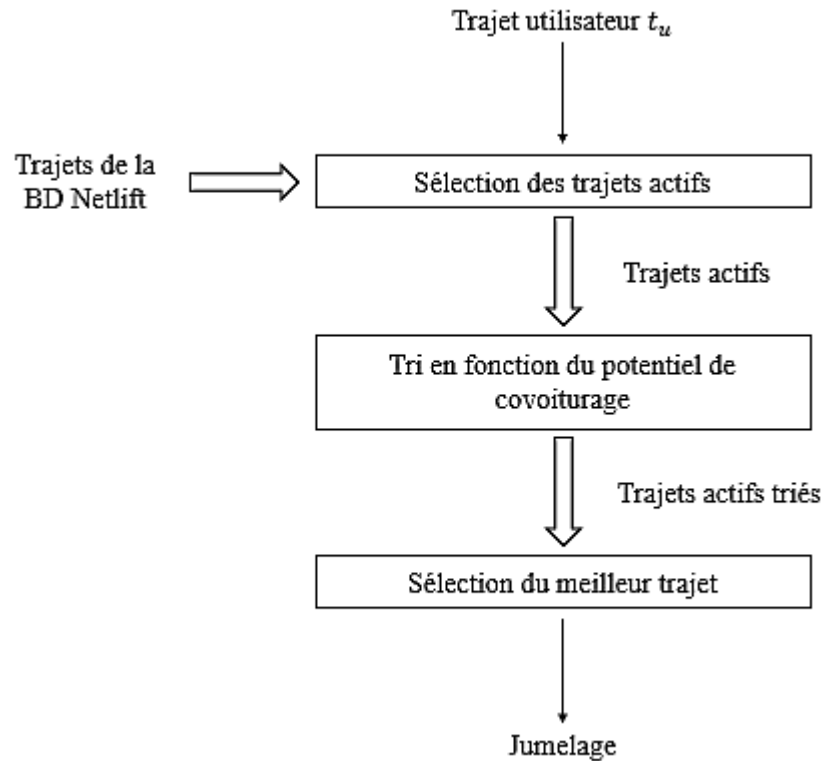


Figure 2.7 : Algorithme initial

En entrée, une demande de covoiturage est associée à un trajet utilisateur t_u .

La première phase de l'algorithme consiste à sélectionner les trajets de la base de données Netlift qui sont susceptibles d'être jumelés avec le trajet utilisateur. Ce sont les trajets actifs :

- Un trajet actif pour t_u est un trajet qui est dans une fenêtre de temps de +/- 2h par rapport à l'attribut When de t_u (heure de départ ou d'arrivée).
- De plus, un trajet est actif pour t_u si parmi u et le titulaire du trajet actif, il y a au moins un conducteur.

La deuxième phase permet de trier ces trajets actifs en fonction de leur potentiel de covoiturage. Une partie est consacrée à l'étude et à la sélection du meilleur critère de tri possible. Cette phase fait appel à la structure de données développée à l'avance et détaillée dans la partie précédente.

La troisième et dernière phase de l'algorithme permet de sélectionner le meilleur trajet parmi ces trajets actifs triés. Les calculs liés à cette phase demandent plus de temps et de puissance de calcul

qu'à la phase précédente, puisqu'il faut calculer des itinéraires précis. C'est ici que la multimodalité est prise en compte, et que des trajets mixant transports en commun et covoiturage sont étudiés.

En sortie, le jumelage pour le trajet utilisateur initial est obtenu. D'autres composantes sont à considérer pour le jumelage proposé à l'utilisateur (acceptation non garantie, autres contraintes de préférences). Cet algorithme initial a pour vocation de détailler les contours de l'algorithme plus général présenté dans le chapitre 3.

L'algorithme présente deux avantages par rapport à l'algorithme de Netlift. Premièrement, il est plus rapide puisqu'il fait appel à une structure de données construite au préalable. Les phases 1 et 2 permettent l'accès rapide à des trajets dont le potentiel de covoiturage est bon. Ainsi en phase 3, les calculs « complexes » sont effectués sur un petit nombre de trajets identifiés. Ces calculs se veulent plus élaborés que ceux réalisés par l'algorithme de Netlift – c'est le second avantage : les jumelages proposés sont plus pertinents, puisqu'ils sont plus précis.

2.4 Tests de l'algorithme initial

Le but de la phase 2 de l'algorithme est d'effectuer un classement des trajets actifs de manière à repérer les trajets dont le potentiel de covoiturage est le meilleur. Un critère efficace permettant de comparer deux trajets est donc nécessaire. Il faut qu'il soit basé sur des calculs relativement complexes pour avoir une précision importante, mais suffisamment simples pour pouvoir les réaliser dans un court laps de temps. Deux critères de tri différents ont été étudiés. Pour évaluer chacun des critères de tri, dans un premier temps il a été question de comparer le classement des trajets actifs aux « jumelages » obtenus par l'algorithme de Netlift. Ce critère n'étant pas suffisant, un autre outil de comparaison a été utilisé, inspiré de la définition qui sera faite de « détour ».

2.4.1 Un score comme critère de tri

2.4.1.1 Définitions

Cette première approche s'appuie sur la structure de données détaillée dans la partie 1. En effet, pour comparer deux trajets, les deux patterns qui les représentent respectivement sont comparés. Le score va représenter le potentiel de covoiturage entre deux trajets (et donc deux patterns). Concrètement, plus ce score est grand, plus le potentiel de covoiturage est grand.

Le but est d'obtenir une liste de trajets actifs classés par ordre de score décroissant, comme dans le tableau suivant :

Tableau 2.1 : Classement de trajets actifs - Exemple

Trajet actif t_a (ici $1 \leq a \leq 5$)	Score entre t_u et t_a
t_1	10 000
t_2	7 000
t_3	6 000
t_4	5 500
t_5	5 000

Plusieurs scores ont été testés :

- **Score simple**

L'idée de ce score est d'incrémenter le score en fonction du nombre de zones en commun.

$$\text{Score simple} = \sum_i c_i * 10^i, \text{ avec } c_i \text{ le nombre d'ensemble de } i \text{ zones en commun}$$

Tableau 2.2 : Score simple - Exemple

Pattern du trajet t1	5	10	19	3	4	8	6	12
Pattern du trajet t2	5	10	19	4	7	8	6	

Dans l'exemple ci-dessus, les deux patterns ont 5 singletons de zones en commun, 3 doublons de zones en commun, 1 ensemble de trois zones en commun, et 0 nombre d'ensemble de quatre zones en commun. Dans cet exemple, le score simple est égal à 1350.

Intuitivement, dans l'exemple, il semble problématique que le fait d'avoir un ensemble de trois zones en commun majore forcément le fait qu'il y ait des doublons ou des singletons en commun.

- **Score complexe**

Ce score dépend de la concordance des zones, et ce pour différentes positions des patterns (début du pattern de plus courte longueur qui varie). Le score complexe représente alors le score obtenu avec la meilleure position.

Tableau 2.3 : Score complexe - Exemple

Pattern du trajet t1	5	10	19	3	4	8	6	12
Pattern du trajet t2	5	10	19	4	7	8	6	
Concordance des zones	1	1	1	0,5	0	1	1	

Si les zones sont les mêmes, 1 est inscrit dans la ligne « concordance des zones ». Si les zones sont voisines, 0,5 est inscrit. Dans l'exemple ci-dessus, les zones 3 et 4 sont voisines.

Au fur et à mesure que le pattern de plus courte longueur est parcouru, un compteur de points et un compteur de zones successives sont mis à jour. Par exemple, le premier compteur de points pour l'exemple est égal à 3,5 (zones 5 à 4) et le compteur de zones successives est égal à 4 (prise en compte des zones voisines). Le deuxième compteur de points est égal à 2, et le deuxième compteur de zones successives est égal à 2 aussi (zones 8 et 6). Une combinaison de ces compteurs permet alors d'obtenir le score.

Il en existe plusieurs en fonction des paramètres ci-dessous :

- Impact de zones égales en début ou en fin de parcours : on rajoute un facteur pour les zones qui sont égales en début ou en fin de trajet.
- Prise en compte des zones voisines (apparition des 0,5 pour la concordance des zones)
- Type de combinaison entre le compteur de points et le compteur de zones successives (multiplication, exponentiel, etc.)

Voici un exemple de combinaison :

$$\text{Score complexe} = \sum_i \text{compteur_points}_i * \text{compteur_zones_successives}_i$$

2.4.1.2 Premier critère d'évaluation : couverture des jumelages Netlift

La BD Netlift est utilisée pour évaluer différents types de scores. Celle-ci compte 1981 trajets qui possèdent des jumelages selon l'algorithme Netlift (cf. 2.1.2). La couverture des jumelages Netlift représente le nombre de trajets sur 1981 dont les jumelages (trouvés par l'algorithme de Netlift) sont inclus dans la liste de trajets actifs du trajet en question, triés par ordre de score décroissant.

2.4.1.2.1 Résultats

Plusieurs critères permettent de faire varier la couverture des jumelages Netlift, comme le nombre de trajets actifs considéré, (10 premiers, 20 premiers, etc.), ou encore le type de couverture (tous les jumelages couverts pour un trajet, ou au moins un des jumelages couverts).

La comparaison entre les deux algorithmes a été effectuée uniquement sur ces 1981 trajets. Les premiers tests ont été réalisés pour trois scores différents :

- Score complexe 1 : $\sum_i \text{compteur_points}_i * \text{compteur_zones_successives}_i$
- Score complexe 2 : $\sum_i \text{compteur_points}_i * \text{compteur_zones_successives}_i^{10}$
- Score simple

Lecture des tableaux : « 10 premiers trajets actifs considérés » signifie que pour chaque trajet utilisateur t_u , les 10 meilleurs trajets actifs en termes de score sont sélectionnés.

Tableau 2.4 : Couverture des jumelages Netlift - Tous

Nombre de trajets actifs considérés pour t_u	Couverture des jumelages Netlift – Tous les jumelages		
	Score complexe 1	Score complexe 2	Score simple
10	699/1981	686/1981	757/1981
20	954/1981	911/1981	1178/1981
30	1293/1981	1326/1981	1438/1981
40	1673/1981	1659/1981	1686/1981
50	1811/1981	1807/1981	1818/1981
60	1895/1981	1914/1981	1892/1981
70	1927/1981	1928/1981	1915/1981
80	1934/1981	1937/1981	1936/1981
90	1956/1981	1968/1981	1969/1981
100	1974/1981	1981/1981	1981/1981

Tableau 2.5 : Couverture des jumelages Netlift - Au moins un

Nombre de trajets actifs considérés pour t_u	Couverture des jumelages Netlift – Au moins un jumelage		
	<i>Score complexe 1</i>	<i>Score complexe 2</i>	<i>Score simple</i>
10	1377/1981	1370/1981	1451/1981
20	1619/1981	1622/1981	1754/1981
30	1758/1981	1786/1981	1819/1981

Pour près de 70 % des trajets utilisateurs, tous les jumelages Netlift sont inclus dans les 30 premiers trajets actifs, et ce pour les trois différents scores. Les jumelages aberrants, cités dans la partie précédente, influencent peut-être les résultats lorsque le nombre de trajets actifs considérés est faible (10 à 30).

C'est en partie confirmé par le tableau 2.5 Les résultats sont meilleurs pour 10, 20 et 30 trajets actifs considérés. En effet 90% de trajets utilisateurs ont au moins un jumelage Netlift inclus dans les 30 premiers trajets actifs.

2.4.1.2.2 Recherche du meilleur score

Plusieurs scores ont été testés, de manière à obtenir les meilleurs résultats possibles en termes de couverture des jumelages Netlift. Une observation du tableau 2.4 permet d'affirmer que le score simple est le meilleur (à priori le moins complexe, et le moins intuitif). Il a en effet un meilleur taux de couverture que les deux autres scores testés (scores 1 et 2).

La recherche du meilleur score a été réalisée par méthode essai-erreur dans un premier temps, et fut portée sur le score complexe, étant le seul des deux scores ayant des paramètres modifiables.

Ainsi plusieurs combinaisons de paramètres ont été testées, dont plusieurs d'entre elles s'avèrent être meilleures que le score simple. En résumé, l'impact des zones voisines n'est pas aussi significatif que cela pourrait l'être intuitivement, ce qui est sûrement dû au fait que le découpage en zones de Montréal est assez irrégulier. L'impact des facteurs multiplicatifs bonifiant les zones similaires en début et en fin de patterns pour différents trajets, est quant à lui assez important. Une combinaison judicieuse permet d'obtenir de meilleurs résultats. Et pour ce qui est du type

d'opérations entre le compteur de points et le compteur de zones successives, il n'y a pas de différence notable.

2.4.1.2.3 Conclusion

Des résultats satisfaisants sont obtenus puisque les mêmes jumelages que l'algorithme de Netlift sont globalement identifiés, et ce de manière plus rapide grâce à la structure de données construite à l'avance.

Cependant, il est difficile de trouver le meilleur score en termes de couverture des jumelages Netlift. Par ailleurs, des anomalies sont à relever parmi les jumelages trouvés par Netlift, et bien que certains de ces problèmes aient été fixés dans des versions ultérieures de l'algorithme, ce système de comparaison peut donc être sujet à la qualité des jumelages trouvés par l'algorithme de Netlift. La couverture des jumelages Netlift n'est donc pas forcément le meilleur critère de comparaison des scores, d'où la nécessité d'en trouver un plus efficace.

2.4.1.3 Second critère d'évaluation : le détour

2.4.1.3.1 Définition

Le détour pour deux trajets $t_1 = O_1 \rightarrow D_1$ et $t_2 = O_2 \rightarrow D_2$ est défini par :

$$\text{Détour}(t_1, t_2) = \text{Min} \left(\frac{\text{temps_covoiturage}(t_1, t_2)}{\text{temps_solo}(t_1)}, \frac{\text{temps_covoiturage}(t_2, t_1)}{\text{temps_solo}(t_2)} \right) * 100$$

$$\text{temps_covoiturage}(t_1, t_2) = \text{temps en min du trajet } O_1 \rightarrow O_2 \rightarrow D_2 \rightarrow D_1$$

$$\text{temps_solo}(t_1) = \text{temps en min du trajet } O_1 \rightarrow D_1$$

Par exemple, un détour de 120% représente un temps de trajet de 20% en plus pour le conducteur. C'est le critère optimum représentant le potentiel de covoiturage pour deux trajets sur lequel la méthodologie d'évaluation des différents scores est basée. On ne considère à ce stade que des trajets effectués en auto. Le covoiturage multimodal n'est pas pris en compte.

2.4.1.3.2 Méthodologie d'évaluation des différents scores

Ainsi deux classements différents pour les trajets actifs sont obtenus et comparés : le classement via le score (que l'on souhaite évaluer) et le classement par le détour, calculé à l'aide de l'API Google Maps, comme dans les tableaux 2.6 et 2.7 :

Tableau 2.6 : Classement via le score

Trajet actif t_a	Score entre t_u et t_a
t_1	10 000
t_2	7 000
t_3	6 000
t_4	5 500
t_5	5 000

Tableau 2.7 : Classement via le détournement

Trajet actif t_a	Détournement entre t_u et t_a
t_2	105%
t_1	110%
t_3	120%
t_5	125%
t_4	130%

Comme expliqué plus tôt le tableau 2.7 classant les trajets actifs grâce au détournement est considéré comme le tableau optimal à atteindre.

Soit un ensemble (t_i) de n trajets jumelés à un trajet utilisateur t_u . Chaque paire (t_u, t_i) correspond donc à un jumelage.

Soit p , un nombre entier compris entre 0 et n . Voici la méthode d'évaluation :

1. La valeur de détournement du $p - ième$ jumelage dans le tableau 2.7 classé par ordre de détournement croissant est sélectionnée. C'est la valeur de référence v_{ref} .
2. Puis, dans le tableau 2.6 classé par ordre de score décroissant, le nombre de jumelages dont le détournement est inférieur à v_{ref} et qui sont classés après le $p - ième$ jumelage est relevé.

C'est m , le nombre de jumelages dont le détournement est « bon » relativement à v_{ref} , mais qui ne sont pas classés dans les p premiers jumelages (du classement via le score), alors qu'ils le devraient (idéalement, étant donné que notre critère optimal est le détournement).

Un pourcentage est alors calculé pour représenter la qualité d'un score :

$$Qualité du score = \frac{m}{p}$$

Celui-ci est toujours inférieur ou égal à 1 puisque $m \leq p$.

Tableau 2.8 : Méthodologie d'évaluation- Exemple

Lecture du tableau : parmi les trajets encadrés, il faut regarder ceux qui devraient être classés parmi les trajets entre parenthèses et qui ne le sont pas. Ici, il y a t_1 et t_4 qui devraient être dans les 5 premiers (p premiers). Il y en a donc 2/5 qui ne sont pas bien classés, ce qui fait 40%.

t_i classés par ordre de détour décroissant calculé via l'API de Google Maps		t_i classés par ordre de score décroissant		
(t_1)	d_1	t_2	s_2	d_2
(t_2)	d_2	t_3	s_3	d_3
(t_3)	d_3	t_7	s_7	d_7
(t_4)	d_4	t_5	s_5	d_5
(t_5)	$d_5 = v_{ref}$	t_6	s_6	d_6
t_6	d_6	t_1	s_1	d_1
t_7	d_7	t_8	s_8	d_8
t_8	d_8	t_4	s_4	d_4
t_9	d_9	t_{10}	s_{10}	d_{10}
t_{10}	d_{10}	t_9	s_9	d_9

2.4.1.3.3 Résultats

Ici 1000 trajets de l'enquête OD sont classés via un score complexe pour un trajet utilisateur t_u donné.

Tableau 2.9 : Évaluation du score par le détour - Résultat

p	Qualité du score
100	59%
200	38%
300	35%
400	30.25%
500	21.6%
...	...
1000	0%

Lecture du tableau : à la première ligne, pour le paramètre $p = 100$, il y a 59% des 100 premiers trajets qui sont classés après la 100^{ème} place dans le tableau de classement via le score, alors que leurs positions respectives sont dans les 100 premières places du classement via le détour.

Les résultats sont du même type pour toutes les sortes de scores. Ils ne sont pas satisfaisants : trop de trajets sont mal classés via le score. En définitive, le score ne semble pas être un bon critère de tri pour les trajets actifs.

2.4.2 Une estimation du détour comme critère de tri

Comme le détour représente notre critère optimum représentant la qualité d'un jumelage, il est logique de penser à l'utiliser comme critère de tri. Cependant, ce détour est calculé via l'API de Google Maps, et s'il était utilisé comme critère de tri, il faudrait effectuer beaucoup de requêtes. Celles-ci ne sont pas illimitées et sont payantes au-delà d'une certaine limite journalière (1250 requêtes par jour).

Un calcul simple permet de se rendre compte du coût d'un tel système. Il y a environ 100 zones pour le découpage de la ville de Montréal, ce qui conduit donc à $100 * 100 = 10^4$ patterns différents. Pour chaque paire de patterns, il faut effectuer une requête (plus d'une en pratique), soit $10^4 * 10^4 = 10^8$ requêtes au total. Chaque requête à un coût de 0,0005 centimes. Le coût total de l'opération serait donc de $10^8 * 0.5 * 10^{-4} = 50\ 000$ \$.

Il paraît alors nécessaire de trouver un autre moyen pour pouvoir évaluer ce détour. Ceci va être réalisé à l'aide d'une régression linéaire multiple, visant à identifier des variables relatives aux trajets (et donc aux patterns les représentant) qui permettent d'expliquer les variations du détour.

