Titre: Title:	Synchronisation en temps réel des correspondances de réseaux de transports publics
Auteur: Author:	Laura Kolcheva
Date:	2025
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Kolcheva, L. (2025). Synchronisation en temps réel des correspondances de réseaux de transports publics [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/68400/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/68400/
Directeurs de recherche: Advisors:	Antoine Legrain, & Martin Trépanier
Programme: Program:	Doctorat en génie industriel

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Synchronisation en temps réel des correspondances de réseaux de transports publics

LAURA KOLCHEVA
Département de mathématiques et de génie industriel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie industriel

Juin 2025

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Synchronisation en temps réel des correspondances de réseaux de transports publics

présentée par **Laura KOLCHEVA**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Michel GENDREAU, président
Antoine LEGRAIN, membre et directeur de recherche
Martin TRÉPANIER, membre et codirecteur de recherche
Issmaïl EL HALLAOUI, membre
Leandro C. COELHO, membre externe

DÉDICACE

На Баба Цеци.

À ma grand-mère.

REMERCIEMENTS

Tout d'abord, je souhaite remercier mes directeurs de recherche, Antoine Legrain et Martin Trépanier. Ils m'ont accompagnée depuis la maîtrise et ont été les premiers à m'encourager à poursuivre en doctorat. Ils m'ont fait confiance dès le départ, même lorsque je n'étais pas certaine de moi. Tout au long de la thèse, ils m'ont poussée à donner le meilleur de moi-même et ils m'ont énormément aidée dans l'avancement de ma recherche ainsi que dans la rédaction de mes articles. Ils m'ont également offert de nombreuses opportunités, en me confiant des projets complémentaires qui m'ont permis de grandir, d'explorer de nouvelles idées et de m'impliquer dans la gestion de projets stimulants. Ces travaux n'auraient pas été possibles sans vos précieux conseils !

Je remercie chaleureusement notre partenaire, la Société de Transport de Laval (STL). Leur intérêt pour mes travaux et leurs retours constructifs ont grandement enrichi ce projet. J'adresse un remerciement particulier à Grzegorz Wielinski et Vincent Dionne pour leur engagement et leurs encouragements au fil des années.

Je suis également reconnaissante envers les participants du projet interdisciplinaire Stratégia pour ces échanges fructueux, qui m'ont permis de découvrir de nombreuses facettes du transport public. Je pense en particulier à Nomessi Kokutse et Emmanuelle Reny-Nolin, du Réseau de transport de la Capitale, pour leur disponibilité et leur intérêt ; nos échanges m'ont souvent inspirée dans ma thèse. J'ai aussi beaucoup apprécié collaborer avec Fanny Tremblay-Racicot et Olivier Roy-Baillargeon, avec qui j'ai découvert les particularités de la mobilité servicielle. Merci également à Marie-France Vincent, avec qui j'ai eu le plaisir de collaborer sur la ludification des transports.

Je remercie les membres de mon jury — Michel Gendreau, Issmaïl El Hallaoui et Leandro C. Coelho — d'avoir accepté d'évaluer ce travail.

Je tiens également à remercier mes collègues de Giro Inc., où j'ai eu la chance de travailler à temps partiel pendant ma thèse. Un grand merci à toute l'équipe, notamment Aurélien B., Aurélien S., Étienne, Fanny, Paul et Thierno., pour votre accueil chaleureux. Merci en particulier à mon gestionnaire Loïc Bodart pour son accompagnement bienveillant et ses retours constructifs. Cette expérience m'a beaucoup appris, et j'ai grandement apprécié travailler avec vous.

Ma reconnaissance va aussi aux équipes du CIRRELT et du GERAD pour leur appui et leur disponibilité : votre travail est très apprécié par tous les étudiants ! Merci en particulier à

Guillaume Michaud, Pierre Girard et Khalid Laaziri, qui m'ont beaucoup aidée et remonté le moral lorsque j'avais des difficultés techniques. Merci à Patrick Munroe pour sa patience et bienveillance face à mes nombreuses questions. Merci à Lucie, Nathalie et Lucie-Nathalie : organiser des événements ensemble et venir travailler quand vous êtes là a été un plaisir. Merci également à tous les membres du CIRRELT et du GERAD que j'ai côtoyés pendant mes années de thèse. J'ai apprécié échanger des idées et partager ces moments avec vous. Merci à Nagisa pour sa compagnie au quotidien dans notre bureau partagé. Merci à toutes celles et ceux qui ont fait vivre le centre avec leur bonne humeur et nos nombreuses soirées karaoké !

Je souhaite terminer ces remerciements en adressant un immense merci à ma famille et à mes amis. Merci à mes parents et à mes frères et sœurs pour leurs encouragements constants, même lorsqu'ils ont suggéré avec bienveillance que je pourrais faire autre chose qu'un doctorat. Votre soutien m'a portée jusqu'au bout. Je suis également très reconnaissante envers mes amis, dont la présence et l'humour ont compté énormément tout au long de ce parcours : Albert, Anamaria, Flore, Jean-Charles, Liébaut, Luca, Lynn, Mohammad, Suzanne et Tenxiang. Vous connaître est une richesse au quotidien, et je suis une personne meilleure grâce à vous. Merci aussi à mes amis en dehors du Canada, qui ont su rester proches malgré la distance.

Enfin, je dédie cette thèse à ma grand-mère qui s'est battue pour accéder aux études supérieures et qui m'a appris à suivre ma propre voie. Elle était une grande femme, malgré sa petite taille, et son exemple m'a inspirée tout au long de ce doctorat.

Je souhaite enfin souligner le soutien financier que j'ai reçu du CIRRELT, du FRQNT, de la Chaire en transformation du transport, du CRSNG, de l'OBVIA et de Polytechnique Montréal.

RÉSUMÉ

Les transports publics jouent un rôle central dans l'organisation des villes en facilitant la mobilité des populations, en réduisant la congestion routière et en limitant les émissions de Gaz à Effet de Serre (GES). La fiabilité des correspondances constitue un enjeu majeur pour améliorer l'expérience des passagers et renforcer l'attractivité du transport collectif.

Cette thèse porte sur la synchronisation en temps réel des correspondances dans les réseaux de transport public, un levier clé pour améliorer la qualité du service et réduire les temps de parcours. L'objectif est de concevoir et d'évaluer des approches d'optimisation dynamique permettant d'améliorer la fiabilité des correspondances à travers des tactiques de contrôle en temps réel, telles que l'attente aux arrêts, le saut d'arrêt et le contrôle de vitesse.

Dans un premier temps, un modèle d'optimisation hors ligne basé sur des graphes temporels intégrant ces tactiques pour un horizon donné est développé. Il vise à minimiser le temps total de parcours des passagers en optimisant les correspondances tout en réduisant les écarts par rapport à l'horaire planifié. Ce modèle hors ligne constitue une preuve de concept et démontre sa capacité à traiter des instances impliquant plusieurs dizaines d'arrêts de contrôle et un nombre élevé de correspondances en temps réel. Pour évaluer l'efficacité du modèle hors ligne dans un environnement à information partielle, une version dynamique générant un scénario unique à chaque réoptimisation est développée. Différents niveaux d'incertitude sont considérés lors de la génération des scénarios. Les deux approches sont ensuite testées sur des données réelles de la ligne 70 du réseau d'autobus de la STL. Les résultats confirment l'efficacité des méthodes et montrent que la génération déterministe de scénarios offre les meilleures performances lorsque l'optimisation dynamique repose sur un unique scénario. Toutefois, un écart de performance subsiste entre le modèle hors ligne à information parfaite et son équivalent dynamique déterministe, soulignant les limites de ce dernier face aux aléas du réseau.

Ensuite, un cadre de simulation à événements discrets est développé pour intégrer les incertitudes inhérentes aux conditions de circulation et aux flux de passagers. Deux algorithmes d'Optimisation Stochastique En Ligne (OSEL) sont adaptés au problème de synchronisation des correspondances, exploitant des données historiques et en temps réel pour générer plusieurs scénarios et prendre des décisions dynamiques. Ces algorithmes sélectionnent les tactiques de contrôle optimales en considérant l'ensemble des scénarios simulés. Testés sur 29 lignes du réseau de la STL, ils permettent une amélioration significative du taux de correspondances réussies et une réduction du temps total de parcours des passagers. Les résultats

montrent que les algorithmes stochastiques en ligne atteignent une performance proche du modèle hors ligne tout en s'adaptant aux conditions dynamiques du réseau.

Enfin, l'approche est étendue à l'ensemble du réseau multiligne de la STL à l'aide d'un simulateur reproduisant les opérations à partir de données historiques détaillées. Le modèle hors ligne et les algorithmes d'optimisation stochastiques sont adaptés et implémentés dans le simulateur. L'objectif est d'évaluer l'impact d'une optimisation simultanée de plusieurs lignes sur la performance globale du réseau. Cette étude permet également d'évaluer différents contextes, comprenant des lignes à haute ou basse fréquence ainsi que des sous-réseaux maillés ou radiaux. L'analyse révèle le potentiel d'une régulation à l'échelle du réseau pour améliorer la fiabilité des correspondances sans générer de perturbations excessives.

Les travaux de cette thèse illustrent l'apport des approches d'optimisation stochastique en temps réel pour la synchronisation des correspondances de réseaux de transport public. Ces méthodologies, alliant efficacité computationnelle et robustesse face aux incertitudes, ouvrent des perspectives d'application concrètes pour les opérateurs de transport.

ABSTRACT

Public Transit (PT) plays a crucial role in urban development and in reducing congestion and greenhouse gas emissions. The reliability and timeliness of transfers are key factors influencing ridership and user experience in PT networks.

This dissertation focuses on real-time transfer synchronization in public transportation networks, an essential approach for improving service quality and reducing passenger travel times. The objective is to develop and evaluate dynamic optimization approaches that enhance transfer reliability by implementing real-time control tactics such as holding, skip-stop, and speed control.

First, an offline arc-flow model is developed using time-expanded graphs to integrate all possible control tactics within a given control horizon. The model minimizes total passenger travel time by optimizing transfers while reducing schedule deviations. As a proof of concept, it demonstrates the ability to handle large-scale instances, involving several dozen control stops and numerous transfer connections, in real time. To assess the performance of the offline model in an environment with partial information, a dynamic version generating a single scenario is designed. Different levels of uncertainty are evaluated in the scenario generation. Both methodologies are tested using real-world passenger demand and ridership data from bus route 70 of the STL network. Results confirm the effectiveness of this approach for transfer synchronization and indicate that, when considering a single scenario, a deterministic scenario generation method yields the best performance. However, a performance gap remains between the perfect information offline model and its deterministic dynamic counterpart, highlighting the latter's limitations in handling network uncertainties.

Next, a discrete-event simulation framework is developed to incorporate the inherent uncertainties of traffic conditions and passenger flows. Two *Online Stochastic Optimization* (OSO) algorithms, *CONSENSUS* (C) and *REGRET* (R), are adapted from the literature for the transfer synchronization problem, leveraging historical and real-time data to generate multiple scenarios and make robust decisions. These algorithms evaluate the solutions of the offline model for all scenarios to determine optimal control tactics at stops. Tested on 29 bus routes within the STL network, the algorithms significantly improve successful transfer rates and reduce total passenger travel time. The results demonstrate that OSO achieves performance levels comparable to the offline model while maintaining adaptability to dynamic network conditions.

Finally, the approach is extended to the full STL network using a network-wide simulator

replicating real-time stochastic conditions based on fine-grained, comprehensive historical data on network operations and passenger demand. The offline model and the online algorithms are adapted and implemented in the simulator. The objective is to evaluate the impact of the simultaneous optimization of multiple routes on overall network performance. The study also examines various contexts, including high- and low-frequency routes as well as grid and radial sub-networks. This analysis highlights the potential of network-wide synchronization to improve transfer reliability without causing excessive disruptions.

The work presented in this thesis demonstrates the contributions of real-time stochastic optimization approaches to synchronizing transfers in PT networks. These methodologies, which combine computational efficiency with robustness, open up concrete application opportunities for transport operators.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

AFC	*Automated Fare Collection*
APC	*Automatic Passenger Counter*
ARTM	Autorité Régionale de Transport Métropolitain
ATUQ	Association du Transport Urbain du Québec
AVL	*Automated Vehicle Location*
C	*CONSENSUS*
CMM	Communauté Métropolitaine de Montréal
CSV	*Comma-Separated Values*
D	*DETERMINISTIC*
E	*EXPECTATION*
GES	Gaz à Effet de Serre
GPS	*Global Positioning System*
GTFS	*General Transit Feed Specification*
NWS	*Network-Wide Simulator*
OSO	*Online Stochastic Optimization*
OSEL	Optimisation Stochastique En Ligne
PI	*Perfect Information*
PIB	Produit Intérieur Brut
PT	*Public Transit*
R	*REGRET*
SCRO	Société Canadienne de Recherche Opérationnelle
STL	Société de Transport de Laval

LISTE DES ANNEXES

CHAPITRE 1 INTRODUCTION

1.1 Contexte

Le transport public joue un rôle central dans le fonctionnement des sociétés modernes. Il facilite l'accès à l'emploi, aux soins de santé, aux services et aux activités de la vie quotidienne. En tant qu'outil structurant du territoire, il contribue au dynamisme des milieux de vie, renforce l'équité sociale et appuie la transition socioécologique.

Au-delà de son rôle social et environnemental, le secteur du transport public constitue un catalyseur économique puissant [1], notamment en stimulant la croissance locale. Au Québec, le secteur du transport représente environ 4% du Produit Intérieur Brut (PIB). Bien que le transport public y occupe une part plus modeste en termes de valeur marchande, ses retombées sont loin d'être négligeables. En 2022, les dix membres de l'Association du Transport Urbain du Québec (ATUQ) ont permis 330 millions de déplacements, soutenu plus de 28 000 emplois, injecté 4,7 milliards de dollars dans l'économie québécoise et généré près de 795 millions de dollars en revenus fiscaux pour les gouvernements provincial et fédéral [2]. À Laval, la STL a contribué à hauteur de 180 millions de dollars en valeur ajoutée. Le rapport entre la richesse créée et les dépenses initiales atteint 82% à l'échelle provinciale, et 79% pour la STL, ce qui illustre la rentabilité économique de ces investissements. Ces résultats témoignent de l'effet multiplicateur du secteur, dont les retombées s'étendent de l'amont (infrastructures, services) à l'aval (emplois, fiscalité, vitalité urbaine). En plus de stimuler l'activité locale, ces investissements contribuent à réduire les coûts sociaux liés à la pollution et à l'exclusion territoriale.

En revanche, la prédominance de la voiture comme mode de déplacement entraîne des coûts élevés pour les individus et pour la collectivité. Au Québec, le transport représente le deuxième poste de dépense des ménages, devant l'alimentation, avec une moyenne annuelle dépassant 10 000 dollars canadiens, dont moins de 10% sont consacrés au transport public. Cette situation s'explique en partie par une motorisation croissante : les ménages possèdent de plus en plus de véhicules, malgré une taille moyenne en diminution. À Laval, on comptait 1,51 véhicule par ménage en 2023, contre 1,27 en région et 0,94 à Montréal [3]. Au-delà de l'impact sur le budget des ménages, cette dépendance automobile a des conséquences économiques plus larges : une part importante des dépenses liées à l'achat de véhicules et de carburant est dirigée vers l'importation, limitant ainsi les retombées locales. Les dépenses annuelles des ménages québécois liées à l'automobile sont estimées à 37 milliards de dollars, dont plus de la moitié pour des biens importés [4]. À cela s'ajoutent des coûts indirects liés à la congestion, à la perte de productivité, à la dégradation de la qualité de l'air et à la pression sur les infrastructures.

Le secteur des transports est également l'un des principaux contributeurs aux émissions de GES, tant à l'échelle mondiale que locale. À l'échelle planétaire, il représente plus de 20% des émissions de GES qui pourraient augmenter de 60% d'ici 2050 en l'absence de mesures structurantes [5]. Au Canada, les transports comptent pour environ 25% des émissions annuelles, se classant au deuxième rang des secteurs les plus polluants [6]. Au Québec, le secteur du transport constitue la première source d'émissions, avec 43,3% du total provincial [7], et jusqu'à 57% sur le territoire lavallois [8]. Les véhicules particuliers jouent un rôle central dans ce bilan : ils sont responsables de 22% des émissions au Québec. La progression constante du parc automobile — près de 5 millions de voitures ou camions légers en 2022 — renforce cette tendance [9]. Dans ce contexte, le transport public représente un levier incontournable pour réduire les émissions, diminuer la pression sur les réseaux routiers et accompagner la transition énergétique.

Au Québec, les sociétés de transport public font face à des défis structurels, accentués par les effets de la pandémie et le développement du télétravail. À Laval, la part modale du transport public atteignait seulement 15% en 2018, contre 26% à Montréal [10]. Cet écart s'explique notamment par une desserte moins fréquente sur un territoire étendu et moins dense, ainsi que par l'absence d'un réseau structurant comparable au métro, beaucoup plus développé sur l'île de Montréal. Bien que la Politique de Mobilité Durable 2030 du Québec prévoit des investissements majeurs – dont plus de 2,2 milliards de dollars pour des projets structurants en transport public [11] - la fréquentation des réseaux peine toujours à retrouver ses niveaux d'avant-pandémie, fragilisant les finances des sociétés de transport [12]. Dans ce contexte, l'ATUQ souligne l'importance d'un financement public stable et prévisible, aligné sur les objectifs de mobilité durable des municipalités.

Malgré ces contraintes, la STL affiche des objectifs clairs : atteindre 30 millions de déplacements d'ici 2031, soit une hausse de presque 50% par rapport à 2023 [8]. La STL fait ainsi face à un double défi : gagner la confiance des usagers dans un contexte de concurrence modale accrue tout en améliorant la performance de son réseau d'autobus. Plusieurs projets soutiennent cette vision, dont la création d'un service rapide par bus, l'étude d'un mode structurant dans l'axe du métro, et l'agrandissement du garage municipal pour soutenir l'électrification de la flotte [13]. La STL s'est également engagée dans une refonte de son réseau, une démarche ambitieuse qui nécessite le recours à des outils d'analyse avancés pour guider les choix de réaménagement, optimiser les correspondances au sein de son futur réseau et structurer l'offre. Dans un contexte budgétaire contraint et face aux nombreux défis auxquels la STL est confrontée, l'optimisation de l'utilisation des ressources existantes s'impose comme un levier stratégique. Elle permet d'améliorer la qualité du service et la fiabilité du réseau sans nécessiter de nouveaux investissements majeurs en infrastructures. Laval apparaît ainsi comme un terrain d'expérimentation particulièrement pertinent pour étudier les conditions d'une amélioration ciblée, réaliste et durable de la performance du transport public en milieu urbain.

C'est dans cette perspective que cette thèse propose de nouvelles méthodes pour améliorer l'utilisation des ressources existantes en transport public par l'optimisation en temps réel des opérations des sociétés de transport. Elle s'intéresse plus particulièrement au problème de la synchronisation en temps réel des correspondances dans les réseaux de transport public, avec un focus opérationnel sur la ville de Laval.

1.2 Problématique

Dans un réseau de transport public, la synchronisation des correspondances consiste à coordonner les arrivées et départs de plusieurs véhicules à un arrêt, afin de permettre aux passagers de poursuivre leur trajet avec un temps d'attente minimal. Cette coordination repose sur des tactiques de contrôle, comme l'attente aux arrêts, et vise à améliorer la qualité de service en optimisant l'utilisation des ressources disponibles. En pratique, elle peut réduire les correspondances manquées (qui sont prévues selon l'horaire planifié, mais n'ont pas pu se réaliser en raison d'aléas opérationnels), améliorer la régularité du service, et, plus globalement, accroître l'efficacité opérationnelle du réseau.

Malgré ces avantages, la synchronisation en temps réel demeure peu exploitée dans la pratique. Cette situation s'explique notamment par les défis liés à sa mise en œuvre dans des contextes opérationnels complexes et dynamiques, ainsi que par la diversité des contextes urbains. L'objectif de cette thèse est de développer des méthodes d'optimisation pour la synchronisation des correspondances en temps réel, qui soient à la fois efficaces et applicables.

Le caractère « efficace » relève plusieurs volets. Premièrement, les algorithmes proposés doivent améliorer significativement les opérations des systèmes de transport. De plus, ces algorithmes doivent être capables de fournir des résultats satisfaisants en temps réel. Le caractère « applicable » recouvre également plusieurs dimensions. Premièrement, les méthodes doivent être reproductibles, s'appuyer sur des standards de données largement utilisés et permettre une analyse transparente des résultats. Elles doivent également être réalistes, c'est-à-dire testées à partir de données empiriques et dans des conditions représentatives de l'exploitation en temps réel. Les méthodologies doivent aussi adopter une approche intégrée, en considérant de manière cohérente les différentes composantes de la mobilité : la circulation des véhicules, les flux de passagers ou encore les interactions entre lignes. Elles doivent être suffisamment flexibles pour s'adapter à la diversité des architectures de réseau et des profils de lignes. Elles doivent en outre être fiables, c'est-à-dire robustes face à l'incertitude et aux aléas courants dans l'exploitation quotidienne. Enfin, elles doivent être évolutives, capables de traiter des lignes comportant de nombreuses correspondances et passer à l'échelle pour traiter des réseaux de grande taille.

Les exigences posées par une méthodologie applicable et efficace soulèvent un ensemble de défis méthodologiques et opérationnels, liés aux réalités complexes des réseaux de transport public. Ces défis concernent à la fois la localisation des tactiques de contrôle, le choix des lignes, les effets à l'échelle du réseau, la modélisation des comportements des usagers, ainsi que la prise de décision en contexte incertain et dynamique. Ces décisions doivent s'appuyer sur des données opérationnelles disponibles en temps réel, telles que les positions *Global Positioning System* (GPS) des véhicules, les historiques d'achalandage ou les validations par carte à puce, tout en tenant compte de leur variabilité et de leur caractère partiellement observable.

L'un des premiers enjeux porte sur la sélection des lignes sur lesquelles appliquer des tactiques de synchronisation. L'impact d'une intervention dépend fortement du profil de la ligne : fréquence de passage, achalandage, rôle structurant dans le réseau. Certaines lignes plus isolées ou faiblement desservies peuvent nécessiter des interventions ciblées pour limiter les correspondances manquées. Des tactiques appliquées à des lignes fortement achalandées vont affecter plus de passagers. De plus, la structure du réseau influence directement les effets d'une décision locale. Une synchronisation réussie sur une ligne peut générer des retards sur une autre, perturber des correspondances futures, ou au contraire, fluidifier l'ensemble d'un corridor de transport. À mesure que le nombre d'arrêts ciblés augmente, la complexité combinatoire du problème s'intensifie, mais les possibilités d'amélioration se multiplient. Une modélisation réaliste et intégrée de la synchronisation doit ainsi tenir compte non seulement de la ligne ciblée, mais aussi de ses interconnexions et des passagers impactés présents et futurs. Dans ce contexte, représenter fidèlement les trajets des usagers et leurs interactions avec l'offre devient essentiel. Chaque tactique appliquée implique un arbitrage entre plusieurs objectifs parfois contradictoires : minimiser les temps de trajet, maximiser la synchronisation, préserver la régularité des véhicules et maintenir l'adhérence aux horaires planifiés. Ces arbitrages doivent être réalisés en temps réel, sous des contraintes computationnelles strictes. Ils s'appuient sur des informations partielles, tant sur les comportements des usagers — souvent modélisés de manière agrégée ou implicite faute d'observabilité directe — que sur l'état du réseau de transport public. Or, ces éléments présentent une forte variabilité et influencent l'efficacité des tactiques de contrôle proposées. La robustesse des décisions devient donc une exigence centrale, qui impose de modéliser explicitement l'incertitude et de concevoir des méthodes capables de s'adapter à des contextes dynamiques tout en maintenant une certaine stabilité et fiabilité opérationnelle.

Un dernier défi concerne la conception de méthodes capables de traiter des lignes d'autobus dans leur intégralité ainsi que des réseaux de grande taille, tout en demeurant applicables en temps réel. La méthodologie développée doit être suffisamment modulable pour s'adapter à différentes échelles d'analyse, qu'il s'agisse d'un réseau multiligne interconnecté complet ou d'un sous-réseau ciblé, identifié comme prioritaire par l'opérateur. Pour maintenir des temps de calcul compatibles avec les exigences opérationnelles, il est nécessaire de découper le problème en sous-problèmes, tout en contrôlant les interactions pour éviter les effets de bord indésirables. Cette articulation entre différents niveaux d'analyse – de la trajectoire individuelle à l'ensemble du réseau – ajoute une complexité supplémentaire à la prise de décision. L'ensemble de ces défis, présenté dans la figure 1.1, est examiné plus en détail dans le chapitre suivant.

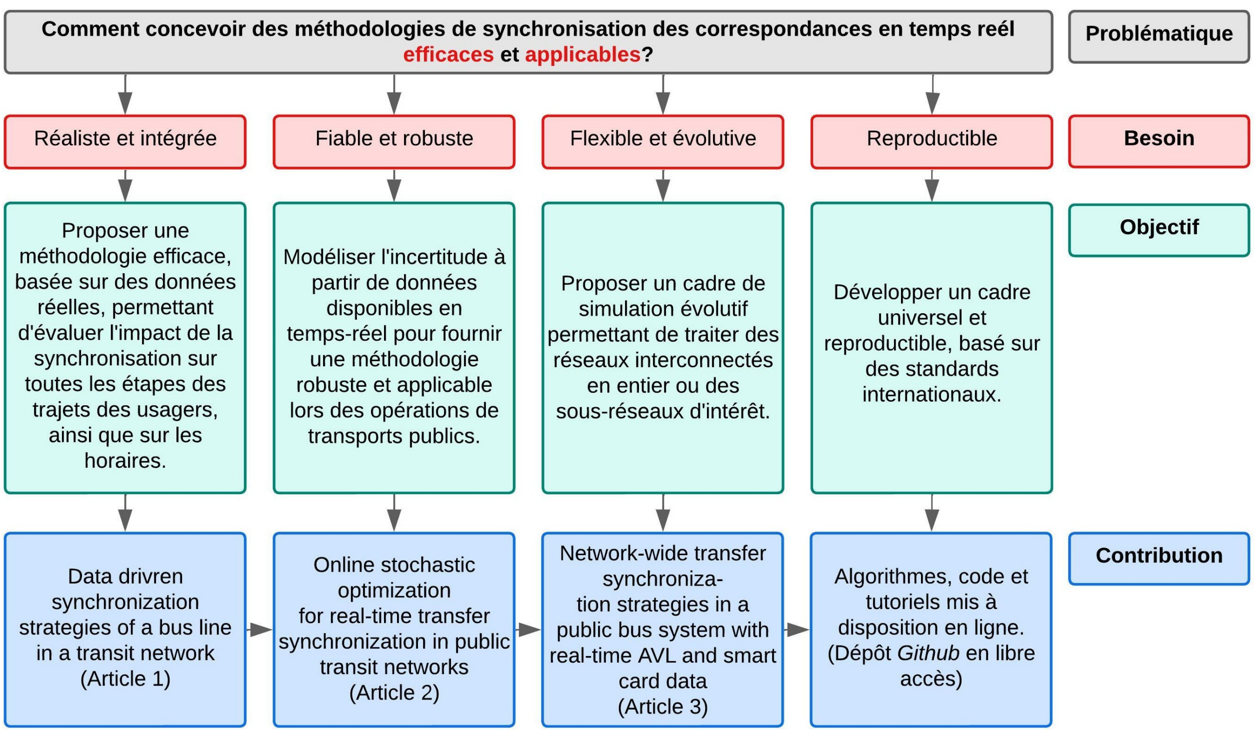

FIGURE 1.1 Problématique, besoins identifiés, objectifs et contributions de la thèse.

Cette thèse est structurée de la façon suivante. Le chapitre 2 introduit une revue de la littérature et adresse les défis majeurs à relever. Le chapitre 3 introduit les données ainsi que la méthodologie utilisées dans les articles qui suivent. Le chapitre 4 présente l'article *Data driven synchronization strategies of a bus line in a transit network* publié dans *Transportation Research Procedia*. Le chapitre 5 contient l'article *Online stochastic optimization for real-time transfer synchronization in public transit networks* soumis à *INFORMS Journal on Computing*. Le chapitre 6 contient l'article *Network-wide transfer synchronization strategies in a public bus system with real-time AVL and smart card data* soumis à *Public Transport*. Le chapitre 7 propose une discussion générale sur les contributions des trois articles précédents. Finalement, le chapitre 8 conclut la thèse.

CHAPITRE 2 REVUE DE LITTÉRATURE

La revue de littérature s'articule autour de deux axes. La section 2.1 présente différentes tactiques de contrôle utilisées pour la synchronisation des correspondances et leur application en temps réel. La section 2.2 présente comment modéliser l'incertitude dans le cadre de la synchronisation des correspondances et discute de plusieurs méthodes pour résoudre des problèmes dynamiques en temps réel.

2.1 Contrôle en temps réel pour la synchronisation des correspondances

La synchronisation des correspondances peut être réalisée soit lors de la phase de planification des horaires, soit durant la phase de contrôle, et a fait l'objet de nombreuses revues [14–16]. Ce domaine suscite un intérêt croissant avec l'émergence et l'accès accru à des données en temps réel, issues de systèmes embarqués ou d'applications utilisateur. Les auteurs de l'article [17] proposent une revue exhaustive des travaux sur la synchronisation des correspondances en temps réel. Cela inclut le contrôle aux arrêts, tel que l'attente aux arrêts ou le saut d'arrêt, ainsi que le contrôle inter-arrêts, comme le contrôle de vitesse.

2.1.1 Tactiques de contrôle

Les tactiques de contrôle aux arrêts et inter-arrêts sont nombreuses, mais cette revue de littérature se concentre sur trois tactiques principales : l'attente aux arrêts, le saut d'arrêt et le contrôle de vitesse. Ces tactiques sont souvent appliquées en combinaison dans les travaux existants.

L'attente aux arrêts consiste à retenir un véhicule à un arrêt pendant une durée prédéterminée. Elle est utilisée pour attendre l'arrivée d'un véhicule de correspondance, améliorer la régularité du service pour les véhicules en avance, ou atténuer le regroupement des véhicules (*bus bunching*). Reconnue comme l'une des tactiques les plus efficaces et simples à mettre en œuvre [14], elle implique néanmoins la résolution de problèmes d'optimisation complexes, dont le nombre de variables croît avec le produit des arrêts ciblés et des véhicules pris en compte. Ces défis expliquent pourquoi de nombreuses études se limitent à une seule ligne et à un nombre restreint d'arrêts [18]. Le saut d'arrêt est le fait de ne pas s'arrêter à un ou plusieurs arrêts consécutifs. Cette tactique réduit les temps de trajet pour les passagers à bord, mais peut pénaliser ceux souhaitant monter ou descendre aux arrêts sautés, en particulier sur des lignes à faible fréquence. Le saut d'arrêt peut permettre un comportement similaire à des lignes rapides ou express. Enfin, le contrôle de vitesse, et notamment l'accélération, est une tactique simple à mettre en œuvre en l'absence de congestion et dans le respect des limitations de vitesse. Elle permet de réduire les retards, de maintenir la régularité du service et de faciliter les correspondances. Dans la littérature, cette tactique est généralement utilisée en combinaison avec d'autres mesures.

La combinaison de plusieurs tactiques de contrôle peut renforcer leur efficacité pour la synchronisation des correspondances. Cependant, cette approche élargit considérablement l'espace décisionnel et augmente la complexité des problèmes à résoudre. Par exemple, les auteurs de l'article [19] proposent une combinaison d'attente et de saut d'arrêt pour minimiser le temps de trajet des passagers tout en augmentant le nombre de correspondances directes (situations où deux véhicules se retrouvent au même arrêt au même moment, permettant des transferts immédiats dans les deux sens). Leur modèle est testé sur un réseau à trois lignes et deux arrêts de correspondance avec des données simulées sur la demande et met en évidence les bénéfices potentiels de cette approche. Cette analyse est approfondie dans l'article [20], qui utilise les mêmes tactiques pour réduire les correspondances ratées. Dans l'article [21], les auteurs y ajoutent une tactique de demi-tour dans une méthodologie visant à maximiser les correspondances directes et à minimiser le temps total de trajet des passagers. La résolution de ces modèles s'effectue en temps exponentiel et limite leur applicabilité à des réseaux de petite taille, ou à des réseaux simplifiés.

2.1.2 Emplacement des points de contrôle

Les arrêts auxquels on peut appliquer une tactique de contrôle sont appelés points de contrôle ou arrêts de contrôle. Le choix du nombre et de l'emplacement des points de contrôle est crucial pour garantir la fluidité du service, réduire les irrégularités et optimiser l'efficacité opérationnelle. Leur sélection, notamment pour l'attente aux arrêts, fait l'objet de nombreuses études soulignant son impact sur la performance des réseaux.

Les auteurs de l'article [22] montrent que multiplier les points de contrôle contribue à régulariser les intervalles. De plus, l'étude menée dans l'article [23] indique qu'introduire un nombre suffisant d'arrêts de contrôle pour l'attente aux arrêts peut améliorer l'adhérence aux horaires sans entraîner des coûts d'attente élevés pour les passagers. De leur côté, les auteurs de l'article [24] observent que considérer tous les arrêts comme points de contrôle permet une gestion distribuée sur l'ensemble de la ligne, ce qui permet un meilleur contrôle des perturbations. Malgré les avantages associés à l'utilisation de plusieurs arrêts de contrôle, la performance des modèles de synchronisation des correspondances est plus sensible à l'emplacement des arrêts de contrôle qu'à leur nombre [25]. Cette sensibilité dépend des caractéristiques du service et de la stratégie d'attente aux arrêts employée. Une forte incertitude inhérente pourrait nécessiter davantage de points de contrôle, tandis qu'une stratégie intégrant la connaissance des positions de tous les véhicules pourrait diminuer leur importance.

L'achalandage aux arrêts est aussi un facteur à considérer lors du choix de points de contrôle. Les arrêts où les véhicules sont très chargés mais présentent peu d'embarquements et débarquements devraient être évités comme points de contrôle d'attente. En effet, ils engendrent des temps d'attente importants pour les nombreux passagers déjà à bord [26]. Enfin, les auteurs de l'article [27] concluent également que les points de contrôle devraient se situer aux arrêts avec des taux d'embarquement élevés, de préférence en début de ligne.

Bien que de nombreuses études se concentrent sur le choix des arrêts de contrôle, aucune n'évalue systématiquement l'impact de la synchronisation des correspondances en fonction du choix des lignes et de la typologie des réseaux.

2.1.3 Accès aux données

L'accès aux données en temps réel est un facteur clé pour améliorer l'efficacité des modèles de synchronisation des correspondances, car il permet d'anticiper plus finement les dynamiques des réseaux de transport et de répondre aux imprévus. Cette section retrace l'évolution des données utilisées et leur impact sur les stratégies de contrôle.

Avant le déploiement des systèmes intelligents de transport, tels que les systèmes *Automated Vehicle Location* (AVL) et *Automatic Passenger Counter* (APC), les stratégies de contrôle en temps réel reposaient sur des décisions prises par du personnel positionné stratégiquement en station [28]. Les premières approches, basées sur des informations limitées, utilisent les horaires planifiés ou les temps entre les passages de véhicules consécutifs pour appliquer des temps d'attente à des points de contrôle prédéfinis [29]. Les auteurs de l'article [18] sont les premiers à utiliser des données de localisation de véhicules précises, supposées disponibles en temps réel. Ils montrent que la connaissance de l'emplacement des véhicules permet d'améliorer la régularisation des intervalles par l'attente aux arrêts, ainsi que la synchronisation des correspondances grâce à des estimations plus fiables des heures d'arrivée.

