UNIVERSITÉ DE MONTRÉAL

OUTILS DE VISUALISATION
DE DONNÉES DE CARTES À PUCE
POUR UNE SOCIÉTÉ DE TRANSPORT COLLECTIF

ANTOINE GIRAUD

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUTRIEL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION

DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES

(GÉNIE INDUSTRIEL)

AVRIL 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

OUTILS DE VISUALISATION DE DONNÉES DE CARTES À PUCE POUR UNE SOCIÉTÉ DE TRANSPORT COLLECTIF

présenté par : GIRAUD Antoine

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

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

M. AGARD Bruno, Doctorat, président

M. TRÉPANIER Martin, Ph. D., membre et directeur de recherche

Mme MORENCY Catherine, Ph. D., membre et codirectrice de recherche

M. NAZEM Mohsen, Ph. D., membre

DÉDICACE

À ma famille et mes amis qui m'ont soutenu
au cours de mes années d'études.

REMERCIEMENTS

Je tiens à remercier Martin Trépanier, directeur de recherche du département de mathématiques et de génie industriel, ainsi que Catherine Morency, co-directrice de recherche du département de génie civil, pour m'avoir conseillé, assisté et subventionné au cours de cette maîtrise recherche à l'École Polytechnique de Montréal. Leur soutien et leur expertise m'ont permis de découvrir encore plus le monde des transports, des systèmes d'information et de confirmer leur pertinence pour l'aide à la décision dans le cœur de métier des entreprises.

Je remercie aussi les représentants du Réseau de Transport de Longueuil pour l'accès à leurs données nécessaires à ce projet de recherche, ainsi que pour leur enthousiasme et leur vif intérêt dans les travaux menés. Je souhaite que les visualisations réalisées puissent être utiles dans leurs activités de gestion et de planification de leur offre de services de transport public.

Je remercie aussi l'équipe du CeNTAI de Thales Communications and Security du site de Paris – Gennevilliers ainsi que celle de Thales TRT à Québec pour leur accueil et leur disponibilité ainsi que pour leur présentation et l'accès à leur interface de visualisation de données. Cela m'a mis le pied à l'étrier pour parfaire ma connaissance des outils de visualisation de données notamment grâce à des outils libres utilisés dans ce domaine. J'espère que les travaux ici menés leur permettront de mieux percevoir les exigences d'une société de transport en commun ainsi que les types de visualisations pouvant les assister dans leur prise de décisions.

Mes remerciements vont aussi au Conseil de Recherches en Sciences Naturelles et en Génie du Canada (CRSNG) et le programme PROMPT qui ont financé cette maîtrise recherche.

Je tiens enfin à remercier d'autres professeurs de Polytechnique notamment ceux du département du génie civil en transport – plus particulièrement les professeurs Robert Chapleau et Nicolas Saunier – pour m'avoir fait découvrir, grâce à leurs cours et à leur savoir, ce monde des transports non plus côté industriel comme vu dans mon école d'origine, l'Institut Catholique d'Arts et Métiers de Lille en France, mais vu également du côté du client et de l'exploitant d'un réseau de transport en commun.

RÉSUMÉ

De plus en plus de sociétés de transport en commun font le choix de systèmes automatisés de perception des titres de transport par carte à puce et réalisent que ces données recueillies au jour le jour, depuis déjà 2008 pour la grande région de Montréal, constituent un potentiel immense à exploiter pour la planification de leur offre de transport.

Dans ce contexte, cette maîtrise recherche s'inscrit dans un projet global mené depuis trois ans en collaboration avec divers partenaires. Elle fait suite aux précédents travaux de recherche menés sur l'enrichissement des transactions des cartes à puce en leur associant les origines et les destinations des trajets. Pour les fins de ce projet, le Réseau de Transport de Longueuil (RTL) a mis à disposition les 3,1 millions de transactions de bus et de métro du mois de mars 2013. La société Thales a rendu disponible son portail « Analytics For Transportation » développé par le département CeNTAI (Centre de Traitement et d'Analyse de l'Information).

L'objectif principal de cette maîtrise recherche est de concevoir des interfaces permettant de visualiser et d'analyser les transactions de cartes à puce, enrichies de leurs destinations, répondant ainsi aux besoins d'un exploitant de transport en commun.

Les sous-objectifs, correspondant aux étapes de la recherche, sont les suivants :

- Rendre opérationnel l'algorithme de détermination des destinations
- Conceptualiser la structure des données la plus adéquate pour permettre leur visualisation
- Créer des interfaces de visualisation répondant aux besoins d'un exploitant de transport en commun.

Ce mémoire commence par une revue de littérature présentant d'une part les projets des années précédentes sur l'estimation de l'origine puis de la destination des déplacements et d'autre part d'autres projets liés à la visualisation de ce type de données. Les raisonnements employés pour répondre aux trois sous-objectifs précités sont exposés dans une section méthodologie. La dernière section présente les résultats et analyses obtenus à partir de ces données enrichies.

Les contributions apportées par ce mémoire sont :

- L'optimisation et la refonte de l'algorithme d'estimation des destinations et son adaptation à un réseau défini selon le format GTFS (General Transit Feed Specification)

- La présentation d'aperçus rapides et ergonomiques obtenus grâce à l'utilisation d'outils libres (Elasticsearch et Kibana) analysant ces données enrichies de carte à puce
- La conception d'une nouvelle interface web personnalisée et développée pour alimenter un tableau de bord présentant des indicateurs clés pour une société de transport en commun à partir des données transmises par le RTL.

En conclusion, ce projet de recherche propose une solution opérationnelle qui, pour un jeu de données de transactions de cartes à puce, permet, en une étape, d'estimer la destination des trajets de chaque transaction des usagers, de préparer des statistiques supplémentaires (distance et temps de trajet, séquences de tronçons …) et de les exporter vers un fichier texte et vers une base de données (Elasticsearch). Le tout est réalisé en un temps relativement court : 20 minutes pour 3 millions de transactions, temps d'exportation compris. Les données sont alors directement disponibles et exploitables dans des portails web configurés ou développés pour l'occasion prenant en compte les besoins des clients.

Parmi les 3,1 millions de transactions disponibles, 20% sont des transactions de métro. Ces dernières permettent d'aider l'algorithme dans l'estimation des destinations. Lorsqu'elles sont prises en compte, elles n'améliorent que de 1% le nombre total de destinations des trajets de bus portant à 79% le nombre de trajets OD en bus recomposés pour notre jeu de données de mars 2013. Les séquences de tronçons ou déplacements ont été recomposées au cours de l'algorithme. Il en ressort par exemple que 66% des déplacements de bus, ou séquences de tronçons, effectués par les usagers sont des trajets directs sans correspondance. La part d'usagers effectuant des déplacements d'une seule correspondance est respectivement 12% du bus vers le bus et 20% du bus vers le métro.

En définitive, ce projet de recherche permet de montrer que l'analyse de gros volumes de données en un temps limité est possible et une solution opérationnelle est présentée. En effet, il faudrait un temps de traitement de seulement 32 heures pour enrichir les transactions des 8 dernières années du RTL, à raison de 3 millions de données par mois. Ces données de type OD seraient alors disponibles pour alimenter les analyses des différents départements d'une société de transport en commun tels que la gestion des opérations du réseau, la planification et même le marketing et la finance. Les outils de visualisation développés permettraient alors d'aider le RTL dans la rédaction d'un cahier des charges auprès d'une entreprise offrant des solutions BI (Business Intelligence) pour visualiser leurs données métier.

ABSTRACT

Public transit authorities are choosing more and more smart card automated fare collection systems and realize that those daily recovered data, since 2008 for the greater Montreal region, have a great potential for their planning and operations.

In this context, this research master is part of a global project held for three-year period in collaboration with various partners. It follows previous research works on data enrichment of smart card transactions by combining their trip origin and destination. For the purpose of this project, the transit authority RTL (Réseau de Transport de Longueuil) provided one month (March 2013) of bus and metro smart card transactions (3.1 million). As far as Thales is concerned, they made available their "Analytics For Transportation" portal developed by its CeNTAI Department (Centre de Traitement et d'Analyse de l'Information).

The main objective of this master research is to design interfaces for viewing and analyzing smart card transactions, enriched of their destination, while meeting the needs of a transit operator.

The sub-objectives, corresponding to the steps of this research, are:

- Make operational the algorithm determining trip destinations
- Conceptualize the most adequate data structure enabling their visualization
- Design visualization interfaces meeting the needs of a transit operator

This thesis starts with a literature review with, on the one hand, the previous works on the estimation of the trips origin and destination, and, on the other hand, other projects on data visualization. The steps followed to meet the above three sub-objectives are described in the methodology section. The final section presents the results and analysis obtained from these enriched data.

The main achievements of this project are:

- The optimization and redesign of the algorithm estimating trip destinations and its adaptation to a network defined with the GTFS format (General Transit Feed Specification)
- The presentation of ergonomic insights, obtained thanks to the use open source tools (Elasticsearch, Kibana), enabling those enriched smart card data to be quickly analyzed

- The design of a new customized web interface developed to present other key indicators used by a public transport company

In conclusion, this research project presents an operational solution, which for a set of smart card transaction data offers, in one step, to estimate the destination of each smart card transaction trip, to prepare additional statistics (distance and travel time, trip-leg sequences …) and to export those enriched transactions to a text file or a data base (Elasticsearch). The whole process is made within a relatively short time: 20 minutes for 3 million transactions, export time included. The data is then directly available and usable in web portals configured or developed for the occasion and which take into account the needs of the customers.

Of the 3.1 million available transactions, 20% are metro transactions. These transactions help the algorithm in the estimation of a trip destination. These metro transactions only help to find 1 more percent of destinations, resulting in 79% of trip destinations recovered for our March 2013 dataset. Trip-legs have also been reconstructed by the algorithm. It shows for example that 66% of bus travels are made without a transfer. The share of users making only one transfer represents respectively 12% from bus to bus and represents 20% from bus to metro.

In the end, this research shows that the analysis of large volume of data within a limited period of time is possible and an operational solution is presented. Indeed, it would require a processing time of 32 hours to enhance the RTL smart card transactions of the last 8 years, with 3 million transactions per month. These OD type of data would then be available to power the analysis of the various departments of a public transit authority such as operations, planning and even marketing and finance. The developed visualization prototypes would then help the RTL in drafting the specifications of a new tool sold and designed by a company selling BI (Business Intelligence) solutions to visualize their business data.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

La liste des sigles et abréviations présente, dans l'ordre alphabétique, les sigles et abréviations utilisés dans le mémoire ainsi que leur signification. En voici quelques exemples :

AFT	Analyze For Transportation – Portail web développé par le CeNTAI et Thales
AMT	Agence métropolitaine de transport
BI	Business Intelligence – Intelligence d'affaire
CeNTAI	Centre de Traitement et d'Analyse de l'Information (laboratoire de Recherche et Développement de Thales)
CRSNG	Conseil de recherches en sciences naturelles et en génie du Canada
CSS	Cascading Style Sheets
CSV	Comma-separated values
D3 (js)	Data-Driven Documents
DOM	Document Object Model
GPS	Global Positioning System
GTFS	General Transit Feed Specification
HTML	Hypertext Markup Language
IP	Internet Protocol
NFC	Near Field Communication
OD	Origine-Destination
PHP	Hypertext Preprocessor
RTL	Réseau de Transport de Longueuil
SDAP	Système de décompte automatique des passagers et de la localisation du bus
SNCF	Société nationale des chemins de fer français
SQL	Structured Query Language
STIF	Syndicat des transports d'Île-de-France

STL	Société de Transport de Laval
STM	Société de Transport de Montréal
STO	Société de Transport de l'Outaouais
SVG	Scalable Vector Graphic
UDS	Université de Sherbrooke (station Longueuil-UDS)
URL	Uniform Resource Locator
XML	eXtensible Markup Language

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Les sociétés de transport en commun sont de plus en plus équipées de solutions électroniques et informatiques pour la validation des titres de transport à l'aide de cartes à puce. Chaque tentative de validation du titre de transport est enregistrée et conservée informatiquement. Les données produites par ces systèmes sont nombreuses et peuvent être considérées comme une problématique de « Big Data ». Des études ont déjà été réalisées pour montrer la pertinence de l'utilisation de ces données à des fins de planification de l'offre de transport.

La région métropolitaine de Montréal dispose d'informations importantes à exploiter avec l'arrivée, dès 2008, de la carte OPUS. Celle-ci est utilisée entre autres par la Société de Transport de Montréal (STM), par la Société de Transport de Laval (STL), par l'Agence métropolitaine de transport (AMT) mais aussi par le Réseau de Transport de Longueuil (RTL). En 2015, ce dernier exploite 91 lignes de bus à l'aide de 451 véhicules.

À l'origine, ces systèmes de carte à puce ont été implantés pour répondre à une problématique financière, à savoir la perception automatisée des titres de transport. De plus, l'extraction de données n'en était qu'à son début et le potentiel de l'exploitation de ces données n'avait pas été pleinement mesuré. Ainsi, lors de leur conception, toutes les informations nécessaires à la planification de l'offre n'ont pas été paramétrées. En effet, les transactions récoltées n'avaient ni besoin d'être géolocalisées, ni besoin de préciser la destination approximative du trajet associé. Désormais, les sociétés exploitantes de transport manifestent un intérêt grandissant pour enrichir ces transactions, recueillies au jour le jour, d'informations sur la demande et l'utilisation du réseau de transport.

Ainsi, le trajet effectué pour chaque transaction n'est pas enregistré et plusieurs traitements sont à concevoir pour que les données soient utilisables pour la planification. Pour chaque transaction, la première étape consiste à la localiser en utilisant par exemple les traces GPS des bus. La seconde étape consiste à retrouver l'origine et la destination du trajet associé. C'est seulement après avoir recomposé une matrice « Origine Destination » (matrice OD) des déplacements des usagers sur le réseau que l'on peut produire des tableaux de bord (graphiques, cartes, indicateurs) fréquemment utilisés par les départements de planification. Pour la visualisation des données, différents outils

d'intelligence d'affaires (BI) sont disponibles sur le marché, mais ceux-ci ne sont pas toujours adaptés aux contraintes propres de chaque société de transport. Ces outils ne peuvent pas visualiser directement les données brutes, il faut au préalable les avoir enrichies tel que précisé précédemment. De plus, ces outils ne disposent pas nécessairement d'objets de visualisation adaptés aux besoins métier d'une société de transport en commun (suivi de l'activité sur le réseau, des profils de charge, des diagrammes espace-temps, etc.)

Ce mémoire s'inscrit d'une part dans cette problématique de valorisation d'un grand volume de données recueillies quotidiennement par une société de transport et d'autre part dans un projet global mené depuis trois ans en collaboration avec plusieurs partenaires : la société Thales, le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), le programme PROMPT, le Réseau de Transport de Longueuil (RTL), l'Université Laval et enfin l'École Polytechnique de Montréal. Ce mémoire poursuit des travaux de recherche menés sur l'exploitation de données : d'une part sur l'assignation géographique des transactions et d'autre part sur la détermination des entrées – sorties sur le réseau (origine/destination). Ce projet est basé sur des données mises à disposition par le RTL (2,5 millions de transactions de bus pour le mois de mars 2013) à partir desquelles l'étudiant Félix Légaré (2014) a retrouvé une localisation de montée pour 91,7% des transactions. Le projet reprend également des éléments de l'algorithme d'estimation des destinations que l'étudiant Li He (2014) a amélioré notamment pour les trajets unitaires.

Enfin, les données de cartes à puce du RTL ont l'avantage d'être issues uniquement d'un réseau de bus. En effet, cela implique que les tracés empruntés par les usagers sont connus, contrairement à un système de métro où l'on peut effectuer des transferts sans revalider son titre de transport. Chaque transaction a été rattachée à une ligne et un voyage GTFS (General Transit Feed Specification) planifié par le RTL. Ces transactions, une fois enrichies, permettent à terme d'obtenir des statistiques précises, obtenues à partir de données réelles, telles que le nombre de kilomètres parcourus par trajet, le temps passé par trajet, etc.

1.2 Objectifs

L'objectif principal de ce mémoire est de répondre aux besoins spécifiques d'un exploitant de transport en commun (le RTL) dans le domaine de la planification de son offre de transport. Il s'agit notamment de concevoir des interfaces permettant de visualiser et d'analyser les données des

transactions des cartes à puce recueillies quotidiennement (« données brutes »). Ces données doivent être enrichies d'informations (origine et destination des trajets effectués) car elles ne figurant pas initialement dans les données brutes.

Les sous-objectifs sont liés aux contraintes propres du système d'information de cette société :

- Rendre opérationnel un algorithme de détermination des destinations (développé dans le cadre d'un précédent projet) pour le cas du RTL;
- Conceptualiser la structure des données la plus adéquate pour permettre leur visualisation et leur analyse;
- Créer des interfaces de visualisation répondant aux besoins d'un exploitant de transport en commun.

1.3 Contenu

Dans un premier temps, la revue de littérature replace ce mémoire dans le contexte des travaux de recherche menés jusqu'à ce jour. Comme le présent projet touche à la fois au domaine des cartes à puce et à celui de la visualisation, deux axes de recherche sont étudiés. Dans un premier temps, la revue de littérature porte sur les travaux précédemment effectués sur l'enrichissement de transactions (origine et destination). Dans un second temps, sont présentées les statistiques métier et visualisations utilisées dans le domaine depuis ces 10 dernières années.

La méthodologie présente les raisonnements employés pour répondre aux problématiques issues des sous-objectifs présentés précédemment. Une première partie relate les travaux effectués pour améliorer l'algorithme destination (développé dans le cadre d'un précédent projet), le rendre plus performant, l'adapter au cas du RTL et faire en sorte que les fichiers de sortie contiennent le plus d'informations possibles pour pouvoir mieux analyser les résultats de l'estimation des destinations. Une seconde partie expose le travail d'intégration des résultats de type OD obtenus préalablement dans un portail web « Analytics For Transportation » développé par le CeNTAI de Thales. Cela permet ensuite la conceptualisation d'une nouvelle structure de données, cette fois désagrégée, pour les transactions enrichies afin de faciliter leur visualisation. Une dernière partie présentera deux outils de visualisation qui ont été paramétrés et développés pour exploiter ces données enrichies par l'algorithme destination : le premier, Kibana, est un logiciel libre, et le second, « Visu Lignes », a été développé spécialement pour les besoins du RTL.

Dans une dernière partie, les expérimentations et résultats obtenus sont présentés ; d'une part les résultats de l'algorithme destination ainsi que plusieurs illustrations d'analyses possibles et d'autre part l'utilisation des visualisations créées avec Kibana. Il y est montré comment ces dernières permettent de faciliter l'analyse des résultats de l'algorithme tout en pouvant facilement comparer plusieurs jeux de données. Pour terminer, la dernière partie présente l'interface « Visu Lignes » conçue spécialement pour mieux répondre aux besoins d'une société exploitante de transport en commun en proposant une vue globale, cartographiée, de la charge du réseau et un tableau récapitulatif des lignes concernées tout en permettant de filtrer temporellement et spatialement les données de transaction enrichies.

En conclusion, les principales contributions et pistes d'améliorations sont exposées et les problématiques de recherche à venir énoncées.

CHAPITRE 2 REVUE DE LITÉRATURE

Cette revue de littérature revient plus en détail sur différents travaux de recherche pertinents par rapport à ce projet. Nous allons revenir dans un premier temps sur le travail effectué pour retrouver la localisation des transactions du Réseau de Transport de Longueuil (Légaré, 2014). Dans un second temps seront présentés différents modèles pour estimer les destinations des déplacements effectués par les utilisateurs des cartes à puce. Suivra enfin une revue de littérature plus large sur les statistiques métier et visualisations attendues pour de telles données de carte à puce. D'autres visualisations ou solutions basées sur d'autres données seront aussi présentées.

2.1 Cartes à puce et autres données disponibles

Ce mémoire s'intéresse principalement au traitement et à la visualisation de données provenant de systèmes de perception par carte à puce en transport collectif urbain. Cependant, c'est en fusionnant celles-ci avec d'autres données que l'on peut en extraire d'avantage d'informations. Un contexte d'utilisation classique de ces données est pour la planification d'un réseau de transport en commun. Pelletier a effectué une revue de littérature (Pelletier, Trépanier, & Morency, 2011) et présente trois catégories d'utilisation de ces données de cartes à puce. On retrouve dans un premier temps une vision stratégique à long terme où l'on va s'intéresser à l'analyse des comportements des usagers du réseau en les catégorisant ou les classifiant ou encore à la prévision de la demande. Dans un second temps, on retrouve le côté « tactique » où l'on travaille plutôt à moyen ou court terme sur des questions comme l'ajustement du service en fonction de l'utilisation du réseau. On retrouve enfin le côté opérationnel où l'on produit des statistiques comme la charge à bord, des indicateurs de performance et où l'on peut s'intéresser à faire remonter les erreurs détectées dans les données collectées du réseau.

2.1.1.1 Données agrégées de comptage sur place ou à bord

Pour alimenter ces différentes phases de planification, on a cherché à utiliser d'autres sources de données bien connues comme des données de comptage agrégées tel que le comptage sur place ou le comptage à bord. Chacun de ces comptages permet d'avoir une perception de l'utilisation du réseau avec par exemple des profils de charge sur une ligne donnée. Cependant, ni l'un ni l'autre ne permet par exemple de lier les montées et descentes aux différents points de la ligne entre elles.

On ne peut pas en déduire de matrice Origine Destination, ou de séquence de tronçons, car on ne peut pas relier l'entrée et la sortie d'une personne sur le réseau.

Par contre, en liant ces données de comptage à bord à celles de carte à puce, on peut par exemple tenter de détecter une évasion tarifaire potentielle et informer les agents de contrôle (Pourmonet, Bassetto, & Trépanier, 2015).

2.1.1.2 Données de localisation - GPS

Les données de localisation GPS des bus équipés permettent de connaître la localisation et la vitesse moyenne de ces derniers au cours de la journée. Il est ainsi possible par extension de visualiser les points sensibles de congestion du réseau ou bien de prévoir l'arrivée du prochain bus aux arrêts suivants. Avec des données GPS sont parfois fournies les données de l'odomètre ou le nombre de mètres parcourus depuis le début de la ligne par le bus. L'acquisition de ces différentes données s'effectue à intervalle régulier et parfois à chaque arrêt du bus ou à chaque ouverture des portes – on dispose alors du temps d'ouverture des portes.

Ces données vont permettre d'imputer la localisation aux transactions de carte à puce, soit les origines des déplacements sur le réseau de transport en commun (Légaré, 2014).

Ces données peuvent de plus être couplées aux données de carte à puce pour alors estimer l'heure d'arrivée à l'arrêt et ainsi estimer le service réellement effectué, afin de le comparer au service planifié (format GTFS par exemple) (Chu & Chapleau, 2008).

2.1.1.3 Données présentant le service planifié - GTFS

Le service planifié peut être représenté par un GTFS (General Transit Feed Specification, (Google)). Il s'agit d'un ensemble de fichiers regroupant les arrêts, lignes, courses des bus et arrêts-lignes du réseau. Ces fichiers représentent le service qui a été prévu par la société exploitante du transport en commun. Comme précisé au paragraphe précédent, en comparant ces données du service planifié à celles de localisation des bus lors de leurs services, on peut estimer l'adéquation entre le service proposé et celui planifié.

2.1.1.4 Données d'enquête des ménages ou enquête Origine-Destination

Ces enquêtes sont effectuées tous les 5 ans (2003, 2008, 2013) dans la grande région de Montréal sur un échantillon de 5% de la population. Ces enquêtes téléphoniques capturent les déplacements d'un foyer sur une journée donnée.

Ces enquêtes sont en particulier très coûteuses en temps de traitement pour redresser les données récupérées des échantillons de population interrogés. De plus, les déplacements rapportés ne sont valables que sur une journée d'une période donnée de l'année.

Des travaux ont été réalisés pour essayer de lier les données d'enquête OD à celles de carte à puce (Spurr, Chu, Chapleau, & Piché, 2015). Ces travaux montrent que ces données présentent des inconvénients, notamment du fait que les personnes répondantes ne sont pas toujours fidèles à se souvenir précisément de leurs déplacements, elles ont le mérite de fournir des informations sociodémographiques dont les données de carte à puce ne disposent pas pour des raisons de confidentialité. Ces données sociodémographiques sont utilisées pour les planifications à long terme. De plus, des outils de visualisations spécialisés ont été développés pour mieux analyser ces données issues des enquêtes (Morency, Trépanier, Piché, & Chapleau, 2010).

2.1.1.5 Données de transaction des cartes à puce

Un système de cartes à puce est une solution technologique, qui est utilisée de plus en plus depuis les années 1990, et qui permet aux usagers d'un réseau de transport en commun de disposer d'une carte à puce qui porte leurs titres de transports ou abonnements. L'usager n'a plus qu'à valider cette carte sur des lecteurs aux entrées du réseau qui valident ou non le titre de transport – typiquement à la montée dans le bus ou à l'entrée des stations de métro. Une transaction numérique est enregistrée, permettant de débiter le compte de la personne ou bien d'autoriser l'accès si l'abonnement en cours est valide. Il faut noter qu'une minorité de systèmes demandent aux usagers de valider à la sortie en plus de l'entrée pour alors effectuer un tarif basé sur le kilométrage. De tels systèmes permettent de réduire considérablement le nombre de tickets ou cartes d'abonnement papier qui étaient utilisés jusque-là. Dans la grande région de Montréal, ce système est en place depuis 2008 par le biais de la carte OPUS.

Ces données représentent une mine d'informations. Cependant elles ne sont pas exploitables directement. Pour en extraire de l'information, il faut les raffiner ainsi que prévenir des erreurs

possibles (Chu & Chapleau, 2007). En effet, les données de transaction ayant été destinées historiquement et avant tout à la perception des titres de transport, celles-ci ne disposent pas forcément de l'arrêt où elles ont été effectuées. Il faut donc les enrichir de leur localisation qui représente alors l'origine d'un déplacement (Légaré, 2014). Une fois cette origine trouvée, il est possible d'estimer la destination (Barry, Newhouser, Rahbee, & Sayeda, 2002), (Tranchant, 2005), (Trépanier, Tranchant, & Chapleau, 2007), (Munizaga & Palma, 2012), (He, 2014). Une fois les destinations retrouvées, il peut être intéressant de préciser si chaque voyage effectué est dans une séquence de tronçons ou non (Chu & Chapleau, 2008).

Ces données de transactions désormais enrichies en déplacements, avec une origine et une destination, offrent de nouvelles perspectives. Celles-ci permettent notamment de générer des matrices Origine-Destination qui sont couramment utilisées dans la planification ; notamment par les outils du groupe MADITUC. Ces matrices ont l'avantage d'être disponibles à chaque jour d'exploitation du réseau. On peut ainsi construire des matrices à différentes résolutions – heure, jour, semaine, mois, année – et ainsi représenter la demande au fil du temps. Une utilisation pourrait être pour des analyses à court terme sur la microélasticité du réseau afin d'examiner la mobilité des usagers suite à une perturbation sur le réseau (par exemple la fermeture du pont Champlain) ou l'instauration d'une nouvelle ligne. Une autre utilisation serait également de modéliser l'évolution de l'utilisation au fil des années.

Des études ont montré qu'il est possible de caractériser le comportement des usagers du réseau en regardant si ces derniers sont « fidèles » et s'ils utilisent peu ou pas de nouvelles stations chaque semaine (Morency, Trépanier, & Agard, 2007). Il est aussi possible de retrouver quels sont les lieux les plus utilisés par une carte à puce. Ainsi, on peut s'intéresser à un groupe de cartes donné se rendant habituellement à un même lieu tel qu'une école et ainsi visualiser d'où proviennent les cartes sur le réseau, avec leur utilisation en fonction de l'heure de la journée (Chu & Chapleau, 2007).

D'autres travaux ont été effectués pour proposer des mesures de performance (Trépanier, Morency, & Agard, 2009) ou encore pour rendre accessibles ces données aux personnes ayant besoin de les utiliser par le biais d'outils de visualisation ou de partage de l'information (Vassivière, 2007), (Lomone, 2014).

Des travaux de visualisation ont été effectués sur ce type de données pour donner un meilleur aperçu de celles-ci (Tao, Corcoran, Mateo-Babiano, & Rohde, 2014), (Anwar, Odoni, & Toh, 2016).

Il est possible de s'inspirer d'autres travaux visualisant aussi des trajets Origine-Destination tels que ceux de (Côme & Oukhellou, 2012) qui proposent un outil pour visualiser des déplacements du réseau de vélopartage Vélib' de Paris.

2.1.1.6 Nouvelles données de transaction

Avec la montée du taux de possession de téléphones intelligents, le bus va pouvoir valider le titre de transport d'une personne étant gardée sur son téléphone grâce par exemple à un lecteur NFC (Near Field Communication) tout en ayant payé en ligne sa transaction. Une autre solution serait de réserver son voyage de tel à tel endroit comme sur le site de vente en ligne de billets de la SNCF (Société nationale des chemins de fer français) avec le trajet emprunté. Ceci permettrait d'avoir les Origines et Destinations plus ou moins précises des utilisateurs directement. Des compagnies telles que Transdev ou Keolis travaillent et réfléchissent à ces nouvelles solutions de monétisation en réalisant des enquêtes sur les différents habitudes et profils d'usagers en lien avec le numérique.

2.2 Retrouver la localisation des transactions

Comme évoqué précédemment, les systèmes de carte à puce servant à la perception des titres de transport n'ont pas été conçus et pensés historiquement pour qu'on utilise les données de transaction à des fins de planifications. Ainsi, même si l'on a un système dans chaque bus qui enregistre les transactions effectuées, ces dernières ne sont pas localisées dans la majorité des systèmes. C'est ainsi le cas pour les données de transactions de carte à puce du RTL qui n'enregistrent pas la localisation lorsque les utilisateurs posent leur carte de transport sur le lecteur à l'entrée du bus. Ce sont des travaux menés à l'École Polytechnique de Montréal (Légaré, 2014) qui ont proposé une méthode pour les retrouver. Pour ce faire, la méthode utilise trois étapes pour l'enrichissement de ces données de transactions.

La première étape se base sur les données issues du système de décompte automatique des passagers à bord (SDAP), qui enregistre le nombre de montants et descendants à des coordonnées GPS rattachées à un arrêt. La deuxième étape s'appuie sur l'historique et l'habitude de l'usager à utiliser certains arrêts pour déterminer les localisations encore manquantes. Enfin, la dernière étape

cherche à attribuer un arrêt à une transaction en regardant l'horaire « prévu » de la transaction à partir des fichiers GTFS (General Transit Feed Specification) qui représentent le service offert par l'exploitant du réseau de transport en commun.

Tableau 2-1 - Récapitulatif de l'apport des différentes étapes d'enrichissement

2 481 977 transactions	0. début	1. SDAP	2. Habitude	3. GTFS
Sans localisation % cumulé	100%	46%	18,80%	8,30%
Avec localisation % cumulé	0%	54%	81,20%	91,70%
Nouvelle localisation par étape		+1 353 372	+659 573	+259 608

La méthode de Légaré a été utilisée sur les données du mois de mars 2013. Elle réussit à affecter une localisation à 92% des transactions soit 2 272 553 transactions localisées sur 2 481 977 possibles. Le Tableau 2-1 récapitule l'avancement des affectations par étape.

2.2.1 Enrichissement par le système de compte à bord et localisation SDAP

Le système de décompte automatique des passagers (SDAP) collecte des données permettant de connaître de façon précise la localisation du tiers des bus du RTL : le numéro du bus – que l'on peut associer à celui inscrit pour chaque transaction – ainsi que le temps d'ouverture des portes, l'heure d'arrivée à l'arrêt, l'heure de départ de l'arrêt et enfin la valeur de l'odomètre, ou position sur la ligne, ou à l'arrêt point de départ de la ligne. Le RTL a filtré ces données en retirant celles qui semblaient erronées.

Légaré réalise en 4 étapes la recherche de la localisation des transactions avec à chaque fois des critères différents.

La première étape traite les transactions en début de ligne (moins de 50 m parcourus). La durée d'ouverture des portes doit être supérieure à 0 seconde. Enfin, on autorise les transactions effectuées 10 secondes avant l'ouverture des portes (arrivée) et jusque 15 secondes après la fermeture des portes (départ). Cette étape est censée empêcher d'attribuer une montée à la fin d'une ligne qui aurait un temps proche de celui de départ du service suivant, qui aurait pu être sans battement. Ainsi, ces transactions de montées au tout début de la ligne seront déjà traitées et ne seront pas retrouvées dans les étapes suivantes disposant de critères plus lâches comme la 4^{e} étape.

La seconde étape traite les correspondances « idéales » en validant des transactions effectuées après l'ouverture des portes (arrivée) et jusqu'à 15 secondes après la fermeture des portes (départ). On garde la contrainte de la durée d'ouverture des portes supérieure à 0.

La troisième étape est la même que la seconde, mais avec des contraintes plus lâches et toujours une durée d'ouverture des portes supérieure à 0. On autorise les transactions effectuées entre 15 secondes avant l'ouverture des portes (arrivée) et jusqu'à 25 secondes après la fermeture des portes (départ). Cela est censé couvrir les décalages d'horloge entre le système de perception des transactions et celui du SDAP.

La dernière étape dispose de contraintes encore plus lâches: on ne tient plus compte de la durée d'ouverture des portes. On autorise les transactions effectuées entre 30 secondes avant l'ouverture des portes (arrivée) et jusqu'à 45 secondes après la fermeture des portes (départ).

Grâce à cette méthode, on arrive à retrouver 54,0% des localisations des transactions avec le système SDAP. Les résultats plus précis par étape sont précisés dans le Tableau 2-2.

Tableau 2-2 - Résultats enrichissement par le système de compte à bord et localisation SDAP – tiré de (Légaré, 2014)

Contraintes *même numéro de bus + …*	**Embarquements recevant une nouvelle localisation**	**En pourcentage**	**Pourcentage cumulatif d'embarquements avec localisation**
10 s avant arrivée 15 s après départ durée d'ouverture des portes > 0 odomètre < 50 m	580 615	23,39%	23,39%
après arrivée 15 s après départ durée d'ouverture des portes > 0	416 211	16,77%	40,16%
15 s avant arrivée 25 s après départ durée d'ouverture des portes > 0	320 463	12,91%	53,07%
30 s avant arrivée 40 s après départ	22 967	0,93%	54,00%

2.2.2 Enrichissement par l'habitude des usagers

Légaré considère que l'historique et les habitudes des usagers sont une méthode plus fiable que celle consistant à comparer l'heure de la transaction à celle du service planifié, car elle aura tendance à ne pas être parfaitement respectée par les autobus. Il se base également sur l'observation portée par des travaux précédents (Morency, Trépanier, & Agard, 2007) sur le fait que les usagers utilisent peu de nouveaux arrêts.

Pour une transaction donnée, on essaie donc de trouver dans l'historique d'autres transactions qui lui sont proches, qui ont donc le même arrêt ou sont à moins de 500 m l'une de l'autre. On peut ainsi attribuer l'arrêt dominant de ces transactions proches. Même si l'arrêt n'est pas absolument exact à 500 m près, on préfère attribuer un arrêt représentatif de la localisation de montée qui est bien sur la même ligne et dans le même sens plutôt que de ne rien considérer. En effet, l'utilisateur peut avoir marché en attendant son bus, avec le choix entre deux arrêts tous deux proches de son domicile.

Pour la démarche à trouver des transactions similaires, on cherche les transactions d'une même carte, sur une même ligne dans un même sens. Une première phase vérifie en plus que l'on a une transaction effectuée dans une même heure de la journée du même type de jour (semaine, fin de semaine). Une seconde phase ne retient pas ces dernières contraintes temporelles.

On arrive ainsi à retrouver la localisation de 27,2% transactions supplémentaires portant à 81,2% le pourcentage de transactions avec localisation. Les résultats plus précis par étape sont précisés dans le Tableau 2-3.

Tableau 2-3 - Résultats de l'enrichissement par l'habitude des usagers – tiré de (Légaré, 2014)

Contraintes *même carte + …*	**Embarquements recevant une nouvelle localisation**	**En pourcentage**	**Pourcentage cumulatif d'embarquements avec localisation**
Même ligne, même direction Même type de jour, même heure de la journée Un seul arrêt correspondant	402 216	16,21%	70,21%
Même ligne, même direction Même type de jour, même heure de la journée Arrêts à moins de 500m entre eux	84 419	3,40%	73,61%
Même ligne, même direction Un seul arrêt correspondant	188 184	7,58%	81,19%

2.2.3 Enrichissement par le service planifié, le GTFS

Légaré met en avant différents inconvénients liés au GTFS. Le premier est que le bus n'est pas toujours à l'heure. Ainsi, un retard de 5 minutes peut engendrer une différence de plusieurs kilomètres entre l'arrêt réellement pris et celui attribué avec le GTFS. Le second est qu'il peut y

avoir plusieurs autobus effectuant le même parcours en même temps. Cela implique que l'on ne peut pas identifier quel est l'autobus réellement emprunté par le passager.

Pour pouvoir affecter un arrêt GTFS à une transaction, on va utiliser comme contraintes d'avoir une même ligne, une même direction, un même numéro de voiture / bus ainsi qu'une même date. On rajoute en plus une tolérance de plus ou moins 60 secondes autour de l'heure de passage à l'arrêt GTFS. Enfin, on valide l'arrêt si toutes les possibilités font partie de la même course GTFS.

Cette étape permet de localiser 259 608 (10.46%) transactions supplémentaires, portant en cumulatif à 91,56% de transactions localisées.

Tableau 2-4 - Tableau récapitulatif des résultats des différents enrichissements avec l'intersection aux résultats de l'enrichissement GTFS – tiré de (Légaré, 2014)

2 481 977 transactions Type d'enrichissement	**Nouvelles transactions localisées**	**∩ GTFS**	**∩ GTFS même arrêt**	**∩ GTFS arrêt proche**	**∩ GTFS autre arrêt**
Match SDAP	1 353 372	715 570	209 526	429 644	76 400
Match Habitude	659 573	354 300	102 702	230 310	21 288
Match GTFS	259 608	1 329 478			
Match SDAP %CAP	54,5%	28,8%	8,4%	17,3%	3,1%
Match Habitude %CAP	26,6%	14,3%	4,1%	9,3%	0,9%
Match GTFS %CAP	10,5%	53,6%			
Match SDAP %∩GTFS		52,9%	29,3%	60,0%	10,7%
Match Habitude %∩GTFS		53,7%	29,0%	65,0%	6,0%

2.2.4 Autres remarques sur les résultats

Légaré a analysé les résultats de sa méthode d'attribution de localisations aux embarquements par jour, par heure de la journée et par ligne en fonction à chaque fois de l'étape d'enrichissement. La Figure 2-1 représentant les taux de réussite des différents enrichissements en fonction du numéro de la ligne montre que lorsque le SDAP n'est pas dominant, ce sont les étapes par habitude puis par GTFS qui prennent le relais. Il y a ainsi des lignes qui sont mieux desservies par le système SDAP.

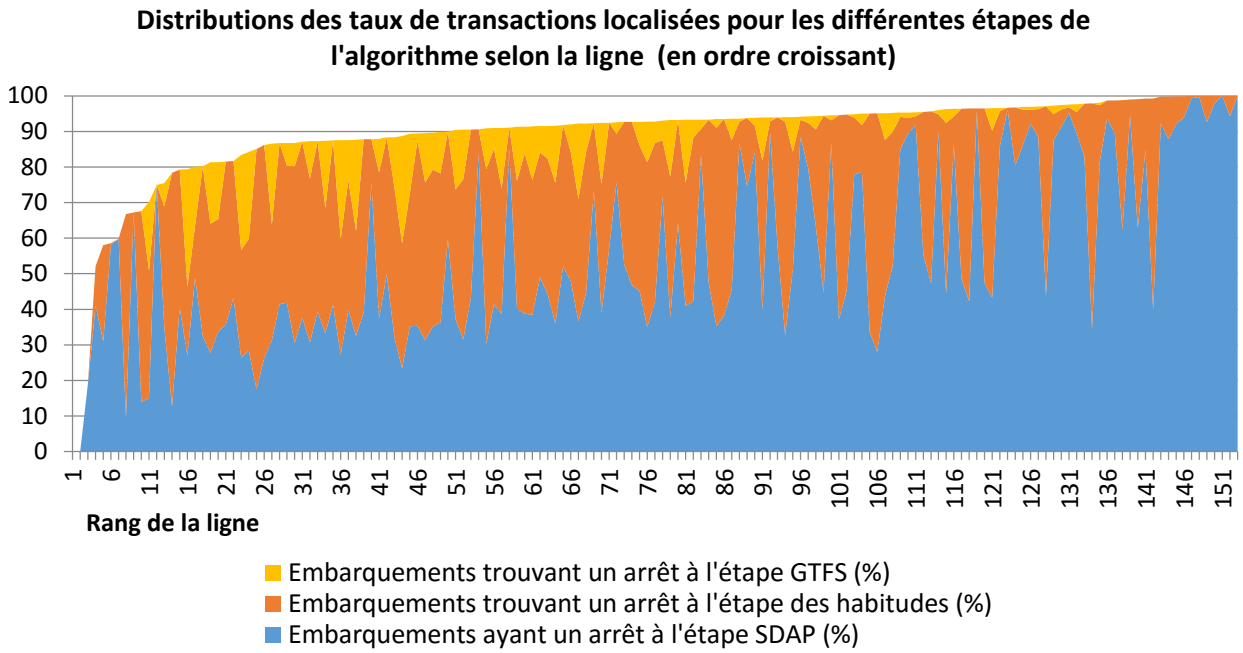


Figure 2-1 - Distributions des taux de transactions localisées pour les différentes étapes de l'algorithme selon la ligne – tiré de (Légaré, 2014)

La Figure 2-2 représentant les taux de réussite des différents enrichissements en fonction de l'heure de la journée montre une proportion stable au long des différentes heures. Cependant, sur cette même Figure 2-2, le graphique de gauche représentant les taux de réussite des différents enrichissements en fonction des jours du mois montre un motif propre aux jours de semaine et ceux de fin de semaine pour l'étape d'enrichissement par habitude – il y a en effet moins d'arrêts retrouvés par habitude en fin de semaine comme il y a moins de déplacements réguliers contrairement à la semaine. L'étape utilisant le SDAP a un taux plutôt stable tournant autour de 50%. Le 31 mars 2013 est une exception.

