

**École polytechnique de Louvain**

# **Évaluation et amélioration d'un réseau de transport en commun en terme de temps de trajet résidence-travail**

Auteur: **Sophie SCHOROCHOFF**  
Promoteurs: **Guillaume DERVAL, Siegfried NIJSSEN**  
Lecteur: **Pierre SCHAUS**  
Année académique 2019–2020  
Master [120] : ingénieur civil en informatique



## Résumé

Dans une société qui vise à réduire son empreinte écologique, les transports en commun sont amenés à évoluer dans les années à venir. Ce mémoire développe un outil visant à améliorer ce type de réseau. L'objectif final est de déterminer quelles sont les nouvelles lignes les plus intéressantes parmi une série de lignes potentielles. Pour ce faire, une métrique d'efficacité du réseau est proposée. Celle-ci est basée sur le temps moyen de trajet résidence-travail des usagers. Les trajets les plus rapides entre chaque combinaison d'arrêts de transports en commun sont utilisés pour déterminer les durées de déplacement. Le gain apporté par une nouvelle ligne est déterminé en l'intégrant au réseau : ce qui implique de réévaluer la mesure. Une technique incrémentale est utilisée afin de réduire le temps de traitement nécessaire par rapport à une réévaluation complète. L'outil présenté est une base pour l'optimisation future d'un réseau de transport en commun. Quelques essais d'optimisation, limités sur l'arrondissement Dixmude, ont toutefois été réalisés.

# Table des matières

<b>Acronymes utilisés</b>	<b>0</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Mesure de l'efficacité des transports en commun</b>	<b>3</b>
2.1 Choix de la mesure . . . . .	3
2.1.1 Estimation de la demande en transport . . . . .	4
2.1.2 Critères de choix . . . . .	5
2.1.3 Mesure choisie . . . . .	5
2.2 Estimation des temps de trajets	
résidence-travail . . . . .	6
2.2.1 Évaluer la durée d'un trajet . . . . .	7
2.2.2 Définir les trajets à effectuer . . . . .	9
2.2.3 Accorder la mesure à une nouvelle configuration du réseau .	10
2.3 Résultats . . . . .	12
2.3.1 Impact de l'estimation de Monte-Carlo	
sur la précision de la mesure . . . . .	12
2.3.2 Évaluation de la mesure dans le Brabant Wallon . . . . .	14
2.4 Pistes d'améliorations pour la mesure . . . . .	15
<b>3 Temps de trajet au sein d'un réseau de transport en commun</b>	<b>17</b>
3.1 Description du problème . . . . .	17
3.2 Méthode de résolution du problème . . . . .	19
3.3 Modélisation d'un réseau	
de transport en commun . . . . .	21
3.3.1 Possibilité de marcher entre les stations . . . . .	23
3.4 All-Pairs Shortest Paths au sein d'un graphe . . . . .	25
3.4.1 Dynamic APSP : État de l'art . . . . .	25
3.4.2 Description des algorithmes de Slobbe . . . . .	26
3.4.3 Adaptation de l'algorithme et stockage des résultats . . . . .	27

3.4.4	Calcul initial de la présence de chemin entre les nœuds . . . . .	28
3.5	Calcul des temps de trajet minimaux . . . . .	28
3.6	Analyse de performances . . . . .	29
3.7	Pistes d'améliorations . . . . .	32
<b>4</b>	<b>Amélioration d'un réseau</b>	<b>33</b>
4.1	Mesurer l'effet d'une modification . . . . .	33
4.2	Algorithme d'optimisation . . . . .	33
4.3	Annuler des modifications : save et restore . . . . .	35
4.4	Résultats et performances . . . . .	37
<b>5</b>	<b>Conclusion et discussion</b>	<b>40</b>
<b>A</b>	<b>Caractéristiques de l'ordinateur</b>	<b>I</b>
<b>B</b>	<b>Brève étude des paramètres influençant la structure du réseau en terme de nombre de stations et de connexions</b>	<b>II</b>
<b>C</b>	<b>Effet de la limite sur le temps de trajet à pied</b>	<b>V</b>
<b>D</b>	<b>Algorithme de recherche de la meilleure paire de modification</b>	<b>VII</b>

# Table des figures

2.1	Évaluation de la mesure : structure . . . . .	8
2.2	Impact de la valeur de C sur la valeur de la métrique . . . . .	13
2.3	Résultats de la mesure . . . . .	16
3.1	Structures et étapes utilisées lors du calcul des temps de trajets minimaux au sein d'un réseau de transport en commun ainsi que lors de la modification de ce dernier. . . . .	20
3.2	Graphe statique du métro de Bruxelles . . . . .	21
3.3	Fully time-expanded graph . . . . .	22
3.4	Déplacements à pied directs et déplacements successifs . . . . .	24
3.5	Pistes de modélisation envisagées pour les déplacements à pied . . . . .	25
3.6	Évaluation des durées d'exécutions . . . . .	31
4.1	Arbre de recherche à un niveau utilisé pour trouver la meilleure modification parmi une série proposée. . . . .	34
4.2	Arbre de recherche à plusieurs niveaux . . . . .	35
4.3	Effet de la meilleure modification en fonction du nombre de modifications testées . . . . .	38
4.4	Répartition des temps d'exécution lors de la recherche de la meilleure amélioration parmi 15 proposées . . . . .	39
B.1	Variation du nombre de connexions dans la journée . . . . .	III
B.2	Analyse du nombre de stations et de connexions de transport en commun . . . . .	IV
C.1	Évolution de la mesure d'efficacité en terme de valeur et de durée d'exécution, en fonction de la limite de temps sur les déplacements maximums à pied. . . . .	VI

# Acronymes utilisés

## **APSP**

All-Pairs Shortest Paths

## **BFS**

Breath First Search

## **BW**

Brabant wallon

## **TC**

Transport en commun

# Chapitre 1

## Introduction

L'amélioration des transports en commun (TC) est une problématique récurrente. En particulier, ces dernières années, où l'écologie a été au cœur de nombreux débats. Le secteur du transport de personnes est connu comme un gros producteur de pollution. En particulier, l'utilisation massive de la voiture privée a un lourd impact sur l'environnement, la congestion routière et la santé [1].

Les services de transport public ont en effet le potentiel d'attirer les usagers de voitures privées en améliorant la qualité de leur service - Redman & al. [2].

De nombreux facteurs peuvent influencer le choix d'utiliser les transports en commun comme moyen de locomotion. Ces derniers peuvent être regroupés en deux catégories : l'efficacité du transport et le confort. La première catégorie reprend des points comme la durée de trajet, la fréquence des transports et répond à la question :

*"Les transports en commun permettent-ils à la population de se déplacer rapidement où et quand elle le désire ?".*

La deuxième catégorie reprend des critères liés au confort, comme la possibilité d'avoir une place assise, la propreté, les aménagements des stations, le service aux clients, etc.

Le présent mémoire développe un outil ayant pour objectif d'aider à améliorer les transports en commun (trains, bus, trams, métros) au regard du premier critère. Le but est de proposer des améliorations du réseau qui permettraient aux usagers d'effectuer leurs trajets souhaités plus rapidement. Les critères liés au confort ne seront pas abordés.

Pour ce faire, la première étape est de fournir et d'estimer une mesure évaluant l'efficacité d'un réseau de transport en commun. Le choix s'est porté sur une estimation du temps moyen nécessaire à la population pour effectuer le trajet

domicile-travail. La mesure a été conçue pour le réseau belge. Néanmoins, elle pourra être adaptée dans d'autres pays. En effet, les données utilisées sont générales et non-confidentielles : horaires de TC, répartition de la population et couples lieu de résidence/lieu de travail des citoyens.

La seconde étape est de pouvoir améliorer un réseau sur base de cette mesure d'efficacité. Il s'agit de déterminer quelles lignes sont les plus intéressantes à ajouter parmi une liste de nouvelles lignes potentielles. Ceci implique de pouvoir réévaluer l'utilisation du réseau modifié, et donc de la métrique choisie. L'idée est de pouvoir connaître l'impact d'une nouvelle ligne sans devoir faire de réelle étude de marché.

Représenter un réseau de TC demande un grand nombre de données. Par exemple, le réseau belge compte 600 gares et plus de 70 000 arrêts de bus. Réduire le temps de calcul est une étape clé pour permettre de considérer des réseaux aussi grands que possible. D'autre part, l'objectif étant de déterminer les meilleures lignes à ajouter, il est primordial de pouvoir en évaluer un maximum. Par conséquent, la limitation du temps d'exécution est constamment recherchée. Dans cette optique, une approche incrémentale est utilisée : la mesure est réévaluée à partir de sa dernière valeur et non depuis zéro.

Le chapitre 2 argumentera le choix de la moyenne des temps de trajet domicile-travail comme mesure pour l'efficacité d'un réseau. Une méthodologie d'évaluation permettant de prendre en compte des modifications du réseau sera également proposée. Cette dernière montrera comment combiner les données sur la population grâce à une distribution dasymétrique et une estimation par la méthode de Monte-Carlo.

La mesure d'efficacité se basant sur les trajets effectués par la population, la durée de trajet entre **toutes** les combinaison d'arrêts de TC est évalué. Cette étape concernant la circulation au sein du réseau sera détaillé dans le chapitre 3. Les temps de trajets sont calculés en respectant les horaires. Afin limiter le nombre d'opérations nécessaires, la durée de chaque trajet est obtenue en l'étendant aux trajets précédemment calculés. Lors d'un changement dans le réseau, suite à l'ajout d'une nouvelle ligne, les durées de trajet seront réévaluées dynamiquement.

Le chapitre 4 portera sur l'amélioration d'un réseau. Il s'agira d'évaluer l'ajout hypothétique d'une série de lignes afin de déterminer la meilleure. Cette optimisation nécessitera de pouvoir annuler les changements afin de tester indépendamment chaque nouvelle ligne du réseau.

En conclusion, le chapitre 5 énoncera les forces et les faiblesses de l'outil développé : les imprécisions liées au choix de la mesure et l'intérêt de l'approche incrémentale. Son code source est disponible à l'adresse : <https://zenodo.org/record/3893851>

# Chapitre 2

## Mesure de l'efficacité des transports en commun

Ce chapitre a pour objectif de proposer une mesure simple qui reflète l'efficacité d'un réseau de transport en commun (TC). Pour commencer, la mesure utilisée sera détaillée ainsi que les choix qui y ont mené. Ensuite, on montrera comment estimer cette mesure et comment l'adapter aux changements dans le réseau. Finalement, les résultats obtenus à l'aide de la mesure seront analysés et interprétés.

### 2.1 Choix de la mesure

L'objectif principal de ce mémoire est l'amélioration d'un réseau de TC. Avant d'entamer toute optimisation, on commence par définir une mesure permettant de représenter sa qualité sous forme d'une valeur unique. Nous avons commencé par une recherche d'articles sur les facteurs influençant la qualité d'un réseau de TC. Redman & al [2] effectuent une revue des études effectuées à ce sujet. Ils développent en particulier les critères de qualité suivants : la fiabilité (liée à la ponctualité), la fréquence, le prix, la vitesse, l'accessibilité, le confort et la convenance. Il en ressort que la fiabilité ainsi que la fréquence, le prix et la vitesse sont des critères importants de qualité. Cependant, l'importance relative de chaque critère dépend des caractéristiques socio-démographique de la population analysée. Par ailleurs, des méthodes permettent de regrouper des critères en une seule valeur, l'une d'entre elle est développée par Eboli et Mazzulla [3].

Lors de la conception de la mesure, on n'a pris en compte que les critères qui dépendent directement des améliorations envisagées : l'ajout de nouvelles lignes. Cela a concentré notre attention sur les aspects géographiques (où placer une ligne) et temporelles (quel sera son horaire). D'autres critères importants de qualité tels

que le prix, la fiabilité, le confort et la convenance ne sont pas abordés. Pour marquer cette distinction, on a opté pour le terme "*efficacité*" plutôt que "*qualité*" des transports en commun.

On définit alors l'efficacité d'un réseau de transport comme son aptitude à permettre d'effectuer rapidement les déplacements que les usagers souhaitent. Par conséquent, il faut donc estimer d'une part les trajets attendus par leur localisation et le moment où ils sont effectués. D'autre part, la notion de rapidité doit être précisée.

### 2.1.1 Estimation de la demande en transport

Estimer les trajets à prendre en compte a été l'une des principales difficultés lors de la conception de la mesure. L'idéal aurait été de prendre en compte tous les trajets que la population concernée a à réaliser. Cependant, définir les lieux d'origine et de destination ainsi que l'heure et la fréquence de chaque trajet est une tâche complexe. Pour résoudre ce type de problème, de nombreux modèles ont été développés.

L'un des premiers est le modèle *four step model* qui permet d'estimer la demande en transport [4]. Comme son nom l'indique, il est composé de 4 étapes :

- **Generation** : estime le nombre de déplacements provenant et arrivant dans chacune des zones étudiées.
- **Distribution** : relie les zones de provenance et d'arrivée pour obtenir une matrice origine-destination.
- **Mode choice** : évalue la proportion de trajets effectués pour chaque type de transport (voiture ,bus, ...).
- **Route assignment** : détermine la route empruntée.

Une faille de cette approche qui utilise les trajets comme unité d'analyse est son manque de précision lors de la représentation de trajets successifs. Le *four step model* considère ces trajets comme indépendants, il ne prend pas en compte leur interaction. Lors d'un enchaînement de plusieurs activités, l'ordre et la localisation de chaque activité va dépendre de l'ensemble des autres activités à effectuer. Imaginons, par exemple, quelqu'un qui désire aller chez le fleuriste et à la pharmacie. Il pourrait s'arranger pour limiter son temps de trajet en allant chez un fleuriste proche d'une pharmacie. L'*Activity-based approach* [5] a été conçue pour répondre à cette lacune. Cette approche plus globale basée sur les activités tient compte des interactions complexes entre les activités et les habitudes de voyage.

## 2.1.2 Critères de choix

Représenter l'efficacité des TC nécessite des données tant sur la structure d'un réseau de TC que sur les trajets que les citoyens désirent effectuer. La mesure emploie des données génériques et faciles d'accès afin de pouvoir être réutilisée dans d'autres pays que la Belgique. Lors du choix, il faut également tenir compte de la phase d'amélioration. La mesure sélectionnée doit permettre l'ajout de nouvelles lignes. Elle doit donc pouvoir juger l'impact de lignes actuellement inexistantes et ne pourra donc pas se limiter à des données sur l'utilisation du réseau actuel. De plus, elle devra pouvoir être réévaluée rapidement afin de pouvoir tester de nombreuses nouvelles lignes de TC durant la phase d'optimisation. La simplicité du calcul de la mesure est un point important, car l'objectif est de réduire la complexité calculatoire.

## 2.1.3 Mesure choisie

On s'est concentré sur la limitation du temps de calcul et des données utilisées en optant pour un modèle simple et probablement moins représentatif que ceux évoqués précédemment. La décision a été de se limiter aux trajets entre la résidence et le lieu de travail. Ces trajets ne représentent évidemment pas l'utilisation réelle et complète des transports en commun. Cependant, ceux-ci ont l'avantage de considérer des lieux fixes (ils ne changent a priori pas lors de l'enchaînement de plusieurs activités). De plus, ils sont accessibles via le recensement belge de 2011<sup>1</sup> qui contient une matrice des déplacements domicile-travail par sexe, donnant pour chaque commune le nombre de personnes travaillant dans chaque autre commune. Il est raisonnable de supposer que chaque personne se rend au travail tous les jours de la semaine et ce indépendamment de la durée du trajet et de sa localisation. Sous ces hypothèses, on obtient une liste trajets (Commune A → Commune B) pour évaluer la mesure.