Historiquement, les données sur les flux de passagers, telles que la charge des véhicules, les arrivées aux arrêts et la demande en correspondances, étaient estimées à partir d'observations empiriques ou même à partir de matrices origine-destination. Ces informations sont désormais accessibles plus rapidement, et même en temps réel, grâce aux systèmes de comptage automatique des passagers (APC), aux validations de cartes à puce ou aux systèmes de tarification anticipée. Lorsque les destinations des passagers ne sont pas directement enregistrées via les validations, elles peuvent être inférées à l'aide d'algorithmes sophistiqués [30,31]. Les auteurs de l'article [32] explorent l'impact de la disponibilité de différents types d'informations, notamment les données en temps réel sur la demande en passagers et les horaires d'arrivée des véhicules. Ils concluent que les données sur la demande en temps réel sont particulièrement bénéfiques pour la synchronisation des correspondances dans des réseaux à faible fréquence. Des travaux récents renforcent cette tendance en utilisant des données passagers réelles plutôt que des moyennes historiques ou des simulations, bien que les études de cas restent de tailles limitées. Par exemple, l'article [33] évalue trois niveaux d'accès à l'information pour optimiser la synchronisation des correspondances et la régularité du service : absence de données en temps réel (demande modélisée par une loi de Poisson), taux d'occupation des véhicules en temps réel, et intégration des validations de cartes à puce avec les taux d'occupation en temps réel. Les tests, effectués avec le modèle de simulation de transports publics BusMezzo, portent sur deux lignes, un arrêt de correspondance et cinq arrêts de contrôle dans le réseau de La Haye, aux Pays-Bas. Les résultats indiquent que ces stratégies sont particulièrement efficaces dans des contextes de faible affluence et que l'utilisation des données issues des cartes à puce améliore significativement la synchronisation des correspondances.

De nombreuses études supposent que les données sur les véhicules et/ou sur la demande sont entièrement disponibles et précises. Par exemple, les auteurs de l'article [34] proposent une stratégie combinant attente aux arrêts et contrôle de vitesse pour la synchronisation des correspondances. Leur modèle d'optimisation, appliqué à un réseau de trois lignes d'autobus et cinq points de correspondance, s'appuie sur des données en temps réel concernant la vitesse et la position des véhicules, tout en supposant la demande en passagers et les correspondances connues. De manière similaire, les auteurs de l'article [35] développent un modèle à base de règles intégrant les mêmes tactiques pour maximiser les correspondances directes et réduire le regroupement des autobus en temps réel. Bien que leur modèle soit résolu en temps polynomial, il repose sur des hypothèses strictes concernant la connaissance des temps de trajet, des arrivées de passagers et de la demande en correspondances.

La mise en œuvre de stratégies de synchronisation efficaces repose sur l'intégration simultanée de données concernant la demande en passagers et les positions des véhicules. Ces informations permettent d'évaluer avec précision l'impact des tactiques de contrôle sur l'ensemble des passagers, qu'ils soient en correspondance ou non, tout en préservant la régularité du service. En combinant ces données, il devient possible de concevoir des solutions adaptées aux besoins des passagers tout en optimisant l'exploitation des véhicules. Pour garantir des stratégies de synchronisation réalistes et applicables, il est essentiel de s'appuyer sur des données réelles reflétant la complexité des réseaux de transport urbains. Toutefois, les stratégies déployées en temps réel doivent se baser exclusivement sur des données disponibles en temps réel, souvent partielles et imprécises, ce qui limite la compréhension complète de l'état du réseau. Ainsi, les modèles doivent intégrer l'incertitude inhérente aux opérations afin de produire des solutions robustes et opérationnelles.

2.2 Modélisation de l'incertitude

Cette section examine les principales sources d'incertitude dans le cadre de la synchronisation des correspondances en temps réel et explore les approches permettant de les modéliser et de les intégrer dans des problèmes dynamiques.

2.2.1 Optimisation dynamique et optimisation en temps réel

Dans la synchronisation des correspondances, l'état du réseau de transport évolue en permanence. Les temps de trajet et les charges à bord sont mis à jour de manière périodique, ce qui exige une réoptimisation dynamique des décisions de contrôle en fonction des informations disponibles. Cette dynamique complexifie la résolution du problème en élargissant l'espace des décisions possibles. Par ailleurs, ces décisions doivent être prises dans des délais courts. Par exemple, déterminer un temps d'attente optimal au prochain arrêt impose de finaliser les calculs avant l'arrivée du véhicule à l'arrêt. La synchronisation des correspondances en temps réel combine ainsi optimisation dynamique, qui intègre l'évolution progressive du système, et optimisation en temps réel, qui impose des temps de calcul réduits malgré la complexité du problème.

Les auteurs de l'article [36] développent un modèle de programmation dynamique combinant les tactiques d'attente aux arrêts et de saut d'arrêt pour minimiser le temps total de trajet des passagers, incluant le temps d'attente pendant les correspondances. Ils introduisent un modèle de simulation pour estimer la probabilité de rencontre entre deux bus partageant plusieurs arrêts communs. La borne supérieure de cette probabilité est ensuite intégrée dans le modèle de programmation dynamique. Le modèle est testé sur un réseau artificiel avec deux lignes et trois arrêts. Cette approche est améliorée dans [37] avec un modèle dynamique en temps polynomial testé sur un réseau artificiel composé d'une ligne principale, trois lignes de correspondances et trois segments de correspondances. Les données sur les temps de parcours et la demande en passagers sont simulées. L'article [38] prolonge cette méthodologie en y ajoutant la tactique de demi-tour, qui consiste à faire retourner un véhicule avant son terminus pour renforcer l'offre dans le sens inverse en cas de déséquilibre de la demande. Cependant, dans [19], les auteurs soulignent que l'approche par programmation dynamique utilisée par [38] est plus difficile à mettre en oeuvre que leur modèle statique résolu à l'aide d'un solveur commercial.

D'autre part, les auteurs de l'article [39] proposent un modèle en temps réel à base de règles pour synchroniser les correspondances dans un réseau multimodal, validé par des simulations impliquant quatre véhicules et environ quarante passagers effectuant une correspondance. Le modèle calcule la durée d'attente maximale admissible pour un véhicule contrôlé à un arrêt de correspondance. Si un véhicule de correspondance est attendu pendant cet intervalle, un temps d'attente est appliqué. Cette approche prend en compte l'incertitude des heures d'arrivée des véhicules et minimise le temps total de trajet des passagers, en considérant le temps d'attente et le temps à bord.

L'optimisation dynamique et en temps réel demeure un défi majeur dans le cadre de la synchronisation des correspondances. Elle requiert le développement de méthodologies alliant rapidité et précision, tout en étant adaptées à des systèmes de transport de grande taille.

2.2.2 Sources d'incertitude

La synchronisation des correspondances dans les réseaux de transport public est soumise à de multiples sources d'incertitude qui affectent tant la planification que les opérations. Les temps de trajet des véhicules fluctuent sous l'effet de la congestion et des conditions météorologiques, tandis que les temps d'arrêt varient selon les flux d'embarquements et de débarquements des passagers. L'incertitude sur les horaires des véhicules impacte directement la faisabilité des correspondances et la répartition des flux de passagers. La demande en passagers dépend de facteurs spatio-temporels tels que l'heure, le jour et la localisation (par exemple, centre-ville ou banlieue). Les comportements des passagers ajoutent une complexité supplémentaire : leurs choix d'itinéraires ne suivent pas toujours une logique rationnelle et sont influencés par des préférences individuelles, ou par une information imparfaite. Ces décisions ne peuvent être reproduites de manière fiable qu'à partir de données historiques. La plupart des modèles dans la littérature adoptent une vision simplifiée de la demande, la traitant comme déterministe [34, 35] ou suivant des distributions à taux constant [40]. Or, ces approches ne reflètent ni la variabilité ni la complexité réelles des réseaux, qui peuvent présenter de fortes fluctuations.

Ces incertitudes impactent directement les décisions de contrôle, mais ne sont révélées qu'après leur mise en œuvre. Une prise en compte rigoureuse de l'incertitude est donc essentielle pour garantir des solutions robustes et performantes. Toutefois, la littérature n'intègre jamais simultanément l'ensemble de ces sources d'incertitude.

2.2.3 Intégration de prévisions

Lorsqu'une décision sur une tactique de contrôle est prise, les données sur les états futurs d'un réseau de transport sont inaccessibles. Toutefois, l'état actuel du système fournit des informations partielles donnant des indices sur les évolutions possibles du réseau. Les décisions doivent donc être ajustées dynamiquement en intégrant à la fois l'état actuel et les évolutions possibles anticipées du système. Dans ce contexte, la modélisation de l'incertitude devient un élément clé pour obtenir des résultats pratiques et pertinents [41]. Elle repose généralement sur des prévisions de certaines variables décrivant le système ou sur la génération de scénarios simulant ses états futurs.

Les auteurs de l'article [42] démontrent que l'intégration de prévisions des horaires d'arrivée des véhicules pour déterminer les temps d'attente aux arrêts réduit significativement les temps de trajet des passagers, comparée à une stratégie sans prévisions. Toutefois, comme le soulignent [43], l'efficacité des méthodes prédictives dépend fortement de la précision des prévisions utilisées. Dans un autre contexte, les auteurs de l'article [44] utilisent des données historiques sur les paires origine-destination et des données issues de systèmes *Automated Fare Collection* (AFC) (embarquement et débarquement) supposées disponibles en temps réel pour effectuer des prévisions sur la demande. Ils synchronisent les correspondances d'un seul arrêt entre une ligne principale et une ligne secondaire. Les auteurs concluent que des prévisions dynamiques sur la demande sont nécessaires dans les réseaux avec de nombreuses correspondances et présentant une forte variabilité de la demande de correspondances. Enfin, les auteurs de l'article [45] proposent un modèle dynamique à base de règles intégrant trois tactiques de contrôle : l'attente aux arrêts, le saut d'arrêt et le contrôle de vitesse. Une procédure de simulation basée sur la méthode de Monte Carlo est utilisée pour évaluer la performance du système face aux fluctuations. Ils modélisent les temps de trajet avec des variables aléatoires ayant des moyennes et des variances dérivées des données historiques. Ce modèle est appliqué à un réseau de deux lignes d'autobus, deux arrêts de correspondance et huit correspondances de passagers. Les simulations montrent que ce modèle peut réduire le temps total de déplacement des passagers, bien qu'il puisse parfois accroître l'irrégularité du service au profit d'une meilleure qualité des correspondances.

Plus récemment, l'utilisation de scénarios englobant l'ensemble des variables décrivant les états futurs du réseau est employée pour la synchronisation des correspondances. Les auteurs de l'article [40] proposent une simulation à événements discrets avec un horizon glissant. Chaque réoptimisation repose sur un scénario unique représentant un état prévisionnel du réseau dans l'horizon, où les temps de trajet sont générés à partir de moyennes historiques et les arrivées des passagers sont modélisées par une loi de Poisson. Leur modèle, combinant les tactiques d'attente aux arrêts et d'accélération, minimise le temps total de déplacement des passagers. Validée à l'aide du simulateur BusMezzo, cette approche est testée sur un réseau de trois lignes et trois arrêts de correspondance du système Métronit à Haïfa, en Israël. L'étude utilise un scénario unique sur l'état du système, ce qui ne permet pas de prendre en compte la variabilité de l'ensemble de ses éléments.

Bien que des outils tels que BusMezzo permettent une simulation détaillée, la synchronisation des correspondances en temps réel nécessite des solutions capables d'intégrer des réoptimisations réseau fréquentes, multi-lignes et avec des tactiques combinées, tout en maintenant une efficacité computationnelle élevée.

2.2.4 Optimisation Stochastique En Ligne (OSEL)

L'OSEL est une discipline essentielle pour la prise de décision dans des environnements dynamiques sous incertitude. Elle vise à élaborer des stratégies robustes, capables de s'adapter aux variations du système.

Les algorithmes d'OSEL basés sur la génération de scénarios multiples, ou l'échantillonnage, comportent trois composantes principales : un générateur de scénarios, un solveur hors ligne qui calcule une solution pour chaque scénario, et un processus décisionnel qui intègre les informations provenant de toutes les solutions [46,47]. Les générateurs de scénarios présentent une grande flexibilité vis-à-vis des hypothèses sur les distributions des variables générées, mais un nombre élevé de scénarios est nécessaire pour représenter fidèlement la variabilité du système.

Les algorithmes d'OSEL les plus couramment utilisés pour générer plusieurs scénarios sont *EXPECTATION* (E) [48], C et R. Ces algorithmes sont adaptés à diverses applications [49], mais n'ont pas encore été appliqués au problème de synchronisation des correspondances. L'algorithme E, appliqué au problème de tournées de véhicules, évalue le coût de visiter chaque client en premier en imposant cette décision dans tous les scénarios, puis optimise complètement pour les clients restants. Bien que l'algorithme E fournisse une évaluation exhaustive, sa complexité computationnelle limite son utilisation en temps réel. L'algorithme C [50, 51] sélectionne la décision qui est optimale pour le plus grand nombre de scénarios. Enfin, l'algorithme R [52] offre une approximation de E en évitant une réoptimisation complète pour tous les scénarios. Bien que ces algorithmes puissent partager le même générateur de scénarios et le même solveur hors ligne, ils se distinguent par leurs processus décisionnels. Dans la littérature sur la synchronisation des correspondances, même si de nombreuses études intègrent des prévisions sur certaines variables du système, aucune n'évalue différents processus décisionnels dans le cadre d'OSEL.

D'autres études améliorent l'efficacité computationnelle en générant moins de scénarios, mais plus pertinents, à partir de données historiques [53–55]. Par ailleurs, des avancées dans la génération de scénarios sont réalisées grâce à des techniques d'apprentissage automatique (*Machine Learning*) et à l'échantillonnage à partir de données historiques [56].

L'utilisation d'algorithmes d'OSEL basés sur l'échantillonage permet de résoudre des problèmes dynamiques rapidement tout en modélisant l'incertitude. Ces méthodes présentent un fort potentiel pour la synchronisation des correspondances en temps réel. Dans cette recherche, nous concentrons notre attention sur les algorithmes C et R, en raison de leur capacité à intégrer l'incertitude via des scénarios multiples tout en respectant les contraintes computationnelles du temps réel. Leur structure décisionnelle permet une sélection rapide de tactiques sans nécessiter de réoptimisation complète de tous les scénarios à chaque événement.

2.3 Lacunes et défis

Les travaux sur la synchronisation des correspondances en temps réel se concentrent majoritairement sur des études de cas de petite taille, en raison des contraintes computationnelles. Cette limitation réduit la portée pratique des résultats et limite leur applicabilité à des réseaux de grande envergure. Par ailleurs, l'efficacité des stratégies de synchronisation n'a pas été systématiquement évaluée selon les différentes typologies de réseaux de transport public. Il devient donc nécessaire de concevoir des approches capables d'optimiser la coordination à l'échelle de lignes complètes, voire de réseaux entiers, et de tester leur robustesse dans des contextes variés.

De plus, les objectifs considérés dans la littérature demeurent souvent locaux, se concentrant principalement sur la réduction du nombre de correspondances manquées, sans prise en compte du temps généralisé de déplacement à l'échelle du système. Une évaluation plus globale, tenant compte de l'impact sur l'ensemble des usagers et non uniquement sur les passagers en correspondance, apparaît essentielle pour refléter la performance réelle des stratégies proposées.

Malgré l'abondance croissante de données en temps réel issues de systèmes embarqués, leur intégration effective dans les processus décisionnels reste limitée. Peu d'approches exploitent ces données pour anticiper l'évolution du réseau ou modéliser dynamiquement le comportement des passagers. En particulier, l'utilisation en temps réel des données issues des systèmes de validation de cartes à puce, pourtant riches en informations sur les flux de passagers, demeure marginale.

Enfin, l'incertitude liée à la demande et aux temps de trajet est rarement modélisée de manière conjointe. Ces composantes sont souvent traitées séparément à partir de données historiques, ce qui limite la robustesse des décisions en temps réel. Deux défis majeurs subsistent : intégrer explicitement la stochasticité de l'ensemble des composantes du système de transport public mentionnées ci-dessus, et développer des méthodes efficaces capables d'exploiter conjointement les données historiques et en temps réel pour améliorer la réactivité et la robustesse des décisions sur les tactiques de contrôle.

Ces constats soulignent la nécessité de développer des approches stochastiques adaptées au temps réel, capables d'intégrer la variabilité opérationnelle tout en maintenant une faible complexité computationnelle. C'est dans cette perspective que s'inscrit la méthodologie proposée dans ce travail.

CHAPITRE 3 DONNÉES ET MÉTHODOLOGIE

Cette thèse se concentre sur la synchronisation en temps réel des correspondances dans les réseaux de transport public. L'objectif principal est de développer des méthodes efficaces capables de résoudre rapidement des problèmes à grande échelle tout en exploitant les données de mobilité en temps réel et en intégrant l'incertitude inhérente aux opérations des réseaux. Les approches proposées reposent sur des techniques identifiées dans la revue de littérature pour produire des solutions réalistes, robustes et applicables à des instances de grande taille. À ce jour, aucune des méthodes recensées dans le chapitre 2 ne répond de manière satisfaisante à ces défis.

Pour répondre à ces enjeux, ce chapitre décrit les données utilisées, les outils d'intelligence artificielle mobilisés pour accompagner certaines étapes du travail, ainsi que la méthodologie développée.

3.1 Données

La méthodologie développée dans cette thèse s'appuie sur un ensemble de données historiques issues du réseau de transport de la STL. Le réseau de transport public de la STL dessert la ville de Laval, au Québec, avec une population de presque 450 000 habitants et une superficie de 267 km^2. Le réseau comporte 45 lignes d'autobus interconnectées, dont plusieurs convergent vers des pôles d'échange multimodaux permettant les correspondances avec le métro de Montréal et d'autres lignes de bus structurantes. Ces lignes couvrent près de 1 500 km de réseau desservant 2 779 arrêts d'autobus. Le service quotidien dure 23 heures en semaine et en fin de semaine. Les autobus de la STL parcourent plus de 14,2 millions de kilomètres tous les ans avec près de 550 000 heures de service annuelles.

Les données partagées par la STL, couvrant l'ensemble du mois de novembre 2019, décrivent de manière détaillée les différentes composantes du système : les passagers, les véhicules, les trajets, les arrêts, et les correspondances possibles. La STL dispose de données riches, alimentées en continu, intégrant des traitements avancés tels que l'inférence automatique des origines et destinations, l'estimation de la charge à bord, ou encore la détection des correspondances effectives. Les données sur les passagers proviennent des systèmes de cartes à puce et ont été enrichies à l'aide d'un algorithme d'inférence des origines et destinations appliqué en amont, avant la transmission des données dans le cadre de cette thèse. Chaque événement de montée ou descente est horodaté avec précision, ce qui permet de reconstituer les trajets individuels et d'identifier avec fiabilité les correspondances effectuées. Les données des véhicules incluent les positions GPS haute fréquence et la charge à bord, utilisée pour valider les inférences origine-destination. Les données de réseau suivent le format standardisé *General Transit Feed Specification* (GTFS) et permettent de reconstruire les horaires planifiés, les trajets des lignes, les arrêts desservis et les possibilités de correspondance.

L'exploitation de ces données nécessite un important travail de prétraitement, entièrement automatisé et réutilisable pour tout réseau disposant de données GTFS et AFC. Celui-ci inclut le nettoyage des voyages invalides (par exemple : directions mal renseignées, temps de trajet incohérents), la correction des absences d'arrêts dans les enregistrements GPS, ainsi que la reconstruction de trajectoires complètes lorsque certaines données sont manquantes. Par ailleurs, les données de cartes à puce ont été croisées avec les données GTFS et GPS afin de retracer avec précision les trajets des usagers dans le réseau. En effet, cette intégration permet d'identifier précisément les heures d'embarquement et de débarquement des usagers grâce aux horodatages issus des traces GPS. Cette analyse exploratoire a permis d'identifier les *hubs* les plus fréquentés, les flux de passagers, les retards causés par des correspondances manquées, ainsi que des points d'amélioration potentiels. Une attention particulière est portée à l'identification des arrêts connectés (ensemble d'arrêts entre lesquels on peut effectuer une correspondance) et des points névralgiques du réseau.

Enfin, les données historiques sont utilisées pour générer des scénarios représentatifs des conditions d'opérations réelles. Le mois de données est divisé en trois ensembles : 80% pour l'entraînement (prétraitement et clustering), 10% pour les tests de calibration (recherche des meilleurs paramètres), et 10% pour la validation finale des algorithmes. Des méthodes de *clustering*, détaillées dans le chapitre suivant, sont utilisées pour regrouper les données historiques selon les heures de la journée afin de construire une base de cas pour la simulation du réseau en temps réel. Lors de la simulation, les scénarios sont générés par tirage aléatoire au sein du *cluster* associé à l'heure de l'événement simulé. Ces scénarios alimentent ensuite les outils d'évaluation et les expérimentations décrits dans les chapitres suivants.

3.2 Outils d'intelligence artificielle

Dans le cadre de la rédaction de cette thèse, j'ai eu recours de manière ciblée à des outils d'intelligence artificielle pour m'accompagner dans certaines étapes du travail. J'ai utilisé *ChatGPT* pour la relecture de textes que j'avais rédigés, notamment pour la détection de coquilles, ainsi que pour la correction grammaticale et orthographique. L'outil m'a également aidée à reformuler certains passages afin d'en améliorer la clarté, la fluidité et le style rédactionnel, tant en français qu'en anglais. Sur le plan technique, j'ai utilisé *ChatGPT* pour m'assister dans la compréhension de messages d'erreur, le débogage de code, l'amélioration de fonctions en Python ainsi que dans l'utilisation de divers outils informatiques essentiels à mon projet, notamment *Slurm, GitHub, Visual Studio Code*, et *Latex*. J'ai eu ponctuellement recours à d'autres outils, tels que *DeepL Write* pour des suggestions de reformulation et *Antidote* pour la correction linguistique. L'ensemble de ces usages est demeuré complémentaire à mon travail intellectuel, dont j'assume pleinement la responsabilité.

3.3 Méthodologie

Le chapitre 4 propose une nouvelle modélisation du problème de synchronisation des correspondances en temps réel intégrant des tactiques de contrôle telles que l'attente aux arrêts, le saut d'arrêts et la modification de vitesse. Cette approche repose sur un modèle de flot hors ligne basé sur un graphe temporel à flots entiers, capable d'énumérer exhaustivement l'ensemble des tactiques possibles dans un horizon donné. Le modèle applique les tactiques de contrôle à un véhicule d'une ligne principale, tout en tenant compte de leurs effets sur les véhicules de correspondance et les déplacements des passagers concernés. L'objectif est de minimiser le temps total de déplacement des passagers en optimisant les correspondances, tout en réduisant les écarts par rapport aux horaires planifiés. Pour évaluer la performance de cette approche dans un environnement réaliste, une version dynamique du modèle est développée et testée sous différents niveaux d'incertitude, calibrés à partir de données empiriques. Un scénario unique est généré à chaque réoptimisation. Un cadre de simulation basé sur des données réelles issues des systèmes de cartes à puce et des positions GPS de la ligne 70 du réseau de la STL est mis en place pour tester ces méthodes. Les résultats confirment l'efficacité du modèle hors ligne, qui produit des solutions optimales de haute qualité avec des temps de calcul réduits. Le modèle dynamique améliore significativement la performance par rapport aux opérations observées dans les données réelles, mais conserve un écart de performance par rapport au modèle hors ligne à information parfaite. Ce travail est présenté dans l'article intitulé *Data Driven Synchronization Strategies of a Bus Line in a Transit Network*, publié dans *Transportation Research Procedia*.

Le chapitre 5 présente deux algorithmes d'OSEL, nommés C et R, pour résoudre le problème de synchronisation des correspondances en temps réel. Ces algorithmes s'appuient sur le modèle de flot hors ligne développé dans le chapitre 4, amélioré afin de réduire davantage le nombre de variables dans les graphes tout en intégrant l'ensemble des tactiques de contrôle possibles. Ils tiennent compte de la nature stochastique des opérations de réseaux de transport public en exploitant des données de mobilité, à la fois historiques et en temps réel, afin de générer des scénarios prédictifs des conditions futures du réseau. Évalués dans un environnement dynamique sur 29 lignes du réseau de la STL, ces algorithmes démontrent leur efficacité par rapport au modèle déterministe. Les résultats révèlent des améliorations significatives du taux de correspondances réussies et des temps de déplacement des passagers. Cette étude illustre le potentiel pratique de l'OSEL pour renforcer la synchronisation des correspondances en temps réel, offrant une solution robuste pour accroître la fiabilité et l'efficacité des systèmes de transport urbain. Ces algorithmes et résultats sont présentés dans l'article *Online Stochastic Optimization for Real-Time Transfer Synchronization in Public Transportation Networks*, soumis à *INFORMS Journal on Computing*. Il a reçu le prix du meilleur article étudiant décerné par la Société Canadienne de Recherche Opérationnelle (SCRO) ainsi que le prix Gilbert Laporte du meilleur article étudiant du groupe d'intérêt en transport et logistique de la SCRO, en 2024.

Le chapitre 6 étend les méthodologies précédentes pour permettre l'application simultanée de tactiques de contrôle à plusieurs lignes de transport interconnectées. Le modèle de flot et les algorithmes d'OSEL développés dans les chapitres précédents sont intégrés dans un simulateur à l'échelle du réseau de transport public de la ville de Laval. En utilisant des données historiques des véhicules et des cartes à puce, ce simulateur reproduit les conditions stochastiques en temps réel du réseau. L'approche décompose le problème en plusieurs sous-problèmes locaux limités à un horizon spatial centré autour du véhicule réoptimisé. Ce découpage permet de contourner les temps de calcul prohibitifs, tout en rendant possible la synchronisation de 29 lignes du réseau de la STL, ce qui dépasse de loin les applications existantes dans la littérature. Les résultats mettent en lumière des améliorations significatives des temps de déplacement et des correspondances des passagers, démontrant la scalabilité et la pertinence de la synchronisation en temps réel pour les réseaux urbains multilignes. De plus, cette méthodologie est testée sur plusieurs typologies différentes de sous-réseaux au sein de la STL, pour établir un diagnostic de l'efficacité des méthodes proposées selon différents contextes d'exploitation. Cet article offre ainsi des lignes directrices claires sur les conditions d'application des stratégies de synchronisation proposées. Ces résultats sont présentés dans l'article intitulé *Network-wide transfer synchronization strategies in a public bus system with real-time AVL and smart card data* soumis à *Public Transport*.

L'ensemble des résultats obtenus, ainsi que leur application potentielle au niveau opérationnel, sont discutés dans le chapitre 7, avant une conclusion générale dans le chapitre 8.

CHAPITRE 4 ARTICLE 1 : DATA DRIVEN SYNCHRONIZATION STRATEGIES OF A BUS LINE IN A TRANSIT NETWORK

L. Kolcheva, A. Legrain et M. Trépanier ont écrit cet article et l'ont publié le 9 janvier 2025 dans *Transportation Research Procedia*.

4.1 Introduction

PT networks are becoming increasingly important considering concerns about climate change. PT companies and services develop inter-route, inter- and intra-modal transfers in order to achieve a better connectivity of the network and more flexible route planning. An efficient PT system can retain existing customers and attract more people to leave their cars behind and opt for public transportation.

Research shows that users are reluctant to engage in multi-segment trips if transfer times are uncertain [57]. The level of satisfaction of users is highly dependent on waiting times, since their personal evaluation of waiting time is higher compared to other parts of a trip (e.g. : access time, in-vehicle time [58]). It is therefore increasingly important to synchronize transfers. There are multiple stages of planning and operating a PT network (e.g. : network design, timetabling, vehicle scheduling, operation, etc.) as defined in [59]. This research will concentrate on the control stage of operating a PT network. The context in which a planned and synchronized bus schedule takes place is stochastic and dynamic. Even an optimal timetable can be subject to unpredictable congestion or route incidents. This is why real-time control strategies are needed to mitigate the undesirable effects of uncertain events.

This research is based on one month of real passenger- and bus-related data provided by the STL. It uses data from AVL, APC and AFC and integrate it in real-time control tactics. Using this historical data, relevant time-dependent travel times, passenger demand and transfer demand are generated to input into a real-time control arc-flow model. First, the model is tested with existing non-stochastic data. Then different levels of uncertainty are introduced in a simulation framework to test the reliability of the model and to examine the relative value of the available data.

4.1.1 Contributions

To the author's knowledge, no research on real-time control strategies has been implemented using an arc-flow model. In our model, a node represents a bus arrival or a passenger arrival at a stop and at a certain time. All tactics are dependent on discrete events such as passenger arrivals, bus arrivals and transfer arrivals. A tactic cannot be implemented if no such event occurs. This limits the number of executable tactics at each stop to only a few possibilities, as opposed to tactics represented by integer variables in the literature, or rule-based models. This allows for timely resolution and thus a possible execution of proposed tactics in real-time. Moreover, different combinations of control tactics are tested with different levels of uncertainty in a rolling-horizon simulation framework. Finally, we test our methodology on a large real data set from a dense urban transit network.

4.2 Literature Review

Control strategies can be divided into three categories : stop control, inter-stop control, and others [60]. Stop control strategies include tactics implemented at stops, for example holding for a certain amount of time at a stop or skipping stops. Inter-stop strategies include speed changes or traffic lights control. The last category includes tactics such as adding or removing vehicles.

4.2.1 The holding problem

The holding tactic is the easiest to implement and the literature review shows that holding, by itself, is the most effective tactic to save time [14]. Holding can be used to either maintain a certain headway between buses [22, 23, 61, 62], or to minimize waiting times [18, 63, 64].

[18] define the holding problem as a deterministic quadratic problem and proposes an algorithm to solve it. The research proposes a rolling horizon for the holding problem. [26] formulates the holding problem as a convex quadratic program at a number of control points. [63] define a control model based on a compromise between passengers on board and passengers further along the line, using stochastic arrivals at stops. The holding problem is also considered with bus capacity constraints using real-time bus locations in a heuristic algorithm [65]. [66] implement two strategies (holding and limiting the number of passengers to be able to get on a bus at certain stops) in a deterministic optimization model considering all waiting times of passengers.

4.2.2 Real-time control

[17] present an extensive literature review of public transport transfer synchronisation at the real-time control phase. [42] prove the impact of real-time control in intelligent systems. In fact, having real-time information on bus locations, passenger demands, arrival times

and passenger origin/destination pairs allows for timed transfers. Later, [32] show that this information was especially useful in networks with many connecting buses.

[67] propose a non-linear integer programming problem using two different stop-skipping tactics for real-time control and test each one's performance with different route scenarios. Real-time information on bus locations is used to predict next-stop departure times in [68], with a genetic algorithm to optimize holding times. Many studies concentrate on synchronizing transfers using real-time control. [36] develop a new definition of synchronized transfers and apply it to multiple consecutive transfers stops (transfer segments) rather than to single transfer points. This approach was improved in later articles using dynamic programming [38]. [69] introduce a dynamic transit-simulation model using real-time data on congestion, passenger demand and bus activity. The article uses the mean headway from the preceding and succeeding buses as a basis for holding tactics. In recent years, the combination of different real-time control strategies has been studied. [57] and [20] propose a combination of holding and skip-stop/skip-segment in order to minimize total passenger travel time by increasing the number of direct transfers. Travel times, passenger demand and transfers are assumed to be known and are deterministic. Moreover, passenger arrivals are independent of bus arrivals. The work is continued in [21] where short turning is implemented as a real-time control action. Travel times are fixed, as well as passenger arrival rates and transfers. [70] use a tactic-based predictive control approach under dynamic and stochastic traffic environment. Finally, [33] proposes two different transfer synchronization controllers using different real-time passenger data to show the importance of passenger data on the performance on transfer control.

4.2.3 Headway control

The literature also addresses the regulation of bus headways using holding and other real-time control tactics. [61] minimize headway variation by minimizing the average waiting time at stops, not considering in-bus delays. [34] implement an optimization model with holding and speed change tactics. Two objective functions are tested (minimizing headway gap or total passenger travel time) using transfers at single points or along shared corridors. [23] studies high frequency bus routes, shows that control is necessary to avoid bus bunching and proposes a real-time holding strategy at pre-defined control points. [62] present a novel approach by defining headways according to system states and behaviour, instead of using static headways. Holding times are calculated using the headway from previous bus. [71] propose a dynamic model to regulate headways and avoid bus bunching.

4.2.4 Dynamic models

More recently, the literature focuses on the dynamical use of real-time data. [43] compare and evaluate different holding methods used in real life and proposed in the literature. The research shows that prediction-based methods achieve the best results when considering both holding time and headway regularity. The drawback of prediction-based methods is their sensitivity to prediction accuracy. [72] propose a model that considers the dynamic nature of travel times and demand and show the importance of accurate estimations of the current state of the PT network. [40] present a simulation framework that considers multiple and entire bus lines (as opposed to single transfer stops in the previous literature) and takes into account all stages of passenger trips (in-bus time, dwell time, transfer time and extra time if a passenger is unable to board a bus) with a capacity constraint. The model defines an optimization horizon at each step considering the actual state of the PT network.

The rest of the article is organized as follows. Section 4.3, describes the case study and the underlying problem. Section 4.4 describes the mathematical formulation of the problem. Section 4.5 presents our optimization model and simulation framework, as well as our tests and results. Finally, section 4.6 presents our conclusion and possible further work.

4.3 Problem description

4.3.1 Data presentation and pre-processing

This work is based on a full month of anonymous data on the whole network of the STL (routes, buses, passengers, transport tickets, etc.). GTFS [1] files are generated for each day of the month and are then used as inputs to our model. The data is organized as follows.

— **Stops, routes and trips :** The data contains route and schedule information on all stops and routes in the STL network. A route is a PT service associated to a bus line that follows a sequence of stops. There can be multiple daily trips along the same route.
— **Dwell times and travel times :** The case study has information on dwell times as well as planned and real travel-times across the PT network. When the real dwell time at a stop is null, the stop is skipped.
— **Passenger demand :** This research uses passenger flow information coming from AFC systems on buses and in metro stations. The STL also estimates individual trip destinations using AFC data, as defined in [30]. All origin-destination (OD) pairs are generated using this data and buses are "filled" accordingly. On the other hand,

1. More information on the GTFS format here : https ://gtfs.org/

passenger information from APC systems was not used as it was not reliable.

— **Transfer demand :** Data generated by the AFC systems, allowed the reconstruction of multi-segment passenger trips. Using the information on transfer demand, this research determines synchronized transfers between buses. A synchronized transfer is when two or more buses arrive at a transfer point at the same time and allow passengers to transfer instantly. If a transfer isn't synchronized, passengers have to wait for their connecting buses. Not all transfers are synchronized.

The GTFS files described above as well as data from the GPS systems in the PT network are used as inputs for the model. The data was pre-processed to create classes of data for the modeling. These instances are used to generate graphs for the multi-commodity arc-flow model. Figure 4.1 illustrates how this data is used to generate instances for our model.

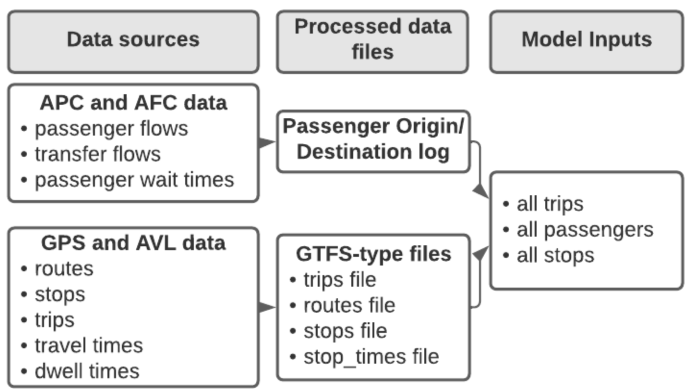

FIGURE 4.1 Data processing and generation.

4.3.2 Initial data analysis

After pre-processing the data, an initial analysis was conducted to determine where the PT network operation could be improved. Passenger flows on all bus lines and along all route segments were analyzed. We then identified corridors with the most passengers and bus stations with the biggest passenger flows. Lines, segments and hubs with the most transfers were identified and used in this research. Finally, the differences between planned and actual travel and transfer times for all passengers were analyzed. The biggest source of delay in passenger travel times was missed or late transfers.

4.3.3 Description of control tactics

Real-time control tactics can help synchronize transfers and ensure planned transfers take place. The following real-time control tactics are implemented in order to improve travel times and transfer times.