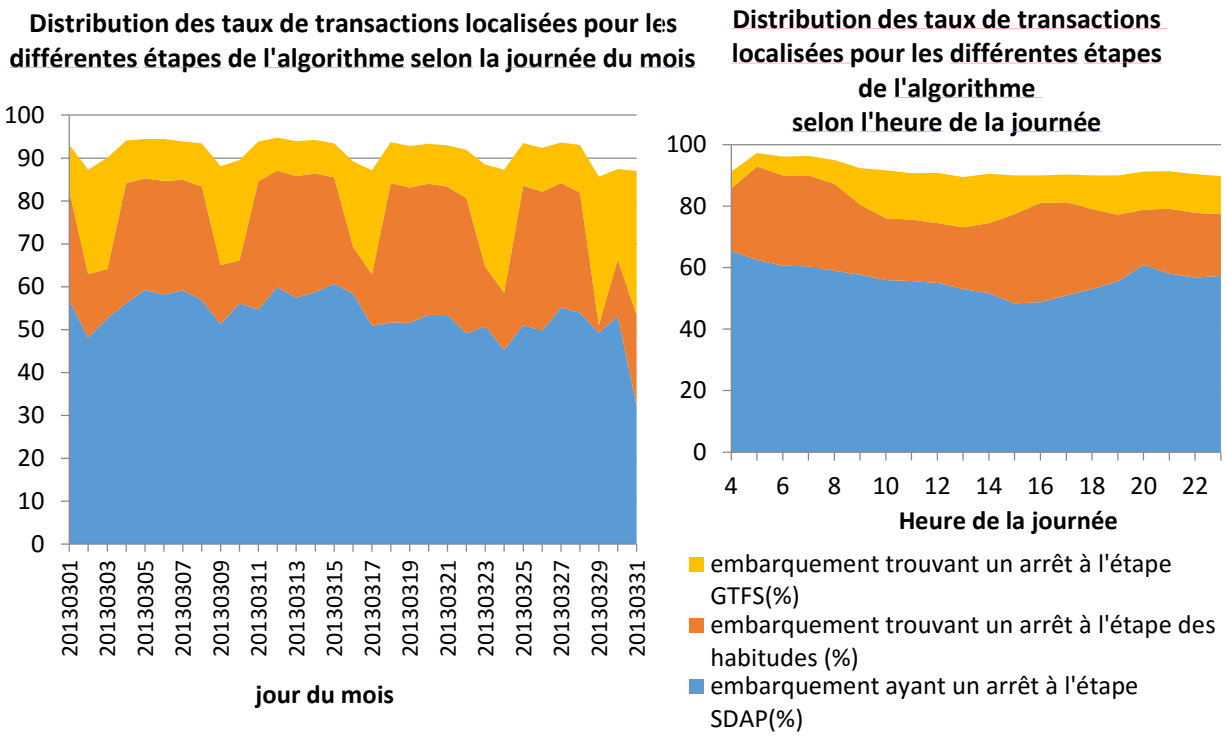


Figure 2-2 - Distributions des taux de transactions localisées pour les différentes étapes de l'algorithme selon le jour du mois et l'heure de la journée – tiré de (Légaré, 2014)

En plus de ces résultats, Légaré a représenté et comparé les transactions ayant obtenu une localisation avec le nombre de passagers montant aux arrêts enregistré par le SDAP. Il montre à l'aide de la Figure 2-3 que le SDAP capture moins de passagers que le nombre de transactions ayant une localisation retrouvée après les trois étapes d'enrichissement décrites.

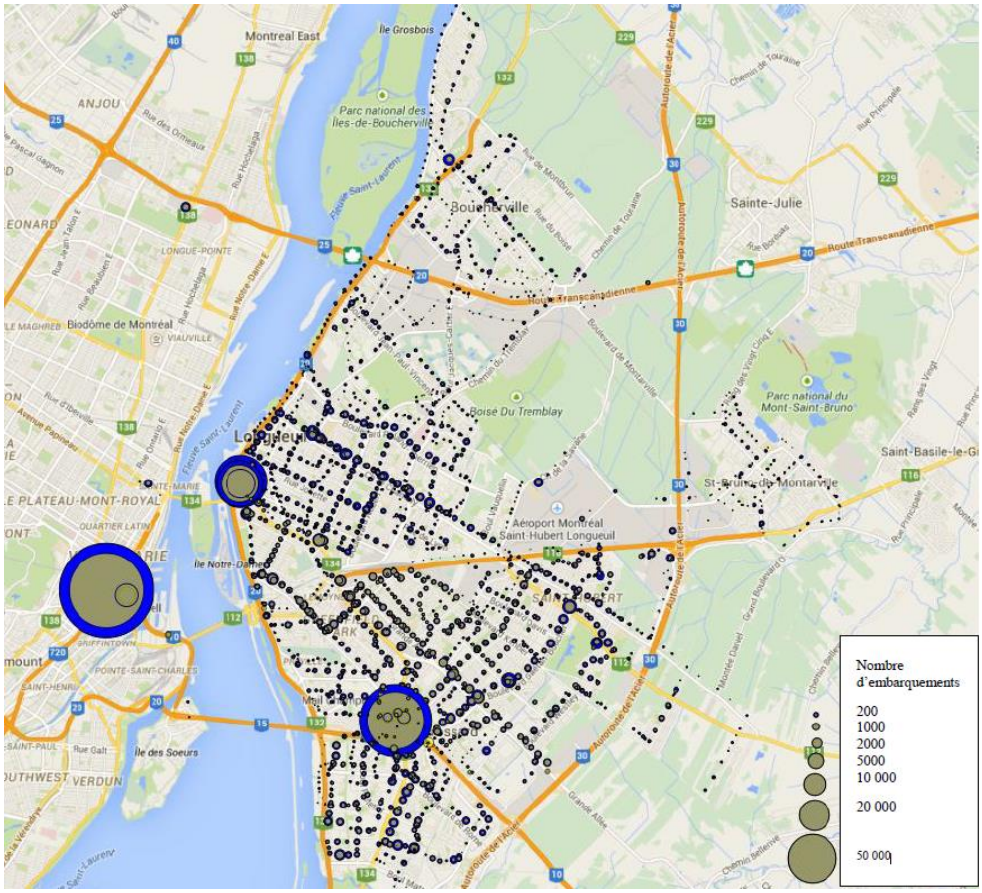


Figure 2-3 - Carte comparative des montées capturées par le SDAP (jaune) et des montées avec une localisation retrouvée après les trois étapes d'enrichissement – tiré de (Légaré, 2014)

2.3 Modèles pour estimer la destination

Une fois les localisations de montées estimées, il faut maintenant essayer de déterminer les points de descente. Pour ce faire, un algorithme a été développé par He au cours de sa maîtrise recherche à Polytechnique Montréal (He, 2014). Plusieurs travaux ont été réalisés auparavant sur les méthodes permettant d'estimer les destinations liées aux transactions. Une des méthodes sur lesquelles se base He est celle développée par (Trépanier, Tranchant, & Chapleau, 2007) suite au mémoire de (Tranchant, 2005). L'hypothèse prise pour retrouver la destination est que l'on suppose que la personne va reprendre le transport en commun à l'arrêt le plus près de l'arrêt où elle en est descendue. On se base alors sur la distance en mètre la plus courte entre les arrêts restants de la ligne prise et l'arrêt du trajet suivant. D'autres travaux (Munizaga & Palma, 2012) se sont basés sur la distance « temporelle » entre les arrêts de la ligne prise et l'arrêt du trajet suivant. Enfin, He

a travaillé pour améliorer cet algorithme en traitant les trajets unitaires et en allant regarder en détail dans l'historique de la carte.

Une fois les destinations retrouvées, afin d'avoir une matrice OD complète, il est important de retrouver les correspondances des destinations finales. (Chu & Chapleau, 2008) et (Munizaga & Palma, 2012) ont travaillé sur l'étude des correspondances ainsi que sur l'estimation du temps d'arrivée à l'arrêt. Pour la logique de cet algorithme, il faut remonter aux travaux de (Barry, Newhouser, Rahbee, & Sayeda, 2002) qui ont travaillé sur le réseau de New York en se basant sur deux hypothèses : qu'une fois arrivés à destination, les utilisateurs retourneront à cette dernière station après leur activité, et qu'en fin de journée, les utilisateurs retourneront à la première station de la journée.

2.3.1 Modèle d'estimation des destinations

Dans le modèle d'estimation des destinations, l'hypothèse courante prise est que les utilisateurs du réseau de transport ne vont normalement pas parcourir de longues distances pour rejoindre la station (arrêt) suivante. Ils devraient de plus choisir l'arrêt le plus proche de leur sortie pour engager leur déplacement suivant.

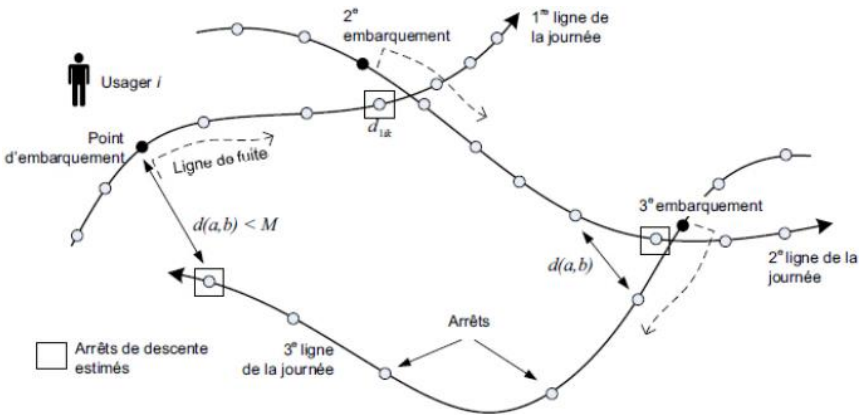


Figure 2-4 - Modèle d'estimation des destinations par une minimisation de la distance entre la station d'arrivée et celle suivante - tiré de (He, 2014)

Ainsi, lorsqu'un utilisateur monte dans le bus dans une certaine direction, on connaît les arrêts restants sur la ligne. Ils correspondent à la ligne de fuite. On suppose alors que l'arrêt de descente est celui parmi ceux de cette ligne de fuite qui se trouve le plus proche à vol d'oiseau de l'arrêt de montée de la transaction suivante. La Figure 2-4 représente ces lignes de fuite, montées et descentes estimées.

Dans la littérature, (Munizaga & Palma, 2012) travaillent aussi sur un algorithme pour retrouver la destination qui s'appuie sur celui de (Trépanier, Tranchant, & Chapleau, 2007). Cependant les auteurs montrent que se baser uniquement sur la distance la plus proche peut masquer des lignes à boucles. La Figure 2-5 le montre.

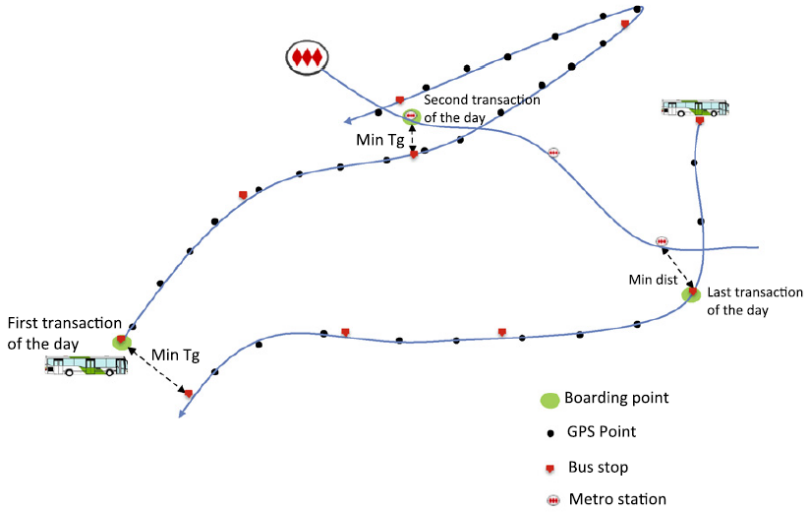


Figure 2-5 - Modèle d'estimation de la destination par une minimisation du temps global - tiré de (Munizaga & Palma, 2012)

En définitive, l'algorithme utilise les possibilités suivantes pour retrouver la destination (Trépanier, Tranchant, & Chapleau, 2007) :

- Séquence de déplacement: la destination est la descente à l'arrêt qui est le plus proche de l'arrêt d'embarquement de la prochaine transaction
- Retour au domicile: on regarde si un arrêt de la ligne peut être assez proche de l'origine du premier déplacement de la journée
- Déplacement du prochain jour: Si notre déplacement est le dernier de la journée, on essaie de trouver une destination avec le premier déplacement du jour suivant.
- Déplacements unitaires: si on échoue dans les cas précédents, on essaie de regarder dans l'historique s'il n'y a pas une transaction dans le jour précédant ou suivant qui est sur la même ligne dans le même sens. Si c'est le cas, on prendra la transaction de l'historique la plus proche temporellement de celle en cours et on lui affectera la même destination que celle définie dans l'historique.

(He, 2014) attribuera respectivement aux quatre cas précédents, dans son algorithme, les codes Origine Destination 11, 12, 13 et enfin 21 ou 22 pour les déplacements unitaires.

2.3.2 Modèle d'estimation des déplacements unitaires

Le travail du mémoire de (He, 2014) a été de développer un algorithme pour déterminer les destinations des déplacements avec une attention particulière sur l'amélioration du traitement des déplacements unitaires. Pour ce faire, l'auteur a utilisé l'historique complet (sur 1 mois) de chaque carte. (Trépanier, Tranchant, & Chapleau, 2007), choisissaient eux forcément une même transaction sur la même ligne pour inférer sa destination. He a proposé la prise en compte du modèle d'activité en cherchant parmi toutes les destinations proches des arrêts potentiels de la ligne. On choisit alors comme destination l'arrêt avec la plus grande probabilité de destination. À cela, He a rajouté une vérification supplémentaire sur la probabilité temporelle que cette destination soit la bonne.

La Figure 2-6 présente graphiquement la logique pour rechercher dans l'historique des déplacements effectués par l'utilisateur. Il faut noter que pour le moment, afin de déterminer les descentes, on ne s'intéresse qu'aux destinations déjà reconstituées.

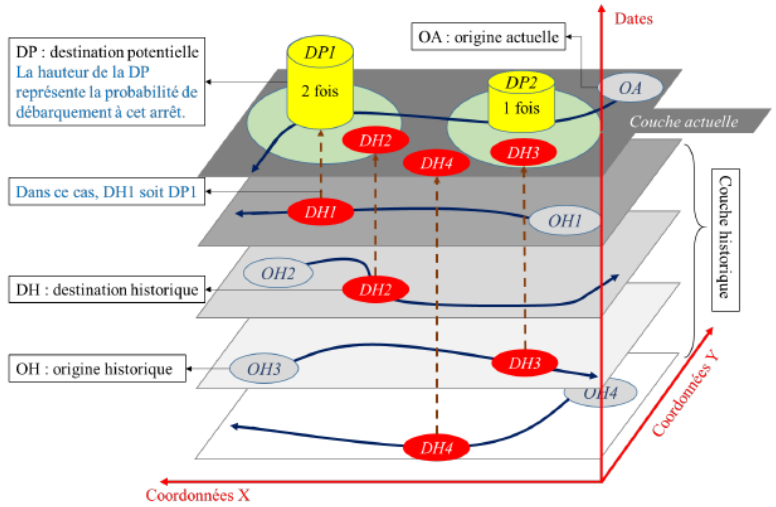


Figure 2-6 - Illustration de la recherche dans l'historique des destinations possibles pour les arrêts de la ligne de fuite prise - tiré de (He, 2014)

Pour ne pas rester uniquement sur une détermination spatiale de la destination, He a aussi travaillé sur une validation supplémentaire temporelle. Pour ce faire, il a utilisé la méthode d'estimation par

noyau qui est présenté dans la Figure 2-7. Cette vérification permet de favoriser une destination qui ressemblerait temporellement mieux au déplacement effectué présentement.

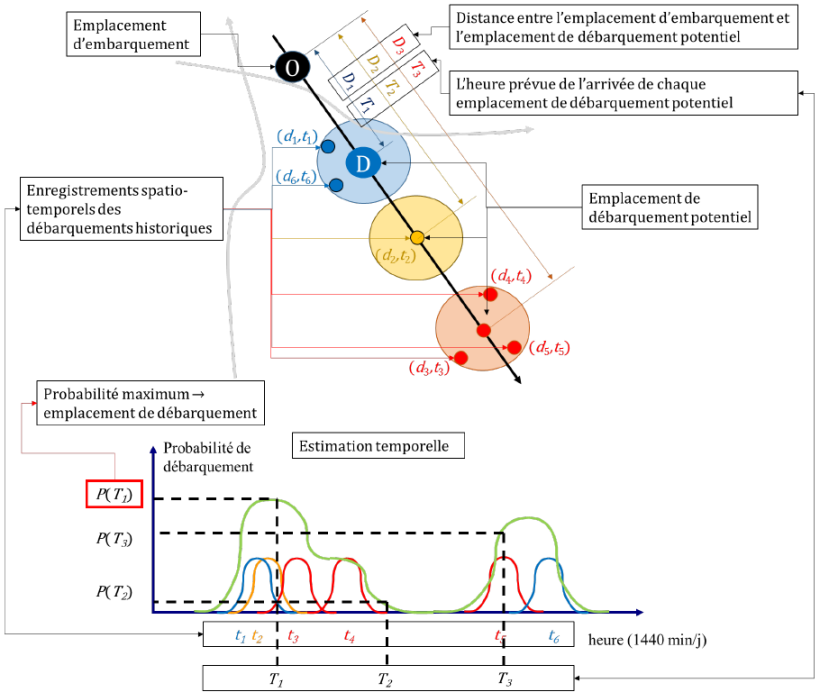


Figure 2-7 - Illustration de l'estimation par noyau pour le traitement temporel d'un déplacement unitaire - tiré de (He, 2014)

À titre d'exemple de données disponibles, la STO a fourni les enregistrements pour le mois d'octobre 2009 qui comportaient 903 333 transactions. Les autres données de travail disponibles étaient les localisations des arrêts ainsi que les différentes lignes avec leurs directions et enfin les lignes-arrêts qui sont les enchaînements des arrêts qui composent les différentes lignes avec la distance de chaque arrêt avec le début de la ligne (He & Trépanier, 2015). Il faut noter que He n'a pas utilisé d'horaire planifié. Ceci implique qu'il a dû utiliser une vitesse moyenne pour déterminer la durée passée dans le transport car il n'a pas recalculé les horaires de passage des bus aux arrêts comme l'ont fait (Chu & Chapleau, 2008).

2.3.3 Calibration de l'algorithme destination

À la suite de son mémoire, He a travaillé avec l'Université de Queensland (He, Nassir, Trépanier, & Hickman, 2015) en Australie qui dispose de données d'exploitation de transport en commun où les utilisateurs doivent valider à l'entrée et à la sortie (« tap-in/tap-out »). L'idée était d'appliquer l'algorithme sur les données pour regarder la précision de ce dernier. Il s'avère qu'avec les données

australiennes, l'algorithme de Li obtient une précision de 79% avec une tolérance de 400 mètres. Dans la première configuration, on a accepté pour les cas 11, 12, 13 – lors de la première phase avec les chaînes de déplacements – une correspondance avec la transaction suivante pour peu que celle-ci se situe à 2000 m autour d'un arrêt de la ligne de fuite. Au cours de la seconde phase pour gérer les déplacements unitaires, il y a une tolérance de 1000 m. Les auteurs, après plusieurs essais, ont conseillé deux tolérances de respectivement 1000 et 250 m pour les deux phases. Ces nouveaux paramètres améliorent la précision de 1,38% ainsi qu'une augmentation de 0,4% de destinations estimées en plus. Ils montrent d'ailleurs qu'il pourrait être intéressant d'adapter cette tolérance en fonction de chaque phase, mais n'ayant pas de caractéristiques sociodémographiques rien n'a été tenté. Les auteurs précisent de plus que ces coefficients pourraient ne pas avoir le même impact sur le réseau de Gatineau car les utilisateurs pourraient avoir des comportements différents de ceux de Brisbane.

2.3.4 Modèle d'estimation des correspondances

Les travaux réalisés par Chu et Chapleau (Chu & Chapleau, 2008) sur l'identification des correspondances et l'estimation des heures de passages réellement effectuées par le bus viennent à la suite de précédents travaux sur le nettoyage et la correction des données de transactions et tracés GPS des bus (Chu & Chapleau, 2007). Il faut noter que pour déterminer les correspondances, Chu et Chapleau ont déjà appliqué un algorithme pour retrouver la destination (Trépanier, Tranchant, & Chapleau, 2007).

Pourquoi s'intéresser aux correspondances effectuées réellement par les passagers quand on connaît déjà la destination d'un voyage ? Chu et Chapleau démontrent que la notion de correspondance tarifaire surestime de 40% le nombre réel de chaînes de trajets. En effet, dans une fenêtre de 60 minutes, certaines personnes ont le temps d'emprunter le transport en commun pour aller faire une activité et de revenir également en transport en commun avant la fin des 60 minutes réglementaires pour une correspondance.

Chu et Chapleau montrent ainsi que seul le critère de fenêtre de temps ne suffit pas pour capturer les correspondances réelles d'une séquence de tronçons qui ne dispose pas d'activités intercalées entre deux voyages. Ainsi, pour mieux déterminer si l'on effectue une correspondance ou non, les auteurs utilisent la destination retrouvée des trajets. (Munizaga & Palma, 2012) ont aussi appliqué des contraintes pour déterminer les correspondances. L'hypothèse prise est essentiellement

temporelle : si une personne reste plus de 30 minutes à un même endroit, il y a une activité. De plus, une transaction ne peut être considérée sur la même ligne.

2.4 Statistiques métier et visualisations

Un réseau de transport en commun est composé de différents objets. La Figure 2-8 tirée des travaux de (Trépanier, Tranchant, & Chapleau, 2007) représente certains objets d'un réseau de transport en commun. Les chiffres ont été adaptés au réseau du RTL et au mois de données disponibles pour ce mémoire (mars 2013). On notera que l'on dispose d'un ensemble de fichiers GTFS pour caractériser l'offre de ce mois, qui possède 404 339 lignes-arrêts-heures pouvant remplacer les 9 047 lignes-arrêts dans l'algorithme de destination et proposer un temps de parcours qui ne se base pas sur une vitesse commerciale moyenne identique pour tout le réseau.

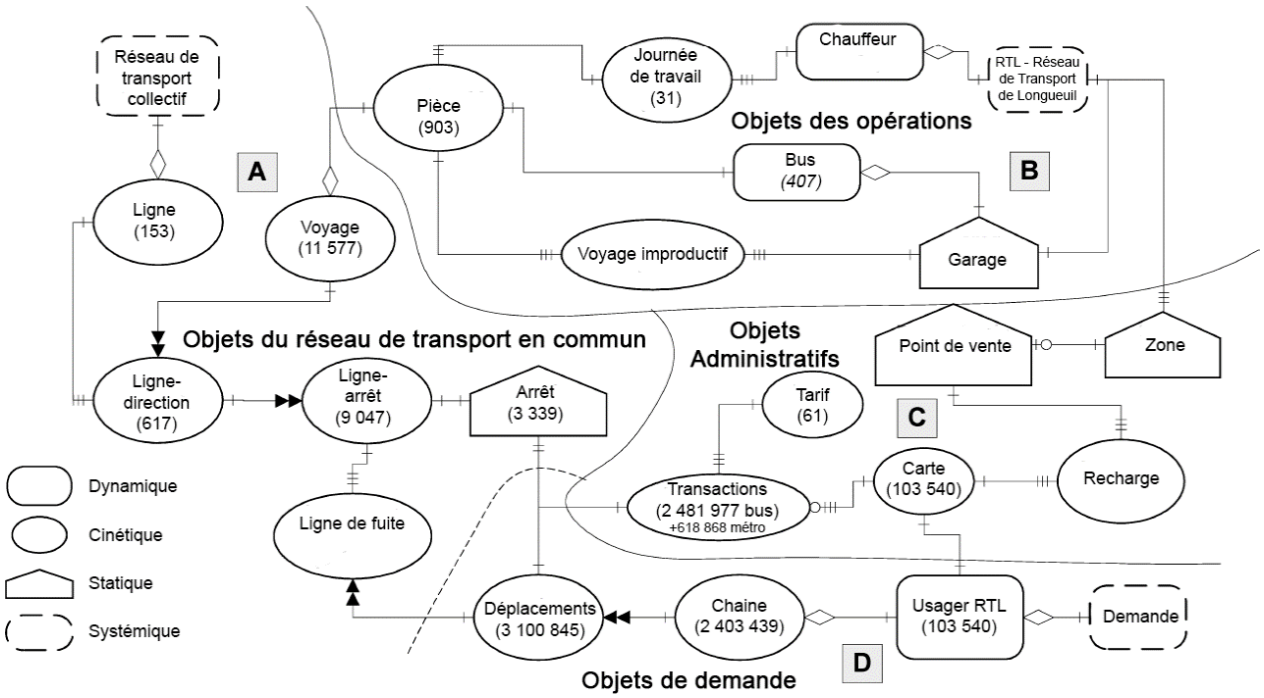


Figure 2-8 - Modèle Objet du Réseau de Transport de Longueuil - inspiré de (Trépanier, Tranchant, & Chapleau, 2007)

Chaque objet dispose de caractéristiques qui lui sont propres. Par ailleurs, le lien entre deux objets peut générer une nouvelle propriété ou indicateur. Différents travaux ont déjà cherché à catégoriser ou à identifier ces indicateurs ou propriétés :

- (Trépanier, Morency, & Agard, 2009) proposent trois catégories de mesures portant sur l'offre (véhicule-kilomètre, véhicule-minute, vitesses moyennes), sur la demande (passager-kilomètre, passager-minute, vitesses moyennes) et sur la performance (taux d'occupation, ponctualité).
- (Vassivière, 2007) insiste sur la diffusion, via un outil web, d'indicateurs et statistiques sur le nombre de passagers montants et descendants, aux arrêts et par ligne, ainsi que sur la reconstitution de la charge à bord des différentes courses.
- (Chu, 2010) propose d'autres indicateurs génériques pour les courses comme le pourcentage de premières montées, le pourcentage de transferts, le temps d'attente moyen par rapport au temps planifié aux montées et descentes ainsi que, de nouveau, les distances cumulées ou moyennes parcourues par les usagers.
- (Lomone, 2014) s'est concentré sur les statistiques propres à un corridor donné en présentant cela pour une ligne donnée. On retrouve la variation de l'horaire d'arrivée aux stations et celle de la durée des haltes aux stations, la différence entre le système de compte à bord et les données de cartes à puce, le nombre de montées et descentes aux arrêts et le cumul de la charge.
- (Tessier, 2015) a de son côté recensé différents indicateurs, cette fois destinés à quantifier la congestion urbaine, en 6 groupes : indicateurs de vitesse, de débit et capacité, spatiaux, de temps et retards, de fiabilité et de coûts.

Ces indicateurs, propriétés ou statistiques peuvent être représentés de différentes manières. On peut utiliser un simple tableau récapitulatif indiquant les chiffres clés. On peut utiliser des graphiques simples ou imbriqués pour représenter différentes distributions. Des graphiques plus poussés permettent de visualiser la combinaison de plusieurs échelles. On retrouve par exemple pour une ligne donnée des profils de charge, des diagrammes espace-temps ou encore des grilles spatio-temporelles. Des tableaux récapitulatifs incluant dans leurs cellules des chiffres-clés et des diagrammes permettent de donner plus d'informations en un simple coup d'œil. Enfin, sachant que ces données sont géolocalisées, la représentation de cartes permet de proposer une vue globale de ces dernières. De plus, si ces cartes sont interactives, avec l'aide d'un outil web par exemple, on peut alors transmettre encore plus d'informations disponibles à l'utilisateur.

2.4.1 Tableaux récapitulatifs

Ces tableaux récapitulatifs sont composés habituellement de chiffres clés. Il est possible de rajouter dans chaque ligne des graphiques plus poussés adaptés au thème du travail de la recherche menée.

Route 44 Direction Ottawa Run	Boarding	% First Boarding	% Transfer Boarding	Adult	Senior	Students over 21	Students 21 or under
5:50	28	82%	18%	28			
6:20	28	89%	11%	26		1	1
6:38	20	70%	30%	17	1		2
6:52	28	100%	0%	23			5
7:10	41	100%	0%	32		3	6
7:24	54	100%	0%	47	1	1	5
7:36	61	100%	0%	53		2	6
7:49	55	98%	2%	39	2	1	13
8:00	37	100%	0%	34			3
8:33	27	100%	0%	22			5
9:00	27	89%	11%	18		3	6

Route 44 Direction Ottawa Run	Boarding at Segment 1	Boarding at Segment 2	Boarding at Segment 3	Boarding at Segment 4	Average delay at boarding (mins)	Average delay at aligting (mins)	Boarding delay outside -2 to 5	Sum of Distance Traveled (km)	Average Distance Traveled (km)	Users using the reverse direction
5:50	15	7		6	-0.6	-1.7		250.8	9.0	4.0
6:20	22	2	1	3	0.0	-2.0		292.5	10.4	7.0
6:38	9	4		7	0.5	-2.2		182.9	9.1	4.0
6:52	23	3	2		1.2	-0.1		354.1	12.6	7.0
7:10	36	5			1.2	0.6		561.9	13.7	12.0
7:24	41	8	5		2.9	5.3	4	570.7	10.6	11.0
7:36	30	21	5	5	2.3	6.4	10	547.7	9.0	19.0
7:49	33	17	3	2	0.9	1.9		499.7	9.1	6.0
8:00	13	19	5		-0.1	-1.3		273.9	7.4	6.0
8:33	14	10	2	1	-1.1	-3.4		234.7	8.7	2.0
9:00	12	8	6	1	3.5	3.3	1	265.1	9.8	2.0

Figure 2-9 - Indicateurs de performance pour une ligne donnée - tiré de (Chu, 2010)

La Figure 2-9 tirée des travaux de (Chu, 2010) récapitule différents indicateurs propres aux données de la STO. Celles-ci ont la particularité de disposer de différents types de cartes (Adultes, Sénior, Écoliers, Étudiants). On retrouve de plus l'horaire de passage aux arrêts ainsi que les retards moyens retrouvés grâce à leurs travaux précédents (Chu & Chapleau, 2008).

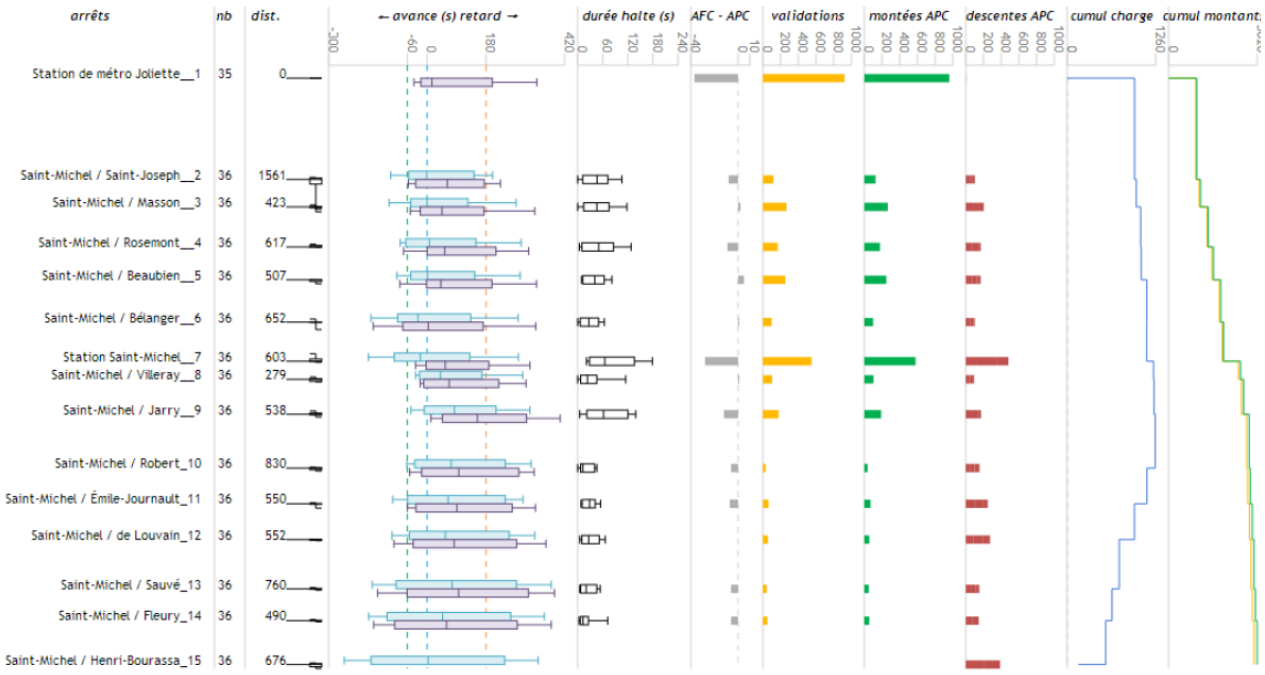


Figure 2-10 - Tableau de bord d'une ligne de la STM dans une direction - tiré de (Lomone, 2014)

À nouveau, en intégrant à un simple tableau des graphiques supplémentaires, on peut transmettre plus d'informations. La Figure 2-10 est un nouvel exemple tiré des travaux de (Lomone, 2014) où l'on affiche des « diagrammes-moustache » sur la variabilité des temps d'arrivée à l'arrêt ainsi que sur la durée des haltes aux arrêts. L'auteur a voulu représenter la distance entre les stations de la ligne par la distance entre les lignes du tableau. Il a aussi placé les montées et descentes à chaque station ainsi que le profil de charge en résultant.

La Figure 2-11 représente un autre tableau récapitulatif issu d'un projet web de visualisation sur le métro de Boston (Barry & Card, 2014). Deux graphiques ou « cartes de chaleur » ont été ajoutés pour représenter le nombre moyen de transactions par heure de la journée, pour les jours de semaine et pour les autres. Cet aperçu est interactif et permet d'extraire la charge moyenne selon l'heure considérée. La carte à gauche offre même une plus grande interaction tout en illustrant la charge des stations du réseau. Quand on se positionne sur une station donnée selon l'heure considérée et le jour de semaine, une carte de chaleur visualise les montées et sorties correspondantes.

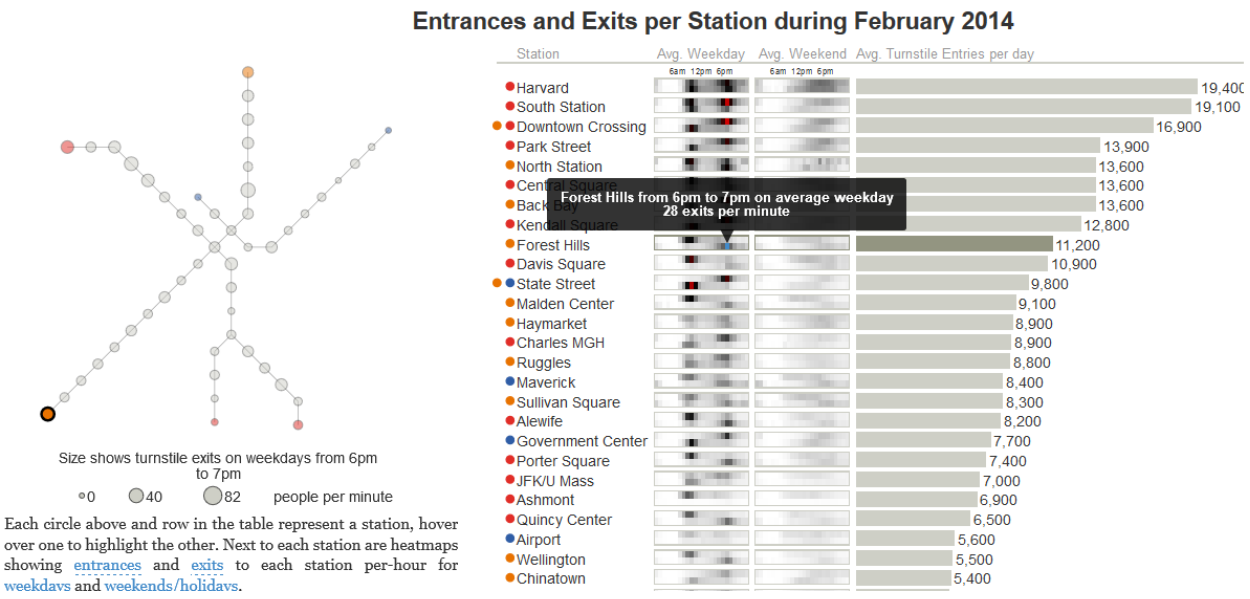


Figure 2-11 - Tableau récapitulatif des entrées et sorties aux stations du métro de Boston - tiré de (Barry & Card, 2014)

2.4.2 Graphiques simples et avancés

Les graphiques usuels simples se présentent sous forme de graphiques avec des courbes ou des barres qui permettent de visualiser, en choisissant les axes des x ou y, les données que l'on souhaite visualiser. On peut ainsi représenter la charge par jour, par heure, par segment d'une ligne, ou encore la distribution des retards, des distances des déplacements des usagers. La Figure 2-12 représente par exemple la variation de l'horaire d'arrivée aux stations par rapport au temps planifié (avances et retards).

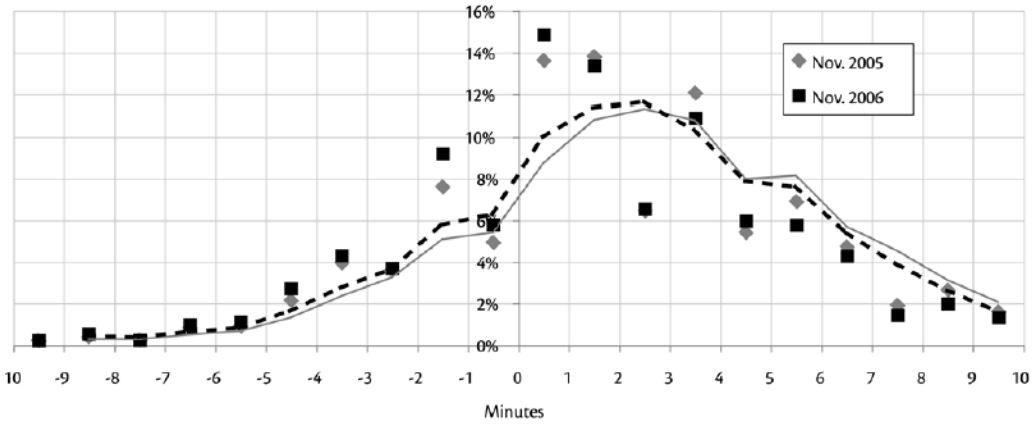


Figure 2-12 - Variation de l'horaire d'arrivée aux stations par rapport au planifié – tiré de (Trépanier, Morency, & Agard, 2009)

2.4.2.1 Profil de charge

Une forme plus avancée de représentation adaptée aux besoins du transport en commun est le profil de charge. Celui-ci peut se décliner de différentes manières. Il peut être reconstitué en utilisant les données du système de comptage automatique à bord qui utilise des capteurs, ou bien en se basant sur les origines et destinations issues des données de cartes à puce. La Figure 2-13 représente la charge à bord d'une ligne ainsi que les déplacements sur la ligne par ses usagers.

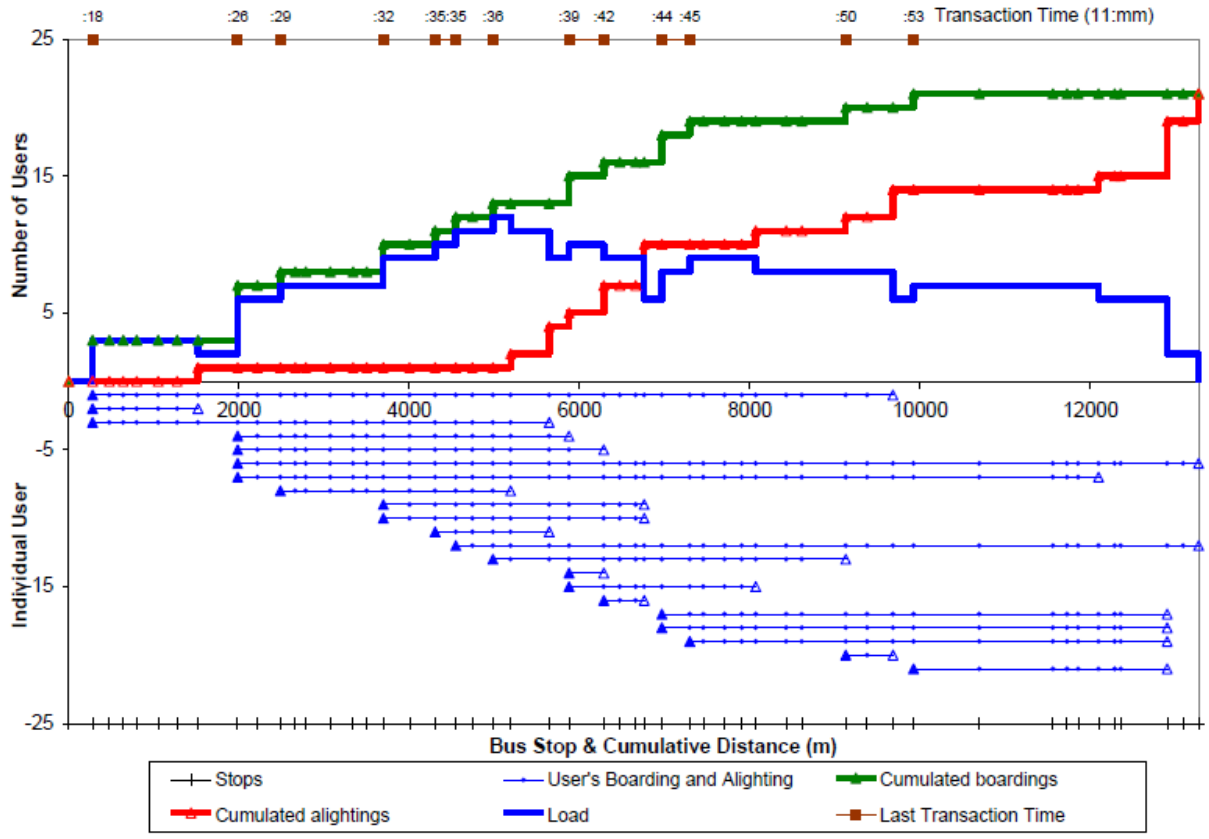


Figure 2-13 - Profil de charge aux niveaux agrégés et désagrégés - tiré de (Chu & Chapleau, 2008)

La Figure 2-14 est un autre exemple de profil de charge qui se propose de comparer deux périodes de temps à la fois.