Vu qu'on ne dispose pas de l'adresse mais uniquement du nom de la commune, une étape sera de générer artificiellement des positions pour les domiciles et le lieu de travail. Pour mieux générer ces positions, on utilise la *cartographie dasymétrique*. Son principe de base est de subdiviser, grâce à des données auxiliaires, les zones sources en unités spatiales plus petites qui possèdent une plus grande cohérence interne de la variable à cartographier [6]. Dans notre cas, les positions des domiciles vont être affinées grâce à la distribution de population par secteur<sup>2</sup> (cf. population par secteur<sup>3</sup>) La population est donc répartie homogénéiquement par secteur.

---

1. **recensement belge de 2011:** <https://data.gov.be/fr/dataset/0782121ea73e0ef80711b358645f74b06ebd3ac8>

2. Un secteur est une subdivision d'une commune.

3. **population par secteur:** <https://statbel.fgov.be/fr/open-data?category=209page=0>

La même méthode sera utilisée pour la localisation du lieu de travail bien qu'elle soit moins appropriée. En effet, il serait plus pertinent d'utiliser des données sur la localisation des lieux de travail plutôt que de celles sur les lieux de résidence.

Concernant le choix du mode de transport, au vu de notre objectif, nous considérons les transports en commun (trains, bus, trams, métros) ainsi que la marche essentielle pour rejoindre un arrêt ou en changer. D'autres modes comme la voiture sont ignorés, ce qui revient à supposer que la possibilité d'utiliser un autre moyen de locomotion n'a pas d'influence sur le temps de trajet moyen en TC. Quant au choix de l'itinéraire, on ne prendra en compte que le trajet le plus rapide durant une tranche horaire définie. Elle a été fixée de 6 h 00 à 10 h 30, cette tranche horaire comprenant l'heure de pointe impose que le déplacement vers le travail ait lieu en matinée.

**Critère de rapidité :** On définit le critère de rapidité comme la moyenne pondérée des durées de trajets pour une liste de couples (origine, destination).

$$\frac{\sum_{(origine,destination)} Temps(origine, destination) \cdot Poids(origine, destination)}{\sum_{(origine,destination)} Poids(origine, destination)} \quad (2.1)$$

L'objectif est donc de minimiser le temps passé dans les transports en commun plutôt que la moyenne des vitesses de déplacement.

## 2.2 Estimation des temps de trajets résidence-travail

Cette section détaille une méthode pour évaluer la mesure d'efficacité des transports en commun basée sur les temps de trajets entre la résidence et le travail. Cette méthode permet également d'adapter la valeur de cette mesure en cas de modifications dans le réseau, telles que l'ajout de nouveaux arrêts et de nouvelles lignes. L'objectif est de calculer la durée de trajet moyenne en TC pour l'ensemble des trajets résidence-travail. Deux points ressortent de cette problématique :

- Évaluer la durée d'un trajet entre deux lieux en se déplaçant à l'aide des transports en commun.
- Définir précisément les trajets à effectuer : lieu d'origine, lieu de destination.

**Notations** : Les notations suivantes seront par la suite utilisées :

commune de résidence ou de travail	→	A ou B
coordonnées spatiales de la résidence ou du travail	→	a ou b
arrêt de transport en commun	→	$S_x$

### 2.2.1 Évaluer la durée d'un trajet

Il existe bien souvent de nombreuses manières pour se déplacer d'un endroit à l'autre et, par conséquent, de nombreux temps de trajet possibles. Nous posons l'hypothèse que l'utilisateur préfère la route la plus rapide.

Contrairement à d'autres moyens de transport tels que la voiture et la marche, les transports en commun suivent des horaires. Ces derniers génèrent des temps d'attente qui font varier les durées de trajets. La durée d'un même itinéraire varie donc au cours d'une journée. À cause de cela, il faut considérer, en plus de l'itinéraire, le moment auquel le trajet est effectué. Pour la modélisation, le choix a été de considérer le trajet de durée minimale parmi tout ceux possibles durant une large tranche horaire : de 6h00 à 10h30. Ce qui revient à supposer que les usagers adaptent leur heure de départ afin de prendre le chemin le plus rapide. Une tranche horaire a été préférée au choix d'une heure de départ arbitraire qui représenterait des temps d'attente inutiles aux arrêts.

Pour plus de clarté, ce chapitre fait abstraction de la structure exacte du réseau de TC (Figure 2.1). Il relègue tout calcul de trajet au sein du réseau au chapitre 3 qui évalue la durée du trajet de durée minimum pour toutes les paires d'arrêts en respectant toutes les hypothèses énoncées. Le calcul du chemin optimal prend en compte toutes les éventuelles correspondances afin de garantir qu'il s'agit du trajet de durée minimale. Les résultats obtenus sont stockés sous forme de tableau qui fournit un accès direct aux temps de trajet minimal entre chaque paire d'arrêt pour une utilisation des TC. De plus, les résultats seront adaptés si la structure du réseau est modifiée. Le présent chapitre utilise ce tableau de résultats pour calculer la mesure. Grâce aux temps de trajet minimaux entre les arrêts, l'évaluation du temps de trajet minimal entre deux points quelconques  $a, b$  de Belgique se simplifie. En effet, il s'agit

- soit d'un trajet sans les transports en commun accompli grâce à un unique déplacement à pied
- soit d'un trajet imposant l'utilisation des TC. Il suffit alors d'identifier la meilleure paire d'arrêts  $(S_x, S_y)$  qui définit le trajet comme composé d'une marche entre la résidence  $a$  et l'arrêt  $S_x$ , suivie d'une utilisation des transports en commun de  $S_x$  vers  $S_y$  incluant d'éventuelles correspondances, pour finir par une nouvelle marche entre l'arrêt  $S_y$  et le lieu de travail  $b$ .

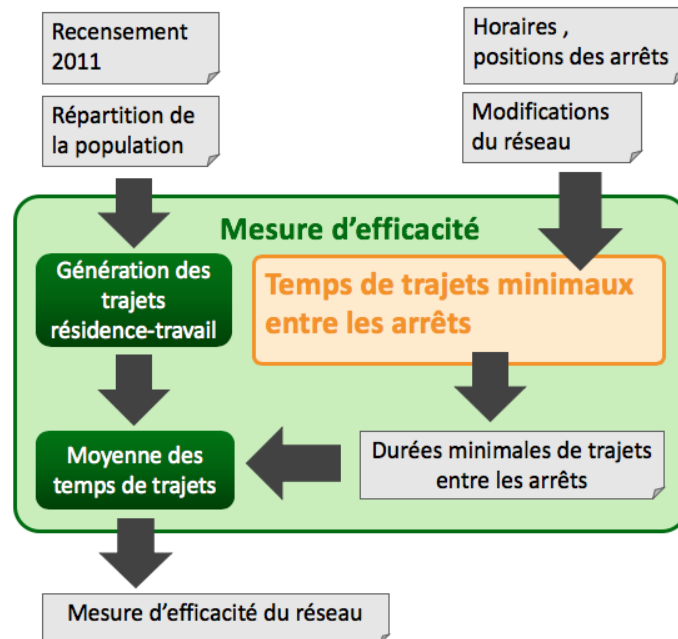


Figure 2.1 – Évaluation de la mesure : structure

**Modélisation de la marche :** Nous approximations le parcours effectué à pied par un trajet à vol d’oiseau à vitesse réduite. Cela permet d’obtenir des temps de trajet plausibles tout en simplifiant le problème. La structure du réseau routier est ignorée, seule compte la position géographique des stations de TC. De plus, le calcul du chemin le plus court à pied au sein du réseau routier est évité. Pour ce faire, le temps de trajet s’obtient en divisant la distance à parcourir par la vitesse de déplacement à vol d’oiseau. Lors de l’estimation de celle-ci, il est pris en compte que le trajet effectué réellement par le piéton est plus long. Selon Héran[7], en centre-ville, le détour moyen pour un piéton est d’environ 20% et de 30% en périphérie. On considère la valeur médiane 25% pour ne pas avoir à différencier la ville de la périphérie. La distance réelle vaut donc  $1,25 \times$  la distance à vol d’oiseau. Vu qu’un piéton avance à une vitesse d’environ 4,5 km/h, sa vitesse à vol d’oiseau est de  $\frac{4,5\text{km/h}}{1,25} = 3,6\text{km/h}$  environ.

Bien qu’en théorie, la durée d’un trajet effectué par un piéton ne soit limitée que par la tranche horaire considérée (6h-10h30), un trajet à pied de Bruxelles à Waterloo est inimaginable au quotidien. Par souci de réalisme, la durée maximale parcourue à pied en une traite a été limitée arbitrairement à 30 minutes (soit 1,8 km) lorsque l’usager prend les transports en commun et 60 minutes (soit 3,6 km) sinon. Cette limite a pour but de réduire le temps de calcul (voir Annexe C).

**Temps de trajet minimal :** Pour calculer le temps de trajet minimal  $T(a, b)$ , le minimum est pris entre la durée d'un trajet effectué uniquement à pied ( $T_{walk}(a, b)$ ) et celle en passant par les TC ( $T_{walk+TC}(a, b)$ ).

$$T(a, b) = \min(T_{walk}(a, b), T_{walk+TC}(a, b)) \quad (2.2)$$

Pour évaluer la durée minimale d'un trajet passant obligatoirement par les TC, on détermine toutes les paires d'arrêts  $(S_x, S_y)$  pouvant constituer l'étape d'utilisation des TC du trajet. Pour cela, on liste tous les arrêts atteignables en 30 min à partir du domicile  $a$  et du lieu de travail  $b$ . Ensuite, on calcule la durée minimale parmi tout trajet défini par une paire d'arrêt atteignable. Soit le resultat du calcul suivant,

$$T_{walk+TC}(a, b) = \min_{\substack{S_x, S_y \in \text{arrêts} \\ T_{walk}(a, S_x) \leq 30 \\ T_{walk}(S_y, b) \leq 30}} T_{walk}(a, S_x) + T_{TC}(S_x, S_y) + T_{walk}(S_y, b) \quad (2.3)$$

où  $T_{walk}$  est la durée de trajet en marchant et  $T_{TC}$  le temps de trajet minimum entre deux arrêt de transport en commun obtenu au chapitre 3.

## 2.2.2 Définir les trajets à effectuer

Le recensement de 2011 reprend le nombre de Belges habitant dans une commune  $A$  et travaillant dans une commune  $B$ . On veut utiliser ces informations pour définir les trajets à effectuer. Cependant, la fonction permettant d'estimer le temps de trajet,  $T(a, b)$ , nécessite de connaître la localisation exacte du lieu départ  $a$  et d'arrivée  $b$  alors que le recensement permet seulement de connaître la commune. Il faut pallier ce manque d'information. Pour cela, on évalue le temps de trajet moyen entre tous les lieux d'habitations et de travail possible. En voyant les lieux,  $a, b$  comme des variables aléatoires dépendant des Communes  $A, B$  et suivant la fonction de densité  $f_{A,B}(a, b)$ , cela revient à calculer l'espérance  $E$  :

$$Temps(A, B) := E[T(A, B)] = \int \int T(o, d) \cdot f_{A,B}(a, b) \delta o, \delta \quad (2.4)$$

On fait l'hypothèse que les fonctions de densité  $f_A(a)$  et  $f_B(b)$  suivent la distribution de la population par secteur. Sous cette condition, elles sont indépendantes et  $f_{A,B}(a, b) = f_A(a) \cdot f_B(b)$ . Bien que l'estimation des positions ait pu être quelque peu améliorée par l'utilisation de la population par secteur comme donnée auxiliaire,

le manque de précision sur l'adresse du domicile et du travail sont des limitations sérieuses liées au manque de données.

La méthode de Monte-Carlo permet de calculer une valeur numérique approchée de l'espérance. L'idée derrière cette méthode est de produire un échantillon aléatoire  $[(a_1, b_1), (a_2, b_2), \dots, (a_{N_{AB}}, b_{N_{AB}})]$  suivant la fonction de densité désirée  $f_{A,B}(a, b)$ . Puis de construire un estimateur à partir de la moyenne empirique :

$$\overline{Temps}(A, B) = \frac{1}{N_{AB}} \sum_{i=1}^{N_{AB}} T(a_i, b_i) \quad (2.5)$$

On utilise cette méthode pour obtenir le temps de trajet moyen entre deux communes. Les éléments de l'échantillon sont sélectionnés en tirant d'abord aléatoirement un secteur, puis en sélectionnant uniformément une position au sein de celui-ci. La probabilité de tirer chaque secteur est directement proportionnelle au nombre de personne y résidant. La taille de l'échantillon  $N_{A,B}$  est égale à une constante  $C$  fois le nombre de trajet entre les deux communes déterminé par le recensement  $NbreTrajet$ .

$$N_{AB} = C \cdot NbreTrajet(A \rightarrow B) \quad (2.6)$$

Ainsi au plus un type de trajet inter-communes est présent dans le recensement au mieux il est évalué. La section 2.3 montrera que la mesure converge bien vers une valeur.

Une fois cette étape franchie, la valeur de la mesure d'efficacité s'obtient en calculant la moyenne pondérée des temps de trajet entre les communes :

$$\frac{\sum_{A,B \in communes} \overline{Temps}(A, B) \cdot NbreTrajet(A \rightarrow B)}{NbreTotalTrajet}$$

### 2.2.3 Accorder la mesure à une nouvelle configuration du réseau

Lorsque le réseau de TC est modifié, la valeur de la mesure doit être adaptée pour correspondre à la nouvelle configuration du réseau. Comme pour le calcul de la valeur initiale, on va se baser sur les résultats qui seront développés au cours du chapitre 3. En effet, il fournit les temps de trajet minimaux pour chaque paire d'arrêts de TC. En cas de modifications, il adapte ces temps pour correspondre au nouveau réseau. De plus, il fournit également une liste de tous les changements effectués. Deux approches ont été envisagées pour passer d'un changement dans les temps de trajet entre les arrêts de TC à un changement de valeur pour la mesure. La première est de réévaluer la mesure depuis zéro sur base de la nouvelle configuration

du réseau. Cependant, de nombreux trajets seraient recalculés inutilement. L'idée derrière la seconde approche est de recalculer uniquement les trajets potentiellement concernés par les améliorations. Elle consiste à déterminer l'ensemble des trajets  $a \rightarrow b$  améliorés suite aux modifications puis, d'ajuster la valeur de la mesure en fonction des nouveaux temps de trajets obtenus. Pour cela, elle part de la liste des améliorations de temps de parcours entre les arrêts dont l'obtention sera détaillée au chapitre 3. L'ensemble des paires d'arrêts modifiées est parcouru. Pour chaque paire d'arrêts  $(S_x, S_y)$ , on recherche toutes les paires de positions  $(a, b)$  dont le trajet pourrait passer par  $S_x \rightarrow S_y$  parmi toutes celles générées précédemment par la méthode de Monte-Carlo. Pour chacune de ces paires, on regarde si le temps de trajet diminuerait en passant par  $S_x \rightarrow S_y$ . Lorsque c'est le cas, on réduit la métrique. Lorsque l'échantillon aléatoire est conservé pour définir les trajets à considérer, la première approche calculant la mesure directement à partir du réseau modifié et l'approche incrémentale aboutissent à la même valeur quelque soit la modification.