2.4.2.1 Modèles de régression linéaire multiple

Le but est de trouver des variables qui pourraient expliquer les variations du détour entre deux trajets. Voici les variables suivantes à disposition obtenues de manière automatique et gratuite :

- x_1 : Nombre de zones en commun entre les deux patterns des deux trajets
- x_2 : Nombre maximum de zones successives en commun (prise en compte des zones voisines)
- x_3 : Distance entre les zones d'origines en km
- x_4 : Distance entre les zones de destinations en km
- x_5 : Différence de nombre de zones entre les deux patterns des 2 trajets

Des modèles du type $Détour_estimé(x_i) = \alpha + \sum_i \beta_i x_i$ sont construits, où les x_i sont les variables explicatives sélectionnées et α et β_i sont des constantes qui dépendent des variables x_i sélectionnées et de l'échantillon considéré pour la régression linéaire multiple.

Ainsi, deux classements différents pour les trajets actifs sont obtenus et comparés : le classement via le détour estimé (à évaluer) et le classement par le détour, calculé à l'aide de l'API Google Maps.

Tableau 2.10 : Classement via le détour estimé Tableau 2.11 : Classement via le détour

Trajet actif t_a	Détour estimé (t_u, t_a)	Trajet actif t_a	Détour (t_u, t_a)
t_1	110%	t_2	105%
t_2	120%	t_1	110%
t_3	125%	t_3	120%
t_4	128%	t_5	125%
t_5	150%	t_4	130%

Comme pour l'évaluation des scores, il est à noter que ces deux classements sont différents. La même méthodologie est utilisée pour évaluer les modèles de régression que pour évaluer les différents scores, i.e. avec un pourcentage représentant la qualité du modèle et dépendant des deux tableaux.

2.4.2.2 Test et résultats

Les régressions linéaires multiples sont effectuées à l'aide de l'outil Excel correspondant, et d'échantillons choisis. Chaque échantillon représente une base de données, dont chaque entrée contient les variables explicatives x_i pour deux trajets différents et la valeur du détour calculé via l'API de Google Maps pour ces deux trajets. Deux échantillons sont utilisés pour amener des modèles de régression différents :

- Premier échantillon comprenant des jumelages obtenus par l'algorithme de Netlift, dont la moyenne du détour se situe autour de 175%.

- Deuxième échantillon avec des jumelages moins bons, tirés de l'enquête OD de 2008, dont la moyenne du détour est environ égale à 260%.

La moyenne des coefficients de détermination des modèles présentés ci-après est proche de 60%, ce qui signifie que 60% des variations de notre variable à expliquer (le détour) sont expliquées par 60% des variations des variables explicatives sélectionnées dans les modèles. Quel que soit l'échantillon utilisé, il n'y a pas de modèle réellement efficace dont le coefficient de détermination serait très élevé. D'autres modèles ont été testés mais ne présentent pas l'intérêt de figurer ici.

Lors de l'élaboration des modèles, il fallait s'attendre à un terme constant avec des β_i tous négatifs (sauf pour la variable « différence de nombre de zones »), puisque chacune de ces variables tend intuitivement à faire diminuer le détour. Or, en pratique, des comportements assez aléatoires sont relevés, avec des β_i négatifs ou positifs. Pour point de satisfaction cependant, β_1 et β_2 sont quasiment toujours négatifs et avec des valeurs assez élevées expliquant peut-être leur influence sur le détour (ce qui était à priori intuitif). Les résultats obtenus pour les deux échantillons sont présentés dans les parties suivantes.

2.4.2.2.1 Modèles de régression sur le 1^{er} échantillon

Le tableau 2.12 présente les statistiques descriptives pour les 600 trajets sélectionnés dans ce premier échantillon. Il est à remarquer qu'il n'est pas gênant d'observer des disparités importantes, avec notamment des écarts-types élevés, puisque le but des modèles est d'identifier aussi les forts potentiels de covoiturage comme les moins bons, dans une optique de classement.

Tableau 2.12 : Statistiques descriptives pour le premier échantillon

	x_1	x_2	x_3	x_4	x_5
Moyenne	0,72	1,03	16	45	3,31
Médiane	0	1	11	41	3
Écart type	0,99	0,80	10	10	1,85
Écart type en pourcentage à la moyenne	1,38	0,78	0,65	0,21	0,56

Ici, les tests sont effectués sur un premier échantillon de 600 trajets sélectionnés parmi la BD Netlift, dont le détour estimé est calculé pour un trajet utilisateur t_u donné. Les variables explicatives sélectionnées pour chaque modèle de régression sont indiquées dans la légende. Les trois modèles suivants sont construits puis évalués sur le même échantillon.

Les résultats de chacune des régressions sont similaires, comme expliqué en 2.4.2.2. Des coefficients de détermination de 0.6 sont obtenus dans les meilleurs cas (cf. Annexe B pour le cas du modèle du tableau 2.14), mais sont relativement loin de 1 pour affirmer que les modèles établis soient de bonne qualité.

Tableau 2.13 : Modèle de régression linéaire avec x_1, x_2, x_3, x_4, x_5

p	<i>Qualité du score</i>
Premier décile : 60	40%
100	33%
200	25.5%
300	19.67%
400	13.5%
500	10.2%
600	0%

Tableau 2.14 : Modèle de régression linéaire avec x_3, x_4, x_5

p	<i>Qualité du score</i>
Premier décile : 60	40%
100	31%
200	25%
300	187%
400	12.75%
500	10.2%
600	0%

Tableau 2.15 : Modèle de régression linéaire avec x_1, x_2, x_5

p	<i>Qualité du score</i>
Premier décile : 60	71.67%
100	55%
200	45.5%
300	34%
400	24%
500	16.6%
600	0%

2.4.2.2.2 *Modèle de régression sur le 2ème échantillon*

Voici les statistiques descriptives pour le deuxième échantillon.

Tableau 2.16 : Statistiques descriptives pour le deuxième échantillon

	x_1	x_2	x_3	x_4	x_5
Moyenne	1,17	1,47	43	18	7,21
Médiane	0	1	43	16	8
Écart type	1,26	1,33	14	8	2,51
Écart type en pourcentage à la moyenne	1,08	0,90	0,33	0,45	0,35

Le second échantillon comporte 1000 trajets sélectionnés parmi la base de données de l'enquête OD de 2008, dont le détour estimé est calculé pour un trajet utilisateur t_u donné. Le modèle suivant est construit et évalué sur cet échantillon. Les mêmes résultats qu'auparavant sont obtenus quant à la qualité du modèle.

Tableau 2.17 : Modèle de régression linéaire avec x_1 , x_2 , x_5

p	<i>Qualité du score</i>
Premier décile : 100	45%
200	29%
300	25.33%
400	22.25%
500	17%
...	...
1000	0%

Le test suivant est effectué sur l'échantillon 1 mais avec le modèle de régression précédent.

Tableau 2.18 : Modèle de régression linéaire avec x_1 , x_2 , x_5

p	<i>Qualité du score</i>
Premier décile : 60	40%
100	34.5%
200	28%
300	24.33%
400	19.25%
500	12%
600	0%

2.4.2.2.3 Résultats

Les résultats sur les deux échantillons sont assez similaires, bien que les échantillons soient différents. Globalement, les modèles de régression sont de meilleurs critères de comparaison que le score développé dans la partie précédente. En effet, environ 40 à 45% du premier décile de trajets sont mal classés pour les meilleurs modèles de régression, contre près 60% pour le score.

Les résultats pour la régression ne sont cependant pas encore assez satisfaisants, avec pour cause partielle la qualité des modèles développés.

2.4.3 Synthèse des tests

Les critères de tri développés (score et détour) ne sont pas suffisants pour conclure quant à l'efficacité de la phase 2 de l'algorithme qui consiste à trier les trajets actifs pour accéder aux trajets dont le potentiel de covoiturage est le plus fort. Théoriquement, l'algorithme semble bon, mais en pratique, il est difficile d'obtenir ces trajets à fort potentiel de covoiturage, sans faire des calculs plus poussés. Les résultats sont logiques puisque le classement des trajets actifs, effectuée de manière « grossière » via le score ou le détour estimé, est comparé à un classement « précis », via le détour calculé par l'API de Google Maps. Cette phase 2 se veut donc trop précise.

2.5 Conclusion

L'étude de l'algorithme initial fut stoppée au vu des résultats concernant les critères de tri des trajets actifs. La partie suivante s'intéresse à un autre algorithme et une autre structure de données, qui s'appuient cependant sur l'idée générale développée pendant ce chapitre.

CHAPITRE 3 STRUCTURE DE DONNÉES ET ALGORITHME FINAUX

3.1 Structure de données

L'idée générale développée dans le chapitre 2 d'avoir une structure de données représentant la ville de Montréal est conservée. La structure de donnée retenue est plus précise que celle imaginée dans le chapitre précédent et permet de faire des calculs d'itinéraires.

3.1.1 Graphe multimodal

L'algorithme de ce chapitre s'appuie sur un graphe mixte multimodal $G_M = (V_M, E_M)$. Il est l'union de trois graphes différents : un graphe routier non orienté $G_R = (V_R, E_R)$, un graphe représentant les transports en commun non orienté $G_{TC} = (V_{TC}, E_{TC})$, et un graphe de transition orienté $G_T = (V_T, E_T)$ qui permet de relier les deux graphes précédents.

Les poids des arêtes du graphe sont les temps de parcours entre les différents nœuds, que ce soit en auto pour G_R , en transports en commun pour G_{TC} , ou à pied, notamment pour G_T .

G_M représente la ville de Montréal de manière approximative de sorte que sa conception ne soit pas trop complexe et chronophage. Il doit cependant être suffisamment précis pour que les calculs réalisés et les décisions prises soient correctes.

3.1.1.1 Graphe routier

Les nœuds du graphe routier sont de deux types différents :

- Les centroïdes sont des nœuds disposés à intervalles réguliers grâce à un quadrillage de Montréal. Chaque zone de 2km² - un carré noir sur la figure 3.1 - contient un centroïde en son centre. Ceux-ci sont reliés entre eux par des arêtes du graphe routier dont les poids respectifs sont égaux à un temps de parcours moyen. Il est égal à la moyenne des temps de parcours de 100 paires de points (10 points disséminés dans chaque zone de manière régulière) dans les deux zones correspondantes aux centroïdes. Le nombre de zones est de 8010 au total.
- Les nœuds artériels qui appartiennent au réseau routier artériel (autoroutes, routes principales) sont quant à eux représentés de manière exacte. En effet, les arêtes entre ces nœuds ont des temps de parcours relativement faibles par rapport à deux points pris au hasard. Ils permettent d'apporter plus de précision aux calculs de plus courts chemins réalisés dans G_M . Les temps de

parcours des arêtes entre centroïdes et nœuds artériels sont approximés par la moyenne des temps de parcours entre les 10 points sélectionnés pour la zone et le nœud artériel.

Voici le quadrillage de la ville de Montréal avec un centroïde (cercle blanc avec un point noir) au centre de chacune des zones, et les arêtes entre les différents centroïdes. Ici le réseau artériel n'apparaît pas par souci de simplicité.



Figure 3.1 : Quadrillage de la ville de Montréal

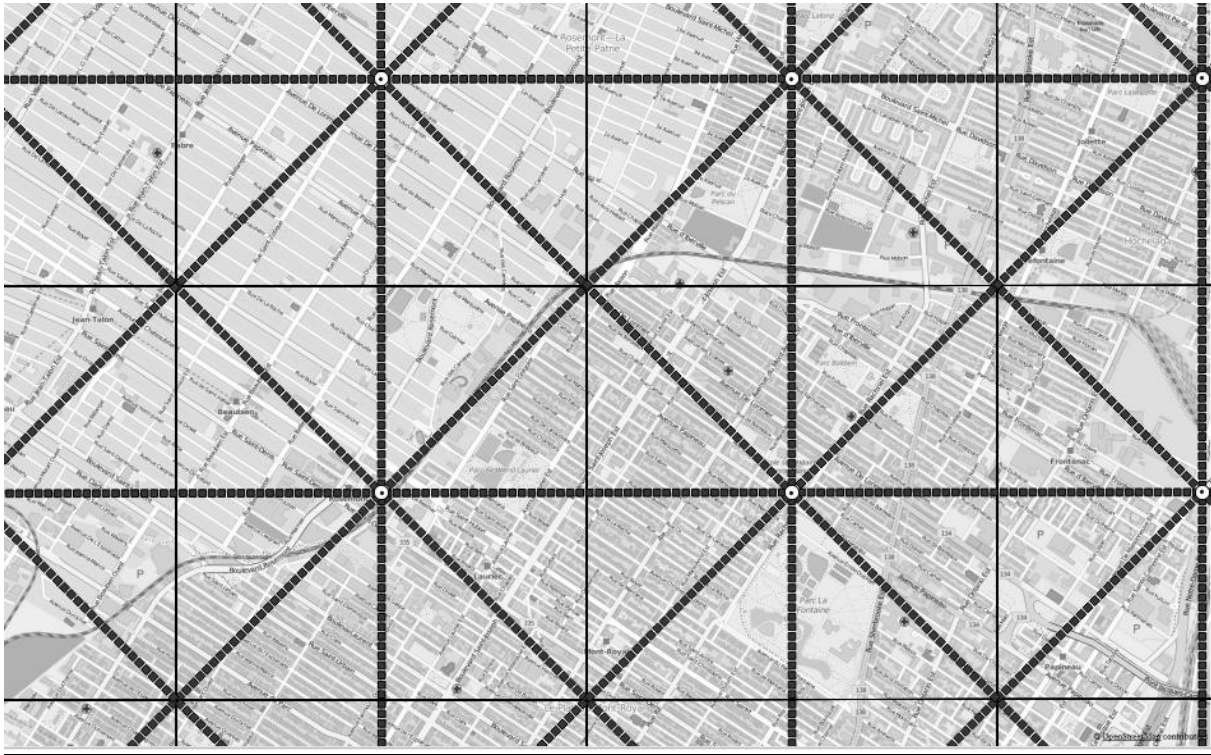


Figure 3.2 : Graphe routier

3.1.1.2 Graphe pour les transports en commun

Les nœuds de G_{TC} représentent les stations de transports en commun (bus, métro, train). Ils sont limités aux stations de l'île de Montréal et aux stations de trains de banlieue. Les arêtes entre ces différentes stations sont les liens entre les différentes stations, dont les poids sont égaux aux temps de parcours. Il existe également des arêtes de correspondance entre différents nœuds de G_{TC} , si ceux-ci sont suffisamment proches (200m de distance) pour effectuer le changement à pied.

Dans la figure 3.3, les arêtes relatives au graphe G_{TC} ont été rajoutées en pointillés. Toujours par souci de simplicité, les nœuds et les arêtes ne sont pas tous représentés.

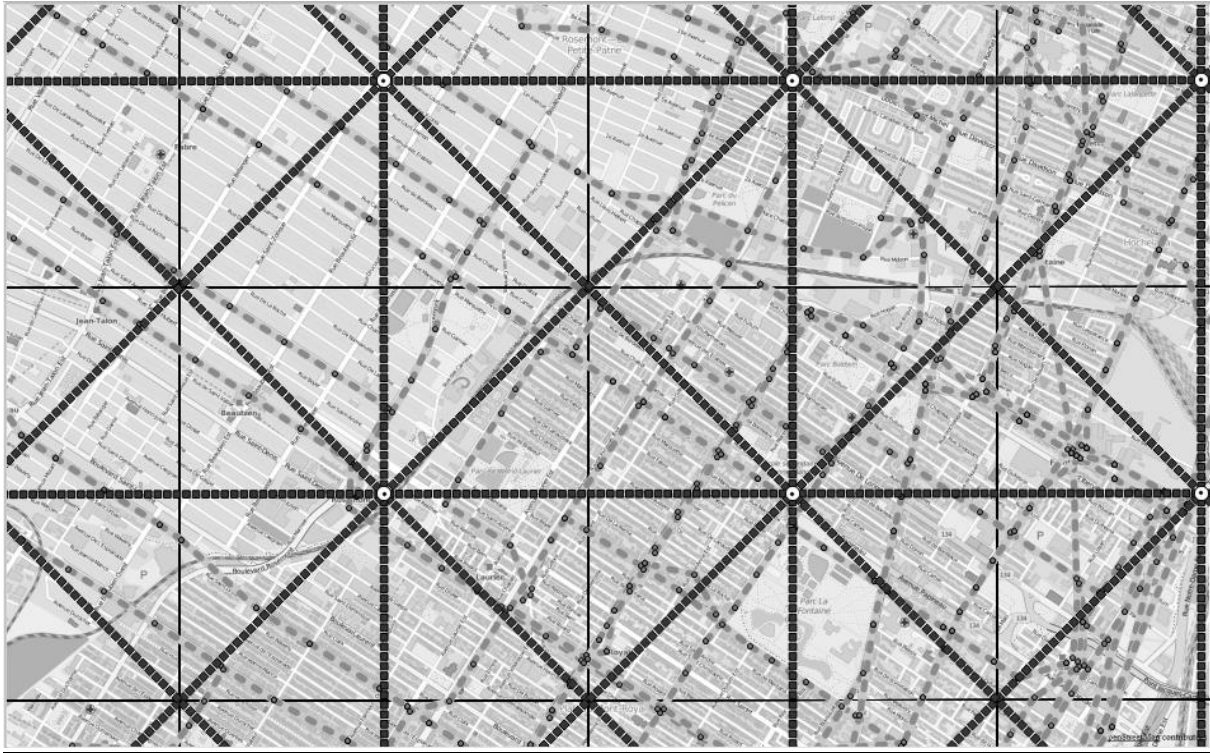


Figure 3.3 : Graphe routier et graphe TC

3.1.1.3 Graphe de transition

Le graphe de transition G_T permet de relier G_R et G_{TC} . Il n'y a pas de nœuds pour ce graphe, mais seulement des arêtes qui relient certains nœuds du graphe G_{TC} au graphe G_R . Les nœuds de G_{TC} qui ont un parking ou une station essence à proximité sont reliés au graphe routier. Ainsi lors de trajets multimodaux, le conducteur dispose d'un lieu de dépose ou de ramassage sécuritaire pour son passager, à proximité de la station de transport en commun faisant partie du trajet du passager. Ces nœuds particuliers peuvent être aussi utilisés pour des déposer ou récupérer des passagers covoiturant de manière classique. Les poids des arcs de transition sont fixés à 1 min dans le sens TC-Routier, puisqu'il n'y a pas de temps d'attente. Dans le sens Routier-TC, ces poids prennent en compte les temps d'attente aux stations de transports en commun.

Les arêtes de transition partant de chaque centroïde sont représentées dans la figure ci-après.