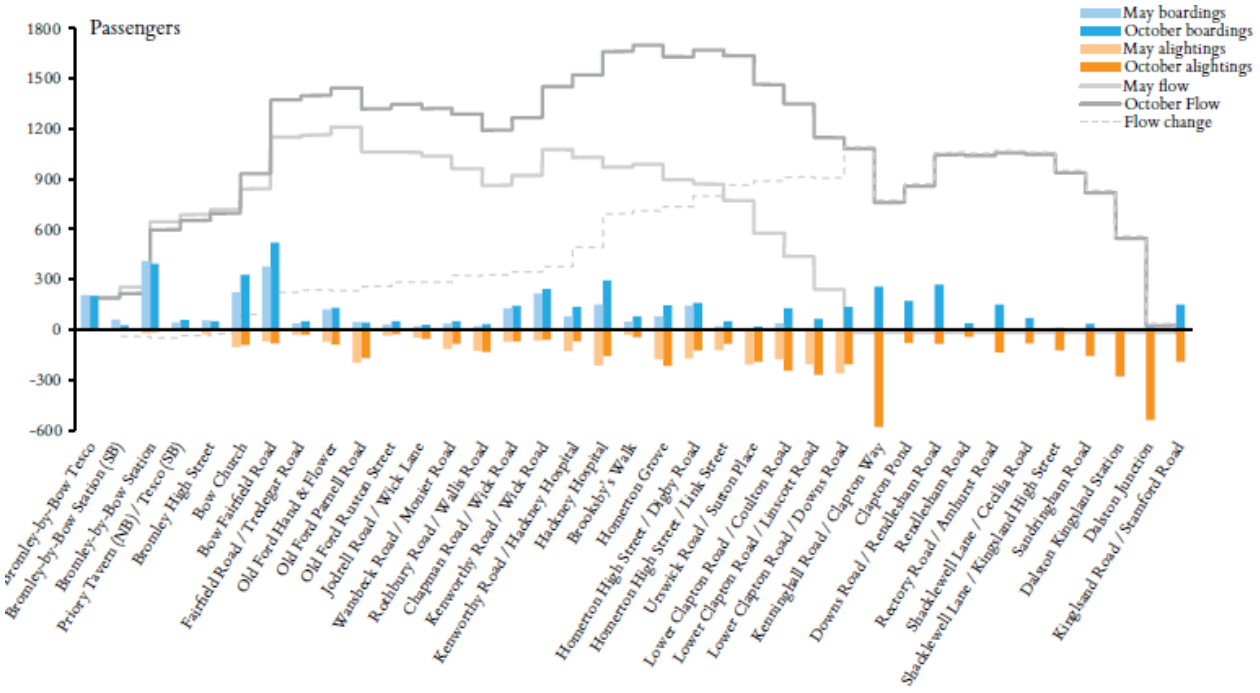


Figure 2-14 - Profil de charge et montées descentes d'une ligne du réseau de transport londonien avec une comparaison entre deux périodes – tiré de (Gordon, 2012)

Des applications permettant de choisir différents tracés de ligne en fonction du jour, de l'heure, de la direction ont été réalisées dès les années 2000. (Tranchant, 2005) a réalisé une telle application, à la Figure 2-15, à l'aide du tableur Excel où ce dernier présente les profil de charge et diagramme espace-temps d'un tracé donné.

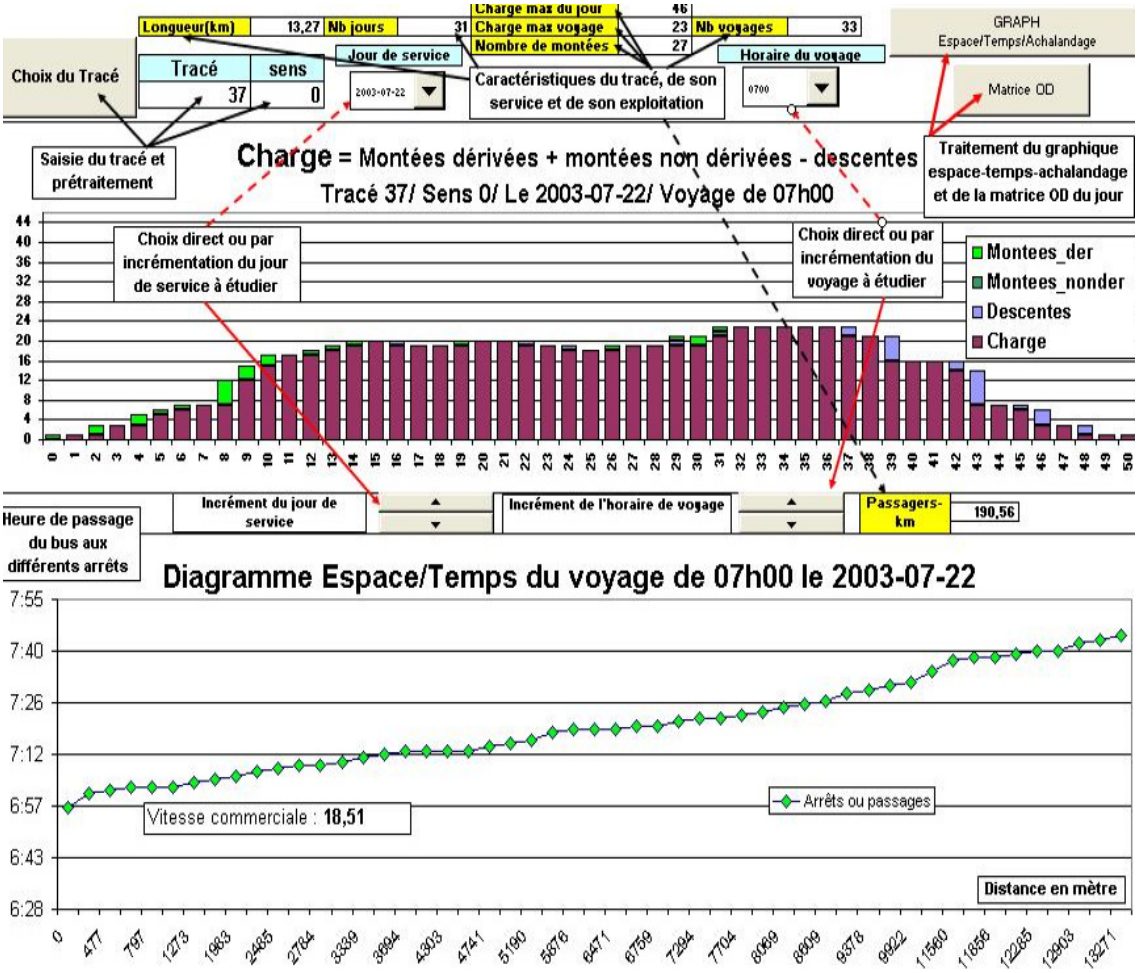


Figure 2-15 - Outil de visualisation et d'analyse du service effectif – tiré de (Tranchant, 2005)

(Vassivière, 2007) a lui réalisé un outil web, à la Figure 2-16, pour présenter un tracé et la charge sur celui-ci afin de rendre ces informations facilement disponibles aux planificateurs ou autres personnes intéressées. Il a utilisé la technologie SVG (Scalable Vector Graphic) pour la réalisation des graphiques et celle du XML (eXtensible Markup Language) pour diffuser les statistiques.

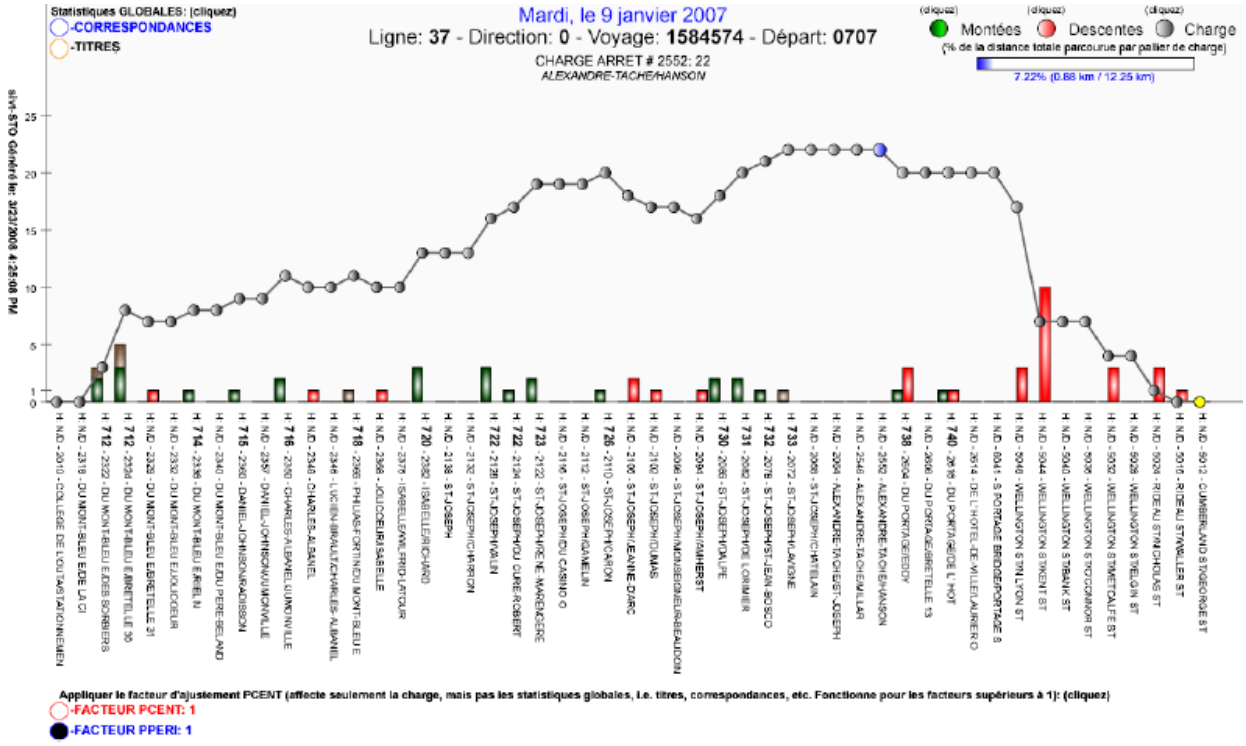


Figure 2-16 - Application web analyse profil de charge – tiré de (Vassivière, 2007)

La Figure 2-17 issue de (Barry & Card, 2014) constitue un autre exemple récent de visualisation du profil de charge avec la possibilité ici de le représenter sur plusieurs jours tout en montrant également une échelle des retards aux différents moments de la journée. Cette visualisation est interactive alors qu'avec le survol des profils de charge, la carte à gauche se met à jour pour montrer la charge et la congestion sur le réseau à l'heure et au jour voulu.

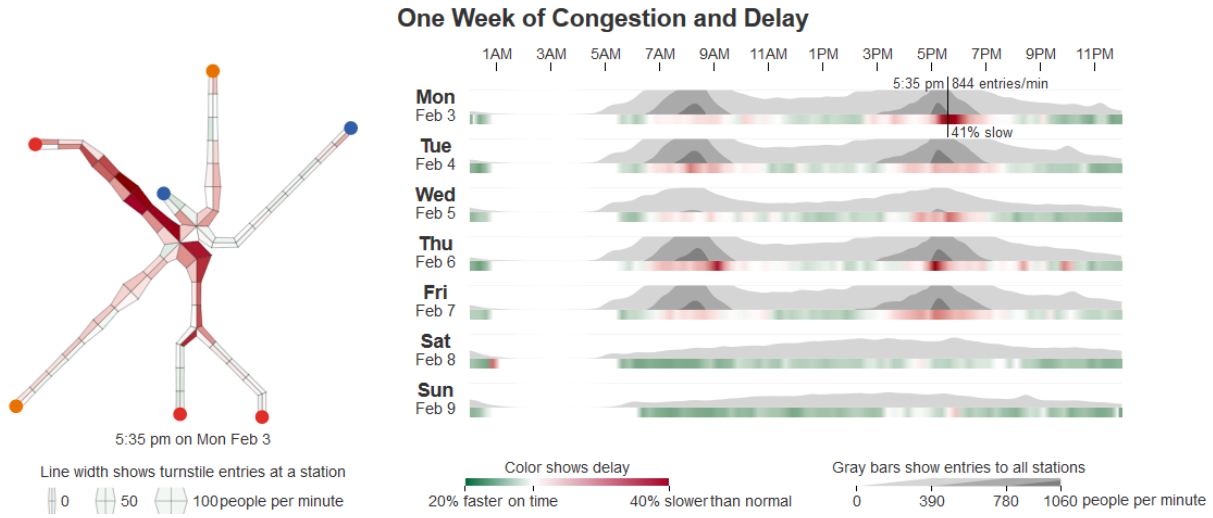


Figure 2-17 - Profil de charge pour différents jours de la semaine - tiré de (Barry & Card, 2014)

2.4.2.2 Diagramme espace-temps

Le diagramme espace-temps est un outil servant à visualiser les différentes courses d'une ligne de transport en commun prévues ou effectuées sur la journée. Habituellement, celui-ci ressemble à la Figure 2-18 qui représente ici en même temps la charge aux différents arrêts. Il est possible d'intervertir les axes.

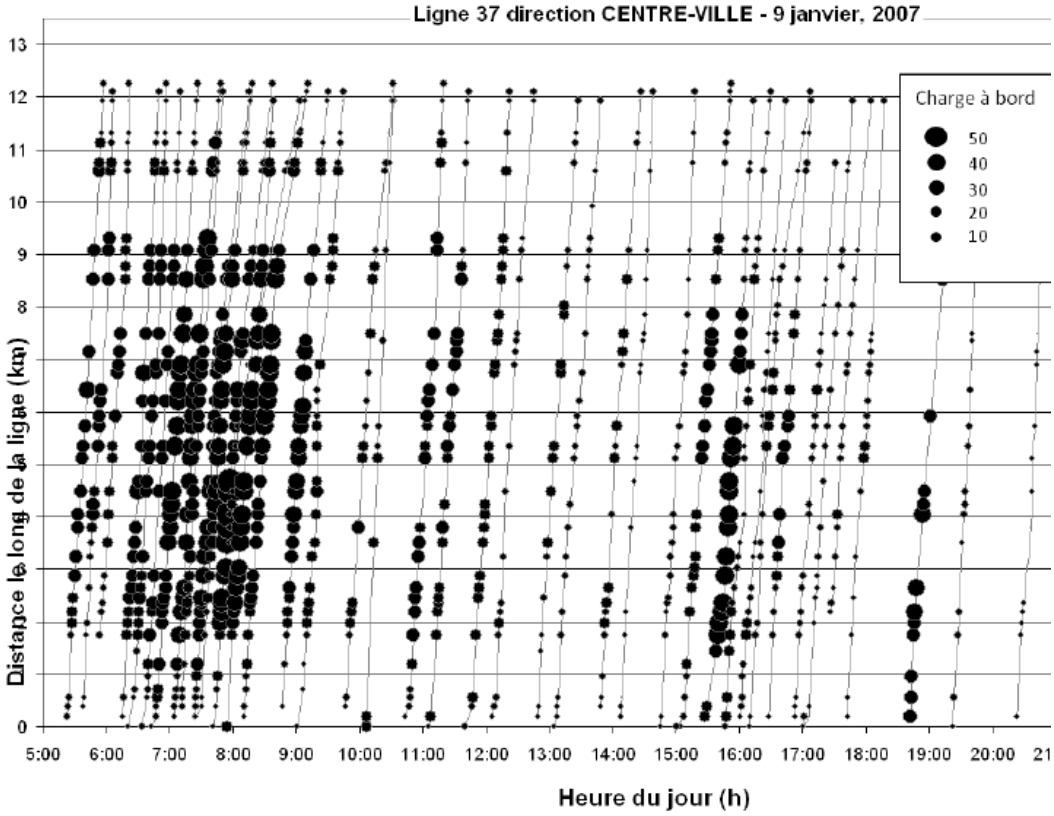


Figure 2-18 - Diagramme espace-temps avec charge – tiré de (Vassivière, 2007)

La Figure 2-19 représente un autre exemple de visualisation interactive d'un diagramme espace-temps appliqué à une ligne de transport en commun. Les auteurs ont ici ajouté des indicateurs de performance pour pouvoir connaître les statistiques exactes à un arrêt donné.

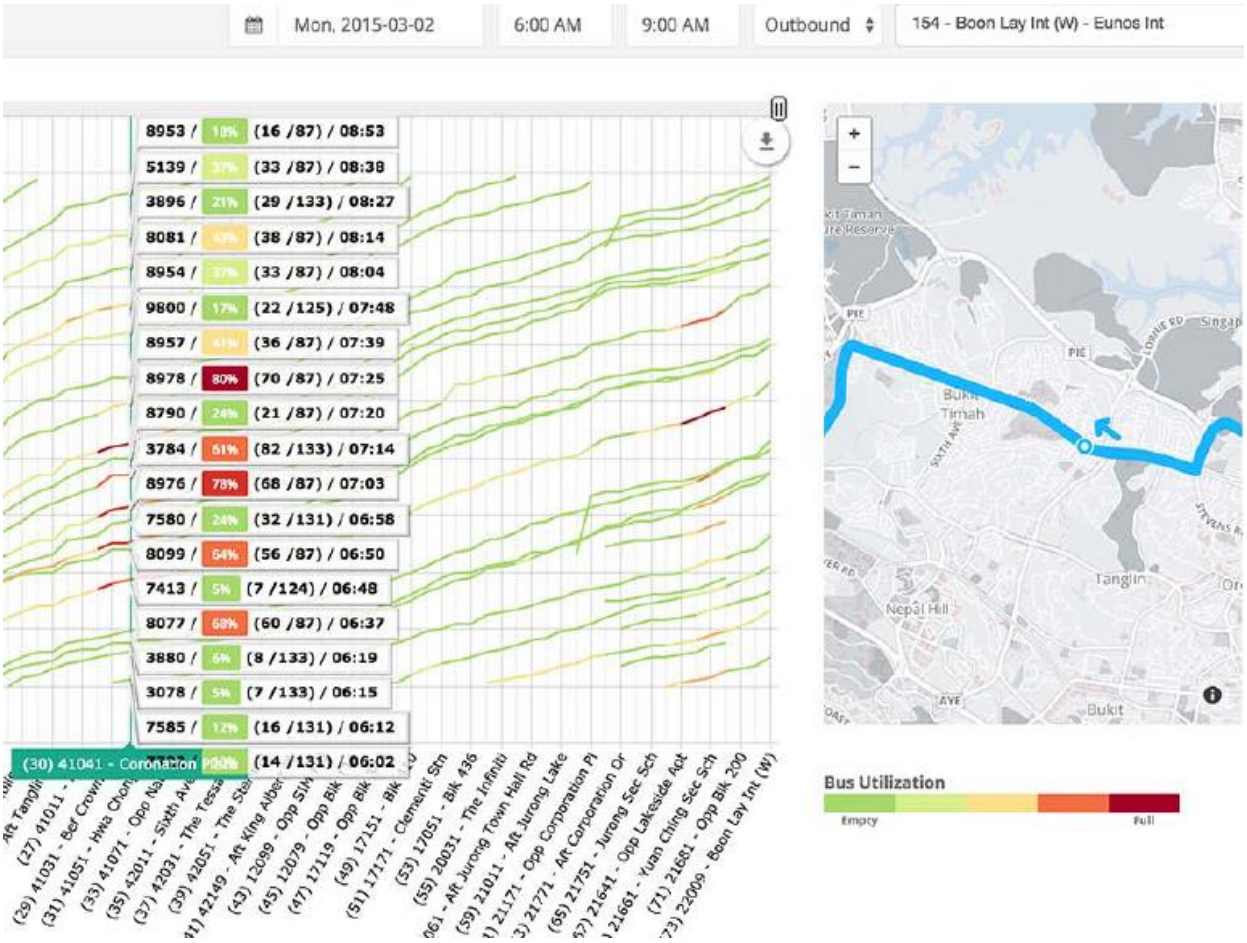


Figure 2-19 - Visualisation interactive d'un diagramme espace-temps – tiré de (Anwar, Odoni, & Toh, 2016)

La Figure 2-20 constitue un exemple poussé et fortement interactif de diagramme espace-temps. Sur la carte à gauche, au survol des heures de la journée, on voit évoluer les différents convois de métros et au survol d'une ligne du diagramme espace-temps, le train concerné est même mis en évidence sur la carte à gauche.

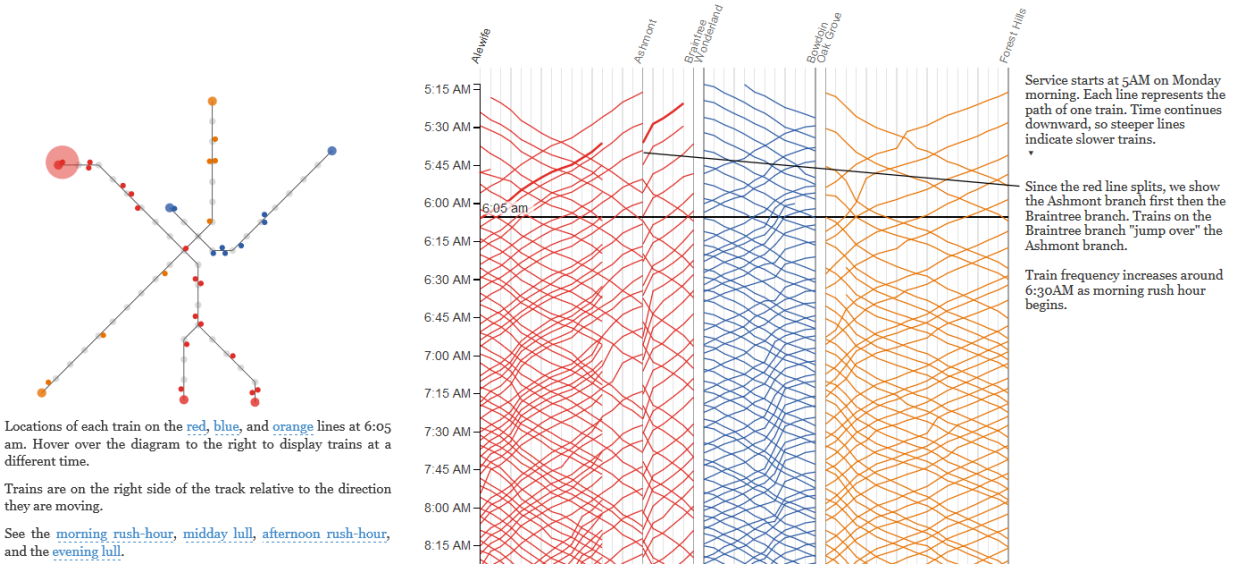


Figure 2-20 - Diagramme espace-temps des courses des lignes du métro de Boston - tiré de (Barry & Card, 2014)

La Figure 2-21 propose de compléter le diagramme espace-temps précédant en ramenant toutes ces courses de voitures de métro dans un même graphique pour observer la distribution des différentes courses. Il est possible de choisir une plage horaire de la journée et de visualiser les différentes courses correspondantes. Cela permet de voir les trajets les plus rapides et les plus lents du réseau. Il est aussi possible d'interagir avec chaque course en visualisant ces données.

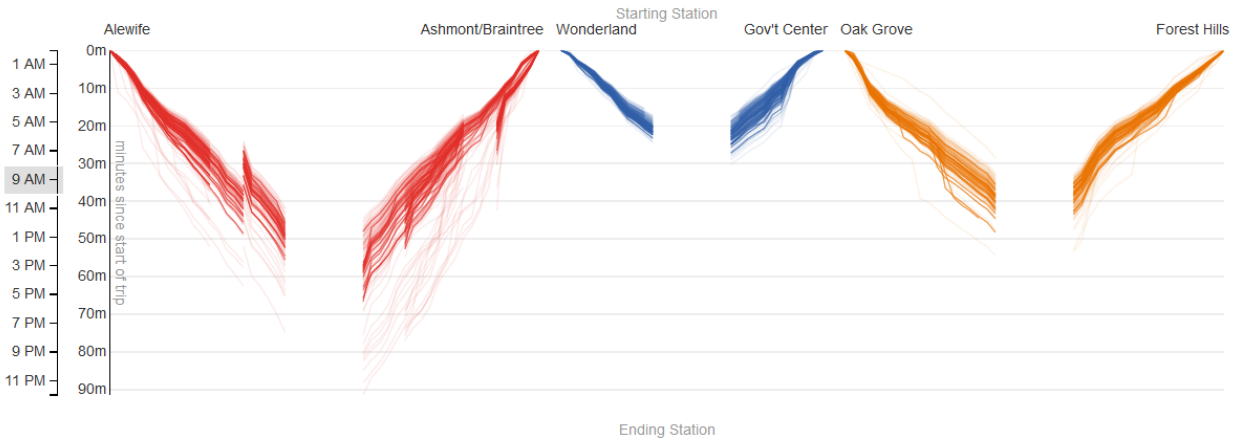


Figure 2-21 - Récapitulatif des courses du métro de Boston superposées dans un même graphique - tiré de (Barry & Card, 2014)

Une évolution possible du diagramme espace-temps consiste à lui ajouter le profil de charge sur les tronçons avec le cas échéant le nombre de montées aux arrêts. Les Figure 2-22 et Figure 2-23 en sont des exemples.

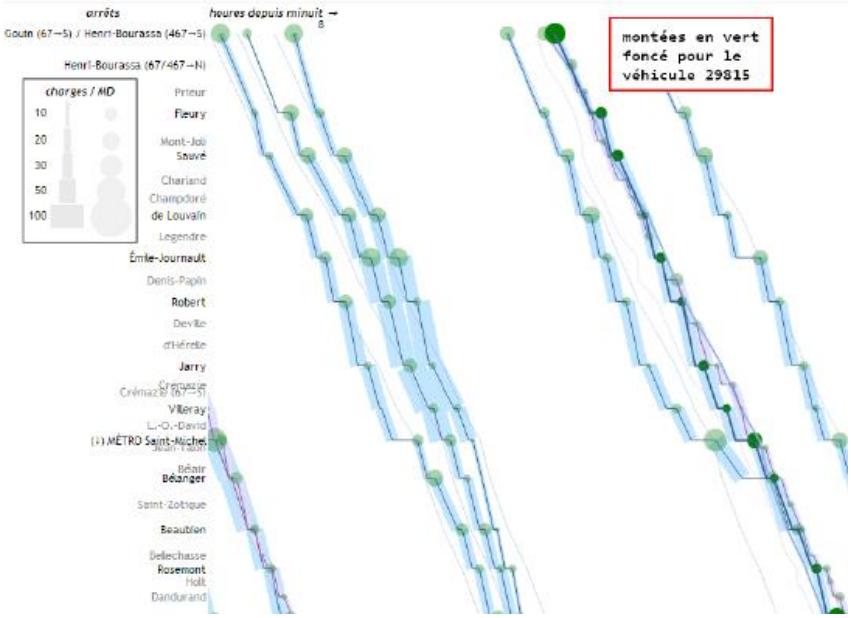


Figure 2-22 - Diagramme espace-temps en 2D avec charge à bord et montées – tiré de (Lomone, 2014)

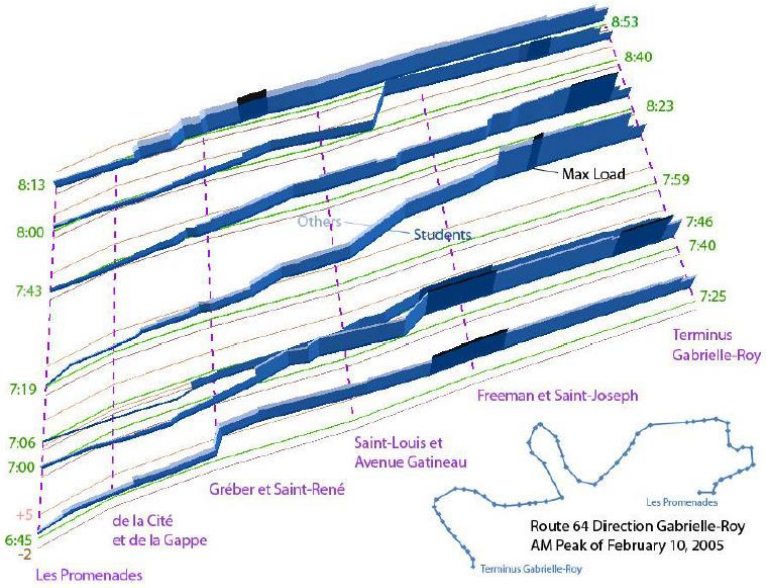


Figure 2-23 - Diagramme espace-temps en 3D avec charge à bord – tiré de (Chu, 2010)

2.4.2.3 Grille spatio-temporelle

La Figure 2-24 illustre un autre type de graphique représentant à la fois une échelle temporelle et spatiale afin d'avoir un aperçu de l'indicateur sur une journée donnée. L'exemple ici représenté

permet de visualiser en rouge la congestion sur une autoroute au fil de la journée aux différents endroits du réseau autoroutier de la grande région de Montréal.

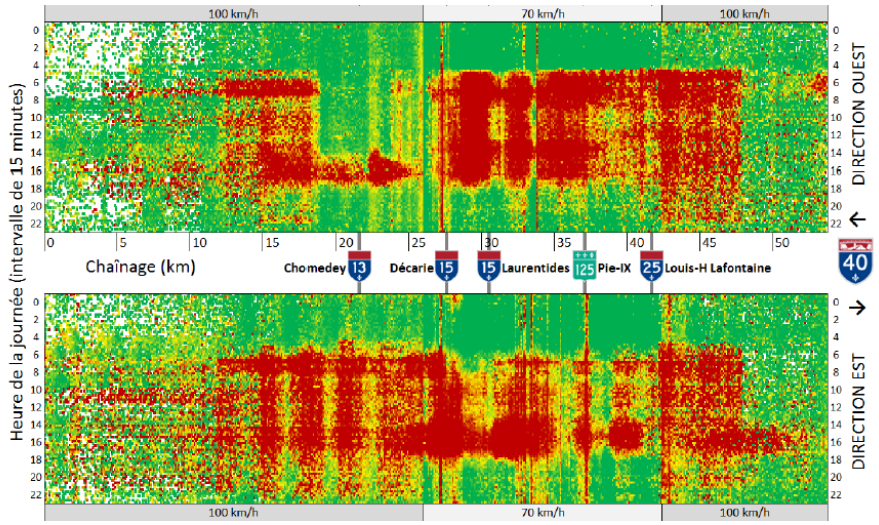


Figure 2-24 - Exemple de grille spatio-temporelle du ratio des vitesses historiques d'une autoroute montréalaise – tiré de (Tessier, 2015)

2.4.1 Géomatique ou visualisations à l'aide d'une carte

2.4.1.1 Vue d'ensemble des arrêts ou des stations du réseau

La Figure 2-25 représente un exemple d'interface de données de type stations de vélo en libre-service. L'auteur (Côme, 2014) propose différents graphiques temporels montrant l'évolution des bornes ou vélos disponibles par exemple sur une période donnée.

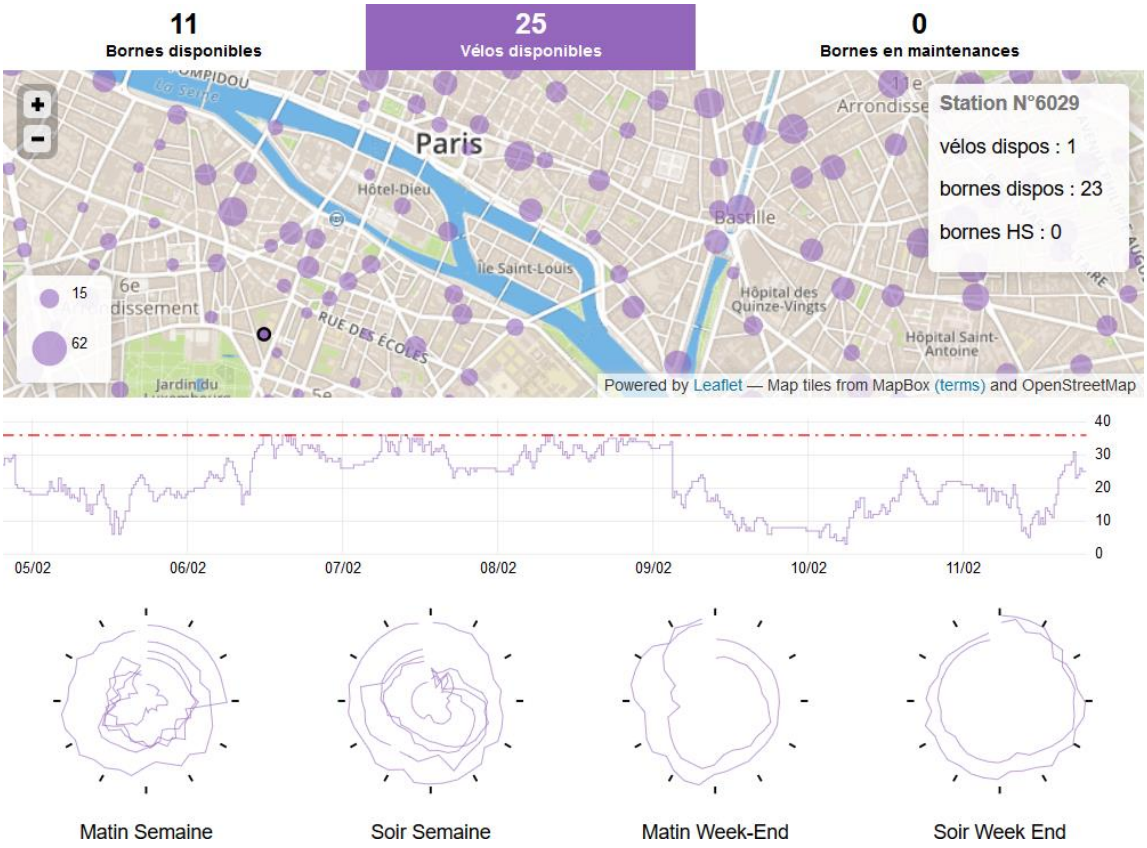


Figure 2-25 - Interface web pour visualiser des statistiques de station de vélo en libre partage - tiré de (Côme, 2014)

Toujours du même auteur (Côme, 2013), la Figure 2-26 permet de visualiser la charge des stations du Vélib à Paris. Au survol d'une station donnée, on peut visualiser le profil de charge des départs et retours en semaine ou bien la fin de semaine. L'auteur a aussi fait apparaître les flux (lignes droites) entre les différentes stations du réseau. L'échelle de couleur représente les résultats d'une étude de classification des stations en fonction de leur utilisation (Côme & Oukhellou, 2012). Les données sont préchargées dans la page, ce qui permet une grande fluidité.

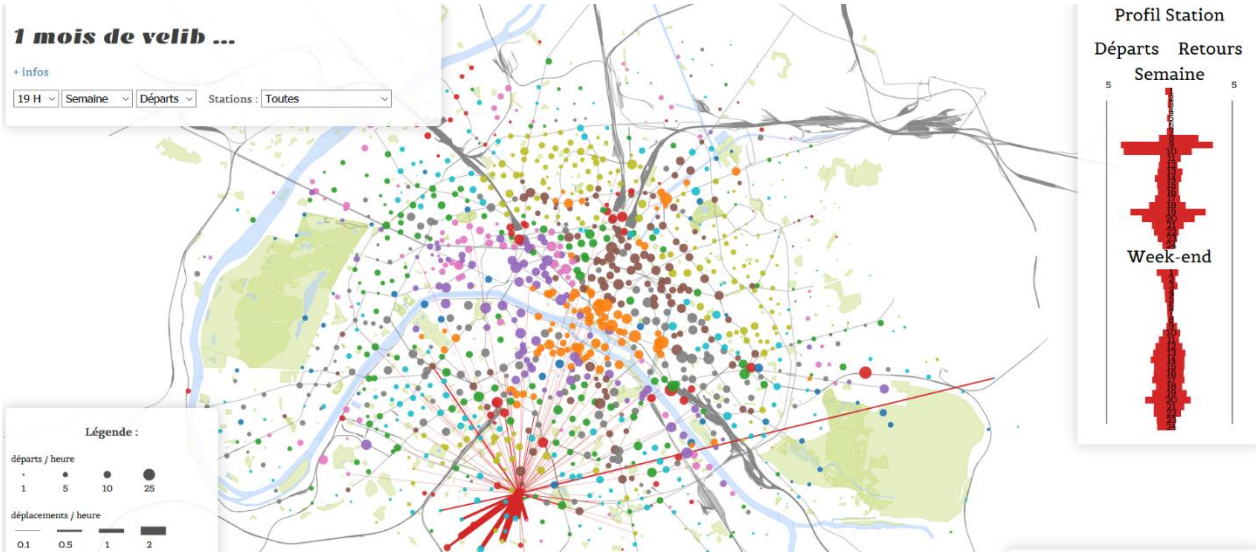


Figure 2-26 - Carte d'ensemble des stations du réseau Vélib' de Paris – tiré de (Côme, 2013)

La Figure 2-27 représente la charge à bord à l'aide d'une échelle de couleur des bus arrivant à une station donnée. Ici, la taille des cercles a été choisie pour représenter si le bus est un modèle à un ou deux étages.

Figure 2-27 - Visualisation de la charge à bord des bus arrivant à une station – tiré de (Anwar, Odoni, & Toh, 2016)

2.4.1.2 Visualisation du profil géospatial d'une carte à puce

À l'aide de données de carte à puce, il est possible d'étudier le comportement d'une carte à puce donnée sur le réseau de transport en commun. La Figure 2-28 permet de visualiser les points d'ancrage et déplacements d'une carte donnée du réseau de la STO étudié par (Chu & Chapleau, 2010). Ceci montre l'utilité des données Origine-Destination retrouvées pour les différentes transactions du réseau. Dans cette même étude, Chu et Chapleau ont aussi sélectionné les cartes à puce se rendant à une école de Gatineau, un point d'intérêt, afin de visualiser d'où viennent les personnes fréquentant cet établissement couramment.

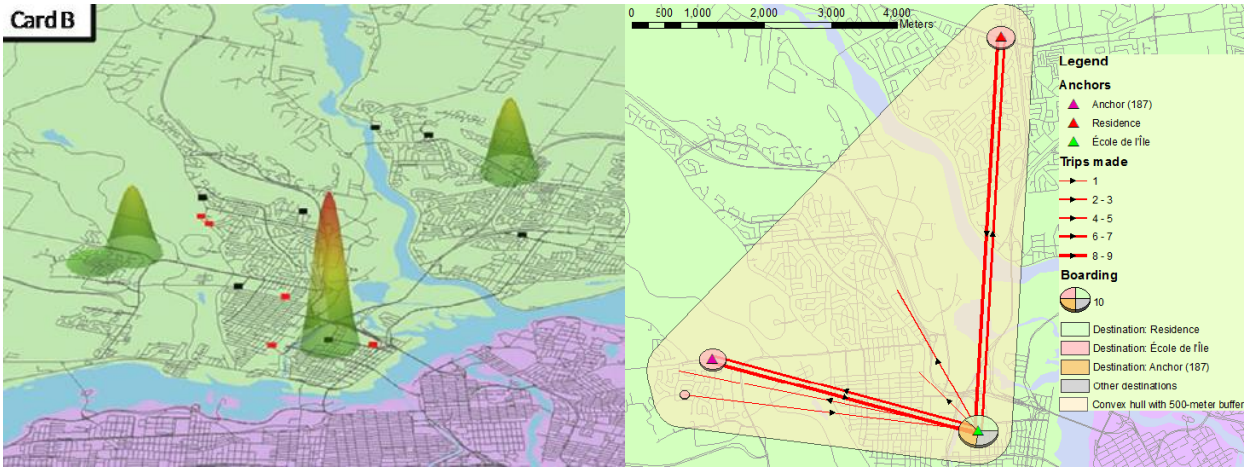


Figure 2-28 - Points d'ancrage et déplacements d'une carte à puce de la STO - tiré de (Chu & Chapleau, 2010)

2.4.1.3 Données OD et vue d'ensemble du réseau

Les données Origine-Destination des transactions permettent de visualiser la charge sur le réseau en fonction des différents moments de la journée. La Figure 2-29 issue des travaux de (Tao, 2015) présente les charges des montées, descentes et transferts d'un côté et de l'autre la charge sur les segments du réseau. L'auteur a aussi ventilé ces visualisations en fonction du type de carte des passagers. Cela permet d'avoir une vue globale de l'état du réseau de transport en commun et de comprendre, de façon macroscopique, la mobilité d'une ville.