---

```

1 #Pour toutes paires d'arrêt dont le temps a diminué suite à un
   changement dans le réseau
2 for (Sx, Sy) in modified_travel_time:
3     #Pour tout trajet a->b qui a été généré par Monte-Carlo
4     for (a,b) in Monte_Carlo_travels:
5         if T_walk(a,Sx) <= max_walk and T_walk(Sy,b) <= max_walk:
6             #Meilleur temps avant modification
7             old_time = get_time_before_modif(a,b)
8             #Temps en passant par une paire d'arrêt avec temps changé
9             new_time = T_walk(a,S_x) + T_TC(S_x, S_y) + T_walk(S_y,b)
10            if new_time < old_time:
11                metric.reduce(a,b,old_time - new_time)
12                set_time(a,b,new_time)

```

---

## 2.3 Résultats

### 2.3.1 Impact de l'estimation de Monte-Carlo sur la précision de la mesure

La Figure 2.2 a été créée dans l'optique d'avoir une idée de l'impact du choix de la constante  $C$  sur la valeur de la mesure et le temps d'exécution. Pour rappel, cette constante  $C$  est utilisée pour définir le nombre d'échantillons utilisé par la méthode Monte-Carlo (voir 2.6). L'impact de  $C$  est évalué pour l'arrondissement de Dixmude à l'aide d'un MacBook Pro de 2009 (caractérisé dans l'Annexe A). Cet arrondissement situé en Flandre, est l'un des arrondissements avec le moins d'arrêts de TC. La Figure 2.2a représente l'évolution de la valeur de la mesure en fonction de  $C$ . Les valeurs de  $C$  analysées vont de  $10^{-3}$  à 10, pour chacune d'entre elles 3 exécutions indépendantes ont été réalisées. La méthode de Monte-Carlo prévoit que les résultats convergent vers une unique valeur lorsque le nombre d'échantillons augmente. On observe une telle convergence vers une unique valeur : 36.6 minutes lorsque  $C$  augmente. À l'aide de la Figure 2.2b, on observe que le temps d'exécution est directement proportionnel à  $C$ .

Ensuite, on s'intéresse à des modifications du réseau. On considère l'ajout arbitraire des lignes suivantes :

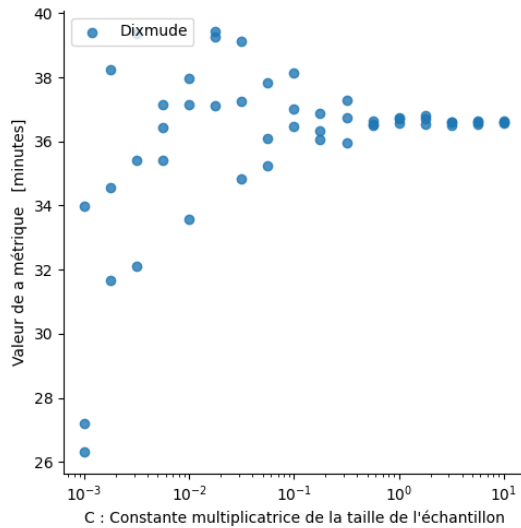
- Modification 1 : delijn42525 à 7h18 → delijn90508 à 7h20
- Modification 2 : delijn42296 à 8h46 → delijn41729 à 8h50
- Modification 3 : delijn87605 à 6h44 → delijn42537 à 6h58

Ces 3 modifications sont effectuées tour à tour sur chacune de 3 initialisations de la métrique. Comme pour l'initialisation, on observe une relation linéaire entre  $C$  et le temps d'exécution (Figure 2.2d). Par ailleurs, le temps d'exécution pour les modifications 2 et 3 est certes plus court que pour l'initialisation avec un même  $C$  mais n'est pas aussi court qu'on aurait pu l'espérer. Dans certains cas, une réévaluation complète de la métrique vaudra mieux qu'un calcul incrémental.

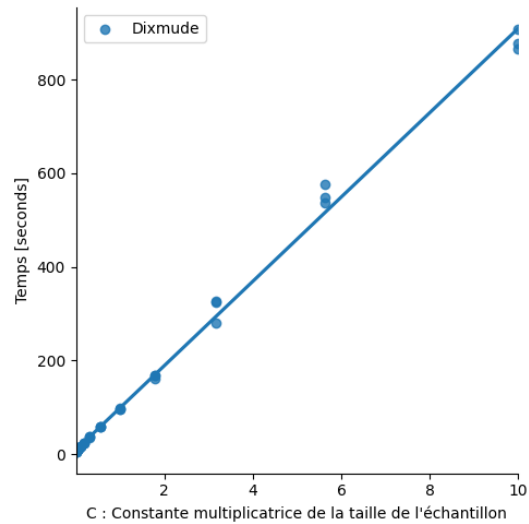
La figure 2.2c montre la différence de valeur entre la mesure après la modification du réseau et celle d'avant. On utilise le même échantillon de trajets afin que la différence de valeurs mesurées porte uniquement sur la modification du réseau. Les modifications 1 et 2 donnent un résultat similaire : peu d'impact sur la valeur de la métrique. Pour la modification 3, on observe une convergence vers une augmentation de 0.3 minutes de temps de trajet.

Vu que l'objectif de ce mémoire est de limiter le temps de trajet en ajoutant des lignes de TC, cette augmentation peut surprendre. Elle provient du traitement de nouveaux trajets auparavant non pris en compte du fait des contraintes posées dans ce chapitre : limitation du temps de marche et utilisation d'une tranche horaire.

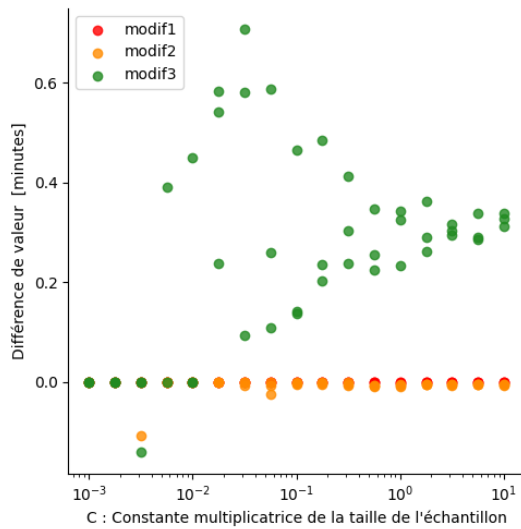
Les trajets non réalisables n'entrent pas le calcul de la moyenne des temps de trajet. Suite à la modification 3 du réseau, de nouveaux trajets jusque là ignorés peuvent être réalisés. Lorsque ces derniers sont de longue durée, la moyenne de l'ensemble des trajets après modification augmente.



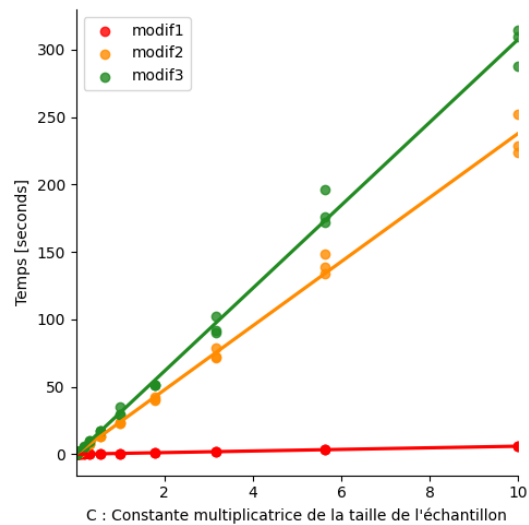
(a) Valeur de la métrique lors de l'initialisation



(b) Temps d'exécution lors de l'initialisation



(c) Différence de valeur pour la métrique entre après et avant modification



(d) Temps d'exécution après modification du réseau

**Figure 2.2** – Impact de la valeur de C sur la valeur de la métrique

### 2.3.2 Évaluation de la mesure dans le Brabant Wallon

Dans ce chapitre, une mesure globale de la moyenne des temps de trajet minimum entre le domicile et le travail a été développée. Afin de déterminer quelles communes sont favorisées par les transports en commun, on peut légèrement modifier la mesure pour lui donner un caractère plus local. Pour chaque commune  $A$ , on restreint le nombre de trajets considérés : seuls les trajets pour lesquels le domicile se trouve en  $A$  sont conservés. La moyenne des temps de trajet est donc calculée pour tous les trajets ayant un domicile en  $A$  et un lieu de travail dans n'importe quelle commune de la zone géographique considérée.

Dans la figure 2.3a, la mesure a été évaluée pour chaque commune du Brabant wallon (BW) avec  $C$  fixé à 1. Les trajets étudiés ont été donc limités à ceux effectués au sein du BW. On observe que les temps de trajets moyens varient entre 27 et 55 minutes. Les trajets prennent plus de temps dans l'est. Cependant, cela ne signifie pas que cette zone est mal desservie par les transports en commun, la distance parcourue en moyenne pourrait être simplement plus grande. En effet, dans la figure 2.3b représentant les vitesses de déplacement (durée de trajet moyenne / distance moyenne parcourue), on observe des déplacements plus rapides dans l'est que dans le nord. Les trajets plus longs semblent donc souvent plus rapides. Par ailleurs à l'ouest, la commune de Rebecq a, à la fois, une faible vitesse et un long temps de trajet. C'est une piste à imaginer pour améliorer l'efficacité des TC.

## 2.4 Pistes d'améliorations pour la mesure

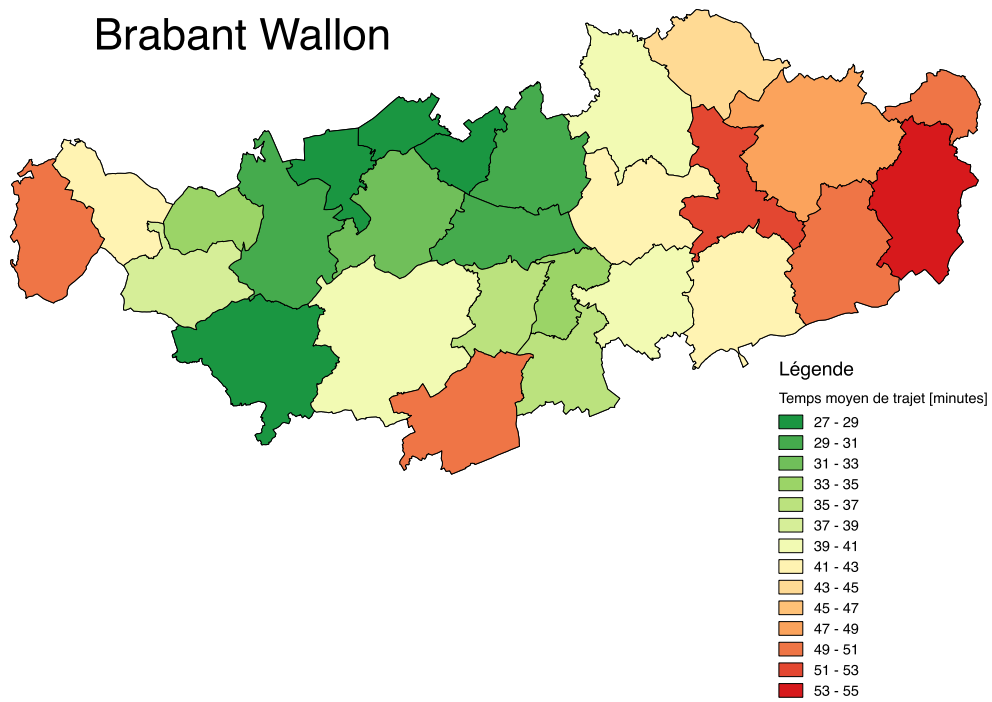
La métrique qui a été choisie est simpliste et peut donc être améliorée. On commencera par proposer des améliorations pour la mesure actuelle. Puis, on envisagera l'utilisation d'un autre type de mesure.

L'une des premières choses à envisager avec la mesure actuelle est de régler le problème des trajets irréalisables. En effet, comme constaté précédemment avec l'exemple de la modification 3, des ajouts de lignes qui permettraient à plus d'utilisateurs de circuler avec les TC peuvent faire augmenter la valeur moyenne du temps de trajet, et donc d'être considérées à tort comme de mauvaises modifications. Une solution rapide qui sera utilisée dans la suite de ce mémoire, est de ne considérer que le trajet réalisable à l'initialisation de la mesure dans le calcul de la moyenne des temps de trajets. Le premier défaut de cette solution est qu'elle n'encourage pas à créer plus de trajets réalisables. Le second défaut de cette solution rapide est que la métrique représentera de moins en moins la moyenne des temps de trajet au fur et à mesure des modifications effectuées. Trouver un moyen de supprimer les trajets irréalisables ou de les introduire dans la métrique sera donc une étape fondamentale pour l'améliorer. Une seconde piste pour améliorer la mesure d'efficacité présentée dans ce chapitre serait de considérer davantage de données. Celles utilisées actuellement sont limitées, une localisation plus précise de la résidence et plus particulièrement du travail serait enrichissante. De plus, la métrique se concentre sur l'aspect rapidité du trajet et néglige le choix de quand l'effectuer. Cet aspect pourrait être représenté en intégrant une heure de départ et/ou d'arrivée souhaitée au trajet et en adaptant la métrique.

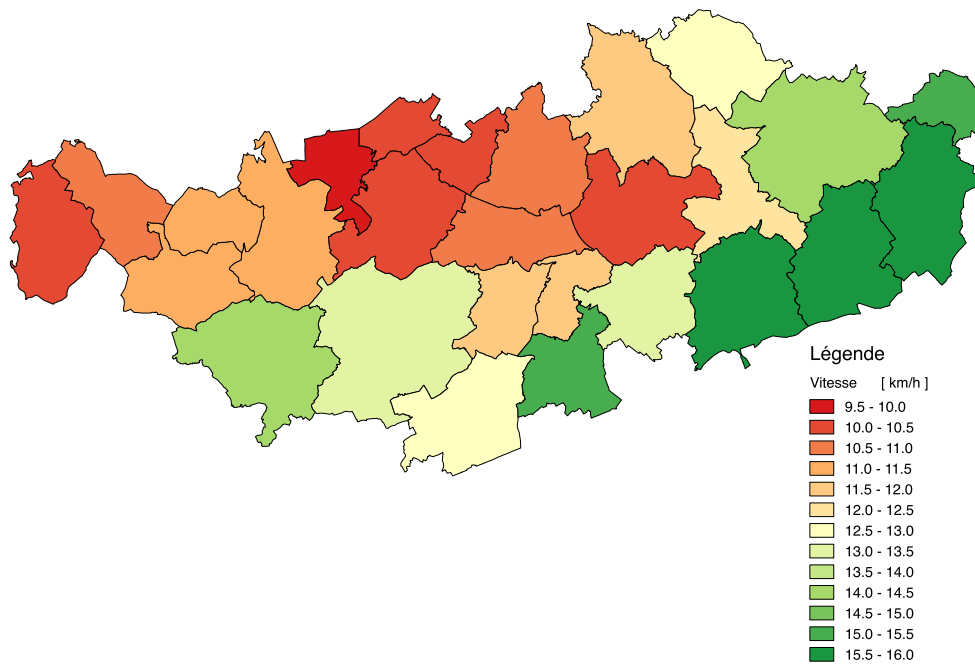
Une autre solution serait de changer complètement la mesure utilisée. La nouvelle mesure pourrait se baser sur une des modélisations de l'utilisation des TC développées dans la littérature (*four step model*, *activity based approach*, etc). Ces dernières pourraient permettre d'améliorer la liste des trajets à effectuer en prenant en compte d'autres types d'activités, d'autres moyens de transport, etc. La pondération des étapes du trajet pour représenter la satisfaction d'un client (évoqué par Frappier [8]) pourrait également être envisagée. Lors du choix de l'une d'elles, il faudra garder à l'esprit la partie d'amélioration du réseau qui implique de pouvoir évaluer l'impact d'une ligne de TC inexistante.