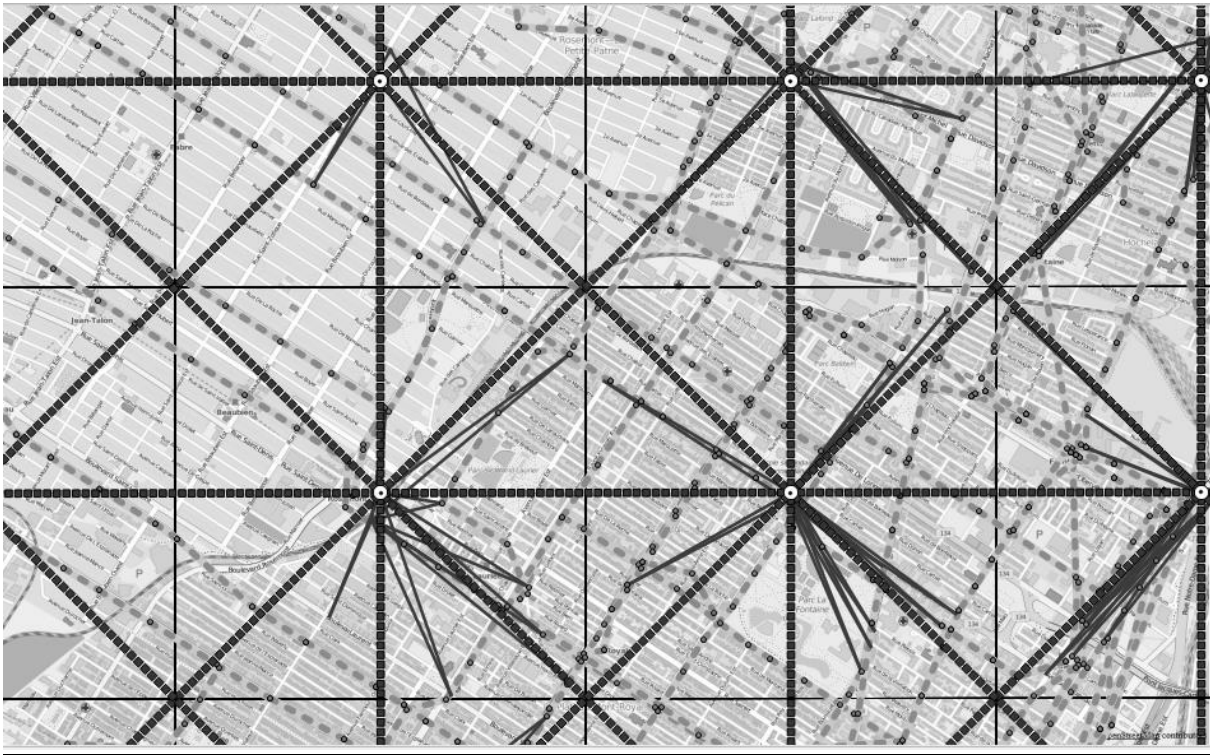


Figure 3.4 : Graphe G_M

3.1.1.4 Chargement des données pour la construction du graphe multimodal

La construction de ce graphe est réalisée par un autre étudiant – Thibaut Jaurou – travaillant sur le projet. Les données utilisées proviennent de différentes sources :

- Réseau artériel de la CMM (Communauté Métropolitaine de Montréal) et OSM (OpenStreetMap) pour le graphe routier G_R . OTP (OpenTripPlanner) pour la génération des temps de parcours.
- Fichiers GTFS (General Transit Feed Specification) de la STM et de l'AMT pour les temps de parcours du graphe G_{TC} .

Les temps de parcours des différentes arêtes de G_R et G_{TC} sont générés par Netlift via un serveur OpenTripPlanner.

Le fichier fourni pour le graphe multimodal G_M est un fichier .csv qui est converti au format .xlsx pour être utilisé dans l'application codée en JAVA. Ce fichier comporte deux feuilles, l'une correspondant aux nœuds du graphe et l'autre comprenant les arêtes du graphe. Les attributs

suivants sont ceux qui sont utilisés comme variables d’instance pour les objets JAVA représentant les nœuds et les arêtes :

Nœuds

Id_vertex	Type_vertex	Lon_vertex	Lat_vertex	Info1_vertex	Info2_vertex
-----------	-------------	------------	------------	--------------	--------------

Arcs

Id_edge	Id_vertex1	Id_vertex2	Temps_parcours	Type_edge	Info1_edge
---------	------------	------------	----------------	-----------	------------

3.1.2 Représentation des trajets OD

Le nœud centroïde de G_R le plus proche de l’origine (resp. destination) du trajet OD est associé à un trajet OD donné. Il est noté n_O (resp. n_D).

3.1.3 Algorithme de recherche de chemins multimodaux

Pour pouvoir proposer des jumelages multimodaux, il est nécessaire de pouvoir calculer des itinéraires multimodaux avec des tronçons routiers et des tronçons en transport en commun.

3.1.3.1 Description

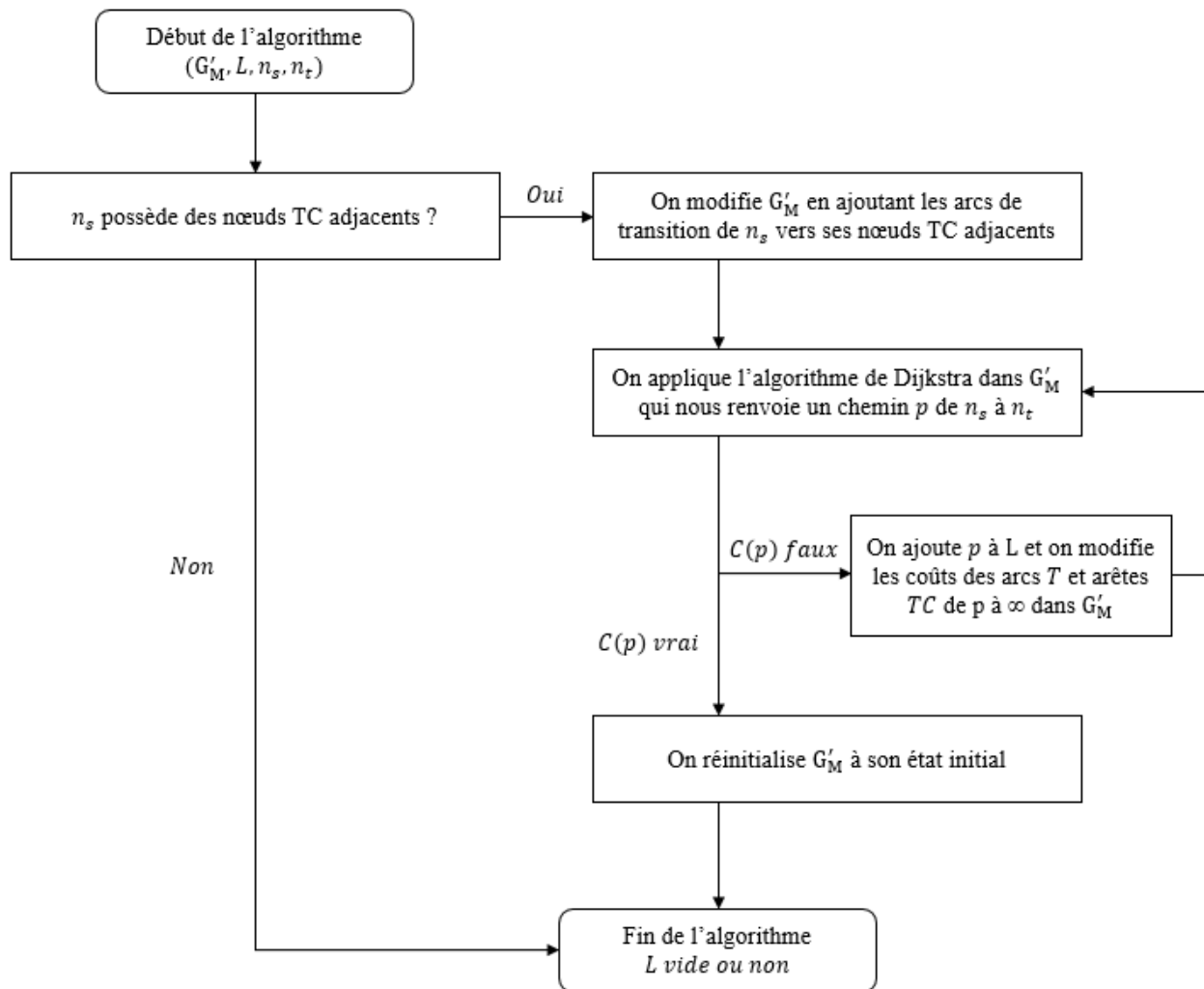
La structure de données que représente le graphe G_M permet de calculer des chemins multimodaux pour un passager donné. Les chemins considérés sont limités à deux arcs de transition, le premier pour entrer dans le réseau de transports en commun, le second pour en sortir, i.e. des chemins de type Routier-TC ou de type TC-Routier.

La conception de l’algorithme de recherche de chemin multimodaux a été notamment motivée par le fait que ces chemins Routier-TC et TC-Routier représentent des cas particuliers, ne nécessitant pas de développer un algorithme complexe de recherche de chemin. Au contraire, cette heuristique est basée sur l’algorithme de Dijkstra – qui est un algorithme de plus court chemin – et sur une modification astucieuse – décrite par la suite – du graphe G_M dépendante du passager pour lequel des chemins multimodaux sont recherchés.

Un graphe auxiliaire G'_M est utilisé et modifié pour chaque passager. A son état initial, c’est une copie quasi-similaire du graphe G_M , à deux différences près. La première concerne les arêtes routières qui ont des poids infiniment plus grands (10^7 secondes au lieu de 10^3 secondes environ

pour les autres), **noté abusivement** ∞ , i.e. $\infty \stackrel{\text{def}}{=} 10^7$. La seconde différence est relative aux arcs de transition. Les seuls arcs de transition conservés sont ceux orientés uniquement du graphe G_{TC} vers le graphe G_R . Ainsi, les arêtes TC et les arcs de transition sont favorisés par rapport aux arêtes routières et il est impossible initialement de rentrer dans le réseau de transports en commun.

L'algorithme est décrit dans la figure ci-dessous. En entrée, voici les données à disposition : une liste de chemins multimodaux L vide que l'on souhaite remplir, un nœud « source » n_s et un nœud « target » à atteindre n_t . En sortie, L peut contenir des chemins multimodaux ou non.



$$C(p) = C_1(p) \vee C_2(p) \vee C_3(p)$$

Figure 3.5 : Algorithme de recherche de chemins multimodaux

Voici le détail des conditions utilisées :

$C_1(p)$: le chemin p est un chemin routier uniquement

$C_2(p)$: le chemin p est déjà dans la liste L

$C_3(p)$: le chemin p est trop long par rapport au chemin $s \rightarrow t$ routier, i.e.

$$\text{durée}(p) > 2 \times \text{durée}(s \rightarrow t \text{ routier})$$

Soit un passager représenté par (n_o, n_d) sa paire OD de centroïdes appartenant à V_R .

Pour calculer des chemins multimodaux de type TC-routier, il faut appliquer l'algorithme avec $n_s = n_o$ et $n_t = n_d$. Pour calculer des chemins multimodaux de type Routier-TC, il suffit d'appliquer le même algorithme, mais avec $n_s = n_d$ et $n_t = n_o$.

Il est à noter que si aucun nœud TC n'est relié aux nœuds n_o et n_d , alors il n'existe pas de trajets multimodaux pour le passager.

3.1.3.2 Exemple

Illustrons l'algorithme sur un exemple : dans les figures de cette partie, le graphe G'_M est représenté à différents moments de l'algorithme. La légende de ces figures est présentée ci-dessous.

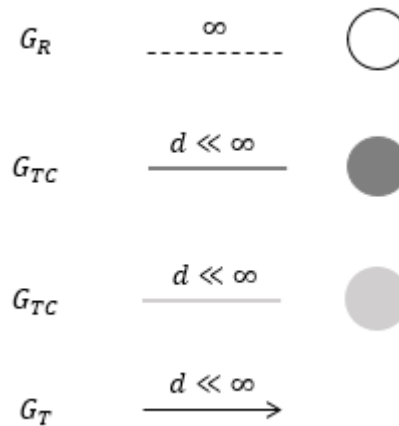


Figure 3.6 : Légende du graphe G'_M

Les nœuds creux noirs et les arêtes noires en pointillés représentent le graphe G_R . Dans ce graphe, il y a deux lignes de transports en commun, l'une en gris foncé (à gauche dans le graphe), l'autre en gris clair (à droite dans le graphe). Le graphe G_{TC} est donc représenté par les nœuds pleins et

arêtes en gris. Les arcs de transition sont distingués du reste des arêtes puisque ce sont les seuls orientés. Comme expliqué auparavant, les poids des arêtes routières sont infinis, tandis que les autres arcs et arêtes (de G_{TC} et G_T) ont des poids négligeables devant l'infini, mais ne sont pas tous égaux. L'origine et la destination du passager sont symbolisés par les nœuds O et D du graphe.

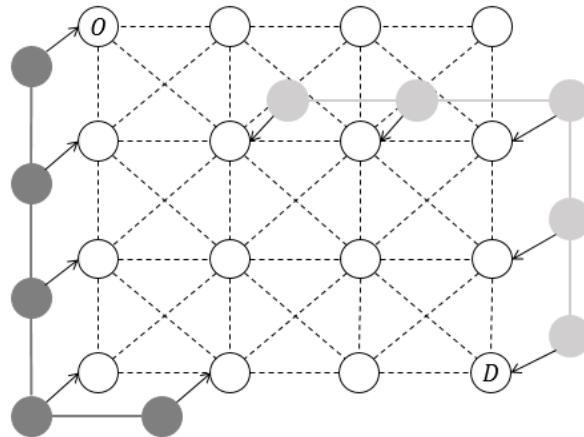


Figure 3.7 : G'_M dans son état initial

L'algorithme est appliqué avec $(n_s, n_t) = (n_o, n_D)$ dans un premier temps pour calculer les chemins multimodaux de type TC-Routier :

- Au début de l'algorithme le graphe G'_M est dans son état initial. Le nœud n_s , i.e. n_o , possède un nœud TC adjacent, appartenant à la ligne gris foncé de transports en commun. G'_M est donc modifié en ajoutant l'arc de transition qui part du nœud TC gris foncé adjacent vers n_o , comme sur la figure suivante (cf. encerclé) :

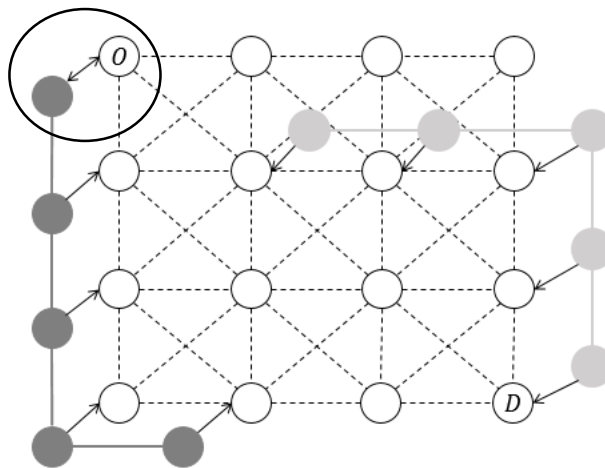


Figure 3.8 : TC-Routier – G'_M modifié

- L'algorithme de Dijkstra est ensuite appliqué et donne le plus court chemin de n_o à n_D dans G'_M . Le chemin p obtenu est l'un des deux chemins de poids similaire représentés en gras :

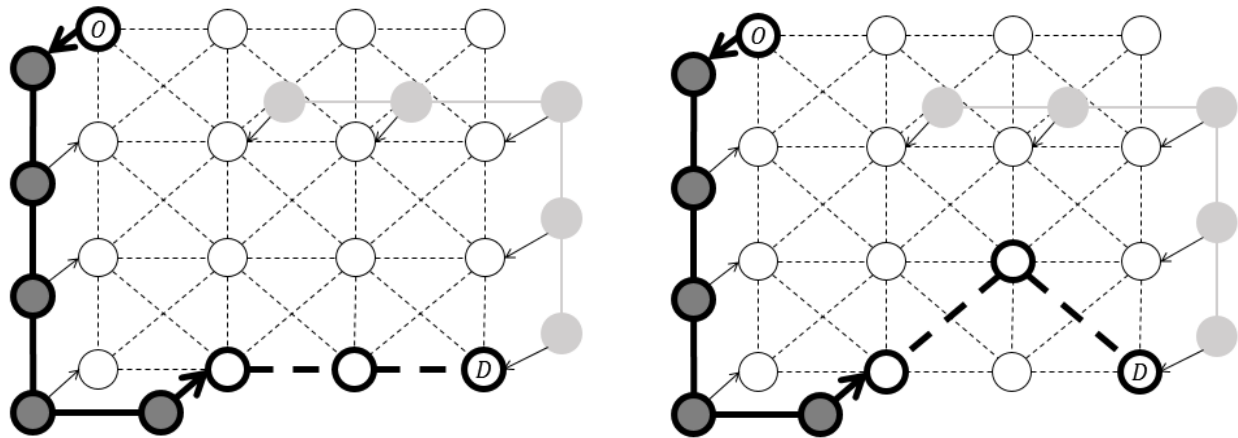


Figure 3.9 : TC-Routier – Plus court chemin p

- Pour simplifier l'exemple, $C(p) = C_1(p) \vee C_2(p)$. Le chemin p n'est pas uniquement routier et n'appartient pas à la liste L (qui est vide initialement), donc $C(p)$ est faux. Les points de jonctions pour le chemin p trouvé sont alors gardés en mémoire dans L : les premier et dernier nœuds de transports en commun et le premier nœud routier. Le graphe G'_M est ensuite modifié de la manière suivante :

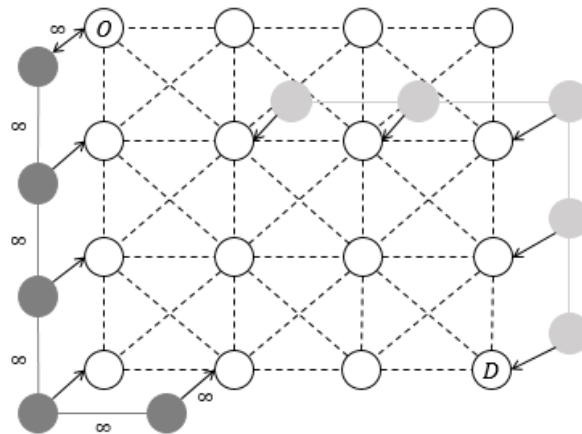


Figure 3.10 : TC-Routier – G'_M modifié après ajout de p dans L

- L'algorithme de Dijkstra est ensuite de nouveau appliqué et renvoie le plus court chemin entre n_o et n_D . Le chemin p renvoyé est uniquement routier, en effet les poids infinis s'additionnent

puisque $\infty \stackrel{\text{def}}{=} 10^7$, et il n'est plus avantageux de passer par la ligne de TC gris foncé ($3 \times \infty$ pour le chemin routier, contre $8 \times \infty$ pour le chemin TC-Routier précédent) :

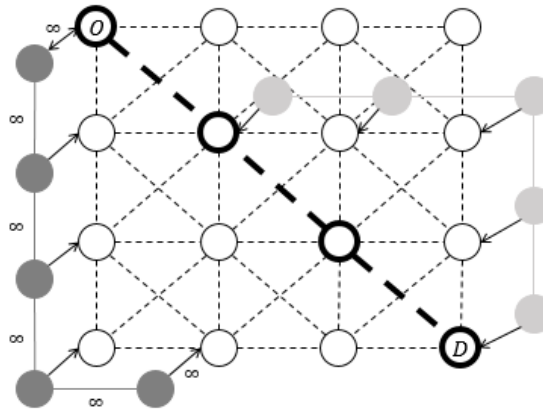


Figure 3.11 : TC-Routier – Chemin renvoyé p routier uniquement

- Le chemin p est uniquement routier, donc la condition $\mathcal{C}(p)$ est vraie. Le graphe G'_M est réinitialisé à son état initial, et c'est la fin de l'algorithme. Au final la liste L ne contient qu'un chemin TC-Routier.