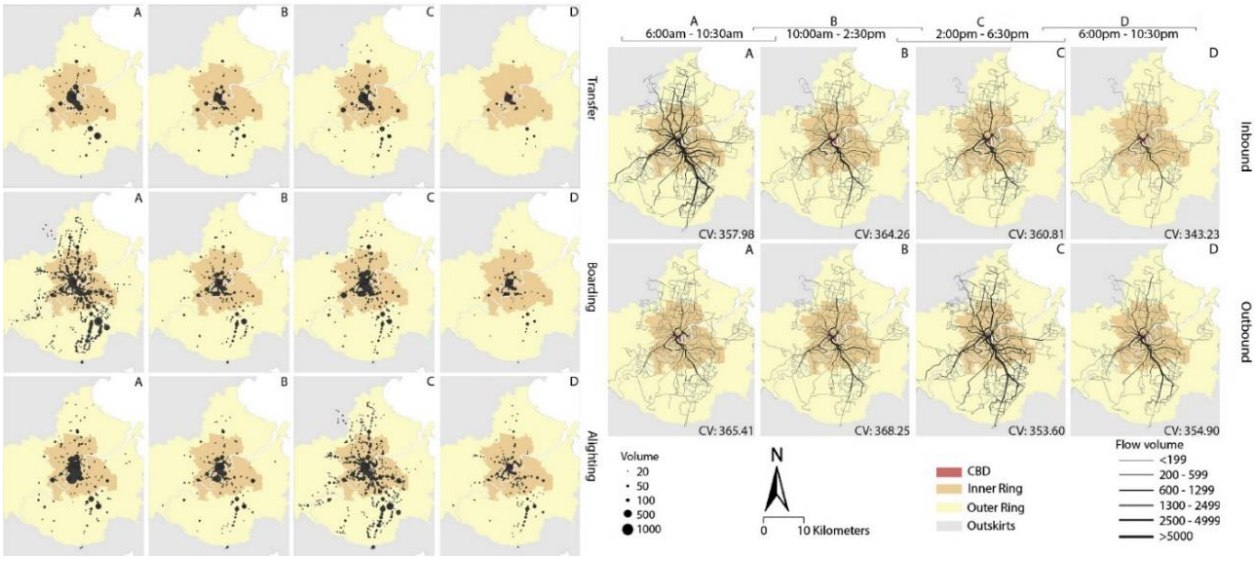


Figure 2-29 - Visualisation des montées/descentes/transferts ainsi que de la charge sur les segments du réseau de transport de Brisbane, Australie – tiré de (Tao, 2015)

2.4.1.4 Données OD pour analyser une ligne de transport en commun

Lorsque l'on dispose de données Origine-Destination, nous avons noté que nous étions capables de reconstruire le profil de charge d'une ligne de transport en commun. Nous avons également noté que nous étions en mesure de retrouver les lieux fréquents associés à une carte à puce. La Figure 2-30 est une autre utilisation de ces données OD qui se propose de visualiser les premières montées des cartes ayant fréquentées une ligne donnée lors d'un jour donné du réseau de transport londonien.

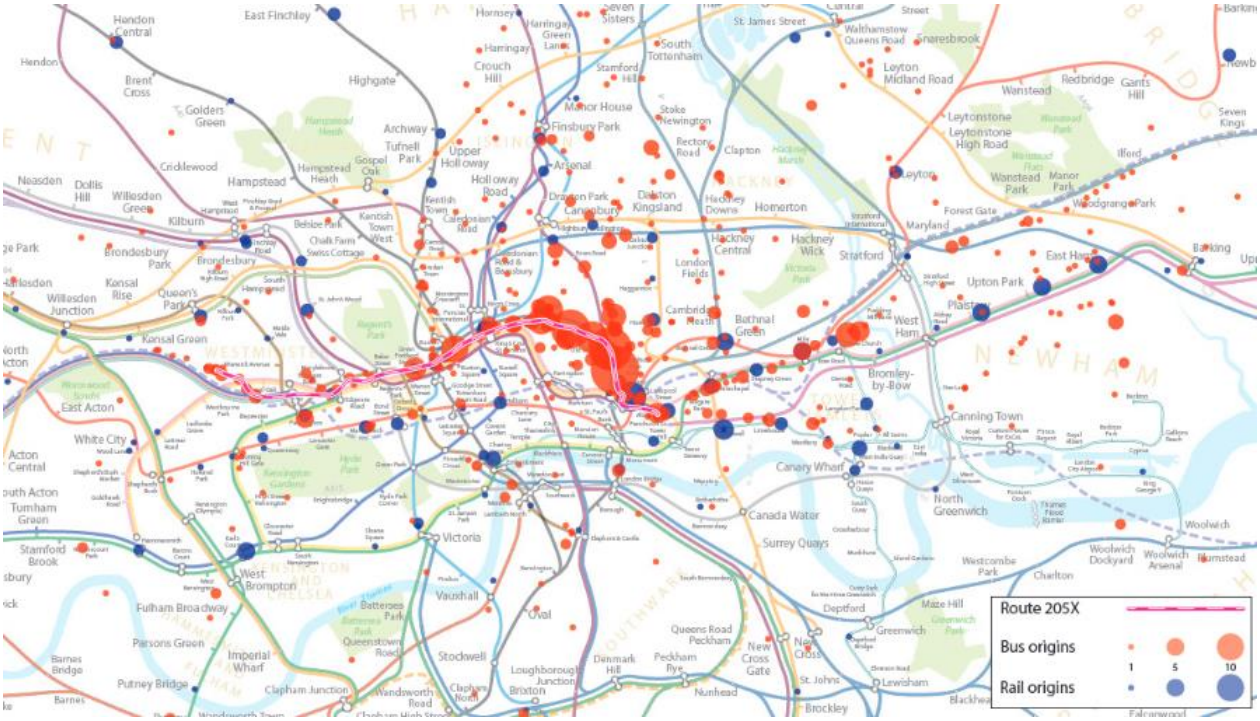


Figure 2-30 - Premières montées du bus et du métro des cartes passant par la ligne 205x du réseau de transport londonien lors d'une journée - tiré de (Gordon, 2012)

2.4.2 Outils et autres solutions commerciales

Il existe d'autres solutions et/ou outils commerciaux qui s'intéressent à ces mêmes questions de visualisation. En voici quelques-unes..

2.4.2.1 Portail Thales Analytics for Transportation

Ce portail est une solution (Gayrard, Naour, & Thales, 2015) développée par Thales et plus précisément par l'équipe du CeNTAI (Centre de Traitement et d'Analyse de l'Information), un laboratoire de recherche et développement sur la gestion et l'analyse de grandes masses de données. Ce portail web se base sur les technologies Elasticsearch et Kibana pour la partie finale de rassemblement des statistiques et présentation de ces dernières. L'équipe du CeNTAI a personnalisé une version de Kibana pour mieux répondre aux besoins de ses clients. Une partie de préparation des données est effectuée à l'aide de Spark. La solution de Thales a été pensée pour pouvoir analyser en temps réel les données de transactions – du Syndicat des Transports d'Île-de-France (STIF), à Paris, dans cet exemple. Cette solution permet aussi l'envoi d'alertes lorsque la demande observée diffère de celle prédite.

2.4.2.2 Kibana et Elasticsearch

Kibana et Elasticsearch sont deux technologies développées par la même compagnie (Elastic). Elasticsearch est une base de données qui est particulièrement adaptée pour faire des agrégations de données à la volée. Elasticsearch peut fonctionner de façon répartie (en ayant différentes instances tournant sur plusieurs machines) lui permettant alors d'être extensible (« scalable ») et de s'adapter à la demande. Kibana est une solution web, basée sur Elasticsearch, simple et intuitive de visualisation et de fouille de données temporelles. Ces deux outils combinés permettent de pouvoir charger de grandes quantités de données dans Elasticsearch d'un côté et les visualiser facilement à l'aide de Kibana de l'autre. Puisque ces outils seront utilisés dans le cadre de ce mémoire, ils seront expliqués plus loin.

2.4.2.3 Solution de vue globale de l'activité d'une ligne

« Touching Bus Rides » développé par (Senseable City Lab, SMART, 2012) propose un outil de visualisation avancé d'une ligne donnée du réseau de Singapour. On retrouve une vue globale des courses et de leurs charges aux arrêts, une vue de cette ligne sur une carte avec les charges ainsi qu'une vue sur les flux entre les arrêts de la ligne. Chaque visualisation peut être mise à jour en temps réel en fonction d'un filtre temporel personnalisable.

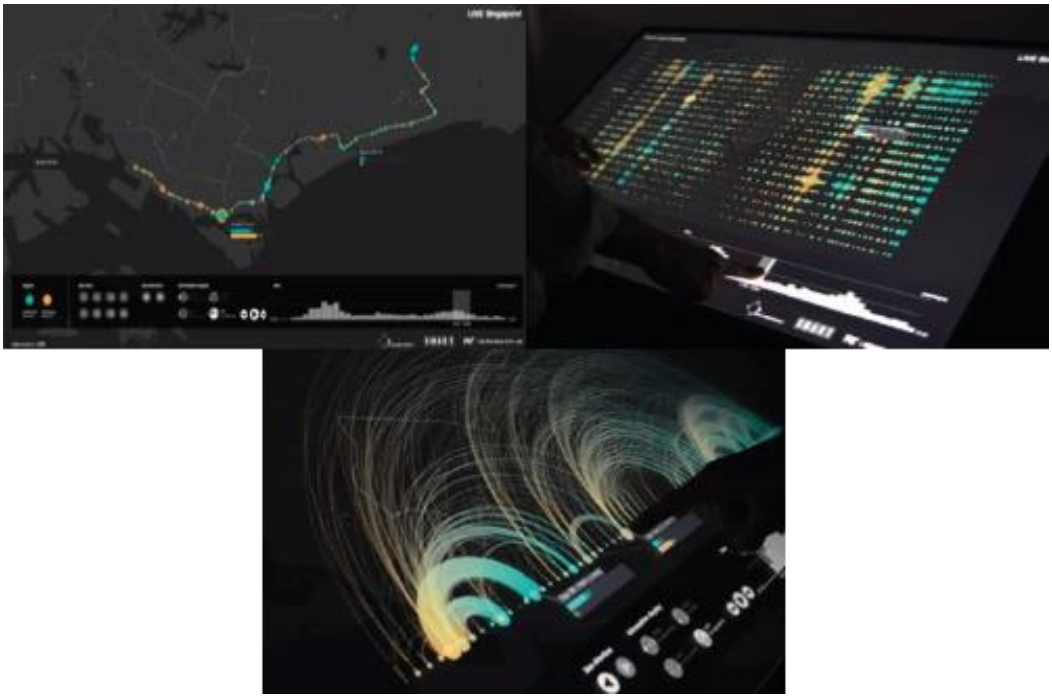


Figure 2-31 - Touching Bus Rides – tiré de (Senseable City Lab, SMART, 2012)

2.4.2.4 Solutions commerciales d'outils pour l'aide à la planification

Il existe plusieurs solutions commerciales d'outils développés spécialement pour les besoins de la planification en transport en commun. Nous présenterons quelques solutions de « start-up » qui nous semblent intéressantes car très à jour technologiquement et esthétiquement parlant.

(Remix) est une solution web qui permet de modéliser à la volée un réseau de transport. Elle offre la possibilité de codifier un réseau en traçant des lignes, choisissant des fréquences et horaires de passage. Remix va afficher en temps réel un coût de mise en œuvre et le faire évoluer en fonction des ajouts et modifications du réseau effectués par l'utilisateur. Si des données de recensement sont disponibles, des indicateurs comme la population, ou le nombre d'emplois desservis seront affichés. Une fois un réseau construit, Remix propose de visualiser à partir d'un point donné toutes les zones accessibles en moins de 30 minutes à partir de ce point. La Figure 2-32 présente les deux solutions évoquées précédemment.

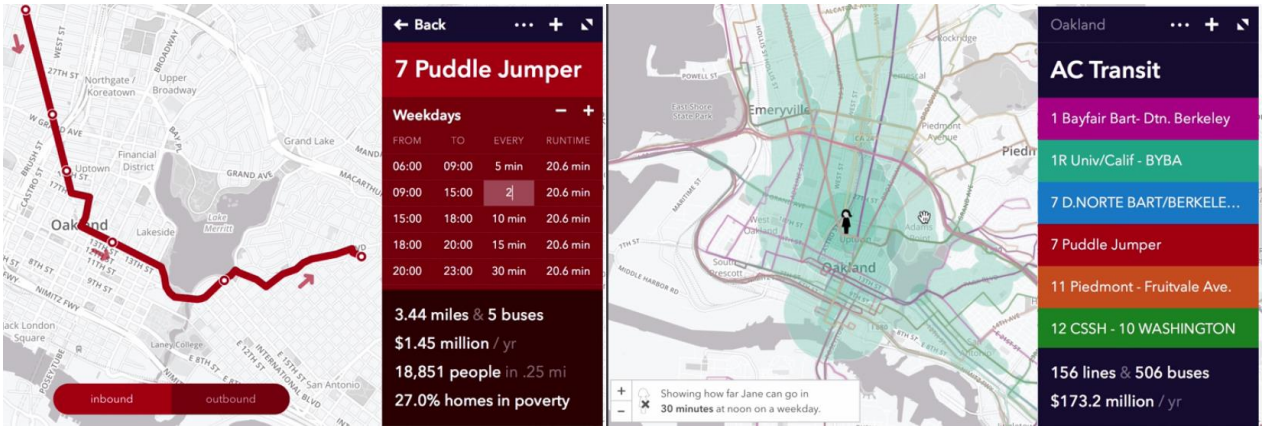


Figure 2-32 - Solutions de codification et visualisation de l'accessibilité – tiré de (Remix)

(Urban Engines) est une autre solution moderne qui utilise la puissance des outils web. Cette start-up propose différents outils de visualisation pour les villes, entreprises ou usagers d'un réseau de transport en commun. La Figure 2-33 présente différentes visualisations interactives permettant de mieux visualiser l'état d'un réseau de transport en commun. On retrouve à gauche une vue d'ensemble d'un réseau de métro qui évolue en fonction du temps avec l'affichage de la charge à bord des trains, le nombre de passagers attendant sur les quais, ainsi que différentes alertes. Une autre visualisation présente les différentes courses d'une ligne donnée de transport avec la visualisation en couleur de la charge à bord. Une dernière visualisation similaire présente en plus

une animation de la charge à bord des bus et l'évolution des montées et descentes aux différents arrêts de la ligne.

Figure 2-33 - Visualisations diverses sur l'état d'un réseau de transport - tiré de (Urban Engines)

Urban Engines présente dans une autre vidéo (Urban Engines, 2016) leur solution pour ajuster manuellement un réseau de transport public et pour ainsi voir les impacts de tels changements dans la planification. Ils proposent une interface web pour faire évoluer la demande à certains arrêts du réseau, retarder/avancer des éléments du service planifié et permettent de visualiser instantanément l'impact de ces modifications sur le comportement des usagers du réseau.

Au-delà des deux sociétés précitées, d'autres entreprises plus établies travaillent aussi dans ce domaine et produisent différents outils de visualisation et d'aide à la planification. On peut citer par exemple Xerox, Thales, IBM, GIRO, INRO mais il en existe bien d'autres encore, alors que le secteur du « big-data » et des outils de visualisation semble voué à un grand essor.

CHAPITRE 3 MÉTHODOLOGIE

3.1 Méthodologie générale

Dans l'optique de produire un outil de visualisation permettant de visualiser et d'analyser les données de transactions des cartes à puce enrichies des destinations du RTL, plusieurs étapes sont nécessaires.

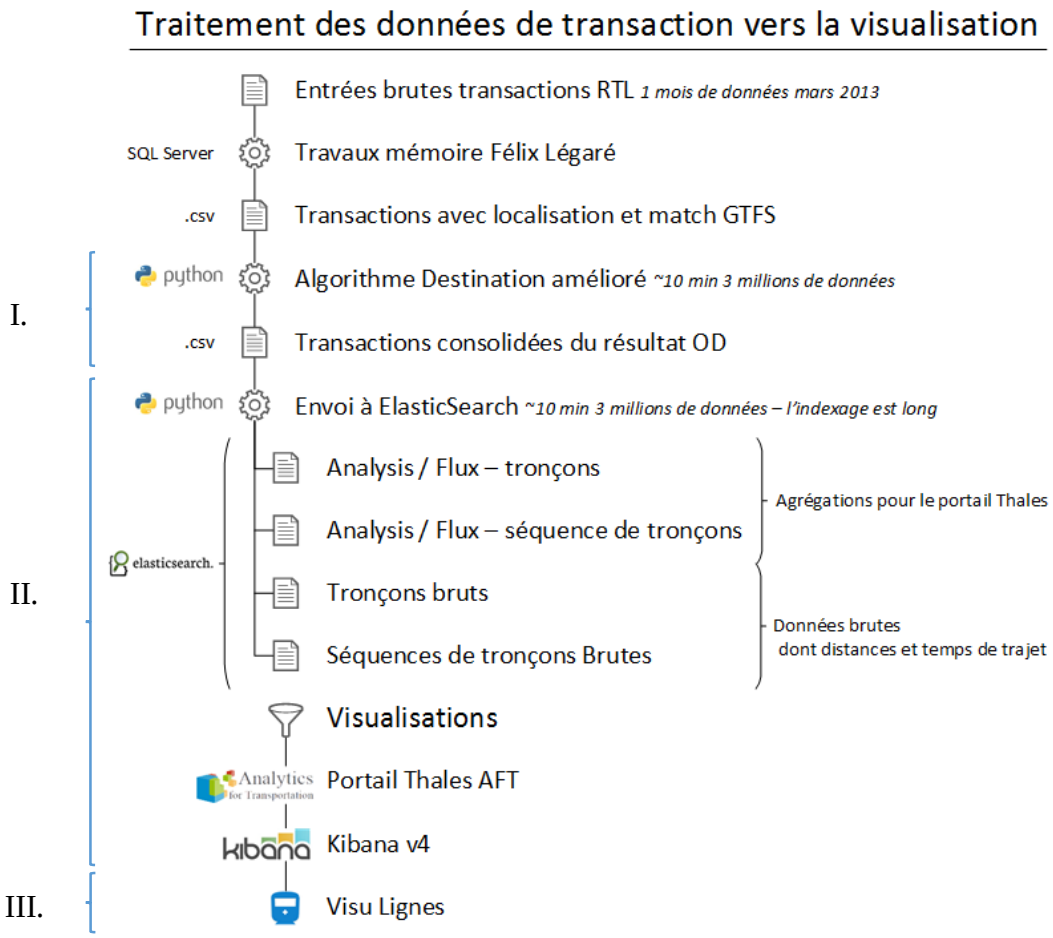


Figure 3-1 - Schéma du traitement des données de transaction vers la visualisation de celles-ci

La Figure 3-1 représente le traitement des données de transaction qui a été développé au Réseau de Transport de Longueuil (RTL), ce qui permet de mieux comprendre les apports et étapes de ce mémoire. Dans un premier temps (Figure 3-1 I.), l'algorithme destination (Chapitre 2.3) est adapté et optimisé pour le cas du RTL (Chapitre 2.2). Dans un second temps (Figure 3-1 II.), ces données enrichies de la destination sont intégrées dans un portail utilisant des technologies similaires à celles utilisées dans la plateforme développée par le CeNTAI de Thales (Chapitres 0 et 2.4.2.2). Le

but est ensuite de voir ce qu'il est possible de réaliser en termes de visualisation des données en les confrontant aux exigences que peut avoir un exploitant de transport en commun. Enfin, suite à l'intégration de ces données et à la lumière des besoins identifiés de l'exploitant et la revue de littérature (Chapitre précédent), ce projet de recherche propose de nouvelles formes de visualisations (Figure 3-1 III.).

3.2 Amélioration de l'algorithme destination

En premier lieu, il convient de revenir sur l'algorithme de He et de présenter plus en détail non pas la logique de l'algorithme, qui a déjà été présentée déjà au chapitre 2.3, mais les données d'entrée et de sortie et la performance de l'outil. Par ailleurs, nous présentons une amélioration que nous avons faite à ce dernier avant même de l'adapter au cas du RTL.

Dans un second temps, les données disponibles et les enjeux liés au cas du RTL en termes d'adaptation sont présentés. Une présentation de la nouvelle structure et de la logique générale de l'algorithme destination est également faite.

Enfin, les derniers chapitres de cette section abordent les différentes améliorations souhaitables pour cet algorithme destination. De nouveaux codes de sortie de l'algorithme ont été ajoutés. Une modélisation objet a été créée pour simplifier le code. Des indicateurs sur la performance et les données lues ont été ajoutés. Une barre de progression pendant son exécution a aussi été implantée. Enfin, des statistiques, déjà calculées pour la résolution de l'estimation des destinations, permettent désormais d'avoir en sortie de l'algorithme des transactions-déplacements enrichis de ces différentes métriques.

3.2.1 Présentation de l'algorithme destination de He

L'algorithme destination de He (He, 2014) est un script python d'environ 500 lignes réalisé pour retrouver les destinations des transactions de la STO. Celui-ci prend en entrées trois sources de données : les arrêts et les lignes-arrêts du réseau et enfin les transactions effectuées à l'aide des cartes à puce. Leur structure est présentée dans la Figure 3-2. Dans le code d'exemple fourni, on dispose de 2010 arrêts, 10109 lignes-arrêts et de 65535 transactions.

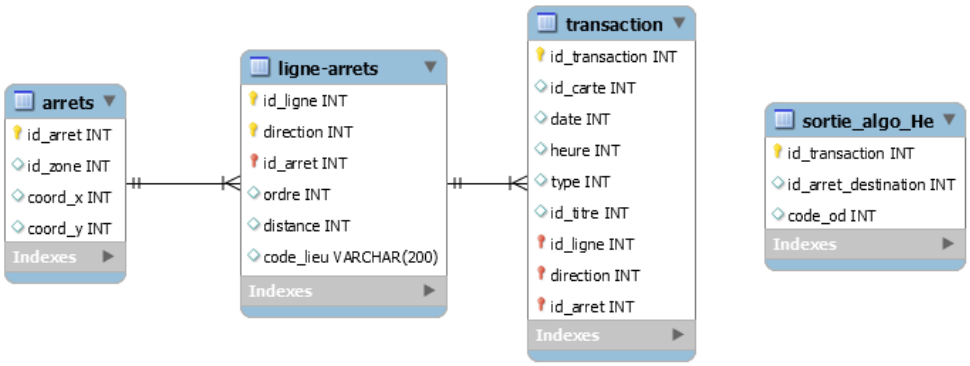


Figure 3-2 - Structure des données d'entrée dans l'algorithme destination de (He, 2014)

L'algorithme commence par charger les données d'arrêts et ligne-arrêts du réseau; il charge aussi l'intégralité des transactions en mémoire. Ce dernier point est problématique, car si on avait par exemple 30 millions de transactions disponibles, celles-ci seraient alors toutes chargées en mémoire. Il faudrait alors avoir un ordinateur disposant de suffisamment de mémoire vive pour pouvoir faire tourner l'algorithme. Même si 30 millions de transactions ne représentent que 2 Go, l'algorithme actuel n'est pas adaptable à un grand nombre de données. De plus, le temps de chargement des données n'est pas linéaire (Figure 3-3a). Plus il y a de données à charger, plus la préparation est longue. Ainsi, pour 3 millions de transactions, il conviendrait d'attendre 35 heures avant même de pouvoir commencer à calculer les premières destinations.

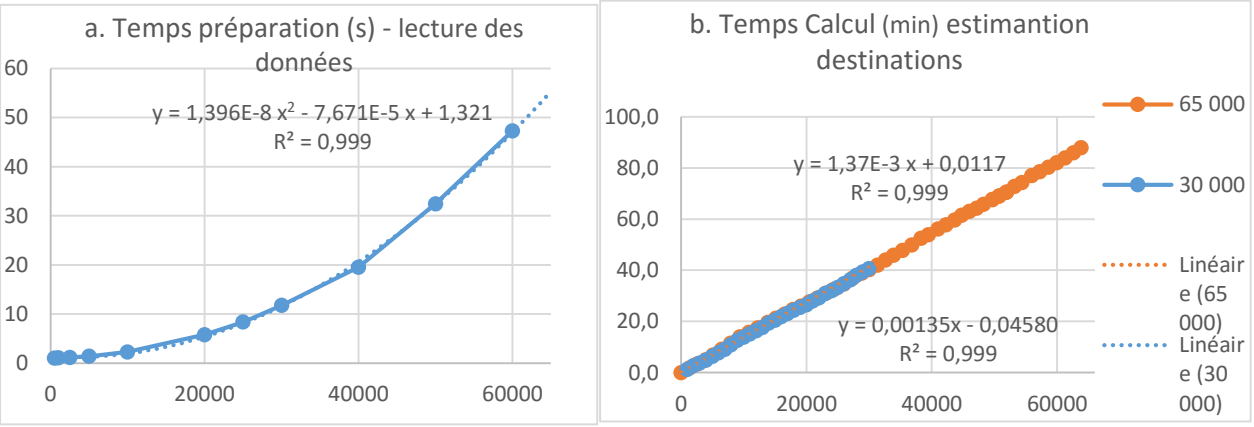


Figure 3-3 - Performances Algorithme destination de (He, 2014) pour 65 000 transactions

Une fois les données chargées, l'algorithme estime les destinations tel que présenté par (He, 2014). Le temps de calcul est cette fois linéaire comme le montre le graphique de droite de la Figure 3-3b – où l'on a relevé le temps de calcul pour 30 000 et 65 000 points et retrouvé les courbes de

régression linéaire. Ainsi, pour 3 millions de transactions, l'algorithme met 68 heures pour estimer les destinations.

L'algorithme remplit au fur et à mesure un fichier de sortie avec le résultat de l'estimation de la destination – table *sortie_algo_He* dans la Figure 3-2. Ce fichier de sortie ne contient que l'id de la transaction, l'id de l'arrêt de destination et le code OD (Origine-Destination) qui précise de quelle manière l'algorithme a déterminé cette destination. La signification de ces codes OD est précisée dans le Tableau 3-1.

Tableau 3-1 - Définitions des différents codes OD de l'algorithme destination – tiré de (He, 2014)

Code	Définition
11	Séquence de déplacement
12	Retour à domicile
13	Déplacement du prochain jour
21	Déplacement unitaire avec plusieurs emplacements de débarquement potentiels
22	Déplacement unitaire avec emplacement de débarquement potentiel unique
30	Pas de résolution encore

3.2.2 Optimisation brute de l'algorithme de He

Au vu du temps mis par l'algorithme pour estimer les destinations et afin de mieux comprendre sa logique, avant même de penser à l'adapter au cas du RTL, il a fallu l'optimiser afin de disposer de temps de calcul raisonnables. En effet, la préoccupation principale de He n'était pas de développer un algorithme performant ; son objectif était avant tout de montrer qu'il était possible d'avoir un algorithme permettant d'estimer des destinations.

Afin de disposer d'une application fonctionnelle, il convient de prendre en compte quelques principes de programmation pour maximiser la performance du code. Il faut que la logique de l'algorithme ne réalise pas de taches inutiles ou déjà faites précédemment. De même, l'affichage d'informations dans la console est coûteux en temps et doit être utilisé à bon escient. De plus, les boucles réalisées nécessitent une attention particulière selon la proportion des données. Ceci est encore plus vrai lorsque l'on souhaite réaliser des boucles imbriquées. Enfin, un langage script de 500 lignes est pratique pour réaliser une preuve de concept. Il est par contre vivement conseillé dans un premier temps de réaliser des fonctions pour éviter une redondance de plusieurs éléments du code.

L'algorithme de He travaillait déjà par carte ; ce principe va être gardé. Cependant, les transactions étaient lues et préparées dans leur totalité en amont et n'étaient alors pas pensées « par carte ». Une solution d'optimisation consiste à travailler tout de l'algorithme par carte. Le fichier d'entrée qui dispose des données de transactions doit être ordonné par identifiant de carte puis par date et heure de chaque transaction. L'idée est de parcourir et de garder en mémoire les transactions de la carte actuellement parcourue. Lorsque l'on détecte un nouvel identifiant de carte, on sait que l'on peut alors estimer les destinations des transactions accumulées jusque-là car appartenant à la même carte. Ceci permet en plus un gain de mémoire utilisé ; on gardera en mémoire non pas la totalité des transactions mais seulement celles reliées à une même carte. Ceci permet d'avoir la phase de préparation des données réalisée au fur et à mesure de la lecture des transactions.

Antérieurement, avec cette phase de préparation des transactions, il y avait une boucle imbriquée parcourant plusieurs fois la totalité des transactions. Il y avait également un classement non optimisé des identifiants des cartes (respectivement les premières, les dernières ou les transactions des cartes étant les seules pour un jour donné sous la forme de trois tableaux). Ces tableaux comportaient l'intégralité des transactions lues. Une première amélioration a été de les segmenter par carte afin de ne parcourir que les transactions premières, dernières ou seules d'une carte donnée. La solution finale a été de calculer directement ces informations dès la lecture de chaque transaction et de les rattacher à leurs tableaux respectifs de données. D'autres cas similaires ont été corrigés dans le code. Enfin, pour permettre une modification plus facile des paramètres-clés de l'algorithme, un fichier de configuration les recensant a été créé.

En suivant ces premiers principes, l'algorithme de He a été rendu plus fonctionnel et rapide avant de commencer ensuite à le complexifier et à l'adapter pour un réseau de transport en commun disposant d'un GTFS.

3.2.3 Données disponibles

Les données disponibles pour ce mémoire sont les données de transactions de bus du RTL du mois de mars 2013 enrichies de leurs localisations par (Légaré, 2014). À ces transactions disponibles, se sont ajoutées celles effectuées le même mois, par les utilisateurs du réseau à 4 stations de métro desservies en bus par le RTL (Longueuil-Université-de-Sherbrooke, Bonaventure, Radisson et Papineau). Enfin, le GTFS représentant le service planifié offert par le RTL a lui aussi été disponible permettant ainsi de connaître les arrêts du réseau ainsi que toutes les courses des bus et

ligne-arrêts-heures (stop-times) sur les différentes lignes et tracés. Par contre, les données GPS ou de comptage à bord n'ont pas été utilisées au cours de ce mémoire (puisqu'elles sont intégrées dans les travaux de Légaré).

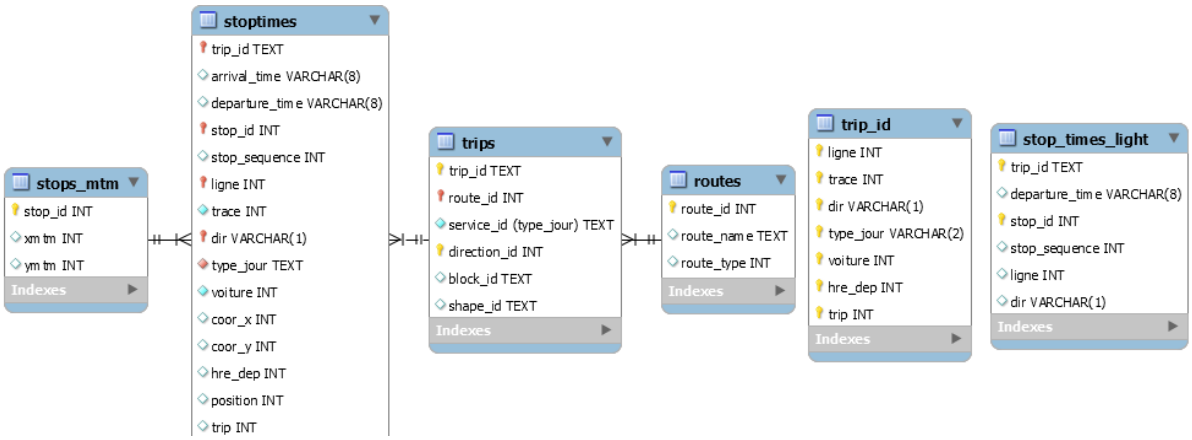


Figure 3-4 - Structure des fichiers composant le GTFS du service planifié du RTL

La Figure 3-4 résume les différents éléments composant le GTFS représentant le service planifié du RTL. Le fichier « stop-times » disposant de 404 339 entrées ligne-arrêt-heure d'une taille de 38 Mo a été simplifié en un fichier représenté par la table « stop_times_light » à droite d'une taille de 18 Mo afin de ne garder que les informations utiles à l'algorithme destination. Les arrêts du réseau ont été complétés des quatre stations de métro desservis par le RTL.

La structure des données de transactions enrichies par Légaré est présentée dans la Figure 3-5. Cette figure indique aussi ici le lien avec la table des courses GTFS. On retrouve les localisations ordonnées dans l'ordre chronologique de leur détermination par l'algorithme pensé par Légaré – chapitre 2.2. C'est dans cet ordre que l'on doit considérer comme valide l'arrêt de chaque transaction : *arret_sdap, arret_habitude*, *arret_ligne* puis enfin *arret_gtfs*.

La structure des données de transactions du métro est similaire à celle du bus. Au final, les transactions doivent être confondues dans une même table et être enregistrées dans un fichier texte qui sera lu en entrée par l'algorithme destination.

Afin d'économiser la taille du fichier, l'identifiant des différentes cartes a été encodé dans une base 36 en partant de 0 jusque 103 540 (27W4), alors qu'avant celui-ci prenait 40 caractères. De plus, seuls les champs pouvant servir durant l'algorithme destination, ou durant les analyses futures, ont été retenus. On a ainsi choisi de garder les variables *xxx_prod* représentant le type de produit de la transaction. La structure de ces données d'entrée est représentée dans la Figure 3-6. On retrouve

des champs présentés par (Légaré, 2014). Un champ pour différencier le bus du métro a été ajouté (typeTransport).

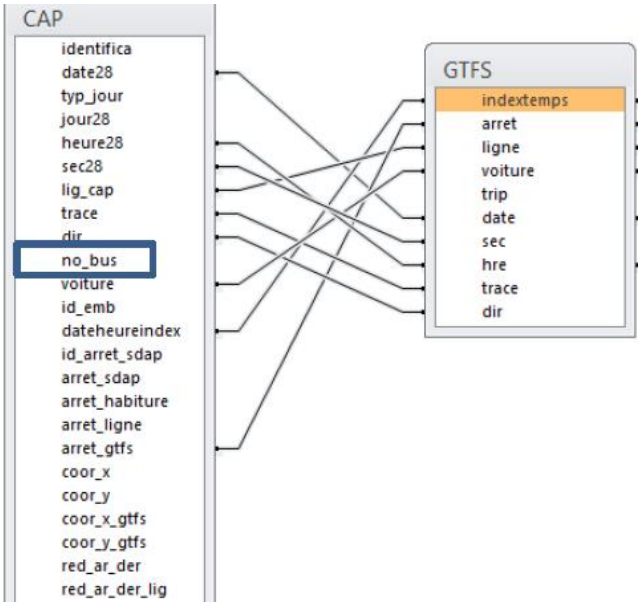


Figure 3-5 - Structure des données de transaction du RTL enrichies de leur localisation – tiré de (Légaré, 2014)

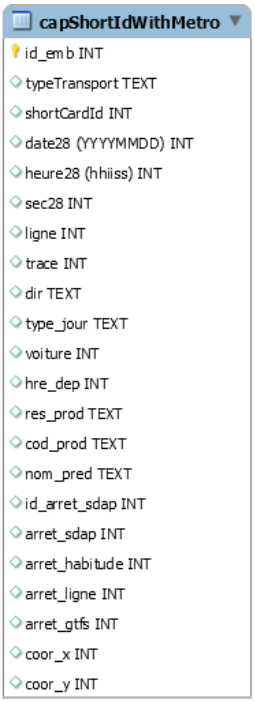


Figure 3-6 - Structure des données entrant dans l'algorithme destination

3.2.4 Enjeux liés au cas du RTL

Comme certaines données de transaction effectuées par des utilisateurs antérieurs du métro (avant de rejoindre le réseau de bus) sont disponibles, il faut penser à gérer celles-ci différemment. Il n'est néanmoins pas nécessaire de chercher à retrouver leur destination car on ne sait pas précisément où les personnes sont allées sur l'île de Montréal. Cependant, il pourrait être intéressant de vérifier si une personne ayant pris le métro revient par cette même station pour retraverser le fleuve. On devrait alors regarder si la transaction suivante est proche ou non de l'arrêt de métro pris.

Contrairement à l'algorithme de He (STO), un GTFS représentant le service planifié est désormais disponible. Les transactions correspondantes ont déjà été affectées à un GTFS par le RTL.

L'avantage du GTFS est que l'on n'a pas à approximer le temps de trajet à bord en se basant sur une vitesse moyenne comme He a dû le faire. Le temps de trajet entre deux arrêts se retrouve en calculant la différence entre les temps de passage à ces deux arrêts. L'idéal serait de faire comme

(Chu & Chapleau, 2008) et de retrouver au préalable, pour chaque course le temps réellement effectué. En effet, si l'on avait pu disposer des données GPS et du comptage à bord, cela aurait permis d'affiner les résultats ici obtenus avec les données de cartes à puce.

Cependant, le fait d'utiliser un GTFS a mis en évidence quelques travers à prendre en compte. En effet, il arrive que des transactions soient affectées à un mauvais GTFS, engendrant alors de possibles incohérences. Ces erreurs ont déjà été identifiées par (Tranchant, 2005). On retrouve dans ces cas une ligne inconnue (erreur ou définition du réseau incomplète), un arrêt inconnu, un arrêt n'appartenant pas à la course prise (ligne, direction et trace) ou encore un arrêt de montée étant le dernier de la ligne. Ces transactions sont donc écartées et n'apparaissent pas, même dans le code « 30 » (échec).

3.2.5 Définition des termes tronçons et séquence de tronçons

Dans la réalité, les usagers du réseau peuvent avoir besoin au cours de leurs déplacements d'emprunter plusieurs lignes du réseau et de réaliser des correspondances pour se rendre à destination.

Un tronçon ou une section représente le voyage d'un usager sur une ligne donnée entre les arrêts de montée et descente sur le réseau. Une séquence de tronçons correspond au déplacement d'une personne ayant utilisé le réseau de transport pour se rendre d'une origine à une destination en ayant utilisé au moins deux lignes différentes et ayant ainsi réalisé une correspondance. On peut assimiler cette notion à un déplacement, bien que les lieux réels d'origine et de destination ne soient pas connus. Attention toutefois à ne pas confondre ce cas de figure avec une chaîne de déplacements qui est une séquence de déplacements réalisée au cours d'une journée, basée de son domicile. Dans les deux cas (tronçon et déplacement), une matrice Origine-Destination (OD) est obtenue.

Si l'on souhaite s'intéresser à la mobilité de la population, on préférera travailler sur la matrice OD des extrémités des séquences de tronçons (déplacements). Par contre, si on souhaite visualiser la charge et l'utilisation du réseau, on restera avec les tronçons simples.

Pour rappel, un GTFS est composé de lignes qui possèdent elles-mêmes différents tracés possibles. Une course représente le trajet planifié d'un bus pour une heure donnée de la journée, pour un tracé d'une ligne donnée.

3.2.1 Structure et logique de l'algorithme destination

Sur le fond, la structure et la logique de l'algorithme destination restent les mêmes que celles pensées par He pour l'estimation des destinations. Des changements apportés, déjà évoqués dans le chapitre 3.2.2, ont néanmoins permis de rendre ce dernier plus performant.

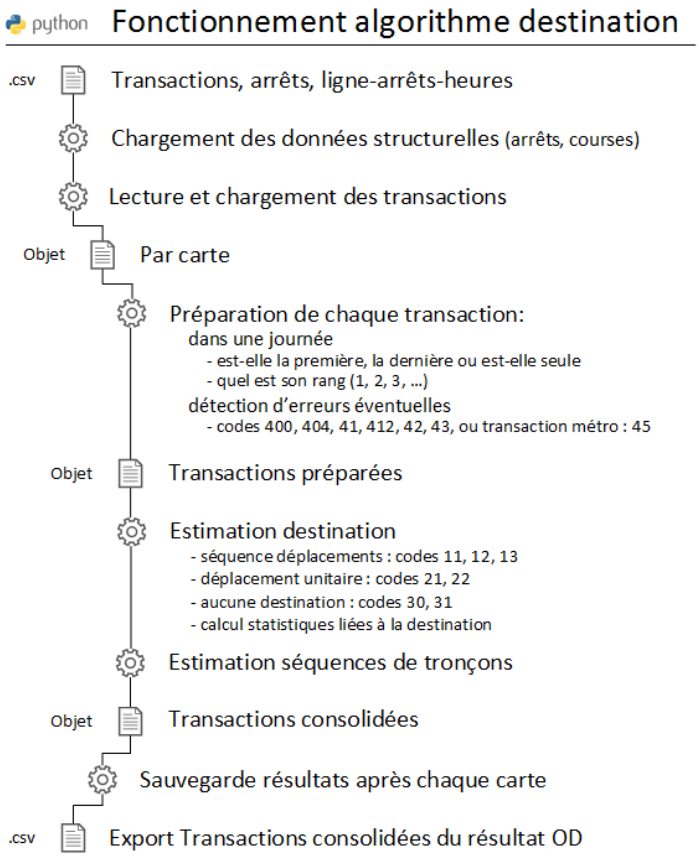


Figure 3-7 - Fonctionnement de l'algorithme destination

La Figure 3-7 représente le fonctionnement de l'algorithme. Dans un premier temps, ce dernier charge les données structurantes (arrêt, GTFS). Il parcourt ensuite les transactions en travaillant une carte à la fois, c'est-à-dire qu'il arrête de lire le fichier de transactions lorsqu'il rencontre la transaction d'une nouvelle carte afin de traiter celles déjà accumulées. On aura pour chaque carte des opérations de préparation, d'enrichissement et d'export des données :

- L'algorithme prépare pour chaque transaction des informations utiles pour l'algorithme destination telles que le rang de la transaction dans la journée et si elle est la première, la dernière ou si elle est seule pour cette journée. Il charge aussi dès maintenant la ligne et la

course GTFS associée à cette transaction. Il peut détecter dès à présent les différents cas d'erreur.

- Une fois toutes les données d'une carte chargées et préparées, il travaille sur les transactions une à une pour estimer leurs destinations. Une fois ces destinations retrouvées, on peut recomposer les séquences de tronçons et indiquer pour chaque transaction la séquence à laquelle elle appartient.
- La dernière étape est d'exporter les transactions enrichies de la carte courante.