Que le choix se porte sur une autre mesure ou une version améliorée de celle existante, une étude analysant sa validité serait une avancée considérable.

## Brabant Wallon



(a) Résultats de la mesure pour les communes du Brabant Wallon



(b) Vitesse de déplacement en transport en commun

**Figure 2.3** – Résultats de la mesure

# Chapitre 3

## Temps de trajet au sein d'un réseau de transport en commun

Ce chapitre vise à déterminer le temps de trajet minimal pour **toutes** les paires de stations/arrêts de transport en commun (TC). Il détaillera également comment mettre à jour ces temps de trajet en cas de légère modification du réseau concerné. L'accent sera particulièrement mis sur la réduction du temps d'exécution. Dans la suite de ce chapitre, on utilisera le terme "station" plutôt que "arrêt" pour éviter toute confusion avec le mot "arête" utilisé en théorie des graphes.

### 3.1 Description du problème

Les façons de voyager sont très diversifiées, allant du simple trajet à pied au voyage en avion. Les trajets peuvent se différencier tant par le moyen de transport utilisé que la manière de les combiner et par le choix de l'itinéraire, etc. Comme énoncé lors du chapitre 2, on se focalise sur les trajets effectués en Belgique combinant l'usage des transports en commun (trains, bus, trams et métros) avec des déplacements à pied. La marche permettra de passer d'une station à l'autre. De plus, on ne considère que les trajets réalisables durant la tranche horaire de 6 h 00 à 10 h 30.

L'ensemble des informations sur la structure du réseau de TC est obtenu grâce aux fichiers GTFS (General Transit Feed Specification) fournis par la SNCB<sup>1</sup>, le TEC<sup>2</sup>, DE LIJN<sup>3</sup> et la STIB<sup>4</sup>. Ces données contiennent les horaires des transports en commun ainsi que les informations qui y sont associées (position, nom, etc). Les retards sont un problème majeur lié aux transports en commun et leurs prédictions constituent un champ de recherche à part entière. Pour une question de simplicité, les temps de trajet calculés ignorent tous retards possibles et se basent uniquement sur l'horaire initialement planifié.

Outre l'utilisation des TC, les voyageurs ont la possibilité de marcher pour passer d'une station à l'autre. Ces déplacements sont modélisés comme dans la section 2.2.1 : ils sont approximés par des traversées à vol d'oiseau effectuées à la vitesse de 3,6 km/h et ont une durée limitée à 30 minutes.

Le premier objectif de ce chapitre est de déterminer, pour **toutes** les paires de stations du réseau considéré, la durée minimale d'un trajet entre celles-ci. Autrement dit, pour chaque couple de station, il s'agit de calculer la durée du trajet effectué à l'heure de départ optimal entre 6h00 et 10h30 et prenant l'itinéraire le plus rapide.

Le second but est de pouvoir modifier légèrement le réseau en vue de l'améliorer. Les modifications autorisées sont l'ajout de nouvelles lignes et de nouvelles stations. Après modifications du réseau, les durées de trajets doivent être rapidement adaptées afin de correspondre à la nouvelle configuration du réseau.

Une des difficultés pour atteindre ces objectifs est de traiter une grande quantité de données nécessaires à la modélisation d'un réseau de TC. La Belgique compte environs 600 gares et plus de 70 000 arrêts de bus. Afin de pouvoir considérer ne serait-ce qu'en partie un tel réseau, un point crucial est la limitation du temps de calcul et de l'utilisation de la mémoire. La réduction du temps d'exécution est particulièrement importante parce qu'au plus il est réduit, au plus il sera possible de tester des différentes améliorations lors de la partie optimisation présentée chapitre 4.

---

1. SNCB /NMBS :  
<https://www.belgiantrain.be/fr/3rd-party-services/mobility-service-providers/public-data>

2. TEC : <https://data.gov.be/en/dataset/tec>

3. De LIJN : <https://data.gov.be/en/dataset/delijn>

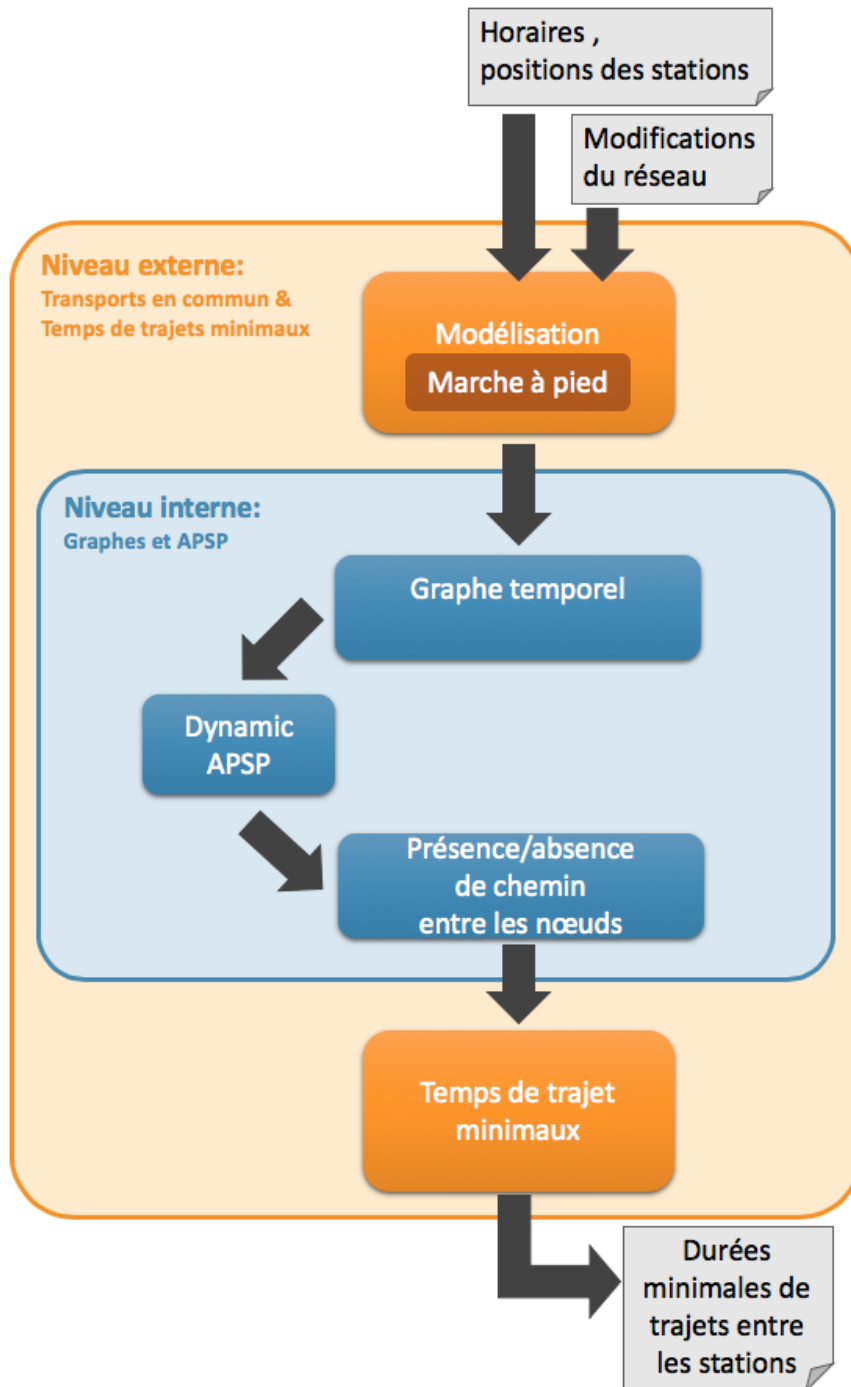
4. STIB : <https://opendata.stib-mivb.be/store/data>

## 3.2 Méthode de résolution du problème

Le problème du calcul du temps de trajet minimal pour chaque paire de stations a été divisé en étapes de résolution. Cette section présentera chacune d'elles et montrera comment elles sont agencées. Les différentes étapes mentionnées seront ensuite détaillées en profondeur dans les sections qui suivent.

La première étape est la modélisation. Elle consiste à transformer les horaires et positions de TC en un format plus adapté à la recherche du plus court chemin : les graphes temporels. Outre les TC, elle représente également la possibilité de marcher entre les stations. Par ailleurs, dès que le réseau est modifié, ces changements sont répercutés sur le graphe. Le rôle de la seconde étape est de déterminer la présence ou l'absence d'un chemin pour toute paire de nœuds du graphe. Puis de stocker les résultats ainsi obtenus sous forme de matrice. Un second rôle de cette étape est de modifier dynamiquement cette matrice lorsque des nœuds ou des arêtes sont ajoutés au graphe. La dernière étape consiste à calculer les temps de trajet minimaux à partir de la matrice indiquant la présence d'un chemin.

Ces étapes de résolution sont conceptuellement séparées en deux niveaux : interne et externe (Figure 3.1). Ces niveaux se distinguent par leur rôle et les structures de données utilisées. Le niveau externe (étapes 1 et 3) utilise les données d'un réseau de transport en commun (lignes, stations, possibilités de marche). Il sert d'interface pour le calcul des temps de trajets minimaux et les modifications du réseau. Pour réaliser cela, il utilise le niveau interne. Ce dernier travaille principalement sur des graphes et détermine dynamiquement la présence de chemin entre les nœuds de celui-ci.



**Figure 3.1** – Structures et étapes utilisées lors du calcul des temps de trajets minimaux au sein d'un réseau de transport en commun ainsi que lors de la modification de ce dernier.

### 3.3 Modélisation d'un réseau de transport en commun

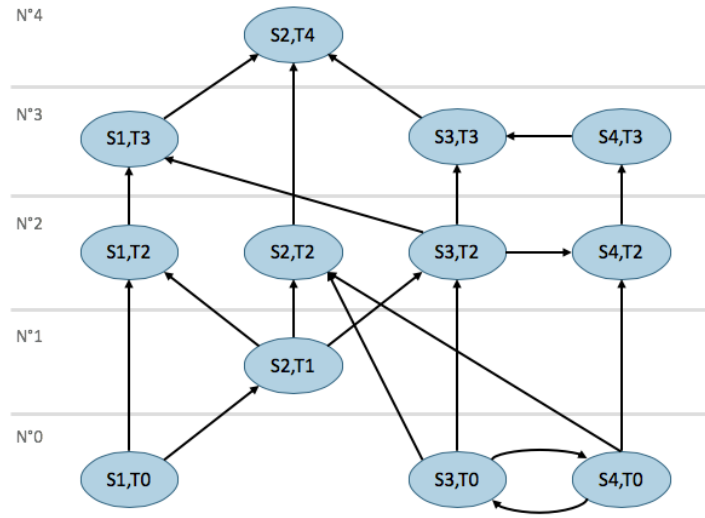
Cette section détaille une manière de modéliser un réseau de TC. L'approche employée permet de calculer la durée du chemin le plus court entre chaque paire de stations. Elle supporte également d'être mise à jour afin de correspondre à une nouvelle configuration du réseau. Les notations suivantes seront utilisées dans la suite de ce chapitre.

Les noms de station de transport en commun	→ $S_x, S_y, S_z$
Les nœud avec station et temps associées	→ $(S_x, T_i), (S_y, T_j)$
Les noms de nœud	→ $s, u, v, d$



**Figure 3.2** – Graphe statique du métro de Bruxelles  
<https://www.visitonsbruxelles.com/metro>

Le réseau de TC est modélisé sous la forme d'un *fully time-expanded graph* tel que proposé par Schulz [9]. Cette modélisation prend en compte le moment de la journée où les trajets sont effectués. Pour cette raison, elle a été préférée à la représentation sous forme de graphes statiques où les nœuds représentent les stations et les arêtes les connexions (Figure 3.2). En effet, considérer le moment de la journée permet de représenter les temps d'attente aux correspondances et de supporter des durées de trajet variant au cours de la journée pour un même déplacement direct entre deux stations. Cette modélisation nécessite d'utiliser des temps discrets. Vu que les horaires sont fournis en minutes, cette unité de temps a été choisie. Schulz [9] construit le *fully time-expanded graph* (Figure 3.3) comme suit :



**Figure 3.3** – Fully time-expanded graph

- **Nœuds** : Pour chaque évènement à une station (départ ou arrivée d'un TC), un nœud est construit sous la forme d'une paire (*station, temps*). Par conséquent, chaque station est représentée par un ensemble de nœuds associés à chacun de ces moments clés.
- **Arêtes** : Les arêtes indiquent la possibilité de se déplacer d'un nœud à l'autre. Une arête est créée pour chaque déplacement direct entre deux stations. Leur création suit donc les horaires de TC. D'autres arêtes sont générées pour représenter la possibilité de rester à une station en attendant un autre TC. Pour chaque station  $S_x$ , une arête est donc placée entre ses paires de nœuds représentant des évènements successifs :  $(S_x, T_i) \rightarrow (S_x, T_{i+1})$ .

À cela, ce mémoire ajoute des arêtes pour représenter la possibilité de marcher à pied entre 2 stations. La création de ces arêtes est détaillée dans la section 3.3.1.

Le graphe résultant de cette modélisation est un graphe dirigé. Ce qui est logique vu qu'on ne peut pas remonter le temps. Par conséquent, pour toute arête  $(S_x, T_i) \rightarrow (S_y, T_j)$ , le temps correspondant au nœud destination sera plus grand ou égal que celui correspondant à l'origine.

$$(S_x, T_i) \rightarrow (S_y, T_j) \implies T_i \leq T_j$$

Le graphe ne contient aucun cycle excepté entre des nœuds correspondants à un même temps. Ce qui peut arriver lorsque des trajets ont une durée arrondie à 0 minute. Les nœuds peuvent donc être hiérarchisés en fonction du temps afin de guider notre recherche d'itinéraire le plus court.

Une autre conséquence de cette représentation est que le temps nécessaire pour parcourir une arête est défini implicitement par les deux nœuds à ces extrémités :  $Temps\_trajet((S_x, T_i), (S_x, T_j)) = T_i - T_j$ . On a opté pour ne pas dupliquer cette information en attribuant un poids représentant la durée de trajet aux arêtes. En poussant le raisonnement un peu plus loin, on remarque qu'il suffit de savoir s'il existe un chemin entre deux nœuds quelconques pour pouvoir retrouver l'**unique** durée de trajet.