Holding : Holding makes a bus wait for a predetermined amount of time at a certain stop. Holding is used in order to reduce bus-bunching, schedule deviation and to synchronize transfers before or at transfer points. The holding tactic can therefore impact the travel time of three groups of passengers. Firstly, holding affects passengers on board the bus who add the holding time to their travel time. Secondly, passengers on board who want to transfer will spend more time in the bus, but their transfer time decreases. Finally, passengers waiting at further stops along the route add the holding time to their waiting times for the bus.

Skip-stop/Skip-segment : Skip-stop is a tactic consisting of skipping a stop along a bus route to avoid dwell times at the stop and deceleration/acceleration time before and after the stop. The skipping of multiple consecutive stops is introduced as the skip-segment tactic in [20]. Skip-stop/skip-segment is a tactic devised to gain time, but it impacts passengers differently. Passengers on board that do not wish to alight on any of the skipped stops have a shorter travel time. Passengers wishing to alight at a skipped-stop, get off at the nearest stop and walk to their destination. Passengers waiting to board the bus at a skipped-stop wait for the next bus and their waiting time increases. Passengers waiting further along the line have shorter waiting times. Finally, passengers on board wishing to transfer arrive at their transfer stops faster but have longer waiting transfer times out of the bus.

Speed control : The speed control tactic consists of making a bus drive faster or slower. In this research, only the effects of speeding up are evaluated. After discussion with the PT operator, a speedup factor of 0.8 was chosen. The speed control tactic affects both passengers on the bus and out of the bus. Passengers on board have a shorter travel time. Those wishing to transfer arrive at their transfer stops faster but have longer waiting transfer times out of the bus. Passengers waiting further along the line wait less for the bus to arrive. Transferring passengers from other lines decrease their transfer time. If the bus arrives at a stop too early before its planned arrival time, passengers arriving on time miss the bus and must wait for the next one.

All control tactics can be implemented individually or at the same time. Unlike in most previous studies where tactics were applied at a few predetermined stops, here all stops are control stops. All impacts of each tactic on passenger transit times are taken into account in our model.

4.4 Problem formulation

This research formulates an arc-flow model to represent the bus network and evaluate the impact of real-time control tactics. The formulation contains all the implemented tactics. The model seeks to minimize total passenger travel times by improving, among others, transfer times and reducing deviations from the bus schedule. The impact of the implemented tactics is evaluated using total passenger travel time, the number of successful transfers, individual passenger travel times and passenger waiting times (in and out of the bus). This research uses a single low frequency line with many feeder lines and transfer points. Tactics can only be applied to the buses of the main line. The model calculates optimal control tactics for all stops and buses in a predefined optimization horizon. The arrival times of feeder lines are not influenced by control tactics.

4.4.1 Methodology

This section describes how the graphs for the arc-flow model are built and how the proposed control tactics are incorporated into the graphs. Figure 4.2 illustrates a simple example of a graph. This example does not contain real data for confidentiality reasons. Figure 4.2a) shows a graph without any control tactics. Figure 4.2b) shows the same case graph with the holding tactic after optimization. The number of passengers along an edge is displayed on it. The time of each node is written on top of it. The exogenous flow of a node, if non-zero, is displayed above it. The horizontal distance between nodes is proportional to the time between two nodes. The vertical distance between nodes represents the distance traveled by the bus. Arrival and departure nodes of the same stop are not aligned in the figures for the sake of clarity. This mock example contains two buses, one departing a time equal to 100 and the next departing at time 300. In the no tactics case, four passengers wish to board the first bus at stop number one but arrive at the stop after the bus has departed. Moreover, two passengers coming from a feeder line wish to transfer at stop number two and board the second bus in the horizon. They arrive too late at stop number two to be able to board the second bus. In the case including the holding tactic, we can see that new possible paths for the buses were added in the graph. The optimization model then makes decisions on the paths for each bus by minimizing the total passenger travel time.

The model uses space-time dependent networks built as follows.

Nodes : A node represents a passenger arrival at a bus stop or a time for a bus to arrive at/depart from a stop. There are arrival nodes and departure nodes to distinguish when buses arrive at and leave stops. Passenger flows are represented by exogenous flows at each node.

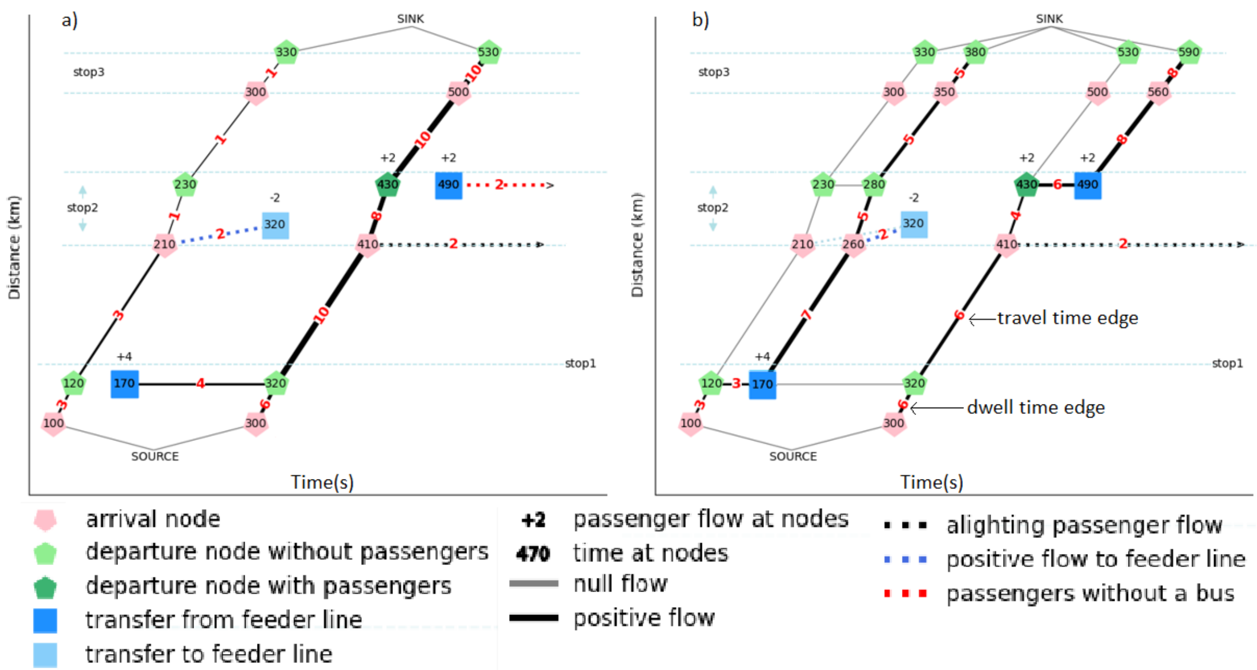

FIGURE 4.2 Example of a graph construction. a) No tactics. b) Hold tactic.

Edges : The weight of an edge represents the time difference between the origin and destination nodes of the edge (the destination node must occur after the origin node). Dwell-times are represented by edges between arrival and departure nodes of the same stop. Travel-times are represented by edges between the departure node of one stop and an arrival node of the next stop (see indications on figure 4.2). It is possible for an edge to link two nodes from different buses (e.g. : a passenger misses their bus and has to wait for the next bus). All edges are directed and have a capacity constraint related to the capacity of the vehicles.

Graphs : A graph has source and sink nodes. The source node has an exogenous bus flow equal to the number of trips. The sink node has an exogenous bus flow opposite to the number of trips. The source node can be seen as a bus depot from which all trips depart. Graphs are built in consecutive steps for each bus trip. For each stop, the travel time, the arrival time, the dwell time and the departure time are represented in the graphs. To begin, an arrival node is created at the time of arrival of the bus at the first stop. The source node and the first stop are linked with an edge of weight equal to the travel time to this stop. The departure of the bus from the stop is represented by a departure node. The arrival and departure nodes are linked with an edge with weight equal to the dwell time at the stop. For the speed control tactic, the travel time to the stop is smaller and so the bus arrives earlier at the stop. The corresponding arrival and departure nodes are added to create an additional possible path for the bus from the source. Any passenger arriving before the departure of a bus from a stop can board the bus. Nodes with positive exogenous flows representing boarding passengers are linked to the bus paths. If any transfer passengers from other lines want to board at this stop, transfer nodes with the corresponding positive exogenous flows are created at the time of arrival of the feeder lines. On the other hand, some passengers will want to alight the bus. This is represented by a negative exogenous flow. If passengers want to transfer to another line at this stop, a transfer node with negative exogenous flow is created at the time of the departure of the feeder line from this stop. If a skip-stop/skip-segment tactic is applied, no passengers can board or alight at this stop. Passengers will then have to walk to the nearest non-skipped stop. Finally, the holding tactic is represented by the time between possible departures. Edges are added between consecutive possible departure nodes. A holding tactic can be implemented only as a waiting time between two existing nodes, e.g., a bus can wait for a transfer passenger coming to the same stop. The last node for the stop is linked to the next bus to allow passengers that missed the current bus to board the next one.

The rest of the graph is constructed by iterating these steps over the remaining stops. Starting from the first stop, all possible paths to the second stop are created and so on. The graphs built in this manner incorporate all possible tactics. The model uses these graphs as inputs to determine optimal bus paths and tactics.

4.4.2 Mathematical formulation

The following assumptions are made for the offline model. We have knowledge on route information, real and planned travel times, real and planned dwell times, passenger demand, transfer demand, transfer stops, bus schedules and delays. In the online application of the model, real information is not available for all stops. The missing information is generated based on historical data and available real-time data as described in section 4.5.3. Road congestion conditions allow the implementation of the speed control tactic. Passengers are informed of the skip-stop/skip-segment tactic before the first skipped stop. Passengers waiting to board at a skipped stop wait for the next bus. Transfer passengers that miss their transfer will wait for the next possible transfer. Passengers always choose the fastest option available to them. Passenger demand does not change because of a bus delay. Finally, passengers arrive at a stop shortly before the scheduled time of arrival of the bus, since the case study is based on a low frequency line.

We use the following notations to describe the model parameters and variables.

Sets		
N		set of nodes with s the source node and t the sink node.
N^-		set of normal and transfer nodes with negative exogenous flows, not including t
A		set of arcs (u,v)
B		set of buses
S		set of stops
A_b^s	$b \in B,\ s \in S$	set of arcs for the bus b between stops s and $s+1$
AN_b^v	$b \in B,\ v \in N^-$	for node u and for the bus b, AN_b^u is the set of arcs passengers alighting at node u take to board the bus b. There is one arc per passenger.
AM_b^v	$b \in B,\ v \in N^-$	for node u and for the bus b, AM_b^u is the set of arcs taken by passengers from the previous bus if they miss their bus and have to alight at node u. There is one arc per passenger.
Parameters		
c_{uv}	$(u,v) \in A$	passenger flow capacity on arc (u,v)
w_{uv}	$(u,v) \in A$	travel time between nodes u and v.
f_v	$k \in K, v \in N$	exogenous passenger flow at node v
g_v	$k \in K, v \in N$	bus departures or bus arrivals at node v
M		bus capacity
p	$1 \leq p$	out of bus waiting times as perceived by passengers
Variables		
x_{uv}	$(u,v) \in A$	$x_{uv} \in \mathbb{N}$, passenger flow on arc (u,v)
y_{uv}	$(u,v) \in A$	binary variable for the bus flow on arc (u,v)
z_{uv}	$(u,v) \in A$	$z_{uv} \in \mathbb{N}$, indicator variable equal to x_{uv} if y_{uv}=0, and 0 otherwise
x_{uv}^+	$(u,v) \in A$	binary variable. $x_{uv}^+ = 1$ if passenger flow on arc (u,v) is positive, and 0 otherwise.

Objective

$$\min \sum_{(u,v)\in A} w_{uv} x_{u,v} + \sum_{(u,v)\in A} w_{uv}(p-1) z_{u,v} \tag{4.1}$$

Constraints

$$\sum_{(v,w)\in A} x_{vw} - \sum_{(u,v)\in A} x_{uv} \quad = \quad f_v, \forall v \in N \setminus N^-, v \neq t \tag{4.2}$$

$$\sum_{(v,w)\in A} x_{vw} - \sum_{(u,v)\in A} x_{uv} = f_v + \sum_{(u,w)\in AN_b^v} (1 - x_{uw}^+) - \sum_{(u,w)\in AM_b^v} (1 - x_{uw}^+), \quad \forall b \in B,\ v \in N^- \tag{4.3}$$

$$\sum_{(v,w)\in A} y_{vw} - \sum_{(u,v)\in A} y_{uv} = g_v, \forall v \in N \tag{4.4}$$

$$x_{uv} \leq c_{uv} y_{uv}, \forall b \in B, s \in S, (u,v) \in A_b^s \tag{4.5}$$

$$x_{uv} - c_{uv} y_{uv} \leq z_{uv}, \forall (u,v) \in A \tag{4.6}$$

$$z_{uv} \leq x_{uv}, \forall (u,v) \in A \tag{4.7}$$

$$x_{uv} - x_{uv}^+ M \leq 0, \forall (u,v) \in A \tag{4.8}$$

$$0 \leq x_{uv} - x_{uv}^+, \forall (u,v) \in A \tag{4.9}$$

$$y_{uv}, x_{uv}^+ \in \{0,1\}, \forall (u,v) \in A \tag{4.10}$$

$$0 \leq x_{uv}, z_{uv}, \forall (u,v) \in A \tag{4.11}$$

Eq. 4.1 describes the objective function of the model, minimizing total passenger travel times as perceived by passengers. This includes in-bus and out-bus travel times. The first sum in the objective function represents total passenger travel time. The second sum in the objective function represents the additional cost perceived by passengers of waiting out of bus. Constraints 4.2 ensure flow conservation for passenger flows. The incoming flow of a node, added to the number of passengers wishing to board/alight at a node must be equal to the number of passengers leaving a node. Constraints 4.3 update flows for alighting passengers. If a passenger didn't board the current bus, then they cannot alight from the current bus. Passengers that missed their bus will board and alight from the next possible bus and are added to the alighting flows for the next bus. Constraints 4.4 ensure that each bus takes a single path. The source node has a positive exogenous bus flow equal to the number of buses departing from the origin of the bus line. The sink node has a negative bus flow equal to the opposite of the source node. Constraints 4.5 ensure that passengers cannot travel between stops without being on a bus. A bus can travel empty. Constraints 4.6 and 4.7 ensure the relationship between variables x, y and z. If $y_{uv} = 1$ then $z_{uv} = 0$ and if $y_{uv} = 0$ then $z_{uv} = x_{uv}$. The variable z is used to linearize the product between the variables x and y. In reality, $z_{uv} = x_{uv}(1 - y_{uv})$. Constraints 4.8 and 4.9 ensure that if $x_{uv} > 0$ then $x^+_{uv} = 1$ and if $x_{uv} = 0$ then $x^+_{uv} = 0$. Constraints 4.10 describe that a bus can either travel on an edge or take another edge. Two buses cannot take the same path at once. Finally, constraints 4.11 indicate that passenger flows must always be non-negative.

4.5 Experiments

This section describes all the experiments made in this research and presents our results. Firstly, we describe the simulation framework and discuss the data generation for the instances used in the simulations. All possible combinations of parameters for the simulation framework are tested and the performances of the model are compared for each case. Five scenarios are used for the simulated network. The first case is the case without optimization. In the second scenario only the holding tactic is applied. The third scenario combines the holding and skip-stop/skip-segment tactics. The fourth scenario combines the holding and speed control tactics. The last scenario combines all three tactics.

All coding is in Python and the Python MIP [2] package is used to solve all instances of the optimization problem. The results presented in this research are based on the data of line 70 of the STL network, but the code works for any other line of the PT network. Line 70 is a line with 91 stops and 30 different feeder lines. Transfers occur at many different stops along line 70, but most feeder lines transfer in 5 major transfer stops along the line.

4.5.1 Simulation Framework

In this research a simulation framework is designed to test and validate the results from the deterministic optimization model. Figure 4.3 illustrates the simulation framework. The simulation evaluates a dynamic optimization algorithm. The discrete-event simulation runs a re-optimization every time a bus reaches a stop within the current optimization horizon. First, the optimization horizon is redefined. The optimization horizon consists of all the buses and stops that will be included in the current step of the simulation. It contains the next few stops on the main line of the bus trip currently being optimized. The optimization horizon

2. Package available here : https ://www.python-mip.com/

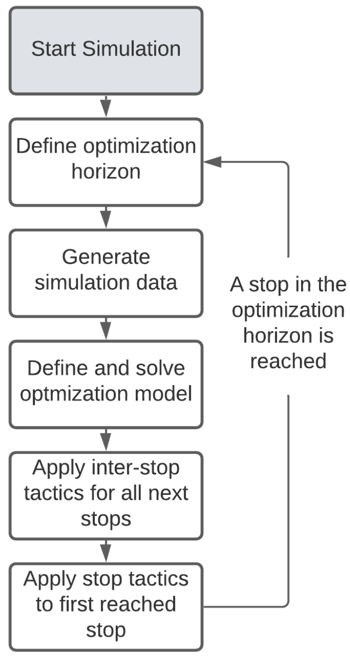

FIGURE 4.3 Simulation framework for the optimization model testing and validation

also includes the previous and next buses on the main line. Finally, the optimization horizon includes buses on feeder lines potentially transferring at the next few stops of the main line. Then data for the re-optimization is collected and generated as described in section 4.5.3. The data generation is based on historic data as well as available real-time data. We use real data for all stops that have already been visited. Passengers that missed their bus or transfer are also taken into consideration. The generation type for the data can be changed between steps of the simulation. Then a graph corresponding to the generated information is built and the associated optimization problem is solved. The solution describes control tactics for all stops in the optimization horizon. We apply the inter-stop speed control tactics for the next stop of each bus. Finally, the stop control tactics are applied to the first stop that is reached in the current optimization horizon. As soon as a bus reaches a stop in the optimization horizon, a new step is started in the simulation. All other control actions are re-evaluated in the next iteration of the simulation.

[40] apply holding and speed control to a stop that has just been reached. This method does not allow for sufficient time to inform the bus drivers of the tactics to be implemented. For this to be possible, the calculations need to be made between stops. If the calculations are made once a stop is reached, they must be intended for tactics starting from the next stop. In this research, there are on average 150 nodes and 200 arcs in the optimization in each step. The mean computation time per step of the simulation is 0.3 seconds. This short

computation time allows for a timely implementation in real-time.

4.5.2 Data sources

The data used in the case study of this research originates from the bus network of the STL, Canada, a city of 436,000 inhabitants. The network has 46 bus lines and about 1,000 stops. Data comes from the AVL systems, the APC systems and the AFC systems. The case study in our research differs from most experiments in the literature for the following reasons. Firstly, the STL has a team of data scientists working on improving the level of service and daily operations. The STL already implements different tactics and communicates with bus drivers when they are behind schedule or when they are running early. From its AVL systems, the STL also has real-time information on bus bunching and can give instructions to drivers on how to mitigate the effects of such delays on passengers. This means that the 'no optimization' scenario in our case study presents a well monitored and dynamically controlled bus network. Secondly, by using the STL smartphone application, passengers have access to real-time data about the positions and time of arrival of buses. In this network, and particularly for the case of the low frequency line 70, passengers can plan their trips before leaving using the most up-to-date information about the state of the bus network. Only trips that are likely to be successful will be offered to passengers planning multi-legged trips. Transfers that are too short and risky (less than a few minutes transfer time between buses) will not be included in the options available to users on the application. Finally, the information on passenger arrivals and transfers is exact and comes from smart card data (AFC). Hence, we do not model passenger arrivals. All stages of each passenger's trip are taken into account : waiting at stops, in bus travel time, in bus waiting time, waiting for transfers, walking time if skipped stops. Any improvements made in the model and simulations are calculated precisely for each passenger in the system (with or without transfers).

4.5.3 Data generation

The real data provided by the STL is used as a basis for the instances generated in the simulations. The month of data is divided into three working sets. The training set consists of the first twenty-four days of the month (or eighty percent of the data set). The testing set contains three weekdays of the remaining week (or ten percent of the data set) and is used to find the best parameters for the simulations. Finally, the remaining three weekdays of the data set are used for the validation set (ten percent of the data set). A second validation set is used consisting of three days of data on line number 42, a busier line of the STL bus network.

Before the simulations, the data of the training set was clustered for dwell times at each stop, travel times between each pair of consecutive stops, number of boarding/alighting passengers at each stop, number of boarding/alighting transfer passengers at each stop and finally, headways between buses. The data is clustered using K-means algorithm. The clusters are based on the time of occurrence. For example, dwell times happening around the same time of the day are in the same group. Extreme data points were removed from all clusters (e.g., dwell times significantly higher than all other dwell times in the same cluster). An example of clustering for the number of passengers boarding at a stop of line 70 is shown in figure 4.4. There are 3 clusters, depending on the time of day. The time stops at 25 hours because the service ends after midnight.

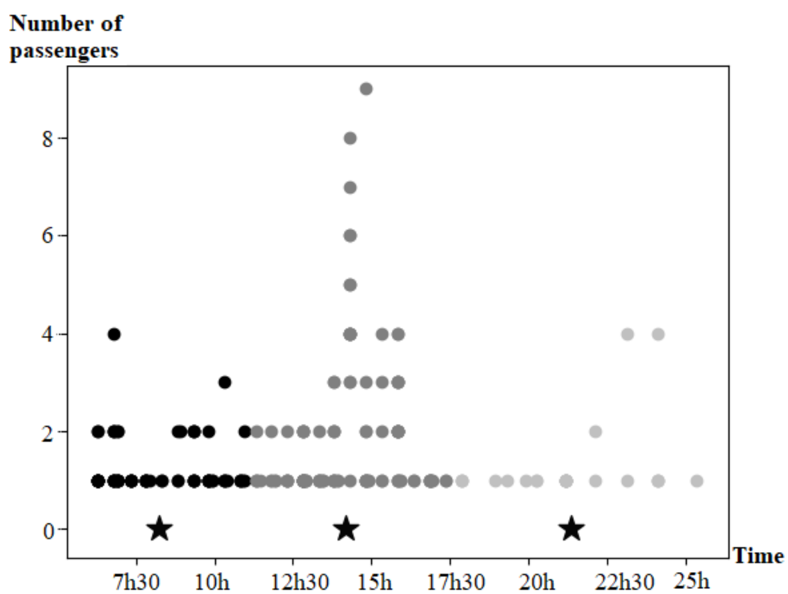

FIGURE 4.4 Clustering of the number of passengers boarding at a specific stop.

Bus trips are then generated using these clusters. First a set of consecutive bus trips is chosen from the testing or validation set and a planning horizon is defined. Then the same number of trips covering the same planning horizon will be generated iteratively to be used in the simulation. When generating bus trips, there are multiple types of data generation for each component of the trip. Type "real" returns the real data from the bus trips in the testing set. If all the components of the trip are generated using real data, the simulation gives the same results as the deterministic model (perfect information scenario). Type "mean" generates the mean of the data points of the cluster corresponding to the time of the day of the generated event. Type "sample" draws randomly from the cluster corresponding to the time of the day of the generated event. Finally, type "planned" returns the planned value of the corresponding event. Boarding and alighting passengers are generated separately and not as origin/destination pairs. Feeder line arrival times can also be generated using the current delay at the time of the simulation, the expected delay calculated at the time of the simulation, the planned arrival time or the real arrival time. Table 4.1 shows what type of data generation is possible for each of the components of the generated instances.

TABLE 4.1 Types of data generation for each component of the trip

Generated Data	**Real**	**Mean**	**Sample**	**Planned**
Dwell times	X	X	X	
Travel times	X	X	X	X
Headway times	X	X	X	X
Boarding/alighting passengers	X	X	X	
Boarding/alighting transfer passengers	X	X	X	

First, the optimal simulation parameters were determined by testing all parameters on buses of the testing set. The optimal parameters for our simulation framework are presented in Table 4.2. The best results were obtained using the real values for all data types (corresponding to the perfect information scenario). Nevertheless, the second-best parameter was chosen for the data types for which real information is not available in real-time (e.g., boarding/alighting (transfer) passengers, dwell times, travel times, arrival times of feeder lines). The generation of dwell times had little impact on the performance of the optimization model. The travel times between pairs of stops had little variation for instances in the same cluster. Planned headway times were well respected by the PT operator and were in most cases equal to the real headways. The number of alighting/boarding (transfer) passengers had the biggest impact on the decisions made in the simulations. The integer mean of the number of passengers boarding/alighting at each stop proved to give the best results. Finally, the predictions of arrival times of feeder lines using available real-time data gave results close to tests with the real values of the data. After the optimal parameters were determined, we tested all instances of the validation set with the optimal simulation parameters. Finally, the optimal parameters and the simulation were tested on another line altogether.

4.5.4 Results

Figure 4.5 shows the distributions of passenger travel times in the experiments. Buses were clustered in five groups depending on the time of day (early morning (before 7AM), morning rush hour (7AM to 10AM), midday (10AM-4PM), evening rush hour (4PM to 8PM) and end-of-service (after 8PM)). For all buses, passenger travel times are divided into waiting for the bus to arrive (with or without transfer), travelling in bus, waiting in bus due to

TABLE 4.2 Optimal types of data generation for each component of the trip

Generated Data	**Optimal Generation Type**
Dwell times	Mean
Travel times	Mean
Headway times	Planned
Boarding/alighting passengers	Mean
Boarding/alighting transfer passengers	Mean
Arrival time of feeder lines	Current Delay

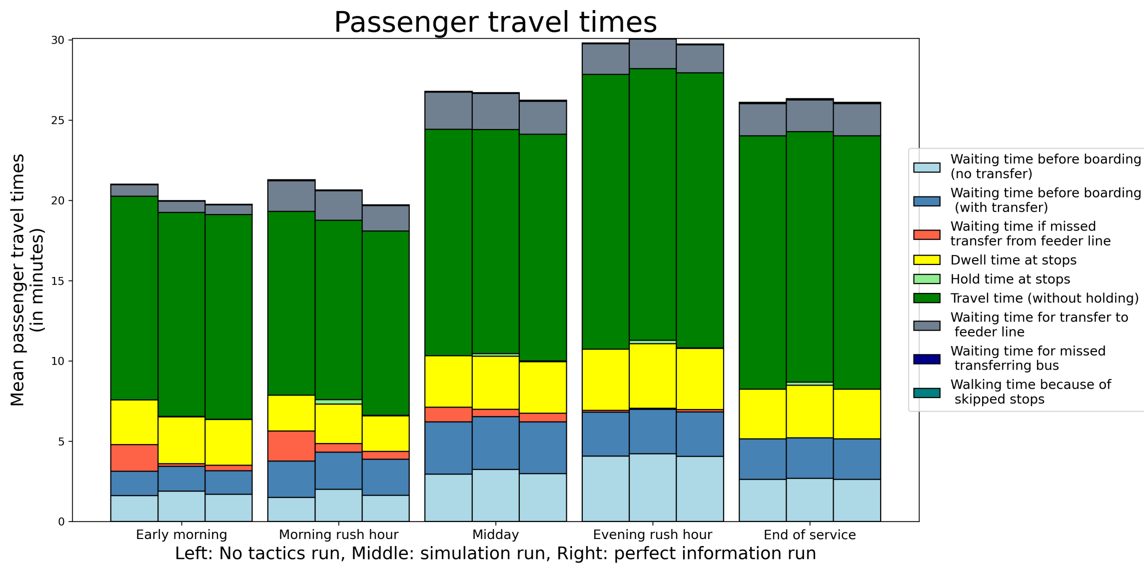

FIGURE 4.5 Passenger travel time distributions for all buses. Holding only.

holding tactics, waiting for a connecting bus of a feeder line (if there is one) and walking (in cases of skipped-stops). Figure 4.5 presents the distribution of passenger travel times when only the holding tactic is allowed. We compare passenger travel times for the no tactics case with the online simulation holding case and the perfect information, offline holding case. Both the simulation and perfect information cases of the model perform better than the no optimization case when there are some missed transfers to improve. Nevertheless, we can see that the simulations perform worse in cases where there were no missed transfers. When the model has perfect information, no tactics are needed when there are no missed transfers. In the simulation case, some predictions about the number of transferring passengers were inaccurate, inducing needless holding time. In general, the simulations activate more holding time than the perfect information offline optimal case. This is due to some inaccuracies related to the data generation in the simulation framework. Figure 4.6 presents the difference in individual travel times between the simulation case and the optimal case of the holding only scenario. We compare individual travel times using boxplots. We can see that most individuals have a very small variation in travel times, and that the time gains are made for the few individuals that managed to get their transfer after applying tactics. Figure 4.6 also presents the percentage of passengers with successful or missed transfers and the percentage of passengers that missed their bus (not counting missed transfers). Finally, table A.1 summarizes this information.

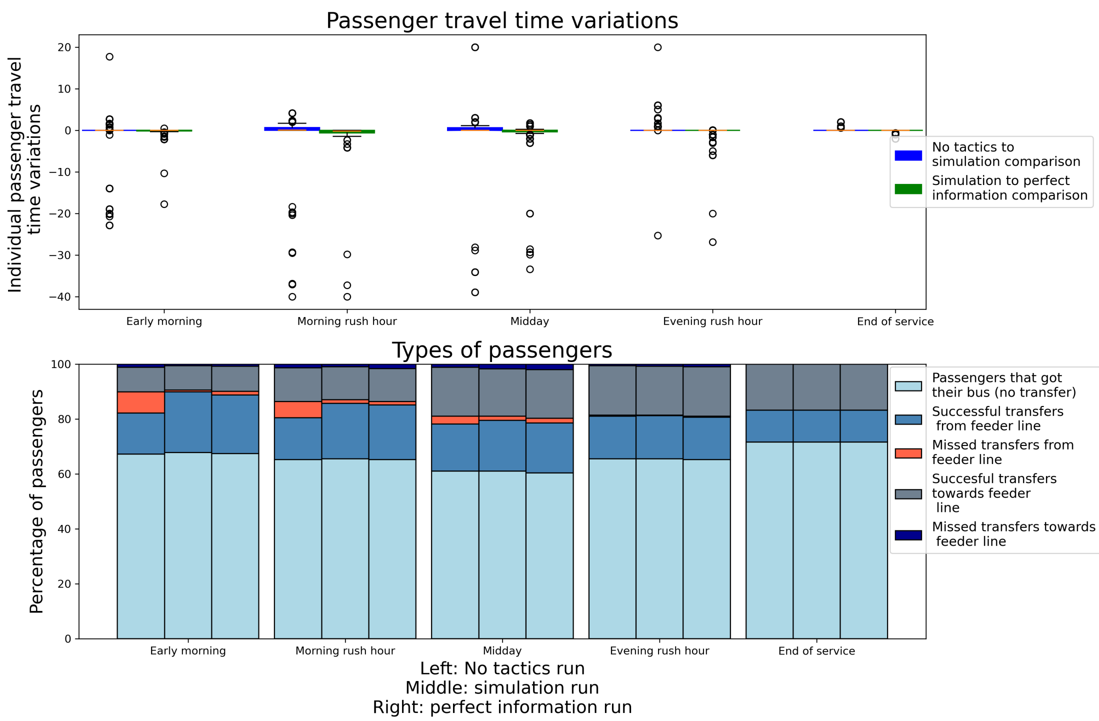

Figure 4.6 Passenger travel time variations for all buses and percentage of passenger transfers. Holding only.

Table A.3 summarizes passenger travel times for the case with holding and skip-stop tactics. We can note that the additional walking time due to skip-stop tactics is minimal. When looking at individual passenger travel time variations for the case with holding and skip-stop tactics, we can note that the boxplots remain compact. This indicates that time gains are again mainly explained by turning missed transfers into successful transfers. Passenger travel time gains because of skipped stops are minimal. Table A.4 presents the passenger travel times for the holding with speed control tactic case. We note that the speed control tactic case has overall passenger travel times smaller than the skip-stop tactic case. We note that the speed control tactic was less effective than the skip-stop tactic in increasing the number of successful transfers. Moreover, there was a bigger variation in individual passenger travel times due to the decrease in in-bus travel time. Finally, table A.5 presents the passenger travel times for the holding with skip-stop and speed control tactics case.

Figure 4.7 presents the bus travel times for holding tactic only, comparing the no tactics case, the online simulation framework case and the offline perfect information case. Figure 4.7 also presents the percentage of stops at which tactics are applied for each cluster. We note that the differences in mean bus travel times are minimal between the three cases. Moreover, holding tactics are applied to a very limited number of stops. Table A.2 further summarizes these results. Table A.6 summarizes bus travel times for the case with holding and skip-stop tactics, and shows the percentage of stops with different tactics. In this case, the skip-stop tactics compensate for any additional time added by the holding tactic. The skip-stop tactic is used more often than the holding tactic. Table A.7 presents the bus travel times the holding with speed control tactics case. It also shows the percentage of stops where tactics were applied. The speed control tactic was applied to a large proportion of stops (more than 30%). This was compensated by more holding time when the bus had to wait for transfer passengers, or in order not to be too early compared to the schedule. Finally, table A.8 presents the bus travel times for the holding with skip-stop and speed control tactics case. It also shows the percentage of stops where tactics were applied.

4.6 Conclusion

Real-time control is a crucial part of planning for PT operators. The waiting times of passengers and the reliability of transfers significantly influence the service quality of PT. This research concentrates on improving transfers times and reducing travel times by using real-time control tactics. The methodology is based on a case study of the city of Laval, Canada with a month of data from their PT network data. A deterministic arc-flow model is developed to integrate three different real-time control tactics : holding, skipping stops/skipping segments and speeding up. Firstly, the model is tested using real data from the case study. Then, to simulate real-time operations, a stochastic simulation framework generating data with different levels of uncertainty is created. This research evaluates five cases in the simulation framework : no tactics, holding only, holding with skip-stop/skip-segment, holding with speed control and finally holding combined with both speed control and skipping stops. For each run of the simulation, the results of the simulation are compared to the results of the perfect information, deterministic case. The model is tested on a large number of instances to determine optimal data generation parameters in the simulation framework. Finally, more simulation runs are made to validate the optimal parameters of the simulation and the results

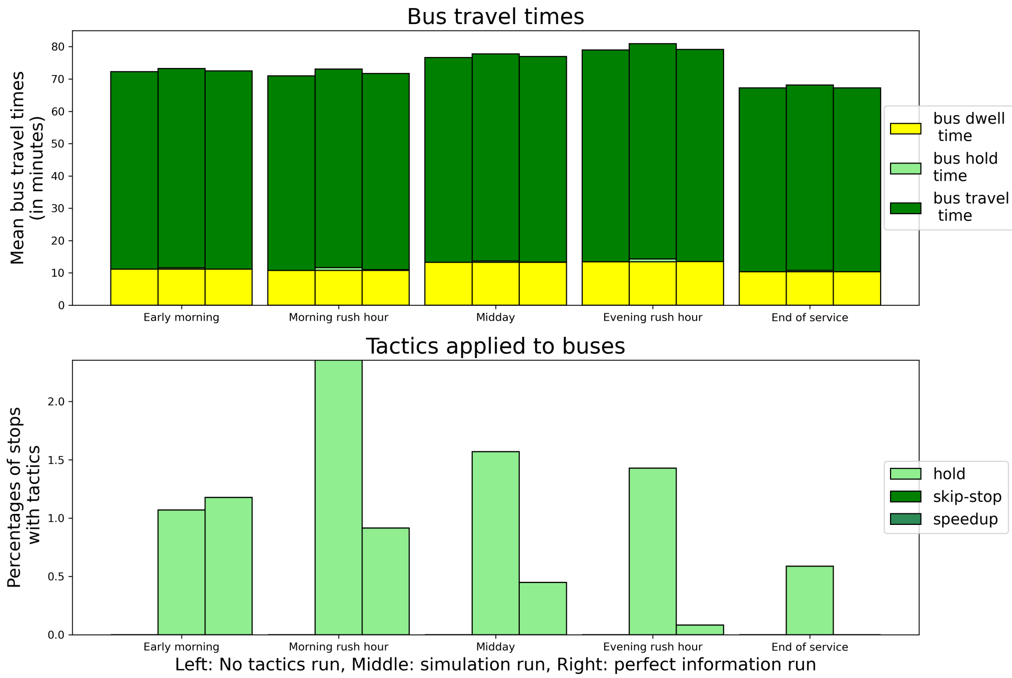

FIGURE 4.7 Bus travel time and percentage of stops with tactics. Holding only.

of the model.