3.2.2 Nouveaux codes de résolution ou d'erreur

Comme évoqué précédemment, les transactions rejetées n'étaient pas catégorisées. Six codes d'erreur ont été ajoutés dans le nouvel algorithme (Tableau 3-2). On retrouve les erreurs relevées par Tranchant (41, 42, 400/404, 44) et les transactions du métro (45). Le code 412 apparaît lorsque, pour un GTFS défectueux, l'algorithme a essayé de retrouver le bon GTFS sans succès. En effet, il est possible de trouver une meilleure course en comparant les temps de départ et d'arrivée aux arrêts des autres courses d'un tracé. De plus, si l'arrêt de montée est le dernier d'une ligne, la personne a probablement pris le voyage GTFS suivant de cette même ligne ou de ce véhicule. Enfin, le code 30 précédent (Tableau 3-1) a été séparé en deux. Le code 31 est levé lorsqu'on n'a pas pu charger un historique de transactions, sinon le code 30 représente les cas restants où l'on n'a pas trouvé de destination.

Tableau 3-2 - Définitions des différents codes d'erreur de l'algorithme destination

Code	Définition
11	Séquence de déplacement
12	Retour à domicile
13	Déplacement du prochain jour
21	Déplacement unitaire avec plusieurs emplacements de débarquement potentiels
22	Déplacement unitaire avec emplacement de débarquement potentiel unique
30	Pas de destination trouvée
31	Pas de destination trouvée - impossible de créer l'historique des transactions
400	Id arrêt vide
404	Id arrêt inconnu
41	ligne inexistante
412	impossible de retrouver la ligne la plus proche
42	arrêt et ligne incompatible
43	arrêt embarquement == terminus ligne
45	transaction métro

3.2.3 Modélisation objet

Comme vu précédemment, on dispose de plusieurs objets déjà utilisés : des cartes, des transactions, des lignes, des arrêts, des « stop-times » ou heures de passage aux arrêts d'une course. Afin de mieux l'organiser et de faciliter sa compréhension, l'algorithme destination a été recodé pour utiliser les concepts de Programmation Orientée Objet (Figure 3-8 et Figure 3-9). De plus, afin de permettre une compatibilité avec différents types de réseau ou permettre d'adapter le code plus facilement à un nouveau réseau, un travail a été réalisé pour définir des classes mères et des classes filles. Les classes mères définissent et contiennent la logique de l'algorithme. Les classes filles reprennent au besoin les fonctions de la classe mère pour les adapter au besoin de la nouvelle structure réseau ou aux nouveaux formats des fichiers d'entrée. Des classes « contrôleur » ont aussi été créées pour gérer l'accès aux informations propres aux lignes ou aux arrêts. On les utilisera par exemple pour leur demander de retourner tel arrêt ou telle ligne, ou bien de retourner la course GTFS la plus près de telle transaction.

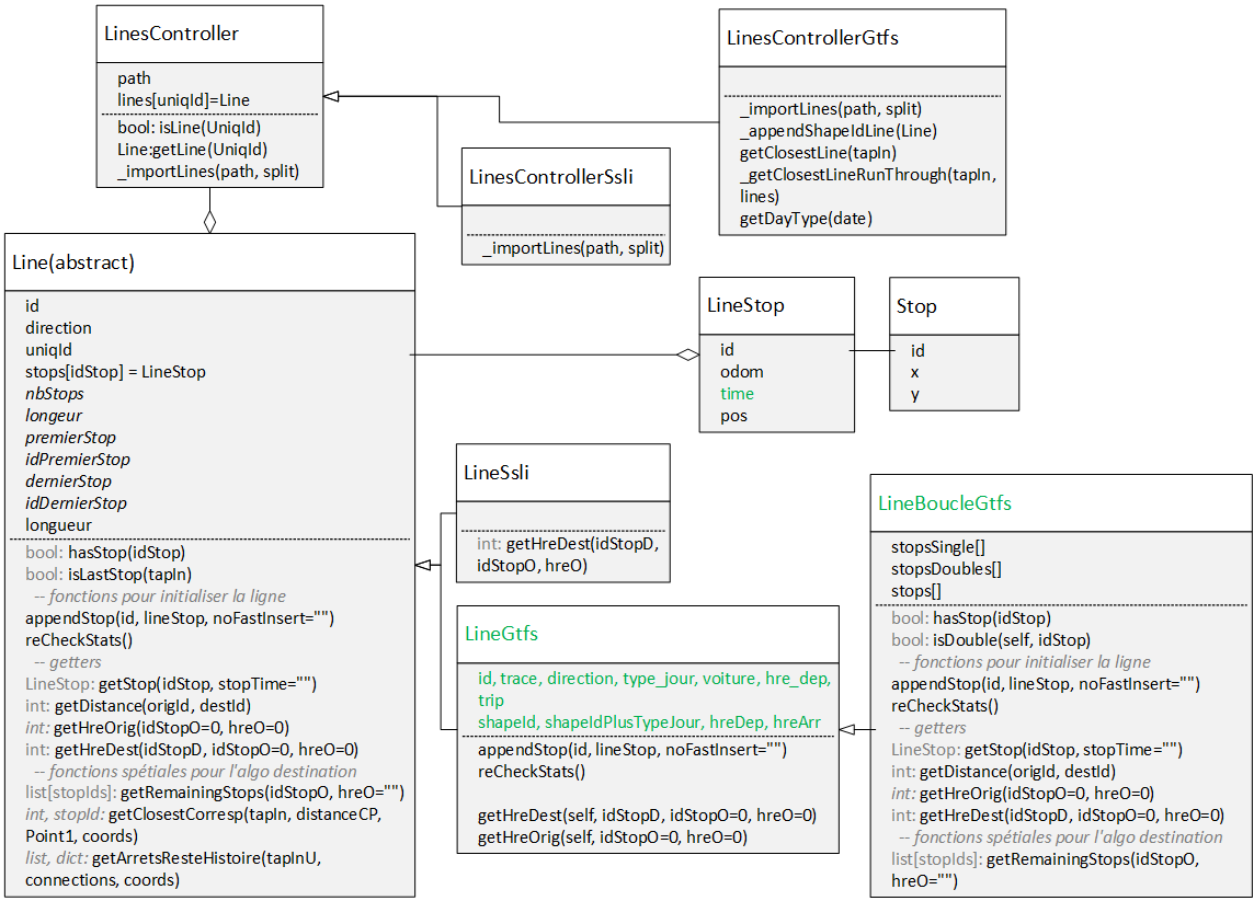


Figure 3-8 - Architecture objet de l'algorithme destination – *Lignes*, *LigneController*

La classe *StopsController* (Figure 3-8) permet de rendre accessibles aux autres classes les coordonnées et autres attributs des arrêts du réseau. Les classes *Line* (Figure 3-9) et ses enfants permettent de représenter les différentes lignes ou courses pour les classes *LineGtfs*. Elles proposent des méthodes simples (*getStop*, *getDistance*, *getHreOrig*, *getHreDest*) ou adaptées pour les besoins de l'estimation des destinations (*getRemainingStops*, *getClosestCorresp*, *getArrets ResteHistoire*) pour interagir avec les éléments d'une ligne, à savoir *LineStop* lui-même lié par l'id à un arrêt contenu par la classe *StopsController*. Comme mentionnée précédemment, la classe *LineController* (Figure 3-9) va permettre aux autres classes (*TapIn*, *Carte*) de lui demander une ligne correspondant à un identifiant donné (*isLine*, *getLine*). La classe *LinesControllerGtfs* va elle permettre en plus de retourner la course d'une ligne GTFS la plus près de l'arrêt et l'heure d'une transaction donnée (*getClosestLine*).

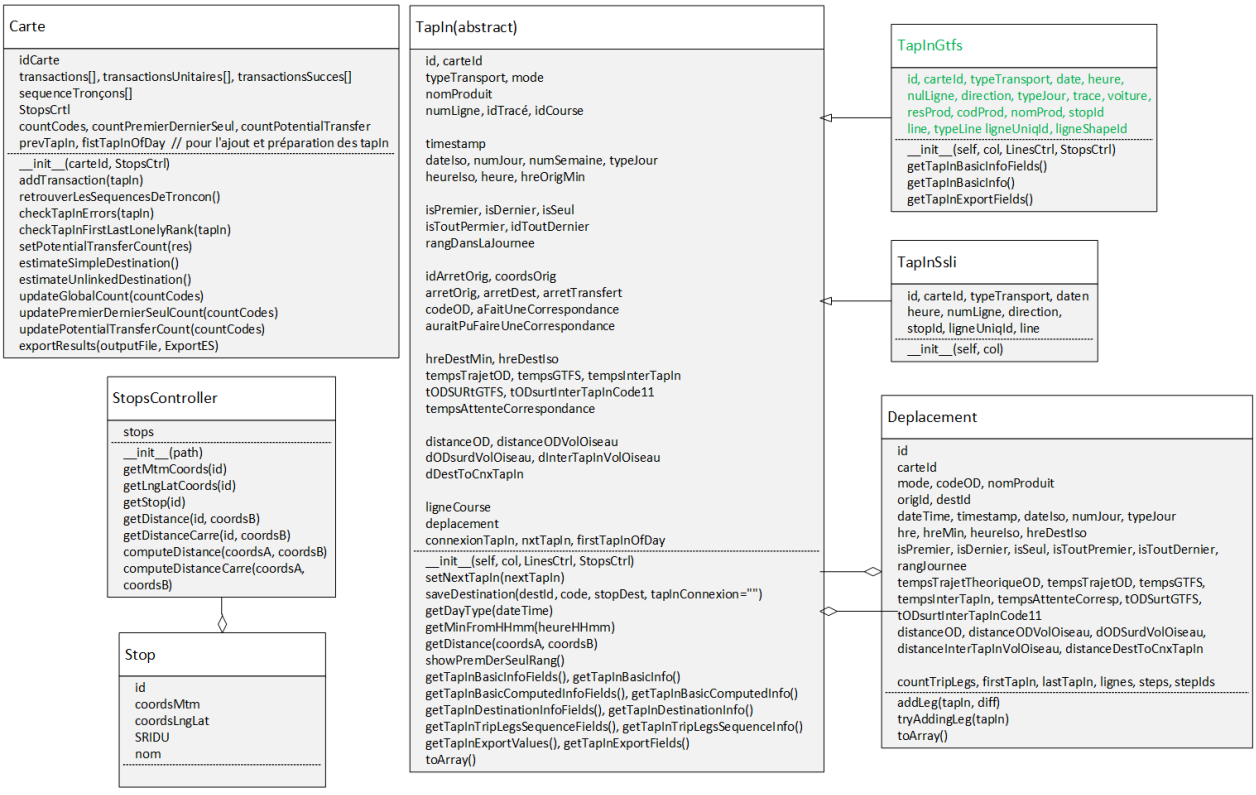


Figure 3-9 - Architecture objet de l'algorithme destination – *Transaction*, *Carte*, *Deplacement*, *StopController*

La classe *Carte* (Figure 3-9) va contenir la logique liée à la préparation (*addTransaction* , *checkTapInErrors*, *checkTapInFirstLastLonelyRank*, *setPotentialTransferCount*), à l'estimation des destinations (*estimateSimpleDestination*, *estimateUnlinkedDestination*), à la résolution des

séquences de tronçons (*retrouverLesSequencesDeTroncon*), à l'export des résultats (*exportResults*) ainsi qu'à la mise à jour des statistiques (*updateGlobalCount, updatePremierDernierSeulCount, updatePotentialTransferCount*). Elle contient la partie principale de la logique de l'algorithme destination. Les autres classes sont là pour servir cette dernière afin de rendre plus simple la lecture et l'écriture du code.

La classe *TapIn* ou *Transaction* (Figure 3-9) contient un grand nombre de champs. Les champs de base lus dans le fichier des transactions sont initialisés dans les classes filles. Ce sont ces classes filles qui sont instanciables. On ne peut pas utiliser la classe mère pour créer un objet Transaction. La classe mère contient elle, par contre, la logique de préparation d'autres statistiques liées aux données brutes (*typeJour, numJour, numSemaine, heureIso, dateIso, stop, coordsMtm, …*). Certains champs (*codeOD, isPremier*) sont initialisés par la classe *Carte* au cours des fonctions *checkTapInErrors* et *checkTapInFirstLastLonelyRank*. Ce sont des informations recomposées à partir de la lecture des autres transactions d'une carte. Cette même classe *TapIn* enregistre les résultats de l'algorithme destination (*codeOD, destId, arretDestination, tapInConnexion*) grâce à la méthode *saveDestination*. Cette méthode permet également d'enregistrer d'autres informations suite à la destination désormais connue tel que des distances (*distanceOD, distanceODVolOiseau, distanceInterTapInVolOiseau, distanceDestToCnx TapIn*), des temps (*tempsTrajetOD, tempsGtfs, tempsInterTapIn, tempsAttenteCorrepondance*) ou des rapports de ces métriques (*tODSURtGTFS, tODSurtInterTapInCode11, dODSurdVolOiseau*). La classe *TapIn* permet de plus d'y ranger les fonctions permettant de choisir quels champs extraire vers le fichier de sortie en rappelant les champs exportés (*getTapInBasicInfoFields, getTapInBasicComputedInfoFields, getTapInDestinationInfoFields, getTapInTripLegsSequence Fields, getTapInExportFields*) ainsi que les valeurs de ces mêmes champs (*getTapInBasicInfo, getTapInBasicComputedInfo, getTapInDestinationInfo, getTapInTripLegsSequenceInfo, getTap InExportValues*). La classe *TapIn* permet de représenter les valeurs clés de chaque transaction en un tableau de clés et de valeurs (*toArray*). Cette fonction sera utilisée au cours du chapitre 3.3 et 3.4 pour exporter les résultats dans Elasticsearch. Les champs de sortie seront reprécisés au chapitre 3.2.5.

Enfin, la classe *Déplacement* (Figure 3-9) est peuplée par la classe *Carte* et la méthode *retrouverLesSequencesDeTroncon*. Cette classe est composée des différentes transactions qui ont effectué une correspondance entre elles et composent une séquence de tronçons (Section 3.2.5). Des statistiques similaires à celles composées pour les transactions une fois dotées de leur

destination sont mises à jour en fonction des transactions composant le déplacement. Il peut arriver qu'un déplacement ne comporte qu'un seul tronçon; cela permet de marquer dans les séquences de tronçons toutes les transactions, même celles étant des échecs (code OD 30, 31) ou des erreurs (code OD > 40). Chaque transaction reçoit aussi le lien vers l'objet *Déplacement* auquel elle appartient. Si la transaction suivante ne dispose pas d'une destination, la destination et l'heure d'arrivée de cette séquence de tronçons sont supposées être l'origine de la dernière transaction et son heure de validation.

3.2.4 Suivi de la progression et statistiques de la résolution de l'algorithme

Un travail a été réalisé pour permettre de présenter différents indicateurs sur les données lues et le temps passé par l'algorithme à chaque étape : préparation, estimation des destinations et récapitulatif de fin. La Figure 3-10 présente un exemple de retour console de l'algorithme.

Chaque transaction est catégorisée en fonction d'un code OD évoqué dans les chapitres précédents (Tableau 3-1**,** Tableau 3-2). La performance et précision de l'algorithme destination peut être améliorée en ayant moins de codes d'erreur liés à une codification imparfaite du réseau (>40).

```
C:\PolyRTL\AlgoDestination (master)
λ python algo_destination.py
HEAD /data_rtl_v4 [status:404 request:0.002s]
Nouvel index de travail data_rtl_v4 créé.
----- Préparation des données -----
3250 stops importés en 0.01723s
404339 stops pour 11577 trips importés en 3.70471s
----- Fin Préparation après 3.83240s -----
 ----- début calcul Origines Destinations -----
Alias data_rtl mis à jour vers data_rtl_v4
---------- Résultats : ----------
----- Temps de calcul : -----
   155.89s (13.0%) : tCalculDestinationsSimples
   270.00s (23.0%) : tCalculDestinationsUnitaires
   550.64s (47.0%) : tExportResultats
   204.59s (17.0%) : tPreparation
  1181.10s ( 100%) : tGlobal
----- Autres statistiques : -----
countPremierDernierSeul [('dernier', 1194831), ('premier', 1194831), ('seul', 311746)]
Transferts: 575030 effectués sur 763452 possibles
  Cas refus transfert:  mêmeOrigine: 7434, mêmeLigne: 33428,
    arretDejaUtilisé: 124, ligneDejaUtilisée: 2886, tropLoinDansLeTemps: 113915
Deplacements: 2153633 effectués sur 2836465 possibles (on a écarté les 264380 transactions sans arrêt connu)
  Distribution des déplacements selon le nombre de segments:
    {1: 1502084, 2: 620765, 3: 30293, 4: 483, 5: 8}
Lignes: GTFS correct: 2061868, GTFS recalculé: 21652, erreurs: 97903
----- Statistiques destinations trouvées : -----
   1963483 / 3100845 (63.3%) destinations trouvées pour 103540 cartes en 1181.124s
ou 1963483 / 2481977 (79.1%) - bus uniquement
ou 1963483 / 2836465 (69.2%) - sans codes 400, 404
----- répartition des codes : -----
code 11:   1145542 - Séquence de déplacement
code 12:    561286 - Retour à domicile
code 13:     65323 - Déplacement du prochain jour
code 21:    141112 - Déplacement unitaire avec plusieurs emplacements de débarquement potentiels
code 22:     50220 - Déplacement unitaire avec emplacement de débarquement potentiel unique
code 30:      3833 - Pas de destination trouvée
code 31:     68075 - Pas de destination trouvée  - Impossible de créer l'historique des transactions
----- Erreurs : -----
code 400:    184429 - Code d'erreur: id arret vide
code 404:     79951 - Code d'erreur: arret inconnu
code 41 :     84443 - Code d'erreur: ligne inexistante
code 412:     13460 - Code d'erreur: impossible de retrouver la ligne la plus proche
code 42 :     36174 - Code d'erreur: arret et ligne incompatible
code 43 :     48129 - Code d'erreur: arret embarquement == terminus ligne
code 45 :    618868 - transaction metro (arrets 150, 168, 252 et 454)
Algo exécuté en 1184.96684s
```

Figure 3-10 - Aperçu exemple de retour console de l'algorithme destination

3.2.5 Autres indicateurs et informations pour le fichier de sortie

Différentes statistiques sont calculées au cours de l'exécution de l'algorithme. Cependant, ces statistiques n'étaient pas initialement enregistrées dans le fichier de sortie et devaient être recalculées en parcourant de nouveau toutes les transactions. Afin de gagner du temps de calcul, ces statistiques sont désormais exportées avec le résultat de l'algorithme destination afin de

disposer de transactions-déplacements enrichis de leurs départs et destinations ainsi que d'autres statistiques.

Comme vu à la section 3.2.3, la classe pour les transactions permet d'enregistrer désormais de nombreuses statistiques. Celles-ci peuvent être rangées dans les catégories suivantes : informations brutes des transactions, informations recomposées des transactions, résultat algorithme destination, autres informations suite à la destination, recomposition des séquences de tronçons. Les sections suivantes recensent les informations ou autres indicateurs lus ou calculés et placés dans le fichier d'export des résultats de l'algorithme. Certains champs peuvent être utiles et ont été rajoutés spécialement en vue des visualisations futures.

3.2.5.1 Informations brutes des transactions

Il s'agit ici des informations tirées des données contenues dans le fichier des transactions. Des informations supplémentaires peuvent dès à présent être récupérées avec ces seules informations telles qu'aller chercher la ligne / course utilisée par la transaction ou encore le numéro de jour dans la semaine, le numéro de semaine dans l'année, etc. Le Tableau 3-3 recense ces différents champs ainsi que leur description.

Tableau 3-3 - Description des informations brutes d'une transaction

Champ	**description**
carteId	Identifiant Carte
tapInId	Identifiant Transaction
typeTransport	Type de transport (bus ou métro)
mode	Mode (2 : Bus seul, 3 : Métro, 7 : Bus ayant une connexion métro)
timestamp	Timestamp : yyyy-MM-dd hh:mm
dateIso	Date au format Iso 8601
heureIso	Heure au format Iso 8601
numLigne	Identifiant Ligne
direction	Direction
stopId	Identifiant arrêt
ligneUniqIdtrace	Identifiant course (trip_id)
voiture	Voiture / identifiant bus utilisé
ligneShapeId	Identifiant tracé (shape_id)
res_prod	Responsable produit
cod_prod	Code produit
nomProduit	Nom produit
hreOrig	Heure Origine en minutes
hreGtfsOrig	Heure Origine prévue par le GTFS
numJour	Numéro du jour dans la semaine
numSemaine	Numéro de la semaine dans l'année
typeJour	Type de jour (SE : semaine, SA : Samedi ; DI : dimanche, F1 : Férié)

3.2.5.2 Informations recomposées des transactions

Lors de l'ajout des transactions à une même carte, on peut tirer de nouvelles informations (Tableau 3-4) telles que le rang de cette dernière dans la journée. On en profite aussi pour indiquer à l'objet transaction courante quel est l'objet de la transaction juste après. On peut aussi dès à présent regarder la possibilité de correspondance avec la transaction juste après car étant à moins de x minutes (ex. 60 min) et n'étant pas sur la même ligne ou à partir du même arrêt.

Tableau 3-4 - Description des informations recomposées d'une transaction

Champ	**description**
estPremier	Première transaction du jour
estDernier	Dernière transaction du jour
estSeul	Seule transaction du jour
estToutpremierD	Toute première transaction de cette carte
estToutdernierD	Toute dernière transaction de cette carte
rangTapInJournee	Rang de la transaction dans la journée
couldHaveTransfered	Transfert possible avec la transaction suivante tout en respectant les contraintes de ligne, arrêt et fenêtre temporelle
nxtTapInId	Identifiant de la transaction suivante
nxtTapInTimestamp	Timestamp de la transaction suivante
nxtTapInStopId	Identifiant Arrêt de la transaction suivante

3.2.5.3 Résultat algorithme destination

Le Tableau 3-5 recense les informations minimales liées à l'export brut de l'algorithme destination. Le champ *hasTransfered* sera validé si la transaction a bien pu faire cette correspondance et si l'algorithme destination a retourné comme transaction de connexion la transaction juste après la transaction courante. Cela élimine alors les codes 12. Attention cependant, cette variable sera mise à jour de nouveau lors de l'étape de recomposition des séquences de tronçons.

Tableau 3-5 - Description des informations liées au résultat de l'algorithme destination

Champ	**description**
cnxId	Identifiant transaction connexion
cnxTimestamp	Timestamp arrêt connexion
cnxStopId	Identifiant arrêt connexion
destId	Identifiant arrêt destination
codeOD	Code Origine Destination
hasTransfered	Est-ce qu'il y a eu un transfert avec cette transaction de connexion

3.2.5.4 Autres informations suite à la destination

Suite au résultat de l'algorithme destination, on peut enregistrer différentes métriques telles que les heures estimées à destination, les distances et temps parcourus ou encore des rapports de ces dernières (Tableau 3-6).

Tableau 3-6 - Description des informations calculées suite au résultat de l'algorithme destination

Champ	**description**
hreGtfsDest	Heure GTFS prévue à l'arrivée
tempsTrajetOD	Différence entre l'heure prévue à l'arrivée et l'heure de la transaction
tempsGtfs	Temps planifié pour se rendre de l'origine à la destination
tempsInterTapIn	Différence entre les heures des deux transactions en correspondance
tempsAttenteCorrepondance	Temps passé avant la prochaine transaction (temps activité ou correspondance)
tODSurtInterTapInCode11	Pour les codes 11, rapport entre tempsTrajetOD et tempsInterTapIn
tODSURtGTFS	Rapport entre le tempsTrajetOD et le tempsGtfs
distanceOD	Distance sur le réseau entre l'origine et la destination
distanceODVolOiseau	Distance à vol d'oiseau entre l'origine et la destination
dODSurdVolOiseau	Rapport entre distanceOD et distanceODVolOiseau
distanceInterTapInVolOiseau	Distance à vol d'oiseau entre l'origine et l'origine de la transaction suivante
distanceDestToCnxTapIn	Distance à vol d'oiseau entre la destination et l'origine de la transaction suivante

3.2.5.5 Recomposition des déplacements

Une fois les destinations estimées, les transactions sont parcourues une nouvelle fois pour déterminer les séquences de tronçons (déplacements). Comme une transaction appartient à une séquence de tronçons, l'objet *Déplacement* auquel fait partie chaque transaction est disponible directement dans l'objet de chaque transaction. Cela va permettre par la suite de disposer des informations sur la séquence de tronçons directement dans le document représentant cette transaction. On sera alors, par exemple, capable de retracer sur une carte toutes les étapes d'une séquence donnée. Les informations rendues disponibles concernant la séquence de tronçons sont expliquées dans le Tableau 3-7.

Tableau 3-7 - Description des informations du déplacement incluant cette transaction

Champ : déplacement…	**description**
Id	Identifiant du déplacement : numCarte_rangDéplacementPourCetteCarte
Mode	2 : bus, 3 : métro, 7 : bus ET métro
CountTripLegs	Nombre de tronçons dans ce déplacement
RangJournee	Rang dans la journée du déplacement
Timestamp	Timestamp
TempsTrajetOD	Temps de trajet entre l'origine et la destination de la séquence de tronçons. Pour plus de précision, on reprend à chaque fois la différence entre l'heure de la dernière transaction et la toute première et on lui rajoute si disponible le temps de trajet du dernier tronçon
TempsEnTransport	Temps cumulé passé en transport
TempsAttenteCorrepondance	Temps de correspondance entre l'heure d'arrivée et l'heure de la transaction suivante
DistanceOD	Distance parcourue sur le réseau
distanceDestToCnxTapIn	Distance parcourue entre les arrêts du réseau lors des correspondances
EstPremier	Est-ce le premier déplacement de la journée ?
EstDernier	Est-ce le dernier déplacement de la journée ?
EstSeul	Est-ce le seul déplacement de la journée ?

3.3 Intégration des données dans une interface existante

Dans un second temps, les données enrichies de la destination sont intégrées dans une version du portail *Analytics for Transportation (AFT)* développée par le CeNTAI de Thales (Section 0). Cela permet d'avoir un exemple d'application web permettant d'analyser une grande quantité de données à l'aide des technologies Elasticsearch et Kibana (Section 2.4.2.2).

Cette section porte dans un premier temps sur l'intégration des données dans Elasticsearch en vue de leur visualisation à travers le portail AFT. Suite à l'intégration de ces données, on présente quelques limites de ce format de donnée face aux besoins d'une société exploitante de transport en commun. Enfin on propose un nouveau format de données désagrégées des transactions pour permettre pour la suite une plus grande liberté en termes d'exploitation et visualisation de ces données.

3.3.1 Données requises en entrée

Le portail AFT de Thales dispose de plusieurs pages web de visualisation. Dans le cadre de ce mémoire, on ne s'intéresse qu'aux parties Analyses et Flux. Le portail AFT est aussi capable de présenter les données issues de transactions en temps réel et d'afficher des alertes levées lors de la

préparation de ces données en cas de comportements anormaux dans l'utilisation du réseau. La structure des documents requis par le portail est précisée ci-après.

3.3.1.1 Structure des documents Elasticsearch

Chaque document Elasticsearch dispose d'une première partie définissant l'index, le type de document et son identifiant puis d'un champ _source dans lequel sont rangées toutes les autres propriétés du document. Voici ci-après un exemple de document *json* pour représenter un arrêt :

```
stations.json {
  "_index": "stations",
  "_type": "stations",
  "_id": "OgV25gcgRtuojyNrebAqLQ",
  "_source": {
    "id": "123",
    "nom": "Station 123",
    "location": [ -45.123, 48.123]
  }
}
```

3.3.1.2 Onglet Analyses

L'ongle analyse permet de visualiser sur une carte le nombre de montées et descentes aux arrêts du réseau. Les agrégations sont réalisées par date (30), heure (24), mode (2), typologie (13), produit (~100) et station (3050). Voici un exemple de document *json* stocké dans Elasticsearch pour une analyse :

```
analysis.json : {
    "_index": "aft",
    "_type": "analysis",
    "_id": "AU_3oBqmFjGrsMU8dtZb",
    "_source": {
        "date": "2013-03-01",
        "hour": 14,
        "typology": "11",
        "product": "CLR",
        "mode": "2",
        "station": {
            "stop_id": 3778,
            "id": "2;4620879.01;3778",
            "name": "ch. de Chambly et boul. Sainte-Foy",
            "mode": 2,
            "SRIDU": 4620879.01,
            "lat": 45.53088471,
            "lon": -73.48560518
            "location": [ -73.48560518, 45.53088471 ],
        },
        "day_of_week": 5,
        "CI": 3,
        "CO": 0
    }
}
```

3.3.1.3 Onglet Flux

L'onglet flux permet de voir sur une carte l'impact d'une région donnée avec le reste du réseau. On peut dessiner une zone géographique et on obtiendra tous les déplacements en lien avec cette zone. On peut choisir d'afficher tous les déplacements ayant une montée dans cette zone ou tous les déplacements ayant une descente dans cette zone. Pour ce faire, une agrégation est réalisée sur les champs date (30), heure (24), mode (2), typologie (13), station montée (3050), station descente (3050), type produit (~100).

Voici un exemple de document *json* stocké dans Elasticsearch pour un flux :


```
{
   "_index": "aft",
   "_type": "flux",
   "_id": "AU_3qzIYFjGrsMU8nmD9",
   "_source": {
      "date": "2013-03-05",
      "hour": 10,
      "mode": "2",
      "typology": "11",
      "product": "Billets O",
      "stationCI": {
         "lat": 45.49844029,
         "stop_id": 3382,
         "name": "TERMINUS CENTRE-VILLE",
         "location": [ -73.56683895 , 45.49844029 ],
         "SRIDU": 4620062,
         "id": "2;4620062.00;3382",
         "mode": 2,
         "lon": -73.56683895
      },
      "stationCO": {
         "lat": 45.46738183,
         "stop_id": 4429,
         "name": "TERMINUS PANAMA",
         "location": [ -73.46853832 , 45.46738183 ],
         "SRIDU": 4620825.04,
         "id": "2;4620825.04;4429",
         "mode": 2,
         "lon": -73.46853832
      },
      "count": 10,
      "day_of_week": 2,
   }
```


3.3.1.4 Limites

Le problème avec ces agrégations est que celles-ci ne sont pas adaptées au nombre de données disponibles pour le RTL. En effet, il peut y avoir respectivement jusqu'à 190 millions et 580 millions de combinaisons possibles d'agrégation d'Analyse et Flux. Il n'y a cependant que 3,1 millions de transactions de bus et de métro pour un mois de donnée au RTL. De plus, de telles agrégations masquent les attributs de la transaction brute telle que la distance parcourue en transport pour chaque déplacement ou le numéro des cartes. On ne peut pas analyser directement le comportement d'un usager-carte avec cette structure de données.

3.3.2 Elasticsearch et l'indexation des données

L'indexation des données est coûteuse en temps. Elasticsearch peut choisir d'interpréter automatiquement quels sont les types des différents champs d'un document (texte, entier, nombre décimal, point géo localisé, date, heure …). Cependant, cette fonctionnalité est à éviter afin d'être sûr de la typologie des champs des familles de documents et pour faciliter la tâche à Elasticsearch lors de l'indexation des données. Ainsi, il vaut mieux définir soi-même la structure d'un type de document. Pour ce faire, il suffit, lors la création d'un index, d'indiquer à Elasticsearch la structure des familles de documents que l'on va exporter par la suite vers cet index.

Pour des questions de performance, il est plus rapide d'utiliser la commande « batch » (commande en lot) d'Elasticsearch plutôt que d'envoyer une requête d'ajout pour chaque nouveau document.

Un équilibre doit être trouvé pour ne pas envoyer de trop gros fichiers à Elasticsearch. Un choix de 10 000 documents par lot a été considéré.

3.3.3 Chargement des données dans Elasticsearch pour le portail AFT

Pour les besoins du RTL, un algorithme a été développé pour réaliser les agrégations Analyses et Flux attendues par le portail AFT et ensuite les exporter vers Elasticsearch.

Deux jeux de données ont été générés. Un premier reprend directement les origines et destinations des transactions. Un second jeu de données reprend les origines et destinations des transactions, agrégées en séquences de tronçons. Chaque jeu de données dispose d'un script qui lui est propre permettant de le générer. Au moment de produire cet algorithme, l'algorithme destination n'avait pas encore calculé dès le début les séquences de tronçons. Le script se chargeant de générer le jeu de données pour les séquences de tronçons a dû déterminer les séquences de tronçons des déplacements en se basant sur les résultats consolidés de l'algorithme destination.

Une fois que les données d'origine et destination des déplacements sont obtenues, la logique de remplissage des agrégations « Analyses » et « Flux » est similaire, peu importe le jeu de données.

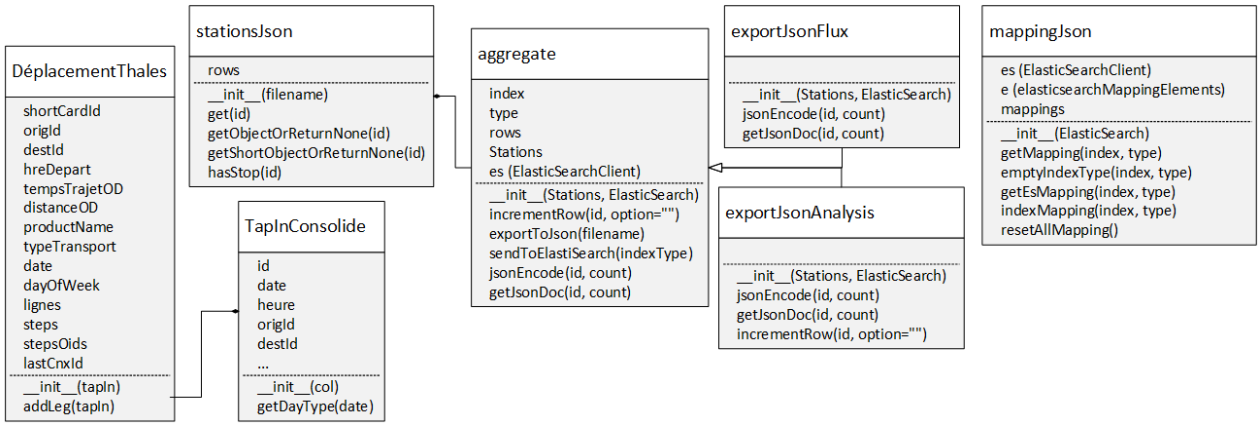


Figure 3-11 - Classes algorithme export et agrégation des transactions vers Elasticsearch

Pour faciliter la programmation, une architecture objet (Figure 3-11) a été développée.

L'objet *mappingJson* permet de lister les champs et leur type des documents Analyse et Flux et de les envoyer à Elasticsearch à chaque nouvel export de données. La librairie python *elasticsearch-py* (Python Elasticsearch Client, 2016) est utilisée pour communiquer directement avec le serveur Elasticsearch.

3.3.4 Limites et autres besoins de visualisation

Le portail AFT se base à l'origine sur des données d'origine et destination de déplacements effectués sur un réseau de transport en commun multimodal. Les séquences de tronçons empruntés lors de chaque déplacement ne sont pas connues ou affectées sur le réseau. Le portail AFT se concentre plutôt sur la visualisation générale de l'utilisation du réseau et de la mobilité de ses utilisateurs, ce qui était son mandat initial.

Lors des premières rencontres avec les spécialistes du RTL, nous avons réalisé que le portail AFT ne répond pas complètement aux besoins de la planification opérationnelle qui nécessitent pour chaque ligne du réseau des indicateurs ou des objets spécifiques tels que des profils de charge ou des diagrammes espace-temps. Ceci sera l'objet de la section suivante (3.4). De plus, le format actuel de données cache les statistiques propres à chaque trajet telles que les temps passés ou distances parcourues en transport en commun.

Pour rappel, l'onglet Analyse du portail AFT permet à l'aide de cartes de chaleurs propres aux montées ou descentes de donner un aperçu de l'utilisation du réseau. L'outil *Leaflet.markercluster* est utilisé pour regrouper les informations des arrêts entre eux. Il peut être intéressant de « forcer » l'algorithme d'agrégation des arrêts à tenir compte des obstacles naturels (voie de chemin de fer, fleuve …). De son côté, l'onglet Flux permet lui de s'intéresser à la mobilité des cartes entrant ou sortant d'une zone géographique sélectionnée en présentant avec une carte de chaleur les autres extrémités des déplacements entrant ou sortant en lien avec la zone choisie.

3.3.5 Nouveau format de données pour les transactions enrichies

Afin de mieux rendre compte des statistiques issues de l'algorithme destination, comme les distances ou temps passés en transport, l'architecture des données va être revue pour revenir à une forme désagrégée des déplacements effectués sur le réseau. Cette architecture désagrégée permettra de recomposer des statistiques pour chaque ligne, arrêt ou même utilisateur. Les propriétés des documents sont celles disponibles à la fin de l'algorithme destination (chapitre 3.2.5).

Dans un premier temps, on génère toujours deux ensembles de données vers Elasticsearch : un pour les tronçons et un autre avec les séquences de tronçons (déplacements). Un exemple de document est présenté à l'Annexe A. Ceux-ci seront ensuite analysés grâce à différents outils de visualisation abordés dans la section suivante.

Dans un second temps, l'algorithme destination est repris pour permettre d'exporter directement vers Elasticsearch les données des transactions consolidées de leur destination ainsi que des informations sur la séquence de tronçons (déplacement) à laquelle une transaction appartient. Il n'y aura ainsi plus besoin de générer deux ensembles car il sera possible d'afficher comme avant les informations liées au déplacement en faisant attention d'avoir un calcul groupant les résultats en fonction du nombre unique d'identifiants de déplacement. Cela équivaut à effectuer en SQL un GROUP BY *deplacementId*.

3.3.6 Adaptation de l'algorithme destination pour exporter directement les résultats vers Elasticsearch

Une nouvelle classe a été développée pour permettre de centraliser les interactions avec Elasticsearch dans un seul endroit. Cela permet au logiciel de gérer lui-même l'ajout de nouveaux documents au « batch » de documents qui lui est envoyé sans qu'il y ait besoin de connaître le type des documents, si ce n'est le type et l'index auquel envoyer ces derniers. L'objet *mappingJson* vu à la Figure 3-11 est toujours utilisé pour préciser à Elasticsearch la structure d'un document donné.

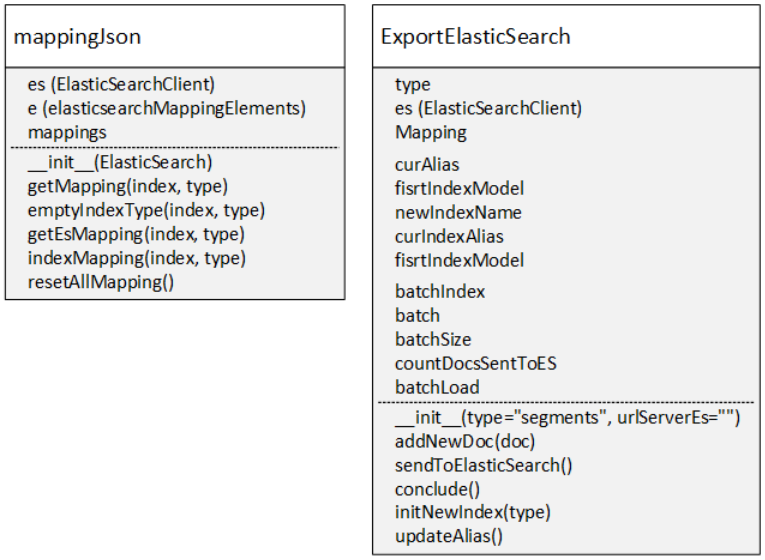


Figure 3-12 - Représentation des classes ExportElasticSearch et mappingJson

Désormais, il est possible d'entrer dans la configuration de l'algorithme si l'on souhaite exporter les résultats vers Elasticsearch. Si c'est le cas, il faut avoir précisé dans la configuration quel alias regarder. En effet, avec la version 2.x d'Elasticsearch, il n'est plus possible de modifier la structure d'un type de document. Si on souhaite le faire, il faut créer un nouvel index et recommencer

l'indexation des données. Cela permet d'éviter des incohérences dans les données qui ainsi sont obligées d'avoir la même structure. Pour permettre de mettre à jour un index en toute transparence, le tutoriel rédigé par (Gormley, 2015) conseille d'avoir un alias permettant d'avoir une façade d'un « index » pouvant être utilisé par exemple dans Kibana. On changera cet alias pour le faire pointer vers le nouvel index une fois le nouvel import de données effectué. Ainsi, les logiciels dépendant de cet index n'auront pas vu l'interruption de service. C'est ce que propose de faire la classe *ExportElasticSearch* si l'on souhaite importer un nouvel index ou reprendre depuis zéro l'import des données. Le nom du nouvel index sera *alias*_**v***XX*. Au besoin, l'algorithme peut continuer d'importer les nouvelles données dans l'index déjà présent.

3.4 Nouvelles formes de visualisation

Précédemment, l'algorithme destination a été appliqué sur les données de carte à puce du RTL qui ont été intégrées dans le portail web AFT permettant de visualiser les origines et destinations des déplacements des usagers du réseau.

On a vu dans la littérature que depuis ces dix dernières années des travaux de visualisations ont été effectués pour représenter ces données par le biais d'éléments plus avancés, comme par exemple un profil de charge ou un diagramme espace-temps dynamique en fonction d'une ligne donnée. On sait par ailleurs qu'un réseau de transport est composé de différents objets possédant chacun des propriétés et que les planificateurs ont besoin de s'y référer.

Cette section va présenter la méthodologie suivie pour produire de nouvelles visualisations. Dans un premier temps, la nouvelle version 4 de Kibana est utilisée. Dans un second temps, un nouvel outil web de visualisation va être développé afin de proposer des statistiques sur les différentes lignes du RTL ainsi qu'une vue globale de l'utilisation du réseau. Ces deux outils utilisent les données enregistrées précédemment dans Elasticsearch.