### 3.3.1 Possibilité de marcher entre les stations

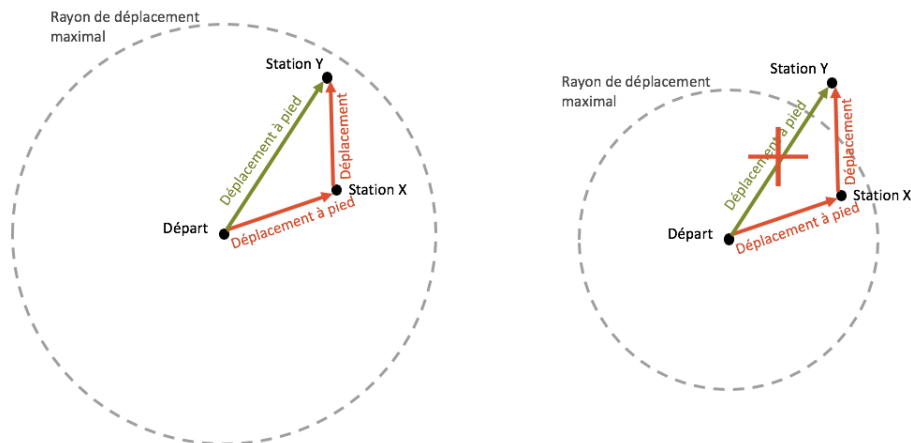
Pour se rapprocher de l'utilisation réelle des transports en commun, il est fondamental d'intégrer la possibilité de marcher à pied entre les stations. Pour ces trajets à pied, on utilisera la même méthode d'évaluation et les mêmes contraintes que dans la section 2.2.1. Les trajets sont approximés par un trajet à vol d'oiseau effectué à la vitesse de 3,6 km/h. De plus, ils sont limités à une durée de 30 minutes, soit un parcours de maximum 1,8 km. Cette dernière hypothèse sert à limiter le nombre de trajets à pied entre 2 stations.

L'utilisateur a la possibilité de partir à n'importe quel moment d'une station quelconques  $S_x$  pour rejoindre une station  $S_y$ . Une première piste pour représenter cela serait de créer un nœud pour chaque temps discret de chaque station et placer des arêtes entre ceux-ci lorsqu'un déplacement est possible (3.5a). Cependant, Schulz [9] avait empêché que le graphe temporel ait une taille gigantesque en considérant uniquement pour chaque station les temps correspondants à des événements. Ce qui a réduit le nombre de nœuds ainsi créés. Cette solution d'intégration de la marche nécessite l'ajout de nombreux nouveaux nœuds et arêtes augmentant ainsi énormément la taille du graphe.

Sans modifier les durées minimales de trajet, une deuxième piste de modélisation des déplacements à pied serait de se limiter à ceux partant et/ou arrivant d'un nœud préexistant 3.5b. Pour chaque nouvelle marche représentée, au plus un nouveau nœud serait alors créé. Cet ajout de nœud reste néanmoins problématique. En particulier si l'on permet à l'utilisateur d'effectuer plusieurs trajets d'affilée à pied. En effet, à partir de chaque départ/arrivée d'un TC, un nœud sera potentiellement créé pour représenter un déplacement à pied. A partir de ces nouveaux nœuds, d'autres nouveaux nœuds pourraient être créés, etc.

Heureusement, il est concevable de supprimer la possibilité d'effectuer plusieurs trajets à pied de suite. En effet, si l'utilisateur a la possibilité d'effectuer un trajet à pied allant directement de la station  $S_x$  à la station  $S_y$ , il s'agit d'un trajet en ligne droite qui est par conséquent le plus court. S'il n'a pas cette possibilité de trajet direct entre  $S_x$  et  $S_y$ , l'utilisateur pourrait passer à pied d'une station à l'autre pour

atteindre la station  $S_y$  qu'il ne pouvait pas atteindre à l'origine. Ces trajets vont à l'encontre de la limite sur la distance maximale parcourue à pied en une fois. Leur modélisation est donc soumise à débat : la représentation de trajets successifs n'est pertinente que lorsqu'une limite sur le temps de trajet à pied a dû être imposée et qu'on souhaite que l'utilisateur puisse contourner cette restriction.



**Figure 3.4** – Déplacements à pied directs et déplacements successifs

La modélisation de la marche utilisée dans ce mémoire (Figure 3.5c) ajoute uniquement des arêtes entre les nœuds pré-existant. Aucun nouveau nœud n'est donc créé. En contrepartie, elle ne représente pas certains enchaînements de trajets à pied. Pour un trajet à pied possible entre deux stations :  $S_x \rightarrow S_y$ , l'idée est de rallonger la durée de trajet en postposant le moment d'arrivée en  $S_y$  pour correspondre à un nœud existant. Afin de réaliser cela, pour chaque nœud associé à la station  $S_x$  :  $(S_x, T_x)$ , on calcule le temps minimal d'arrivée en  $S_y$  ( $T_{Ymin} = T_x + tempsAPied(S_x \rightarrow S_y)$ ) puis on crée une arête vers le premier nœud atteignable à pied de la station  $S_y$ . C'est à dire une arête entre  $(S_x, T_x)$  et  $(S_y, T_y)$  tel que

$$T_y = \min_{\substack{\forall T_j \geq T_{Ymin} \\ (S_y, T_j) \in noeuds}} T_j$$

Avec cette technique, les trajets sont limités arbitrairement à ceux commençant et finissant par une utilisation des TC. Les trajets calculés ne sont pas forcément les plus courts vu qu'une possibilité de marche est omise en début et en fin de parcours. Cependant, ce n'est pas un souci avec la mesure d'efficacité des TC utilisée actuellement (voir chapitre 2). En effet, un trajet à pied est effectué avant et après l'utilisation du réseau de transport en commun. Ce qui remplace donc les marches manquantes. Par ailleurs, si on en avait eu besoin, on aurait pu intégrer une étape de marche à pied avant et après l'utilisation du réseau de TC afin d'obtenir les vrais temps minimaux.

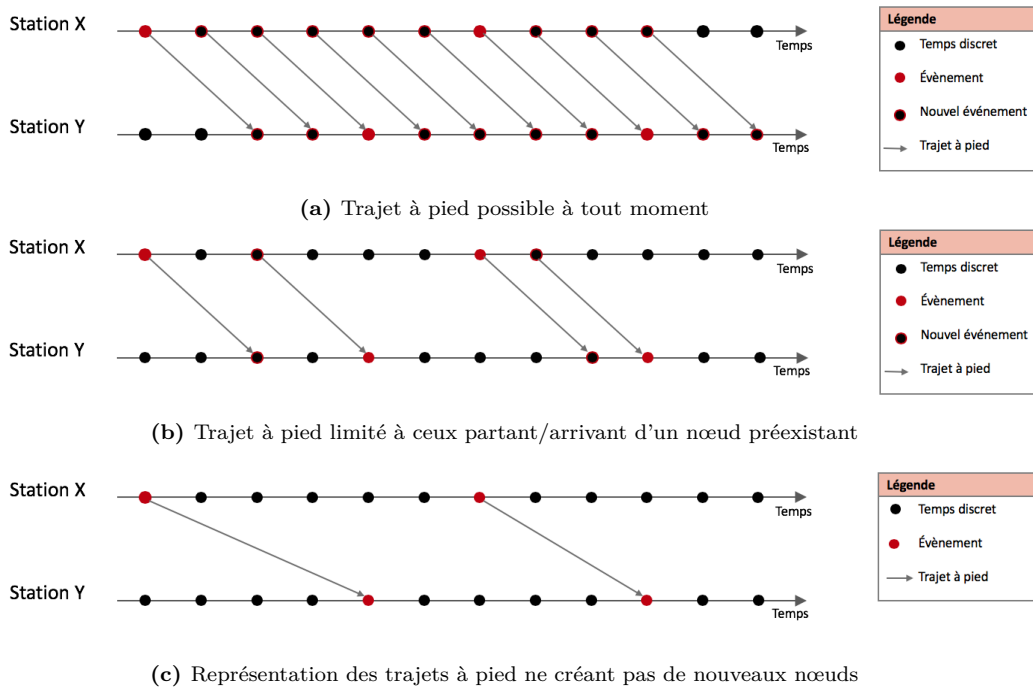


Figure 3.5 – Pistes de modélisation envisagées pour les déplacements à pied

## 3.4 All-Pairs Shortest Paths au sein d'un graphe

Cette section montre comment calculer le chemin le plus court entre tous les nœuds du graphe temporel représentant le réseau. Elle correspond à la seconde étape : niveau interne, de la découpe en sous problème présentée section 3.2. Dans un premier temps, on commencera par l'état de l'art des algorithmes permettant de calculer dynamiquement le chemin le plus court entre toutes les paires de nœuds. L'algorithme adopté sera ensuite détaillé. Finalement, on montrera comment initialiser l'algorithme et comment il a été adapté aux graphes temporels.

### 3.4.1 Dynamic APSP : État de l'art

Le problème consistant à trouver le plus court chemin entre toutes les paires de nœuds est connu, en théorie de graphes, sous le nom de *All-Pairs Shortest Paths (APSP)*. En outre, des algorithmes dynamiques ont été développés afin de pouvoir mettre à jour l'intégralité des distances en cas de légère modification du graphe. Ces algorithmes se divisent en plusieurs catégories en fonction des modifications autorisées :

- *Incremental APSP* : Ces algorithmes permettent d'améliorer le réseau. Les modifications permises sont l'ajout d'arête et/ou de nœud ainsi que la diminution du poids d'arête.
- *Decremental APSP* : Ces algorithmes permettent de détériorer le réseau. Les modifications permises sont la suppression d'arête et/ou de nœud ainsi que l'augmentation du poids d'arête.
- *Fully dynamic APSP* : Ces algorithmes sont à la fois incrémentaux et décrementaux.

Le problème du *dynamic APSP* a une longue histoire. En 1967, Loubai [10] développe une méthode pour évaluer l'effet de l'ajout ou de l'amélioration d'une connexion dans un réseau. Au cours des années suivantes, Ramalingam et Reps [11] proposent un algorithme incrémental pour résoudre dynamiquement le problème de la recherche du chemin le plus court. Cependant, la complexité d'une mise à jour, dans le pire des cas, équivaut à celle du calcul à partir de rien :  $O(mn + n^2 \log(n))$  où  $n$  est le nombre de nœuds et  $m$  le nombre d'arêtes. Cette méthode reste néanmoins intéressante car elle est plus rapide en pratique. En 2004, Demetrescu et Italiano [12] proposent un algorithme complètement dynamique qui est asymptotiquement plus rapide qu'un calcul depuis zéro. Cet algorithme est ensuite amélioré par Thorup [13] pour atteindre une complexité temporelle de  $O(n^2(\log(n) + \log^2(\frac{m+n}{n})))$  et une complexité spatiale de  $O(mn)$ .

Comme pour le *fully-dynamic APSP* problème, des avancées ont été faites pour l'*incrémental APSP*. Dernièrement, Slobbe [14] a développé, en se basant sur les travaux de Ramalingam et Reps, un algorithme incrémental plus rapide en pratique que les autres méthodes préexistantes et possédant une complexité spatiale et temporelle de  $O(n^2)$ . De plus, sa complexité spatiale est en  $O(n^2)$ , soit la même que le stockage des plus courtes distances sous forme de matrice.

### 3.4.2 Description des algorithmes de Slobbe

Au vu de la quantité de données à traiter pour modéliser le réseau belge, on a préféré les algorithmes incrémentaux proposés par Slobbe [14] afin de privilégier la vitesse d'exécution et limiter l'utilisation de la mémoire. Ce choix est fait au détriment de la possibilité de supprimer des stations et des lignes de TC. Slobbe propose deux algorithmes l'un dédié à l'ajout d'arêtes et l'autre à celui des nœuds.

#### Algorithme pour l'ajout d'arête

Lors de l'ajout d'une arête  $u \rightarrow v$ , on observe quelques propriétés clés. La première est que le chemin le plus court entre n'importe quel nœud source  $s$  et nœud destination  $d$  reste soit inchangé soit passe par la nouvelle arête  $u \rightarrow v$ . Cette propriété

est fondamentale. En effet, grâce à elle, calculer la distance après la modification  $dist'(s, d)$  se réduit à trouver le maximum entre l'ancienne distance  $dist(s, d)$  et  $(dist(s, u) + w'(u, v) + dist(v, d))$  avec  $w'(u, v)$  le poids de la nouvelle arête.

On parlera de sources (resp. destinations) affectées pour les nœuds dont la distance pour atteindre (resp. être atteinte par) au moins un autre nœud a été modifiée. Une autre propriété est qu'une source  $s$  ne peut être affectée que si tous les nœuds sur le chemin optimal de  $s$  à  $u$  sont aussi affectés. Le même principe est applicable pour les destinations : une destination ne peut être affectée que si tous les nœuds sur le chemin optimal de  $v$  à  $d$  sont aussi affectés. De plus, pour le nœud  $i$  qui précède  $j$  dans le chemin le plus court à partir  $v$ , l'ensemble  $A(i)$  des sources affectées par rapport à la destination  $i$  inclus  $A(j)$

La première étape de l'algorithme est d'identifier toutes les sources affectées. Pour cela, il exécute un *Breath First Search (BFS)* dont la racine est  $u$  et parcourant les arêtes à contresens. En outre, cette recherche n'explore que les arêtes des sources affectées. Déterminer si une source est affectée revient à vérifier si sa distance par rapport à  $v$  a changé. La seconde étape est d'identifier les destinations affectées et de mettre leurs distances à jour. Les destinations sont trouvées à l'aide d'un *BFS*. Pour chaque destination  $d$ , les distances sont mises à jour à l'aide de la formule  $dist'(s, d) = \max(dist(s, d), (dist(s, u) + w'(u, v) + dist(v, d)))$ . De plus, on ne parcourt que les sources pour lesquels la distance vis à vis du nœud précédant  $d$  a été affectée vu que  $A(d) \in A(\text{prédécesseur}(d))$ .

### Algorithme pour l'ajout de nœuds

L'ajout d'un nœud  $v$  possède des propriétés similaires à celles obtenues lors de l'ajout d'une arête : si la distance entre deux nœuds diminue alors le nouveau chemin doit passer par  $v$ . On a aussi les propriétés qui en découlent.

L'algorithme pour l'ajout de nœuds est similaire à celui pour les arêtes. La seule différence notable est l'utilisation de *Best First Search* à la place des *Breath First Search*.

### 3.4.3 Adaptation de l'algorithme et stockage des résultats

Dans la partie modélisation (3.3), on a relevé que la durée de trajet entre deux nœuds du graphe temporel était d'une part unique et d'autre part calculable à partir des dits nœuds : la durée de trajet est la différence entre les temps liés aux deux nœuds. Suite à cela, déterminer la durée du chemin le plus court entre une paire de nœuds se réduit à trancher entre l'existence et l'absence de chemin entre ceux-ci. L'algorithme de Slobbe [14] a été adapté à ce changement d'objectif.

- Au lieu d'enregistrer des distances, la présence ou l'absence d'un chemin est sauvegardée dans une matrice de booléens.
- Les conditions analysant si la distance entre deux nœuds a diminué ont été remplacées par des conditions testant si un nouveau chemin a été créé entre deux nœuds anciennement non-connectés. Plus concrètement, lorsque l'algorithme pour les graphes pondéré utilise une condition du type :  $dist(u, w) > w'(u, v) + dist(v, w)$ , cette dernière est remplacée par  $not(is\_path(u, w)) \& is\_path(u, v) \& is\_path(v, w)$  avec  $is\_path(u, v)$  qui renvoie `true` ssi il existait un chemin entre  $u$  et  $v$  avant la modification.