A présent, l'algorithme est appliqué avec $(n_s, n_t) = (n_D, n_O)$ pour calculer les chemins multimodaux de type Routier-TC :

- Le graphe G'_M est modifié en ajoutant l'arc TC qui relie n_D au nœud TC de la ligne gris clair :

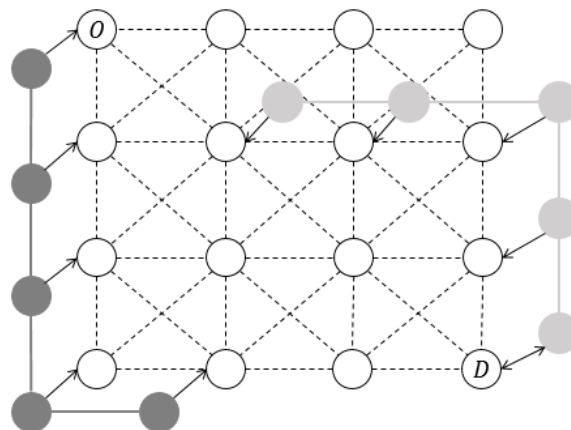


Figure 3.12 : Routier-TC – G'_M modifié

- En appliquant l'algorithme de Dijkstra de n_D vers n_O , le chemin TC-Routier p suivant est renvoyé :

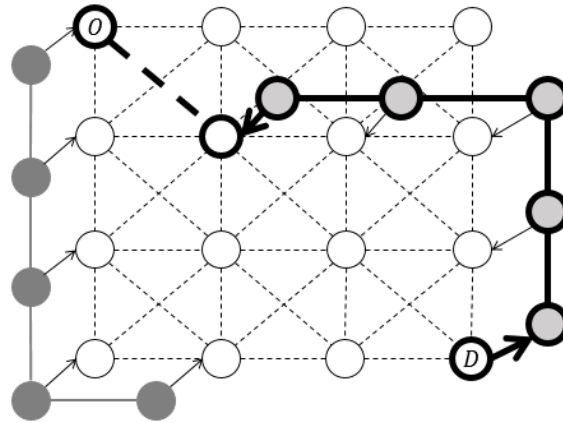


Figure 3.13 : Routier-TC – Plus court chemin p

- En inversant ce chemin p , un chemin Routier-TC allant de n_O à n_D est obtenu, noté \bar{p} :

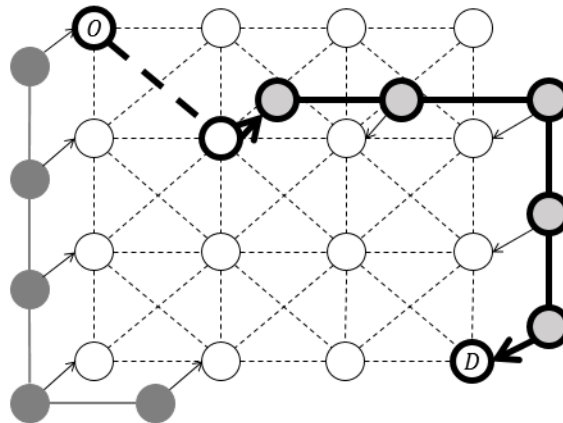


Figure 3.14 : Routier-TC – Plus court chemin \bar{p}

- Le chemin p vérifie $\mathcal{C}(p)$. Comme pour les chemins de type TC-Routier, ses points de jonctions sont sauvegardés dans L , à la différence près que le premier nœud TC de p correspond au dernier nœud TC de \bar{p} , le dernier nœud TC de p au premier nœud TC de \bar{p} et le premier nœud routier de p au dernier nœud routier de \bar{p} . L'algorithme se poursuit et le graphe G'_M est modifié avec les coûts infinis sur les arcs T et arêtes TC de p . L'algorithme de Dijkstra renvoie le chemin uniquement routier illustré sur la figure 3.15 G'_M est alors réinitialisé et l'algorithme est terminé.

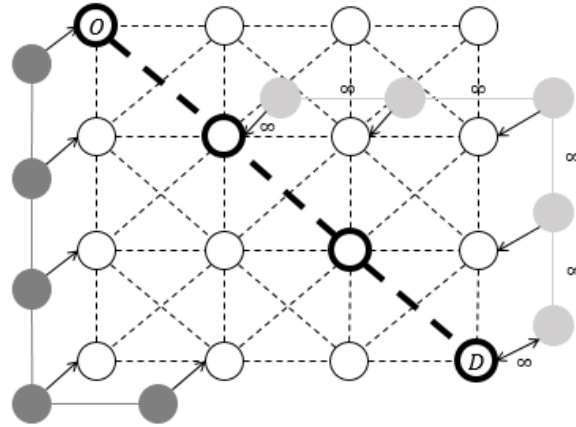


Figure 3.15 : Routier-TC – Chemin renvoyé p routier uniquement

3.2 Algorithme de jumelage

L'algorithme de ce chapitre s'appuie sur le graphe multimodal G_M et est ajustable en fonction de l'offre de conducteurs disponibles pour du covoiturage.

De manière simplifiée, un passager souhaite réaliser son trajet et une liste de conducteurs est disponible dans la base de données du service de covoiturage.

La première étape de l'algorithme consiste à réduire cette liste de conducteurs de manière à sélectionner les n meilleurs candidats potentiels pour le covoiturage classique et les m meilleurs candidats potentiels pour le covoiturage multimodal. C'est à ce niveau que les paramètres d'ajustement à l'offre et le graphe multimodal G_M sont utilisés.

La seconde étape consiste à sélectionner le meilleur jumelage possible pour le passager, classique ou multimodal. C'est l'option qui sera proposée à l'utilisateur. Il est également possible de choisir de proposer plusieurs solutions (une classique et une multimodale par exemple).

Voici le schéma qui représente l'algorithme de manière simplifiée.

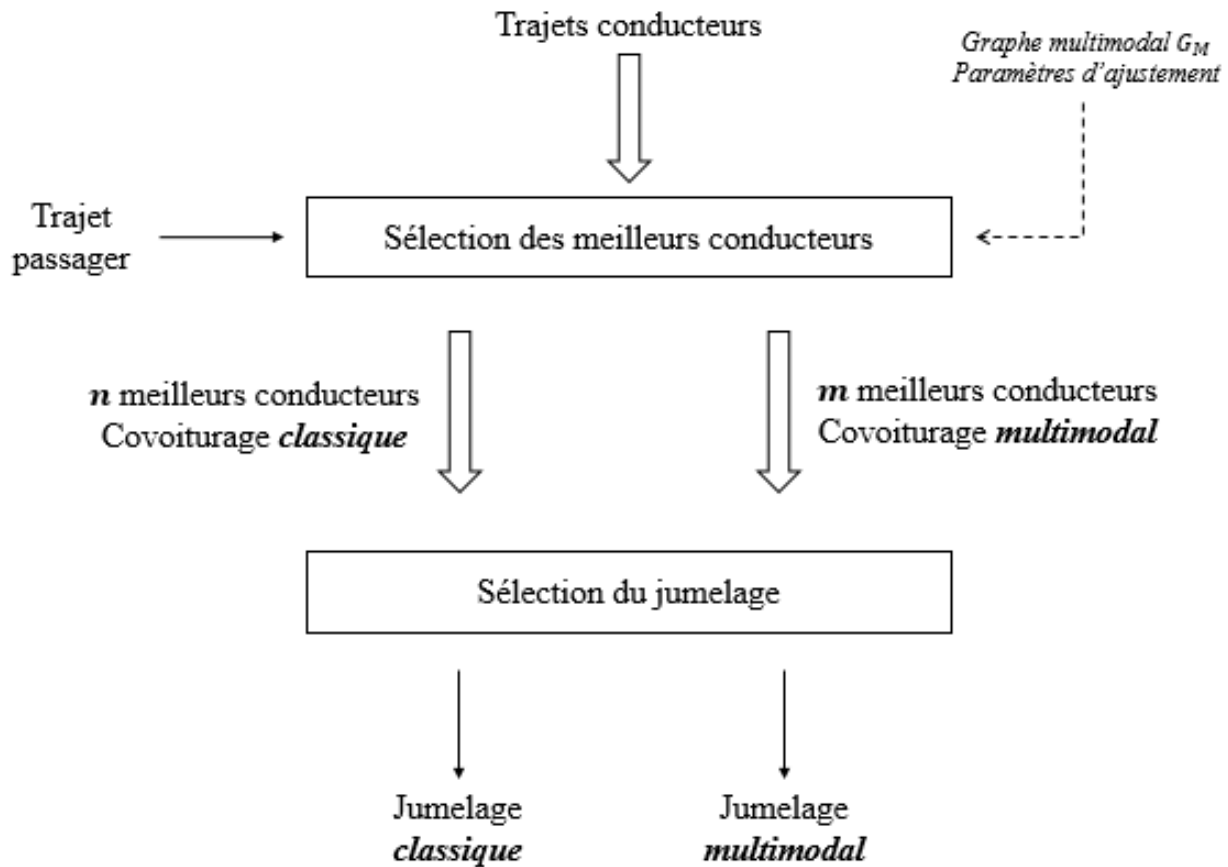


Figure 3.16 : Algorithme simplifié

Les prochaines parties décrivent en détail chaque étape de l'algorithme.

3.2.1 Sélection des meilleurs conducteurs

L'algorithme qui présente le détail de la sélection des meilleurs conducteurs est sur la figure 3.17 présentée ci-après.

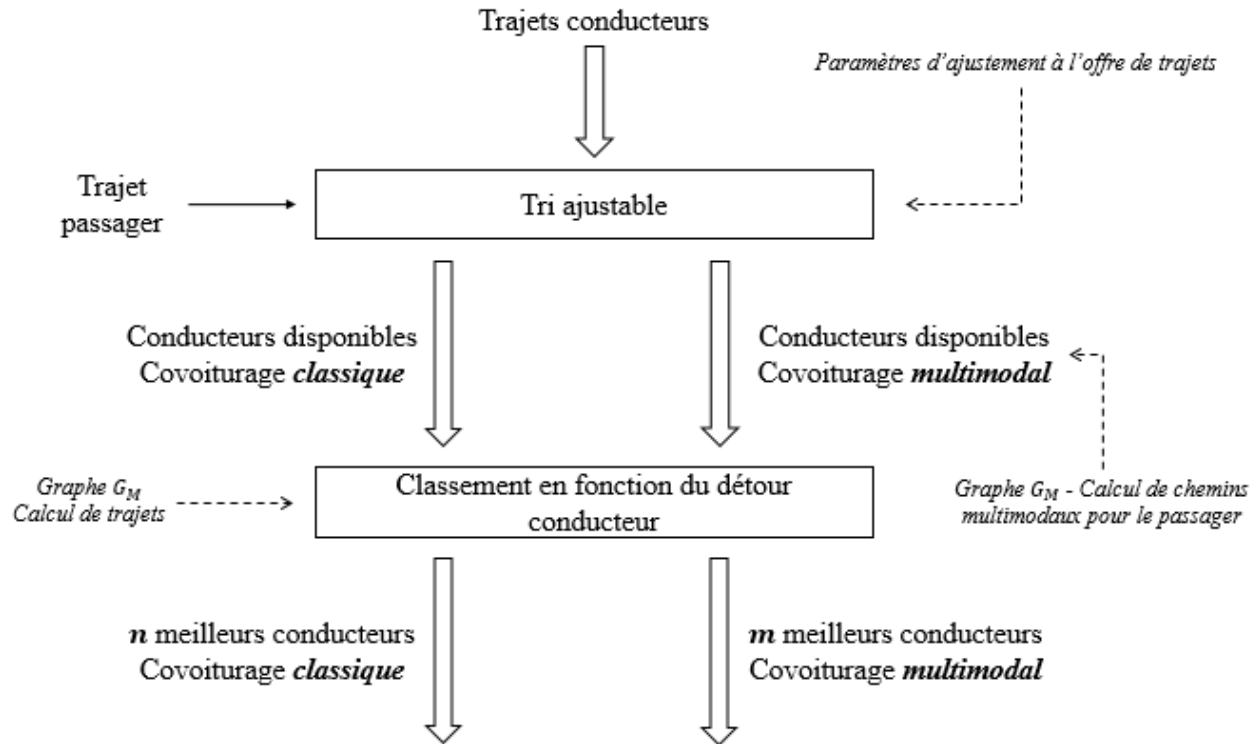


Figure 3.17 : Algorithme – Sélection des meilleurs conducteurs

3.2.1.1 Tri ajustable

Le tri ajustable permet d'identifier et d'éliminer les conducteurs qui ne peuvent pas covoiturer avec le passager en question. Grâce à des critères simples, la liste de conducteurs est réduite de manière rapide. Les conducteurs disponibles pour du covoiturage classique et ceux disponibles pour du covoiturage multimodal sont distingués.

3.2.1.1.1 Paramètres ajustables à l'offre et préférences utilisateur

Voici les paramètres indépendants des utilisateurs qui permettent d'ajuster l'algorithme à l'offre de covoitureurs :

- Seuil horaire t_{seuil} en minutes
- Angle limite pour le covoiturage classique $\theta_{classique}$ en degrés
- Angle limite pour le covoiturage multimodal $\theta_{multimodal}$ en degrés
- Vitesse moyenne v_{moy} en km/min

Et les préférences propres à chaque utilisateur, l'indice c correspond au conducteur et p au passager :

- Détour acceptable pour le conducteur d_c en km
- Tolérance horaire pour le conducteur $t_c = d_c/v_{moy}$ en minutes
- Tolérance horaire pour le passager t_p en minutes

Ces différents paramètres seront expliqués dans les parties qui suivent.

3.2.1.1.2 Conducteurs disponibles pour le covoiturage classique

Le trajet du passager est noté $O_p D_p$ et celui du conducteur est noté $O_c D_c$. Dans la suite, des temps d'arrivée à destination sont utilisés. S'il n'est pas précisé par l'utilisateur (passager ou conducteur), il peut être estimé par rapport à son temps de départ à l'origine en calculant un plus court chemin dans le graphe routier G_R . Pour qu'un conducteur soit disponible pour un passager, il doit répondre à plusieurs critères :

- Il doit se situer dans la même fenêtre de temps que le passager. En effet, un conducteur est éliminé si la différence entre les heures de départ du passager et du conducteur est supérieure au seuil horaire (qui correspond en principe au temps de parcours de la zone en question, ici Montréal, donc environ 120 min), i.e. si :

$$|t_{\text{départ conducteur}} - t_{\text{départ passager}}| > t_{\text{seuil}} \quad (1)$$

De plus, un conducteur est éliminé si son heure de départ minimum (t_2 dans la figure 3.18) est déjà supérieure à l'heure de départ maximum du passager (t_1 dans la figure 3.18) (en supposant qu'il y ait un trajet à faire pour aller chercher le passager chez lui), i.e. ceux qui vérifient la condition suivante sont éliminés :

$$t_{\text{départ passager}} + t_p \leq t_{\text{départ conducteur}} - t_c \quad (2)$$

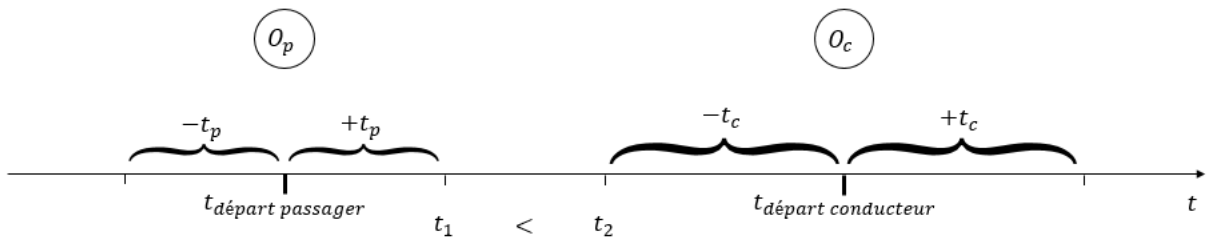


Figure 3.18 : Fenêtres de temps de départ - Exemple

Si l'heure d'arrivée maximum d'un conducteur (t_3 dans la figure 3.19) est plus petite que l'heure d'arrivée minimum du passager (t_4 dans la figure 3.19), il est également éliminé :

$$t_{\text{arrivée conducteur}} + t_c \leq t_{\text{arrivée passager}} - t_p \quad (3)$$

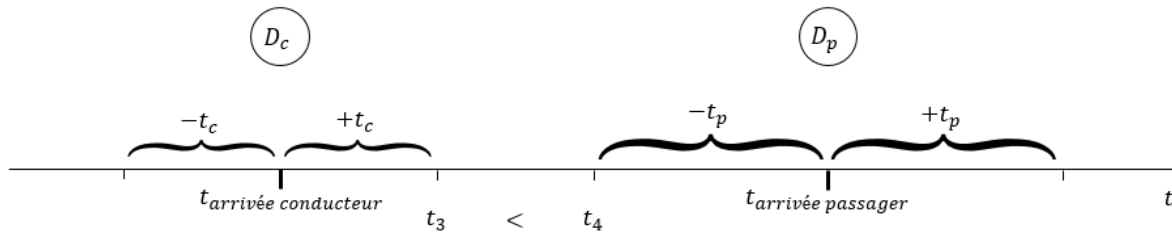


Figure 3.19 : Fenêtres de temps d'arrivée - Exemple

- Il doit avoir approximativement la même direction que le passager, i.e. l'angle α formé par les vecteurs $\overrightarrow{O_p D_p}$ et $\overrightarrow{O_c D_c}$ doit être inférieur à l'angle limite $\theta_{\text{classique}}$ pour le covoiturage classique, i.e. si la condition suivante est vérifiée, le conducteur est éliminé :

$$* (\overrightarrow{O_p D_p}, \overrightarrow{O_c D_c}) > \theta_{\text{classique}} \quad (4)$$

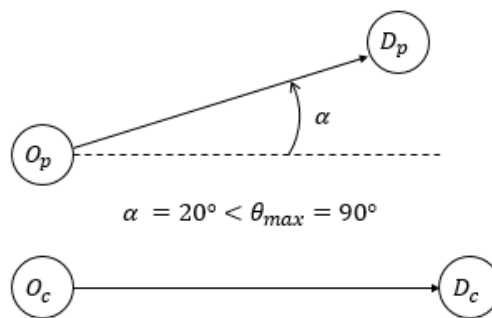


Figure 3.20: Direction des trajets – Exemple

Dans l'exemple ci-dessus, l'angle α est de 20° et $\theta_{\max} = 90^\circ$, donc la condition n'est pas vérifiée et le conducteur est conservé.

- Il doit avoir approximativement les mêmes zones d'origine et destination spatio-temporelles que le passager. Pour ce faire, à l'aide de v_{moy} (cf. 3.2.1.1.1), la différence entre les temps de départ – du passager et du conducteur – et la différence en distance à vol d'oiseau – entre l'origine du conducteur et l'origine du passager – est comparée. L'opération est également réalisée pour les temps d'arrivée et la distance à vol d'oiseau entre les destinations. Ainsi, un conducteur est éliminé si :

$$* \delta_{origines} + \delta_{destinations} > d_c \quad \vee \quad \delta_{origines} > v_{moy} * t_p \quad (5)$$

$$\delta_{origines} = \left| |O_c O_p| - v_{moy} * (t_{départ\ conducteur} - t_{départ\ passager}) \right|$$

$$\delta_{destinations} = \left| |D_c D_p| - v_{moy} * (t_{arrivée\ conducteur} - t_{arrivée\ passager}) \right|$$



Figure 3.21 : Zones spatio-temporelles – Exemple

Par exemple, dans la figure ci-dessus, $\delta_{origines} = 0$ et $\delta_{destinations} = 25$, avec $v_{moy} = 1km/min$. En considérant que $d_c = 30$, i.e. que le conducteur est prêt à faire un détour de 30km (aussi équivalent à 30min dans le cas où $v_{moy} = 1km/min$) et que $t_p = 10$, i.e. $v_{moy} * t_p = 10$, alors la condition (5) est fausse et le conducteur est conservé.

Pour chaque conducteur de la liste de conducteurs initiale, les critères sont utilisés dans l'ordre de description précédent, i.e. du moins chronophage en temps de calcul au plus chronophage. Si le conducteur vérifie ces critères successifs alors il est conservé pour intégrer la liste de covoitureurs disponibles pour le covoiturage classique.