3.4.1 Kibana et l'exploration de données temporelles

Pour visualiser ces données, nous allons utiliser dans un premier temps une version plus récente de Kibana (v4). L'avantage que présente Kibana est de préparer des visualisations et tableaux de bord qui vont utiliser les données d'un index donné. Une fois que l'usager est satisfait des visualisations réalisées, il est alors facile de passer d'un ensemble de données à un autre depuis Elasticsearch. En

effet, comme précisé dans le chapitre 3.3.6, on peut indiquer un alias à Kibana et changer l'index de cet alias à souhait dans Elasticsearch. Ceci est transparent pour Kibana qui visualise sans problèmes un ensemble de données ou un autre. Cela permet de comparer différents scénarios dans la génération des jeux de données.

3.4.1.1 Rappel sur l'utilisation de Kibana et la génération de visualisations

Rappelons que Kibana est une application web construite en complément d'Elasticsearch pour analyser des données temporelles. Kibana, dans sa version 4, offre un outil d'analyse de données temporelles en trois étapes. On commence par visualiser sur une période choisie la répartition des données. Une fois une période intéressante repérée, on peut passer à l'onglet « Visualize » permettant de construire une visualisation que l'on paramètre soi-même et que l'on peut sauvegarder une fois satisfait. Le dernier onglet « Dashboard » permet d'afficher différentes visualisations côte à côte dans une même page. Toujours comme dans les versions précédentes de Kibana, en appliquant un filtre à un panneau, tous les panneaux du tableau de bord courant seront impactés par ce même filtre.

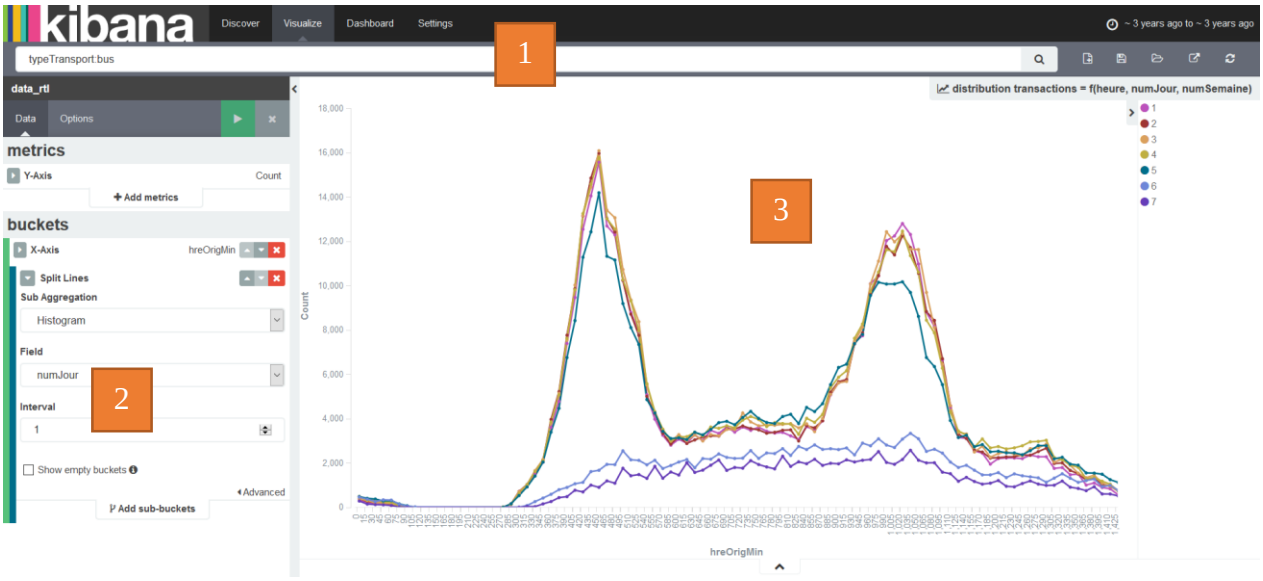


Figure 3-13 - Exemple de construction de visualisation dans Kibana v4

La Figure 3-13 est un exemple de visualisation créée dans Kibana. Cette figure montre aussi l'interface disponible pour générer des visualisations qui dispose de trois grandes sections :

1. Une barre de navigation qui permet de choisir une plage temporelle, de rajouter des filtres (ici on ne veut voir que des données de type bus, par exemple), de sauvegarder la visualisation …
2. Une section pour préparer les agrégations. Ici, pour un graphique à lignes, on choisit les valeurs en Y et X de ce dernier.
    - Pour les données selon l'axe des Y, on peut choisir entre différentes métriques d'agrégations telles que des comptes, sommes, moyennes, minimum, maximum ou encore des comptes uniques (tel un regroupement *GROUP BY* par le champ qui nous intéresse, par exemple par identifiant unique de séquence de tronçon). Attention : pour l'agrégation par compte unique, le nombre de valeurs est approximé avec Kibana, et des résultats « faux » peuvent être retournés. Pour y remédier, il faut rajouter en paramètre de cette agrégation le champ *precision_threshold* (Kibana - Cardinality Aggregation, 2016).
    - Pour les agrégations restantes (« buckets »), une attention particulière doit être portée sur le type d'agrégations choisies et la résolution de ces dernières. Cela implique de connaître les données disponibles. Par exemple, si l'on choisit une agrégation par histogramme selon la distance parcourue et que l'on indique un pas de 1, cela revient à demander à Kibana d'afficher un graphique avec pour résolution 1 mètre. Le navigateur ne va vraisemblablement pas réussir à charger la page. Il faudrait plutôt choisir une résolution de 500 m ou 1 km par exemple.
3. La dernière section est l'espace dans lequel la visualisation est générée. Il est possible d'interagir avec cette dernière. Dans l'exemple présenté ici, on peut voir la distribution du nombre de transactions par période de 15 minutes de la journée et en fonction du type de jour dans la semaine (1 : lundi, 2 : mardi, …, 7 : dimanche).

Pour plus d'informations sur les possibilités de Kibana en termes de visualisation et sur les autres types de visualisation, la documentation de (Kibana by Elastic, 2016) est disponible. Sommairement, Kibana propose différents types de visualisations : des graphiques à aires, à lignes, à barre, des diagrammes camembert, des cartes pour représenter des données géospatiales, des panneaux pour résumer différentes métriques, des tableaux où afficher des résultats bruts et enfin un panneau permettant d'afficher un texte statique en *markdown*.

3.4.1.2 Différentes catégories de visualisations

Les graphiques que l'on peut qualifier de base dans notre cas sont ici les distributions du nombre de transactions (par section, par déplacement) en fonction de différentes agrégations des champs des documents. Il n'y a pas de filtre ou d'imbrications appliquées. Ce sont des informations brutes et au besoin regroupées déjà en agrégations telles qu'un histogramme des valeurs possibles de ce champ.

On retrouve en premier toutes les agrégations temporelles : en fonction du jour dans la période temporelle sélectionnée, de l'heure de la journée, du numéro du jour dans la semaine, du numéro de la semaine dans l'année.

On dispose ensuite des autres champs disponibles dans le document tel que présenté précédemment (chapitre 3.2.5). On dispose alors des diagrammes camembert ou graphiques à barres pour les champs suivant : mode, type de transport, code OD, rangs des transactions dans la journée, nombres de tronçons par séquence de tronçons, rangs des séquences de tronçons dans la journée, type d'arrêt issu des travaux de Légaré, lignes, type de produit. On peut ensuite s'intéresser aux distributions des distances et temps parcours.

Il est intéressant de signaler qu'il sera possible par la suite de lier ensemble différents indicateurs. Il peut être considéré que les vues générales des documents constituent des points d'entrée qui permettent ensuite de réaliser des tableaux de bord encore plus poussés autour d'un champ donné.

3.4.2 Réalisation d'une nouvelle application web basée sur Elasticsearch

Kibana est pratique pour rechercher et trier des données temporelles et pour réaliser rapidement des graphiques sur une sélection de données choisies tout en permettant déjà des agrégations et des liaisons entre les différents champs des documents. Cependant, Kibana ne répond pas à tous les besoins.

Dans notre cas plus particulier et dans un contexte de planification opérationnelle, Kibana ne propose pas de panneaux ou d'outils spécialisés pour les besoins de visualisation des éléments d'un réseau de transport en commun.

Nous avons vu dans la revue de littérature des exemples de visualisations plus poussées, telles que la représentation d'un profil de charge pour une course donnée ou celle d'un diagramme espace-

temps pour une journée donnée de service. Dans ce mémoire, nous proposons des éléments permettant d'avoir une vue globale de l'utilisation du réseau ainsi que des statistiques sur les différentes lignes impliquées, intégrés sous le vocable *Visu Lignes*.

3.4.3 Tableau récapitulatif des lignes et de leurs tracés

Un premier tableau permet de mettre en avant différents résultats vers l'utilisateur. L'idée ici est de disposer d'un tableau récapitulatif interactif permettant de représenter les indicateurs et éléments graphiques propres à chaque ligne du réseau et au besoin de voir pour chaque ligne des indicateurs similaires propres à chaque tracé des lignes.

Les indicateurs et visualisations souhaités sont les suivants :

- Identifiant de la ligne et du nombre de tracés ou identifiant du tracé
- Nombre de transactions rattachées à cette ligne ou à ce tracé
- Proportion des transactions rattachées à un tracé aller, retour ou nul
- Proportion des transactions avec destination rattachées à un tracé aller ou retour
- Pourcentage des destinations retrouvées pour les transactions d'une ligne ou tracé
- Distribution des transactions par jour de la période temporelle choisie
- Distribution des transactions par heure de la journée de la période temporelle choisie
- Pour les tracés : profil de charge le long du tracé
- Schéma représentatif du parcours de la ligne ou tracé

Pour les profils de charge, il faut les recomposer en se basant sur les montants et descendant aux arrêts concernés. L'avantage de recomposer ainsi chaque profil de charge est que lors d'un filtre spatio-temporel donné, ce dernier sera limité aux transactions répondant à ce filtre.

L'avantage d'un tel tableau permet de répondre à plusieurs interrogations. On peut ainsi savoir quelles sont les lignes les plus achalandées du réseau ou encore quelles sont celles avec le meilleur taux de destinations retrouvées à l'issue de l'algorithme destination. On peut aussi savoir quand et à quelles heures ces lignes, ainsi que leurs différents tracés, sont utilisées. On peut aussi avoir un aperçu du profil de charge global des tracés. De plus, le tracé représentatif à l'aide des points clés du réseau (stations de métro, terminus de la ligne) permet de rappeler à un utilisateur non connaisseur du réseau de savoir où se situe la ligne grossièrement sur le territoire.

Une telle visualisation propose un index des lignes et tracés du réseau. L'utilisateur peut ensuite être redirigé vers une page détaillée qui reprendra des graphiques et indicateurs plus poussés tels

que le profil de charge détaillée ou le diagramme espace-temps du tracé choisi. Il est possible dès cette page à l'instar du portail AFT ou de Kibana de permettre à l'utilisateur de choisir une période temporelle d'analyse.

3.4.4 Carte des lignes et du réseau avec leurs charges respectives

La visualisation précédente a comme plus grosse limite le manque de vue globale des lignes sur le réseau. Pour ce faire, une carte interactive permettant de visualiser les différents éléments du réseau a été réalisée en plusieurs phases. Dans un premier temps, cette carte permet de visualiser la charge des tronçons de toutes les lignes du réseau. Dans un second temps, la fonctionnalité de filtre spatial, tel que proposé dans le portail AFT pour les Flux ou encore dans le portail Kibana dans les visualisations de type carte, a été ajoutée afin de permettre de visualiser les interactions entre certaines zones du réseau. Dans un troisième temps, les charges aux arrêts du réseau ont été représentées. Pour finir, une animation temporelle en fonction des heures a été implémentée pour les arrêts du réseau.

3.4.5 Technologies disponibles

Afin de réaliser une application web dynamique, plusieurs technologies, libraires sont disponibles pour faciliter la tâche et sont recensées au Tableau 3-8.

Tableau 3-8 - Récapitulatif technologies et librairies utilisées pour la réalisation de Visu Lignes

Création et structure page web	Outils de visualisation de données	Technologies côté serveur
HTML CSS JavaScript A B	Leaflet	php Slim elasticsearch.

Côté serveur, afin de sécuriser l'accès à des pages d'un site internet ou pour sécuriser Elasticsearch dans notre cas, il est possible d'utiliser différentes technologies. De nos jours, des solutions comme *Django* et *Python*, *Ruby on rails*, *Node JS* sont de plus en plus utilisées. Pour plus de simplicité, car maîtrisant déjà la technologie, le traditionnel PHP a été utilisé mais avec l'aide d'une librairie minimaliste, *Slim* en version 3, pour aider à la construction des routes.

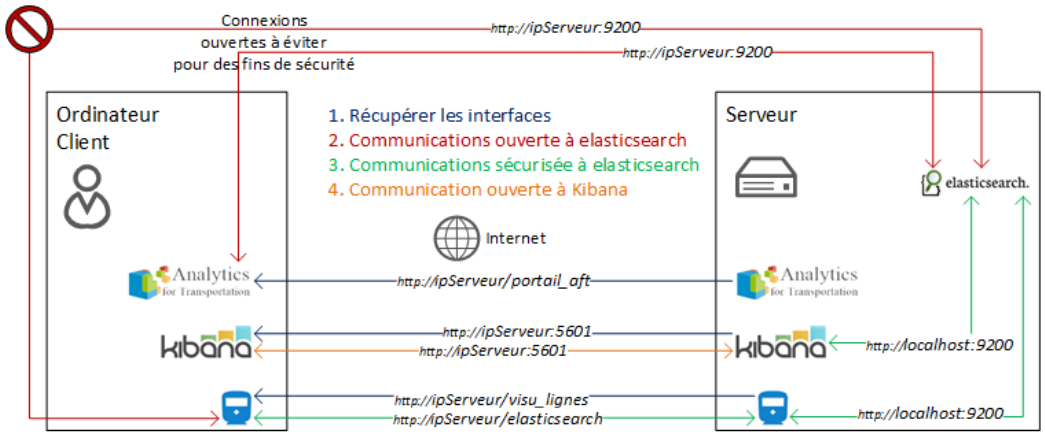


Figure 3-14 - Schéma accès et communications entre les visualisations – client / serveur

La Figure 3-14 présente les communications effectuées entre les applications ainsi que les adresses web (URL) à appeler. Il est nécessaire d'englober notre page HTML simple dans une application pour ne pas avoir à ouvrir « au monde entier » notre serveur Elasticsearch. En effet, si quelqu'un connaît l'adresse IP (Internet Protocol) de notre serveur et que notre serveur Elasticsearch est accessible depuis Internet, des personnes mal avisées pourraient en prendre le contrôle et altérer les données disponibles. Le portail AFT basé sur la version 3 de Kibana a besoin d'un serveur web et Elasticsearch accessible directement. La nouvelle version 4 de Kibana comprend directement un serveur web et les communications à Elasticsearch sont gérées par l'instance de Kibana côté serveur. Kibana offre déjà auparavant des solutions pour sécuriser l'accès à celui-ci et aux visualisations qu'il contient. Cependant, dans sa version gratuite, n'importe qui ayant accès à l'adresse web de Kibana peut modifier toutes les visualisations qui s'y trouvent. Pour ce faire, il n'est pas conseillé non plus d'avoir une instance de Kibana ouverte au monde entier. Un effort a été fourni dans *Visu Lignes* pour demander une authentification simple par courriel/mot de passe pour pouvoir accéder à la page web mais aussi pour rediriger les requêtes à Elasticsearch par l'instance côté serveur de *Visu Lignes*.

Côté client, comme pour Kibana et le portail AFT, nous allons nous appuyer sur des librairies ouvertes couramment utilisées dans le domaine :

- La librairie thème *Bootstrap* est utilisée pour aider dans le design et l'ergonomie de la page.
- *Angular JS* est utilisé pour aider à structurer la page

- *D3.js* est utilisé pour faciliter la création de graphiques et aide à la manipulation de la page pour réaliser des animations. *D3 (Data-Driven Documents)* va être utilisé ici surtout pour générer et contrôler des SVG
- *Leafleat* est utilisé pour gérer la cartographie
- Des tuiles d'*OpenStreetMap* sont utilisées pour le fond de carte

Côté performance, il convient de savoir que charger de nombreux éléments dans le *DOM* (Document Object Model) d'une page HTML est coûteux si cela est effectué en grande quantité. Cependant, il n'est pas aussi coûteux d'en changer les propriétés. Dans notre cas, nous disposons de près de 3000 arrêts de bus, et prêt de 9000 tronçons de ligne. De plus, nous disposons de 600 tracés et plus de 150 lignes pour lesquels nous souhaitons afficher pour chacun différents indicateurs et graphiques (distributions, profils de charge …). Cela fait beaucoup d'éléments à considérer. Une attention particulière doit donc être portée sur le sujet.

Nous entrerons dans les détails au chapitre suivant, qui présente des résultats d'implantation des outils développés.

CHAPITRE 4 EXPÉRIMENTATION ET RÉSULTATS

4.1 Résultats Algorithme Destination

L'algorithme destination dans sa dernière forme permet de retrouver les destinations de 1 963 483 transactions de bus sur 2 481 977 disponibles, soit un rendement de 79,2%. 618 868 transactions de métro effectuées par les utilisateurs du réseau ont été utilisées afin d'obtenir de meilleurs résultats. La Figure 4-1 recense les différents résultats des codes OD et permet de visualiser le rendement de ce dernier sans les transactions métro dans un diagramme de type camembert.

code	count	%	groupes	sous-groupes
11	1 145 542	37,0%	réussite	réussite dest. simple
12	561 286	18,1%	réussite	réussite dest. simple
13	65 323	2,1%	réussite	réussite dest. simple
21	141 112	4,6%	réussite	réussite dest. unitaire
22	50 220	1,6%	réussite	réussite dest. unitaire
30	3 833	0,1%	échec	échec
31	68 075	2,2%	échec	échec
400	182 726	5,9%	erreur	erreur arrêt vide
404	78 483	2,5%	erreur	erreur arrêt inconnu
41	84 443	2,7%	erreur	erreur lignes
412	13 460	0,4%	erreur	erreur lignes
42	36 174	1,2%	erreur	erreur lignes
43	48 129	1,6%	erreur	erreur lignes
45	618 868	20,0%	erreur	Métro

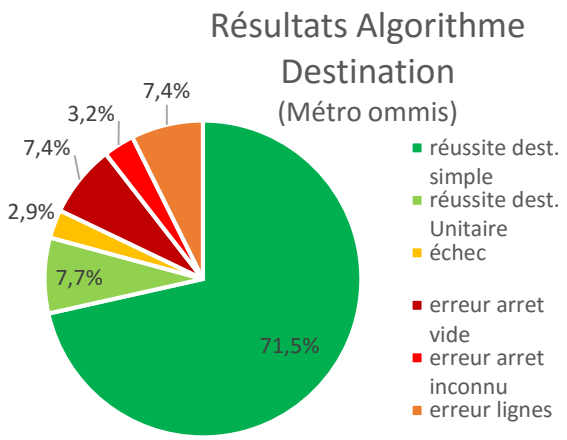


Figure 4-1 - Distribution des codes Origine Destination de l'algorithme destination

Le temps de correspondance choisi pour recomposer les séquences de tronçons (déplacements) repose sur une différence maximum de 60 minutes entre la première transaction et la dernière. 2 153 633 séquences de tronçons ont été recomposées à partir de 2 836 465 transactions, 264 380 ont été écartés pour cause d'arrêt inconnu. Parmi ces déplacements, la majorité des usagers l'effectuent avec un trajet direct (1 502 084 ou 70%). 620 765 ou 29% des déplacements disposent de deux tronçons ou encore d'une correspondance entre deux lignes du réseau. Seul 1% des séquences de tronçons disposent de trois tronçons ou plus.

L'algorithme destination a pris au total 20 minutes, soit une durée 300 fois plus rapide que l'algorithme original qui lui aurait pris plus de 100 heures. L'algorithme a passé 3,2 minutes (17%) à préparer les données, 2,5 minutes (13%) à retrouver les destinations simples, 4,3 minutes (23%) à retrouver les destinations unitaires et enfin 8,8 min (47%) à exporter les résultats vers Elasticsearch, et dans un fichier CSV (comma-separated values).

Pour les lignes, 2 061 868 de transactions ont réussi à « trouver » leur GTFS du premier coup. 21 652 transactions ont pu être récupérées car le match GTFS ne fonctionnait pas. Cependant, 97 903 lignes n'ont pas pu être retrouvées. En rajoutant les 264 380 transactions sans arrêts connus et les 36 174 transactions dont l'arrêt ne coïncide pas avec la ligne, on obtient bien de nouveau les 2 481 977 transactions de bus disponibles.

4.1.1 Un temps de calcul linéaire pour l'algorithme destination

Un des gros points faibles de l'algorithme de He était sa performance qui ne le rendait pas fonctionnel pour des jeux de données importants. On a vu que désormais l'algorithme est capable en 20 minutes de préparer et d'estimer les destinations des transactions pour toutes les transactions du mois de mars 2013 du RTL et pour exporter les résultats vers Elasticsearch pour l'autre.

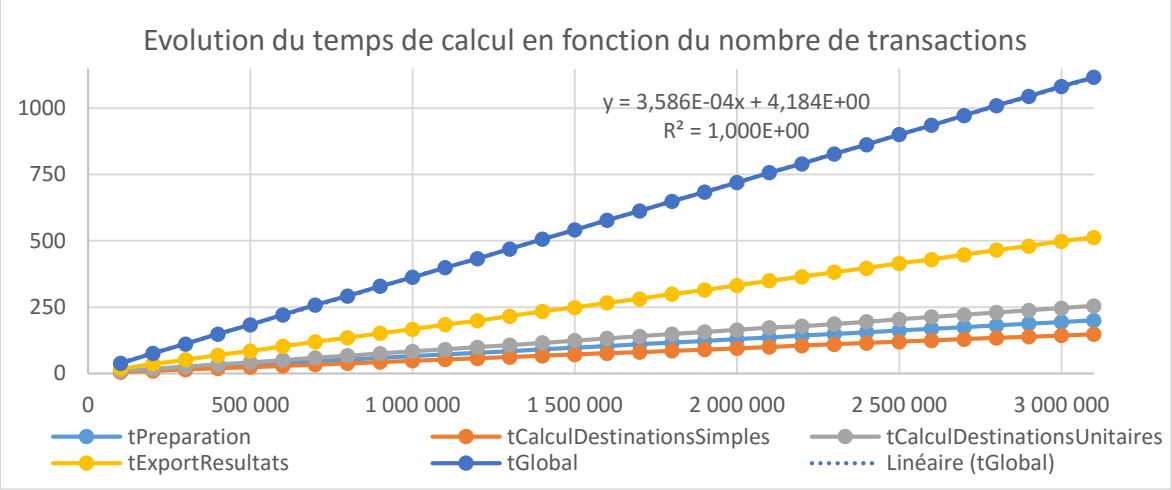


Figure 4-2 - Évolution du temps de calcul en fonction du nombre de transactions

La Figure 4-2 montre l'évolution du temps de calcul de l'algorithme au fur et à mesure de l'avancement de celui-ci. Les temps sont bien cette fois linéaires à tout moment, contrairement à la phase de préparation précédente qui était de forme polynomiale du second degré (Figure 3-3a).

4.1.2 Première analyse des données avec Access et Excel

Dans un premier temps, pour analyser les résultats de l'algorithme destination, les outils Access et Excel ont été utilisés. On importe le fichier de résultat consolidé dans Access, on effectue des requêtes SQL sur ces données et on exporte le résultat de ces dernières vers Excel. Cela a permis dans un premier temps de vérifier la cohérence des données. C'est grâce à ces outils que l'on a pu

découvrir une mauvaise gestion dans les fichiers GTFS des horaires qui allait au-delà de 24 heures. Ainsi, une attention plus particulière a été portée lors du développement de l'algorithme destination concernant la gestion des heures. Ces outils nous ont aussi permis de visualiser pour la première fois les distributions des distances et des temps passés par trajet sur les tronçons du réseau ventilé par code OD. La Figure 4-3 montre un pic très distinct dans la distribution des distances, mais qui n'apparaît pas dans la distribution des durées. Ce pic représente une particularité du réseau qui dispose de lignes permettant de traverser le fleuve Saint-Laurent par la voie réservée du pont Champlain. Ces trajets n'ont pas d'arrêt au milieu des ponts et sont en effet longs d'à peu près 11 kilomètres, ce qui correspond au pic observé.

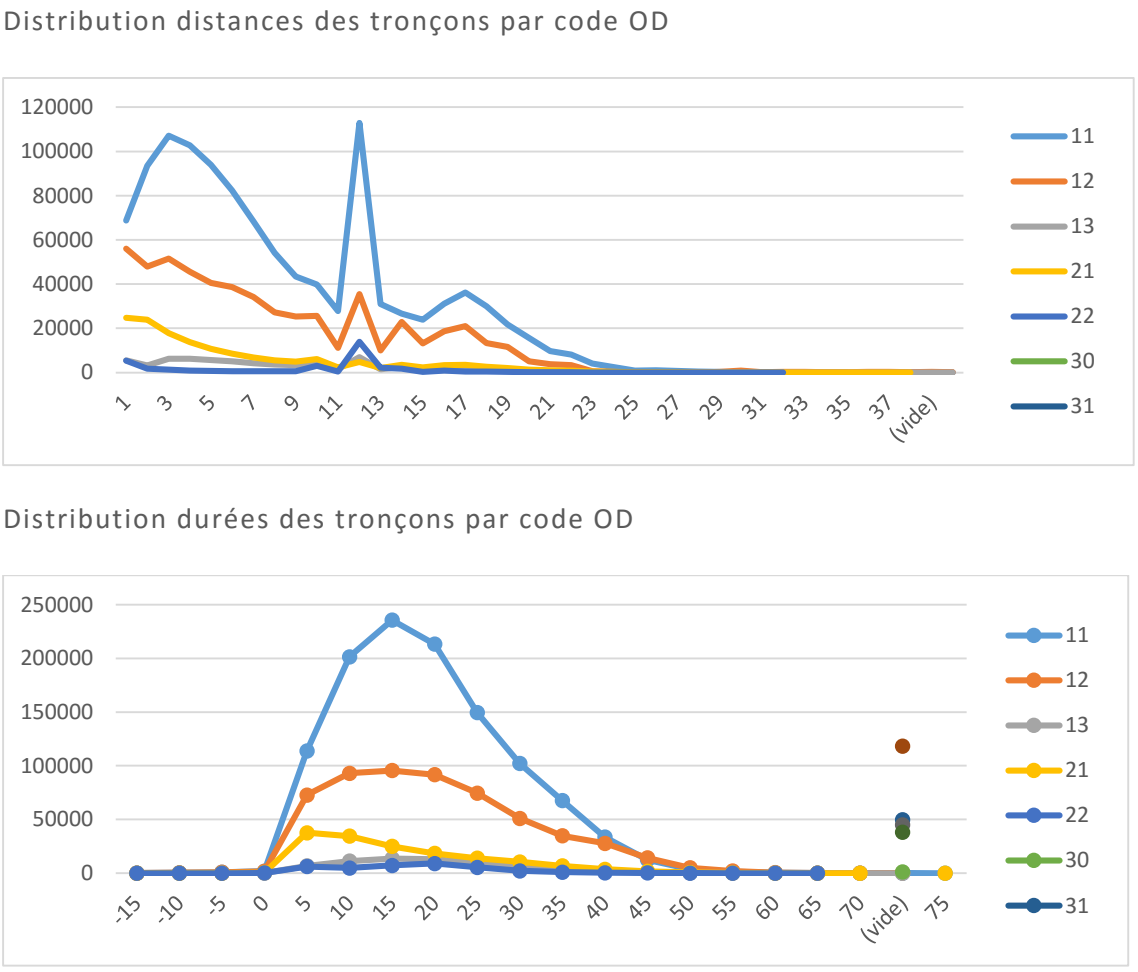


Figure 4-3 - Distributions des distances et durées des tronçons par code OD

D'autres premières visualisations rapides dans Excel ont permis d'apprécier la charge par jour et de repérer deux vendredis avec moins de trafic que d'habitude. Le premier s'explique par la présence d'un jour férié, le 29 mars 2013 (Vendredi Saint) et possède une charge (38%) proche des

samedis ou dimanches (25%) par rapport à la moyenne par jour de semaine. Le vendredi 8 mars dispose, lui, d'une charge inférieure de moitié aux jours habituels de la semaine (probablement un congé scolaire).

Une matrice OD par secteur de recensement a aussi été produite et transmise au RTL. En effet, les compagnies de transport de la grande région de Montréal utilisent ces divisions du territoire notamment pour recouper les résultats des enquêtes OD permettant de recenser la mobilité de la population. Le fait de générer ces matrices OD par secteur de recensement peut permettre au RTL de les comparer avec d'autres matrices OD du même type dont il dispose déjà.

Pour la suite des analyses, l'utilisation de Kibana et Elasticsearch permet d'éviter de devoir entrer les données dans Access, de traiter les requêtes SQL de façon manuelle et de devoir exporter les résultats de ces requêtes vers Excel.

4.1.3 Impact des transactions de métro disponibles

Un deuxième jeu de données a été généré pour permettre d'analyser la différence entre la présence ou non des transactions de métro. Pour finir, l'algorithme a tout de même réussi à retrouver 78,1% des destinations contre 79,2% précédemment.

code	count	%	groupes	sous-groupes
11	978 565	39,4%	réussite	réussite dest. simple
12	585 777	23,6%	réussite	réussite dest. simple
13	86 036	3,5%	réussite	réussite dest. simple
21	204 608	8,2%	réussite	réussite dest. unitaire
22	84 625	3,4%	réussite	réussite dest. unitaire
30	6 135	0,2%	échec	échec
31	89 645	3,6%	échec	échec
400	184 429	7,4%	erreur	erreur arrêt vide
404	79 951	3,2%	erreur	erreur arrêt inconnu
41	84 443	3,4%	erreur	erreur lignes
412	13 460	0,5%	erreur	erreur lignes
42	36 174	1,5%	erreur	erreur lignes
43	48 129	1,9%	erreur	erreur lignes
45	-	0,0%	erreur	Métro

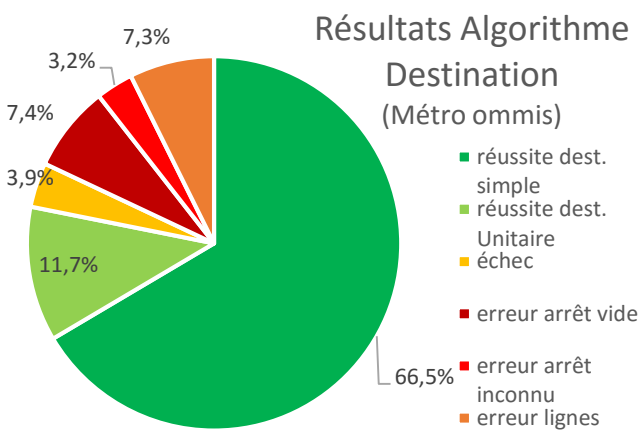


Figure 4-4 - Distribution des codes Origine Destination de l'algorithme destination – sans métro

4.1.3.1 Précision sur les parts de déplacements de type bus-direct, bus-bus, bus-métro, bus-bus-métro

Les différences entre ces deux jeux de données vont naturellement porter sur les types de déplacements. On retrouve 1 933 558 pour 2 481 977 transactions (on a 618 868 transactions de

métro en moins). Ces séquences de tronçons sont davantage sans correspondance (1 659 970 ou 86%), et désormais on a 263 474 séquences de tronçons (soit 14%) avec une correspondance. Cela est normal et s'explique par le fait que dans le jeu de données précédent beaucoup de transactions de bus effectuaient une connexion avec le métro.

Plus précisément, la Figure 4-5 permet deux graphiques avec à l'extérieur le nombre de tronçons dans un déplacement (1 : violet, 2 : rouge, 3+ : jaune) et à l'intérieur les modes (2 : bus seul : rouge foncé, 7 : bus vers métro : orange). Le graphique de gauche est le jeu de données avec données de métro (présence du mode 7) et le graphique de droite le jeu de données avec uniquement des transactions de bus. Il reste une part anecdotique de 9000 transactions de bus empruntant au moins 3 tronçons que l'on retrouve de nouveau dans le jeu de données avec transactions de métro (a.). Des 260 000 transactions du jeu de droite faisant une correspondance, a fortiori avec de nouveau du bus, on n'en retrouve que 240 000 (b.) correspondant aux mêmes critères (mode 2 et 1 correspondance). Cependant, on retrouve 20 000 transactions dans le mode 7 et ayant au moins deux correspondances représentant des personnes qui ont dû faire un transfert de bus avant de pouvoir prendre le métro. Enfin, le reste des transactions de bus ayant un mode 7 montre les transactions ayant une correspondance, mais cette fois avec le métro (toutes ces statistiques ne comprennent pas les correspondances effectuées au sein du réseau de métro).

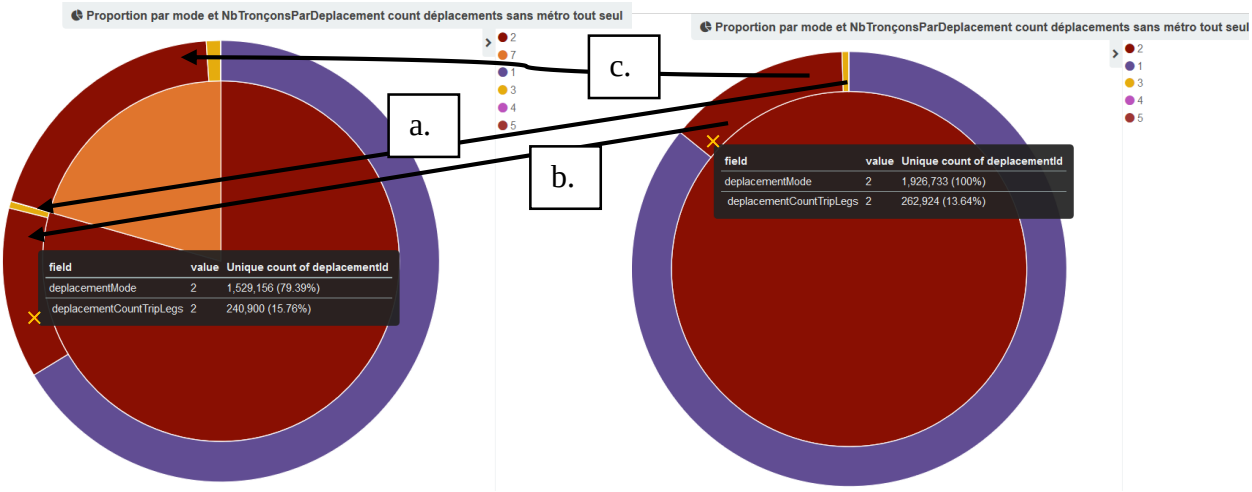


Figure 4-5 - Présentation de la part de trajets directs faits en bus, de correspondances en bus et de correspondances bus vers métro

Pour résumer, on a :

- 66% des trajets en bus sont directs entre l'origine et la destination finale
- 20% des déplacements effectuent une correspondance avec le métro
- 12% de déplacements effectuent une correspondance bus-bus
- 2% de déplacements effectuent au moins 2 correspondances, dont 20 000 transactions (1%) devant faire une première correspondance avec le bus avant de pouvoir prendre le métro

4.1.3.2 Évolution des codes OD

Une autre différence également en lien avec la remarque du paragraphe précédent est l'évolution de la proportion des codes OD. Les codes 11 (séquences de déplacements) vont être moins nombreux, car il y aura moins de correspondances effectuées dans le même jour à cause de l'absence des transactions métro. Ces codes 11 vont être transformés vers les autres codes OD de succès comme le montre le Tableau 4-1. Une grande partie est transférée aux cas de déplacements unitaires, d'où l'évolution de cette part des codes OD de 7,7 dans Figure 4-1 vers 11,7% dans la Figure 4-4. L'Annexe B montre aussi cette différence, mais cette fois-ci en utilisant un même tableau de bord chargé pour chaque jeu de données. Ce dernier permet de visualiser les différentes distributions des codes OD selon leur type : succès ou échec/erreur par heure, date ou juste globalement. Une particularité que montre ce tableau de bord en annexe est que la pointe du matin pour les codes d'erreur est à 8h contre 7h pour les codes de succès.

Tableau 4-1 - Évolution des codes OD entre un jeu de données avec et un sans transactions de métro

code	avecMétro	%avecMétro	sansMétro	%sansMétro	evolution	evolutionGlobale
11	1 145 542	46%	978 565	39%	-15%	-7%
12	561 286	23%	585 777	24%	4%	1%
13	65 323	3%	86 036	3%	32%	1%
21	141 112	6%	204 608	8%	45%	3%
22	50 220	2%	84 625	3%	69%	1%
30	3 833	0%	6 135	0%	60%	0%
31	68 075	3%	89 645	4%	32%	1%
400	182 726	7%	184 429	7%	1%	0%
404	78 483	3%	79 951	3%	2%	0%
41	84 443	3%	84 443	3%	0%	0%
412	13 460	1%	13 460	1%	0%	0%
42	36 174	1%	36 174	1%	0%	0%
43	48 129	2%	48 129	2%	0%	0%

4.1.3.3 Analyse comparative sur le rang dans la journée, le nombre de tronçons et le rang du déplacement

La Figure 4-6 permet de comparer les impacts de l'ajout ou le retrait des transactions métro pour les variables rang transaction, nombre de tronçons par déplacement et rang déplacement. La colonne de gauche présente le jeu de données comportant uniquement des transactions de bus, contrairement à la colonne de droite qui dispose, elle, des transactions de métro comme l'atteste la présence des codes 7. Les transactions de métro de ce dernier jeu ont été filtrées pour ne pas apparaître afin de pouvoir mieux comparer les transactions de bus entre elles. On peut percevoir grâce aux différents graphiques que les transactions de métro sont en majorité le matin. La première ligne de graphique le montre avec un nombre de transactions de rang 2 plus important en fin de journée. La seconde ligne le confirme aussi avec une nette baisse du nombre de déplacements avec une correspondance. Ce même graphique laisse apparaître d'ailleurs qu'il y a un pic dans le nombre de séquences de tronçons en deux étapes aux heures de pointe pouvant suggérer que l'utilisateur a effectué une correspondance sur le réseau du RTL avant de prendre une ligne lui permettant de traverser le Saint Laurent.

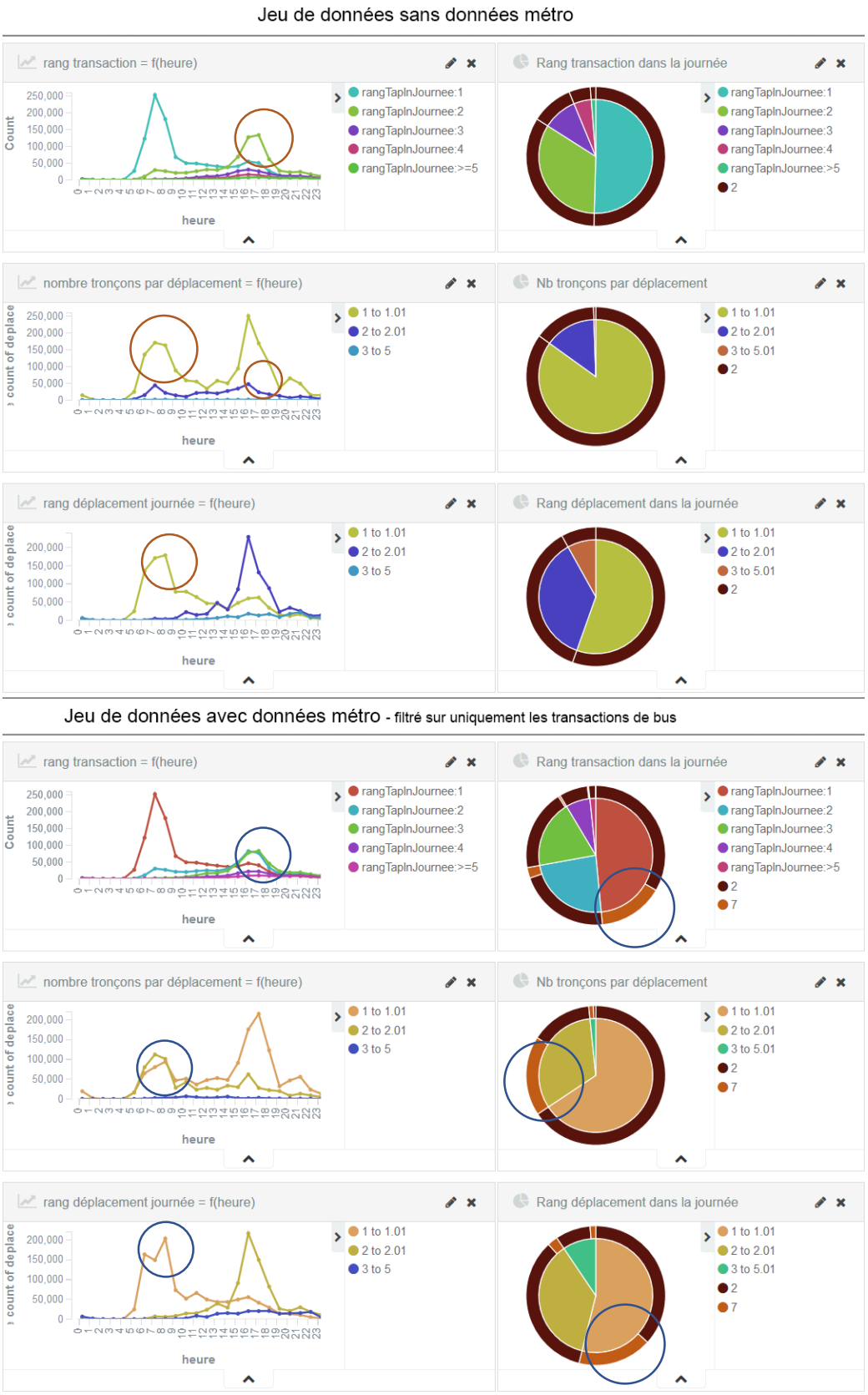


Figure 4-6 - Comparaison entre deux jeux de données – avec et sans transactions métro

Enfin, la dernière ligne montre que le rang de la séquence de tronçon dans la journée est bien similaire. Pour rappel, le mode 7 signifie d'une part pour une transaction de bus qu'elle de bus réalise une correspondance avec le métro. D'autre part, ce même mode 7 signifie pour une séquence de tronçon qu'elle comporte une transaction de bus et une de métro. On peut voir que les transactions ou séquences de tronçons de mode 7 sont bien réalisées en grande majorité en tout début de journée et font bien partie de déplacements à au moins deux tronçons. Cela confirme la remarque précédente montrant que les utilisateurs se dirigent vers Montréal et le métro en début de journée. Il faut noter que l'on ne dispose pas des transactions de métro pour les déplacements de retour.