### 3.4.4 Calcul initial de la présence de chemin entre les nœuds

Maintenant que la technique de mise à jour de la matrice de présence de chemin à été définie, il reste un dernier point à aborder : son initialisation. Pour toutes les paires de nœuds, il s'agit trancher entre la présence et l'absence de parcours les reliant. On parlera d'un nœud initialisé lorsqu'on a déterminé tous les nœuds qu'il pouvait atteindre.

Le calcul de la matrice de présence de chemin repose sur ce principe suivant : en partant d'un nœud quelconque  $u$ , il est possible d'atteindre le nœud lui-même, les successeurs de  $u$ , tout nœud atteignable depuis un successeur de  $u$ . Par conséquent, lorsque tous les successeurs de  $u$  sont initialisés, il devient aisé d'initialiser  $u$ . Par ailleurs, les graphes temporels peuvent être hiérarchisés par temps. On regroupe les nœuds par niveaux correspondants à un même temps (Figure 3.3). Puis, les niveaux sont ordonnés par temps décroissant. On commence par initialiser les nœuds en partant de ceux du niveau de temps maximal. Ces derniers ne peuvent avoir de chemin que vers eux-même et entre-eux. On descend ensuite dans les niveaux de temps jusqu'à ce que tous les nœuds soient initialisés. L'initialisation des nœuds de chaque niveau est facilitée grâce aux résultats des niveaux de temps supérieur.

## 3.5 Calcul des temps de trajet minimaux

Cette section est la dernière étape de résolution dans le calcul des temps de trajet minimaux entre chaque paire de stations d'un réseau de transport en commun. Elle montre comment passer de la matrice représentant la présence d'un chemin entre les nœuds du graphe temporel aux temps de trajet minimaux entre stations.

**Calcul initial de l'ensemble de temps minimaux** Pour toute paire de stations, on cherche le temps de trajet minimal parmi tous ceux possibles dans la

tranche horaire observée. Pour cela, on trouve les deux ensembles de nœuds  $S$ ,  $D$  associés respectivement à la station source et destination. On parcourt ensuite l'ensemble des paires  $\{(s, d) | s \in S, d \in D\}$  à la recherche du temps de trajet minimum. Le temps de trajet de  $s$  à  $d$  est obtenu en vérifiant qu'il existe un chemin liant  $s$  à  $d$  puis en calculant la différence entre les temps liés aux nœuds  $s$  et  $d$ .

**Mettre à jour les temps minimaux** Comme toutes les autres structures, les temps de trajet minimaux peuvent être impactés par l'ajout d'une nouvelle ligne de TC. À cette étape, ces modifications sont perçues comme des changements au sein de la matrice de présence de chemin. Pour éviter de devoir parcourir l'entièreté de la matrice, l'ensemble des changements sera rapporté sous forme de liste. Il est à noter que les changements iront toujours vers une augmentation du nombre de chemins possibles vu que seules des ajouts au réseau sont permis. Par conséquent, chaque changement représente pour une paire de nœuds le passage d'une absence de chemin vers une présence de chemin. Mettre à jour l'ensemble des temps de trajet minimaux par rapport à une nouvelle configuration de réseau revient donc à voir, pour chaque changement, si le nouveau trajet qu'il permet est plus court que celui enregistré et le cas échéant à adapter le temps de trajet minimum concerné. Par ailleurs, les temps de trajet n'ont pas besoin d'être mis à jour à chaque modification mais uniquement quand on cherche à accéder à ces valeurs.

## 3.6 Analyse de performances

L'objectif de cette section est d'évaluer le temps d'exécution pour une modification du réseau ainsi que celui nécessaire à l'initialisation des temps de trajet entre toutes les paires de stations.

On a commencé par observer les paramètres influençant la structure du réseau en terme de nombre de stations et de connexions. En particulier, on a regardé l'effet de l'intervalle de temps choisi et la zone géographique considérée sur le nombre de stations et connexions présentes dans le réseau. Ces observations se trouvent en annexe. Pour la suite de notre analyse, quelques arrondissements ont été sélectionnés pour représenter des configurations de réseau possibles. Nous avons fait le choix d'analyser les performances sur un réseau réaliste. Cependant créer aléatoirement un réseau fictif pourrait également être intéressant pour fixer le nombre de nœuds et arêtes.

Afin d'évaluer le temps d'exécution nécessaire pour effectuer une modification, on a mesuré le temps d'exécution avec un MacBook Pro (Annexe A) pour 4 types de modifications :

- L'ajout d'une **arête** entre deux nœuds préexistants. On sélectionne aléatoirement les deux stations, puis le temps de trajet est défini de manière à avoir

une vitesse tirée entre 10 km/h et 20 km/h<sup>5</sup>. Finalement, les temps de départ et d'arrivée sont choisis aléatoirement parmi ceux **existant** et permettant le trajet.

- L'ajout d'une **arête** entre deux nœuds chacun associé à une station préexistante et à un temps aléatoire. On utilise quasiment la même procédure que pour les nœuds préexistants.
- L'ajout d'un nœud représentant un nouveau temps dans une station déjà existante. Aucun trajet en transport en commun n'est généré à partir de ce nœud. Seuls les trajets à pied sont possibles.
- L'ajout d'un nœud représentant une nouvelle station (nouveau nom et position).

La Figure 3.6a indique, pour chaque type d'ajout, le temps moyen d'exécution mesuré sur 25 modifications aléatoires. Ces temps sont calculés pour les arrondissements de Dixmude, Ath, Tournai, Nivelles et Charleroi placés en abscisse en fonction de leurs nombres d'arêtes. On observe que le temps d'exécution augmente avec le nombre d'arêtes. Chaque type d'ajout n'a pas le même temps d'exécution moyen. En particulier l'ajout d'une connexion réaliste entre deux nœuds existants est significativement plus rapide.

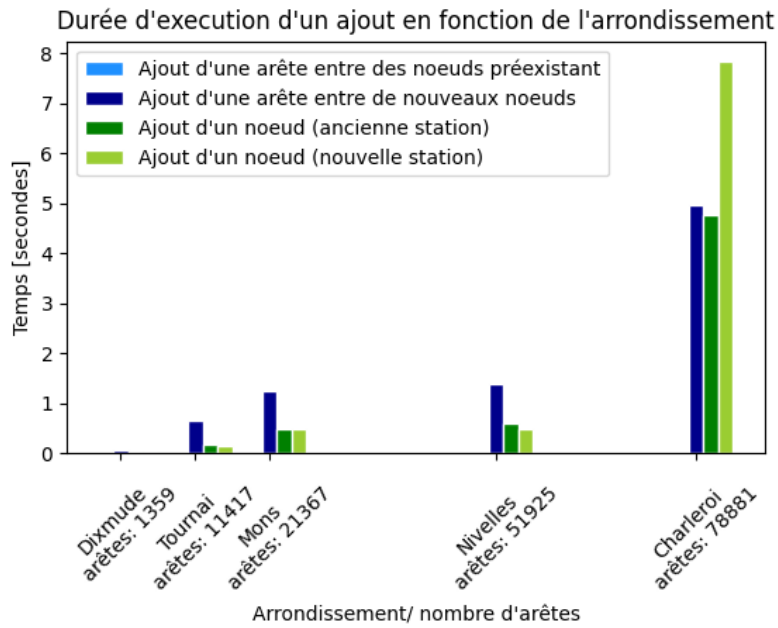
L'histogramme (Figure 3.6b) montre la différence de temps d'exécution entre les phases de modification et la phase d'initialisation où l'ensemble des temps de trajet est calculé. Il est en échelle log-log pour des raisons de lisibilité. On observe que le temps d'initialisation augmente avec le nombre de connexions. Il semble environ 100 fois plus long que le temps pour les 3 derniers types de modifications et ce quel que soit l'arrondissement sélectionné.

Par ailleurs, on a également analysé le temps d'initialisation en fonction de la limite sur le temps de trajet à pied. Les résultats se trouvent en annexe.

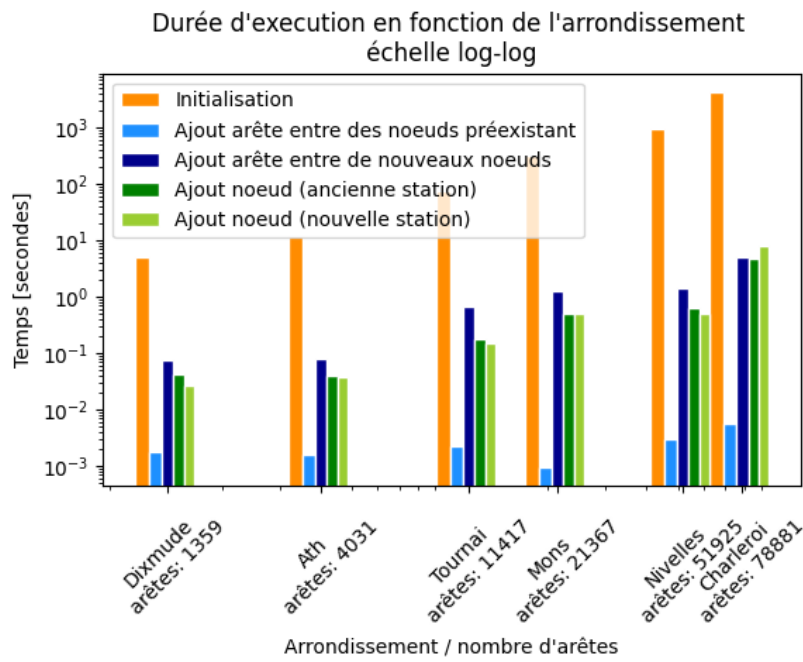
---

5. Ce qui correspond à une estimation de la vitesse commerciale pour des lignes d'autobus se déplaçant en site banal

Source : <http://mobilite.wallonie.be/files/eDocsMobilite/Centre%20de%20doc/CeMathèque/cematheque33.pdf>



(a) Durée d'exécution lors d'une modification en fonction de l'arrondissement



(b) Comparaison des durées d'exécution : initialisation et modifications

**Figure 3.6** – Évaluation des durées d'exécutions

## 3.7 Pistes d'améliorations

Au sein du graphe temporel, nous avons utilisé un algorithme incrémental pour la recherche du chemin le plus court entre toutes les paires. Cependant, vu la structure du graphe, on s'intéresse uniquement à la présence d'un chemin entre chaque paire de nœuds. Suite à ce constat, deux pistes d'améliorations peuvent être envisagées. La première consiste à voir si un algorithme vérifiant la présence de chemin entre toutes les paires de nœuds serait plus performant. La seconde piste consiste à utiliser le potentiel de l'*APSP* en attribuant un poids aux arêtes du graphe afin de minimiser à la fois le temps et un autre critère (par exemple, le nombre de correspondances à l'aide d'un poids de 1 aux arêtes).

# Chapitre 4

## Amélioration d'un réseau

L'objectif de cette section est de pouvoir comparer plusieurs nouvelles lignes potentielles de transport en commun (TC) afin de déterminer laquelle est la plus intéressante. Pour déterminer l'intérêt d'une ligne, la mesure d'efficacité développée au chapitre 2 sera utilisée. Il s'agit donc de trouver la ligne apportant la plus grande différence d'efficacité parmi plusieurs lignes possibles. En outre, l'optimisation d'un réseau de TC peut aller au-delà de la recherche d'une unique meilleure ligne. D'autres types de recherches pourraient être effectués tels que l'amélioration avec le meilleur rapport efficacité/coût ou encore la meilleure paire d'amélioration, etc. Pour laisser la porte ouverte à l'implémentation de ces autres méthodes d'optimisation, les fonctions utilisées sont assez génériques.

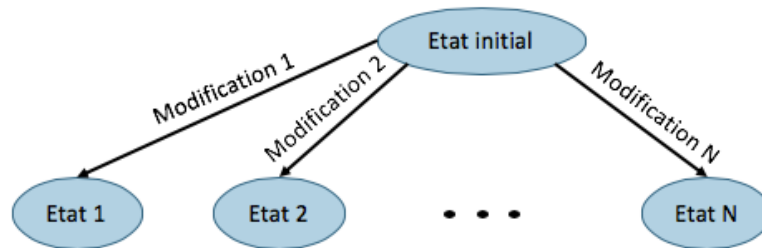
### 4.1 Mesurer l'effet d'une modification

Les chapitres 2 et 3 vus précédemment développent une mesure pour l'efficacité d'un réseau de TC. Celle-ci est utilisée pour quantifier l'effet de chaque modification. Ces chapitres montrent également comment représenter des modifications apportées à un réseau tel que l'ajout de nouvelles stations de TC et de lignes entre stations. Ils expliquent également comment la mesure peut-être rapidement recalculée suite à ces changements. Par conséquent, pour n'importe quelle modification ou série de modifications, on dispose d'une méthode pour mesurer la différence d'efficacité.

### 4.2 Algorithme d'optimisation

Pour déterminer la meilleure modification parmi celles proposées, une tactique consiste à tester indépendamment chaque modification et sauvegarder la meilleure (c'est à dire, celle qui a le plus faible temps de trajet moyen). Afin d'éviter de

recalculer l'état initial, un arbre de recherche est créé pour tester tout à tour chaque modification (Figure 4.1).



**Figure 4.1** – Arbre de recherche à un niveau utilisé pour trouver la meilleure modification parmi une série proposée.

Pour parcourir cette structure, il est nécessaire de pouvoir appliquer chaque modification sur l'état initial du réseau. Cependant, chaque modification change la structure du réseau. Face à ce problème, la solution employée consiste à annuler les changements effectués. Cette annulation est réalisée par le biais de deux fonctions : **save** et **restore**<sup>1</sup>.

- **save**, crée une sauvegarde de l'état actuel de la structure de données.
- **restore**, revient à l'état enregistré par le dernier **save** effectué. La sauvegarde est ensuite supprimée.

Certaines structures de données implémentent également la fonction **get\_change** qui renvoie une liste contenant tous les changements effectués depuis le dernier appel à **save**.

L'algorithme 4.1 permet de trouver la meilleure amélioration d'un réseau en testant tour à tour chaque modification proposée. Avant chacune d'entre elles, il utilise la fonction **save** permettant de sauvegarder l'état du réseau. Puis, il transforme le réseau et mesure son efficacité. Ensuite, il se sert de la fonction **restore** pour retrouver l'état initial du réseau. Un fois chaque amélioration envisagée, la plus intéressante est renvoyée ainsi que sa valeur.

---

1. Le mécanisme de **save** et **restore** est inspiré de celui développé dans le cours *LINGI2365 - Constraint Programming* dispensé à l'UCLouvain

---

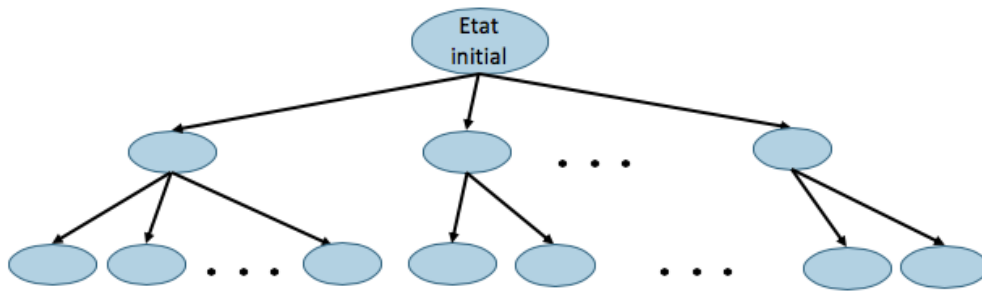
**Listing 4.1** – Trouver la meilleure modification parmi celles proposées

---

```
1 def find_best_modification(network, modifications: list):
2     best = None
3     min_value = math.inf
4     for modif in modifications:
5         network.save()          # sauvegarde l'état actuel du réseau
6         value = network.modify(modif)      # modifie le réseau
7         if value < min_value:
8             best = modif
9             min_value = value
10        network.restore()      # reviens à l'état avant les modifications
11    return best, min_value      # meilleure modif, valeur de la métrique
```

---

Le mécanisme de `save` et `restore` peut également être utilisé pour une recherche à plusieurs niveaux afin de trouver, par exemple, la meilleure combinaison de changements. En effet, plusieurs `save` et `restore` peuvent être effectués d'affilée afin d'aller plus profondément dans l'arbre de recherche (Figure 4.2).