The results show that an improvement in total travel times, in individual travel times and in the number of successful transfers is made for any type of data generation compared to the no tactics case. The best improvements occur in the cases when buses are in advance compared to their planned schedules. In these cases, small holding times allow for great improvements in passenger travel times.

The performance of the model is limited by the assumption that passengers do not leave the system but wait for the next bus. In an urban environment, it is unlikely that a passenger will always wait 20 to 30 minutes for the next bus. For future research, the performance of the model could be improved by an in-depth analysis of travel origin/destination pairs. In that case, boarding and alighting passengers could be generated together and not separately. Moreover, the research on a single main line could be expanded to multiple lines in the bus network.

CHAPITRE 5 ARTICLE 2 : ONLINE STOCHASTIC OPTIMIZATION FOR REAL-TIME TRANSFER SYNCHRONIZATION IN PUBLIC TRANSIT NETWORKS

L. Kolcheva, A. Legrain et M. Trépanier ont écrit cet article et l'ont soumis le 19 février 2025 dans *INFORMS Journal on Computing*.

5.1 Introduction

Public transit (PT) systems play an increasingly vital role in addressing urban growth, traffic congestion, and sustainable development. PT networks are developed through several phases, including planning, operation, and control, with network redesigns and operational adjustments often being costly and challenging. However, innovative control strategies offer a more cost-effective means of enhancing PT network performance.

Research shows that transfer speed and reliability are critical factors influencing passengers' willingness to use public transit [19]. While transfers are typically synchronized during the planning phase, buses operate in a stochastic environment that often leads to schedule deviations, resulting in missed transfers and reduced ridership. Consequently, real-time transfer synchronization strategies are attracting increasing attention. This study proposes an arc-flow formulation to address the real-time transfer synchronization problem within a dense PT network. Three control tactics are employed, either individually or in combination, to optimize transfers and minimize passenger travel times. These tactics are integrated into the arc-flow formulation, where variables capture passenger flows across the network. We then implement two online stochastic optimization (OSO) algorithms, incorporating the offline arc-flow model in a dynamic, event-based environment with a rolling control horizon. Predictions of the future state of the PT system are generated using a combination of real-time and historical data provided by the *Société de Transport de Laval* (STL). The key contributions of this study are as follows :

- Development of an arc-flow formulation that enumerates all control tactics for the real-time PT transfer synchronization problem, designed to solve large-scale instances with multiple feeder lines and extended planning horizons in real time.
- Introduction of two online stochastic optimization algorithms specifically tailored for the real-time transfer synchronization problem.
- Integration of real-time data from multiple sources, including AVL systems and smart card data, to generate relevant scenarios and address uncertainty in decision-making.

— Performance evaluation on a large-scale real dataset with 29 lines from a mid-size PT network.

5.2 Literature review

Transfer synchronization has been extensively studied in the PT literature [14–16]. More recently, [17] provide a comprehensive overview of transfer synchronization strategies specifically in the real-time control phase.

5.2.1 Hold tactic

The hold tactic requires a bus to wait at a designated stop for a specified period, which can result in reduced operating speed and increased passenger travel or waiting times as a trade-off. Despite these drawbacks, the hold tactic is widely regarded in the literature as one of the most effective control measures [14]. Its effectiveness can be further enhanced by using multiple control stops. [24] demonstrate that when all stops serve as control points, disruptions are more effectively contained through distributed control across the entire line.

The hold tactic can be used to ensure successful planned transfers. For instance, [73] apply this tactic using real-time bus location data to support transfers synchronized during the planning phase, validating their method on an artificial network with three transfer points. Similarly, [74] implement hold times to protect planned transfers by waiting for delayed buses, taking into account the stochastic nature of travel times and modeling random passenger arrivals. Their model is tested with two transfer points. Lastly, [39] propose a dynamic model for the hold tactic to synchronize transfers within a multimodal transportation network. They demonstrate their approach using simulations involving four bus trips and around 40 transferring passengers.

Recent studies have increasingly examined the use of real-time passenger data. In a study by [44], real-time synchronization of transfers between a main line and a feeder line is implemented at a single stop using dynamic predictions of passenger transfer demand. This approach combines historical data on passenger origin-destination pairs with real-time information derived from smart card usage. The study assumes that AFC data of both boarding and alighting is available in real time. [33] evaluate holding strategies aimed at balancing service regularity with transfer synchronization on a high-frequency bus line. Real-time passenger demand is integrated into this study through bus occupancy data and smart card validation records from the tramway network in The Hague, The Netherlands. Tests are conducted using BusMezzo, a traffic simulation model, and encompass two lines with two transfer stops and a total of five control stops.

5.2.2 Skip-stop and combined tactics

While the skip-stop tactic can reduce travel times for onboard passengers, it may inconvenience passengers who wish to board or alight at skipped stops, especially on low-frequency bus lines. [37] introduce the concept of transfer segments - consecutive stops where two vehicles can exchange passengers. The authors present a dynamic programming model that minimizes total passenger travel time by applying both hold and skip-stop tactics. They test it on an artificial network with three bus lines and fourteen transfer segments. Subsequently, [19] develop a mixed-integer program running in exponential time using the hold and skip-stop tactics for the transfer synchronization problem. The model is tested on a similar artificial network with simulated demand on three bus lines with two transfer points.

In recent years, an increasing number of studies have employed simulation to address the transfer synchronization problem. [75] combine the skip-stop and hold tactics within a rule-based method with polynomial running time. They use microsimulation to evaluate the performance of different rules for applying these tactics at a single, high-ridership transfer stop, relying on historical data for passenger demand. Similarly, [40] develop an event-based simulation using a rolling prediction horizon where each re-optimization considers a single predicted state of the PT network and includes approximately forty variables. Their optimization model minimizes total passenger travel time by combining the hold and speedup tactics. The authors test the model on a network with three lines and three transfer stops, achieving an average runtime of 1.25 seconds over a horizon of the same scale.

5.2.3 Online stochastic optimization algorithms

Considering uncertainty has become essential for obtaining practical and relevant results [41]. Uncertainty can be modeled by generating multiple scenarios to represent possible future states of the system, thus providing more robust solutions. OSO algorithms based on multiple scenarios generally comprise three main components : a sampler that generates scenarios, an offline solver that provides a solution for each scenario, and a decision process that integrates information from all solutions [46, 47].

The most commonly used OSO algorithms generating multiple scenarios are E [48], C and R. These algorithms have been adapted for various applications [49], but not for the transfer synchronization problem. Algorithm E, when applied to the Vehicle Routing Problem, evaluates the cost of visiting each customer first by enforcing this decision across all scenarios, then performs a complete optimization for the remaining customers. While Algorithm E provides a comprehensive evaluation, its computational intensity can make it less suitable for online applications. Algorithm C [50, 51] selects the statistically optimal decision over all scenarios. Lastly, Algorithm R [52] approximates Algorithm E by avoiding full re-optimization of all scenarios, selecting a decision that performs reasonably well across the scenario set. While these algorithms may share the same sampler and offline solver, they differ in their decision processes. Other studies have enhanced computational efficiency by generating fewer, yet more relevant, scenarios based on historical data [53–55]. Additionally, advancements in scenario generation have been achieved through the use of machine learning techniques and sampling from historical data [56].

5.2.4 Research gap

Existing studies on transfer synchronization primarily focus on a single stop or a limited number of stops, constrained by high computational costs. Passenger demand is frequently modeled using average arrival rates, rather than leveraging actual historical data. Moreover, the utilization of real-time data is generally confined to AVL and APC systems, with other data sources being incorporated only in offline methodologies. Real-time integration of smart card data, which provides precise passenger demand information and facilitates improved demand forecasting, remains largely under explored. Lastly, while OSO algorithms have demonstrated potential in related fields, their application in the context of transfer synchronization has yet to be investigated.

The rest of this article is organized as follows. Section 5.3 presents the offline arc-flow formulation for the transfer synchronization problem. Section 5.4 describes the implementation of OSO algorithms based on the offline solver. Section 5.5 introduces the case study used to test the methodology, provides a sensitivity analysis of the models, and details the results of all experiments. Lastly, Section 5.6 offers our conclusions, including a discussion of findings and suggestions for future research.

5.3 Offline model

We formulate an arc-flow model to tackle the offline transfer synchronization problem through the application of control tactics, minimizing total passenger travel time by improving transfer times. This model builds upon and refines the authors' previous work [76] enhancing both computational efficiency and variable reduction by limiting the number of paths added through predefined rules. The arc-flow formulation uses time-expanded graphs incorporating control tactics. Upon launching the model, users can specify which tactics—individually or in combination—should be considered during graph construction.

5.3.1 Arc-flow formulation

The arc-flow model optimizes transfer synchronization along a main line within a predefined control horizon. Feeder lines, connecting to the main line at transfer stops, are treated as fixed inputs with predetermined arrival times, unaffected by control tactics. Each main line stop within the control horizon, up to the final transfer stop, serves as a control point enabling more efficient control. The model accounts for all trip stages—waiting, travel, and transfers—for both onboard passengers and those intending to board. In the arc-flow model, graph nodes represent stops at specific times, with boarding and alighting passengers modeled as positive and negative exogenous flows, respectively. Transfers are determined by feeder line arrival times at transfer stops, while direct boarding passengers are assumed to arrive shortly before the main line bus's arrival time. Graph arcs capture travel or waiting times, weighted by the time difference between origin and destination nodes. Passengers can wait for either a main line or feeder line bus at their current stop, but travel between stops in the horizon is only allowed when onboard a main line bus. To accurately represent flow dynamics, travel arcs link consecutive main line stops, while feeder lines are modeled only by their arrival and departure times at transfer points.

In the graph construction phase, control tactics are incorporated to achieve two objectives : (1) enabling missed transfers and (2) maintaining adherence to planned schedules for timely boarding of non-transfer passengers. The hold tactic is modeled by adding arcs between existing nodes to represent waiting times at stops, simplifying the graph by avoiding the creation of additional nodes and reducing possible hold durations. Before integrating speedup and skip-stop tactics at a stop, the model evaluates whether the bus is behind schedule or if applying these tactics could enable missed transfers. If neither condition applies—or if the bus is significantly ahead of schedule—these tactics are excluded for that stop. The speedup tactic, with a speedup factor of 0.8 (based on STL recommendations), reduces travel time between stops by allowing the bus to reach the next stop earlier. A "faster" arc is added with a new node representing an earlier arrival time. This tactic is applied only when inter-stop travel times exceed 30 seconds to ensure safety and meaningful time savings. The skip-stop tactic bypasses selected stops by adding a node and arc where boarding is prohibited. Passengers needing to alight at skipped stops are accommodated with walking arcs reflecting the required walking time from the closest non-skipped stop. Transfer stops, however, cannot be skipped to maintain transfer reliability.

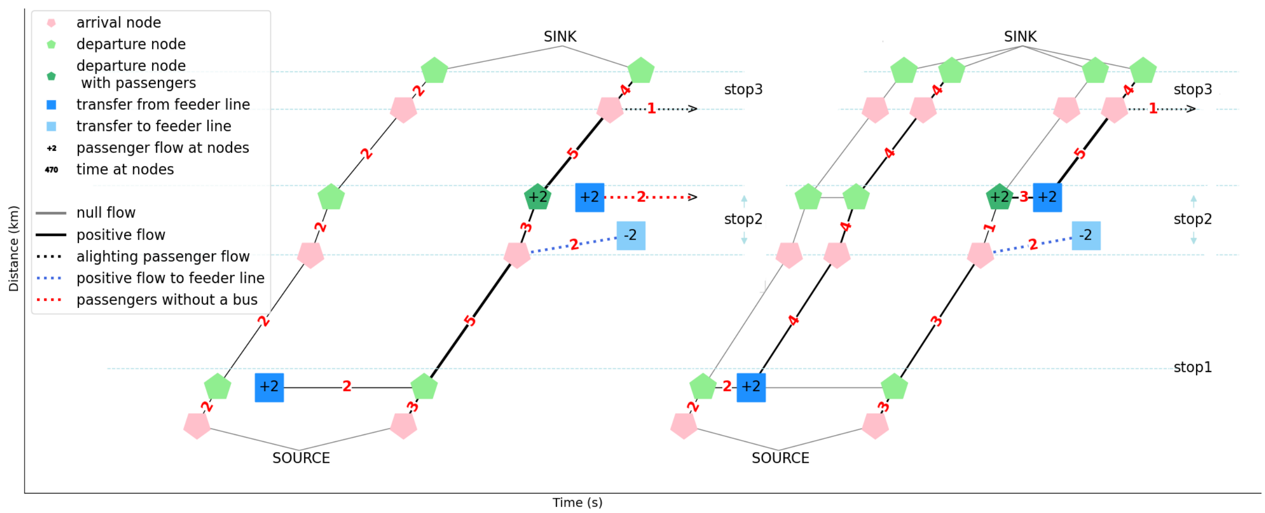

FIGURE 5.1 Mock example of a graph construction. Left-no tactics, right-with tactics.

Figure 5.1 provides a conceptual visualization of the graph construction, illustrating how control tactics create additional bus paths and enable more transfer opportunities. Arcs are labeled with non-zero passenger counts for clarity, with arrival and departure nodes slightly offset on the y-axis for better visualization. Bold black arcs indicate the optimized path derived from the arc-flow model solution, while gray arcs represent alternative routes. Exogenous flows are depicted on the nodes. The figure example includes two main line vehicles and three stops within the control horizon. In the "no tactics" case (left), two passengers miss the first bus at "stop 1" and must wait for the next one. At "stop 2," a late-arriving transfer bus causes two passengers to miss their intended connection to the main line, leaving them unable to board a main line bus within the control horizon, as shown by the red dotted arrow. In the "with tactics" case (right), additional arcs create alternative paths for both buses. In the optimized solution, specific hold times enable passengers to make their transfers, thereby minimizing total passenger travel time.

The time-expanded graphs, which integrate all possible tactics, serve as inputs for a minimum-cost flow problem. The integration of tactics—and the resulting addition of new paths—is automated and constrained by the previously mentioned conditions. This approach ensures that only realistic alternative paths within a defined time window around the planned schedule or current delay are considered. All paths incorporated into the graphs are then evaluated in the optimization problem.

Table 5.1 Sets, parameters and variables used in the mathematical formulation.

Sets		
N		Set of nodes with s as the source node and t as the sink node.
N^{-}		Set of nodes with negative exogenous flows, not including t.
A		Set of arcs $(u,\ v)$ in the graph.
B		Set of main line buses available for routing in the graph.
S		Set of main line stops within the control horizon.
A_b^s	$b \in B,\ s \in S$	Set of arcs for the bus b between stops s and $s+1$, used to define stop-to-stop travel.
$A_{v,b}^{travel}$	$b \in B,\ v \in N^{-}$	Set of arcs used to board bus b by passengers alighting from bus b at node v. There is one arc per passenger.
$A_{v,b}^{missed}$	$b \in B,\ v \in N^{-}$	Set of arcs used to board bus b by passengers who missed a previous bus, and are now alighting from bus b at node v. There is one arc per passenger.
Parameters		
c_{uv}	(u,v) $\in A$	Passenger flow capacity on arc (u, v), representing bus capacity.
w_{uv}	(u,v) $\in A$	Travel time between nodes u and v.
f_v	$v \in N$	Exogenous passenger flow at node v (passengers boarding or alighting at stops).
g_v	$v \in N$	Exogenous bus flow at node v ($g_s = \lvert B \rvert$, $g_t = -\lvert B \rvert$, $g_v = 0, \forall v \neq s, t$) for source, sink, and intermediate nodes respectively.
p	$p \geq 0$	Additional perceived cost for waiting outside the bus relative to in-bus waiting.
Variables		
x_{uv}	$(u, v) \in A$	Passenger flow on arc (u, v), where $x_{uv} \in \mathbb{N}$
y_{uv}	$(u, v) \in A$	Binary variable for bus flow on arc (u, v), where $y_{uv} \in \{0, 1\}$. Here, $y_{uv} = 1$ if a bus is present on arc (u, v), and $y_{uv} = 0$ otherwise.
z_{uv}	$(u, v) \in A$	Indicator variable defined as $z_{uv} = \begin{cases} x_{uv}, & \text{if } y_{uv} = 0 \\ 0, & \text{if } y_{uv} = 1 \end{cases}$ captures out-of-bus waiting time for passengers when no bus is present on arc (u, v).

5.3.2 Mathematical formulation

The following assumptions are made for the offline model. When faced with multiple possible paths in the time-expanded graphs, passengers choose the fastest route. Due to access to real-time information, passengers arrive shortly before the bus arrival. Street conditions permit the application of speedup tactics between stops. Passengers are informed in advance if their intended stop will be skipped. If they cannot alight due to this tactic, they will disembark at the closest available stop and proceed on foot to their original destination through a walking arc with the corresponding walking time. Passengers who miss boarding due to a skipped stop or who miss their transfer, wait at their current stop for the next available bus.

Objective

$$\min \quad \sum_{(u,v)\in A} w_{uv}x_{u,v} + \sum_{(u,v)\in A} w_{uv}pz_{u,v} \tag{5.1}$$

Constraints

$$\sum_{(v,w)\in A} x_{vw} - \sum_{(u,v)\in A} x_{uv} = f_v, \forall v \in N \setminus N^-, v \neq t \tag{5.2}$$

$$\sum_{(v,w)\in A} x_{vw} - \sum_{(u,v)\in A} x_{uv} = f_v + \sum_{(u,w)\in A^{travel}_{v,b}} (1 - x_{uw}) - \sum_{(u,w)\in A^{missed}_{v,b}} (1 - x_{uw}),$$

$$\forall b \in B, v \in N^- \tag{5.3}$$

$$\sum_{(v,w)\in A} y_{vw} - \sum_{(u,v)\in A} y_{uv} = g_v, \forall v \in N \tag{5.4}$$

$$x_{uv} \leq c_{uv}y_{uv}, \forall b \in B, s \in S, (u,v) \in A^s_b \tag{5.5}$$

$$x_{uv} - c_{uv}y_{uv} \leq z_{uv}, \forall (u,v) \in A \tag{5.6}$$

$$y_{uv} \in \{0,1\}, \forall (u,v) \in A \tag{5.7}$$

$$x_{uv}, z_{uv} \geq 0, \forall (u,v) \in A \tag{5.8}$$

The objective 5.1 minimizes the perceived passenger travel time, with an additional cost associated with out-of-bus waiting time. The first term minimizes the total passenger travel time across all arcs. The second term accounts for the additional cost of out-of-bus waiting time, using the parameter p to weigh the discomfort of waiting outside the bus relative to in-bus travel. Constraints 5.2 ensure the flow conservation for passenger flows at nodes without negative exogenous flow. Constraints 5.3 ensure the flow conservation for passengers at nodes where passengers may alight, adjusting for passengers who miss their intended bus and rejoin the flow with the next available bus. These constraints maintain the alignment of solution flows with historical passenger origin/destination pairs, and allow the calculation of the number of missed transfers. Constraints 5.4 ensure the flow conservation for buses. The source node has a positive exogenous bus flow, while the sink node has a negative exogenous bus flow, both corresponding to the number of main line buses in the control horizon. Constraints 5.5 ensure passengers can only travel between stops while aboard a main line bus. If no bus is present (i.e., $y_{uv} = 0$), the flow x_{uv} is restricted to 0. Constraints 5.6 calculate out-of-bus waiting times by defining z_{uv} to capture out-of-bus waiting periods on arcs where no bus is present (i.e., $y_{uv} = 0$). Constraints 5.7 define the bus flow variables and constraints 5.8 ensure that passenger flows are non-negative.

5.4 Online stochastic optimization

The offline model presented in this work assumes perfect knowledge of the system for an unlimited control horizon, allowing control tactics for improving passenger transfers to be predetermined. However, PT operations are dynamic and often uncertain, with stochastic passenger demand, travel times, and bus arrival times. To address these challenges, we formulate a dynamic environment that leverages real-time data. Then we implement two OSO algorithms (C and R) based on the offline model to make real-time operational decisions under uncertainty. To evaluate their performance, we implement two additional approaches : a **Deterministic (D)** algorithm, which relies on average historical data, and a **Perfect Information (PI)** solution, which assumes complete knowledge of the system state. By contrasting these approaches, we evaluate the added value of real-time data and the use of scenarios to model uncertainty in the real-time dynamic transfer synchronization problem.

5.4.1 Dynamic environment

To account for the gradual reveal of real-time data in PT operations, we implement a discrete-event dynamic environment, illustrated in Figure 5.2. This environment triggers a system re-optimization each time a main line bus departs from its current stop.

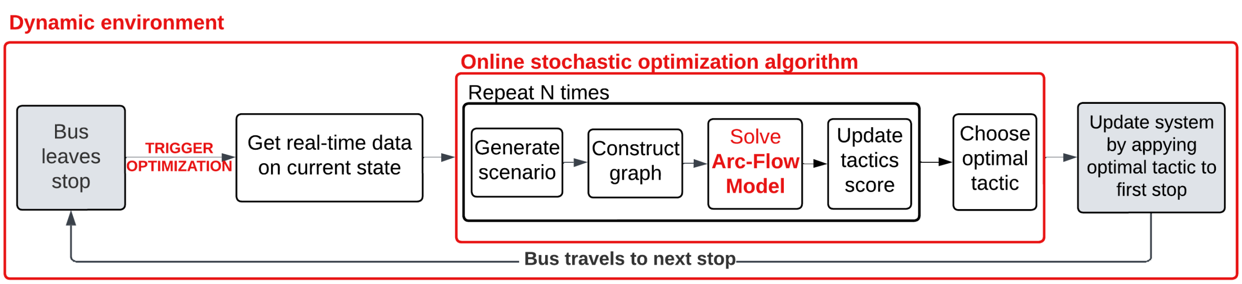

FIGURE 5.2 Dynamic environment for the online application of the arc-flow model.

At the start of each iteration, the system state S is updated with real-time data, including bus positions and delays from AVL systems, occupancy levels from APC systems, and passenger demand at previous stops from real-time smart card data, all within a defined control horizon h. This updated state serves as input for the OSO algorithms. The first main line stop in S is designated as s_0, where an OSO algorithm selects a control tactic σ_{s_0} from tactics hold, speedup, a combination of both, or skip-stop for non-transfer stops. The selected tactic σ_{s_0} is applied and the system updates the positions of buses, the number of passengers onboard, and those waiting at stops within the control horizon according to the impact of σ_{s_0}. The re-optimization process is then complete, with s_0 becoming the new current stop. This iterative process continues until the bus reaches the end of the line.

5.4.2 Sources of uncertainty and scenario generation

While S captures the current state of the transit system, key information about future travel and dwell times, passenger arrivals, transfer demand, and feeder line arrival times remains unknown, introducing uncertainty into the optimization process.

To address this uncertainty, our OSO algorithms generate scenarios integrating historical and real-time data to model possible realizations of the system state. To generate a scenario, we consider the following parameters of the state S : the main line, the starting time corresponding to the time of departure from the current stop and the control horizon of h stops. Travel times, dwell times, and passenger demands (including transfers) for all stops in h are independently and randomly sampled from historical data clusters categorized by time, route, and boarding/alighting stop. Passenger and transfer demand generation takes into account current bus occupancy derived from real-time smart-card data. Real-time information on current feeder line delays is incorporated to generate arrival times for transfer vehicles within the horizon. This data is then used to define bus trips along the main line and allocate passenger flows on the main and feeder lines, resulting in a scenario ω. Random sampling from the historical data clusters allows the creation of diverse random scenarios using the same inputs, providing the basis for robust decision-making in real time optimization. The specific clustering of the data used in this study is detailed in section 5.5.1.

5.4.3 Online stochastic optimization algorithms : C and R

This section introduces two OSO algorithms, adapted from the literature [52], to address the transfer synchronization problem. Algorithm 1 provides the general framework for these algorithms within the dynamic environment.

Both C and R algorithms use the system state S from the dynamic environment as input to generate a number N of scenarios. For each scenario ω, a graph G^{ω} is constructed to incorporate control tactics in T, a subset of all possible tactics. The offline model defined in Section 5.3.2 solves the problem corresponding to graph G^{ω}. From the solution we derive a sequence of optimal tactics $\boldsymbol{\sigma}^{*}=(\sigma^{*}_{s_i})_{i=0}^{h-1}$, which specifies the tactics applied at each stop

Algorithm 1 Online Stochastic algorithm

function ONLINE_STOCHASTIC_ALGORITHM(S, T, N, ALG)
  score(σ) $\leftarrow 0, \forall \sigma \in T$
  repeat N times
    $\omega \leftarrow$ GENERATE_SCENARIO(S, ALG)
    $G^\omega \leftarrow$ BUILD_GRAPH(ω, T)
    $\boldsymbol{\sigma}^* \leftarrow$ SOLVE_OFFLINE_MODEL(G^ω)
    score $\leftarrow$ UPDATE_SCORE(G^ω, $\boldsymbol{\sigma}^*$, ALG, score)
  return $\arg\max_{\sigma \in T}$ score(σ)
end function

within the control horizon h. The OSO algorithms then evaluate the solution, focusing on the first stop s_0 in the control horizon, and update the scores of tactics in T. After considering the scores for all scenarios, the algorithms choose the tactic with the highest score to be applied at s_0. Tactics for subsequent stops are re-evaluated during the next re-optimization, ensuring adaptability to updated system states.

Algorithm C evaluates multiple scenarios and selects the tactic that is most frequently optimal for s_0 across all scenarios. For each scenario where a tactic $\sigma^*_{s_0}$ is optimal for s_0, its score is incremented by one (i.e., the function UPDATE_SCORE updates the score as score($\sigma^*_{s_0}$) += 1). The final score of a tactic is determined by the number of scenarios for which it is optimal. While Algorithm C returns the most frequently optimal tactic, it does not provide information on tactics that perform similarly. Additionally, it does not evaluate how effectively a tactic performs across all scenarios or recognize that a tactic might never be optimal yet still be robust overall.

On the other hand, Algorithm R aims to select a tactic that performs consistently well across all scenarios. Algorithm 2 outlines the process for updating the scores of all tactics after evaluating each scenario. Adapted to the transfer synchronization problem, the R algorithm

Algorithm 2 R algorithm

function UPDATE_SCORE(G^ω, $\boldsymbol{\sigma}^*$,R, score)
  $obj \leftarrow$ GET_OBJECTIVE_VALUE(G^ω,$\boldsymbol{\sigma}^*$)
  for $\sigma \in T \setminus \{\sigma^*_{s_0}\}$ **do**
    $\boldsymbol{\sigma'} \leftarrow (\sigma, \sigma^*_{s_1}, ..., \sigma_{s^*_{h-1}})$
    $obj' \leftarrow$ GET_OBJECTIVE_VALUE(G^ω,$\boldsymbol{\sigma'}$)
    score(σ)$+ =$($obj - obj'$)
  end for
  return score
end function

calculates the loss, or "regret", incurred by using sub-optimal tactics at the first stop in the control horizon while retaining optimal tactics for subsequent stops. In this formulation, the "regret" quantifies the additional travel and wait times passengers experience due to deviations from the optimal tactic at the first stop. For a given scenario ω, the "regret" of a tactic $\sigma \neq \sigma^*_{s_0}$ is defined as the increase in the objective function caused by replacing the optimal sequence of tactics $\boldsymbol{\sigma}^*$ with the sub-optimal sequence $\boldsymbol{\sigma'}$=$(\sigma, \sigma^*_{s_1}, ..., \sigma^*_{s_{h-1}})$. This restriction fixes all bus flow variables and limits passenger flows to a predefined path, simplifying the objective function calculation to a straightforward computation of costs along the used arcs in graph G^ω. This approach avoids solving for all possible paths in G^ω, significantly reducing complexity. Additionally, the same underlying graph G^ω is reused to compute "regret" values for all tactics, enabling the evaluation of the performance of each tactic without substantially increasing computation time. Algorithm R then selects the tactic with the smallest "regret" across all scenarios, effectively penalizing poorly performing tactics that accumulate large "regret" values and rendering them non-competitive. Consequently, tactics returned by Algorithm R are robust, avoiding poor performance in any scenario. Additionally, similar-performing tactics tend to have closely aligned "regret" values. To fit the framework of Algorithm 1, which returns the argument of the maximum score, the algorithm uses the negative of the "regret" values for evaluation.

The implementation of Algorithm R is more computationally demanding than Algorithm C, as it calculates "regret" values for all possible tactics in each scenario, while Algorithm C updates the score for only one tactic per scenario. However, this additional complexity enables Algorithm R to provide a more comprehensive evaluation of tactic performance, ensuring that the selected tactic is robust and effective across a wide range of conditions.

5.4.4 Perfect Information solution and Deterministic algorithm

This section presents two approaches using Algorithm 1 with a single scenario ($N = 1$), the *Perfect Information* (PI) solution and Algorithm *DETERMINISTIC* (D), both implemented to benchmark the performance of the proposed OSO algorithms. The PI solution applies the offline model in a dynamic environment, where decisions on tactics are made based on the real historical system state within the control horizon, without any uncertainty, and thus setting a theoretical upper bound on Algorithms C and R's performance. The single scenario used corresponds to the true state of the system, and the optimal tactic $\sigma^*_{s_0}$ for stop s_0, as determined by the offline model, is returned. When only the hold tactic is allowed, the offline model is capable of solving large-scale instances in real time (computation times of less than 1 second), including scenarios with three consecutive buses on a main line with over 80 control stops and numerous transferring stops and feeder lines. However, when the speedup or skip-stop tactics are permitted, the number of possible paths for each bus increases significantly. In such cases, the offline model cannot handle instances involving many transfers and three consecutive buses for horizons with more than 30 control stops (no solution after 15 minutes of computation). As a result, the PI solution is used instead of the offline model to evaluate the performance of the OSO algorithms in this study. Notably, when only the hold tactic is considered, the performance of the PI solution with a limited horizon of 15 stops closely matches that of the offline model with perfect information and an unlimited horizon.

Algorithm D does not model uncertainty either. It generates a single scenario ω using the mean historical values for all unknown variables describing the system state S, representing the average expected state of the system. The optimal tactic $\sigma^*_{s_0}$ for stop s_0, as computed by the offline model, is returned. Algorithm D is based on a static view of the system and provides a baseline to determine if the use of scenario generation to model uncertainty in Algorithms C and R provides significant improvements over a deterministic approach. Additionally, Algorithm D is simpler to implement than Algorithms C and R, as it requires only one scenario based on historical data. Consequently, its computation time is significantly lower.

5.5 Experiments

This section describes the case study used for the numerical experiments, conducts a sensitivity analysis of the OSO algorithms, and presents results from tests performed on 29 different bus lines within a dense urban PT network.

5.5.1 Case study

The experiments are based on data provided by the STL, serving a Canadian city with a population of 436,000. The STL's PT network consists of 46 bus lines, many of which connect to the Montreal subway system, and includes more than a thousand bus stops. The majority of passengers rely on mobile applications for real-time schedule updates and estimated arrival times. The dataset spans one month of operations and includes detailed information on bus routes, schedules, ridership, and bus locations. This data is collected in real time from multiple sources, including AFC or smart cards, AVL and APC systems installed on all vehicles. By leveraging accurate smart card data and insights from previous research on individual trip destinations [30], the dataset provides reconstructed passenger origin-destination pairs and multi-segment trips. This rich dataset enables the use of real passenger demand and actual transfer flows in the experiments, ensuring high fidelity in the numerical analysis.

To ensure the dataset is suitable for scenario generation and testing, a preprocessing step is conducted. Weekends and holidays are excluded from the dataset, and the remaining weekdays are divided into three working sets : training (80%), testing (10%) and validation (10%). The training set is used to derive structured representations of key variables describing the state of the PT system, including travel times, dwell times, passenger boarding and alighting counts from smart card data, transfer flows, and headway times for all lines. Data for each variable is clustered based on the time of day, capturing distinct patterns for peak and off-peak periods for each line and stop. Outliers are removed from these clusters to enhance reliability. Instead of generating values randomly from statistical distributions, scenario generation samples data from real observations in the clusters. The testing set is then used to evaluate the scenario generation process and conduct sensitivity analyses of the algorithms, while the validation set supports numerical experiments.

5.5.2 Sensitivity analysis

This section examines the sensitivity of the OSO algorithms to variations in the control horizon length, the weight assigned to out-of-bus waiting time for passengers, and the number of scenarios considered during each re-optimization.

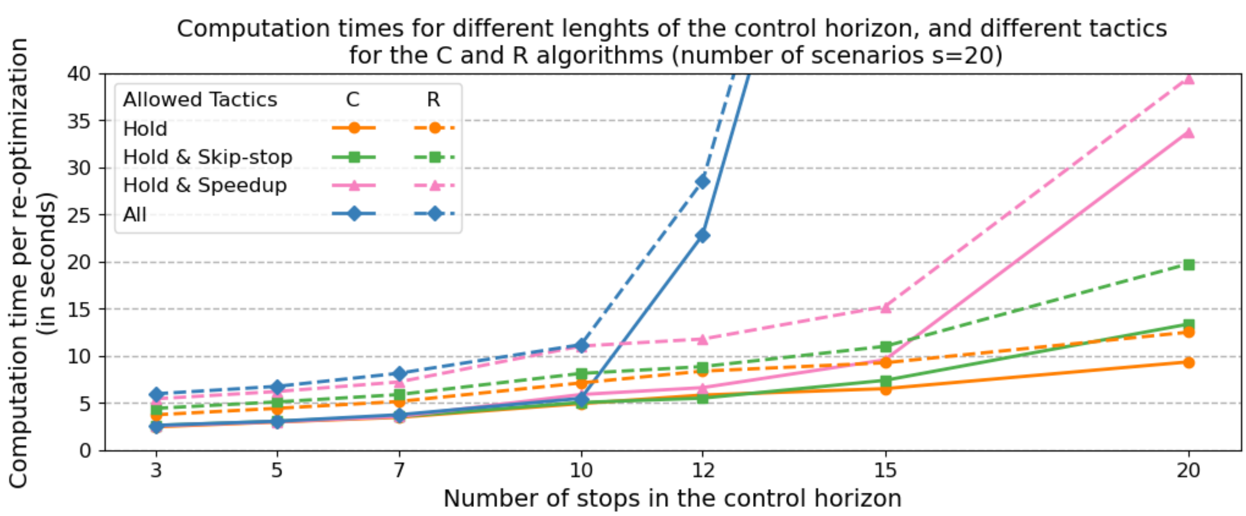

Figure 5.3 Computation times per re-optimization for different control horizon lengths, algorithms and tactics.

Figure 5.3 illustrates the average computation time per re-optimization in the dynamic environment for Algorithms C and R under varying control horizon lengths. These times include the entire re-optimization process, including updating the system state, scenario generation, construction of a time-expanded graph for each scenario, solving the offline problem for each scenario, updating tactic scores for each scenario, choosing an optimal tactic and applying the optimal tactic in the dynamic environment. Each re-optimization considers 20 scenarios per stop, and computation times scale linearly with the number of scenarios. Algorithm D and the PI solution are excluded from the figure due to their significantly shorter computation times, as they solve only one scenario per re-optimization.

As previously explained, the time-expanded graphs used as inputs for the arc-flow model are generated by automatically incorporating paths for all allowed tactics. Consequently, the inclusion of additional tactics and longer control horizons increases the complexity of these graphs. When the underlying graphs include only the hold tactic or a combination of hold and skip-stop tactics, computation time grows linearly with respect to the length of the control horizon. In contrast, the inclusion of the speedup tactic causes an exponential increase in computation time as the control horizon lengthens. When all tactics are permitted, computation time becomes prohibitively large for real-time implementation in scenarios with control horizons exceeding 12 stops. Nonetheless, the results show that all algorithms can be executed within ten seconds for a control horizon of ten stops, which is sufficient for the STL's operational needs.