Une exception est également gérée : elle concerne uniquement le critère (1) de direction des trajets. Si le trajet du passager est très petit et très proche de l'origine ou de la destination du conducteur, alors le conducteur intègre la liste, quel que soit l'angle entre les trajets i.e. si :

$$* |O_c O_p| + |O_p D_p| + |D_p O_c| < d_c \quad \vee \quad |D_c O_p| + |O_p D_p| + |D_p D_c| < d_c \quad (6)$$

Cette exception est illustrée dans la figure ci-dessous, où le conducteur n'est pas rejeté :

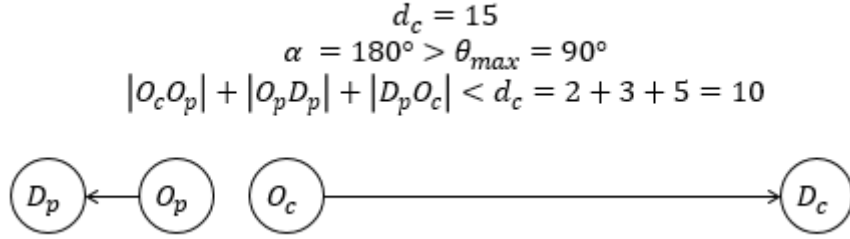


Figure 3.22 : Exception pour la direction des trajets - Exemple

3.2.1.1.3 Liste de conducteurs pour le covoiturage multimodal

Comme expliqué dans la partie 4.1.3 les chemins multimodaux sont obtenus pour le trajet passager $O_p D_p$. Dans le cas où il n'y a pas de chemins multimodaux, la liste de conducteurs pour le covoiturage multimodal est vide. Sinon, il faut distinguer les chemins multimodaux Routier-TC des chemins multimodaux TC-Routier.

- Pour les chemins multimodaux Routier-TC, les mêmes critères que pour le covoiturage classique sont employés, sauf que les trajets comparés sont différents : les trajets conducteurs $O_c D_c$ sont comparés au trajet $O_p O_{TC}$, où O_{TC} est le dernier nœud routier du chemin Routier-TC en question. De plus $t_{arrivée\ passager}$ devient $t_{arrivée\ passager\ à\ O_{TC}}$, qui est estimé grâce à un calcul de plus court chemin dans le graphe G_M . Ainsi, les critères (1) et (2) sont les mêmes tandis que les critères (3), (4), (5) et (6) changent :

$$* t_{arrivée\ conducteur} + t_c \leq t_{arrivée\ passager\ à\ O_{TC}} \quad (3_{Routier-TC})$$

$$* (\overrightarrow{O_p O_{TC}}, \overrightarrow{O_c D_c}) > \theta_{multimodal} \quad (4_{Routier-TC})$$

$$* \delta_{origines} + \delta_{destinations} > d_c \quad \vee \quad \delta_{origines} > v_{moy} * t_p \quad (5_{Routier-TC})$$

$$\delta_{origines} = \left| |O_c O_p| - v_{moy} * (t_{départ\ conducteur} - t_{départ\ passager}) \right|$$

$$\delta_{destinations} = \left| |D_c O_{TC}| - v_{moy} * (t_{conducteur} - t_{arrivée\ passager\ à\ O_{TC}}) \right|$$

$$* |O_c O_p| + |O_p O_{TC}| + |O_{TC} O_c| < d_c \quad \vee \quad |D_c O_p| + |O_p O_{TC}| + |O_{TC} D_c| < d_c \quad (6_{Routier-TC})$$

- Pour les chemins multimodaux TC-Routier, $O_c D_c$ est comparé au trajet $D_{TC} D_p$, où D_{TC} représente le premier nœud routier. Ici, $t_{départ\ passager}$ devient $t_{départ\ passager\ à\ D_{TC}}$. Les critères modifiés sont (1), (2), (4), (5) et (6), tandis que (3) reste inchangé :

$$* |t_{arrivée\ conducteur} - t_{arrivée\ passager}| > t_{seuil} \quad (1_{TC-Routier})$$

$$* t_{départ\ passager\ à\ D_{TC}} \leq t_{départ\ conducteur} - t_c \quad (2_{TC-Routier})$$

$$* (\overrightarrow{D_{TC} D_p}, \overrightarrow{O_c D_c}) > \theta_m \quad (4_{TC-Routier})$$

$$* \delta_{origines} + \delta_{destinations} > d_c \quad \vee \quad \delta_{origines} > v_{moy} * t_p \quad (5_{TC-Routier})$$

$$\delta_{origines} = \left| |O_c D_{TC}| - v_{moy} * (t_{départ\ conducteur} - t_{départ\ passager\ à\ D_{TC}}) \right|$$

$$\delta_{destinations} = \left| |D_c D_p| - v_{moy} * (t_{arrivée\ conducteur} - t_{arrivée\ passager}) \right|$$

$$* |O_c D_{TC}| + |D_{TC} D_p| + |D_p O_c| < d_c \quad \vee \quad |D_c D_{TC}| + |D_{TC} D_p| + |D_p D_c| < d_c \quad (6_{TC-Routier})$$

Comme un passager peut posséder plusieurs chemins multimodaux, lorsqu'un conducteur vérifie les conditions citées pour l'un des chemins multimodaux du passager, ce dernier est mis en mémoire et associé au conducteur en question, ceci pour chaque chemin multimodal qui fonctionne. Certains chemins multimodaux peuvent fonctionner pour un conducteur A mais ne pas fonctionner pour un conducteur B .

3.2.1.2 Classement en fonction du détour

La partie précédente décrit la façon d'obtenir la liste de conducteurs disponibles pour le covoiturage classique et la liste de conducteurs disponibles pour le covoiturage multimodal.

Une fois ces listes obtenues pour un passager donné, elles sont classées par ordre de détour du conducteur, en fonction du type de covoiturage :

$$\text{Détour covoiturage classique} = \frac{\text{temps du trajet } (O_c, O_p, D_p, D_c)}{\text{temps du trajet } (O_c, D_c)}$$

$$\text{Détour covoiturage multimodal Routier} - TC = \frac{\text{temps du trajet } (O_c, O_p, O_{TC}, D_c)}{\text{temps du trajet } (O_c, D_c)}$$

$$\text{Détour covoiturage TC} - \text{Routier} = \frac{\text{temps du trajet } (O_c, D_{TC}, D_p, D_c)}{\text{temps du trajet } (O_c, D_c)}$$

Ainsi, les n meilleurs conducteurs pour le covoiturage classique et les m meilleurs conducteurs pour le covoiturage multimodal peuvent être sélectionnés. L'idée est d'avoir une liste suffisamment grande pour pouvoir proposer des conducteurs au passager, mais suffisamment petite pour ne pas augmenter le temps de calcul passé dans la partie suivante de l'algorithme.

3.2.2 Sélection des jumelages

En effet, la phase de sélection des jumelages fait appel à un calculateur d'itinéraire précis de type Google Maps ou OpenTripPlanner. Les itinéraires de chacun des conducteurs obtenus dans les listes précédentes sont calculés. Il suffit juste de vérifier que le passager et le conducteur arrivent et partent à l'heure indiquée, en prenant en compte la tolérance de chacun en termes de temps perdu. Pour les itinéraires multimodaux du passager, les points de jonctions sont déjà connus, et des itinéraires en transports en commun nécessitent d'être calculés.

Pour proposer un ou plusieurs jumelages au passager, les listes classées de conducteurs obtenus à l'étape précédente – pour le covoiturage classique et le covoiturage multimodal – sont parcourues dans l'ordre et ceux pour lesquels les horaires ne sont pas respectés sont écartés. Dès que le nombre et le type de jumelages voulus est obtenu, l'algorithme s'arrête. Ainsi, le calculateur d'itinéraire est peu sollicité.

Il est donc important d'avoir des conducteurs de bonne qualité en début de liste. C'est ce qui est vérifié avec les tests dans la partie 4.3.

3.2.3 Processus de jumelage dynamique

L'algorithme est appliqué à chaque fois qu'un utilisateur offre ou demande un nouveau trajet. Il peut s'appliquer avec une liste de conducteurs de taille unitaire, donc il peut aussi bien être appliqué pour un passager et un conducteur à la fois.

Ainsi, lorsqu'un utilisateur souhaite covoiturer, deux cas sont distingués : s'il est passager, l'algorithme est lancé avec tous les conducteurs non jumelés, s'il est conducteur l'algorithme est lancé avec tous les passagers non jumelés.

Des notifications sont alors envoyées aux utilisateurs si des jumelages sont trouvés. Les passagers peuvent ensuite choisir les conducteurs en fonction de leur préférences (type de voiture, non-fumeurs, etc.) ou de leur réputation, au même titre que les conducteurs peuvent refuser certains passagers.

L'algorithme peut également être appliqué avec des critères plus souples si le taux d'acceptation n'est pas assez élevé et proposer des solutions moins avantageuses en termes de détour aux utilisateurs, de manière à constamment proposer des solutions aux utilisateurs.

Le schéma du processus de jumelage dynamique est présenté ci-dessous.

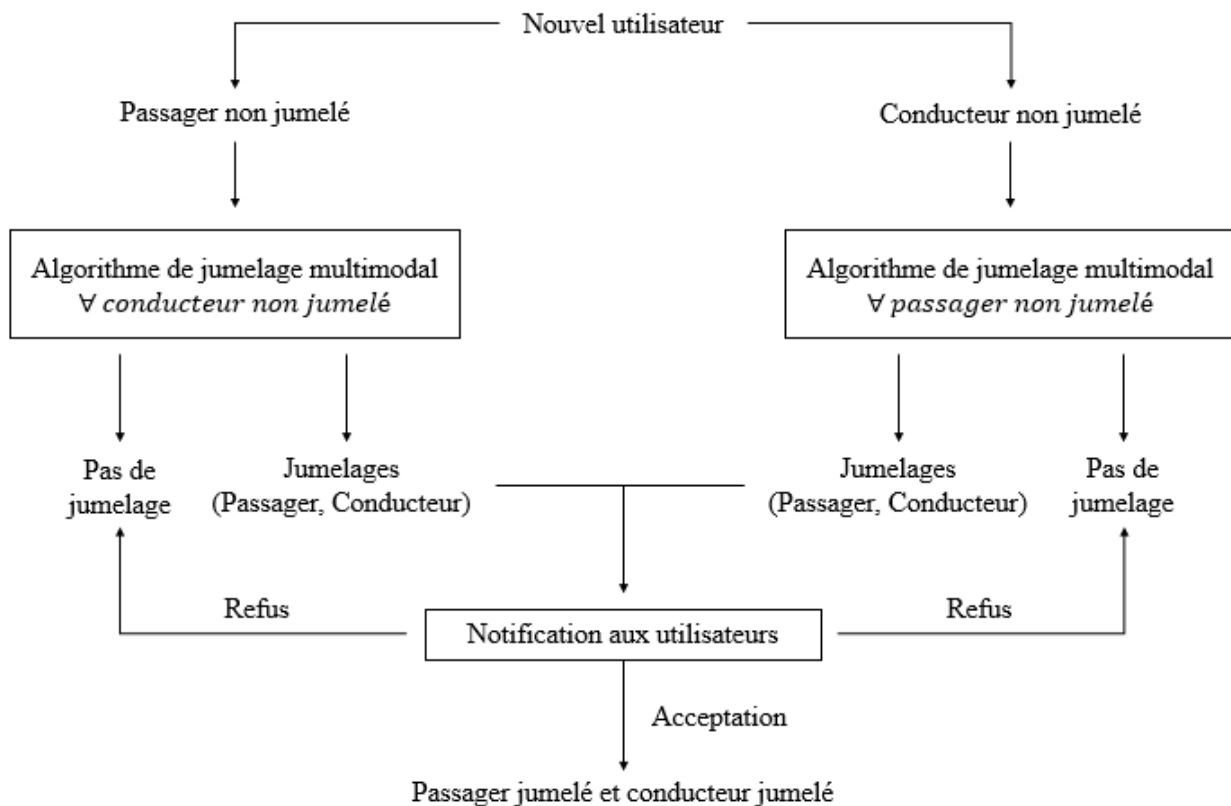


Figure 3.23 : Processus de jumelage dynamique

3.2.4 Avantages et principe de fonctionnement

Pour un passager donné, un calculateur d'itinéraire de type Google Maps permet de trier et de classer de manière précise et efficace les conducteurs disponibles pour du covoiturage. Cependant, trier une base de données de quelques milliers d'utilisateurs prend du temps (et de l'argent avec Google Maps). C'est l'intérêt du tri ajustable, qui permet de réduire considérablement et rapidement la taille de la base de données, grâce aux calculs préalables qui peuvent être réalisés avant que tout utilisateur ne fasse de demande.

Après avoir effectué ce premier tri, les conducteurs peuvent alors être classés grâce à G_M , et les calculs coûteux sont effectués seulement sur une infime portion de conducteurs de la base de données initiales. Seuls les premiers de chaque liste nécessitent une vérification précise via Google Maps, et dès qu'un de ces premiers conducteurs fonctionne il peut être proposé au passager, sans calcul supplémentaire.

3.3 Tests et résultats sur les données de l'enquête OD

Dans cette partie, les données de l'enquête OD sont utilisées. Il y a 23 998 trajets OD dont le nœud d'origine est différent du nœud de destination dans G_M . En reprenant la figure 3.16 représentant l'algorithme de manière simplifié, les trajets conducteurs entrants sont représentés par ces 23998 trajets OD, et **le trajet passager entrant correspond à l'un de ces conducteurs considéré comme passager**, de manière à lui trouver des covoitureurs. En d'autres mots, un des conducteurs de l'enquête OD est sélectionné pour que des jumelages lui soient trouvés. C'est pour cette raison qu'il est considéré comme « passager ».

Les tests sont réalisés avec un processeur Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz.

3.3.1 Validation du graphe multimodal

La représentation du réseau routier et du réseau de transports en commun par le graphe multimodal G_M étant approximative, il est nécessaire de s'assurer que les calculs de trajets effectués dans G_M soient relativement corrects par rapport à un calculateur d'itinéraire exact de type OpenTripPlanner ou Google Maps. La partie qui suit sur les temps de parcours s'inspire largement du travail réalisé par Thibaut Jaurou dans le cadre de son mémoire.

3.3.1.1 Temps de parcours des trajets de l'enquête OD

Dans un premier temps les temps de parcours calculés via le graphe sont comparés à ceux calculés via Google Maps. Ces calculs sont réalisés sur les trajets de l'enquête OD, pour les nœuds routiers (artériel et centroïde) et TC qui sont les plus proches de l'origine et de la destination de chacun des trajets sélectionnés. Chacune des courbes ci-dessous donne le temps de parcours (en ordonnée) du trajet OD en question (en abscisse). Les courbes en gris représentent les temps de parcours via le graphe G_M , tandis que les temps de parcours via Google Maps sont représentés par la courbe noire.

La figure 3.24 concerne les temps de parcours des trajets de l'enquête OD : pour un trajet OD, le nœud artériel le plus proche de son origine et le nœud artériel le plus proche de sa destination lui sont associés. Ainsi le temps de parcours entre ces deux nœuds artériels est calculé, via le graphe G_M et via Google Maps. Les trajets sont classés par ordre de temps de parcours croissant.

Les figures 3.25 et 3.26 représentent les temps de parcours des trajets de l'enquête OD, mais pour les nœuds centroïdes et TC les plus proches.