**Figure 4.2** – Arbre de recherche à plusieurs niveaux

Un algorithme est fourni en Annexe D à titre d'illustration.

### 4.3 Annuler des modifications : save et restore

Pour réaliser la partie optimisation, toutes les structures de données utilisées supportent le mécanisme de `save` et `restore` permettant d'annuler les modifications effectuées. Pour cela, deux stratégies sont utilisées pour implémenter ce mécanisme.

L'idée de la première stratégie est d'annuler chaque action réalisée en effectuant celle qui lui est opposée (par exemple : ajouter/retirer, augmenter/diminuer, etc).

Celle-ci est concrétisée de la manière suivante : lors d'un appel à la fonction `save`, une nouvelle liste est créée et elle servira à enregistrer les actions accomplies. Pour chaque appel à une fonction modifiant la structure de données concernée, le type d'action effectuée ainsi que ses arguments sont ajoutés à la liste. Quand `restore` est appelé, la liste est parcourue en partant de la dernière action effectuée. Pour chacune d'elles, l'action opposée est réalisée afin de l'annuler. La liste est ensuite supprimée.

Cette stratégie d'annulation nécessite de disposer, pour chaque modification possible, d'un opposé permettant de l'annuler. Elle sera donc utilisée pour le graphe temporel où le retrait de nœuds et d'arêtes est aisé à implémenter. Pour tout ce qui concerne le calcul des plus courts chemins (*APSP*), on a implémenté uniquement l'ajout de nœuds et d'arêtes et non leur suppression. Une seconde stratégie est donc utilisée.

La seconde stratégie est une version améliorée de la technique consistant à copier l'entièreté de la structure de données. L'astuce est de sauvegarder uniquement les valeurs modifiées. Lors d'un `save`, une nouvelle zone de sauvegarde est créée. Dès qu'une variable est sur le point d'être modifiée, son nom et sa valeur sont ajoutées à la zone de sauvegarde, à moins qu'elle ne s'y trouve déjà. Ensuite la modification a lieu. De cette façon, n'importe quelle valeur de l'état sauvé peut être récupérée soit via la sauvegarde, soit directement sur la structure de données. Lorsque `restore` est appelé, on rétablit l'ancienne valeur de toutes les variables modifiées en parcourant la sauvegarde.

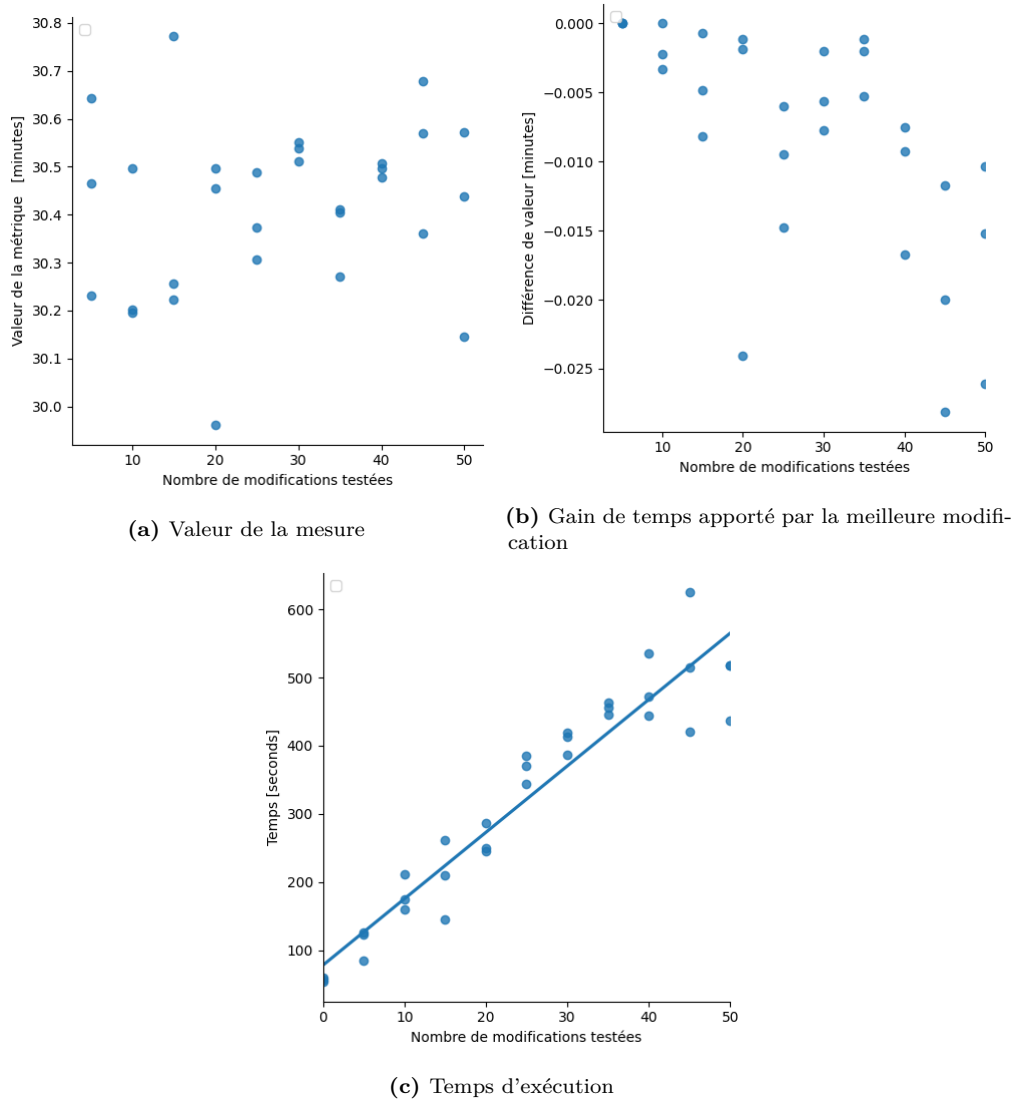
Afin de permettre plusieurs `save` ou `restore` d'affilée, on ajoute à ces deux stratégies une technique pour stocker des sauvegardes successives. Cette dernière s'imbrique dans les stratégies d'annulation. Elle repose sur l'utilisation d'une pile. Lors d'un `save`, une étape est ajoutée : elle consiste à mettre l'ancienne sauvegarde sur la pile. Pour le `restore`, une étape finale est ajoutée : retirer la sauvegarde du sommet de la pile et la considérer comme la zone de sauvegarde courante.

## 4.4 Résultats et performances

On observera les résultats produit par l'optimisation tel que présenté par l'algorithme 4.1. On s'intéressera ensuite au temps nécessaire pour exécuter cette optimisation.

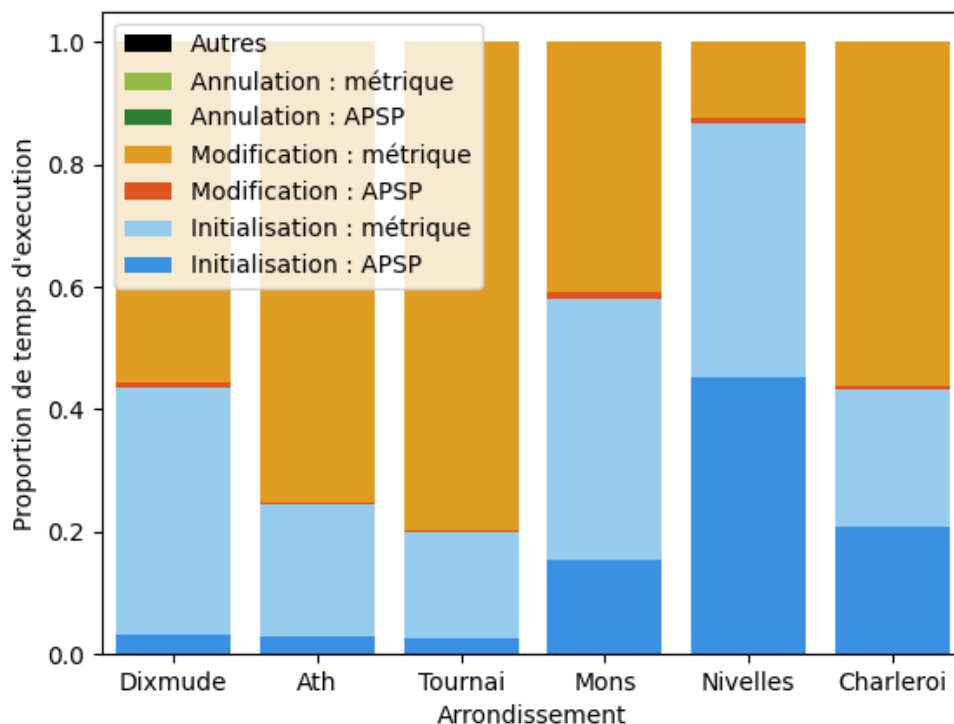
On veut voir l'effet du nombre de modification effectué vis à vis de la mesure d'efficacité. Pour cela, on a mesuré la valeur de la mesure avec la meilleure modification, le gain de valeur et le temps d'exécution pour un nombre de modification allant de 0 à 50. La version de la métrique utilisée ne prend en compte que les trajets initialement réalisables lors du calcul du temps de trajet moyen. L'arrondissement considéré est Dixmude avec un valeur de  $C = 1$ . Chaque modification testée correspond à l'ajout d'une ligne de TC entre 2 arrêts existants, effectué à une vitesse de 10-20km/h et démarrant à temps aléatoire. Pour chaque nombre de modification, on a effectué l'optimisation 3 fois afin d'avoir une vague idée de la variance.

Dans la Figure 4.3a, on a dû mal à déterminer l'impact de la meilleure ligne ajouté sur la métrique. Cela est dû à l'introduction d'aléatoire par la méthode de Monte Carlo. La valeur de la métrique varie d'une exécution à l'autre. Pour percevoir l'apport d'une nouvelle ligne il est donc plus adéquat d'évaluer la différence de valeur entre avant et après son intégration au réseau quant terme de valeur absolue. La Figure 4.3b confirme que de meilleurs ajouts de ligne sont trouvé lorsque de nombreuses modifications sont envisagées. Tester plus de ligne à cependant un coût en termes de temps d'exécution (Figure 4.3c).



**Figure 4.3** – Effet de la meilleure modification en fonction du nombre de modifications testées

On s'intéresse ensuite à la part du temps d'exécution utilisé par chaque étape de l'optimisation. Pour cela, on effectue une optimisation comme précédemment avec 15 modifications aléatoires. La Figure 4.4 représente les parts de temps d'exécution pour le calcul de la valeur initiale de la mesure, les 15 modification et l'annulation des modifications (*save* et *restore*). De plus, on a séparé le temps associé à la recherche des temps de trajet minimum entre chaque paire de stations (*APSP*) et le reste de calcul concernant l'évaluation de la mesure d'efficacité (*métrique*).



**Figure 4.4** – Répartition des temps d’exécution lors de la recherche de la meilleure amélioration parmi 15 proposées

Les étapes d’initialisation et d’amélioration prennent la quasi-totalité du temps d’exécution. La part associée à l’annulation des modifications est imperceptible. Cela permet d’identifier ce qu’il faudra améliorer en priorité. Grâce à la séparation *APSP/métrique*, on observe que l’évaluation de la mesure par la méthode de Monte Carlo prend énormément de temps. Revoir la méthode d’évaluation de la mesure d’efficacité sera donc une étape clé pour limiter le temps d’exécution. Par ailleurs, la part de temps associé à la recherche des temps de trajet minimum entre chaque paire de station semble raisonnable. Le mécanisme de `save` et `restore` implémenté permet d’annuler extrêmement rapidement par rapport au reste de l’algorithme. Optimiser le choix de la ligne de TC avec ce mécanisme semble donc pertinent.

# Chapitre 5

## Conclusion et discussion

Ce mémoire a détaillé le développement d'un outil ayant pour objectif d'améliorer l'efficacité des transports en commun. Le but était de déterminer parmi plusieurs nouvelles lignes de TC possibles laquelle serait la plus intéressante. Dans cette optique, une métrique afin de départager les lignes a été conçue. Celle-ci se base sur la moyenne des temps de trajets domicile-travail de l'entièreté de la population. Seuls les transports en commun et la marche ont été considéré en ignorant les autres moyens de transports : voiture, vélos, etc. Supposant que toute la population se déplace en transport en commun, l'impact du changement de comportement ne peut pas être étudié (par exemple, une personne qui troquerait la voiture pour le bus). Une solution serait d'exclure les personnes dont le temps de trajet en TC excède une certaine valeur en fonction de la distance à parcourir (typiquement le temps de trajet en voiture). Cependant, dans pareil cas, la mesure de temps de trajet moyen n'est plus suffisante pour juger le réseau : une personne passant de la voiture aux transports en commun pourrait faire augmenter la valeur moyenne des temps de trajet alors que son changement de comportement est positif. Des effets similaires ont été observés suite à la limitation du temps de la marche (effet présenté Figure C.1a). La mesure d'efficacité temporelle nécessiterait d'être complétée par une mesure de popularité. Notons également qu'il pourrait être intéressant d'étudier la combinaison de la voiture et des transports en communs. Par exemple, rouler jusqu'à la gare puis prendre le train.

Seuls les déplacements domicile-travail ont été considérés comme trajet à effectuer. Il est cependant évident que les transports ne sont pas utilisés que pour se rendre au boulot. Il serait intéressant de se baser sur une enquête afin d'évaluer la légitimité de cette hypothèse. Cette même enquête pourrait aussi valider le temps moyen de déplacement domicile-travail des belges.

Une autre hypothèse qui mériterait d'être discutée est le choix de l'itinéraire

uniquement basé sur le temps le plus court. Une plage horaire plus restreinte serait plus réaliste. En effet, l'utilisateur vise généralement une heure d'arrivée plus ou moins précise et pas seulement à faire son trajet entre 6h00 et 10h30. De plus, le choix du trajet dépend de bien d'autres facteurs, comme par exemple le nombre de correspondances et le délai entre celles-ci. L'utilisateur aura tendance également à se prémunir face à des retards éventuels de sa part ou des TC (privilégier les lignes à haute fréquence).

La mesure d'efficacité a été séparée en 2 parties (tant dans ce rapport que dans l'implémentation du code) : l'une se focalisant sur le réseau de transport en commun et l'autre l'utilisant tout en se concentrant sur les trajets souhaités par la population belge. L'outil étant conçu pour une utilisation future, la mesure d'efficacité peut être transformée sans avoir à réimplémenter tous les calculs de temps de trajet. Une version légèrement modifiée de la métrique a d'ailleurs été utilisée pour évaluer les temps moyens de trajet pour dans les communes du Brabant Wallon (Figure 2.3).