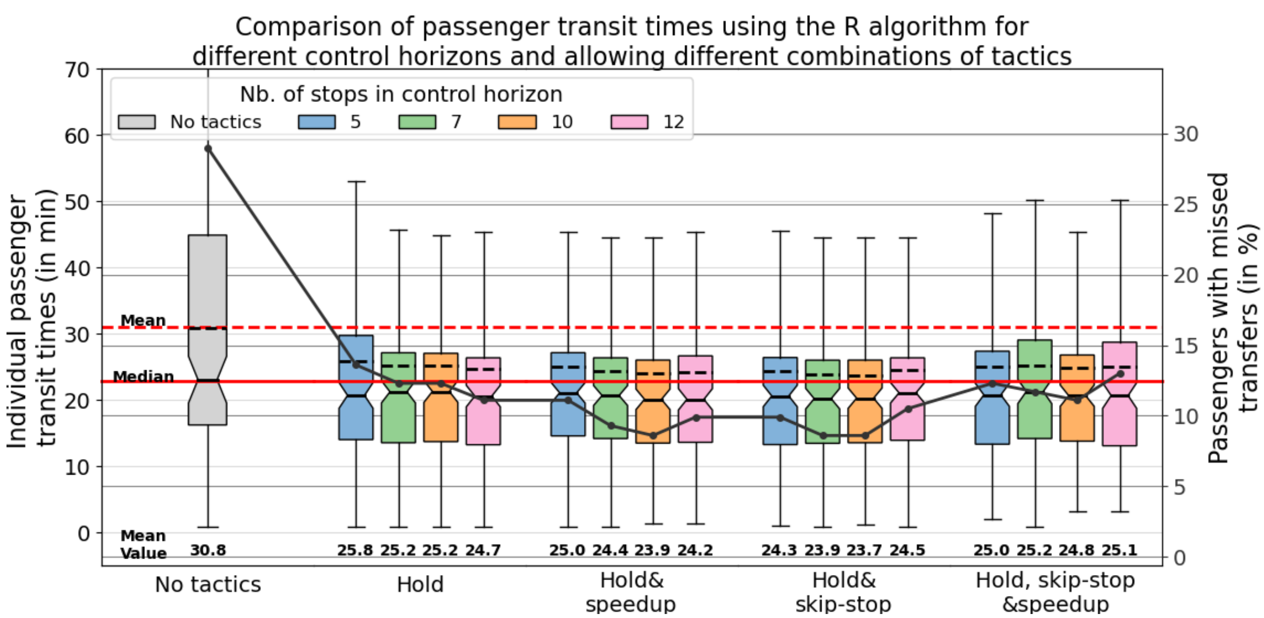

FIGURE 5.4 Total passenger travel times and percentage of passengers with missed transfers for different lengths of the control horizon for Algorithm R.

Figure 5.4 illustrates the performance of Algorithm R for various control horizon lengths and different sets of allowed tactics. When only the hold tactic is permitted, extending the control horizon reduces the number of missed transfers and improves total passenger travel time. A similar trend is observed when the speedup and/or skip-stop tactics are included, though these improvements plateau at a horizon of ten stops. For horizons of twelve stops, a slight increase in passenger travel times is observed compared to a ten-stop horizon. This is attributed to Algorithm R overusing the speedup and skip-stop tactics in an attempt to recover transfers at much later stops in the scenarios. Such overuse can lead to buses arriving at stops earlier than scheduled or before connecting buses, disrupting transfer synchronization. When the control horizon exceeds ten stops, the tactics applied at early stops anticipate transfers further down the horizon, which can lead to inefficiencies and overuse of certain tactics. Shorter control horizons provide more accurate arrival time approximations, as these are based on the current delay of vehicles and the generated travel times between stops, reducing the likelihood of over correction.

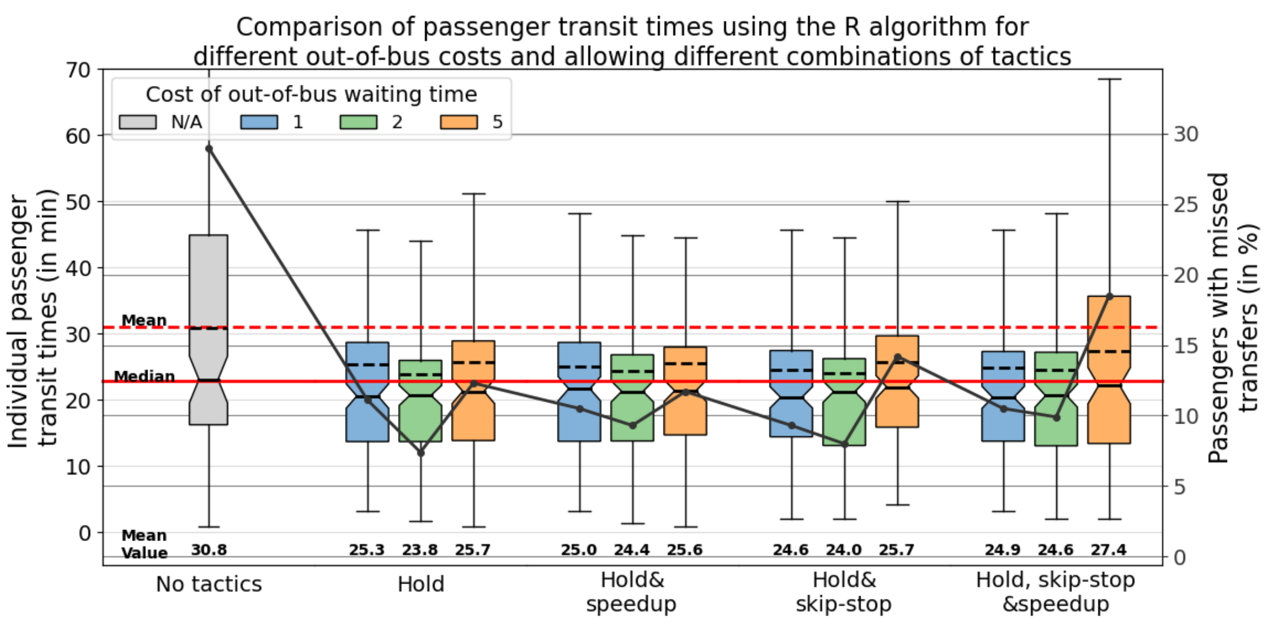

Figure 5.5 Total passenger travel times and percentage of passengers with missed transfers for different out-of-bus waiting time costs for Algorithm R.

Figure 5.5 illustrates passenger travel times and the percentage of passengers with missed transfers for different out-of-bus waiting time costs using Algorithm R. The tests are conducted with a control horizon of ten stops. An improvement in both travel times and successful transfers is observed when increasing the out-of-bus waiting time cost from one to two. In this range, the model effectively prioritizes minimizing out-of-bus waiting times (associated with transfer delays) over in-bus travel times, leading to better transfer synchronization and fewer missed transfers. However, for costs greater than two, Algorithm R begins to excessively prioritize minimizing out-of-bus waiting times in scenarios with a high number of anticipated transfers. This results in disproportionately large "regret" values for sub-optimal tactics. Conversely, scenarios with fewer expected transfers generate smaller "regret" values, introducing a bias in Algorithm R that overestimates transfer demand and promotes the overuse of control tactics. As a result, excessively high out-of-bus waiting time costs degrade both passenger travel times and the percentage of successful transfers achieved by Algorithm R.

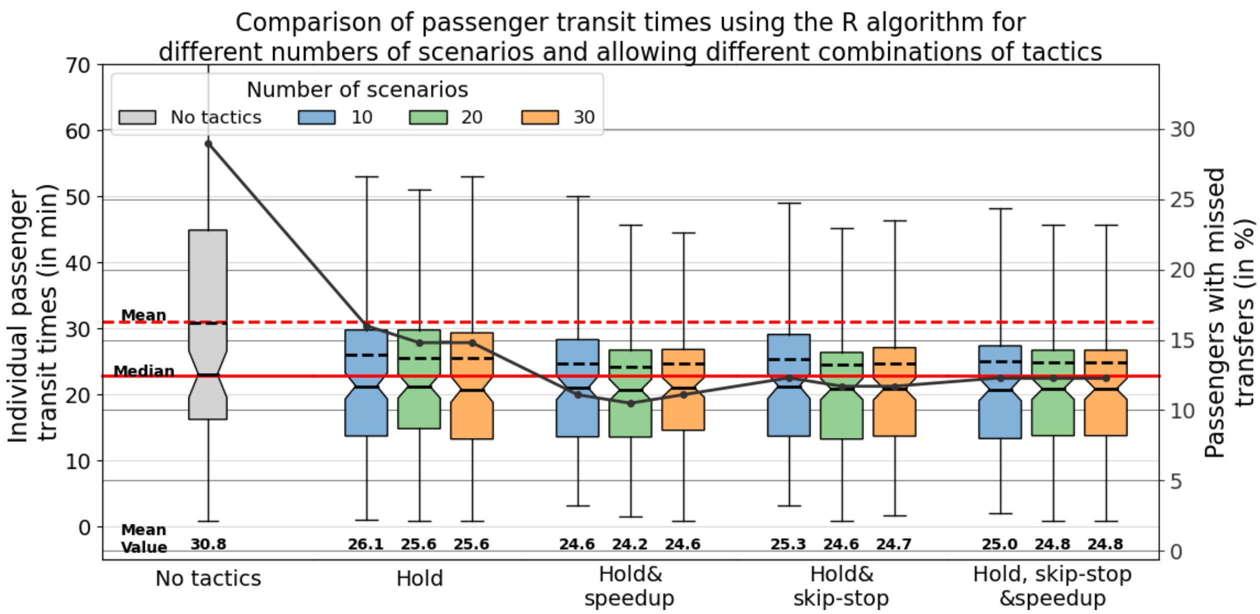

FIGURE 5.6 Total passenger travel times and percentage of passengers with missed transfers for different numbers of scenarios solved at each re-optimization with Algorithm R.

Figure 5.6 illustrates passenger travel times for varying numbers of scenarios solved during each re-optimization with Algorithm R for control horizon of ten stops. Increasing the number of scenarios leads to improvements in total passenger travel times and a reduction in the percentage of passengers with missed transfers. However, these benefits plateau between 20 and 30 scenarios, indicating diminishing returns beyond this range. It is important to note that for a given number of scenarios, the same set of scenarios is used across different combinations of tactics in both Algorithms C and R to ensure comparability of their performance.

Figure 5.7 presents passenger travel times when using a control horizon of ten stops and varying numbers of scenarios solved at each re-optimization with Algorithm C. As the number of scenarios increases, total passenger travel times and the percentage of passengers with missed transfers decrease. However, in the case where hold, skip-stop, and speedup tactics are allowed, Algorithm C performs worse than the case without any tactics. This is because a tac-

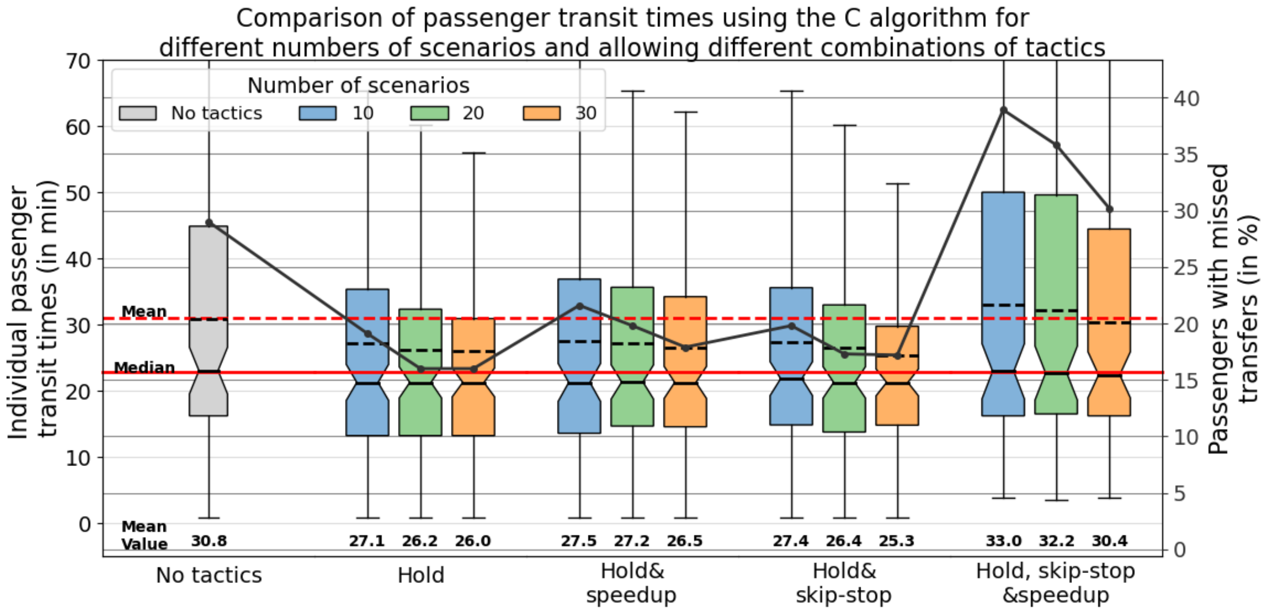

FIGURE 5.7 Total passenger travel times and percentage of passengers with missed transfers for different numbers of scenarios solved at each re-optimization with Algorithm C.

tic may be optimal for certain scenarios but can lead to significant time losses in others. The added complexity of coordinating all three tactics exacerbates this limitation in Algorithm C, potentially introducing inefficiencies in transfer synchronization.

We conclude the sensitivity analysis by selecting a horizon of fifteen stops for tests restricted to the hold tactic and a horizon of ten stops for tests involving speedup and/or skip-stop tactics. For all tests conducted with Algorithms C and R, twenty scenarios are used. Additionally, an out-of-bus cost of two is adopted for all subsequent numerical experiments. As shown in Figure 5.3, this combination of horizons and number of scenarios enables computation times per re-optimization that are generally under ten seconds, making it suitable for real-time bus operations. Furthermore, this configuration yields the greatest reduction in passenger travel time and the percentage of missed transfers across the sensitivity analyses.

5.5.3 Results

This section presents the results from experiments conducted on 29 lines, with detailed analyses focusing on two specific lines. Line 42 features high frequency, high occupancy, and significant transfer demand. In contrast, line 33 operates with medium frequency and average occupancy but includes multiple transfer stops, highlighting its role in enhancing network connectivity.

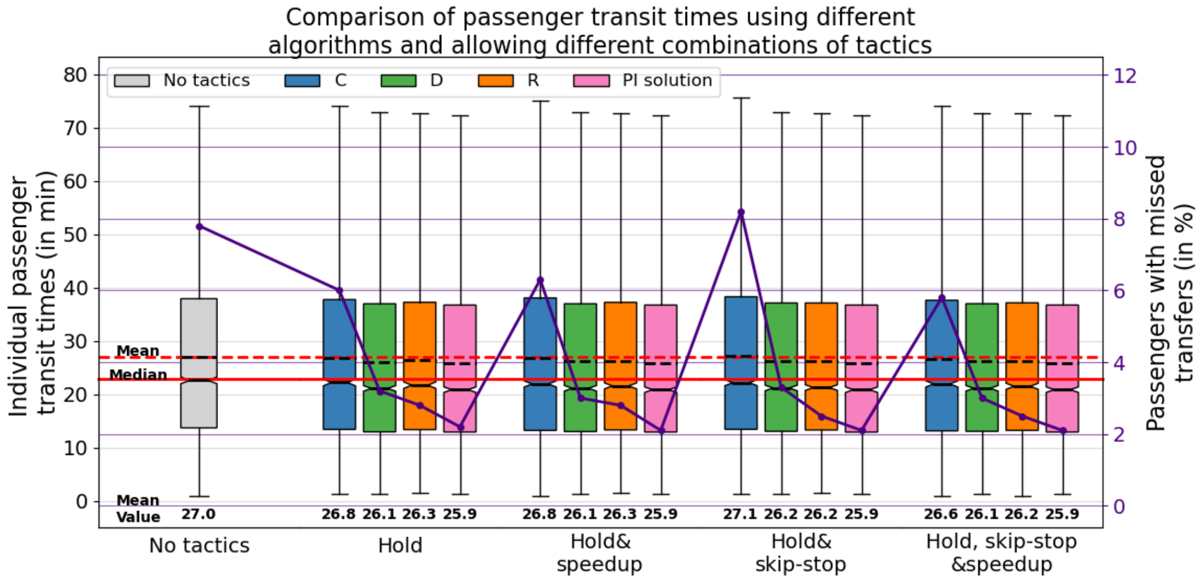

FIGURE 5.8 Total passenger travel times and percentage of passengers with missed transfers for different algorithms and tactics for line 42.

Note : The figure displays fifteen boxplots, each representing the transit times of individual passengers in minutes (left y-axis). Outliers, which account for 0.7 percent of passengers, are not shown for clarity. The right y-axis shows the percentage of passengers who missed their transfer, calculated for each boxplot. The first boxplot represents passenger travel times without any tactics, representing the observed real-world performance. The next four boxplots display passenger travel times when only the hold tactic is allowed. The following four boxplots display the case where both the speedup and hold tactics are allowed. The following boxplots illustrate the case where the hold and skip-stop tactics are allowed. The final group of boxplots represents the case where all three tactics are allowed. For each group of four boxplots, each box represents a different algorithm used to compute optimal tactics : the first box represents Algorithm C, followed by Algorithm D, Algorithm R, and the PI solution. The mean for each boxplot is displayed below it. A line indicating the median and mean of the "no tactics" case is drawn across all boxplots to facilitate comparisons across different cases.

Figure 5.8 presents detailed results from computations on three days of buses from Line 42, encompassing over 6,500 passengers and close to 2,000 transfers. The figure provides a comparison of passenger travel times and the percentage of missed transfers for different algorithms and combinations of tactics. Algorithm C exhibits the worst performance, even under performing the "no tactics" case when the hold and skip-stop tactics are applied. This poor performance arises from Algorithm C's inability to consider that a tactic optimal in certain scenarios may perform poorly in others. Consequently, the algorithm often selects tactics that seem advantageous in specific scenarios but result in significant costs when implemented in the real instance. This limitation is particularly pronounced with the skip-stop tactic, which can force boarding passengers to wait for the next bus, thereby increasing delays.

In contrast, Algorithm D consistently outperforms both the "no tactics" case and Algorithm C, achieving significant reductions in missed transfers without negatively impacting passenger travel times. For the hold tactic only, the slight increase in travel times due to holding buses is offset by the significant improvement in travel times for passengers who successfully make their transfer. Both the mean and median travel times for all passengers benefit from this trade-off. Similar trends are observed for the speedup and skip-stop tactics, with the greatest improvement in successful transfers occurring when all three tactics are allowed.

Algorithm R consistently delivers better results than the "no tactics" case and outperforms both Algorithms C and D in terms of transfer synchronization. When comparing Algorithms D and R in the hold tactic-only case, Algorithm R achieves greater improvements in the percentage of successful transfers. However, this comes at the cost of slightly longer passenger travel times, as Algorithm R applies the hold tactic more frequently. A similar observation is made for the hold with speedup tactics case. The percentage of successful transfers improves further when the hold tactic is combined with the skip-stop tactic. The skip-stop tactic proves more effective than the speedup tactic for stops with long dwell times. Moreover, Algorithm R's ability to consider uncertainty allows it to deploy the skip-stop tactic more effectively, enhancing its overall performance. Finally, when all three tactics are allowed, Algorithm R achieves a slightly higher percentage of missed transfers compared to the case with hold and skip-stop tactics. The added complexity of coordinating all tactics may diminish performance. Overall, Algorithm R's performance demonstrates the benefits of considering uncertainty by generating random scenarios compared to the deterministic approach of Algorithm D.

The PI solution attains the lowest passenger travel times and percentage of missed transfers, serving as a benchmark for assessing the performance of other algorithms. While the performance gap between Algorithm R and the PI solution varies across different tactics, it remains generally small, indicating that Algorithm R can effectively approximate the optimal outcomes achieved with complete system knowledge.

Figure 5.9 presents results from computations on three days of buses from line 33, considering around 1000 passengers who transferred close to 250 times. For a medium frequency line, each successful transfer saves more time compared to a high frequency line. Although the percentage of passengers with missed transfers for the "no tactics" case of line 33 is lower than for line 42, the performance difference between Algorithms D and R is more pronounced. Algorithm D struggles to compensate for poor decisions when only the hold tactic is allowed. Its performance improves when the speedup and/or skip-stop tactics are enabled, but it

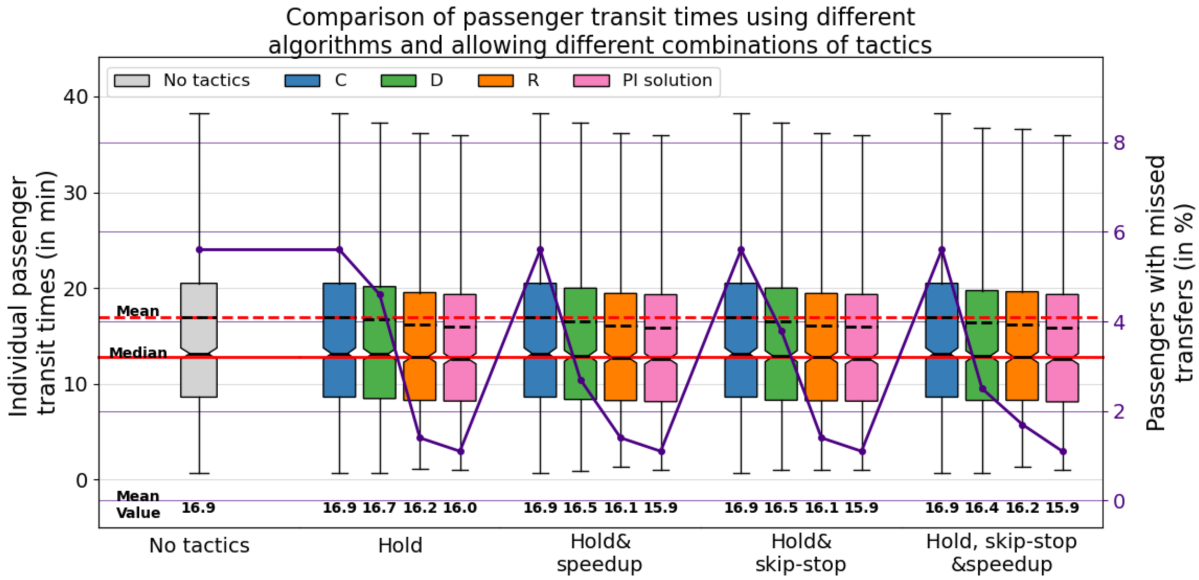

FIGURE 5.9 Total passenger travel times and percentage of passengers with missed transfers for different algorithms and tactics for line 33.

still falls short compared to Algorithm R. This highlights the advantage of modeling future conditions using sampling, as done by Algorithm R. Moreover, the results indicate that the models are particularly effective on medium- and lower-frequency lines or lines with fewer passengers. In these cases, a small number of well-applied tactics can significantly benefit transferring passengers while inconveniencing fewer passengers already onboard. Algorithm R achieves the best performance when both the hold and skip-stop tactics are allowed. However, when all three tactics are allowed, a slight degradation in results is observed due to the overuse of tactics.

TABLE 5.2 Mean percentage of missed transfers and mean change in total passenger travel time compared to the real case for 27 lines of the STL network.

Tactics	Average	Algorithm D	Algorithm R	PI Solution
Hold	Passengers with missed transfers	4.21%	3.47%	1.62%
	Reduction in total passenger travel time	1.53%	2.39%	4.21%
Hold& speedup	Passengers with missed transfers	4.81%	3.41%	1.50%
	Reduction in total passenger travel time	1.27%	2.89%	4.78%
Hold& skip-stop	Passengers with missed transfers	5.02%	3.35%	1.48%
	Reduction in total passenger travel time	0.50%	2.49%	4.31%

The average percentage of passengers with missed transfers for the **"no tactics"** case is **6.09%**.

Tests were conducted on 27 additional lines from the STL bus network, using the same methodology. The results are summarized in Table 5.5.3, with detailed findings presented in Figures B.1, B.2, and B.3 in the Appendix. Algorithm C is excluded from the analysis, as it did not demonstrate any improvement over the "no tactics" case. Algorithm R outperforms Algorithm D for all tactics combinations, reducing both the percentage of passengers with missed transfers and total passenger travel time. Moreover, Algorithm R mirrors the behaviour of the PI solution. They both achieve a lower percentage of missed transfers for the hold with skip-stop case compared to the hold with speedup case, at the cost of a smaller reduction in total passenger travel time. Algorithm R's performance is closest to that of the PI solution in the case with hold and skip-stop tactics.

5.6 Conclusion

Synchronizing transfers is a critical aspect of PT network planning, as transfer reliability directly influences both service quality and passenger perception of the network. This research introduces a novel approach to addressing the real-time transfer synchronization problem employing three control tactics : hold, speedup, and skip-stop. An arc-flow formulation is proposed to integrate combinations of all possible tactics while minimizing the number of variables in the offline problem. A discrete-event dynamic environment is developed to simulate real-time operations, enabling the evaluation of transfer synchronization strategies based on real-time data. Two OSO algorithms, C and R, are adapted to the transfer synchronization problem and tested within this dynamic environment using historical and real-time data. Algorithm C generates multiple scenarios at each iteration and selects the tactic that is optimal across the majority of scenarios. Algorithm R, in contrast, selects the tactic with the smallest cumulative "regret" value across all scenarios. Both algorithms achieve low computation times, making them suitable for real-time implementation. To benchmark their performance, Algorithm D and a PI solution are also implemented. Algorithm D generates a single scenario representing the average future state of the system during each re-optimization, while the PI solution assumes perfect foresight, providing an upper bound on performance.

The study tested the two OSO algorithms on 29 lines from a large-scale real dataset provided by the STL. The results were compared to the "no tactics" case, to Algorithm D and to the PI solution. Algorithm D showed improvement in passenger travel times and the number of successful transfers for any combination of tactics when compared to the "no tactics" case. Algorithm C's performance is hindered by its tactic selection process, which failed to adequately account for the variability in scenarios. Algorithm R outperformed both Algorithm C and Algorithm D, significantly increasing the number of successful transfers while reducing passenger travel times. Notably, Algorithm R's performance closely approximated that of the PI solution, demonstrating its potential for practical real-time applications.

The quality of scenario generation is a key factor influencing the performance of AlgorithmsC and R, as it significantly affects the accuracy of decision-making. Future research could investigate advanced scenario generation methods to enhance scenario quality while using a smaller number of scenarios, thereby improving efficiency and further reducing computation times. Additionally, this study, which primarily examines individual main lines linked to fixed feeder lines, could be extended to explore the interactions and coordinated optimization of multiple main lines within a PT network.

CHAPITRE 6 ARTICLE 3 : NETWORK-WIDE TRANSFER SYNCHRONIZATION STRATEGIES IN A PUBLIC BUS SYSTEM WITH REAL-TIME AVL AND SMART CARD DATA

L. Kolcheva, A. Legrain et M. Trépanier ont écrit cet article et l'ont soumis le 24 avril 2025 dans *Public Transport*.

6.1 Introduction

Public transit (PT) systems play a critical role in reducing traffic congestion and promoting sustainable urban transport. They provide a cost-effective, environmentally friendly alternative to private car use, fostering more equitable access to mobility. Within this context, transfer reliability between transit lines is critical to passenger satisfaction and overall system effectiveness. Reliable transfers not only minimize travel times but also enhance the overall appeal of PT, encouraging higher ridership and contributing to the system's long-term viability. In urban settings, unpredictable traffic patterns, variable passenger demand, and a myriad of operational uncertainties often disrupt planned schedules, resulting in missed transfers, prolonged waiting times, and diminished passenger confidence in the system. These issues are particularly acute in multi-line transit networks, where coordinating transfers becomes increasingly complex under stochastic operating conditions.

Transfer synchronization was initially addressed during the network planning phase, relying on static schedules and historical data. While this provides a solid foundation, even well-designed schedules are vulnerable to real-world variability and disruptions. Real-time synchronization therefore emerges as a necessary complement, enabling dynamic adjustments during operations. Although schedule adjustments and network redesigns can improve performance, they are often resource-intensive and logistically complex. This highlights the need for innovative, cost-efficient strategies to enhance network performance in real time. Advances in data collection technologies, such as Automatic Fare Collection (AFC), Automatic Passenger Counting (APC), and Automatic Vehicle Location (AVL) systems, generate a wealth of real-time data, enabling dynamic predictions of vehicle and passenger movements. By leveraging these insights, transit agencies can implement transfer synchronization strategies that address disruptions and improve network efficiency in real time.

Recent work has addressed real-time transfer synchronization using an offline arc-flow model and integrating it into two online stochastic optimization (OSO) algorithms. This approach applied synchronization tactics — *hold*, *skip-stop*, and *speedup* — on a single main line at a time while accounting for transfer impacts from all connecting feeder lines within a predefined control horizon. Tested under stochastic conditions on 29 bus lines from the transit network of Laval, the framework demonstrated significant improvements in both transfer reliability and passenger travel times.

Building on this foundation, the present work extends the methodology to apply tactics across multiple interconnected transit lines. This presents new challenges, including increased computational complexity and the necessity for seamless real-time data integration. The existing arc-flow model and the OSO framework are integrated into a network-wide simulator (NWS) of Laval's bus network, enabling real-time optimization in a complex urban environment. The simulator incorporates historical vehicle movement and smart-card data to replicate real-time stochastic conditions, providing a practical and scalable framework for evaluating real-time synchronization tactics across an entire network.

This paper makes the following key contributions :

- it integrates the offline arc-flow model and the OSO framework into a NWS for bus transit networks ;
- it leverages historical and real-time data to enable robust simulations under stochastic bus operations ;
- it validates the approach using real data from a mid-size bus network, demonstrating significant improvements in transfer reliability and passenger travel times across multiple lines.

6.2 LITERATURE REVIEW

Transfer synchronization has been extensively studied in the PT literature. Early works focused on offline strategies, such as network design and timetable coordination, to improve planned transfers. More recent studies have addressed real-time approaches, leveraging advances in data availability and computational power. [17] provide an extensive overview of real-time transfer synchronization tactics and their applications.

6.2.1 Control tactics

This literature review examines three primary control tactics : *hold*, *skip-stop*, and *speed control*. These tactics can be applied at stops called control points along transit lines. *Hold*, which delays a vehicle at a stop for a predetermined duration, is widely regarded as the most effective and straightforward to implement [14]. However, its application involves complex optimization challenges, with the number of variables increasing proportionally to the product of the number of targeted stops and vehicles, often limiting studies to a single line or a small set of stops [18]. *Skip-stop* reduces travel times for onboard passengers by skipping one or more consecutive stops, but may inconvenience passengers needing to board or alight at those stops. Lastly, *speed control* or *speedup* is a simple yet effective tactic for reducing delays, maintaining service regularity, and improving transfer reliability. These strategies are applied either independently [23] or in combination [19–21] to enhance transit performance. [24] highlight that designating all stops as control points facilitates disruption management through distributed control throughout the entire line.

6.2.2 Data integration

Before the deployment of intelligent transportation systems, such as AVL and APC, real-time control strategies relied on decisions made by strategically positioned personnel [28]. Historically, passenger flow data — such as vehicle occupancy, at stop arrivals, and transfer demand — were derived from empirical observations. Today, real-time insights are made possible by technologies like APC systems, smart card validations, and advanced fare collection systems. When passenger destinations are not directly recorded, they can be inferred using sophisticated algorithms [31]. [32] conclude that real-time demand information significantly improves transfer synchronization, particularly in low-frequency networks. Recent studies have further emphasized the value of real-world passenger data over historical averages or simulations. For instance, [33] evaluated three levels of passenger information for optimizing transfer synchronization and service regularity : no real-time data, real-time vehicle occupancy, and the integration of smart card validations with real-time occupancy data. Using the BusMezzo simulation model, they tested these scenarios on two lines with one transfer stop and five control stops in The Hague, Netherlands. The results showed that access to smart card data significantly improves transfer synchronization. Nevertheless, many studies rely on the assumption that vehicle and/or passenger demand data are fully available and accurate [34, 35].

6.2.3 Dynamic, real-time and stochastic optimization

In transfer synchronization, the state of the PT network evolves dynamically, with vehicle travel times and onboard loads updated as vehicles reach stops. Control tactics must be re-optimized dynamically based on the latest information, while decisions must be made within tight time frames. This dual challenge requires algorithms that combine the flexibility of dynamic optimization with the speed of real-time optimization to address evolving network conditions and the urgency of operational constraints.

[36] developed a dynamic programming model combining *hold* and *skip-stop* tactics to minimize total passenger travel time, including transfer waiting time. Their model was tested on a small artificial network with two buses and three stops. Building on this, [37] developed a polynomial-time dynamic model applied to a network featuring a main line, three feeder lines, and three transfer segments, using simulated travel time and passenger demand data. Although further refined with the addition of a short-turn tactic [38], [19] argued that their static model, solved with a commercial solver, was simpler to implement than the dynamic programming approach.

Each update of the transit network state provides real-time data that captures current conditions while also enabling predictions of future network states through techniques like sampling and scenario generation. Effective decision-making relies on dynamically adapting to these predictions, with uncertainty modeling playing a critical role in delivering practical and reliable outcomes [41]. In this context, [40] proposed a discrete-event simulation with a rolling prediction horizon, applying *hold* and *speed control* to minimize total passenger travel time. Their approach, tested on Haifa's Métronit network with three lines and three transfer stops using the BusMezzo simulator, relied on a single forecast scenario, limiting its ability to account for system variability. In contrast, [76, 77] employed two OSO algorithms generating multiple random scenarios to represent potential future states of the public transit network. By integrating these scenarios into a unified decision-making process, the methodology accounted for variability in travel times and passenger demand.

The remainder of this paper is organized as follows. Section 6.3 outlines the methodology for embedding the optimization model into a NWS. Section 6.4 details the data sources and their integration into the proposed framework. Section 6.5 describes our case study and discusses results from applying the approach to Laval's transit network. Section 6.6 concludes with a summary of findings and directions for future research.

6.3 Methodology

The proposed methodology builds upon an arc-flow model and an OSO framework introduced in previous work. While the original approach focused on applying transfer synchronization tactics to a single main line connected to multiple feeder lines, this study implements real-time control tactics on multiple interconnected main lines.

The methodology, depicted in Figure 6.1, integrates three key components. First, an offline arc-flow model integrating all control tactics for a given control horizon. Second, the

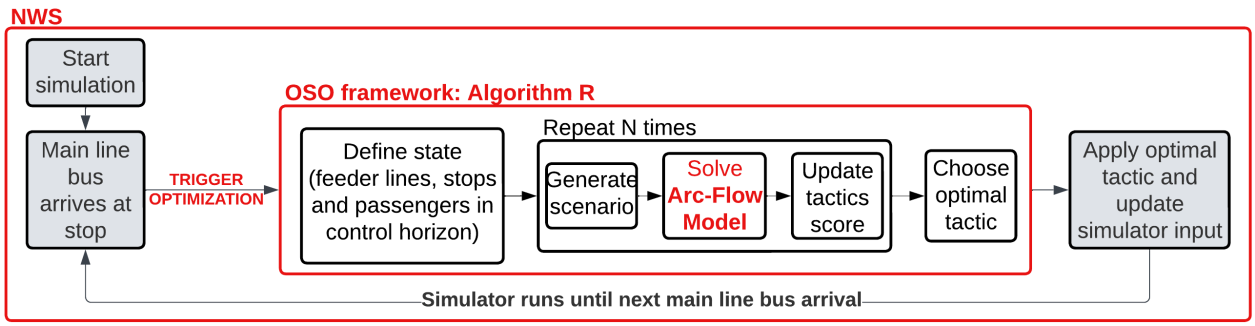

FIGURE 6.1 Overview of the optimization process within the NWS (N : number of stochastic scenarios evaluated by Algorithm R).