4.1.4 Impact du temps de correspondance sur les séquences de tronçons

Pour en tester la sensibilité, on a changé le temps de correspondance pour deux jeux de données, mais les résultats de l'algorithme destination ne changent pas car ils ne sont pas sensibles à cet indicateur. Il en va de même pour l'estimation du rang des transactions dans la journée. Cependant, la différence intervient au niveau des séquences de tronçons.

Tableau 4-2 - Évolution du nombre de tronçons dans les séquences de tronçons et du nombre d'erreurs lors de la modification du temps possible de correspondance de 60 à 90 min

code d'erreur	60min	90min	%	%global	#tronçons	60min2	90min2	%global2
mêmeOrigine	7434	8873	19%	0,1%	1	1502084	1466814	-1,6%
mêmeLigne	33428	47429	42%	0,7%	2	620765	626414	0,3%
arretDejaUtilisé	124	363	193%	0,0%	3	30293	36983	0,3%
ligneDejaUtilisée	2886	6746	134%	0,2%	4	483	1378	0,0%
tropLoinDansLeTemps	113915	111554	-2%	-0,1%	5	8	70	0,0%

Le Tableau 4-2 nous permet de bien voir que le nombre de séquences de tronçons a augmenté en termes de déplacements effectuant des correspondances. Ceci est confirmé par la Figure 4-7 à la première ligne qui permet de montrer qu'à la pointe de matin, on a plus de déplacements avec une correspondance. Cependant, ce même Tableau 4-2 nous indique que 1% des 1,5 million de déplacements a aussi disparu. On les retrouve dans les codes d'erreur, avec certes moins d'erreurs pour des transactions trop lointaines dans le temps mais répercutées sur les autres cas d'erreur, ce qui souligne l'importance de rajouter de telles conditions. Ces erreurs se voient dans la Figure 4-7 avec une perte de déplacements dans la pointe de l'après-midi.

Figure 4-7 - Comparaison entre deux jeux de données – variations du temps de correspondance

Cela montre d'autant plus l'importance de ne pas se fier uniquement au temps de correspondance tarifaire mais bien plutôt d'essayer d'aller chercher précisément, en s'appuyant sur le GTFS et les

temps estimés d'arrivée aux arrêts tel que proposé par (Chu & Chapleau, 2008), ou d'appliquer une méthode géométrique comme celle de (Munizaga & Palma, 2012).

4.2 Autres cas d'utilisation de Kibana

Des graphiques tirés de Kibana ont déjà été présentés à la section précédente (Figure 4-6 et Figure 4-7) pour appuyer les analyses des résultats de l'algorithme destination. Un tableau de bord a été créé pour visualiser ces graphiques pour le jeu de données des transactions enrichies. Cela illustre qu'il est bien facile de changer un alias dans Kibana afin de comparer différents jeux de données et d'obtenir des visualisations identiques propres à chacun.

Cette section montre d'autres cas d'utilisation de Kibana ainsi que des exemples de visualisations ou tableaux de bord permettant de mieux extraire l'information contenue dans les trois millions de transactions.

4.2.1 Vue globale des données

La Figure 4-8 est un tableau de bord permettant d'obtenir une vue globale des données du réseau. Certains de ces panneaux (F) ont été repris et présentés précédemment (Figure 4-6 et Figure 4-7). Dans le cas de la Figure 4-8, un filtre spatial a été appliqué (A) sur les cartes tout en bas pour avoir les tronçons ayant comme destination la zone du métro Longueuil-Université de Sherbrooke (Longueuil-UDS). Comme le filtre s'applique sur les arrêts de destination des trajets, on a comme résultat uniquement des transactions disposant d'un arrêt de destination non vide et situé dans la zone sélectionnée. C'est pour cela que l'on ne voit que des modes 2 et 7 dans les visualisations (D, E, F) mais c'est aussi pour cette raison que les indicateurs montrant le pourcentage de réussite (E, E') sont tous à 100%, car seules des transactions avec destinations ont été sélectionnées.

On retrouve dans les produits les plus importants (C) beaucoup de cartes TRAM3 ou supérieur car ce titre permet de se rendre sur l'île de Montréal. Les graphiques en E et F indiquent que les personnes se dirigent vers le métro Longueuil-UDS le matin pour se rendre sur l'île de Montréal. Cependant, il existe tout de même une part d'usagers moins importante qui effectue le trajet inverse en se rendant sur la rive-sud le matin et qui rentrent à Montréal en fin de journée. On peut le voir sur une petite pointe en soirée dans les graphiques temporels ou par le nombre de destinations sur le quatrième panneau en bas à droite.

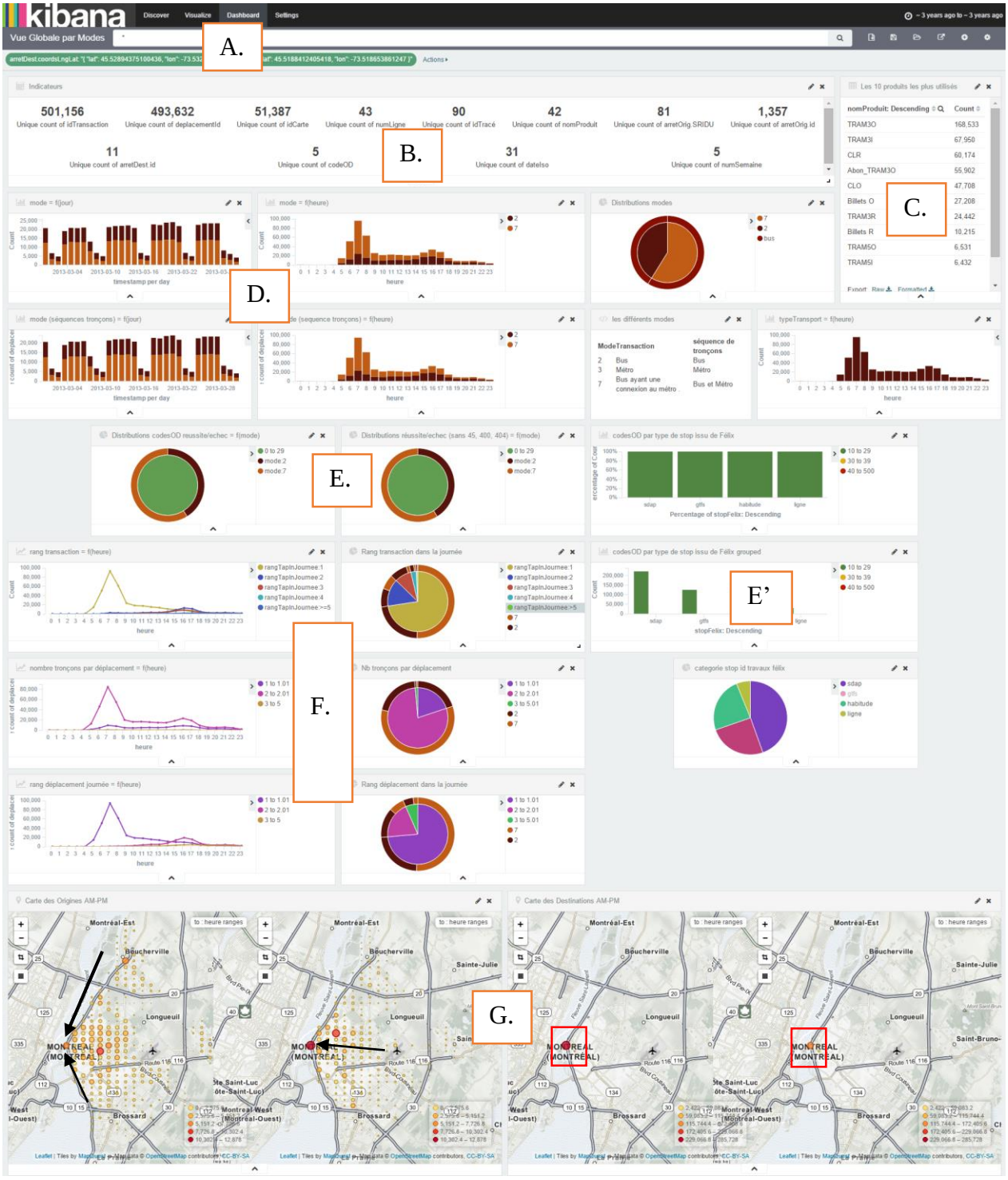


Figure 4-8 - Vue globale des données RTL – Filtre spatial : arrivées au métro Longueuil-Université de Sherbrooke

4.2.2 Impact des travaux de Légaré

Figure 4-9 - Graphiques distributions méthode d'attribution de l'arrêt d'origine des transactions

La Figure 4-9 permet de visualiser sur l'ensemble des données de bus les distributions des différents modes par lesquels (Légaré, 2014) a obtenu la localisation des transactions. Les taux de réussite d'estimation de destination dépassent 80%. Cependant, ce taux varie en fonction de la manière avec laquelle les arrêts ont été retrouvés lors des travaux de Légaré (section 2.2). Le nombre de destinations estimées pour les arrêts retrouvés à l'aide du GTFS est de 95% contre 84% pour les autres méthodes. Cela est dû en partie à cause de l'attribution par Légaré d'identifiants d'arrêts du SDAP et non du GTFS.

4.2.3 Vue d'ensemble des distances et temps moyens de parcours

Kibana présente l'avantage de pouvoir facilement rechercher et trier ses données. Dans un tableau de bord, on peut interagir avec les panneaux et choisir de nouveaux filtres qui vont se répercuter sur les autres éléments de la page. La Figure 4-10 présente ici deux semaines du mois de mars 2013 avec un attention particulière portée sur les distances et temps de parcours.

Cette Figure 4-10 dispose de plusieurs panneaux. On retrouve sur la première ligne des panneaux d'ordre général sur les données. Les panneaux de type A proposent de visualiser les distances et temps moyens mis par les usagers en fonction de l'heure ou de la date. On peut y voir deux pics lors des pointes du matin et du soir, tant pour la distance que le temps de trajet, laissant entendre que les usagers parcourent de plus longues distances à ces heures. De même, les déplacements effectués le samedi et le dimanche sont moins longs que ceux de semaine. La carte D représente la distance moyenne des déplacements se terminant aux arrêts du réseau. On observe que les déplacements sont de plus en plus longs lorsque l'on s'éloigne du métro Longueuil-UDS ou lorsque l'on traverse le Saint-Laurent, rappelant de nouveau la grande attractivité du centre-ville de Montréal. Les panneaux B et C présentent eux respectivement les distributions des tronçons en

fonction de leur distance ou temps de trajet. On retrouve le pic à 11 km pour les traversées du Saint-Laurent par le corridor du pont Champlain.

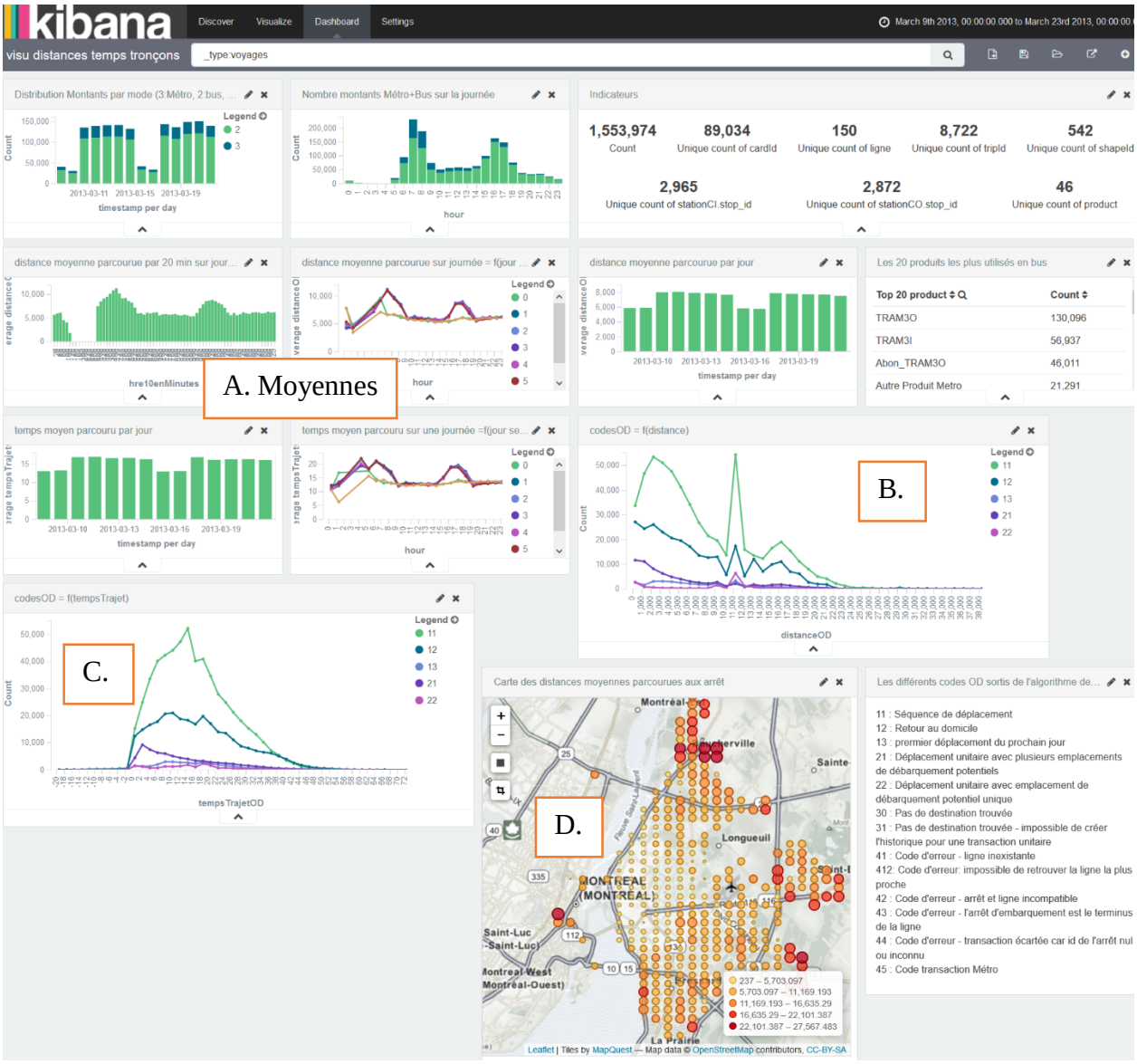


Figure 4-10 - Vue d'ensemble des distances et temps moyens de parcours

4.2.4 Différence entre le mode d'une transaction et celui d'un déplacement

Dans le dernier format de données, un document de transaction dans Elasticsearch comprend aussi les informations de la séquence de tronçons (déplacement) à laquelle elle appartient. Les figures suivantes vont montrer les différentes significations d'agrégations possibles sur le mode de la

transaction ou du déplacement ainsi que la différence dans l'agrégation pour la métrique entre un compte normal ou un compte unique par identifiant de déplacement.

La Figure 4-11 présente dans la partie A la distribution du nombre de transactions en fonction de l'heure (par pas de 20 min), du nombre de tronçons de la séquence de tronçons à laquelle elle appartient, enfin le mode de ces transactions. La partie B représente la même distribution mais cette fois par séquence de tronçons (« Unique count of deplacementId ») et par le mode de ces déplacements.

Pour rappel, on a choisi de catégoriser les transactions en trois modes :

- 2 pour une transaction de bus simple;
- 3 pour une transaction de métro;
- 7 pour une transaction de bus qui effectue une correspondance avec le métro.

Ces modes ont été repris pour marquer les séquences de tronçons :

- 2 pour une séquence de tronçons composée uniquement de transactions de bus;
- 3 pour une séquence de tronçons composée uniquement de transactions de métro;
- 7 pour une séquence de tronçons composée de transactions de bus et de métro.

Cette Figure 4-11 permet de constater que les transactions de métro sont bien en majorité effectuées le matin comme pressentis au chapitre 4.1.3 (voir seconde ligne de la partie A : répartition des transactions de métro en fonction de l'heure de la journée). De plus, ces deux graphiques mettent en lumière le transfert que l'on peut voir entre les courbes et la différence entre une métrique sur le nombre de transactions et une sur le nombre unique de déplacements.

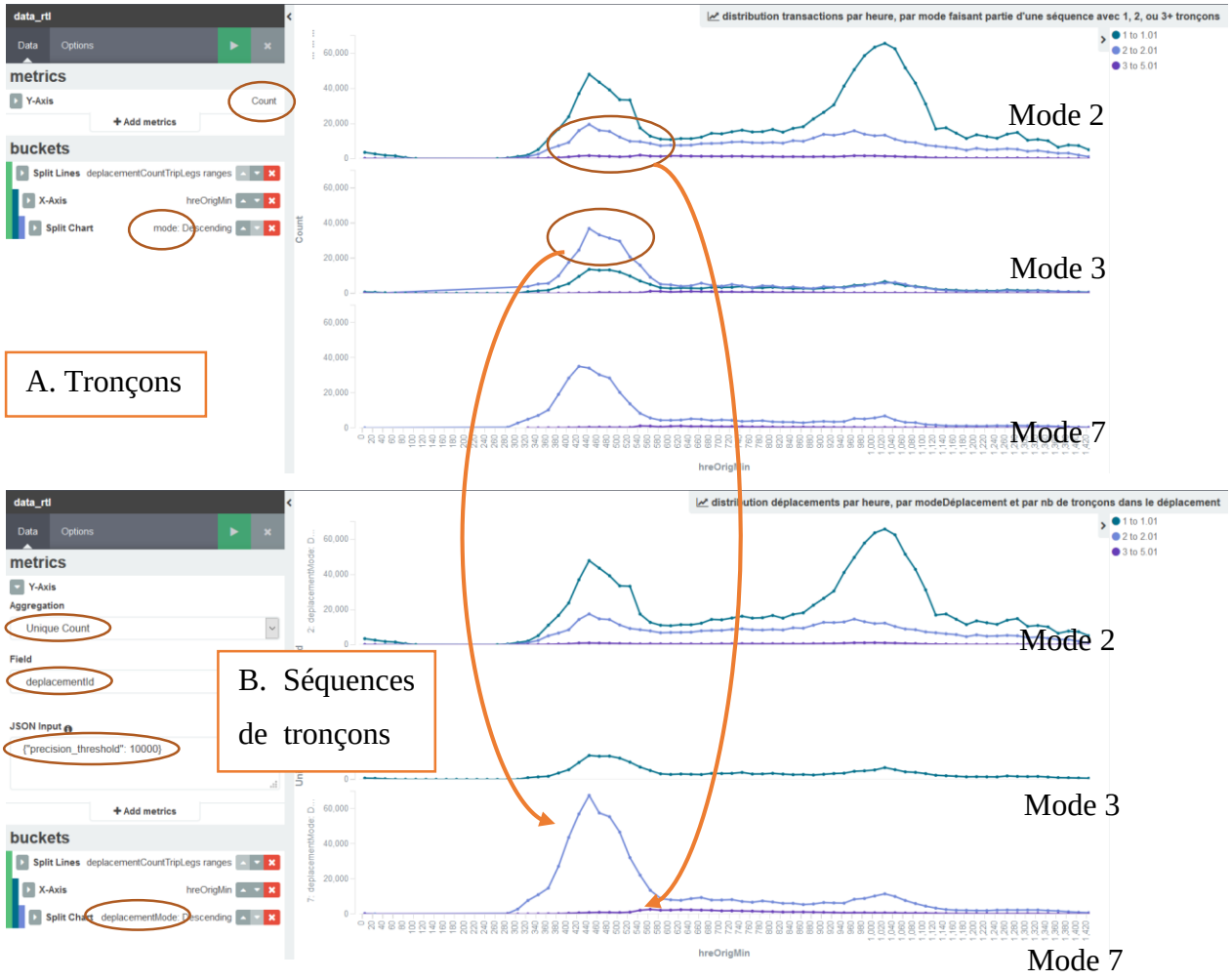


Figure 4-11 - Comparaison entre le nombre de transactions et le nombre de séquences de tronçons ; les deux en fonction par heure, par nombre de tronçons les composant et par mode

La Figure 4-11 a montré l'impact et la différence de signification sur la métrique d'agrégation pour l'axe des Y (compte des transactions ou compte des identifiants uniques de déplacement). Il faut faire attention au champ à sélectionner dans la sous-agrégation. La Figure 4-12 le rappelle en montrant la différence pour une agrégation sur l'identifiant unique d'un déplacement mais en changeant les champs de la sous-agrégation du mode des transactions d'une séquence de tronçons au mode même de la séquence de tronçon. On peut y voir que les transactions de métro de mode 2 se retrouvent en grande majorité dans des séquences de tronçons de mode 7. De plus, le graphique de gauche recense plus de déplacements car il affiche le nombre de modes différents contrairement à celui de droite qui n'aura plus qu'un seul mode pour un déplacement.

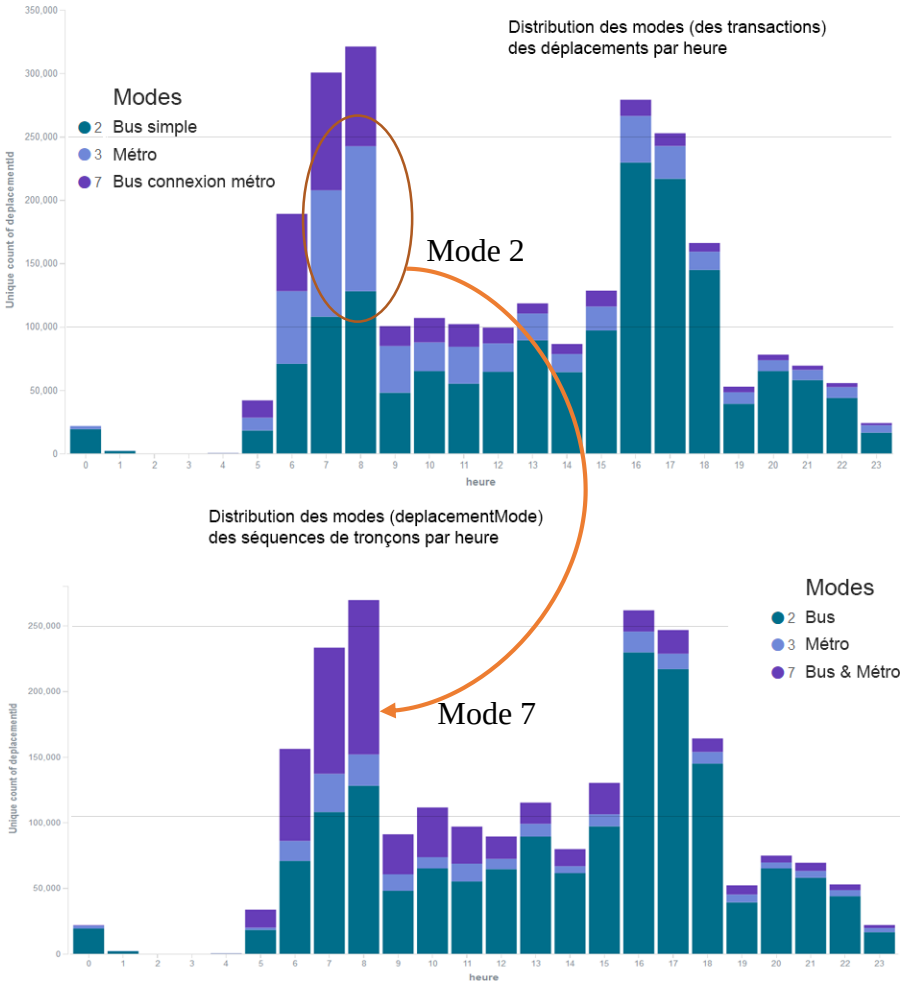


Figure 4-12 - Impact et différence du champ mode dans une sous agrégation

4.2.5 Détection d'erreurs (ligne 90)

Cette section présente un cas d'utilisation de Kibana pour détecter et chercher à comprendre la source d'erreurs levées par l'algorithme destination. Comme vu lors du chapitre 4.1, il existe encore une grosse partie de transactions qui ne peuvent être résolues.

Dans le cas du réseau du RTL, comme le relate la section 4.1.2, celui-ci a la particularité d'avoir des lignes de bus traversant le Saint-Laurent, principalement via le pont Champlain. La Figure 4-13 est un tableau de bord permettant de visualiser les transactions importées pour la seconde fois dans Elasticsearch sous forme désagrégée. Toutes les transactions ici ont été importées, quel que soit leur code d'erreur comme l'indique le compte du nombre de documents à 3,1 millions. Ce dernier nous indique, grâce au panneau 1, les trois lignes possédant le plus de transactions. On y retrouve trois lignes : la 8, la 45 et la 90.

Tableau 4-3 - Tracés des lignes 8, 45 et 90 – tiré de AMT

Ligne 8	Ligne 45	Ligne 90
		

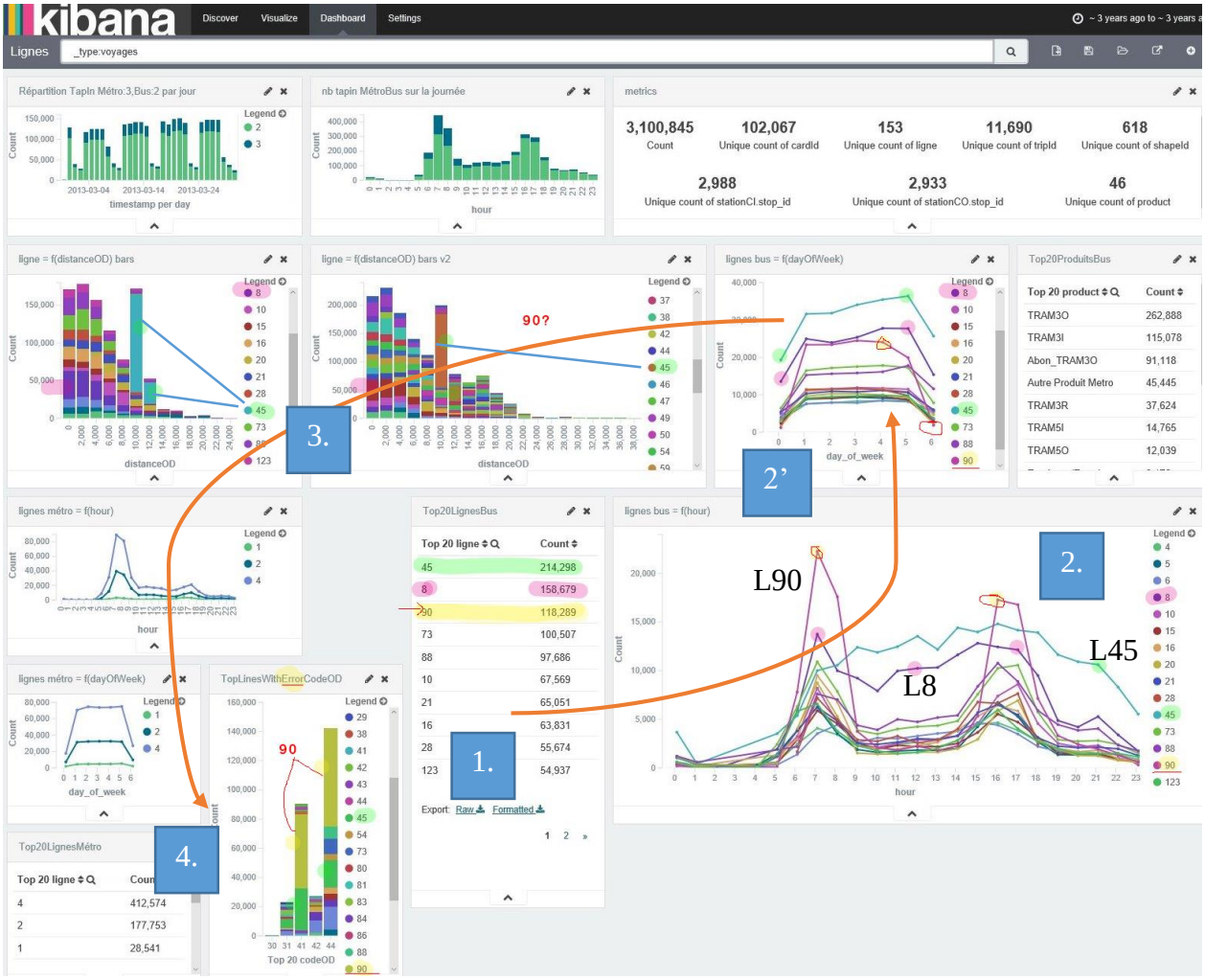


Figure 4-13 - Tableau de bord récapitulatif des lignes du réseau

Le Tableau 4-3 recense les tracés de ces lignes. La ligne 8 est une grande ligne permettant de se rendre au métro Longueuil-UDS. Les lignes 45 et 90 sont-elles des lignes express pour traverser le pont Champlain afin de se rendre à la station de métro Bonaventure (terminus centre-ville). Grâce

au panneau 2 de la Figure 4-13, on peut voir que les lignes 8 et 45 sont plutôt constantes tout au long de la journée en termes d'utilisation et sont aussi fortement utilisées le samedi et le dimanche (panneau 2'). On voit en plus la ligne 90 apparaître mais celle-ci est utilisée principalement aux heures de pointe et pendant la semaine.

Le panneau 3 de la Figure 4-13 montre quelles sont les lignes principales pour une tranche de distance de trajet donné. On retrouve le pic des 10-12 km correspondant aux lignes traversant le pont Champlain. Cependant, on ne peut y retrouver que la ligne 45. La ligne 90 est en effet absente.

Une recherche dans l'onglet « Découvrir » de Kibana, en filtrant sur la ligne 90 (Figure 4-14), permet de retrouver les 118 289 transactions effectuées sur cette ligne. On y voit que les codes de ces dernières sont principalement 44 pour arrêt inconnu et 41 pour erreur chargement de la ligne.

C'est pour cela que le panneau 4 de la Figure 4-13 a été ajouté. On y visualise que la ligne 90 est bien présente et majoritaire dans les cas d'erreur. En fait, la ligne 90 n'est simplement pas codifiée dans les fichiers GTFS disponibles de mars 2013, car celle-ci était opérée à l'époque par l'AMT et ne faisait donc pas partie de l'ensemble GTFS du RTL. L'outil de visualisation a permis de détecter cet oubli.

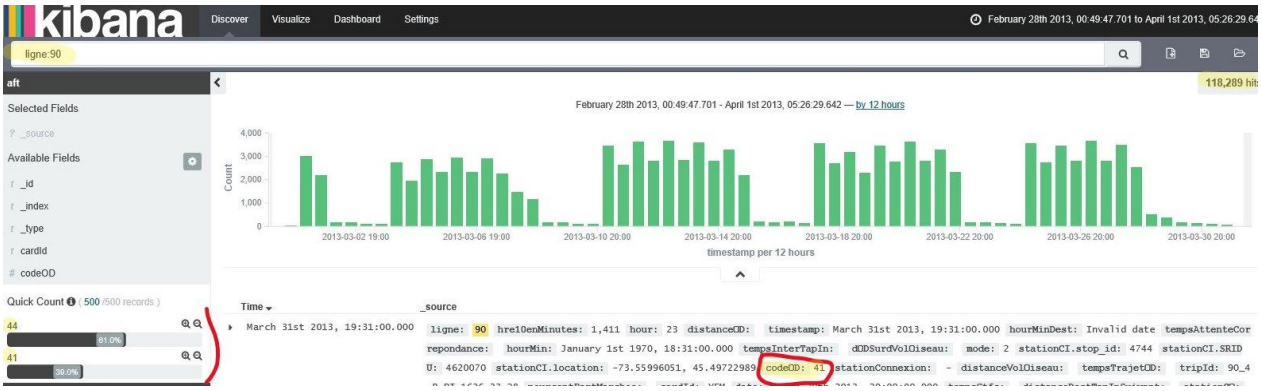


Figure 4-14 - Découverte des transactions de la ligne 90

4.2.6 Analyse comparative sur plusieurs semaines

Un aspect non évoqué jusqu'à présent est la comparaison sur plusieurs semaines de certaines caractéristiques liées à l'utilisation du réseau. Il est bien possible de naviguer et de générer un tableau de bord pour une semaine donnée puis de le recharger pour visualiser une autre semaine ou période. Cependant, cela nécessite de le faire manuellement.

La Figure 4-15 présente les 5 semaines du mois de mars 2013 ainsi que la charge globale par jour. On observe que les vendredis de la seconde semaine (2e ligne) et celui de la dernière semaine (5e ligne) sont bien en dessous des autres journées des semaines. On peut apercevoir que le vendredi a une tendance à avoir moins de transactions comparé aux autres jours de la semaine (lignes 3 et 4).

Figure 4-15 - Évolution au fil des semaines de la charge globale par heure de la journée

4.3 Visu Lignes

Cette section présente la nouvelle interface *Visu Lignes* développée pour visualiser l'utilisation du réseau de transport en commun. Pour ce faire, l'interface propose sur une même page web un tableau récapitulatif d'indicateurs et des figures-clés des lignes du réseau s'appuyant sur une carte qui permet de visualiser l'ensemble des charges sur les arrêts et tronçons du réseau. La page propose également de sélectionner une plage temporelle entre deux dates données. Une sélection spatiale est aussi permise afin d'analyser les interactions du réseau avec une zone donnée ou encore pour visualiser les flux entre deux zones données. Ces filtres se répercutent sur tous les éléments de la page.

4.3.1 Présentation vue globale du réseau, et de Visu Lignes

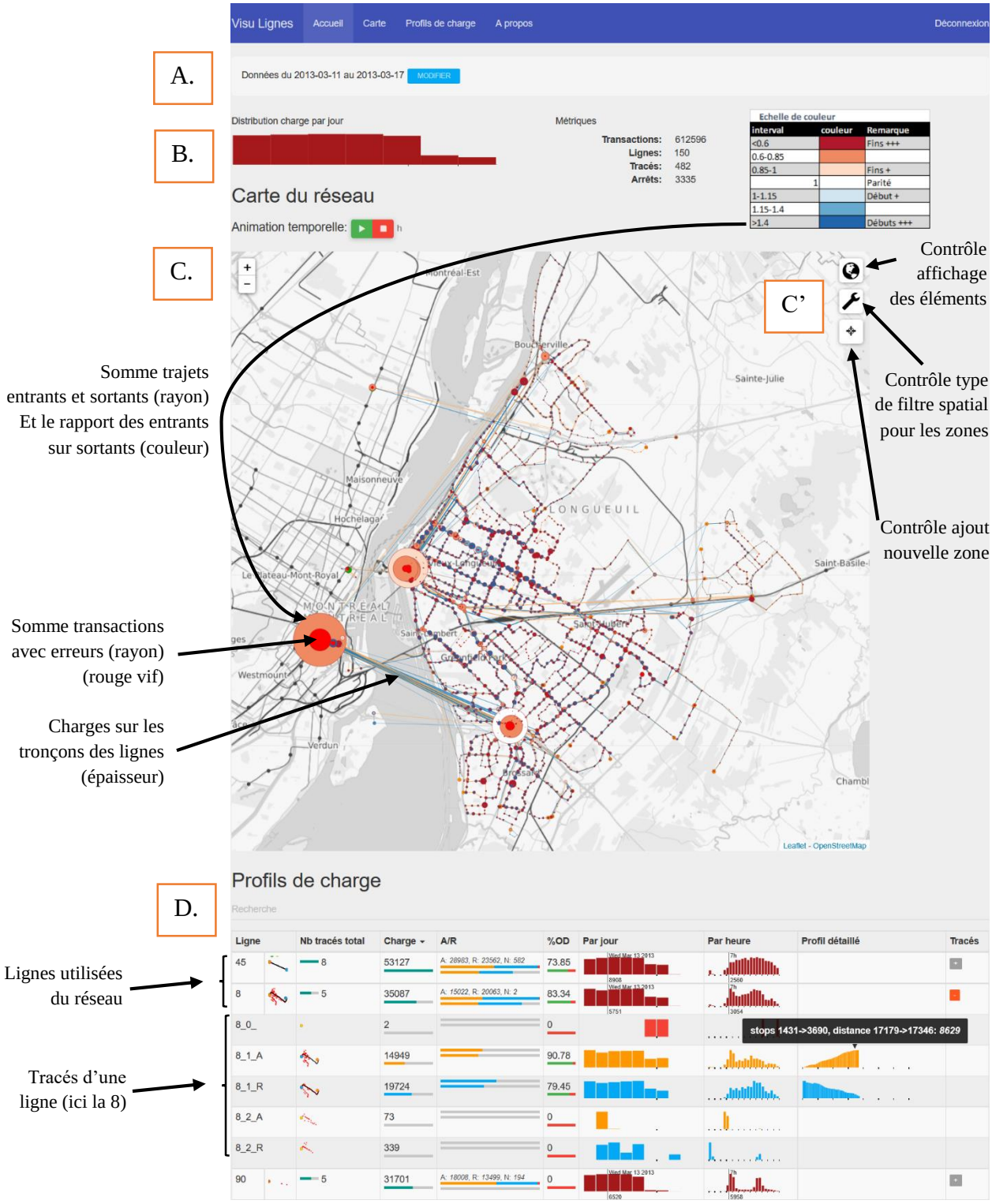


Figure 4-16 - Vue globale du réseau grâce à l'interface Visu Lignes

4.3.1.1 Présentation et fonctionnement

La Figure 4-16 présente l'interface Visu Lignes. On y retrouve en premier lieu (A) une section interactive pour sélectionner une période temporelle entre deux dates (ici, la semaine du 11 au 17 mars 2013). Ensuite, une seconde section (B) permet de recenser quelques indicateurs clés sur la sélection spatio-temporelle (ici temporelle uniquement) du jeu de données de transactions de bus disponible. Les transactions de métro n'ont pas été représentées ici.

La section (C) est la carte interactive qui permet de visualiser les charges sur les tronçons des lignes du réseau (par l'épaisseur du trait), la charge cumulée des entrants et sortants (par la taille du symbole) aux arrêts avec le rapport entre les deux (échelle de couleur), ainsi que le nombre d'erreurs par arrêts (aire). Des contrôles (C' et Figure 4-17) permettent de choisir quels éléments afficher (C1) ainsi que de rajouter des zones spatiales (C3) pour filtrer les données (C2). En ayant placé au moins une zone sur la carte, il est possible de visualiser les transactions ayant au moins un départ ou une arrivée dans la zone (C2a), ce qui sera présenté à la section 4.3.2. En créant deux zones ou plus sur la carte, il est possible de choisir de visualiser les interactions entre ces deux zones (C2b), voir section 4.3.3.

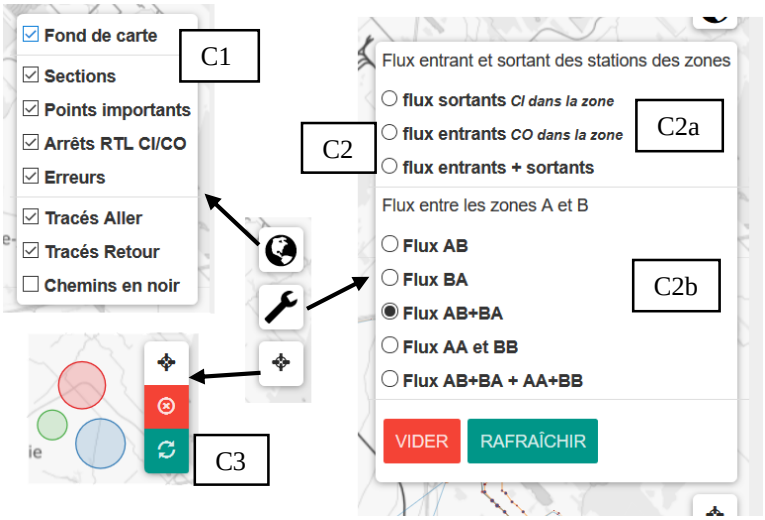


Figure 4-17 - Différents contrôles de la carte

La section (D) propose un tableau récapitulatif des lignes concernées par la sélection spatiotemporelle ainsi que les indicateurs-clés de ces dernières tels que le nombre de transactions effectuées sur cette ligne globalement, ou bien par jour et heure de la journée ou même encore le pourcentage de destinations retrouvées. Ces lignes de tableau peuvent être étendues pour afficher

les informations sur les tracés d'une ligne donnée en reprenant des informations similaires à celles des lignes mais en affichant en plus le profil de charge de ce tracé en partant du kilomètre 0. La colonne A/R permet de visualiser la distribution des transactions entre les tracés aller et retour et dans un deuxième temps la même distribution mais pour les transactions ayant reçues une destination. Le tableau peut être ordonné selon les colonnes de « Charge » ou « %OD ». Un schéma de la ligne permet de visualiser le tracé de cette dernière ainsi que la dispersion avec les points rouges des transactions à erreurs. Si ces points ne sont pas sur la ligne, cela indique une erreur de concordance GTFS. Les graphiques sont interactifs et permettent d'afficher la charge sur chaque barre à l'aide d'une infobulle.

4.3.1.2 Analyse des informations de la vue globale

La Figure 4-16 nous présente également une vue globale du réseau. On visualise sous cette forme principalement le cumul des départs et arrivées des transactions effectuées aux arrêts (C). Trois gros hubs se démarquent des autres arrêts du réseau : Le terminus centre-ville au-dessus de la station de métro Bonaventure, près de la gare centrale de Montréal, le terminal Métro Longueuil-UDS ainsi qu'en troisième le terminus Panama au nord de Brossard le long de la 10. Les couleurs rouges de ces points (à l'exception des erreurs en rouge vif) semblent indiquer un déséquilibre au cours de la journée avec plus d'arrivées que de départs aux terminus Bonaventure et Longueuil-UDS. Cependant, il faut rappeler que toutes les transactions n'ont pas pu être traitées et une ligne entière, la ligne 90 desservant la station de métro Bonaventure sur l'île de Montréal est absente et pourrait être utilisée par les gens pour revenir sur la rive sud.