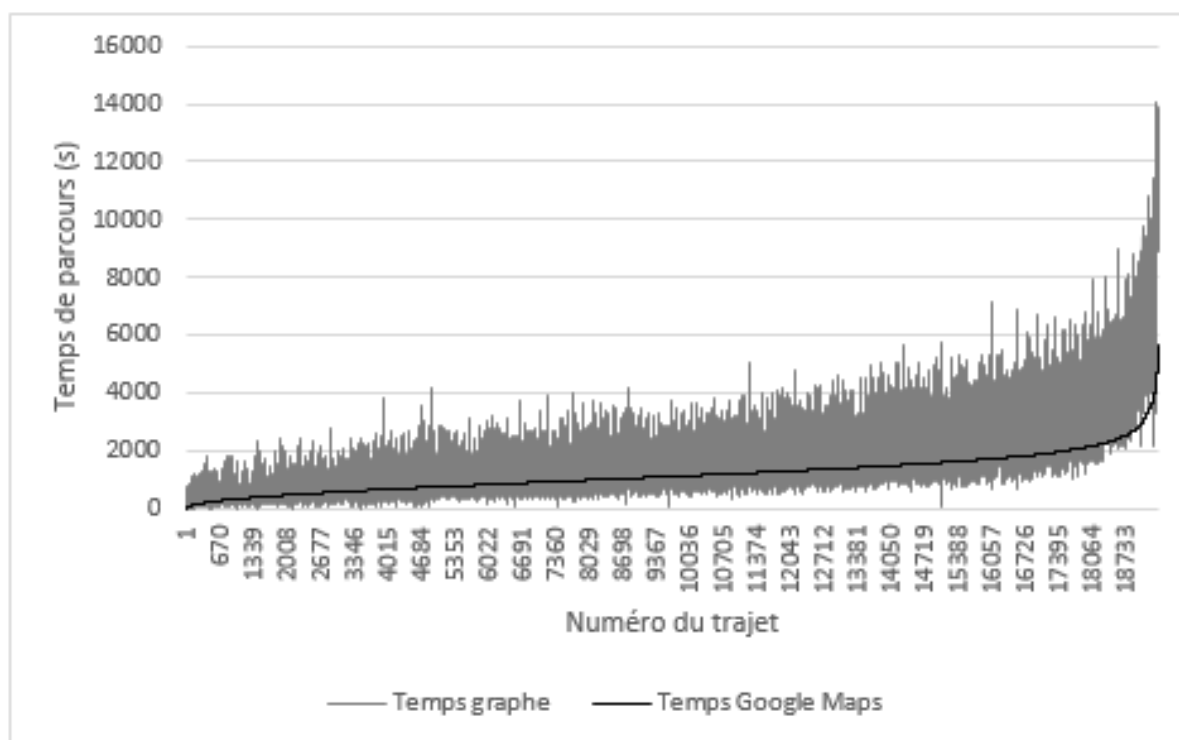


Figure 3.24 : Temps de parcours des trajets de l'enquête OD – Nœuds artériels

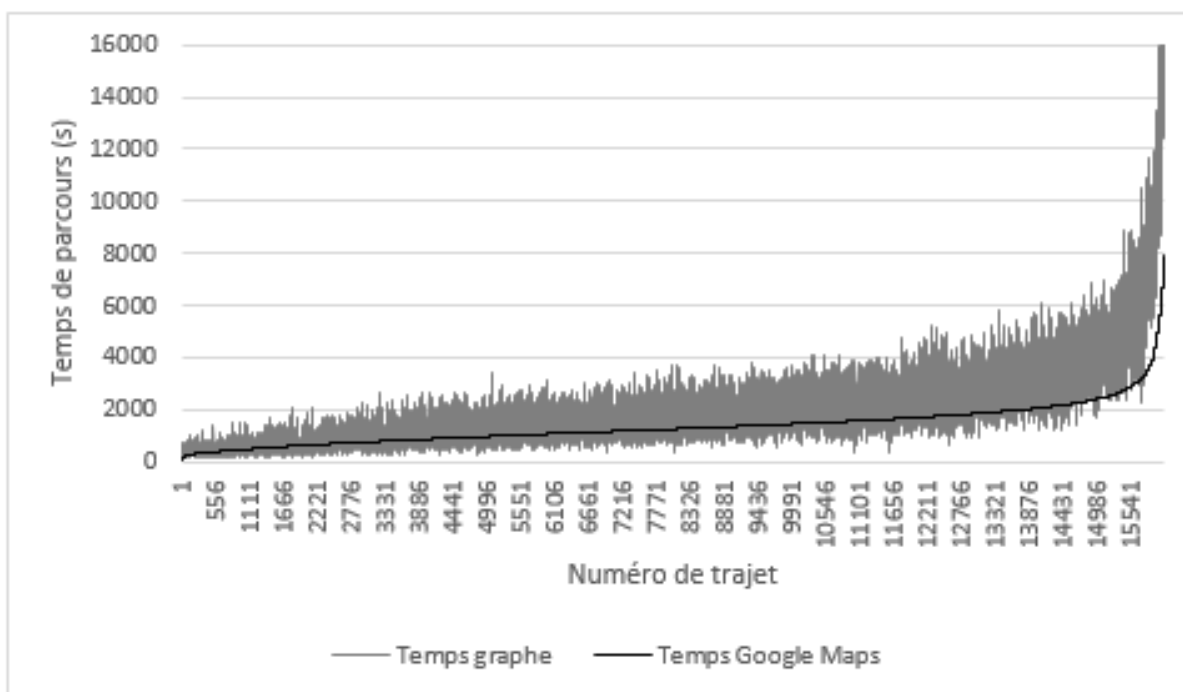


Figure 3.25 : Temps de parcours des trajets de l'enquête OD - Nœuds centroïdes

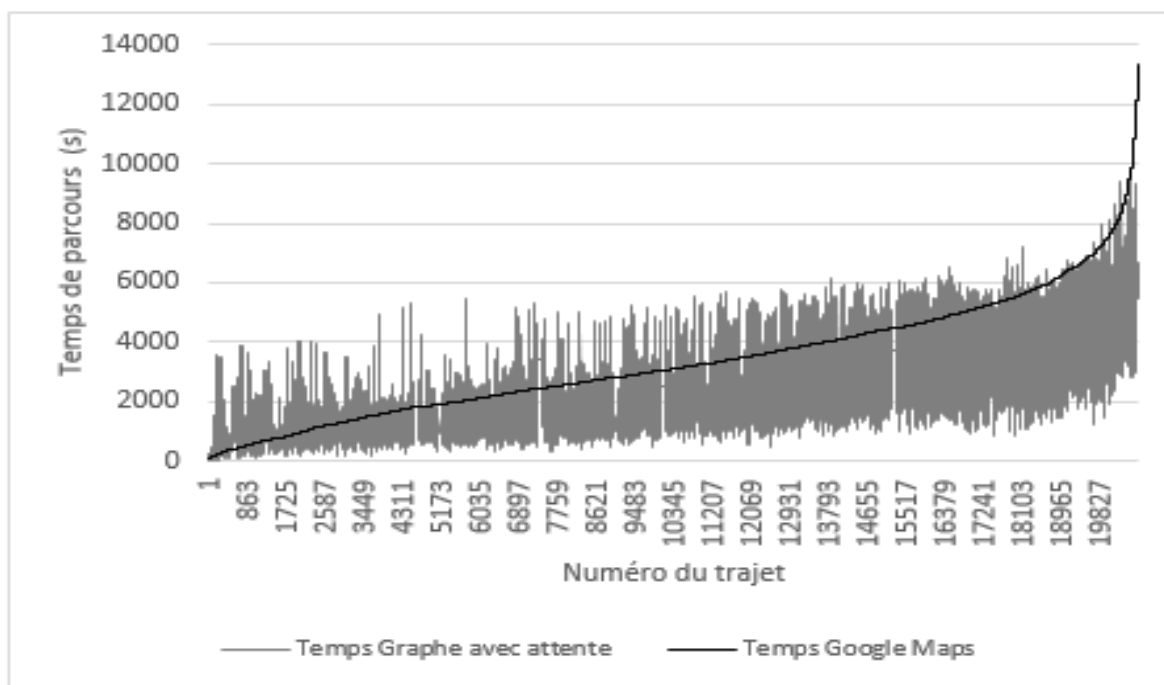


Figure 3.26 : Temps de parcours des trajets de l'enquête OD - Nœuds TC

Dans les figures relatives aux graphes routiers (figure 3.24 et 3.25), les deux courbes coïncident approximativement, bien que les temps de parcours calculés par le graphe soient surévalués. Le résultat est moins évident pour les temps de parcours du graphe TC (figure 3.26) : les temps de parcours via le graphe sont ici assez nettement sous-évalués.

Les tableaux présentés ci-après sont des statistiques descriptives des figures précédentes permettant de quantifier l'erreur commise par le calcul des temps de parcours via le graphe G_R .

Tableau 3.1 : Temps de parcours nœuds artériels - Statistiques descriptives

Comparaison entre T_G et T_{maps} (min)			
	Trajets	$T_G - T_{maps}$	T_G / T_{maps}
Moyenne	Ensemble	9.54	1.42
	< 30 minutes	6.28	1.31
	> 30 minutes	28.44	2.02
Mediane	Ensemble	5.43	1.31
	< 30 minutes	3.67	1.24
	> 30 minutes	23.57	1.65
Ecart-type	Ensemble	16.18	2.02
	< 30 minutes	11.26	2.18
	> 30 minutes	25.01	0.53

Tableau 3.2 : Temps de parcours nœuds centroïdes - Statistiques descriptives

Comparaison entre T_G et T_{maps} (min)			
	Trajets	$T_G - T_{maps}$	T_G / T_{maps}
Moyenne	Ensemble	10.78	1.36
	< 30 minutes	5.96	1.28
	> 30 minutes	28.00	1.64
Mediane	Ensemble	6.57	1.33
	< 30 minutes	4.00	1.24
	> 30 minutes	22.78	1.62
Ecart-type	Ensemble	17.00	0.50
	< 30 minutes	9.60	0.48
	> 30 minutes	24.77	0.44

Pour les nœuds artériels (Tableau 3.1), la différence entre les calculs des temps de parcours (G_M et Google Maps) est en moyenne égale à 10 min (9.54min). Ce résultat relativement bon est à nuancer par l'écart type élevé qui est de 16min (16.18min). L'écart à la moyenne ne suit donc pas une distribution homogène, ce qui se confirme par la valeur médiane, qui est de 5 min environ. En ce qui concerne le rapport entre les méthodes de calculs, c'est un rapport relativement bon en moyenne de 1.42, qui est lui aussi tronqué par l'écart-type élevé. Les trajets de moins de 30 minutes obtiennent de meilleurs résultats.

Pour les nœuds centroïdes, les résultats sont similaires.

3.3.1.2 Détours

Comme expliqué auparavant (cf. partie 4.2.1.2), il est essentiel d'avoir des conducteurs de bonne qualité au début des listes de conducteurs classés en fonction du détour via G_M . Pour un passager donné, il faut comparer les classements obtenus via G_M aux classements obtenus via un calculateur d'itinéraire précis, ici Google Maps.

Ainsi 100 passagers sont considérés. Le tri ajustable sur ces trajets donne entre 80 et 120 trajets parmi les trajets de l'enquête OD, pour le covoiturage classique, avec les paramètres suivants :

- Seuil horaire $t_{seuil} = 120min$
- Angle limite pour le covoiturage classique $\theta_{classique} = 90^\circ$
- Vitesse moyenne (en km par minute) $v_{moy} = 1km/min$
- Détour acceptable pour le conducteur $d_c = 10 km$
- Tolérance horaire pour le passager $t_p = 10 min$

Le premier critère observé concerne la position du premier conducteur de la liste obtenue via G_M dans la liste obtenue via Google Maps. Pour chacun de ces 100 trajets sont donnés la position moyenne, la valeur médiane et le nombre de fois où le conducteur est en première position dans les deux listes.

Le second critère observé est la moyenne des positions des trois premiers conducteurs de la liste obtenue via G_M dans la liste obtenue via Google Maps. La moyenne de ces positions et la valeur médiane sont recensées.

Les résultats figurent dans le tableau 3.3.

Tableau 3.3 : Détours - Critères 1 et 2

Critère	Moyenne	Médiane	Nombre de fois en première position
1. Position du premier	14 ^{ème}	7 ^{ème}	19%
2. Position moyenne des trois premiers	17 ^{ème}	15 ^{ème}	-

Le premier conducteur obtenu par le classement via G_M est donc en moyenne 14^{ème} dans le classement via Google Maps. Il est près d'une fois sur cinq le premier, et dans la moitié des cas il est dans une position plus faible inclusivement que la 7^{ème} place.

Les trois premiers conducteurs via G_M sont classés 17^{ème} en moyenne dans le classement via Google Maps,

En pratique, il est raisonnable de penser que la phase de sélection des jumelages n'ira pas au-delà des 10 premiers conducteurs de la liste calculée via G_M .

Ceci justifie l'observation du troisième et dernier critère (tableau 3.4), qui concerne le nombre de fois où le premier conducteur de la liste obtenue via G_M est dans les 10 premiers conducteurs obtenus via Google Maps. Cette valeur est également calculée pour les deuxième et troisième conducteurs.

Tableau 3.4 : Détours - Critère 3

Critère	Pourcentage
Premier via G_M dans les 10 premiers via Google Maps	64%
Deuxième via G_M dans les 10 premiers via Google Maps	44%
Troisième via G_M dans les 10 premiers via Google Maps	47%

Il y a donc de très fortes chances de tomber sur l'un des 10 premiers conducteurs du classement via Google Maps parmi les trois premiers classés via G_M , car il y a près d'une chance sur deux de tomber sur l'un des 10 premiers conducteurs du classement via Google Maps pour ces derniers.

En résumé, le classement obtenu via G_M est relativement bon. Il faut noter que la base de données des trajets de l'enquête OD utilisée est plus grande que la base de données de Netlift. Cette

dernière est en effet plus petite et il y a plus de disparité entre les différents trajets. Les différences entre les détours calculés via G_M sont alors plus importantes, et il est alors plus facile d'identifier les bons conducteurs des mauvais. Les résultats sur la base de données Netlift serait potentiellement de meilleure qualité.

3.3.1.3 Conclusion

Bien que les résultats sur les temps de parcours ne soient pas très convaincants, les résultats de meilleure qualité obtenus pour le classement des trajets en fonction du détour permettent l'utilisation du graphe multimodal G_M pour la validation de l'algorithme de jumelage multimodal. Celui-ci reste néanmoins perfectible. Un quadrillage plus précis (zones de 1km*1km par exemple) permettrait plus de précision.

3.3.2 Validation de l'algorithme de jumelage multimodal

3.3.2.1 Validation de l'algorithme de recherche de chemins multimodaux

Différents paramètres sont calculés de manière à illustrer les chemins obtenus à l'aide de l'algorithme de recherche de chemins multimodaux (cf. 3.1.3). La condition $C(p)$ est modifiée ici puisque les trajets de l'enquête OD recherchés ici sont ceux pouvant être remplacés – totalement ou en partie – par un tronçon en transports en commun. Ainsi :

$C(p)$: la taille de la liste L de chemins multimodaux est égale à 1

Pour chacun des trajets de l'enquête OD, ceci permet de calculer p_1 , le premier chemin renvoyé par l'algorithme de Dijkstra. Si celui-ci est uniquement composé d'arêtes du graphe G_R , c'est que le nombre d'arêtes routières de n'importe quel chemin multimodal est au moins égal au nombre d'arêtes routières de p_1 (puisque les poids des arêtes routières sont « infinis »). Si p_1 est uniquement composé d'arêtes du graphe G_{TC} et de deux arcs de transition, alors c'est que le trajet peut s'effectuer uniquement en transports en commun. Dans les deux cas précédents, le covoiturage multimodal n'est donc pas conseillé. Le dernier cas correspond au cas où p_1 est un trajet multimodal, qui n'est pas forcément plus rapide qu'un chemin uniquement routier, mais qui possède strictement moins d'arêtes routières que n'importe quel chemin routier correspondant au trajet OD en question.

Ce test permet également de distinguer les trajets de l'enquête OD qui ne sont pas connectés au réseau de transports en commun de l'île de Montréal et des trains de banlieue.

Dans les deux tableaux suivants, pour chacune des catégories, le pourcentage de trajets de l'enquête OD est présenté. Voici les résultats pour la recherche de chemins Routier-TC et TC-Routier.

Tableau 3.5 : Algorithme de recherche de chemin Routier-TC - Résultats

Catégorie	Pourcentage
Pas de connexion de la destination du trajet OD au réseau TC	54%
Trajet uniquement routier	3%
Trajet uniquement TC	21%
Trajet multimodal Routier-TC	22%

Tableau 3.6 : Algorithme de recherche de chemin TC-Routier - Résultats

Catégorie	Pourcentage
Pas de connexion de l'origine du trajet OD au réseau TC	66%
Trajet uniquement routier	2%
Trajet uniquement TC	21%
Trajet multimodal TC-Routier	11%

Il est à noter dans cette partie que les trajets de l'enquête OD considérés sont ceux de la **période de pointe du matin**, et que les conclusions suivantes ne sont valables que pour **les trajets du matin**. Logiquement, l'algorithme de recherche de chemins multimodaux trouve deux fois plus de trajets Routier-TC que de trajets TC-Routier (22% contre 11%). En effet, les zones de destination des conducteurs (en auto solo durant la période de pointe du matin) sont majoritairement situées sur l'île de Montréal et sont reliées au réseau de transport en commun de l'île ou des trains de banlieue. Ceci est également confirmé par la proportion des trajets connectés au réseau de transport en commun, puisque les destinations sont plus fortement reliées que les origines. Les résultats obtenus sont donc cohérents.

Voici quelques exemples de trajets multimodaux obtenus (pas de notion de covoiturage à ce stade).
Les arêtes routières sont en bleu, les arcs de transition en rouge et les arêtes TC en vert.



Figure 3.27 : Exemple de trajet Routier-TC 1



Figure 3.28 : Exemple de trajet Routier-TC 2

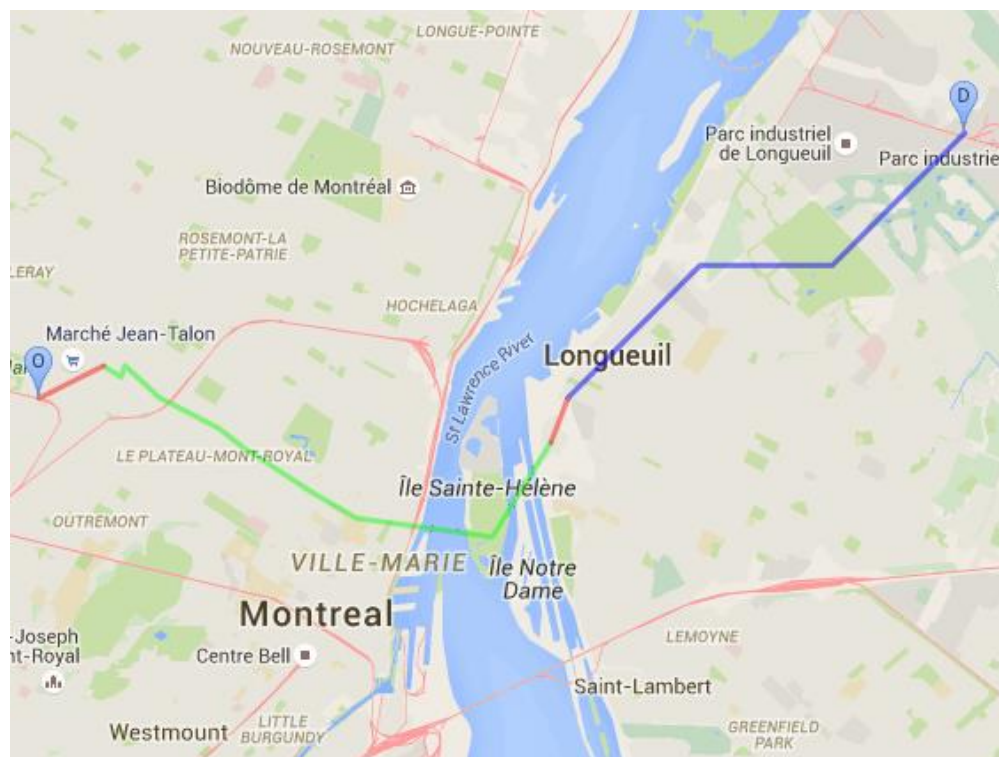


Figure 3.29 : Exemple de trajet TC-Routier 1

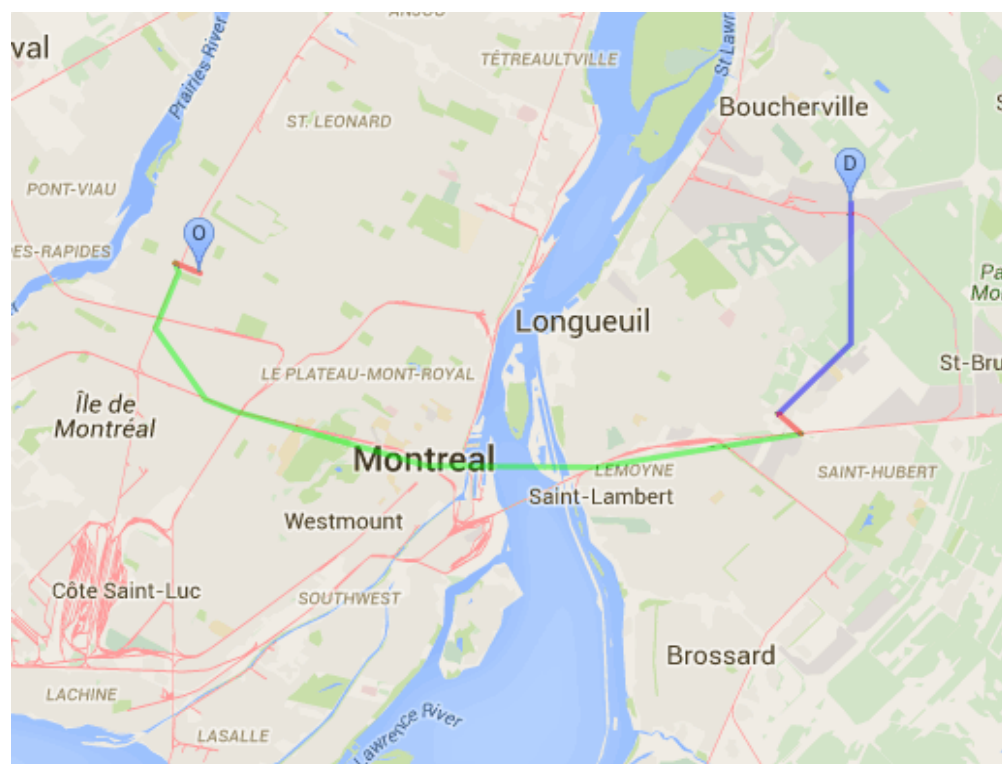


Figure 3.30 : Exemple de trajet TC-Routier 2

3.3.2.2 Validation du tri ajustable

Voici les paramètres utilisés pour les tests de cette partie :

- Seuil horaire $t_{seuil} = 120min$
- Angle limite pour le covoiturage classique $\theta_{classique} = 90^\circ$
- Angle limite pour le covoiturage multimodal $\theta_{multimodal} = 135^\circ$
- Vitesse moyenne (en km par minute) $v_{moy} = 1km/min$

3.3.2.2.1 Covoiturage classique

Le tri ajustable a été effectué pour chaque passager de l'enquête OD pour le covoiturage classique et pour différents paramètres. Comme dans la partie précédente, lorsqu'un pourcentage est indiqué, il correspond à une proportion de trajets de l'enquête OD.

Pour le covoiturage classique, quatre tests différents sont réalisés pour illustrer le tri effectué. Les seuls paramètres qui varient sont les détours acceptables du conducteur d_c et du passager t_p .

Tableau 3.7 : Tri ajustable pour le covoiturage classique – Tests

Lecture du tableau : par exemple, pour les paramètres $d_c = 20 km$ et $t_p = 20 min$, 99% des passagers de l'enquête OD possèdent au moins un conducteur disponible. Il y en a en moyenne 459.