Algorithm *REGRET* (R) is implemented in real time in an OSO framework, building on the offline model to solve multiple scenarios and support decision-making under stochastic operating conditions. Third, the OSO framework is implemented into a NWS to test the methodology's scalability and effectiveness on the transit network of the city of Laval. Together, these components provide a robust framework for real-time transfer synchronization aiming to minimize passenger travel times across complex, multi-line systems.

6.3.1 Offline arc-flow model for transfer synchronization

The arc-flow model forms the backbone of the methodology, capturing the temporal and spatial dynamics of passenger flows and bus movements. Specifically designed to minimize passenger travel times, the model enables efficient transfers through the systematic application of control tactics.

The model focuses on a main line and its connecting feeder lines within a defined control horizon. Feeder lines are modeled as exogenous inputs with fixed arrival times at transfer points, while main lines are subject to control tactics. The model represents the transit network as a time-expanded graph, where nodes correspond to specific events, such as passenger and bus arrivals/departures at stops, and arcs represent possible movements, including passenger entering/exiting the system, waiting, or transferring. Each arc is weighted by the associated travel or waiting time, ensuring all segments of passenger trips are explicitly modeled. This detailed representation enables the model to precisely evaluate the effects of control tactics on passenger travel times.

Three control tactics are automatically and systematically incorporated into the arc-flow model to improve transfer reliability and reduce disruptions for passengers. The *hold* tactic adds arcs between existing nodes at stops to allow buses to wait for delayed transfer passengers, ensuring connections that would otherwise be missed. The *skip-stop* tactic introduces bypass arcs, enabling buses to skip stops with low demand to recover schedules or enable transfers at later points. The *speedup* tactic implements faster travel arcs to simulate reduced travel times under favorable conditions, such as light traffic or operational slack, helping buses catch up to their planned schedules or transfers.

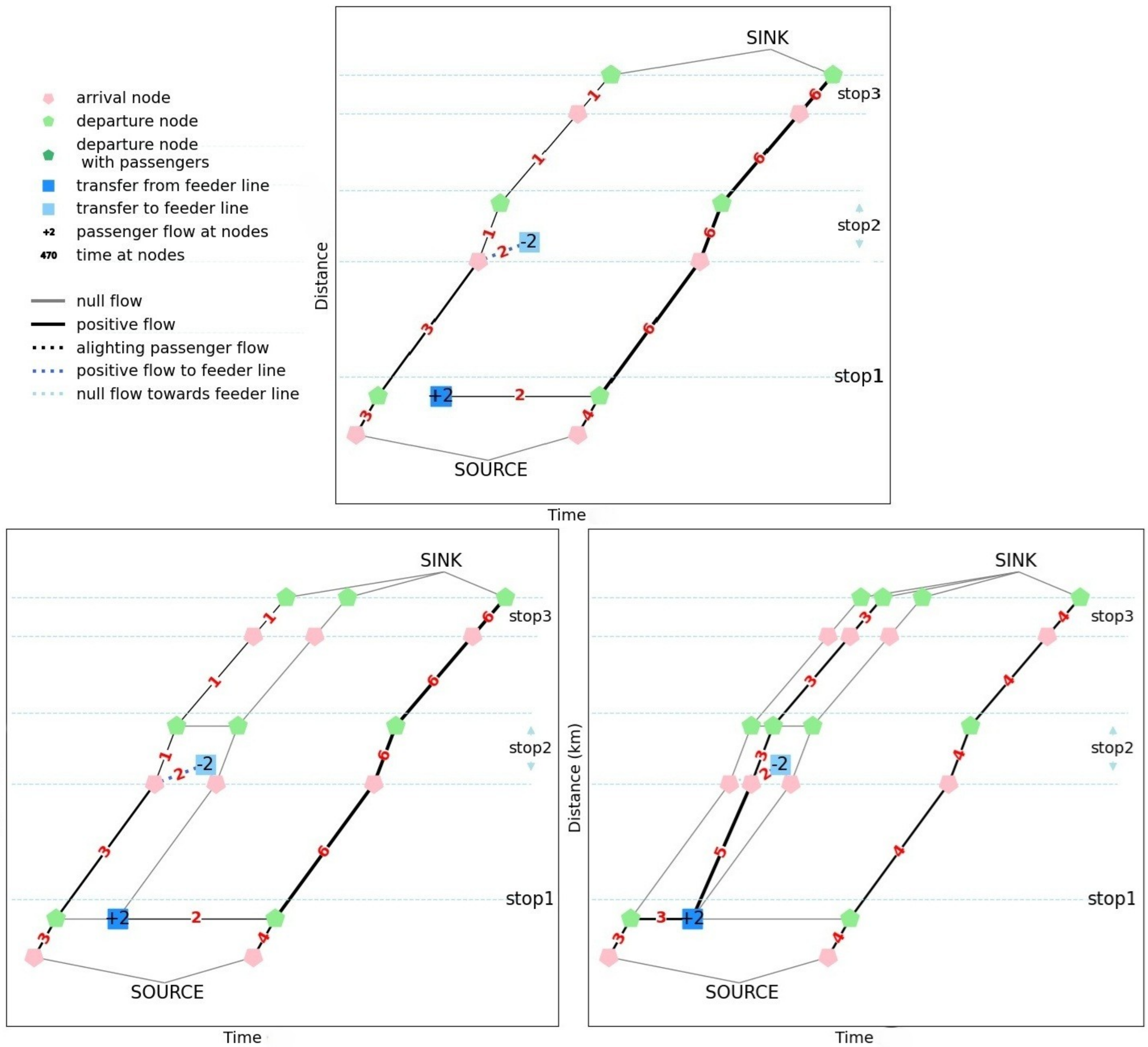

FIGURE 6.2 Example of an arc-flow model graph of a small instance. Up - no tactics. Lower left - *hold* only. Lower right - *hold* and *speedup*.

Figure 6.2 illustrates the graph construction process, highlighting how control tactics create additional bus paths and introduce new transfer opportunities. Non-zero passenger counts are displayed on the arcs for clarity, while arrival and departure nodes are slightly offset vertically to improve readability. Bold black arcs represent the optimal path returned by the arc-flow model, while gray arcs depict alternative paths. Exogenous passenger flows (e.g., passenger arrivals or transfers) are shown on the nodes. The example includes two main line vehicles and three stops within the control horizon. In the scenario without control tactics (top), two transferring passengers miss the first bus at "stop 1" and must wait for the next one. At "stop 2," two passengers alight from the main line to transfer to a feeder line. In the case where only the *hold* tactic is permitted (bottom left), additional paths are introduced to allow *hold* time at the first stop. However, if the bus takes this path, the transfer at "stop 2" becomes impossible, and the optimal path excludes the *hold* tactic. In contrast, when both *hold* and *speedup* tactics are allowed (bottom right), new paths account for both strategies. This combination allows for a *hold* time at "stop 1" and a *speedup* tactic towards "stop 2," ensuring successful transfers at both stops. The optimal path utilizes these two tactics together.

Rather than applying the arc-flow model directly to the entire network, it is used to solve multiple localized problems. This modular approach ensures scalability without compromising accuracy, enabling the integration of the arc-flow model into the OSO framework and the NWS.

6.3.2 Online stochastic optimization (OSO) framework and REGRET (R) algorithm

The OSO framework is central to the proposed methodology, enabling the implementation of the R algorithm and facilitating real-time decision-making under uncertainty. Operating within a rolling control horizon, the framework re-optimizes decisions each time a main line bus arrives at a stop, remaining responsive to real-time data as it becomes available.

While the offline model assumes perfect knowledge of the system and does not adapt to changing network conditions, Algorithm R extends this foundation by dynamically applying control tactics — *hold*, *skip-stop*, and *speedup*. To address uncertainty, the OSO framework employs sampling to generate scenarios representing multiple possible realizations of future system states by combining historical and real-time data. Each scenario is used to optimize transfer synchronization within the control horizon through the arc-flow model, yielding an optimal tactic. Algorithm R then quantifies the potential 'regret' of suboptimal tactics by evaluating their costs across all scenarios. By selecting the tactic with the lowest cumulative regret, the algorithm ensures robust performance across diverse conditions, including variability in travel times, passenger demand, and transfer bus delays. To benchmark the performance of Algorithm R, two additional approaches are used. The DETERMINISTIC (D) algorithm generates a single scenario based on average historical values for all relevant variables. This approach represents a simplified, static view of the system, providing a baseline for evaluating the benefits of incorporating stochastic sampling. In contrast, the PERFECT INFORMATION (PI) solution assumes complete knowledge of the true future state across all stops within the control horizon. This approach enables decision-making under ideal conditions, setting a theoretical upper bound on Algorithm R's performance.

6.3.3 Network-wide simulator (NWS)

This study extends the arc-flow model and OSO algorithms from the authors' previous work by extending their application to a network-wide scale by integrating them into the open-source NWS. This flexible platform enables detailed discrete-event simulations of multimodal transportation systems.

The NWS models transportation systems by simulating events involving two key agents : passengers and vehicles. Passenger trips are reconstructed using smart card data, ensuring that the simulation captures real-world travel patterns. Passenger trips are defined by origins, destinations, intermediate stops, and temporal constraints. Vehicles operate along predefined routes consisting of past, current, and future stops, while events such as passenger boarding, alighting, and vehicle movements are processed sequentially. This event-driven approach allows for dynamic interactions between passengers and vehicles across the network and enables a realistic evaluation of the impact of control tactics on transfer reliability and passenger travel times.

To incorporate stochasticity and enable real-time decision-making, the OSO framework is implemented as an additional module within the NWS. This module allows for the designation of multiple (or all) lines in the network as main lines, providing flexibility in defining the scope of optimization. Each time a main line bus arrives at a stop, an optimization event is triggered. The OSO framework then defines the system state within a rolling optimization horizon and applies tactics to the current main line while considering all other lines as fixed for that step. It generates multiple scenarios based on historical and real-time data and solves localized arc-flow problems to determine the optimal control tactics — *hold*, *skip-stop*, and *speedup* — for the relevant main line.

Once an optimal tactic is selected, the state of the simulation in the NWS is updated and future stops, dwell times, and travel times for the optimized vehicle are adjusted accordingly. Onboard passengers affected by a skipped stop are reassigned to alight at the nearest available stop, with an additional walking time to their final destination. Similarly, passengers wishing to board at a skipped stop are rerouted to the next available bus. The *hold* tactic modifies passenger travel times and possibly transfer feasibility. Once the system is updated, the simulation proceeds until a vehicle of any of the designated main lines arrives at a stop, triggering a new re-optimization step.

This sequential and iterative optimization approach ensures that the methodology captures interdependencies between lines while maintaining computational efficiency. The method adapts dynamically to evolving network conditions, and takes into account the current positions of all vehicles and passengers. Although optimizations are applied to individual lines and vehicles, frequent re-optimizations — potentially occurring every few seconds — allow different lines to indirectly coordinate through successive decision updates.

The NWS, integrated with the OSO framework, offers a robust and flexible platform for validating the proposed methodology. Its modular design and computational efficiency enable network-wide transfer synchronization while ensuring a detailed evaluation of the effects of control tactics under realistic stochastic conditions.

6.4 Data processing

This section details the data sources used in the study, explains their integration into the NWS, outlines their role within the OSO framework, and specifies the inputs for the arc-flow model. All data were provided by the *Société de Transport de Laval* (STL), a city of 450,000 inhabitants in Canada. Figure 6.3 illustrates the data integration.

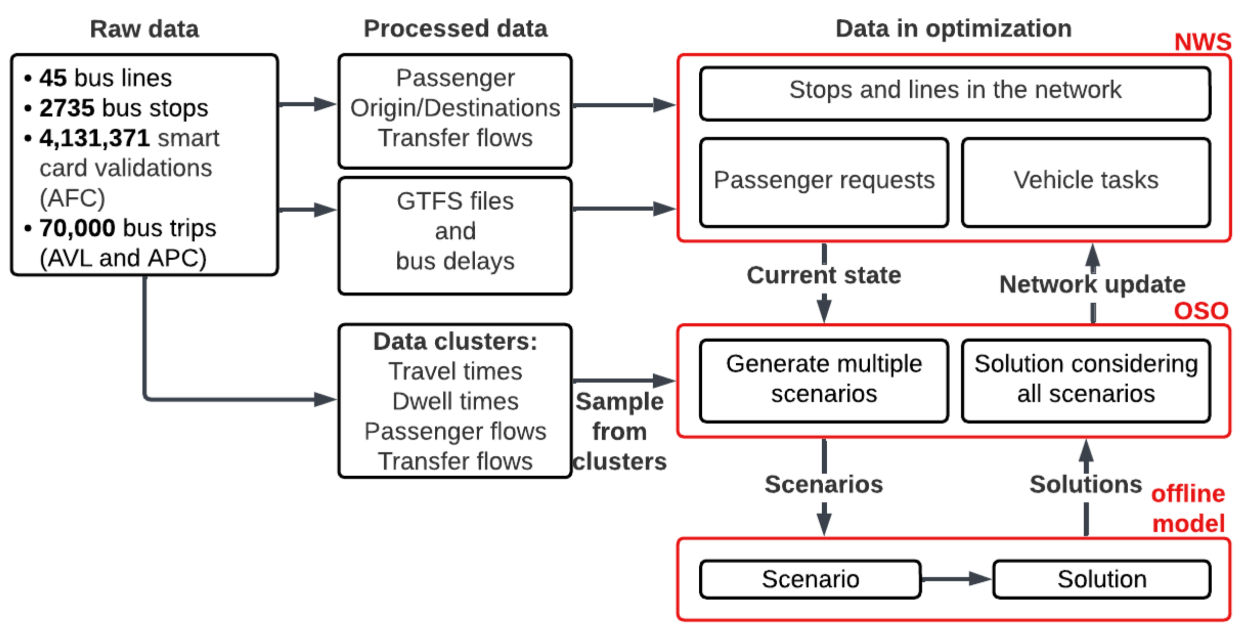

Figure 6.3 Data pre-processing and integration in the NWS, OSO framework and arc-flow model.

6.4.1 Input data

This study uses multiple datasets to accurately model and evaluate transit operations within the NWS :

- General Transit Feed Specification (GTFS) data provides static schedule information, including planned routes, stops, and timetables. This static dataset serves as the foundation for modeling the transit network, enabling the reconstruction of historical operations and the evaluation of schedule deviations.
- AVL data, collected via Global Positioning System (GPS) devices installed on buses, offers detailed information on vehicle trajectories, such as locations, speeds, and adherence to planned schedules.
- APC data records passenger boardings and alightings at each stop along a route.
- Anonymized AFC records, derived from smart-card transactions, provide precise timestamps for boardings, transfers, and inferred destinations, as determined by STL's proprietary algorithms.

These datasets collectively enable the reconstruction of complete passenger journeys and the estimation of transfer demands across the network.

6.4.2 Data integration in the NWS

The NWS reconstructs historical transit operations by integrating data from AVL, APC, AFC, and GTFS sources. Passenger interactions are modeled as trip "requests" derived from smart-card data. Each request comprises sequential legs corresponding to the onboard segments of passenger trips, with transferring passengers having multiple legs. The NWS reconstructs these trips by linking passengers to the specific vehicles they boarded. Boarding and alighting events are processed dynamically, capturing real-time interactions between passengers and vehicles. To ensure high fidelity, the NWS adjusts vehicle operations using historical AVL data, which accounts for variations in travel and dwell times, while GTFS data maintains alignment with planned schedules. Although APC data are not directly integrated into the simulation, it is employed to validate reconstructed passenger flows and transfer patterns, ensuring their accuracy.

The NWS interacts dynamically with the OSO framework, which applies control tactics and modifies input data in real-time. After these adjustments, the NWS updates bus travel times and reevaluates passenger trips to reflect the impact of these changes. For instance, passengers who miss their originally scheduled bus due to delays are automatically reassigned to the next available vehicle on the same route. Similarly, when control tactics improve transfer synchronization, passengers may board earlier vehicles on the same route. This clear separation of responsibilities ensures that the NWS provides a stable and accurate foundation, supporting the stochastic optimization processes performed in the OSO framework.

6.4.3 Data integration in the OSO framework and offline arc-flow model

When an optimization event is triggered, the OSO framework takes into account available real-time data and generates scenarios representing potential future system states, which are then used as inputs for the offline arc-flow model. Unlike the NWS, which replicates historical operations based solely on input data, the OSO framework dynamically adjusts this data through control tactics.

When generating a scenario, the OSO framework distinguishes between known and unknown variables. Firstly, historical AVL data is clustered by time of day, such as peak and off-peak hours, to create realistic distributions for bus travel and dwell times. Similarly, historical smart-card data is aggregated and clustered to provide detailed information on passenger demand, categorized by time, route, and boarding/alighting stop. Unknown variables — such as future travel times, dwell times, passenger arrivals at stops, and transfer intentions — are then stochastically sampled from the clustered datasets. Secondly, known variables, such as current bus positions obtained from real-time AVL data and vehicle occupancy derived from real-time APC and smart-card records, are directly integrated into the scenarios. These scenarios are then provided as inputs to the offline arc-flow model, which operates under the assumption of complete knowledge of the generated system state.

6.5 Experiments

This section evaluates the performance of the proposed methodology by assessing the impact of transfer synchronization control tactics on passenger travel and transfer times. The methodology is applied to the full STL network as well as to several sub-network configurations, with the aim of testing its effectiveness across diverse operating environments. The goal is to derive practical insights and establish clear guidelines on the conditions under which these synchronization strategies are most effective. The evaluation is based on a case study of weekday evening peak hours, covering a four-hour period with 425 bus trips, and 5,000 passengers. The following sections describe each network configuration and present the results of the methodology.

6.5.1 Test cases

Grid sub-network

A sub-network with a grid configuration was designed to evaluate the effectiveness of the proposed method in scenarios where pairs of intersecting lines share only one or a few common transfer stops, while each line connects with multiple others. This type of network is a classic and widely studied configuration in public transit planning, and a predominant form in many North American cities. Grid networks — such as those in Los Angeles and Vancouver — minimize route overlap and allow passengers to travel between most origin - destination pairs with a single transfer. In such settings, transfers are distributed across the network, requiring synchronization at multiple points along the lines.

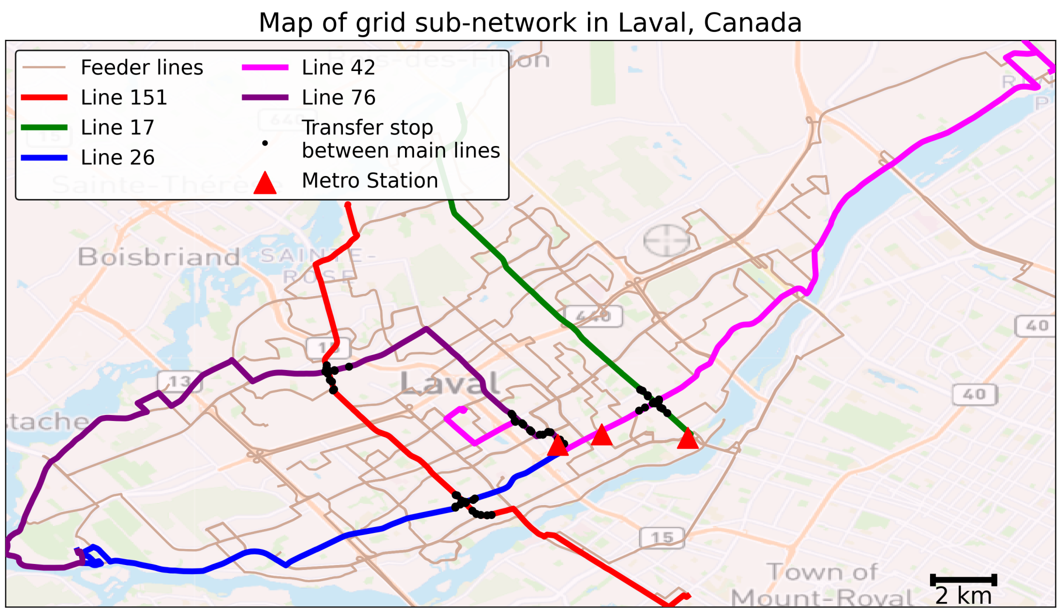

FIGURE 6.4 Map of lines in grid sub-network.

This instance focuses on high-frequency and some medium-frequency lines that structure the STL network and carry significant transfer demand. It includes 2,500 passengers with five main lines (17, 26, 42, 76, and 151) where control tactics are applied. The remaining routes function as feeder lines. It is important to note that the transfer points between these lines are the most demand intensive in the STL network. Figure 6.4 presents the grid sub-network test instance.

Radial sub-network

A radial sub-network is considered to reflect the structure of several STL bus routes converging toward the three metro stations that connect Laval to Montréal. This layout follows the classical radial pattern found in many metropolitan transit systems, where services are designed to direct passenger flows toward central hubs — typically downtown areas or major interchanges. In the STL context, the metro stations serve as these focal points, making this instance a relevant and realistic setting for evaluating synchronization strategies in radial networks.

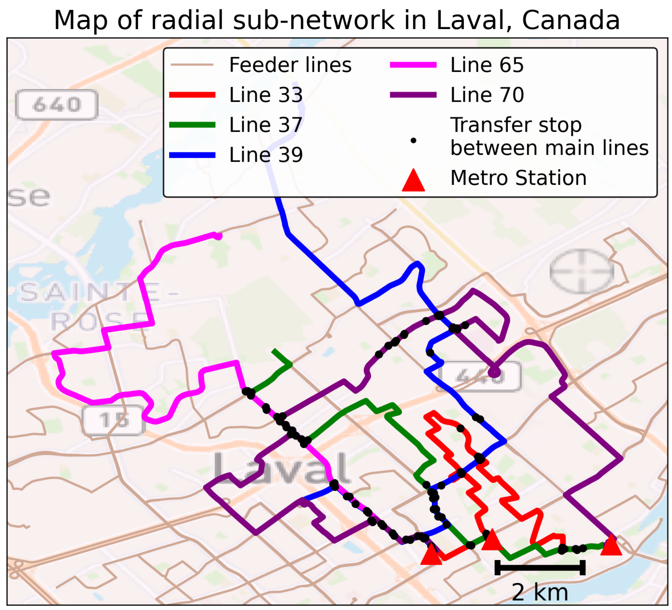

FIGURE 6.5 Map of lines in radial sub-network.

The radial sub-network, shown in Figure 6.5, includes two lines (33 and 70) forming concentric loops — one smaller, one larger — both beginning and ending at metro stations. Three additional lines (37, 39 and 65) pass through the metro stations and extend beyond these hubs. Synchronization in this setting occurs at multiple transfer points, where the looping lines intersect with the radial lines at several locations along their trajectories. This instance serves a dual purpose : to evaluate the effectiveness of synchronization strategies in a highly interconnected environment, and to assess their performance within a radial network structure. It comprises 1,000 passengers and 194 bus trips, with five main lines where control tactics are applied. The remaining routes function as feeder lines.

Corridor sub-network

In the current STL network, multiple lines converge toward the three previously mentioned metro stations, forming several corridors with overlapping service and varying headways. This layout reflects a common evolution in transit systems : while many networks originate as radial, they often transition toward grid-like structures, with the grid becoming distorted near central areas to maintain strong connectivity to key hubs. As observed in cities like Portland, this results in parallel routes converging toward the core, while others complete the grid without entering it. The corridor sub-network instance focuses on synchronizing transfers along these overlapping lines, providing a relevant context for evaluating strategies in hybrid networks that combine grid and radial characteristics.

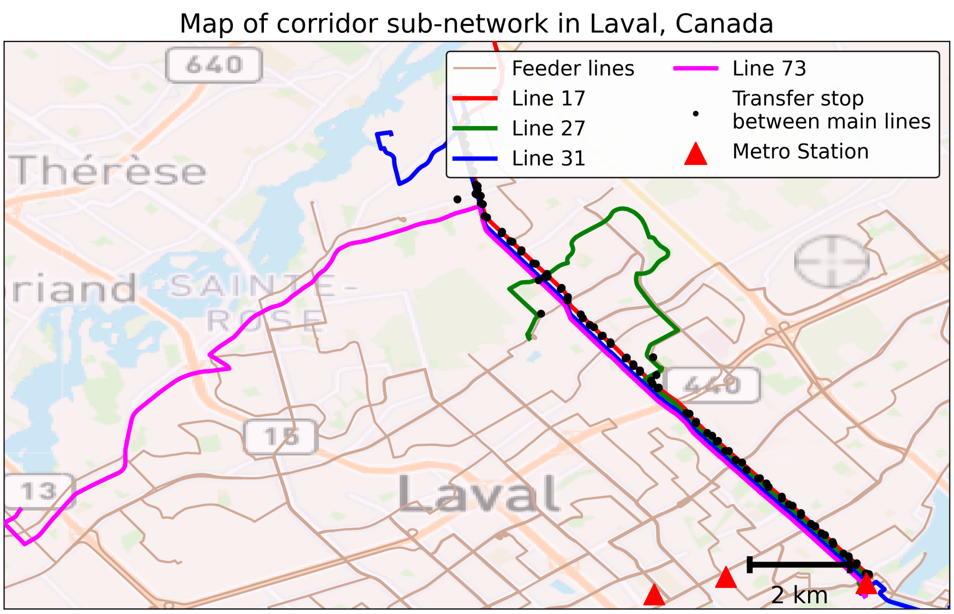

FIGURE 6.6 Map of lines in corridor sub-network.

A distinctive feature of this instance is the demand pattern : many users board the first available bus along a corridor and transfer as needed along shared segments or towards feeder lines to reach their destination. This instance includes 850 passengers and 150 bus trips, with lines 17, 27, 31, and 73 designated as main lines. Figure 6.6 presents the corridor instance, with lines slightly offset to improve readability.

Full network

This instance serves as a baseline for evaluating the performance of our methodology, where synchronization tactics are applied sequentially to almost all lines. The selected main lines

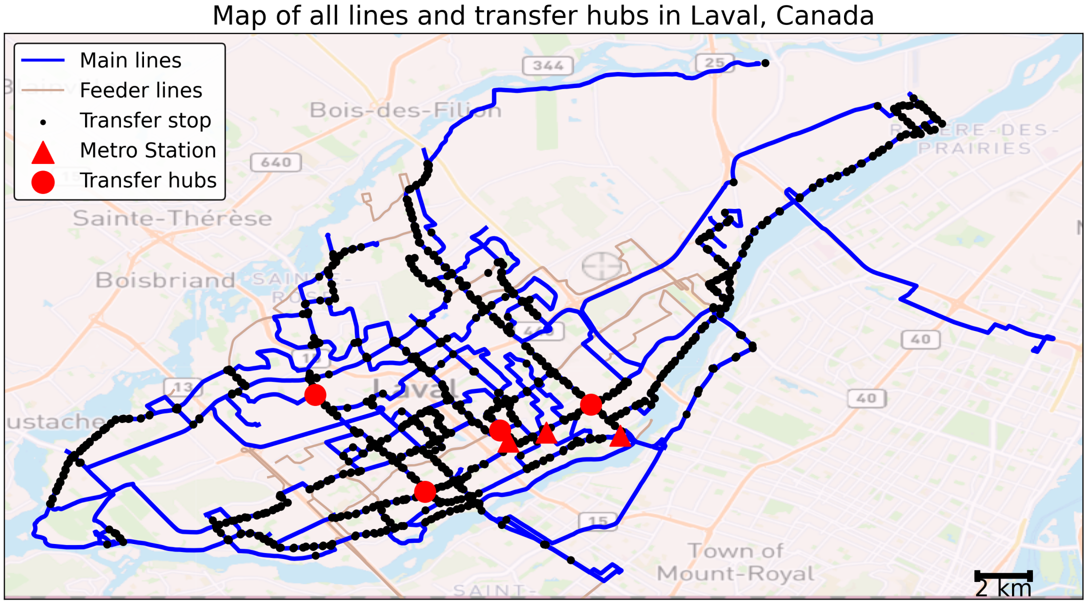

FIGURE 6.7 All lines and transfer hubs map.

are shown in Figure 6.7. As illustrated on the map, several corridors feature overlapping lines that share the same segments. While the overlap may obscure the number of lines, the density of black dots — indicating transfer stops — reveals areas with frequent inter-line transfers. Treating all lines as main lines introduces significant computational challenges. During peak periods, bus arrivals can occur every few seconds, and each re-optimization step — depending on the number of feeder lines, transfer stops, and passengers — can take between 0.1 and 10 seconds. As the process is executed sequentially without parallelization, total computation times become prohibitive for real-time application. This issue did not arise in the previous instances, which involved fewer lines and more localized synchronization scopes.

Transfer hubs

Building on the previous instance, where synchronization tactics were applied network-wide, this instance retains the same set of main lines but restricts the application of control tactics to the vicinity of selected transfer hubs. These transfer hubs account for over 25% of all transfers in the STL network during the study period. Tactics are applied only to buses operating within a limited number of stops from these designated transfer hubs, with no synchronization outside them. This setup allows us to assess the trade-off between comprehensive, system-wide optimization and more localized interventions focused on major transfer points. It also reflects a practical application scenario in which real-time dispatching efforts are geographically bounded to reduce operational and computational complexity. All lines used in this instance, as well as in the previous one, are shown in Figure 6.7.

6.5.2 Results

This section presents the results for all test cases. Detailed analyses of passenger travel times, transfer times, and travel time variations, as well as a comparison between single-line and multi-line optimization, are provided for the grid sub-network instance, as it reflects the predominant structure of the STL network. Summary results for all other test instances are presented in Table 6.1.

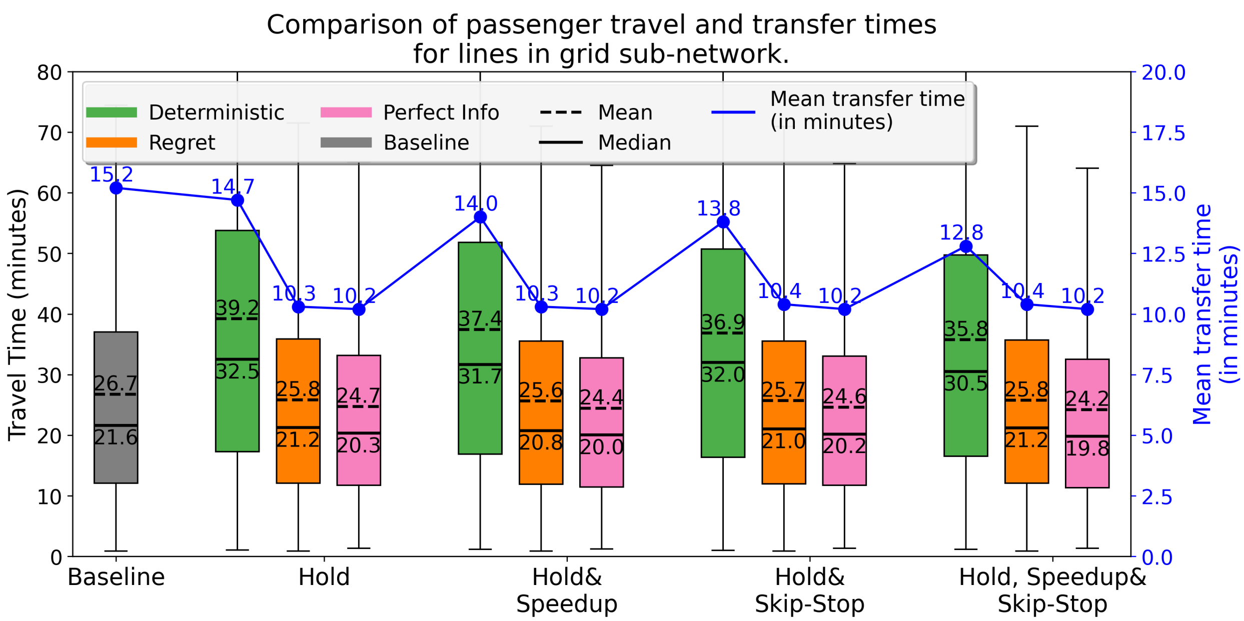

FIGURE 6.8 Grid sub-network travel and transfer times.

Figure 6.8 shows the passenger travel and transfer times for all passengers in the grid sub-network instance, for different control tactics and algorithms. The figure presents thirteen boxplots illustrating passenger travel times (in minutes) on the left y-axis. The first boxplot represents reconstructed passenger trips in the NWS based on historical data, serving as the baseline. Subsequent boxplots are organized into groups corresponding to different optimization algorithms and control tactics. The first group, consisting of three boxplots, compares the performance of Algorithm D, Algorithm R, and the PI solution when only the *hold* tactic is allowed. The second group evaluates the same algorithms under a combination of *hold* and *speedup* tactics. The third group examines the impact of applying both *hold* and *skip-stop*, while the final group assesses outcomes when all three tactics (*hold*, *speedup*, and *skip-stop*) are allowed. Each boxplot displays the mean (dashed line) and median (solid line), with these values explicitly labeled. The figure also includes mean transfer times across all individual transfers, illustrated on the right y-axis. These mean values are annotated above the plotted line connecting them.

We first observe that Algorithm D, when applied to instances involving multiple synchronized lines, substantially increases passenger travel times, despite improving mean transfer times. This outcome is due to the compounding effect of errors made by this approach. Algorithm D generates a single scenario representing the mean state of the network and applies the offline arc-flow model to compute optimal tactics. However, these tactics lack robustness to variations in passenger demand and vehicle travel times. Small delays resulting from inefficient control decisions accumulate across lines, leading to degraded overall performance. In particular, Algorithm D tends to apply a high number of tactics to secure connections, specifically *hold* tactics, which introduces delays along individual lines. These delays propagate, triggering further *hold* actions on connecting lines and initiating a vicious cycle with negative compounding effects. This phenomenon is most pronounced when only the *hold* tactic is available. When *speedup* or *skip-stop* tactics are permitted, some delays can be mitigated.

In contrast, Algorithm R applies fewer *hold* or other tactics. It is more conservative in its decisions, proposing tactics only when they are robust under uncertainty in the system state. As a result, the tactics selected by Algorithm R generate a positive compounding effect. Secondly, we observe that the performance of Algorithm R remains relatively stable across different combinations of control tactics. In previous studies focusing on single-line instances, Algorithm R's performance varied more noticeably depending on the allowed tactic combinations. In the current multi-line configuration, this gap is reduced, as the flexibility of optimizing multiple lines simultaneously allows for a more balanced distribution of tactics across the network. This added flexibility enables the algorithm to compensate for the limitations of certain tactics on one line by leveraging more effective strategies on others, resulting in more consistent overall performance.

Finally, we note that passenger travel times under Algorithm R are improved relative to the baseline but remain higher than those achieved by the PI solution. However, the mean transfer times under Algorithm R are very close to those of the PI solution across all tactic configurations, highlighting the effectiveness of Algorithm R in real-time transfer synchronization. These results suggest that Algorithm R provides a strong compromise between robustness and efficiency, delivering competitive performance while maintaining reliable connections in a dynamic operating environment.