Le tableau récapitulatif a été coupé pour montrer ici les trois lignes les plus importantes du réseau. On retrouve comme présenté au chapitre 4.2.5 les lignes 45 et 8 mais aussi la ligne 90 qui ne s'est pas vu attribuer de destinations pour ses transactions (0 %OD). La ligne 8 est ici « explosée » pour montrer ses 5 tracés. Seuls deux paraissent utilisés : 8_1_A et 8_1_R. Le tracé de retour possède plus de transactions comparé à celui de l'aller avec 19 724 contre 14 949 (colonnes *charge* et *A/R*) mais a un moins bon rendement dans l'attribution de destinations 79% contre 90% (colonne *%OD*). Le tracé aller présente un pic le matin contre un pic en fin de journée pour le tracé retour confirmant les remarques des chapitres précédents sur le fait que les usagers du RTL effectuent des déplacements pendulaires vers Montréal le matin et reviennent en fin de journée sur la rive-sud.

4.3.2 Interactions entre une zone donnée et le réseau

La Figure 4-18 présente un cas d'utilisation du portail web pour filtrer sur uniquement les transactions interagissant avec une zone dans le quartier Saint-Hubert autour du croisement entre la rue Davis et la montée Saint Hubert. Les arrêts interagissant le plus avec cette zone sont encore une fois les métros Longueuil-UDS et Bonaventure ainsi que le Terminus Panama. L'épaisseur des tracés des lignes indique comme pour la taille de ces arrêts que la ligne vers le Métro Longueuil-UDS est plus utilisée par les usagers de cette zone.

De plus, cette Figure 4-18 grâce au tableau récapitulatif permet d'afficher le profil de charge des lignes ayant des transactions répondant à ce filtre spatio-temporel. On retrouve la ligne 19 qui dessert le métro Longueuil-UDS mais aussi la ligne 5 qui dessert le métro Bonaventure en semaine (tracés 5_1 et 5_3) ou seulement le Terminus Panama le samedi et dimanche (tracés 5_2). Ces différences de réseau montrent l'utilité d'avoir une carte pour voir les éléments mais aussi un tableau récapitulatif pour aller vérifier les informations cartographiées. Pour finir, la ligne 5 possède 14% de transactions en moins que la ligne 19, mais cette différence monte à 18% en prenant en compte le nombre de destinations retrouvées.

Les tracés 5_2 ont la particularité d'être utilisés tout au long de la journée contrairement aux tracés 5_1et 5_3 qui traversent le pont Champlain mais seulement aux heures de pointe et prennent la charge du tracé 5_2 à ces heures. Ces phénomènes montrent sûrement une utilisation prévue par le GTFS. Les traversées du pont Champlain sont visibles par les gros blocs à la fin des tracés Aller et au début des tracés Retour. Pour rappel, le kilomètre de ces profils est placé à l'extrémité gauche des graphiques. L'infobulle affichée en survolant chaque segment le rappelle. Les pics sont placés dans la zone sélectionnée ce qui est logique car on a filtré pour n'avoir que des transactions entrant ou sortant depuis cette zone. Le tracé 5_2_A a un pourcentage de réussite bien inférieur aux autres tracés avec 75.03% contre une moyenne de 95% pour les autres. Cela indique qu'il pourrait être intéressant d'aller chercher avec Kibana plus d'explications.

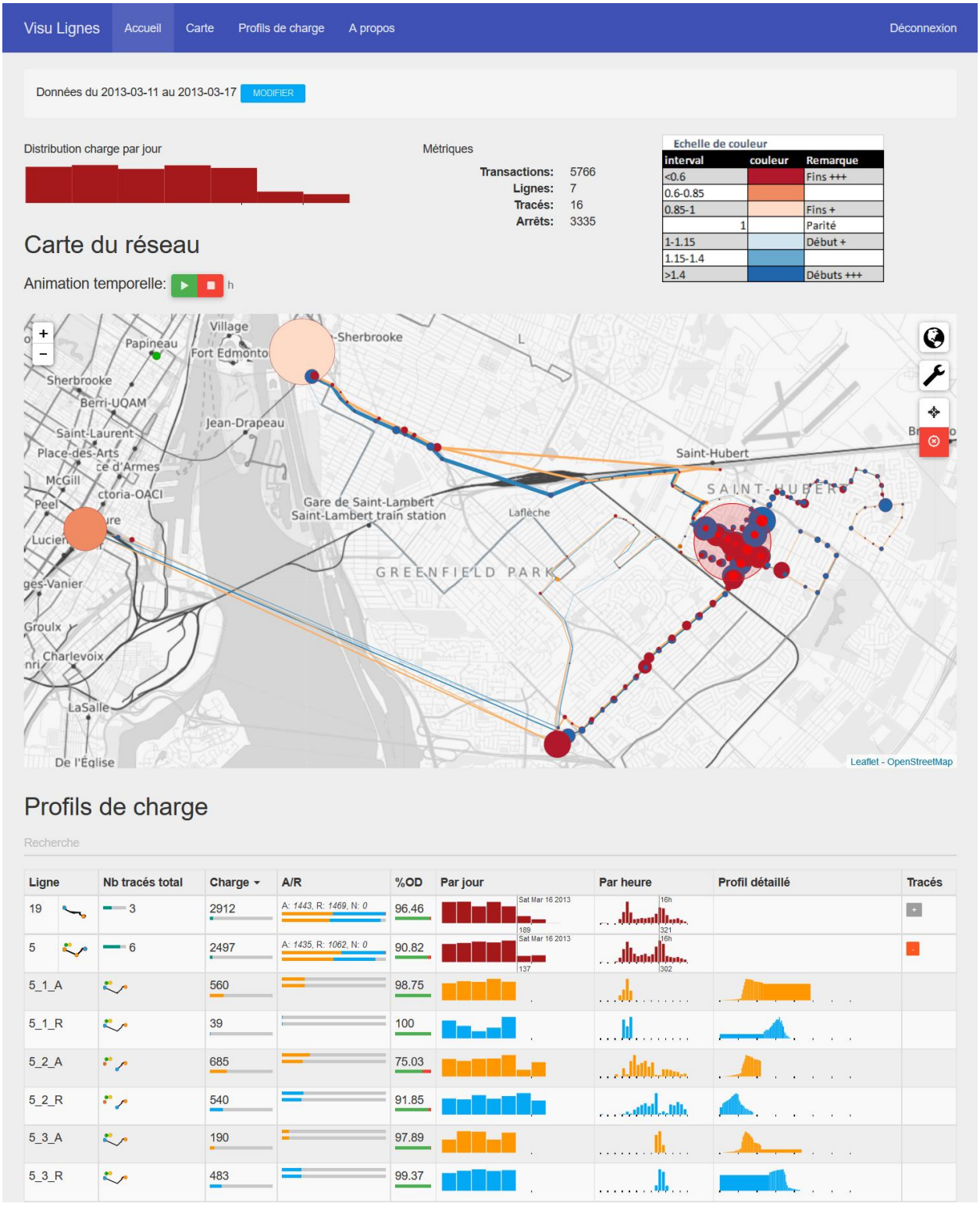


Figure 4-18 - Aperçu Visu Lignes – interactions d'une zone donnée avec le reste du réseau

4.3.3 Interactions entre deux zones données

La Figure 4-19 présente un filtre spatiotemporel de deux zones permettant de visualiser les flux échangés entre celles-ci. La première zone est le métro Longueuil-UDS. La seconde zone est centrée sur un quartier résidentiel ceinturé par la voie de chemin de fer utilisée par l'AMT, par le boulevard Jacques-Cartier et enfin par le chemin de Chambly. La zone comprend tout de même les arrêts de ces boulevards.

La fonctionnalité présentée ici permet de savoir combien de personnes se sont déplacées d'une zone vers l'autre. Le profil de charge de la ligne 73 montre qu'il augmente progressivement le temps de sortir de la zone avant d'être constant jusqu'au terminus situé dans l'autre zone. Ceci est bien l'effet attendu car on a sélectionné les transactions ayant une origine dans une zone et une destination dans l'autre zone. La carte montre un déséquilibre avec une tendance à avoir plus de déplacements vers le métro Longueuil-UDS que de retours. On le voit aussi dans la colonne AR avec 3043 allers contre 2251 retours pour la ligne 73, mais avec 570 allers contre 597 retours pour la ligne 8 (phénomène déjà observé précédemment sur la vue générale, voir chapitre 4.3.1).

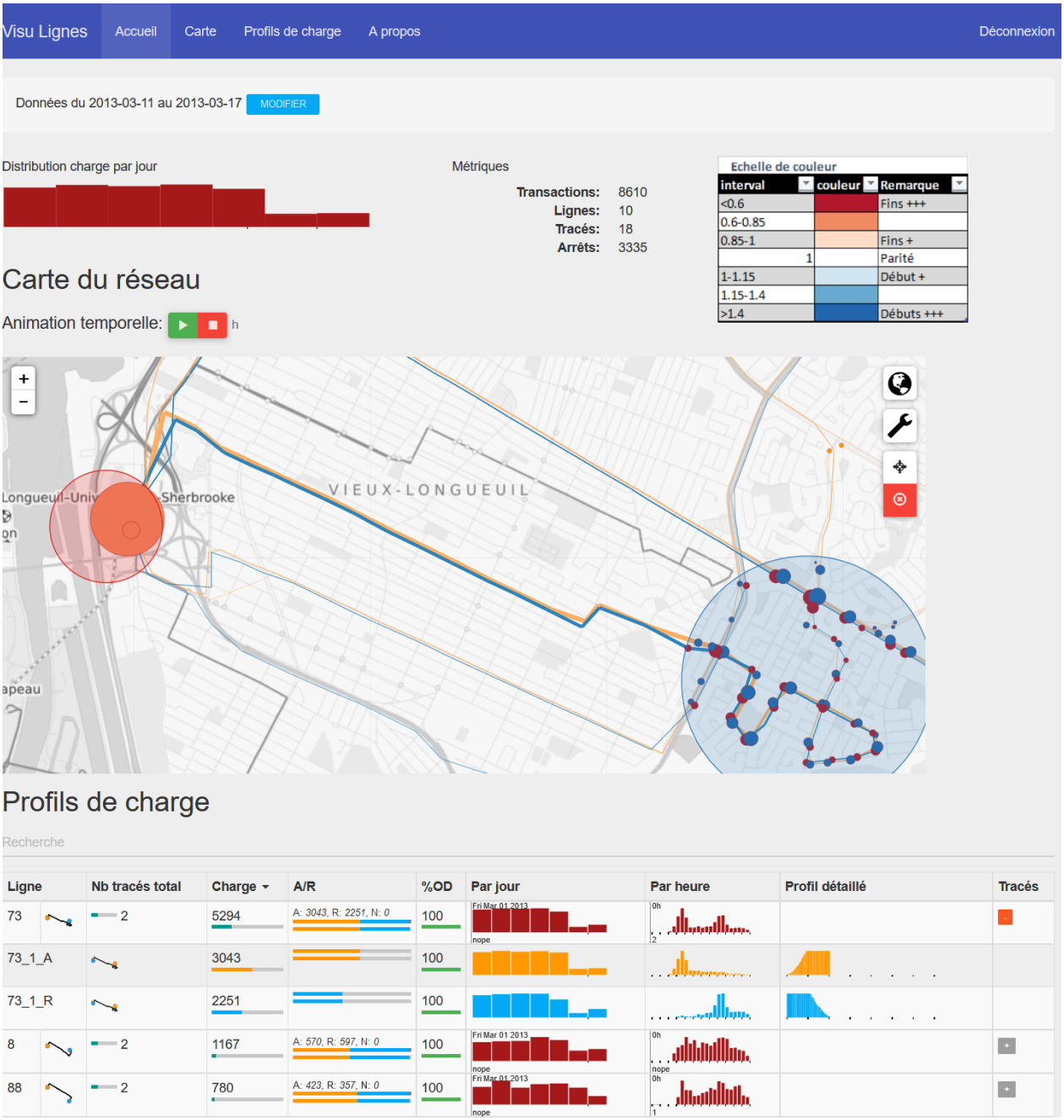


Figure 4-19 - Aperçu Visu Lignes – flux entre deux zones données

4.3.1 Animation temporelle

L'interface web Visu Ligne propose une fonctionnalité supplémentaire – Figure 4-20 – qui permet d'animer les points du réseau en fonction de l'heure de la journée afin de proposer une vue globale sur les différentes heures de la journée de l'évolution de la mobilité des utilisateurs à travers les charges et rapports départs sur arrivées aux différents arrêts. On voit clairement sur la première

ligne l'arrivée de l'heure de pointe avec un flux de déplacements vers Montréal et le métro Longueuil-UDS qui sont en rouge foncé signifiant qu'il y a 1,6 fois plus d'arrivées à ces arrêts que de départs. Pour rappel, les transactions de métro ne sont pas visualisées, car ces dernières causeraient un déséquilibre car on ne dispose que des départs des stations concernées. On observe l'inverse pour la pointe du soir. On remarquera que le terminus Panama suit une logique inverse à celle des arrêts de métro de la STM montrant bien son utilisation en tant que *hub* avant la traversée du pont Champlain.

Figure 4-20 - Animation de la charge aux arrêts sur la journée aux heures de pointe

Côté erreurs, elles sont majoritaires dans la période de 8h le matin, peu visible ici, mais néanmoins confirmé par l'annexe B. Pour la pointe de fin de journée, les erreurs semblent être en majorité liées au métro Bonaventure. Pour rappel, la ligne 90 représente 118 000 transactions mais que seulement une moitié des codes 41, sachant que l'autre n'a pas d'arrêts connus et qu'il y a 250 000 erreurs avec arrêt connu (Figure 4-13). La ligne 45 est elle aussi fortement représentée. On pourrait alors en conclure que les codes d'erreurs de l'après-midi à Bonaventure sont des codes d'erreur 41 marquant l'échec à retrouver la bonne ligne pour la transaction.

CHAPITRE 5 CONCLUSION ET PERSPECTIVES

Pour répondre au besoin d'un exploitant de transport en commun, et dans le cadre d'un projet de collaboration mené depuis trois ans, l'objectif de ce projet de recherche était de concevoir et développer des interfaces permettant de visualiser et d'analyser les transactions journalières de cartes à puce des usagers d'une société de transport, enrichies de leurs destinations.

5.1 Synthèse des travaux

La revue de littérature a permis de resituer l'objet de ce mémoire dans le contexte actuel des recherches menées à ce jour. On retrouve d'une part des travaux portant sur l'enrichissement des données de cartes à puce (origine et destination des trajets) et d'autre part des travaux portant sur les statistiques et visualisation métier utilisées dans le domaine.

La méthodologie a présenté les raisonnements employés pour répondre aux trois sous-objectifs de ce mémoire. On y retrouve dans un premier temps les travaux menés sur l'algorithme destination : comment l'optimiser et le rendre opérationnel, comment l'adapter au cas du RTL, comment faire pour mesurer sa performance, et comment enrichir un peu plus les transactions (distance et temps d'un trajet, recomposition des séquences de tronçons, etc.). Dans un second temps est présenté le travail d'intégration des résultats de l'algorithme dans le portail « Analytics For Transportation » du CeNTAI de Thales. Suite à cette intégration, une nouvelle structure de données pour les transactions enrichies est proposée afin de faciliter leur visualisation. La technologie de gestion des données Elasticsearch utilisée par le CeNTAI est reconduite. Dans un troisième temps, pour faire suite à la nouvelle intégration des documents des transactions désagrégées et enrichies dans Elasticsearch, une dernière partie présente les possibilités d'utilisation du logiciel Kibana ou encore les suggestions de visualisations utiles pour le développement d'un nouvel outil. L'idée est ici de proposer des outils permettant de visualiser et analyser ces données enrichies par l'algorithme destination tout en répondant aux besoins d'une société telle que le RTL.

Le fruit de ce mémoire a été de proposer une solution opérationnelle efficace et relativement rapide de consolidation et enrichissement de transactions de cartes à puce. Ainsi, à partir d'un jeu de données de transactions, cette solution permet en une seule étape d'estimer les destinations des trajets correspondants, de préparer des statistiques supplémentaires (distance et temps de trajet, séquences de tronçons …) et d'exporter les résultats directement dans un fichier texte ainsi que

vers un outil de gestion de données, Elasticsearch. Le temps de traitement est d'environ 20 minutes pour 3 millions de transactions, temps d'export compris. Pour rappel, le RTL a rendu disponibles 3,1 millions de transactions de bus et métro du mois de mars 2013. 20% des transactions sont effectuées dans 4 stations de métro desservies par le RTL. Ces transactions de métro permettent d'aider un peu l'algorithme dans l'estimation des destinations : lorsqu'elles sont prises en compte, elles n'améliorent que de 1% le nombre total de destinations des trajets de bus, portant à 79% le nombre de trajets OD en bus recomposés pour notre jeu de données de mars 2013. Les séquences de tronçons ou déplacements ont été recomposées au cours de l'algorithme. Il en ressort par exemple que 66% des déplacements de bus, ou séquences de tronçons, effectués par les usagers sont des trajets directs sans correspondance. La part d'usagers effectuant des déplacements d'une seule correspondance est respectivement 12% du bus vers le bus et 20% du bus vers le métro. Dans les 2% de déplacements restants, on retrouve 1% (~20 000) de déplacements de type bus-bus-métro.

Par ailleurs, ce mémoire recherche a proposé des solutions logicielles, au moyen de portails web configurés ou développés pour l'occasion, pour analyser et exploiter ces données enrichies à l'étape précédente, données qui sont directement disponibles après avoir été produites et exportées dans Elasticsearch. On retrouve d'une part les travaux menés dans Kibana avec différents exemples d'utilisation : vue globale des données, possibilité de comparaisons entre différents jeux de données sans avoir à recréer les tableaux de bord et analyses sur des points particuliers tels que l'étude des modes de correspondance, la détection d'erreurs, etc. D'autre part, les résultats du développement d'une nouvelle interface web « Visu Lignes » pour les besoins du RTL sont présentés. Cette interface permet d'avoir une vue globale cartographiée sur l'utilisation du réseau ainsi que la charge sur les différentes lignes du réseau. Ce nouvel outil possède des fonctionnalités de filtres spatiotemporels à l'instar du portail du CeNTAI et de Kibana afin d'étudier les interactions de zones choisies avec le reste du réseau ou encore les flux entre des zones données. La possibilité d'animer heure par heure de la journée la charge globale du réseau en fonction de la sélection spatiotemporelle permet d'aider à visualiser les dynamiques de la mobilité des usagers du réseau.

Pour terminer, ce projet de recherche a permis de montrer que l'analyse de gros volumes de données en un temps limité était possible et une solution opérationnelle a été présentée. En effet, il faudrait un temps de traitement de seulement 32 heures pour enrichir les transactions des 8 dernières années du RTL, à raison de 3 millions de données par mois. Ces données de type OD seraient alors

disponibles pour alimenter les analyses des différents départements d'une société de transport en commun tels que la gestion des opérations du réseau, la planification et même le marketing et la finance. Les outils de visualisation développés permettraient alors d'aider le RTL dans la rédaction d'un cahier des charges auprès d'une entreprise offrant des solutions BI pour visualiser leurs données métier, sans compter les avancements possibles pour Thales dans le développement de ses solutions d'affaires.

5.2 Contraintes

La fiabilité de la visualisation et de l'analyse des flux de transport à partir des transactions de cartes à puce dépend grandement de la manière dont les données ont été enrichies en amont. Cette section traite de différentes limites rencontrées lors du travail sur l'enrichissement des données pour le cas du RTL.

Une première limite est de ne pas avoir revu le processus de pairage ayant mené à l'identification des lieux de montées. Ces transactions ont été fournies par Légaré. Plusieurs codes d'erreur de l'algorithme destination suggèrent une mauvaise assignation pour une transaction de sa course GTFS (arrêt incompatible avec la ligne, arrêt étant le terminus de la ligne). Ainsi comme amélioration, le contrôle de la fiabilité du pairage entre les transactions et le GTFS ou alors une meilleure manière de retrouver le meilleur GTFS pourraient être envisagé.

Une deuxième limite est de ne pas disposer de l'horaire réel d'arrivée à destination. Une solution est proposée par (Chu & Chapleau, 2008) pour retrouver dans un premier temps les horaires de passage des courses aux arrêts. L'utilisation des données GPS des bus permettrait d'améliorer ces heures effectives de passage aux arrêts. Cela permettrait une meilleure estimation du temps passé en transport en commun. Les temps de correspondance ou d'activité seraient ainsi plus fiables et des analyses supplémentaires utilisant ces informations pourraient alors se baser sur des résultats moins sujets à être erronés. Ces temps de passage aux arrêts pourraient aussi être utilisés pour réaliser des diagrammes espace-temps permettant de comparer le service offert et planifié d'une journée.

Une troisième limite est un certain nombre de courses non disponibles dans le GTFS utilisé (cas de la ligne 90 manquante). En effet, des courses peuvent être absentes car desservies à l'époque par une autre compagnie, ou peuvent être sciemment rendues non disponibles dans le GTFS public

(exemple : un bus scolaire). S'il n'est pas possible humainement de retrouver les courses manquantes, deux options sont envisageables pour retrouver tout de même les destinations des courses GTFS manquantes. Une première option serait d'avoir au préalable traité les courses GPS comme suggéré précédemment pour connaître les heures de passage aux arrêts et ainsi la structure de la course avec l'odomètre depuis le premier arrêt. Une seconde solution serait de revenir à une définition du réseau semblable à celle utilisée par He mais en faisant attention de disposer du temps planifié en plus de la distance réseau entre deux arrêts. En effet, He estimait le temps de parcours entre deux arrêts à l'aide d'une vitesse moyenne de service, ce qui n'est pas souhaitable car certaines lignes de bus peuvent emprunter des voies rapides. Leur adaptation serait possible sans avoir à réécrire toute la logique de l'algorithme grâce au système de classes mères et filles.

Une dernière limite est le risque d'avoir une destination estimée qui ne soit pas valable. (He, Nassir, Trépanier, & Hickman, 2015) ont présenté que l'algorithme de He dans sa dernière version adaptée à un autre réseau que 79% des destinations estimées étaient en effet à 400m de la destination effectuée (car le réseau disposait des *Tap-In* et *Tap-Out* pour valider). Pour le cas du RTL, une solution pour estimer le pourcentage de précision de l'algorithme destination serait de comparer les destinations du jeu de destinations calculé avec uniquement les transactions de bus et celles du jeu de destinations calculé avec toutes les transactions (bus plus métro).

Résoudre ces problèmes ou défis permettrait d'augmenter la fiabilité des données traitées et donc le nombre de destinations retrouvées. L'analyse de l'utilisation du réseau (trajets directs / trajets avec correspondance / lien métro, bus …) en serait améliorée. Enfin, les analyses et visualisations utilisant le temps de trajet en transport généreraient des données plus fidèles.

5.3 Perspectives

L'enrichissement et la visualisation de données de transactions est un enjeu que les sociétés de transport doivent relever. Celles-ci doivent réussir leur transition vers l'exploitation « automatisée » des données numériques relevées chaque jour.

Outre les contraintes précédentes, des améliorations sont encore à apporter sur le raffinage et l'amélioration des données disponibles pour permettre une meilleure estimation des destinations. Cette estimation peut encore être améliorée. Par exemple, l'estimation des déplacements unitaires ne prend en compte pour le moment que des destinations déjà trouvées. Une solution serait aussi

d'explorer les lieux déjà visités afin d'avoir plus de cas possibles de destination, le tout en réutilisant la méthode développée par He.

Des travaux vont devoir être menés par les entreprises de transport en commun pour raffiner leurs données. Le temps de calcul brut pourrait prendre jusqu'à 20 minutes * 12 mois * 8 années de transactions disponibles, soit 35h de calcul pour toutes les données du RTL sans compter le chargement d'un nouveau GTFS pour chaque période (3 secondes chacun). Pour aider à faire baisser le temps de calcul, l'algorithme pourrait utiliser différents processeurs et pourrait répartir la charge de calcul sur plusieurs machines. Réécrire l'algorithme dans un langage de programmation non script tel que C++ pourrait permettre une meilleure performance. Quant au questionnement à savoir si la solution proposée, Elasticsearch, va être capable de tenir la charge face à la grande quantité de données disponibles, le cas du portail présenté par le CeNTAI de Thales montre qu'ils ont été capables d'analyser les données du STIF qui reçoivent chaque jour en région parisienne 10 millions de déplacements.

Visu Lignes est un prototype d'application et son développement peut être poursuivi. Il existe plusieurs améliorations possibles tant pour le côté technique que pour le côté expérience utilisateur. Les éléments de la page pourraient être plus interactifs et présenter les données déjà disponibles dans la page mais pas encore affichées (charge aux arrêts, profils de charge par heure des arrêts …). Des solutions de filtrage supplémentaires pourraient être ajoutées : afficher les trajets précédant et suivant les trajets d'une ligne donnée, filtrer sur tel type de produit, filtrer sur telle ligne uniquement, sur telle course en particulier, etc. Des statistiques entre zones peuvent être déjà affichées sans avoir à recharger les données. On pourrait songer à afficher des légendes dynamiques, à pouvoir paramétrer les tailles maximales des charges sur les arrêts ou tronçons depuis l'interface. Il pourrait être envisagé d'améliorer le contrôle de l'animation, de proposer de travailler par secteur de recensement, d'intégrer et s'inspirer des travaux vus dans la revue de littérature pour ajouter une nouvelle page dans le site web permettant d'avoir une vue détaillée d'une ligne, d'une course donnée. Les visualisations métier telles que des profils de charge détaillés et des diagrammes espace-temps pourraient être recréées.

Un besoin relevé était de pouvoir comparer différentes périodes temporelles entre elles sans avoir à charger à la main deux instances des tableaux de bord pour chaque période à comparer. Il faudrait

ainsi développer un outil permettant de comparer des séries temporelles. L'extension TimeLion de Kibana (Khan, 2015) est un exemple de solution expérimentale pour réaliser cette tâche.

Les données de transactions des cartes à puce ne sont pas les seules données utilisées. D'autres travaux de fusion de données peuvent être réalisés afin de proposer une meilleure vue d'ensemble prenant en compte plus d'éléments explicatifs, en faisant alors attention à ne pas noyer l'utilisateur sous les informations. Voici une liste exhaustive de nouvelles sources de données possibles à intégrer, à cartographier, à juxtaposer et à comparer entre elles :

- les points d'intérêts du réseau (écoles, grosses entreprises, centres commerciaux …) et si possible les horaires d'ouverture de ces lieux;
- les données de recensement;
- les événements et perturbations connues du réseau (accidents, fermeture pont Champlain, météo …);
- les déplacements tous-modes issus de l'enquête Origine Destination;
- les données de maintenance de la société de transport pour lier le nombre de réparations à réaliser par rapport à l'occupation d'un bus au cours de sa vie ou des profils d'usagers qu'il transporte (si ces données ne sont pas disponibles, on pourrait au moins détecter quand un service n'a pas été effectué suggérant un retard de chauffeur ou un problème mécanique);
- les données d'autres moyens de transport tels que les autres compagnies de bus privé, les trains de banlieue, les taxis, les données de covoiturage, les données de vélo en libre partage tel que Bixi (même si peu utilisé dans le territoire du RTL), la congestion du réseau autoroutier, etc.

En définitive, offrir un moyen d'analyser et d'explorer des données de transactions est important, cependant, face au grand nombre de jours et de mois à analyser, le risque est de ne pas savoir où regarder à moins d'avoir une idée précise de ce que l'on recherche. C'est pour cela qu'une nouvelle partie de traitement des données doit être réalisée pour détecter par exemple les points de rupture dans le réseau entre ce qui est planifié ou habituel et ce qui est observé.

BIBLIOGRAPHIE

Anwar, A., Odoni, A., & Toh, N. (2016). BusViz: Big Data for Bus Fleets. *TRB 2016 Annual Meeting*.

Barry, J., Newhouser, R., Rahbee, A., & Sayeda, S. (2002). Origin and destination estimation in New York City with automated fare system data. *Transportation Research Record 1817*, 183–187.

Barry, M., & Card, B. (2014 , Juin 10). *An interactive exploration of Boston's subway system*. Récupéré sur Visualizing MBTA Data: http://mbtaviz.github.io/

Chu, K. K. (2010). *Leveraging data from a smart card automatic fare collection system for public transit planning.* Thèse de doctorat en Génie Civil (PhD) de l'Ecole Polytechnique de Montréal.

Chu, K. K., & Chapleau, R. (2007). Imputation Techniques for Missing Fields and Implausible Values in Public Transit Smart Card Data. *Presented at the 11th World Conference on Transportation Research, Berkeley, CA.*

Chu, K. K., & Chapleau, R. (2007). Modeling Transit Travel Patterns from Location-stamped Smart Card Data Using a Disaggregate Approach. *Presented at the 11th World Conference on Transportation Research, Berkeley, CA.*

Chu, K. K., & Chapleau, R. (2008). Enriching Archived Smart Card Transaction Data for Transit Demand Modeling. *Transportation Research Record: Journal of the Transportation Research Board, (2063)*, 63-72.

Chu, K. K., & Chapleau, R. (2010). Augmenting transit trip characterization and travel behavior comprehension: Multiday location-stamped smart card transactions. *Transportation Research Record: Journal of the Transportation Research Board, (2183)*, 29-40.

Côme, E. (2013). Récupéré sur Un mois de Vélib: http://www.comeetie.fr/galerie/velib/

Côme, E. (2014). *Stations*. Récupéré sur Vélos en Libre Service et Stats: http://vlsstats.ifsttar.fr/stationinfo.html?city=Paris&stationid=12001

Côme, E., & Oukhellou, L. (2012). Model-based count series clustering for Bike-sharing system usage mining, a case study with the Vélib'system of Paris.

Elastic. (s.d.). Récupéré sur Elastic: https://www.elastic.co/

Gayrard, H., Naour, J., & Thales. (2015). *Complex Analysis in Public Transportation: A Step towards Smart Cities.* Récupéré sur SlideShare: http://pt.slideshare.net/Hadoop_Summit/complex-analysis-in-public-transportation-a-step-towards-smart-cities

Giraud, A. (2015, Novembre). *AntoineGiraud/sendDataToElasticSearch*. Récupéré sur Github: https://github.com/AntoineGiraud/sendDataToElasticSearch

Google. (s.d.). *General Transit Feed Specification Reference.* Récupéré sur Google developers: https://developers.google.com/transit/gtfs/reference

Gordon, J. B. (2012). *Intermodal passenger flows on London's public transport network: automated inference of full passenger journeys using fare-transaction and vehicle-location data.* Mémoire de maîtrise en Transport (MScA) du Massachusetts Institute of Technology.

Gormley, C. (2015, Juin 17). *Modification du mapping à chaud.* Récupéré sur Elastic.co: https://www.elastic.co/fr/blog/changing-mapping-with-zero-downtime

He, L. (2014). *Contributions à l'amélioration d'un algorithme d'estimation des destinations des déplacements unitaires dérivés des validations d'un système de perception par carte à puce.* Mémoire de maîtrise en Génie Industriel (MScA) de l'École Polytechnique de Montréal.

He, L., & Trépanier, M. (2015). Estimating the destination of unlinked trips in public transportation smart card fare collection systems. *Transportation Research Board 94th Annual Meeting (No. 15-3433).*

He, L., Nassir, N., Trépanier, M., & Hickman, M. (2015). Validating and calibrating a destination estimation algorithm for public transport smart card fare collection systems. *CIRRELT-2015-52*.

Khan, R. (2015, Novembre). *Timelion : la composante chronologique de Kibana.* Récupéré sur elastic.co: https://www.elastic.co/fr/blog/timelion-timeline

Kibana - Cardinality Aggregation. (2016). *Cardinality Aggregation.* Récupéré sur elastic.co: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-aggregations-metrics-cardinality-aggregation.html

Kibana by Elastic. (2016). *Visualize.* Récupéré sur Elastic.co: https://www.elastic.co/guide/en/kibana/current/visualize.html

Légaré, F. (2014). *Appariement des données provenant d'un système de paiement par cartes à puce et d'un système de compte à bord en transport collectif.* Mémoire de maîtrise en Génie Industriel (MScA) de l'École Polytechnique de Montréal.

Lomone, A. (2014). *Exploration et traitement multidonnées appliqués à des corridors d'autobus.* Mémoire de maîtrise en Génie Civil (MScA) de l'École Polytechnique de Montréal.

Morency, C., Trépanier, M., & Agard, B. (2007). Measuring Transit Use Variability with Smart-Card Data. *Transport Policy 14(3)*, 193–203.

Morency, C., Trépanier, M., Piché, D., & Chapleau, R. (2010). Bridging the gap between complex data and decision-makers: an example of an innovative interactive tool. *Transportation Planning and Technology, 33:6*, 465-479.

Munizaga, M. A., & Palma, C. (2012). Estimation of a disaggregate multimodal public transport Origin–Destination matrix from passive smartcard data from Santiago, Chile. *Transportation Research Part C: Emerging Technologies, 24*, 9-18.

Pelletier, M.-P., Trépanier, M., & Morency, C. (2011). Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies, 19(4)*, 557-568.

Pourmonet, H., Bassetto, S., & Trépanier, M. (2015). Vers la maîtrise de l'évasion tarifaire dans un réseau de transport collectif. *11e Congrès International De Génie Industriel–CIGI2015.* CIRRELT.

Python Elasticsearch Client. (2016, Février). *API Documentation*. Récupéré sur elasticsearch-py: http://elasticsearch-py.readthedocs.org/en/master/api.html

Remix. (2016, Février). Récupéré sur Remix: Transit planning for the 21st century.: http://getremix.com/

Senseable City Lab, SMART. (2012). *Touching Bus Rides*. Récupéré sur Visual explorations of urban mobility.

Spurr, T., Chu, A., Chapleau, R., & Piché, D. (2015). A smart card transaction “travel diary” to assess the accuracy of the Montréal household travel survey. *Transportation Research Procedia, 11,* 350-364.

Tao, S. (2015). *Investigating the travel behaviour dynamics of Bus Rapid Transit passengers.* Thèse de doctorat (PhD) de l'University of Queensland.

Tao, S., Corcoran, J., Mateo-Babiano, I., & Rohde, D. (2014). Exploring Bus Rapid Transit passenger travel behaviour using big data. *Applied Geography, 53*, 90-104.

Tessier, M.-A. (2015). *Développement d’indicateurs d’analyse et de suivi de la congestion routière.* Mémoire de maîtrise en Génie Industriel (MScA) de l’École Polytechnique de Montréal.

Tranchant, N. (2005). *Modèle de dérivation des déplacements en transport collectif à partir de données de cartes à puce.* Mémoire de maîtrise en Génie Industriel (MScA) de l’École Polytechnique de Montréal.

Trépanier, M., Morency, C., & Agard, B. (2009). Calculation of transit performance measures using smartcard data. *Journal of Public Transportation, 12(1), 5*.

Trépanier, M., Tranchant, N., & Chapleau, R. (2007). Individual trip destination estimation in a transit smart card automated fare collection system. *Journal of Intelligent Transportation, 11(1),* 1-14.

Urban Engines. (2016, Février). *Cities*. Récupéré sur Urban engines: https://www.urbanengines.com/cities/

Urban Engines. (2016, Janvier 28). *Urban Engines Scenario Planning Overview.* Récupéré sur YouTube: https://www.youtube.com/watch?v=CBAwXvG06pA

van Rossum, G., Warsaw, B., & Coghlan, N. (2013). *PEP 0008 -- Style Guide for Python Code.* Récupéré sur Python: https://www.python.org/dev/peps/pep-0008/

Vassivière, F. (2007). *Diffusion de statistiques d’achalandage de transport collectif provenant d’un système de paiement par cartes à puces à l’aide des technologies xml et svg.* Mémoire de maîtrise en Génie Industriel (MScA) de l’École Polytechnique de Montréal.

ANNEXE A – EXEMPLE DE DOCUMENTS ELASTICSEARCH DE DÉPLACEMENTS

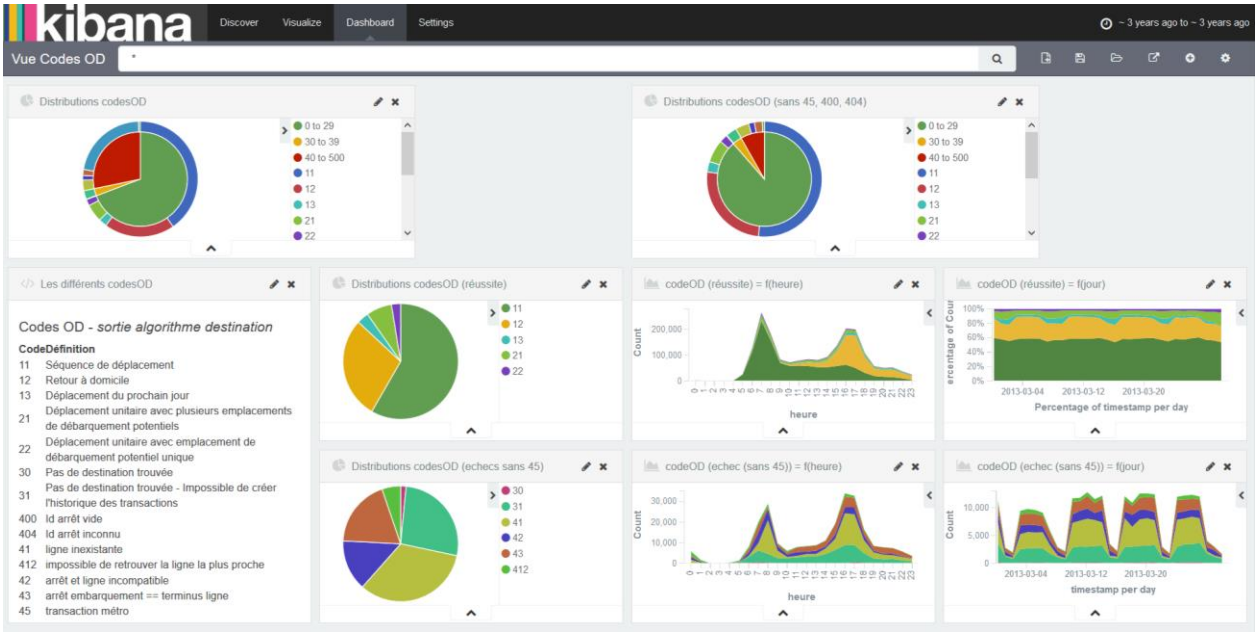
Voici deux exemples de documents représentant des déplacements. Un pour les tronçons et un autre pour les séquences de tronçons. Le document pour le tronçon possède une correspondance à un arrêt de métro. Le document pour la séquence de tronçons reprend comme origine la montée de ce tronçon et comme destination l'arrêt de métro comme on ne peut déterminer la destination finale à la sortie du métro.


```
{
   "_index": "aft",
   "_type": "voyages",
   "_id": "AVARVu7IkDOn4f4Yp6I6",
   "_source": {
      "cardId": "0",
      "ligne": "10",
      "shapeId": "10_2_A",
      "mode": "2",
      "product": "TRAM3O",
      "tripId": "10_2_A_SE_2178_21_31",
      "timestamp": "2013-03-05 21:41",
      "date": "2013-03-05",
      "day_of_week": "2"
      "typ_jour": "SE",
      "hourMin": "21:41",
      "hour": 21,
      "hre10enMinutes": 1301,
      "codeOD": 11,
      "stationCI": {
         "stop_id": 1499,
         "SRIDU": 4620876.04,
         "location": [ -73.45210896 , 45.52842802 ] },
      "stationCO": {
         "stop_id": 4415,
         "SRIDU": 4620884.01,
         "location": [ -73.52108716 , 45.52429759 ] },
      "stationConnexion": {
         "stop_id": 454,
         "SRIDU": 4620884.01,
         "location": [ -73.521961 , 45.525191 ] },
      "distanceOD": 9563,
      "hourMinDest": "21:57",
      "tempsTrajetOD": 21,
      "tempsGtfs": 21,
      "tODSurtGTFS": 1,
      "tempsInterTapIn": 16,
      "tempsAttenteCorrepondance": -5,
      "distanceVolOiseau": 5407,
      "dODSurdVolOiseau": 1.7686332531903088,
      "distanceDestTapInSuivant": 179,
      "pourcentPartMarchee": 0.018374050502976802,
      "tODSurtInterTapInCode11": 1.3125,
   }
}
```

```
{
   "_index": "aft",
   "_type": "deplacements",
   "_id": "AVAR2ARvkDOn4f4Y2CHe",
   "_source": {
      "cardId": "0",
      "mode": "7",
      "product": "TRAM3O",
      "timestamp": "2013-03-05 21:41",
      "day_of_week": "2",
      "date": "2013-03-05",
      "typ_jour": "SE",
      "hourMin": "21:41",
      "hour": 21,
      "hre10enMinutes": 1301,
      "stationCI": {
         "stop_id": 1499,
         "location": [ -73.45210896 , 45.52842802 ],
         "SRIDU": 4620876.04 },
      "stationCO": {
         "stop_id": 454,
         "location": [ -73.521961 , 45.525191 ],
         "SRIDU": 4620884.01 },
      "countVoyages": 2,
      "distanceOD": 9563,
      "hourMinDest": "21:57",
      "tempsTrajetOD": 21,
      "tempsAttenteCorrepondance": null,
      "distanceDestTapInSuivant": null,
   }
}
```

ANNEXE B – EXEMPLE DE COMPARAISON DE CODES OD ENTRE DEUX JEUX DE DONNÉES GRÂCE À KIBANA (SANS 45, 400, 404)

Jeu de données utilisant des données Métro pour aider dans l'estimation de la destination



Jeu de données ayant uniquement des données de bus