Test	Paramètres (d_c, t_p)	Passagers avec conducteurs disponibles	Nombre de conducteurs disponibles moyen
1	(20,20)	99%	459
2	(10,10)	98%	95
3	(5,5)	92%	22
4	(1,1)	38%	2

Lorsqu'il n'y a pas de conducteurs disponibles, le tri ajustable est trop fort, i.e. pour chacun des passagers considérés, aucun des conducteurs ne valide les conditions du tri ajustable.

Logiquement, plus les paramètres d_c et t_p sont petits, moins le nombre de passagers avec conducteurs disponibles est grand, et plus le nombre de conducteurs disponibles en moyenne est faible.

Pour les 23 998 passagers et avec les paramètres $d_c = 10 \text{ km}$ et $t_p = 10 \text{ min}$, le tri ajustable prend en moyenne 151ms. Ceci pour passer d'une liste d'environ 24000 conducteurs à une liste réduite d'une centaine de conducteurs classés par ordre de détour croissant. La majeure partie du temps d'exécution est prise par le calcul des plus courts chemins pour calculer les détours et ainsi classer les conducteurs disponibles.

3.3.2.2.2 Covoiturage multimodal

Pour le covoiturage multimodal, le tri ajustable a été effectué pour les passagers dont le trajet peut comporter un tronçon en transports en commun. Ici, seuls sont considérés les 22% de trajets dont l'algorithme de recherche de chemins multimodaux renvoie un chemin Routier-TC en premier, et les 11% de trajets pour les chemins TC-Routier. Ceci correspond respectivement à 5324 trajet pour le covoiturage Routier-TC et 2674 trajets pour le covoiturage TC-Routier.

Les paramètres d_c et t_p sont toujours variables. Les résultats obtenus sont dans les deux tableaux qui suivent.

Tableau 3.8 : Tri ajustable pour le covoiturage Routier-TC - Tests

Test	Paramètres (d_c, t_p)	Passagers avec conducteurs disponibles	Nombre de conducteurs disponibles moyen
1	(10,10)	99,5%	171
2	(5,5)	97%	38
3	(1,1)	57%	2

Tableau 3.9 : Tri ajustable pour le covoiturage TC-Routier - Tests

Test	Paramètres (d_c, t_p)	Passagers avec conducteurs disponibles	Nombre de conducteurs disponibles moyen
1	(10,10)	98,3%	239
2	(5,5)	97%	61
3	(1,1)	75%	3

Le temps d'exécution pour le tri ajustable multimodal est similaire au temps d'exécution pour le tri ajustable classique.

3.3.2.3 Jumelages obtenus

Afin d'évaluer l'efficacité de l'algorithme, il est nécessaire d'évaluer les jumelages obtenus par l'algorithme. Les paramètres communs aux tests de cette partie sont les suivants :

- Seuil horaire $t_{seuil} = 120min$
- Vitesse moyenne (en km par minute) $v_{moy} = 1km/min$

3.3.2.3.1 Covoiturage classique

La série de tests suivante s'appuie sur une liste de 1000 passagers sélectionnés parmi les 23 998 passagers de l'enquête OD. Chacun des passagers de cette liste possède au moins trois conducteurs disponibles après le tri ajustable, et ce pour différents paramètres de tri ajustable :

- $P_1 : \theta_{classique} = 90^\circ, d_c = 10 km$ et $t_p = 10 min$.
- $P_2 : \theta_{classique} = 45^\circ, d_c = 5 km$ et $t_p = 5 min$.
- $P_3 : \theta_{classique} = 90^\circ, d_c = 1 km$ et $t_p = 1 min$.

Pour chaque ensemble de paramètres P_i un pourcentage est recensé. C'est le pourcentage de passagers dont le premier jumelage classé via le tri ajustable respecte les horaires du passager et du conducteur au regard des paramètres $(d_c, t_p) \in \{(10,10); (15,15); (20,20)\}$.

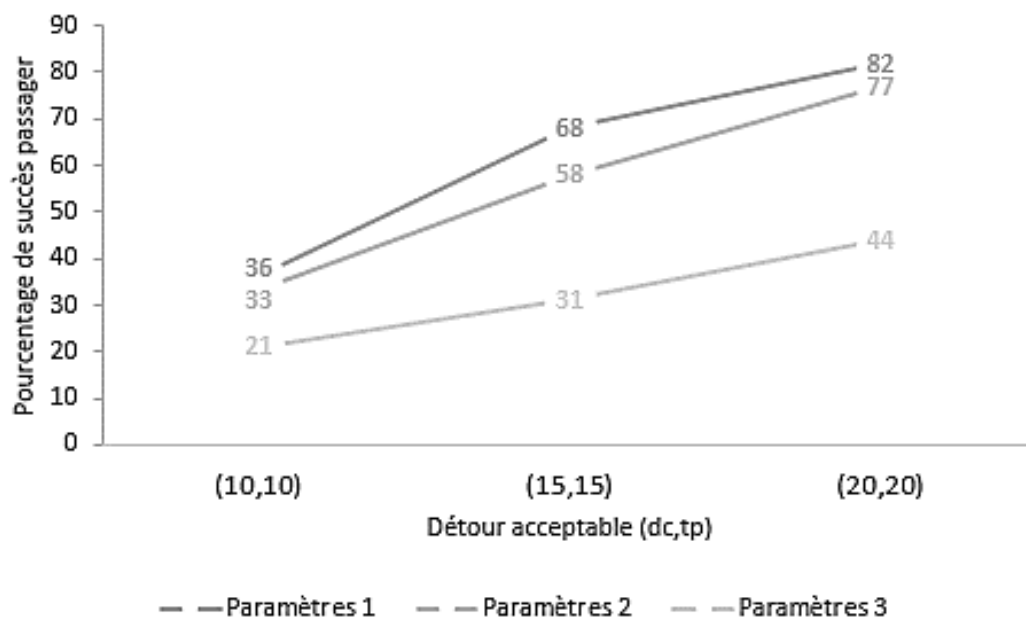


Figure 3.31 : Covoiturage classique – Pourcentage de passagers dont le 1er jumelage satisfait les horaires

Lecture du graphique : par exemple, 68% des 1000 passagers sélectionnés de l'enquête OD disposent d'un premier jumelage satisfaisant les horaires du passager et du conducteur pour un détour conducteur de 15 km et une tolérance passager de 15 min, pour les paramètres P_1

Le second test consiste à observer si parmi les trois premiers conducteurs classés, au moins l'un d'entre eux satisfait les horaires au regard des mêmes paramètres (d_c, t_p) que dans le test précédent. Voici les résultats :

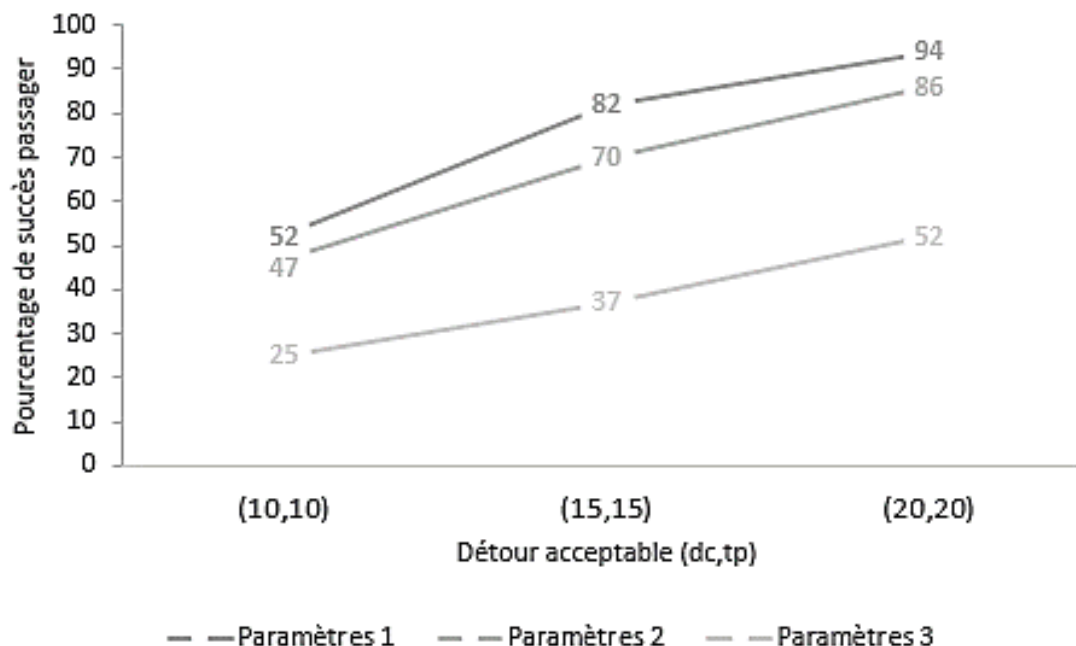


Figure 3.32 : Covoiturage classique – Pourcentage de passagers dont au moins un des trois premiers jumelages satisfait les horaires

Lecture du graphique : par exemple, 94% des 1000 passagers de l'enquête OD sélectionnés disposent d'un jumelage parmi les trois premiers classés satisfaisant les horaires du passager et du conducteur pour un détour conducteur de 20 km et une tolérance passager de 20min pour les paramètres P_1

Les meilleurs résultats sont obtenus pour les paramètres P_1 où plus d'un passager sur deux (52%) dispose d'un conducteur respectant les horaires parmi leurs trois premiers jumelages (pour $d_c = 10 \text{ km}$ et $t_p = 10 \text{ min}$). Il faut noter qu'en cherchant un bon conducteur parmi les 10 premiers conducteurs, les pourcentages seraient plus élevés.

3.3.2.3.2 Exemples de jumelages obtenus pour le covoiturage classique

Dans les figures ci-dessous, les conducteurs sont représentés en rouge et les passagers en jaune.



Départ habituel du passager et du conducteur : 7h



Départ habituel du conducteur : 7h

Départ habituel du passager : 7h15

Figure 3.33 : Exemple 1 de jumelage classique

Figure 3.34 : Exemple 2 de jumelage classique

Les jumelages représentés satisfont les horaires pour $d_c = 10$ et $t_p = 10$, et sont ceux qui sont classés en premier dans la liste de jumelages pour les passagers en question.

Le conducteur et le passager du premier exemple partent habituellement à la même heure 7h20. Dans le deuxième exemple le conducteur part habituellement à 7h, tandis que le passager part à 7h15, ce qui explique la raison pour laquelle le jumelage est correct.

3.3.2.3.3 Covoiturage multimodal

Les jumelages pour le covoiturage multimodal de type Routier-TC obtenus par l'algorithme sont ici analysés. La série de tests suivante s'appuie sur une liste de 500 passagers sélectionnés parmi les 23 998 passagers de l'enquête OD. Chacun des passagers de cette liste possède au moins trois conducteurs disponibles après le tri ajustable, et ce pour différents paramètres de tri ajustable. Les meilleurs résultats ont été obtenus pour l'ensemble de paramètres P : $\theta_{multimodal} = 135^\circ$, $d_c = 10 \text{ km}$ et $t_p = 10 \text{ min}$.

Comme pour le covoiturage classique, les trois premiers conducteurs classés sont observés, pour différents paramètres (d_c, t_p) . Voici les résultats :

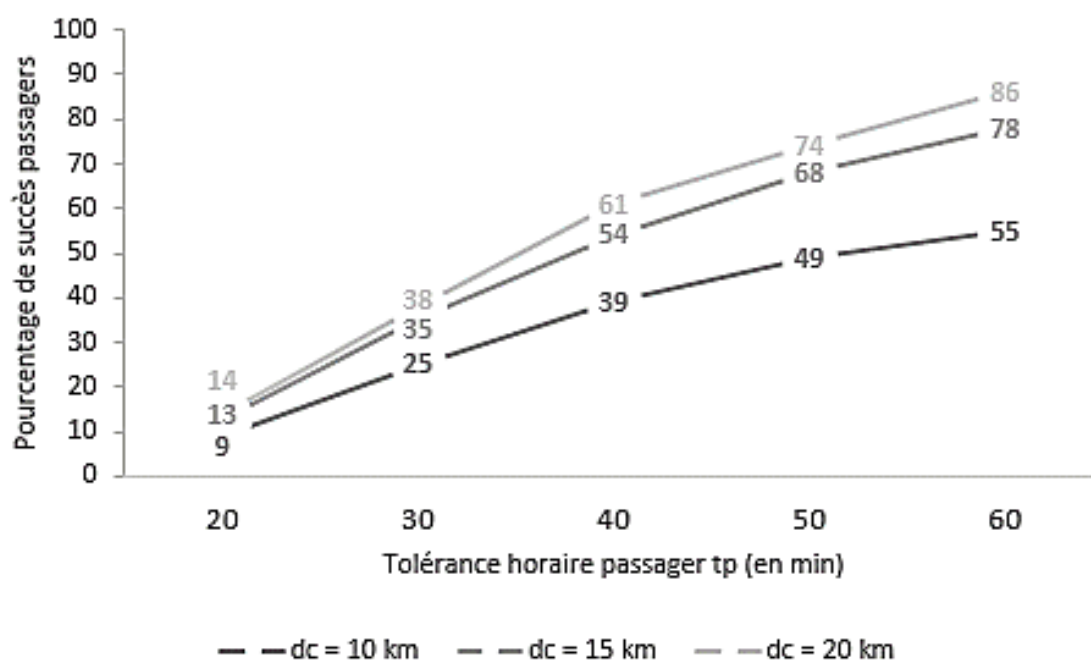


Figure 3.35 : Covoiturage multimodal Routier-TC – Pourcentage de passager dont au moins un des trois 1ers jumelages satisfait les horaires

Lecture du graphique : par exemple, 86% des 500 passagers de l'enquête OD sélectionnés disposent d'un jumelage parmi les trois premiers classés satisfaisant les horaires du passager et

du conducteur pour un détour conducteur de 20 km et une tolérance passager de 60min pour les paramètres P

La satisfaction de la tolérance horaire t_p du passager est plus limitante que le détour acceptable du conducteur d_c . En effet, l'augmentation des pourcentages est beaucoup plus marquée en faisant varier la tolérance horaire passager (de gauche à droite sur la figure 3.35) qu'en faisant varier le détour acceptable du conducteur (en changeant de courbe de bas en haut).

Ceci s'explique par le fait que le trajet multimodal du passager génère plus d'aléas qu'un trajet en voiture. De plus, les temps de parcours en transports en commun sont nettement plus élevés qu'en auto, ce qui est à prendre en compte quant à l'analyse des résultats. Les temps de parcours calculés ne prennent pas en compte la congestion, donc le détour du passager pourrait être inférieur à celui qui est affiché.

Ce sont de bons résultats malgré tout pour le covoiturage multimodal de type Routier-TC.

Pour ce qui est du covoiturage multimodal de type TC-Routier, les détours calculés pour les jumelages trouvés sont plus importants que les détours pour le covoiturage multimodal de type Routier-TC. Avec différents paramètres, les tests ne permettent pas d'arriver à des résultats concluants. Ceci pourrait prouver que le covoiturage multimodal de type TC-Routier est une solution de covoiturage qui nécessite beaucoup plus de précision dans les modèles déployés, et qui n'est viable uniquement dans des cas particuliers, **pour la période de pointe du matin**. En effet, si le covoiturage de type Routier-TC semble viable le matin pour aller au travail, il semble logique que le covoiturage de type TC-Routier soit une solution pour rentrer chez soi. Ceci nécessiterait de nouveaux tests sur des trajets pendant la période de pointe du soir.

3.3.2.3.4 Exemples de jumelages obtenus pour le covoiturage multimodal

Comme dans les parties précédentes, les arêtes routières sont en bleu, les arcs de transition en rouge et les arêtes TC en vert. Le conducteur est représenté en rouge et le passager en jaune.

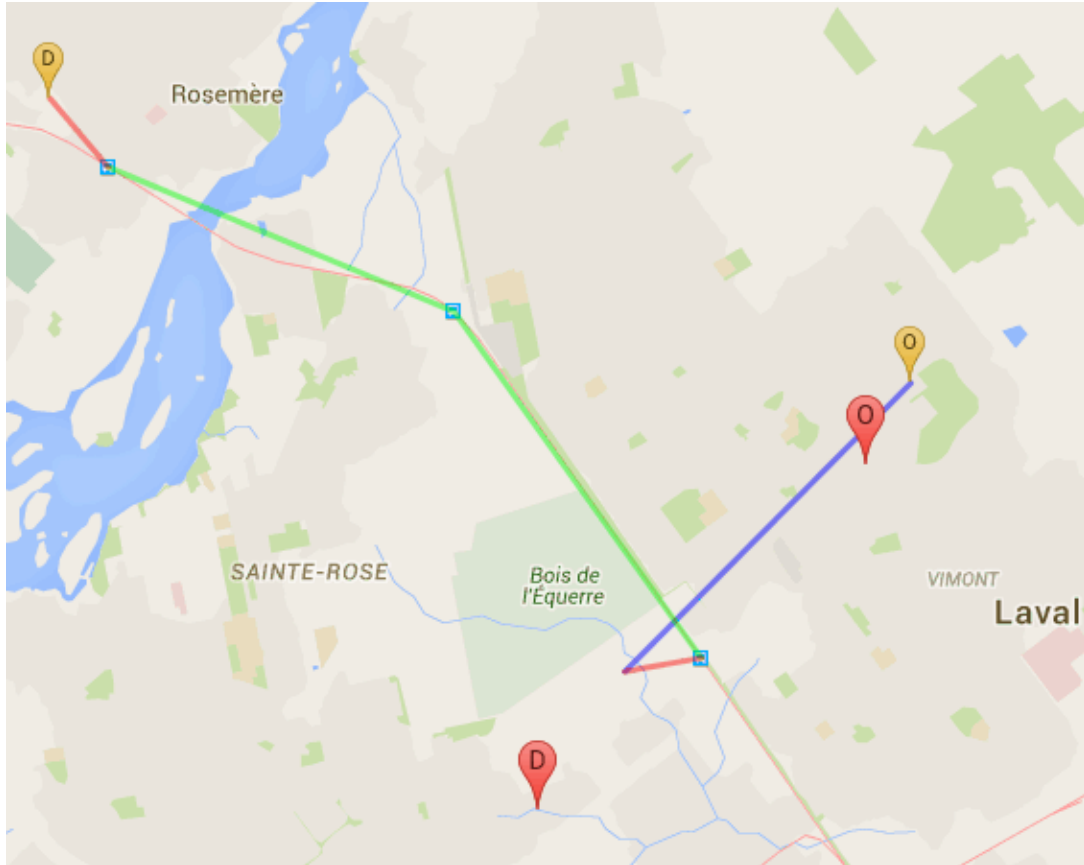


Figure 3.36 : Exemple 1 de jumelage multimodal Routier-TC

Les jumelages représentés satisfont les horaires pour $d_c = 10 \text{ km}$ et $t_p = 20 \text{ min}$, et sont ceux qui sont classés en premier dans la liste de jumelages pour les passagers en question.

Le conducteur et le passager du premier exemple partent habituellement à la même heure 7h30, tandis que ceux du deuxième exemple partent à 7h45.