Figure 6.9 shows the distribution of passenger travel time variations for Algorithms D and R,

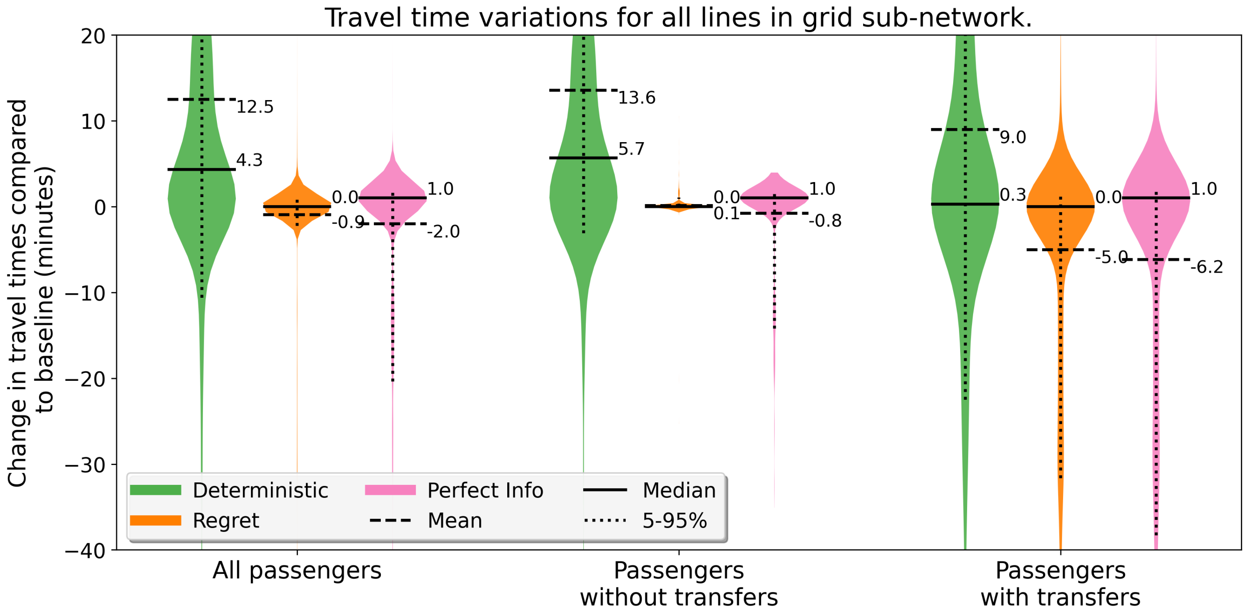

FIGURE 6.9 Grid sub-network travel time variation distributions.

as well as for the PI solution, when only the *hold* tactic is allowed. The figure displays nine violin plots representing the distribution of individual passenger travel time variations (in minutes) relative to the baseline. The plots are grouped into three categories : all passengers, passengers without transfers, and passengers with transfers. Within each group, three violins are shown — from left to right — for Algorithm D, Algorithm R, and the PI solution. Each plot includes mean (dashed line) and median (solid line) values, which are explicitly annotated. A dotted vertical line indicates the interval between the 5th and 95th percentiles, highlighting the spread of the central 90% of the data.

In the first group (all passengers), the violin plot for Algorithm R is narrow and skewed slightly toward negative values, with a mean of –0.9 minutes and a median of exactly zero. This reflects consistent performance with modest time savings. In contrast, Algorithm D shows a wider, positively skewed distribution, indicating a higher frequency of extreme delays and an overall increase in passenger travel times. Compared to both Algorithm D and the PI solution, Algorithm R displays a more concentrated distribution, highlighting its conservative behavior under uncertainty.

This pattern is even more pronounced for passengers without transfers. For this subgroup, Algorithm R has minimal impact, with mean and median travel time variations of 0.1 and 0 minutes, respectively. Its highly concentrated distribution indicates that it avoids applying risky control tactics in stable areas of the network. This conservative approach — rooted in the explicit modeling of uncertainty — helps preserve service quality for non-transferring passengers. This outcome is particularly encouraging for transit operators, as it demonstrates that real-time control can improve transfer reliability without compromising the experience of through passengers.

For passengers with transfers, Algorithm R achieves travel time variations that closely match those of the PI solution, despite its more cautious control policy. This confirms its ability to synchronize transfers effectively in real time while maintaining robustness to uncertainty. The strong alignment between Algorithm R and the PI solution in this subgroup underscores the value of online stochastic optimization for achieving high-quality transfer synchronization without relying on perfect foresight.

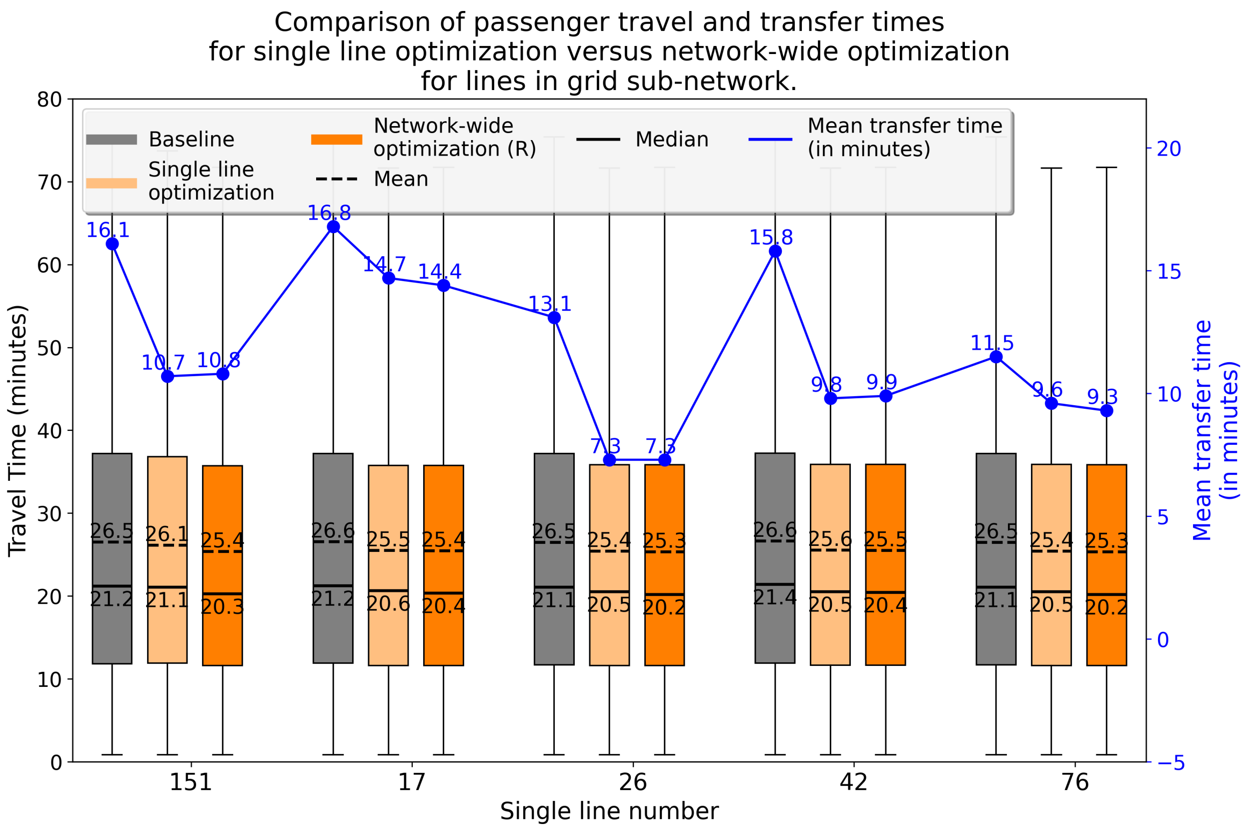

FIGURE 6.10 Grid sub-network travel and transfer times for single and network-wide optimization

Figure 6.10 compares the performance of Algorithm R when applied to individual lines in the grid sub-network, versus optimizing all lines jointly. As in Figure 6.8, passenger travel times are shown on the left y-axis (in minutes), and mean transfer times on the right y-axis. Mean and median values are annotated directly on the boxplots. For each line, three boxplots are shown, representing travel times only for passengers who boarded that line during their trip

— including any transfers made to complete their journey. The first boxplot corresponds to the baseline (no control tactics), the second shows results when Algorithm R is applied to that line in isolation, and the third reflects the performance of Algorithm R for the same passengers when all lines in the sub-network are optimized jointly.

When a line is optimized independently, control tactics can be tailored exclusively to the benefit of its own passengers. In contrast, joint optimization across multiple interconnected lines may require trade-offs to ensure that control tactics provide balanced benefits across the network. However, this integrated approach also enables the application of reinforcing tactics across lines, creating more opportunities for improvement.

Overall, passenger mean and median travel times are slightly lower across all lines when Algorithm R is applied network-wide, compared to when it is applied to individual lines. However, a slight increase in mean transfer time is observed for lines 151 and 42. These lines carry high passenger demand and involve a substantial number of transfers with other lines. In the network-wide setting, some tactics may be adjusted to favor transfers involving other lines, resulting in marginal trade-offs for the most interconnected lines.

Finally, Table 6.1 summarizes the results for all instances. For each test case, and for Algorithm R and the PI solution, it reports the percentage changes relative to the baseline in mean passenger travel time, median travel time, and mean transfer time. It also reports the 5th and 95th percentile variations in passenger travel times (in minutes). For each case, results correspond to the best-performing combination of allowed control tactics.

Across the grid, radial, and corridor sub-networks, our methodology yields the best performance on the grid instance in terms of transfer time improvements. This setting features transfers between main lines that are more spatially distributed, with lines intersecting only once or at a few locations. The limited overlap between lines reduces the likelihood of negative interactions between control tactics, and the high transfer volume combined with dispersed transfer points provides a favorable structure for local synchronization strategies. In the radial network, greater improvements are observed in overall travel times, but the proportional gains in transfer times are smaller. This can be attributed to the presence of loop structures, where control tactics can have delayed and cumulative effects as vehicles complete their loops and re-encounter other lines. Despite these challenges, the results show that substantial transfer time reductions can still be achieved, highlighting the adaptability of the proposed approaches even in networks with more complex connectivity patterns. Nevertheless, the performance gap between Algorithm R and the PI solution is more pronounced in this case, reflecting the increased difficulty of maintaining optimal synchronization tactics under such dynamic conditions.

Table 6.1 Comparison of travel time and transfer time variations under various synchronization tactics and algorithms (H-Hold, SP-Speedup)

Sub-network style	Tactics (best perf.)	Algorithm	Travel time variations (%)				Mean transfer time var. (%)
			Mean	Median	P5	P95	
Grid	H, SP	R	-4.1	-3.7	-1.1	-6.4	-32.2
		PI	-8.6	-7.4	+4.5	-10.5	-32.9
Radial	All	R	-4.8	-5.3	0.0	-4.3	-17.3
		PI	-11.4	-12.5	+6.5	-12.4	-18.6
Corridor	H	R	-4.4	-3.0	+0.1	-7.5	-13.0
		PI	-9.2	-9.4	+10.9	-7.8	-14.8
All	H	R	-3.3	-2.0	+0.7	-7.6	-28.6
		PI	-7.4	-5.0	+10.9	-11.3	-29.2
Transfer hubs	H	R	-2.5	-0.5	+0.4	-6.8	-27.9
		PI	-6.2	-3.0	+9.6	-11.4	-28.6

In the corridor sub-network, although passenger transfer times improve, the overall variation in travel and transfer times is less pronounced. The optimized lines in this instance share common segments and serve overlapping demand, with passengers frequently boarding the first available vehicle and transferring opportunistically. Travel time improvements are partly attributable to enhanced schedule adherence and more evenly spaced vehicle arrivals along these segments, rather than solely to the direct impact of synchronization tactics. These findings suggest that the proposed methodology also contributes to mitigating bus bunching when applied to multiple lines operating along a shared corridor. Moreover, internal comparisons reveal that the corridor instance exhibits the largest performance gap between single-line and network-wide optimization, with significantly greater improvements achieved when optimizing across multiple lines.

Across all sub-network configurations, we observe a consistent pattern in the behavior of the 5th and 95th travel time percentiles. Algorithm R has little to no impact on the 5th percentile, indicating that passengers with the shortest travel times are generally unaffected by the applied control tactics. In contrast, the PI solution often increases travel times for these passengers, sometimes substantially (e.g., +10.9% in the corridor sub-network). However, the PI solution yields consistently greater improvements for the 95th percentile, with reductions of up to 12.4%, thus benefiting passengers experiencing the longest travel times.

For the instance including all lines, significant improvements are achieved in both travel and transfer times. Under Algorithm R, travel time gains primarily benefit passengers with transfers, whereas the PI solution leads to substantial improvements for all passengers. In the transfer hub instance — which involves the same set of passengers but restricts control tactics to the vicinity of the STL's main transfer hubs, accounting for more than 25% of all transfers in the test instance — Algorithm R achieves results that are comparable, albeit slightly inferior, to those obtained when tactics are allowed at any stop across the full network.

6.6 Discussion

The results highlight the effectiveness of the proposed methodology in enhancing transfer synchronization within complex, multi-line transit networks. By integrating the OSO framework and the offline arc-flow model into a NWS, the approach dynamically adapts to real-time conditions, reducing passenger travel times and improving the reliability of transfers. Algorithm R consistently outperforms Algorithm D, underscoring the importance of explicitly addressing uncertainty in real-time transfer synchronization. The modular design of the methodology also supports its scalability to larger and more complex transit systems.

Results indicate that the methodology is particularly effective in grid network configurations, while also showing strong potential in highly frequented corridors of radial systems. Furthermore, applying the methodology to all lines operating near major transfer hubs yields substantial benefits, suggesting that synchronization efforts could be spatially centralized. This could help streamline dispatch operations and reduce the burden on drivers by concentrating tactical interventions in strategically important areas of the network.

From a practical perspective, the proposed framework can serve as a real-time decision support tool for transit agencies, contributing to improved operational efficiency and enhanced passenger experience. The methodology relies primarily on real-time AVL data to track vehicle locations across the network and to estimate feeder line arrival times at transfer stops. In addition, accurate estimation of passenger loads onboard the main line vehicles is required at each bus arrival to validate the generation of alighting passengers and to correctly account for onboard waiting times when applying control tactics. This passenger load can be inferred in near real-time by combining AFC data for boardings and APC data for alightings, both of which are collected in real-time but typically transmitted with some delay. In contrast, full trip inference and origin-destination reconstruction, which require more extensive processing, remain outside the real-time scope and are employed primarily for calibration and the clustering procedures supporting scenario generation. These characteristics make the proposed methodology realistically implementable by transit agencies.

The framework could also be extended to multimodal networks, including micromobility and on-demand transportation services, to evaluate its effectiveness in more heterogeneous systems. From a practical standpoint, further research could explore integration with real-time dispatch tools to support operational decision-making in complex urban environments.

CHAPITRE 7 DISCUSSION

Dans cette thèse, le problème de la synchronisation des correspondances en temps réel a été abordé de manière progressive et structurée, allant de la preuve de concept à l'expérimentation à grande échelle.

7.1 Synthèse des travaux

Le chapitre 4 introduit un modèle de flot hors ligne pour la synchronisation des correspondances entre une ligne principale et de nombreuses lignes secondaires. Cette approche sert de preuve de concept pour l'application d'un modèle de flot au problème de synchronisation des correspondances en temps-réel. Le modèle est également testé dans un environnement dynamique avec information imparfaite, à l'aide d'un algorithme déterministe *DETERMINISTIC* (D) fondé sur un scénario unique représentant l'état moyen du système. Cette approche permet de capturer avec précision les effets de la synchronisation sur l'ensemble du trajet des usagers, tout en adressant explicitement les dimensions dynamiques et temps réel du problème.

Cette base méthodologique est étendue dans le chapitre 5, où deux algorithmes d'Optimisation Stochastique En Ligne (OSEL), *CONSENSUS* (C) et *REGRET* (R), sont dérivés du modèle initial. Ces algorithmes, spécifiquement adaptés au contexte de la synchronisation des correspondances, sont intégrés dans un cadre de simulation à événements discrets. Ils reposent sur la génération de scénarios multiples à partir de données historiques et en temps réel, exploitant ainsi pleinement les sources d'information disponibles aux opérateurs. Cette modélisation permet de représenter de manière explicite l'incertitude du système avec un ensemble fini de prévisions, et d'évaluer, pour chaque scénario, les effets des tactiques de contrôle avant de prendre une décision robuste.

Enfin, le chapitre 6 généralise la méthodologie à l'échelle d'un réseau complet, en l'intégrant dans un simulateur multimodal. Cette dernière contribution permet l'optimisation conjointe de plusieurs lignes interconnectées et l'évaluation de la performance des algorithmes dans des configurations variées de réseaux de transport public. Elle confirme la robustesse, l'efficacité et l'évolutivité de l'approche proposée, ouvrant ainsi la voie à des applications opérationnelles à grande échelle.

Cette trajectoire de recherche respecte l'objectif initial formulé dans l'introduction : proposer une méthodologie à la fois réaliste, robuste, et applicable à grande échelle.

7.1.1 Une méthodologie réaliste et intégrée

Les algorithmes développés dans cette thèse reposent sur l'exploitation de données empiriques issues d'un réseau de transport en exploitation réelle. Un prétraitement structuré, incluant des opérations de nettoyage et de *clustering* par jour, ligne, arrêt et heure, permet de valoriser ces données pour générer dynamiquement des scénarios représentatifs des conditions d'exploitation. Ces scénarios intègrent également l'état courant du système à partir de données en temps réel, telles que les positions des autobus, assurant ainsi un cadre expérimental fidèle aux contraintes opérationnelles.

Les modèles proposés capturent l'ensemble des composantes du parcours usager — attente, trajet à bord, correspondances, effets des tactiques de contrôle comme la marche induite — ainsi que les interactions entre flux de passagers et circulation des véhicules. Cette représentation intégrée permet d'évaluer de manière cohérente les impacts systémiques des tactiques de synchronisation, tant sur les passagers que sur les véhicules. Les performances sont mesurées à l'échelle du réseau à l'aide d'indicateurs globaux, notamment les temps de parcours individuels de l'ensemble des usagers. Enfin, les approches développées permettent d'analyser les effets de bord induits par les décisions locales dans des contextes topologiques variés.

7.1.2 Une méthodologie fiable et robuste

Les approches proposées intègrent explicitement l'incertitude inhérente aux systèmes de transport public en contexte dynamique. Celle-ci est modélisée au sein des algorithmes d'OSEL par la génération de scénarios multiples représentant des futurs possibles du réseau. Cette démarche permet de produire des solutions robustes face aux variations des temps de trajet, des flux de correspondances et de la demande en passagers.

L'algorithme R, en particulier, se distingue par des performances proches de celles de la solution à information parfaite, bien qu'il opère sans accès aux données futures. Il permet de réduire significativement les temps de parcours ainsi que les temps de correspondance, tout en assurant une stabilité des résultats dans des conditions opérationnelles variées. Lorsqu'il est déployé à l'échelle de sous-réseaux, les tactiques appliquées par R peuvent engendrer des effets de synergie entre lignes, amplifiant leur impact mutuel. Au contraire, notre algorithme déterministe D provoque des interférences négatives à l'échelle des sous-réseaux. Par ailleurs, l'analyse de la distribution des temps de parcours individuels montre que R applique les tactiques de manière plus ciblée que l'algorithme D et la solution à information parfaite. Celles-ci affectent principalement les usagers en correspondance, limitant les perturbations pour les autres passagers.

Ces résultats confirment que l'algorithme R constitue une solution à la fois robuste et efficace, capable de soutenir une prise de décision en temps réel.

7.1.3 Une méthodologie flexible et évolutive

La méthodologie développée se caractérise par une grande flexibilité d'application et une capacité d'adaptation à la complexité croissante des réseaux de transport. Elle permet d'optimiser des lignes présentant un nombre important d'arrêts et de correspondances. La génération de scénarios est paramétrable en fonction des niveaux d'incertitude, des horizons temporels et des besoins décisionnels propres à chaque contexte. Enfin, à l'échelle du réseau, la méthodologie permet de cibler des sous-réseaux structurants ou des typologies particulières, assurant ainsi une adaptabilité à des contextes opérationnels variés et à des réseaux de toute taille.

7.1.4 Une méthodologie ouverte et reproductible

Afin de favoriser la reproductibilité des résultats et de faciliter l'adoption des outils développés, l'ensemble du code source associé à cette thèse est mis à disposition dans un dépôt *GitHub* en accès libre. Ce dépôt inclut des modules de traitement de données, des jeux de tests, ainsi que des tutoriels d'utilisation. L'objectif est de permettre à d'autres chercheurs, praticiens ou partenaires industriels de reproduire l'analyse, de tester la méthodologie sur leurs propres données, ou d'adapter les outils à leurs besoins.

Le dépôt propose une interface pour charger et prétraiter des données issues de formats standards (notamment GTFS et fichiers *Comma-Separated Values* (CSV) dérivés de cartes à puce). Les colonnes d'attributs attendues sont documentées de manière explicite et des exemples concrets sont fournis pour faciliter la prise en main. Des modules permettent également de générer automatiquement des instances de test à partir d'un jeu de données GTFS, avec une flexibilité sur le choix des lignes, des périodes d'analyse, ou du sous-ensemble d'usagers pris en compte (par exemple : tous les usagers ou seulement ceux affectés par certaines lignes).

Du côté algorithmique, des outils de regroupement et de prétraitement de données historiques sont fournis, permettant de générer des scénarios à partir de données historiques propres à l'utilisateur, peu importe la quantité de données disponibles (par exemple : une semaine, un mois ou même une année entière). Tous les algorithmes mentionnés dans cette thèse sont disponibles sur le dépôt et peuvent être lancés facilement. Différents outils proposent une analyse ainsi qu'une visualisation des résultats obtenus. L'ensemble de la démarche vise à rendre la méthodologie non seulement reproductible, mais également transférable et adaptable à une variété de réseaux et d'environnements opérationnels.

7.1.5 Une méthodologie efficace

Les expérimentations menées démontrent l'efficacité des méthodologies proposées selon plusieurs axes. D'abord, les résultats produisent une réduction significative des temps de parcours pour les usagers en correspondance, ainsi qu'une amélioration du taux de correspondances réussies et des temps d'attente associés. Ces bénéfices sont obtenus sans détériorer les trajets des usagers sans correspondance, préservant ainsi l'équité du service. Dans certains contextes, les interventions ont également permis une meilleure adhérence aux horaires planifiés, en limitant les effets de propagation des retards. En outre, les algorithmes proposés présentent des temps de calcul compatibles avec une application en temps réel, même dans des situations complexes impliquant de nombreuses correspondances.

7.2 Application au niveau de l'opérateur

Au-delà de la conception algorithmique, l'objectif principal de cette recherche est de proposer une méthodologie applicable en contexte réel, en tant qu'outil d'aide à la décision pour la STL et les opérateurs de transport collectif. Cette démarche s'accompagne d'une réflexion sur les conditions pratiques d'intégration des tactiques de synchronisation des correspondances en temps réel dans la gestion opérationnelle des réseaux.

Sur le plan des données, la mise en œuvre des méthodes proposées repose sur la disponibilité de données historiques et en temps réel. Les données historiques — GTFS pour l'offre de service, AFC pour les validations de cartes à puce, et AVL pour la localisation des véhicules — sont prétraitées en amont et sont hors du périmètre de l'optimisation en temps réel. L'enrichissement de ces données, selon les méthodes décrites dans les chapitres précédents, permet de générer via les algorithmes d'OSEL des scénarios réalistes menant à des décisions robustes sans accès à une information parfaite. Lors de chaque réoptimisation en temps réel, seules deux informations sont nécessaires pour la génération des scénarios : (i) la charge à bord du véhicule principal, estimée par les données issues de cartes à puce et/ou des compteurs de passagers, et (ii) les positions des véhicules sur les lignes de correspondance, obtenues via les données AVL en temps réel. Toutes les autres variables utilisées dans la prise de décision sont simulées dans les scénarios.

Plusieurs enseignements clés émergent de cette recherche en ce qui concerne l'implémentation des tactiques en contexte opérationnel. Par exemple, appliquer des tactiques de contrôle trop en amont ne procure pas de gains significatifs, tout en augmentant la complexité computationnelle. C'est pourquoi un horizon flottant de la bonne taille (10 à 15 arrêts dans le cas de la STL) maximise l'efficacité. De plus, l'application de tactiques à une seule ligne s'avère particulièrement efficace sur les lignes à basse fréquence, où chaque correspondance réussie génère un gain de temps considérable pour les usagers concernés. Ce phénomène est encore plus marqué lors de l'heure de pointe du soir, quand de nombreux usagers effectuent des correspondances depuis des lignes à haute fréquence vers des lignes à plus basse fréquence pour le dernier segment de leurs trajets. Ensuite, nos algorithmes appliquent des tactiques à un nombre restreint d'arrêts, ce qui faciliterait leur transmission aux chauffeurs d'autobus. C'est notamment le cas de l'algorithme R, dont le comportement conservateur limite les perturbations sur l'ensemble du réseau. À l'échelle du réseau, la synchronisation des correspondances en temps réel est plus bénéfique dans des structures maillées que dans des réseaux radiaux. Toutefois, sur des corridors structurants au sein d'un réseau radial, la coordination de quelques lignes clés peut s'avérer très efficace. Finalement, synchroniser l'ensemble des lignes d'un réseau en temps réel est trop coûteux en temps de calcul et en coordination humaine. En revanche, une stratégie ciblée, concentrée autour de *hubs* de correspondances majeurs, permet d'atteindre des résultats similaires avec une charge opérationnelle réduite.

Enfin, il est essentiel de replacer la synchronisation en temps réel dans une perspective plus large. Elle ne constitue pas une solution miracle, mais bien un levier d'adaptation face aux aléas opérationnels affectant l'exécution des horaires planifiés. Elle vise à limiter les effets négatifs des fluctuations de la demande et des temps de parcours, mais ses bénéfices restent conditionnels à la qualité de la planification initiale. En d'autres termes, si les correspondances ne sont pas synchronisées dès la conception des lignes et des horaires, les marges d'amélioration lors des opérations en temps réel sont limitées. À l'inverse, même une planification optimale est exposée à l'incertitude inhérente aux opérations, ce qui souligne la nécessité d'une complémentarité étroite entre planification robuste et contrôle en temps réel.

7.3 Limitations de la solution proposée et améliorations futures

Les travaux présentés dans cette thèse explorent différents compromis entre efficacité des solutions et complexité de mise en œuvre. À travers les trois articles, nous proposons des méthodes de plus en plus performantes pour la synchronisation des correspondances, au prix d'une complexité croissante, tant algorithmique qu'opérationnelle.

Dans le chapitre 4, nous introduisons un premier modèle hors ligne, relativement simple à implémenter. Ce modèle est intégré à l'algorithme D qui, bien qu'il ne prenne pas en compte l'incertitude, modélise le caractère dynamique du problème. Les résultats obtenus sont prometteurs, mais un écart subsiste avec la solution à information parfaite, en particulier dans les contextes à forte variabilité.

Le chapitre 5 propose deux algorithmes d'OSEL, dont l'implémentation est plus complexe. Cette complexité accrue permet à l'algorithme R de générer des tactiques robustes, performantes sur l'ensemble des scénarios considérés. Toutefois, cette robustesse s'accompagne de temps de calcul plus élevés, limitant la taille de l'horizon temporel flottant utilisable en pratique. À l'inverse, l'algorithme C, plus léger à exécuter, affiche des performances nettement inférieures, car il néglige les conséquences d'une tactique sur les scénarios où elle n'est pas optimale. Par ailleurs, ces deux algorithmes sont appliqués à une seule ligne principale à la fois.

Dans le chapitre 6, la méthodologie est étendue à plusieurs lignes simultanément, à l'échelle de sous-réseaux. Les résultats montrent que l'algorithme R conserve d'excellentes performances qui se rapprochent de la solution à information parfaite, pour des groupes de lignes structurantes. Cependant, les réoptimisations sont effectuées de manière séquentielle, sans parallélisation, ce qui freine l'applicabilité à l'ensemble du réseau de la STL en temps réel. Une amélioration envisageable consisterait à paralléliser l'optimisation pour différents véhicules ou lignes, en maintenant la cohérence globale des décisions.

Enfin, bien que les algorithmes proposés aient démontré leur efficacité, une piste de recherche complémentaire consisterait à les comparer à des heuristiques simples, appliquées localement à proximité des *hubs* de correspondance. Ces approches, moins exigeantes en calcul et en coordination, pourraient offrir une solution plus facilement déployable pour les opérateurs.

CHAPITRE 8 CONCLUSION

Cette thèse a abordé le problème de la synchronisation des correspondances en temps réel en tenant compte de l'ensemble des étapes du parcours des passagers, tout en intégrant explicitement l'incertitude inhérente aux réseaux de transport public. Nous avons d'abord proposé un modèle de flot hors ligne, puis développé deux algorithmes d'Optimisation Stochastique En Ligne capables d'en exploiter la structure dans un cadre de simulation à événements discrets. L'étude montre que la prise en compte explicite de l'incertitude permet de produire des solutions robustes et applicables. Cette approche est ensuite étendue à l'échelle de réseaux entiers, tout en maîtrisant la complexité associée.

Les résultats obtenus confirment que la synchronisation en temps réel, lorsqu'elle est ciblée avec précision et appuyée par des données historiques et en temps réel, constitue un levier opérationnel concret pour améliorer la qualité de service. L'ambition de ce travail est de fournir des algorithmes efficaces et exploitables par notre partenaire, la STL, ainsi que par les opérateurs de transport collectif.

Enfin, cette thèse a été l'occasion pour moi de mieux comprendre les enjeux liés au transport public, d'en apprécier la complexité, et de confirmer mon souhait de poursuivre une carrière professionnelle dans ce domaine porteur de sens.

RÉFÉRENCES

[1] G. Héon, "L'ATUQ dévoile les résultats de l'étude sur les impacts économiques du transport en commun au Québec," Publication officielle de l'Association du transport urbain du Québec, 2024. [En ligne]. Disponible : https://atuq.com/fr/latuq-devoile-les-resultats-de-letude-sur-les-impacts-economiques-du-transport-en-commun-au-quebec/

[2] "Étude sur les impacts économiques des sociétés de transport en commun membres de l'association du transport urbain du québec (données de 2022)," Association du transport urbain du Québec, Rapport technique, 2023. [En ligne]. Disponible : https://atuq.com/wp-content/uploads/2024/02/Etude-Impacts_econo-complete-ATUQ-2023-.pdf

[3] "Enquête origine-destination 2023 – faits saillants," https://www.artm.quebec/wp-content/uploads/2024/10/ARTM_Enquete2023_Faits_Saillants.pdf, Autorité régionale de transport métropolitain (ARTM), Rapport technique, 2024, consulté en mars 2025.

[4] J. Laviolette, "L'état de l'automobile au québec : constats, tendances et conséquences," https://fr.davidsuzuki.org/wp-content/uploads/sites/3/2020/10/Rapport_Fondation-David-Suzuki-Final-Part1-Dependance-auto-10.2020.pdf, 2020, consulté en mars 2025.

[5] H. Lee et J. Romero, "Summary for policymakers," dans *Climate Change 2023 : Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change.* Geneva, Switzerland : IPCC, 2023, p. 1–34, intergovernmental Panel on Climate Change (IPCC). [En ligne]. Disponible : https://www.ipcc.ch/report/ar6/syr/downloads/report/IPCC_AR6_SYR_SPM.pdf

[6] Gouvernement du Canada, "Plan d'action du canada pour un transport routier propre," https://tc.canada.ca/fr/transport-routier/publications/plan-action-canada-transport-routier-propre, 2022, tP 15406F, Nº de cat. T42-30/2022F-PDF, ISBN 978-0-660-46759-7. Consulté en mars 2025.

[7] Ministère de l'Environnement, de la Lutte contre les changements climatiques, de la Faune et des Parcs, "Inventaire québécois des émissions de gaz à effet de serre en 2022 et leur évolution depuis 1990," Bibliothèque et Archives nationales du Québec, 2024, ISBN 978-2-550-99147-2 (PDF).

[8] Ville de Laval, "Laval 2031 : Une ville en mouvement – plan de mobilité durable," https://www.laval.ca/wp-content/uploads/2025/02/plan-mobilite-durable-consultation.pdf, 2011.

[9] Société de l'assurance automobile du Québec (SAAQ), "Données et statistiques 2022,"

2023, ISBN 978-2-550-96519-0. [En ligne]. Disponible : https://saaq.gouv.qc.ca/blob/saaq/documents/publications/donnees-statistiques-2022.pdf

[10] Autorité régionale de transport métropolitain (ARTM), “Enquête origine-destination 2018,” https://www.artm.quebec/wp-content/uploads/2020/01/CA_Faits-saillants_EOD_COMPLET_WEB_14012020_R002.pdf, 2018, consulté en mars 2025.

[11] Ministère des Transports, de la Mobilité durable et de l'Électrification des transports, “Transporter le québec vers la modernité – politique de mobilité durable 2030,” https://cdn-contenu.quebec.ca/cdn-contenu/adm/min/transports/ministere-des-transports/publications-amd/Plan_de_mobilite_durable/PO_politique-mobilite-durable_MTMDET.pdf, Gouvernement du Québec, Québec, Rapport technique, 2018.

[12] T. Mehdi et R. Morissette, “Travail à domicile et utilisation du transport en commun au canada, 2016 à 2023,” *Rapports économiques et sociaux*, vol. 4, n°. 1, 2024. [En ligne]. Disponible : https://doi.org/10.25318/36280001202400100002-fra

[13] Ministère des Transports et de la Mobilité durable, “Politique de mobilité durable 2030 – bilan synthèse 2022-2023,” https://cdn-contenu.quebec.ca/cdn-contenu/adm/min/transports/ministere-des-transports/publications-amd/Plan_de_mobilite_durable/pmd-bilan-2022-2023.pdf, Gouvernement du Québec, Québec, Rapport technique ISBN 978-2-550-96143-7, 2023.

[14] O. Ibarra-Rojas *et al.*, “Planning, operation, and control of bus transport systems : A literature review,” *Transportation Research Part B : Methodological*, vol. 77, p. 38–75, 2015. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0191261515000454

[15] T. Liu, O. Cats et K. Gkiotsalitis, “A review of public transport transfer coordination at the tactical planning phase,” *Transportation Research Part C : Emerging Technologies*, vol. 133, p. 103450, 2021. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X21004393

[16] K. Gkiotsalitis et O. Cats, “At-stop control measures in public transport : Literature review and research agenda,” *Transportation Research Part E : Logistics and Transportation Review*, vol. 145, p. 102176, 2021. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S136655452030819X

[17] K. Gkiotsalitis, O. Cats et T. Liu, “A review of public transport transfer synchronisation at the real-time control phase,” *Transport Reviews*, vol. 43, n°. 1, p. 88–107, 2023. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0144164722004378

[18] X. J. Eberlein, N. H. M. Wilson et D. Bernstein, “The holding problem with real–time information available,” *Transportation Science*, vol. 35, n°. 1, p. 1–18, 2001. [En ligne].

Disponible : https://doi.org/10.1287/trsc.35.1.1.10143

[19] A. A. Ceder *et al.*, “Transfer synchronization of public transport networks,” *Transportation Research Record*, vol. 2350, nº. 1, p. 9–16, 2013. [En ligne]. Disponible : https://doi.org/10.3141/2350-02

[20] M. M. Nesheli et A. A. Ceder, “Optimal combinations of selected tactics for public-transport transfer synchronization,” *Transportation Research Part C : Emerging Technologies*, vol. 48, p. 491–504, 2014. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X14002605

[21] ——, “A robust, tactic-based, real-time framework for public-transport transfer synchronization,” *Transportation Research Part C : Emerging Technologies*, vol. 60, p. 105–123, 11 2015.

[22] A. Sun et M. Hickman, “The holding problem at multiple holding stations,” dans *Computer-aided Systems in Public Transport*. Springer Berlin Heidelberg, 01 2008, p. 339–359.

[23] C. Daganzo, “A headway-based approach to eliminate bus bunching : Systematic analysis and comparisons,” *Transportation Research Part B : Methodological*, vol. 43, p. 913–921, 12 2009.

[24] O. Cats *et al.*, “Bus-holding control strategies : Simulation-based evaluation and guidelines for implementation,” *Transportation Research Record*, vol. 2274, nº. 1, p. 100–108, 2012. [En ligne]. Disponible : https://doi.org/10.3141/2274-11

[25] O. Cats, F. Rufi et H. Koutsopoulos, “Optimizing the number and location of time point stops,” *Public Transport*, vol. 6, p. 215–235, 10 2014.

[26] M. D. Hickman, “An analytic stochastic model for the transit vehicle holding problem,” *Transportation Science*, vol. 35, nº. 3, p. 215–237, 2001. [En ligne]. Disponible : http://www.jstor.org/stable/25768957

[27] P. Furth et T. Muller, “Optimality conditions for public transport schedules with time-point holding,” *Public Transport*, vol. 1, p. 87–102, 06 2009.

[28] M. D. Abkowitz et M. Lepofsky, “Implementing headway-based reliability control on transit routes,” *Journal of Transportation Engineering*, vol. 116, nº. 1, p. 49–63, 1990. [En ligne]. Disponible : https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-947X%281990%29116%3A1%2849%29

[29] A. Carrel *et al.*, “Decision factors in service control on high-frequency metro line,” *Transportation Research Record : Journal of the Transportation Research Board*, vol. 2146, 12 2010.