La limitation du temps d'exécution a été une grande préoccupation dans le développement de l'outil. Certaines hypothèses ont permis de réduire le nombre de trajets à examiner comme, par exemple, la limitation du temps de trajet à pied et son approximation par un déplacement "à vol d'oiseau". L'usage d'algorithmes incrémentaux a permis d'obtenir des temps d'exécution plus faible que ceux pour une réévaluation complète lorsque le réseau est modifié. Ce gain de temps est surtout notable dans le calcul des temps de trajet entre les stations de TC où un facteur 100 a été observé. La durée d'exécution peut toujours être diminuée : il a été observé que l'initialisation et la modification de la métrique par la méthode de Monte Carlo prenait la majeure partie du temps de calcul. Un point d'amélioration pour réduire la complexité calculatoire serait de repenser la méthode d'évaluation.

Malgré le peu de résultats finaux présenté, l'outil développé ouvre la voie à une optimisation exhaustive des transports en communs. Ce qui est aujourd'hui une thématique importante face, par exemple, au bannissement de l'automobile des centres urbains.

# Annexe A

## Caractéristiques de l'ordinateur

Les résultats ont été obtenus à l'aide d'un MacBook Pro de 2009.

<b>Processeur</b>	Intel Core 2 Duo - 2,93 GHz
<b>Mémoire vive</b>	4 Go - 1067 MHz
<b>Stockage</b>	Disque SSD - 250 Go

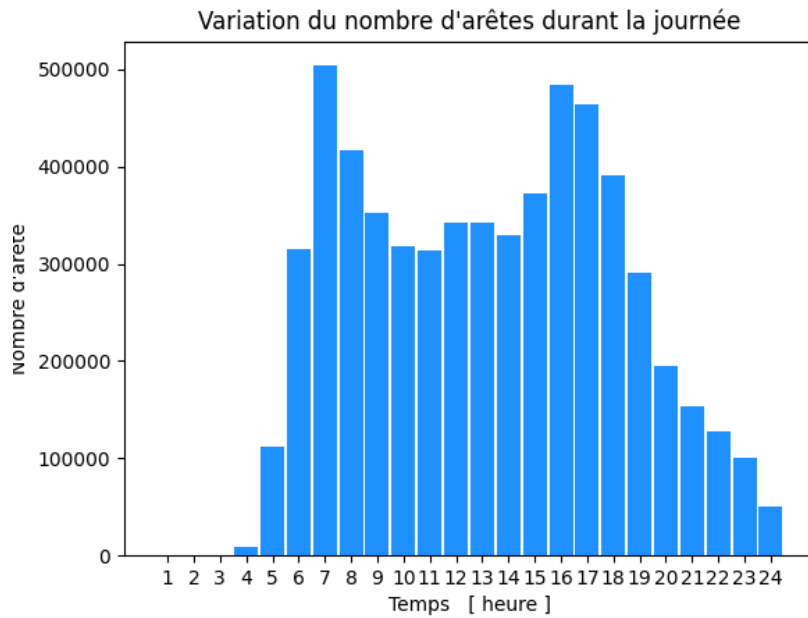
# Annexe B

## Brève étude des paramètres influençant la structure du réseau en terme de nombre de stations et de connexions

Cette annexe montre comment varient les paramètres influençant la structure statique du réseau de transport en commun. On regardera l'effet de l'heure de la journée sur le nombre de connexions en transport en commun. On montrera également le nombre de stations et de connexions présents dans chaque arrondissement de Belgique. Les arrondissements sélectionnés pour l'évaluation des performances et des résultats de ce mémoire seront mis en évidence.

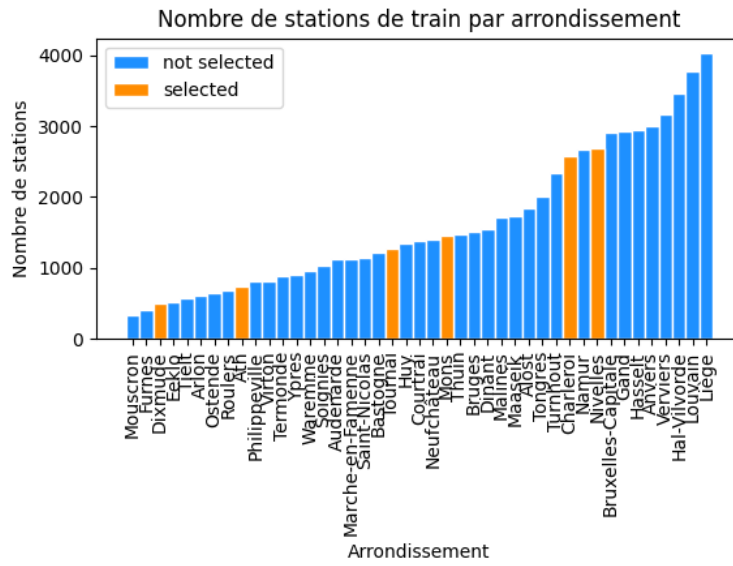
L'histogramme B.1 montre comment le nombre de connexions entre des stations de transport en commun varie durant la journée en Belgique. Il reprend, pour chaque tranche horaire d'une heure (ex : de 10h00 à 10h59), le nombre de connexions différentes démarrant dans ce laps de temps. On observe que le transport en commun circule principalement entre 6h et 20h avec des pics pour les tranches horaires de 7-8h et celles de 16-17h. La tranche horaire observée (de 6h00 à 10h30) profite donc du pic matinal.

La figure B.2 a été faite pour évaluer le nombre de stations et de connexions présentes dans chaque arrondissement. On observe que le nombre de stations varie entre 300 et 4000, tandis que le nombre de connexion observée va de 1400 à 650000. En outre, le nombre de connexions ne dépasse pas 100 000 pour la quasi totalité des communes, excepté pour Liège (environ 300 000) et Bruxelles (environ 650 000). Le nombre de connexions est donc plus important et varie plus que le nombre de stations. De plus, le nombre de stations et de connexions semblent liés (Figure B.2c).

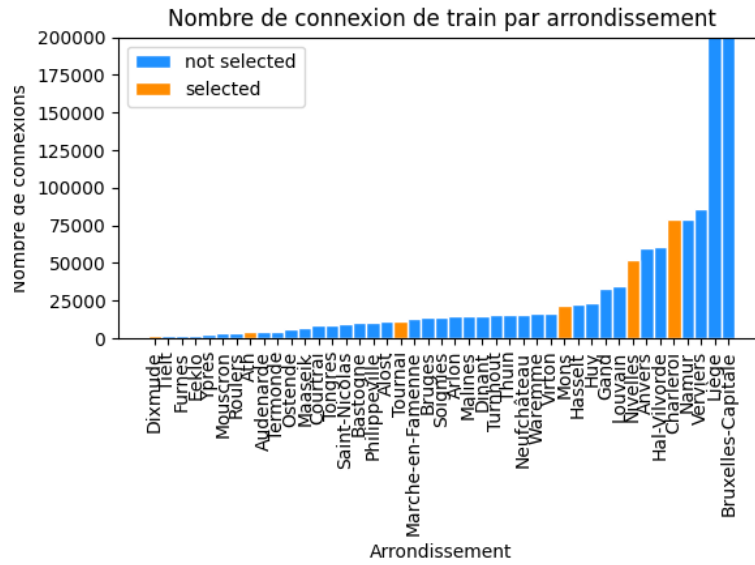


**Figure B.1** – Variation du nombre de connexions dans la journée

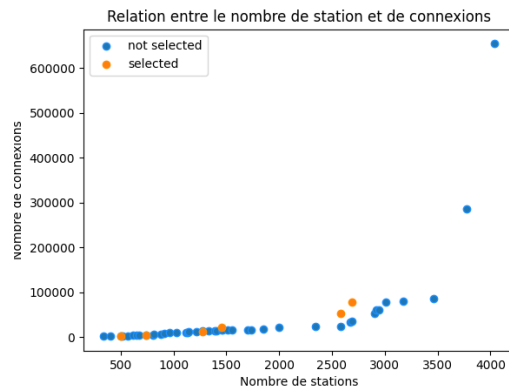
Pour l'évaluation des performances et des résultats présentés dans ce mémoire, nous avons préféré analyser les performances sur un réseau réaliste plutôt que de créer aléatoirement un réseau fictif. Quelques arrondissements ont été sélectionnés pour représenter des configurations de réseau possibles. Ils sont indiqués Figure B.2c. L'idéal aurait été de sélectionner également Bruxelles ou Liège mais leur évaluation demandait trop de temps que pour être effectué dans un délai raisonnable



(a) Nombre de stations par arrondissement



(b) Nombre de connexions par arrondissement



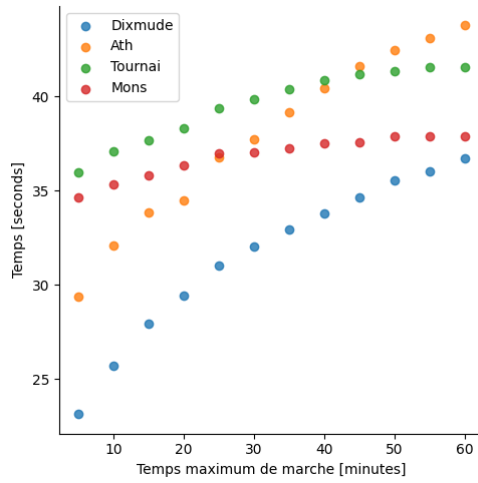
(c) Relation entre le nombre de stations et de connexions

**Figure B.2** – Analyse du nombre de stations et de connexions de transport en commun

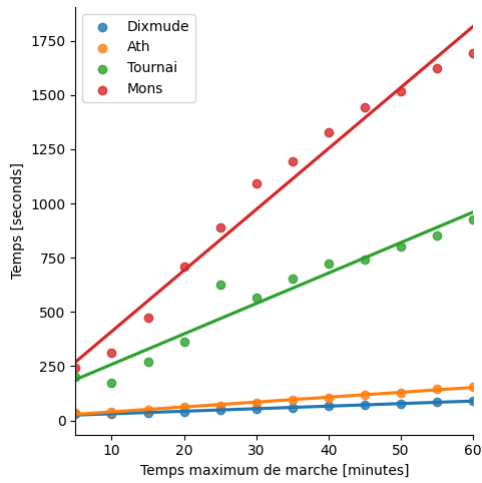
## Annexe C

# Effet de la limite sur le temps de trajet à pied

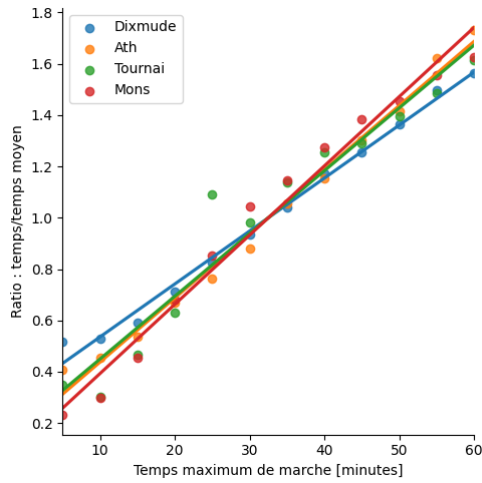
Les figures C.1 montre l'effet de la limite sur le temps maximale de trajet à pied en une traite (*MarcheMax*). La Figure C.1a montre la relation entre la valeur de la mesure d'efficacité et *MarcheMax* pour les arrondissement de Dixmude, Ath, Tournai et Mons. On observe que le temps de trajet augmente avec *MarcheMax*. En effet, autoriser des durées de marche plus longue a pour effet d'intégrer à la métrique des trajets de longue durée qui était jusque là ignorés. Les figures C.1b et C.1c donnent une idée de l'évolution du temps d'exécution nécessaire à l'initialisation de la mesure d'efficacité. La première indique la durée d'exécution absolue. On observe que le temps d'exécution semble relativement linéaire avec le temps de marche. La seconde figure met en évidence l'accroissement du temps de calcul. Pour chaque arrondissement, les durées d'exécution sont normées par la moyenne des durées d'exécution associés aux valeur de *MarcheMax*. On observe que les quatre arrondissement ont un taux d'accroissement similaire.



(a) Valeur de la mesure



(b) Temps d'exécution



(c) Ratio de temps

**Figure C.1** – Évolution de la mesure d'efficacité en terme de valeur et de durée d'exécution, en fonction de la limite de temps sur les déplacements maximums à pied.

# Annexe D

## Algorithme de recherche de la meilleure paire de modification

L'algorithme ci-dessous calcule grossièrement la meilleure paire de changements dans le réseau de TC en envisageant chacune d'entre elles.

**Listing D.1** – Trouver la meilleure paire de modifications parmi celles proposées

```
1 def find_best_Duo_modification(network, modifications: list):
2     best = None
3     min_value = math.inf
4     for i in range(len(modifications)):
5         network.save()
6         network.modify(modifications[i])
7         for j in range(i+1, len(modifications)):
8             network.save()
9             value = network.modify(modifications[j])
10            if value < min_value:
11                best = (modifications[i], modifications[j])
12                min_value = value
13            network.restore()
14        network.restore()
15    return best, min_value
```

---

# Bibliographie

- [1] David L Greene and Michael Wegener. Sustainable transport. *Journal of Transport Geography*, 5(3) :177–190, 1997.
- [2] Lauren Redman, Margareta Friman, Tommy Gärling, and Terry Hartig. Quality attributes of public transport that attract car users : A research review. *Transport policy*, 25 :119–127, 2013.
- [3] Laura Eboli and Gabriella Mazzulla. A stated preference experiment for measuring service quality in public transport. *Transportation Planning and Technology*, 31(5) :509–523, 2008.
- [4] Patrick Bonnel. Estimating transport demand. In *Geographical information and urban transport systems*, pages 117–165. 2011.
- [5] Chandra R Bhat and Frank S Koppelman. Activity-based modeling of travel demand. In *Handbook of transportation Science*, pages 35–61. Springer, 1999.
- [6] Andrey Petrov. One hundred years of dasymetric mapping : back to the origin. *The Cartographic Journal*, 49(3) :256–264, 2012.
- [7] Frédéric Héran. Des distances à vol d’oiseau aux distances réelles ou de l’origine des détours. *Flux*, (2) :110–121, 2009.
- [8] Alexis Frappier. *Méthode d’évaluation de la diversité et de la qualité des alternatives de déplacement de transport en commun*. PhD thesis, École Polytechnique de Montréal, 2015.
- [9] Frank Schulz. *Timetable information and shortest paths*. PhD thesis, Citeseer, 2005.
- [10] Peter S Loubai. A network evaluation procedure. *Highway Research Record*, (205), 1967.
- [11] Ganesan Ramalingam and Thomas Reps. An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms*, 21(2) :267–305, 1996.
- [12] Camil Demetrescu and Giuseppe F Italiano. A new approach to dynamic all pairs shortest paths. page 8.

- [13] Mikkel Thorup. Fully-dynamic all-pairs shortest paths : Faster and allowing negative cycles. In *Scandinavian Workshop on Algorithm Theory*, pages 384–396. Springer, 2004.
- [14] Arie Slobbe, Elisabetta Bergamini, and Henning Meyerhenke. Faster incremental all-pairs shortest paths. 2016.

**UNIVERSITÉ CATHOLIQUE DE LOUVAIN**  
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | [www.uclouvain.be/epl](http://www.uclouvain.be/epl)