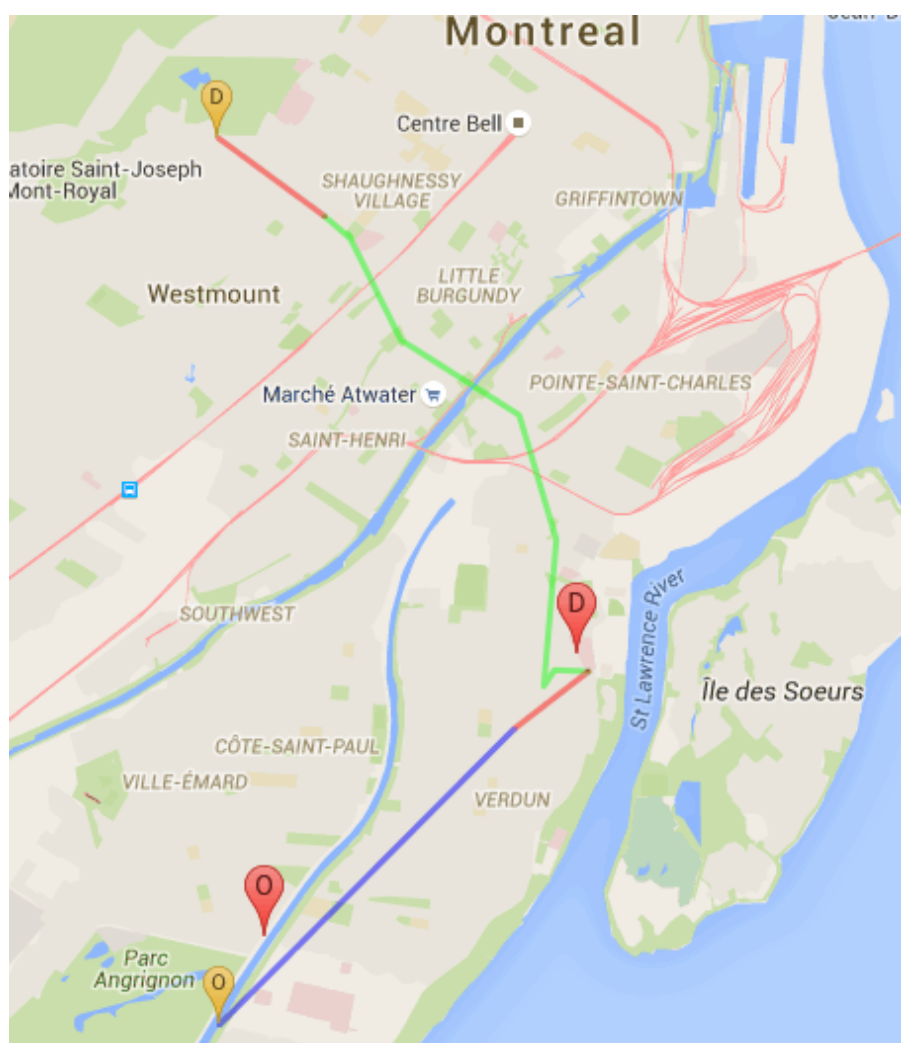


Figure 3.37 : Exemple 2 de jumelage multimodal Routier-TC

CHAPITRE 4 RECOMMANDATIONS ET CONCLUSION

4.1 Recommandations

4.1.1 Calculs hors-ligne

De manière à accélérer l'algorithme de jumelage multimodal, les poids des plus courts chemins et les chemins multimodaux entre chaque paire de nœuds routiers (centroïdes) du graphe G_M peuvent être calculés à l'avance. Ils correspondent aux trajets OD des passagers et conducteurs qui utiliseront le service.

Ceci réduirait considérablement la durée d'exécution du tri ajustable.

Afin de réaliser les calculs hors-ligne, il est nécessaire d'implémenter un algorithme de plus court chemin efficace. L'algorithme A* permettrait d'obtenir de meilleurs résultats en temps de calcul.

4.1.2 Calibrage du tri ajustable

Il est nécessaire d'ajuster les paramètres du tri ajustable en fonction du nombre de trajets de la base de données de conducteurs et de passagers à jumeler. Un petit nombre d'utilisateur est compatible avec des paramètres peu stricts, puisque le temps alloué au calcul peut être concentré sur les calculs de détours réels et la sélection des jumelages, tandis qu'un grand nombre d'utilisateurs implique un tri ajustable conséquent, et donc des paramètres plus stricts.

De plus, il convient de définir le nombre de jumelages à considérer lors de la phase de sélection des jumelages. Comme en atteste les résultats des tests pour le covoiturage classique, il n'est pas toujours assuré d'obtenir des jumelages satisfaisants parmi les trois premiers conducteurs. Il peut donc être parfois nécessaire de s'attarder sur des conducteurs moins bien placés, de manière à essayer de fournir de bons covoitureurs.

4.1.3 Pistes d'amélioration

La qualité de l'algorithme repose en partie sur la qualité du graphe. Plus il est précis, plus l'algorithme propose des solutions exactes. L'une des pistes d'amélioration consiste donc à affiner le graphe, en effectuant un découpage plus précis (zones de 1km*1km) et en considérant différents graphes (avec différents poids) correspondants aux différentes heures de la journée – période de

pointe du matin et période hors-pointe par exemple. Il faudrait également ajouter les lignes de transports en commun manquantes (celles qui sont en dehors de l'île de Montréal et qui ne sont pas les trains de banlieue).

Les résultats obtenus pour le covoiturage multimodal sont plus encourageants pour le type Routier-TC que pour le type TC-Routier, pour la période de pointe du matin. Il faut néanmoins rappeler que les hypothèses et modèles développés ne sont pas parfaits, et qu'un résultat énoncé dans ce mémoire ne saurait justifier d'une vérité générale sur le potentiel d'un type de covoiturage multimodal par rapport à un autre. Il semble nécessaire que les critères du tri ajustable pour ce type de covoiturage soient très précis, puisque les aléas générés par les embouteillages et par les trajets en transports en commun empêchent de combiner les deux modes de transports. Ces résultats sont aussi à nuancer positivement car les temps de parcours calculés ne prennent pas en compte les embouteillages : dans le calcul des détours, le temps de parcours habituel d'un utilisateur est comparé au temps de parcours en multimodal mais considération du trafic. Les tolérances horaires des passagers pourraient donc basculer du côté positif en multimodal en prenant en compte les embouteillages.

Des calculs sont également nécessaires avec des trajets effectués en période de pointe du soir. Ceci permettrait d'évaluer le potentiel du covoiturage multimodal pour le retour à la maison.

Il faudrait également intégrer les points sécuritaires de dépose et de récupération des passagers dans l'algorithme.

De manière générale, il est également possible de laisser une certaine liberté à l'utilisateur. Il est aisé d'imaginer un réseau social où chaque utilisateur contribuerait à cartographier Montréal avec les points de dépose sécuritaires. Pour le covoiturage multimodal, l'algorithme pourrait seulement indiquer au passager des lignes de transports en commun permettant d'accéder à destination et proposer des jumelages possibles pour accéder à la ligne en question. Ceci serait possible grâce à l'algorithme de recherche de chemins multimodaux.

4.2 Contributions

La classification établie dans le premier chapitre suite à la revue de littérature permet à d'autres chercheurs d'avoir une vue d'ensemble sur les techniques qui ont été imaginées pour le

covoiturage. Elle définit également les contours de ce que doit être un algorithme de jumelage multimodal entre passagers et conducteurs : rapide et précis.

Le projet relatif à ce mémoire présente plusieurs façons d’appréhender le potentiel de covoiturage entre deux utilisateurs. Le second chapitre s’attarde sur deux modélisations simplifiées de ce potentiel : un score le représentant et diverses régressions linéaires tentant d’approximer le détour entre deux trajets (notion définie dans le premier chapitre). Le troisième chapitre présente l’élaboration du graphe G_M multimodal représentant Montréal, ainsi que ses usages pour l’algorithme développé. Les résultats de validation du graphe et de l’algorithme sont présentés et sont encourageants pour le covoiturage classique tandis que le covoiturage multimodal s’avère plus difficile à mettre en œuvre, mais présente une marge d’amélioration (cf. 4.1). Il est à noter que les résultats présentés dans ce mémoire sont purement mathématiques et preuves de concepts algorithmiques, et n’ont pas vocation à définir quel détour ou quelle tolérance horaire sont acceptables pour un utilisateur final.

4.3 Conclusion

Il est possible de classer l’algorithme développé en reprenant la classification établie dans le chapitre 1. Celui-ci est plutôt décentralisé et il utilise un graphe comme structure de données. Il fait aussi appel à des outils relatifs à la théorie des graphes pour la modélisation (plus court chemin notamment). La résolution est en deux parties : une première phase de tri puis une seconde phase de calcul d’itinéraire plus précis. Les décisions prises sont les horaires des conducteurs et passagers (H), les points de transferts pour les passagers utilisant d’autres modes de transport (T), le rôle des utilisateurs (R), et l’algorithme prend en compte les détours acceptables (D). En termes de fréquence de décisions, il peut être envisagé à différents niveaux. La proposition de la partie 3.2.3 concernant le processus de jumelage dynamique considère qu’il faut relancer l’algorithme à chaque nouvelle demande de trajet.

L’algorithme final de jumelage multimodal développé présente donc l’avantage de reposer sur une structure de données – le graphe G_M – qui nécessite des mises à jour ponctuelles. Les calculs de jumelages en sont ainsi accélérés, puisque les plus courts chemins entre les différents nœuds de G_M et les chemins multimodaux peuvent être calculés au préalable.

Il est flexible et ajustable au nombre d’offres de trajets grâce à différents paramètres qui prennent en compte les préférences des usagers. Grâce à un premier tri, les « mauvais » conducteurs peuvent être rapidement éliminés, et les calculs complexes de jumelage sont ainsi effectués sur une liste réduite de conducteurs disponibles : calcul de détour et d’itinéraires précis.

Les résultats des tests sur l’enquête OD – en période de pointe du matin – sont encourageants pour le covoiturage classique, puisque les jumelages obtenus après classement sont relativement de bonne qualité. Les résultats pour le covoiturage multimodal sont moins convaincants, même s’ils sont encourageants pour le covoiturage multimodal de type Routier-TC. De futures recherches permettraient d’établir plus précisément le potentiel du covoiturage multimodal, notamment avec des tests sur des trajets pendant la période de pointe du soir. Il faut cependant nuancer ces résultats positivement, car les embouteillages ne sont pas pris en compte dans le modèle développé, ce qui permettrait d’avoir de meilleurs résultats quant au respect des tolérances horaires des utilisateurs.

La qualité du graphe G_M construit pourrait aussi expliquer en partie la qualité des résultats obtenus pour le covoiturage multimodal.

Avec l’algorithme développé et la structure de données imaginée, Netlift dispose malgré tout d’un système permettant de proposer des jumelages de qualité, plus rapidement que l’algorithme qu’ils utilisent actuellement.

BIBLIOGRAPHIE

- Agatz, N., Erera, A., Savelsbergh, M., & Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2), 295-303. doi:10.1016/j.ejor.2012.05.028
- Aissat, K., & Oulamara, A. (2015). The Round-Trip Ridesharing Problem with Relay Stations. 9335, 16-30. doi:10.1007/978-3-319-24264-4_2
- Aissat, K., & Varone, S. (2015). Multi-modal Transportation with Public Transport and Ride-sharing - Multi-modal Transportation using a Path-based Method. 479-486. doi:10.5220/0005366204790486
- Armant, V., & Brown, K. N. (2014). Minimizing the Driving Distance in Ride Sharing Systems. 568-575. doi:10.1109/ictai.2014.91
- Baldacci, R., Maniezzo, V., & Mingozzi, A. (2004). An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation. *Operations Research*, 52(3), 422-439. doi:10.1287/opre.1030.0106
- Bruglieri, M., Ciccarelli, D., Colorni, A., & Luè, A. (2011). PoliUniPool: a carpooling system for universities. *Procedia - Social and Behavioral Sciences*, 20, 558-567. doi:10.1016/j.sbspro.2011.08.062
- Cao, B., Alarabi, L., Mokbel, M. F., & Basalamah, A. (2015). SHAREK: A Scalable Dynamic Ride Sharing System. 4-13. doi:10.1109/mdm.2015.12
- Coltin, B., & Veloso, M. (2014). Ridesharing with passenger transfers. 3278-3283. doi:10.1109/iros.2014.6943018
- Dakroub, O., Boukhater, C. M., Lahoud, F., Awad, M., & Artail, H. (2013). An intelligent carpooling app for a green social solution to traffic and parking congestions. 2401-2408. doi:10.1109/itsc.2013.6728586
- Drews, F., & Luxen, D. (2013). *Multi-hop ride sharing*. Communication présentée à Sixth Annual Symposium on Combinatorial Search, SOCS 2013, Leavenworth, Washington, USA,.
- Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., & Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57, 28-46. doi:10.1016/j.trb.2013.08.012
- Galland, S., Knapen, L., Yasar, A.-U.-H., Gaud, N., Janssens, D., Lamotte, O., . . . Wets, G. (2014). Multi-agent simulation of individual mobility behavior in carpooling. *Transportation Research Part C: Emerging Technologies*, 45, 83-98. doi:10.1016/j.trc.2013.12.012
- Geisberger, R., Luxen, D., Sanders, P., Neubauer, S., & Volker, L. (2010). *Fast Detour Computation for Ride Sharing*. Communication présentée à Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'10).

- Guo, Y., Goncalves, G., & Hsu, T. (2013). A multi-destination daily carpooling problem and an ant colony based resolution method. *RAIRO - Operations Research*, 47(4), 399-428. doi:10.1051/ro/2013049
- He, W., Hwang, K., & Li, D. (2014). Intelligent Carpool Routing for Urban Ridesharing by Mining GPS Trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 15(5), 2286-2296. doi:10.1109/tits.2014.2315521
- Herbawi, W., & Weber, M. (2011). Evolutionary Multiobjective Route Planning in Dynamic Multi-hop Ridesharing. 6622, 84-95. doi:10.1007/978-3-642-20364-0_8
- Hrnčíř, J., Rovatsos, M., & Jakob, M. (2015). Ridesharing on Timetabled Transport Services: A Multiagent Planning Approach. *Journal of Intelligent Transportation Systems*, 19(1), 89-105. doi:10.1080/15472450.2014.941759
- Huang, S.-C., Jiau, M.-K., & Lin, C.-H. (2015). Optimization of the Carpool Service Problem via a Fuzzy-Controlled Genetic Algorithm. *IEEE Transactions on Fuzzy Systems*, 23(5), 1698-1712. doi:10.1109/tfuzz.2014.2374194
- Huang, Y., Bastani, F., Jin, R., & Wang, X. S. (2014). Large scale real-time ridesharing with service guarantee on road networks. *Proceedings of the VLDB Endowment*, 7(14), 2017-2028. doi:10.14778/2733085.2733106
- Meng, C., Wang, H., Shao, Z., & Ding, Y. (2013). Heuristic Optimization Algorithms for Solving MRMPT. 256, 81-91. doi:10.1007/978-3-642-38466-0_10
- Naoum-Sawaya, J., Cogill, R., Ghaddar, B., Sajja, S., Shorten, R., Taheri, N., . . . Wirth, F. (2015). Stochastic optimization approach for the car placement problem in ridesharing systems. *Transportation Research Part B: Methodological*, 80, 173-184. doi:10.1016/j.trb.2015.07.001
- Nourinejad, M., & Roorda, M. J. (2016). Agent based model for dynamic ridesharing. *Transportation Research Part C: Emerging Technologies*, 64, 117-132. doi:10.1016/j.trc.2015.07.016
- Pelzer, D., Xiao, J., Zehe, D., Lees, M. H., Knoll, A. C., & Aydt, H. (2015). A Partition-Based Match Making Algorithm for Dynamic Ridesharing. *IEEE Transactions on Intelligent Transportation Systems*, 16(5), 2587-2598. doi:10.1109/tits.2015.2413453
- Sghaier, M., Zgaya, H., Hammadi, S., & Tahon, C. (2011). A novel approach based on a distributed dynamic graph modeling set up over a subdivision process to deal with distributed optimized real time carpooling requests. 1311-1316. doi:10.1109/itsc.2011.6082983
- Ta Anh, S., Le Thi Hoai, A., Pham Dinh, T., & Khadraoui, D. (2012). A distributed algorithm solving multiobjective dynamic car pooling problem. 231-236. doi:10.1109/ICCISci.2012.6297245
- Vargas, M. A., Sefair, J., Walteros, J. L., Medaglia, A. L., & Rivera, L. (2008). Car pooling optimization: A case study in Strasbourg (France). 89-94. doi:10.1109/sieds.2008.4559691
- Xu, H., Pang, J.-S., Ordóñez, F., & Dessouky, M. (2015). Complementarity models for traffic equilibrium with ridesharing. *Transportation Research Part B: Methodological*, 81, 161-182. doi:10.1016/j.trb.2015.08.013

- Yan, S., & Chen, C.-Y. (2011). A model and a solution algorithm for the car pooling problem with pre-matching information. *Computers & Industrial Engineering*, 61(3), 512-524. doi:10.1016/j.cie.2011.04.006
- Yousaf, J., Li, J., Chen, L., Tang, J., Dai, X., & Du, J. (2012). Ride-Sharing: A Multi Source-Destination Path Planning Approach. 7691, 815-826. doi:10.1007/978-3-642-35101-3_69
- Zhao, D., Zhang, D., Gerding, E. H., Sakurai, Y., & Yokoo, M. (2014). *Incentives in ridesharing with deficit control*. Communication présentée à AAMAS. 2014.
- Zhu, J., Feng, Y., & Liu, B. (2013). PASS: Parking-Lot-Assisted Carpool over Vehicular Ad Hoc Networks. *International Journal of Distributed Sensor Networks*, 2013, 1-9. doi:10.1155/2013/491756

Accord de Paris sur le climat. Dans Wikipédia. Repéré en 2016 à :

https://fr.wikipedia.org/wiki/Accord_de_Paris_sur_le_climats

Consommation d'énergie finale dans le monde. Repéré en 2016 à :

<http://www.connaissancedesenergies.org/fiche-pedagogique/consommation-d-energie-finale-dans-le-monde-0>

ANNEXES

ANNEXE A – IMPLÉMENTATION

Le langage de programmation et l'environnement de développement intégré sont respectivement JAVA et Eclipse. Chaque ressource était sauvegardée et réutilisée sous la forme d'objets JAVA.

Les APIs de Google Maps ont été utilisées pour les calculs précis d'itinéraires. Les calculs d'itinéraire en auto sont effectués à minuit, et les calculs d'itinéraire en transports en commun sont effectués à 8h en semaine.

Les données géographiques utilisées dans le premier chapitre proviennent d'OpenStreetMap - principalement le graphe de la ville de Montréal permettant de faire les calculs de trajets.

Pour ce qui concerne l'implémentation en elle-même, plusieurs librairies ont été utilisées :

- OpenTripPlanner, pour les calculs de trajets sur la base du graphe de la ville de Montréal dans le chapitre 2
- GeoTools et Java Topology Suite dans le chapitre 2 pour la conception et l'affichage des différentes formes géométriques : zones, trajets, points du graphe.
- Apache POI
- Common collections
- JGraphT pour la modélisation et les calculs de trajets dans le chapitre 3

ANNEXE B – EXEMPLE DE RÉSULTAT DE RÉGRESSION LINÉAIRE

RAPPORT DÉTAILLÉ	Variables x3, x4 et x5			
Statistiques de la régression				Valeur critique de F
Coefficient de détermination multiple	0,77			
Coefficient de détermination R^2	0,59			5,3128E-118
Erreur-type	28,83			
Observations	600			
ANALYSE DE VARIANCE				
	Degré de liberté	Somme des carrés	Moyenne des carrés	F
Régression	3	728158	242719	292
Résidus	618	513712	831	
Total	621	1241870		
	Coefficients	Erreur-type	Statistique t	Probabilité
Constante	174,13	4,26	40,91	5,2107E-178
x3	203,67	8,67	23,50	9,17737E-88
x4	121,66	8,09	15,04	8,47205E-44
x5	-4,62	0,53	-8,74	2,22806E-17
	Limite inférieure pour seuil de confiance = 95%	Limite supérieure pour seuil de confiance = 95%	Limite inférieure pour seuil de confiance = 95%	Limite supérieure pour seuil de confiance = 95%
Constante	165,77	182,49	165,77	182,49
x3	186,65	220,69	186,65	220,69
x4	105,78	137,54	105,78	137,54
x5	-5,66	-3,58	-5,66	-3,58