[30] M. Trépanier, N. Tranchant et R. Chapleau, "Individual trip destination estimation in a transit smart card automated fare collection system," *Journal of Intelligent Transportation Systems*, vol. 11, n°. 1, p. 1–14, 2007. [En ligne]. Disponible : https://doi.org/10.1080/15472450601122256

[31] L. He *et al.*, "Validating and calibrating a destination estimation algorithms for public transport smart card fare collection systems," *CIRRELT*, vol. 2015-52, p. 1–13, 2015.

[32] M. Dessouky *et al.*, "Real-time control of buses for schedule coordination at a terminal," *Transportation Research Part A : Policy and Practice*, vol. 37, p. 145–164, 02 2003.

[33] A. Gavriilidou et O. Cats, "Reconciling transfer synchronization and service regularity : real-time control strategies using passenger data," *Transportmetrica A : Transport Science*, vol. 15, n°. 2, p. 215–243, 2019. [En ligne]. Disponible : https://doi.org/10.1080/23249935.2018.1458757

[34] T. Liu *et al.*, "Synchronizing public transport transfers by using intervehicle communication scheme : Case study," *Transportation Research Record*, vol. 2417, n°. 1, p. 78–91, 2014. [En ligne]. Disponible : https://doi.org/10.3141/2417-09

[35] M. M. Nesheli, A. A. Ceder et V. A. Gonzalez, "Real-time public-transport operational tactics using synchronized transfers to eliminate vehicle bunching," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, n°. 11, p. 3220–3229, 2016.

[36] Y. Hadas et A. A. Ceder, "Improving bus passenger transfers on road segments through online operational tactics," *Transportation Research Record*, vol. 2072, n°. 1, p. 101–109, 2008. [En ligne]. Disponible : https://doi.org/10.3141/2072-11

[37] ——, "Public transit simulation model for optimal synchronized transfers," *Transportation Research Record*, vol. 2063, n°. 1, p. 52–59, 2008. [En ligne]. Disponible : https://doi.org/10.3141/2063-07

[38] ——, "Optimal coordination of public-transit vehicles using operational tactics examined by simulation," *Transportation Research Part C : Emerging Technologies*, vol. 18, n°. 6, p. 879–895, 2010. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X10000586

[39] C. Daganzo et P. Anderson, "Coordinating transit transfers in real time," Institute of Transportation Studies, UC Berkeley, Rapport technique qt25h4r974, mai 2016. [En ligne]. Disponible : https://ideas.repec.org/p/cdl/itsrrp/qt25h4r974.html

[40] H. Manasra et T. Toledo, "Optimization-based operations control for public transportation service with transfers," *Transportation Research Part C : Emerging Technologies*, vol. 105, p. 456–467, 2019. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X18308878

[41] W. B. Powell, "A unified framework for stochastic optimization," *European Journal of Operational Research*, vol. 275, n°. 3, p. 795–821, 2019. [En ligne]. Disponible : https://ideas.repec.org/a/eee/ejores/v275y2019i3p795-821.html

[42] M. Dessouky *et al.*, "Bus dispatching at timed transfer transit stations using bus tracking technology," *Transportation Research Part C : Emerging Technologies*, vol. 7, n°. 4, p. 187–208, 1999. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X99000194

[43] S. J. Berrebi *et al.*, "Comparing bus holding methods with and without real-time predictions," *Transportation Research Part C : Emerging Technologies*, vol. 87, p. 197–211, 2018. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0968090X17302000

[44] L. M. Kieu *et al.*, "Transfer demand prediction for timed transfer coordination in public transport operational control," *Journal of Advanced Transportation*, vol. 50, n°. 8, p. 1972–1989, 2016. [En ligne]. Disponible : https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.1440

[45] T. Liu et A. A. Ceder, "Communication-based cooperative control strategy for public transport transfer synchronization," *Transportation Research Record*, vol. 2541, n°. 1, p. 27–37, 2016. [En ligne]. Disponible : https://doi.org/10.3141/2541-04

[46] A. Hernandez-Arauzo *et al.*, "Dynamic scheduling of electric vehicle charging under limited power and phase balance constraints," dans *Proceedings of SPARK 2013-Scheduling and Planning Applications Workshop.*, 06 2013.

[47] V. Pillac *et al.*, "A review of dynamic vehicle routing problems," *European Journal of Operational Research*, vol. 225, p. 1–11, 02 2013.

[48] H. S. Chang, R. Givan et E. K. P. Chong, "On-line scheduling via sampling," dans *International Conference on Artificial Intelligence Planning Systems*, 2000. [En ligne]. Disponible : https://api.semanticscholar.org/CorpusID:2755062

[49] R. Bent et P. V. Hentenryck, *Online Stochastic Combinatorial Optimization.* The MIT Press, 2009.

[50] ——, "The value of consensus in online stochastic scheduling," dans *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, ser. ICAPS'04. AAAI Press, 2004, p. 219–226.

[51] ——, "Online stochastic optimization without distributions," dans *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling*, ser. ICAPS'05. AAAI Press, 2005, p. 171–180.

[52] ——, “Regrets only ! online stochastic optimization under time constraints,” dans *AAAI Conference on Artificial Intelligence*, 2004. [En ligne]. Disponible : https://api.semanticscholar.org/CorpusID:1819722

[53] A. Philpott et V. de Matos, “Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion,” *European Journal of Operational Research*, vol. 218, p. 470–483, 01 2012.

[54] M. Lin, K. Tang et X. Yao, “Dynamic sampling approach to training neural networks for multiclass imbalance classification,” *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, p. 647–660, 04 2013.

[55] A. D. Filippo, M. Lombardi et M. Milano, “How to tame your anticipatory algorithm,” dans *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, p. 1071–1077. [En ligne]. Disponible : https://doi.org/10.24963/ijcai.2019/150

[56] P. Van Hentenryck, R. Bent et E. Upfal, “Online stochastic optimization under time constraints,” *Annals OR*, vol. 177, p. 151–183, 06 2010.

[57] A. A. Ceder *et al.*, “Transfer synchronization of public transport networks,” *Transportation Research Record*, vol. 2350, nº. 1, p. 9–16, 2013. [En ligne]. Disponible : https://doi.org/10.3141/2350-02

[58] A. Boardman *et al.*, *Cost-Benefit Analysis : Concepts and Practice, 2nd edition*. Cambridge University Press, 01 2001.

[59] A. A. Ceder, *Public Transit Planning and Operation : Theory, Modeling, and Practice*. Oxford, United Kingdom : Butterworth-Heinemann, Elsevier, 2007.

[60] X. Eberlein, N. H. M. Wilson et D. Bernstein, “Modeling real-time control strategies in public transit operations,” *Lecture Notes in Economics and Mathematical Systems*, p. 325–346, 1999.

[61] L. Fu et X. Yang, “Design and implementation of bus–holding control strategies with real-time information,” *Transportation Research Record*, vol. 1791, nº. 1, p. 6–12, 2002. [En ligne]. Disponible : https://doi.org/10.3141/1791-02

[62] J. J. Bartholdi et D. D. Eisenstein, “A self-coordinating bus route to resist bus bunching,” *Transportation Research Part B : Methodological*, vol. 46, nº. 4, p. 481–491, 2012. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0191261511001676

[63] J. Zhao, S. Bukkapatnam et M. Dessouky, “Distributed architecture for real-time coordination of bus holding in transit networks,” *Intelligent Transportation Systems, IEEE*

Transactions on, vol. 4, p. 43 – 51, 04 2003.

[64] A. Puong et N. H. M. Wilson, *A Train Holding Model for Urban Rail Transit Systems*, M. Hickman, P. Mirchandani et S. Voß, édit. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008.

[65] S. Zolfaghari, N. Azizi et M. Y. Jaber, “A model for holding strategy in public transit systems with real-time information,” *International Journal of Transport Management*, vol. 2, n°. 2, p. 99–110, 2004. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S1471405105000030

[66] F. Delgado, J. C. Munoz et R. Giesen, “How much can holding and/or limiting boarding improve transit performance ?” *Transportation Research Part B : Methodological*, vol. 46, n°. 9, p. 1202–1217, 2012. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0191261512000653

[67] A. Sun et M. Hickman, “The real–time stop–skipping problem,” *Journal of Intelligent Transportation Systems*, vol. 9, n°. 2, p. 91–109, 2005. [En ligne]. Disponible : https://doi.org/10.1080/15472450590934642

[68] B. Yu et Z. Yang, “A dynamic holding strategy in public transit systems with real-time information,” *Applied Intelligence*, vol. 31, n°. 1, p. 69–80, aug 2009. [En ligne]. Disponible : https://doi.org/10.1007/s10489-007-0112-9

[69] O. Cats *et al.*, “Impacts of holding control strategies on transit performance : Bus simulation model analysis,” *Transportation Research Record*, vol. 2216, n°. 1, p. 51–58, 2011. [En ligne]. Disponible : https://doi.org/10.3141/2216-06

[70] T. Liu et A. A. Ceder, “Synchronization of public transport timetabling with multiple vehicle types,” *Transportation Research Record*, vol. 2539, n°. 1, p. 84–93, 2016. [En ligne]. Disponible : https://doi.org/10.3141/2539-10

[71] Y. Ji et H. M. Zhang, “Dynamic holding strategy to prevent buses from bunching,” *Transportation Research Record*, vol. 2352, n°. 1, p. 94–103, 2013. [En ligne]. Disponible : https://doi.org/10.3141/2352-11

[72] G. Sánchez-Martínez, H. Koutsopoulos et N. Wilson, “Real-time holding control for high-frequency transit with dynamics,” *Transportation Research Part B : Methodological*, vol. 83, p. 1–19, 01 2016.

[73] C.-J. Ting et P. Schonfeld, “Dispatching control at transfer stations in multi-hub transit networks,” *Journal of Advanced Transportation*, vol. 41, n°. 3, p. 217–243, 2007. [En ligne]. Disponible : https://onlinelibrary.wiley.com/doi/abs/10.1002/atr.5670410302

[74] E.-H. Chung, M. M. Nesheli et A. Shalaby, “Transit holding control model for real-time connection protection,” *Journal of Transportation Engineering, Part*

A : Systems, vol. 146, nº. 4, p. 04020021, 2020. [En ligne]. Disponible : https://ascelibrary.org/doi/abs/10.1061/JTEPBS.0000332

[75] C. A. Guevara et G. A. Donoso, “Tactical design of high-demand bus transfers,” *Transport Policy*, vol. 32, p. 16–24, 2014. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0967070X13001777

[76] L. Kolcheva, A. Legrain et M. Trépanier, “Data driven synchronization strategies of a bus line in a transit network,” *Transportation Research Procedia*, vol. 82, p. 3333–3351, 2025. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S2352146524003892

[77] L. Kolcheva, A. Legrain et M. Trépanier, “Online stochastic optimization for real-time transfer synchronization in public transit networks,” *CIRRELT*, vol. 2024-19, p. 1–25, 2024.

ANNEXE A ANNEXES DE L'ARTICLE 1 : *DATA DRIVEN SYNCHRONIZATION STRATEGIES OF A BUS LINE IN A TRANSIT NETWORK*

TABLE A.1 Average passenger travel times (holding only). EM-Early Morning, RH-Rush Hour

Time of day	Case	Wait for boarding	Wait for transfer (boarding)	Missed transfer (boarding)	Dwell time	Hold time	Travel time	Wait for transfer (alighting)	Missed transfer (alighting)	Walking time if skipped stop	Total time	Passengers with missed boarding transfers (%)	Passengers with missed alighting transfers (%)
EM	No Tactics	1.6	1.5	1.7	2.8	0.0	12.7	0.7	0.0	0.0	21.0	7.7	1.1
EM	Online	1.9	1.5	0.2	2.9	0.0	12.7	0.7	0.1	0.0	20.0	0.7	0.6
EM	Offline	1.7	1.5	0.3	2.9	0.0	12.7	0.6	0.0	0.0	19.8	1.3	0.7
RH AM	No Tactics	1.5	2.3	1.9	2.2	0.0	11.4	2.0	0.1	0.0	21.3	6.0	1.3
RH AM	Online	2.0	2.3	0.5	2.5	0.3	11.2	1.9	0.1	0.0	20.7	1.3	1.0
RH AM	Offline	1.6	2.3	0.5	2.2	0.0	11.5	1.6	0.1	0.0	19.8	1.3	1.7
Midday	No Tactics	2.9	3.3	0.9	3.2	0.0	14.1	2.4	0.2	0.0	26.9	2.9	1.2
Midday	Online	3.2	3.3	0.4	3.3	0.2	14.0	2.3	0.3	0.0	27.0	1.6	1.7
Midday	Offline	3.0	3.2	0.5	3.2	0.1	14.1	2.1	0.2	0.0	26.4	1.7	2.0
RH PM	No Tactics	4.1	2.7	0.1	3.8	0.0	17.1	1.9	0.0	0.0	29.8	0.4	0.6
RH PM	Online	4.2	2.8	0.1	4.0	0.2	16.9	1.9	0.1	0.0	30.2	0.2	0.7
RH PM	Offline	4.1	2.8	0.1	3.8	0.0	17.1	1.8	0.0	0.0	29.8	0.4	0.9
Night	No Tactics	2.6	2.5	0.0	3.1	0.0	15.8	2.1	0.0	0.0	26.1	0.0	0.0
Night	Online	2.7	2.5	0.0	3.3	0.2	15.6	2.1	0.0	0.0	26.3	0.0	0.0
Night	Offline	2.6	2.5	0.0	3.1	0.0	15.8	2.1	0.0	0.0	26.1	0.0	0.0

TABLE A.2 Average bus travel times and tactics (holding only). EM-Early Morning, RH-Rush Hour

Time of day	Case	Dwell time	Hold time	Travel time	Total time	Stops with hold(%)	Skipped stops(%)	Stops with speedup(%)
EM	No Tactics	11.2	0.0	61.1	72.3	0.0	0.0	0.0
EM	Online	11.2	0.4	61.7	73.3	1.1	0.0	0.0
EM	Offline	11.2	0.0	61.4	72.5	1.2	0.0	0.0
RH AM	No Tactics	10.8	0.0	60.2	71.0	0.0	0.0	0.0
RH AM	Online	10.8	0.9	61.4	73.1	2.4	0.0	0.0
RH AM	Offline	10.8	0.2	60.7	71.7	0.9	0.0	0.0
Midday	No Tactics	13.2	0.0	63.4	76.6	0.0	0.0	0.0
Midday	Online	13.2	0.5	64.0	77.8	1.6	0.0	0.0
Midday	Offline	13.2	0.1	63.7	77.0	0.4	0.0	0.0
RH PM	No Tactics	13.4	0.0	65.6	79.0	0.0	0.0	0.0
RH PM	Online	13.4	0.9	66.6	80.9	1.4	0.0	0.0
RH PM	Offline	13.5	0.0	65.6	79.1	0.1	0.0	0.0
Night	No Tactics	10.4	0.0	56.9	67.2	0.0	0.0	0.0
Night	Online	10.4	0.4	57.4	68.1	0.6	0.0	0.0
Night	Offline	10.4	0.0	56.9	67.2	0.0	0.0	0.0

TABLE A.3 Average passenger travel times (holding and skip-stop). EM-Early Morning, RH-Rush Hour

Time of day	Case	Wait for boarding	Wait for transfer (boarding)	Missed transfer (boarding)	Dwell time	Hold time	Travel time	Wait for transfer (alighting)	Missed transfer (alighting)	Walking time if skipped stop	Total time	Passengers with missed boarding transfers (%)	Passengers with missed alighting transfers (%)
EM	No Tactics	1.6	1.5	1.7	2.8	0.0	12.7	0.7	0.0	0.0	21.0	7.7	1.1
EM	Online	1.6	1.4	0.2	2.8	0.0	12.6	0.8	0.0	0.0	19.4	1.1	0.5
RH AM	No Tactics	1.5	2.3	1.9	2.2	0.0	11.4	2.0	0.1	0.0	21.3	6.0	1.3
RH AM	Online	1.5	2.1	0.2	2.2	0.1	11.3	1.9	0.0	0.0	19.3	0.6	0.6
Midday	No Tactics	2.9	3.3	0.9	3.2	0.0	14.1	2.4	0.2	0.0	26.9	2.9	1.2
Midday	Online	2.6	2.9	0.2	2.7	0.1	13.6	3.1	0.1	0.0	25.4	0.7	0.8
RH PM	No Tactics	4.1	2.7	0.1	3.8	0.0	17.1	1.9	0.0	0.0	29.8	0.4	0.6
RH PM	Online	3.6	2.6	0.0	3.5	0.0	17.0	2.2	0.1	0.0	29.0	0.2	0.6
Night	No Tactics	2.6	2.5	0.0	3.1	0.0	15.8	2.1	0.0	0.0	26.1	0.0	0.0
Night	Online	2.4	2.4	0.0	3.0	0.1	15.7	2.1	0.0	0.0	25.7	0.0	0.0

TABLE A.4 Average passenger travel times (holding and speedup). EM-Early Morning, RH-Rush Hour

Time of day	Case	Wait for boarding	Wait for transfer (boarding)	Missed transfer (boarding)	Dwell time	Hold time	Travel time	Wait for transfer (alighting)	Missed transfer (alighting)	Walking time if skipped stop	Total time	Passengers with missed boarding transfers (%)	Passengers with missed alighting transfers (%)
EM	No Tactics	1.6	1.5	1.7	2.8	0.0	12.7	0.7	0.0	0.0	21.0	7.7	1.1
EM	Online	1.1	1.4	0.7	2.8	0.3	11.5	0.9	0.0	0.0	18.7	3.3	0.7
RH AM	No Tactics	1.5	2.3	1.9	2.2	0.0	11.4	2.0	0.1	0.0	21.3	6.0	1.3
RH AM	Online	1.5	2.2	0.9	2.2	1.6	9.3	2.0	0.1	0.0	19.7	2.6	1.5
Midday	No Tactics	2.9	3.3	0.9	3.2	0.0	14.1	2.4	0.2	0.0	26.9	2.9	1.2
Midday	Online	2.4	2.8	0.5	3.2	0.8	12.1	3.0	0.1	0.0	24.8	1.5	0.6
RH PM	No Tactics	4.1	2.7	0.1	3.8	0.0	17.1	1.9	0.0	0.0	29.8	0.4	0.6
RH PM	Online	3.4	2.6	0.3	3.8	0.5	15.3	2.5	0.0	0.0	28.4	0.8	0.8
Night	No Tactics	2.6	2.5	0.0	3.1	0.0	15.8	2.1	0.0	0.0	26.1	0.0	0.0
Night	Online	2.0	2.2	0.0	3.1	0.4	14.3	2.7	0.0	0.0	24.7	0.0	0.0

TABLE A.5 Average passenger travel times (holding, skip-stop and speedup). EM-Early Morning, RH-Rush Hour

Time of day	Case	Wait for boarding	Wait for transfer (boarding)	Missed transfer (boarding)	Dwell time	Hold time	Travel time	Wait for transfer (alighting)	Missed transfer (alighting)	Walking time if skipped stop	Total time	Passengers with missed boarding transfers (%)	Passengers with missed alighting transfers (%)
EM	No Tactics	1.6	1.5	1.7	2.8	0.0	12.7	0.7	0.0	0.0	21.0	7.7	1.1
EM	Online	1.2	1.4	0.8	2.8	0.4	11.9	0.8	0.1	0.3	19.7	5.4	1.8
RH AM	No Tactics	1.5	2.3	1.9	2.2	0.0	11.4	2.0	0.1	0.0	21.3	6.0	1.3
RH AM	Online	1.5	1.9	0.9	2.0	1.0	11.0	1.2	0.0	0.3	20.8	3.6	1.2
Midday	No Tactics	2.9	3.3	0.9	3.2	0.0	14.1	2.4	0.2	0.0	26.9	2.9	1.2
Midday	Online	2.2	3.4	0.9	2.6	0.5	12.4	3.5	0.1	0.5	26.1	2.2	0.9
RH PM	No Tactics	4.1	2.7	0.1	3.8	0.0	17.1	1.9	0.0	0.0	29.8	0.4	0.6
RH PM	Online	2.4	2.2	1.0	3.4	0.2	16.0	2.5	0.1	0.8	28.6	3.3	1.5
Night	No Tactics	2.6	2.5	0.0	3.1	0.0	15.8	2.1	0.0	0.0	26.1	0.0	0.0
Night	Online	1.6	1.8	1.0	3.0	0.6	14.3	3.2	0.0	0.2	25.7	3.8	0.9

TABLE A.6 Average bus travel times and tactics (holding and skip-stop). EM-Early Morning, RH-Rush Hour

Time of day	Case	Dwell time	Hold time	Travel time	Total time	Stops with hold(%)	Skipped stops(%)	Stops with speedup(%)
EM	No Tactics	11.2	0.0	61.1	72.3	0.0	0.0	0.0
EM	Online	11.2	0.2	61.1	72.5	1.5	4.0	0.0
RH AM	No Tactics	10.8	0.0	60.2	71.0	0.0	0.0	0.0
RH AM	Online	10.5	0.3	60.5	71.3	1.9	4.7	0.0
Midday	No Tactics	13.2	0.0	63.4	76.6	0.0	0.0	0.0
Midday	Online	11.6	0.3	62.3	74.2	1.4	6.4	0.0
RH PM	No Tactics	13.4	0.0	65.6	79.0	0.0	0.0	0.0
RH PM	Online	12.6	0.0	64.9	77.5	0.3	6.4	0.0
Night	No Tactics	10.4	0.0	56.9	67.2	0.0	0.0	0.0
Night	Online	10.1	0.1	56.6	66.9	0.3	2.5	0.0

Table A.7 Average bus travel times and tactics (holding and speedup). EM-Early Morning, RH-Rush Hour

Time of day	Case	Dwell time	Hold time	Travel time	Total time	Stops with hold(%)	Skipped stops(%)	Stops with spee-dup(%)
EM	No Tactics	11.2	0.0	61.1	72.3	0.0	0.0	0.0
EM	Online	10.9	1.6	58.7	71.3	4.1	0.0	30.16
RH AM	No Tactics	10.8	0.0	60.2	71.0	0.0	0.0	0.0
RH AM	Online	10.9	3.1	59.9	73.9	5.6	0.0	33.38
Midday	No Tactics	13.2	0.0	63.4	76.6	0.0	0.0	0.0
Midday	Online	12.8	2.1	61.5	76.4	4.5	0.0	36.2
RH PM	No Tactics	13.4	0.0	65.6	79.0	0.0	0.0	0.0
RH PM	Online	12.8	1.7	61.3	75.8	2.9	0.0	45.13
Night	No Tactics	10.4	0.0	56.9	67.2	0.0	0.0	0.0
Night	Online	10.3	0.8	54.8	66.0	4.5	0.0	39.61

Table A.8 Average bus travel times and tactics (holding, speedup and skip-stop). EM-Early Morning, RH-Rush Hour

Time of day	Case	Dwell time	Hold time	Travel time	Total time	Stops with hold(%)	Skipped stops(%)	Stops with spee-dup(%)
EM	No Tactics	11.2	0.0	61.1	72.3	0.0	0.0	0.0
EM	Online	9.8	2.5	60.8	73.0	4.3	5.9	16.03
RH AM	No Tactics	10.8	0.0	60.2	71.0	0.0	0.0	0.0
RH AM	Online	9.8	3.6	63.8	77.2	5.2	7.3	14.82
Midday	No Tactics	13.2	0.0	63.4	76.6	0.0	0.0	0.0
Midday	Online	10.1	2.5	61.2	73.8	4.9	7.4	16.47
RH PM	No Tactics	13.4	0.0	65.6	79.0	0.0	0.0	0.0
RH PM	Online	12.3	0.6	64.9	77.8	1.6	10.5	22.0
Night	No Tactics	10.4	0.0	56.9	67.2	0.0	0.0	0.0
Night	Online	9.8	1.5	57.0	68.2	2.9	5.6	18.24

ANNEXE B ANNEXES DE L'ARTICLE 2 : *ONLINE STOCHASTIC OPTIMIZATION FOR REAL-TIME TRANSFER SYNCHRONIZATION IN PUBLIC TRANSIT NETWORKS*

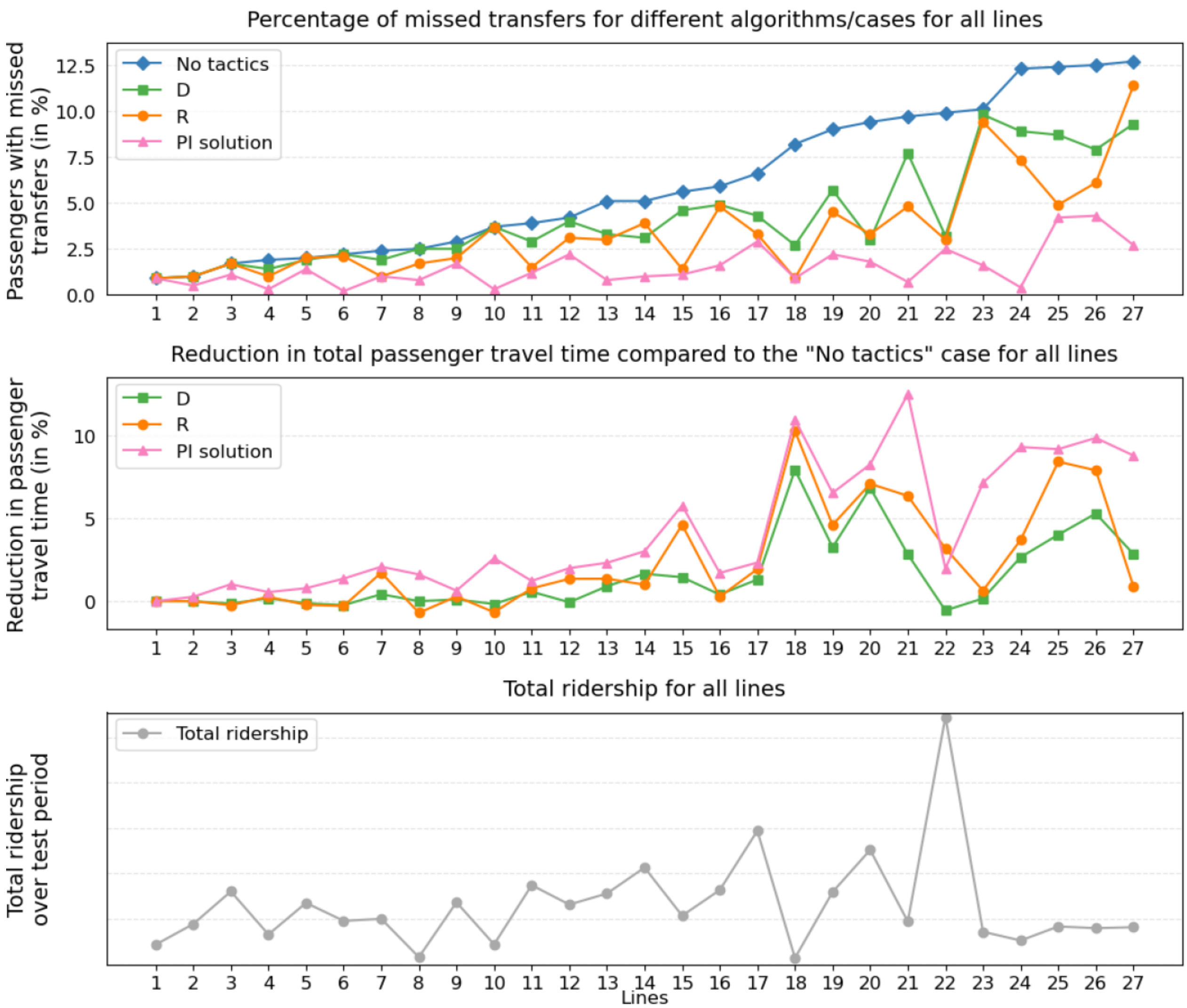

FIGURE B.1 Percentage of passengers with missed transfers, reduction in passenger travel time and total ridership for different algorithms using the hold tactic for 27 lines in the STL bus network.

The lines are sorted by percentage of passengers with missed transfers in the "no tactics" case.

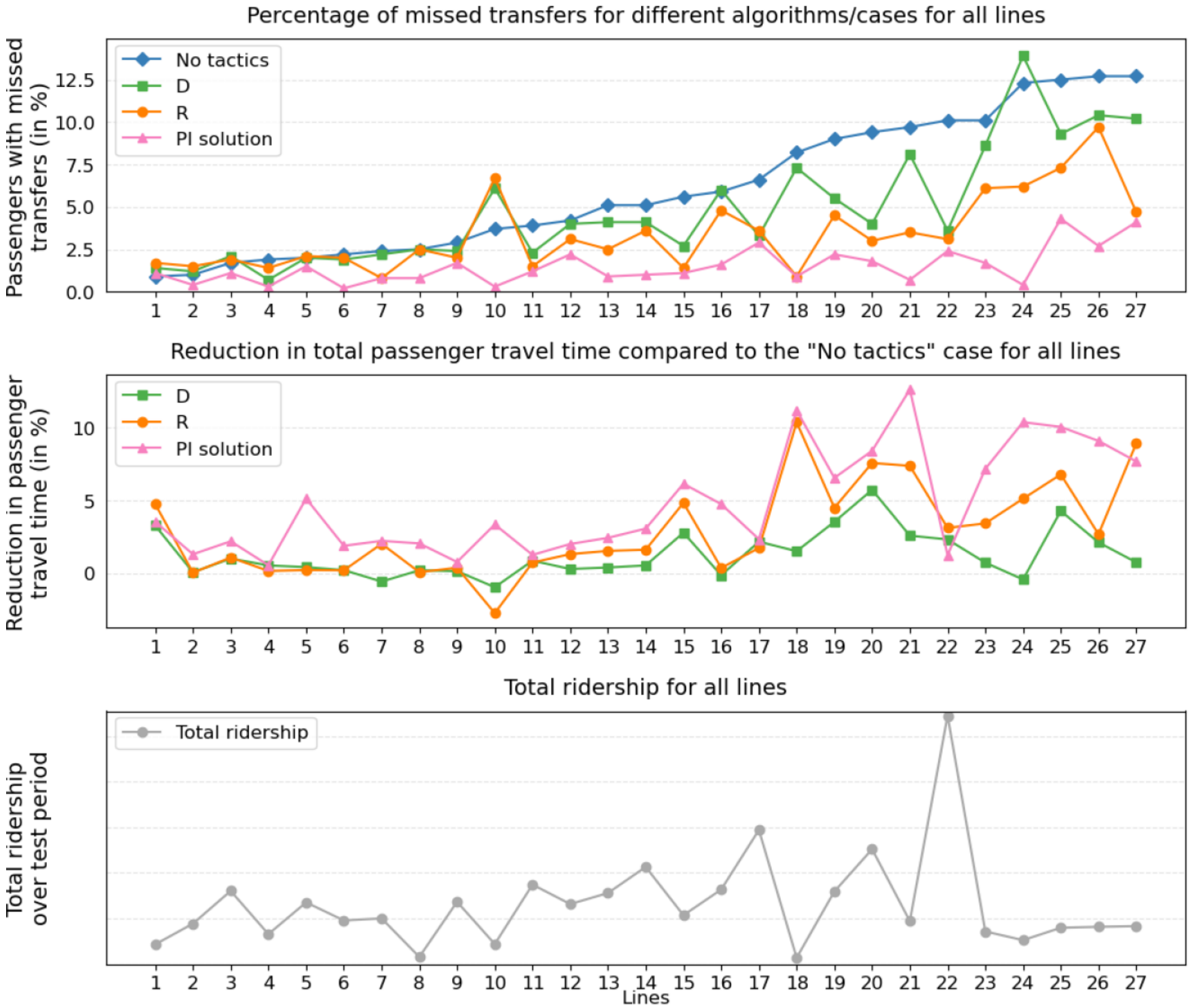

Figure B.2 Percentage of passengers with missed transfers, reduction in passenger travel time and total ridership for different algorithms using the hold and speedup tactics for 27 lines in the STL bus network.

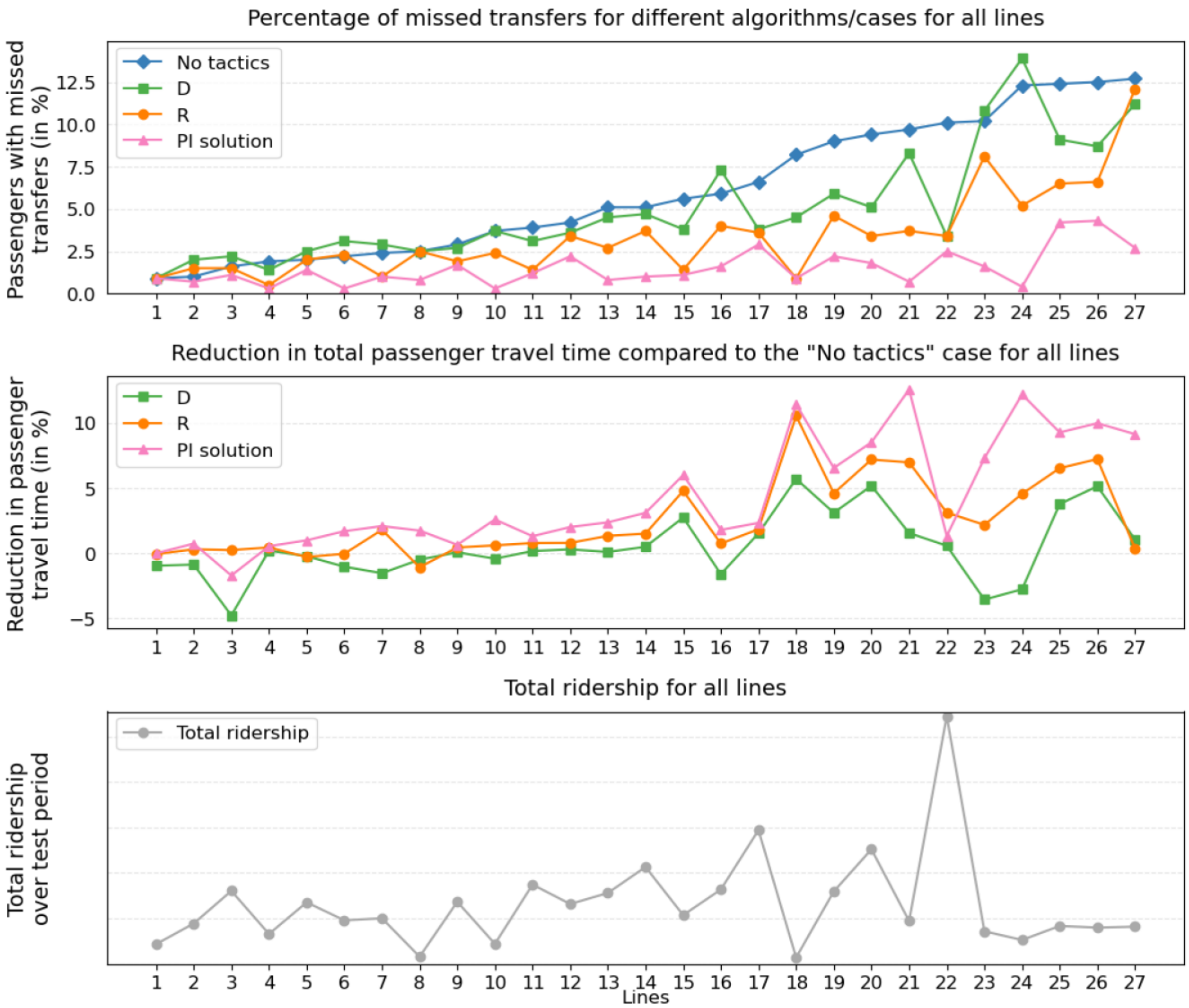

FIGURE B.3 Percentage of passengers with missed transfers, reduction in passenger travel time and total ridership for different algorithms using the hold and skip-stop tactics for 27 lines in the STL bus network.