UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

TRANSFERS IN MASSIVE TRANSPORTATION SYSTEMS UNDER FIXED AND VARIABLE ROUTE SCHEMES.

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN SISTEMAS DE INGENIERÍA

CRISTIAM ANDRÉS GIL GONZÁLEZ

PROFESOR GUÍA:
CRISTIÁN CORTÉS CARRILLO

PROFESOR CO-GUÍA:
PABLO ANDRÉS REY

MIEMBROS DE LA COMISIÓN:
MICHEL GENDREAU
ANTONIO GSCHWENDER KRAUSE
EDUARDO MORENO ARAYA
FERNANDO ORDOÑEZ PIZARRO

SANTIAGO DE CHILE
2020

ABSTRACT
AUTHOR: CRISTIAM ANDRÉS GIL GONZÁLEZ
DATE: JANUARY 2020

TRANSFERS IN MASSIVE TRANSPORTATION SYSTEMS UNDER FIXED AND VARIABLE ROUTE SCHEMES.

In this thesis we present modelling contributions for performing transfer operations in fixed and flexible route transit systems to achieve good solutions in real size instances associated with both problems. In the case of fixed route services, the specific problem addressed in the specialized literature is denoted the Bus Synchronization Timetabling Problem (BST). In the case of flexible route services, our focus is in the Pickup and Delivery Problem with Transfers (PDP-T).

In Chapter 1, we propose an exact method to address the Pickup and Delivery Problem with Transfers (PDP-T). Exact methods in the PDP-T literature were only employed for solving small instances with large computational times due to the complexity of the problem.In this thesis, we developed cutting-edge solution methods to PDP-T based on Column Generation. We propose a new methodology to solve the problem including precedence, route synchronization and capacity constraints, involving several kinds of columns. To the best of our knowledge, this research is innovative and pioneer in addressing the problem under a column based methodology. We improve the algorithm performance applying a branch and price strategy achieving optimality in instances from 5 requests previously using only a column generation framework, up to 30 requests under a B&P framework.

In Chapter 2, we study the Bus Synchronization Timetabling Problem (BST) including bus dwelling times motivated by the night shift of transit system operating in Santiago-Chile. Instead of following timetables, in the operation the dispatching of trips when there are high frequencies it is not necessary to publish timetables, since the waiting times are low (frequency based methods). This approach works well enough during the day operation when demand and frequencies are high. However, during the operation at night, when demand is low and there are a reduced number of services operating with lower frequencies, this type of policy may result in longer waiting times (not only to take the service, but also to transfer at bus stops) and a bad service quality for passengers. To address these inefficiencies, the authorities has defined that all of their night services will operate according to fixed and coordinated timetables. For this, we propose a mixed integer programming model to define the specific timetables and the duration of the vehicle dwelling periods, analyzing the model performance under a set of different real size instances.

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE DOCTOR EN SISTEMAS DE INGENIERÍA
POR: CRISTIAM ANDRÉS GIL GONZÁLEZ
FECHA: ENERO 2020

TRANSFERS IN MASSIVE TRANSPORTATION SYSTEMS UNDER FIXED AND VARIABLE ROUTE SCHEMES.

En este documento, se presentan contribuciones al estado del arte en el modelamiento matemático de sistemas de transporte de bienes y pasajeros tanto en ruta flexible como ruta variable incorporando operaciones de transferencia, ya sea de carga o pasajeros, alcanzando soluciones de calidad en instancias de tamaño real. En la literatura especializada, los sistemas de servicios de rutas fijas y rutas variables, incluyendo transferencias son conocidos como: Bus Synchronization Timetabling Problem (BST) y Pickup and Delivery Problem with Transfers (PDP-T), respectivamente.

En el capítulo 1 de este documento, se aborda el Pickup and Delivery Problem with Transfers (PDP-T) a través de un método exacto. Los métodos exactos en la literatura de PDP-T han sido empleados para solucionar instancias pequeñas con grandes tiempos de cómputo debido a su complejidad en las formulaciones agregando esta característica. En esta tesis, desarrollamos métodos de solución tipo cutting-edge (corte de aristas del poliedro matemático que describe el problema) al PDP-T: basado en generación de columnas. Se propone una nueva metodología para solucionar este tipo de problemas incluyendo restricciones de precedencia, sincronización de rutas y capacidad de vehículos; adicionalmente el problema involucra diferentes tipos de columnas. Según lo que se ha investigado, este trabajo es innovador y precursor en aplicar este enfoque metodológico a este problema específico. El desempeño del algoritmo fue mejorado aplicando diferentes estrategias Branch-and-Price. Esta implementación logró pasar de solucionar 5 solicitudes de servicios en el trabajo preliminar que solo empleaba generación de columnas hasta alcanzar soluciones óptimas en instancias de 30 solicitudes de servicio.

En el capítulo 2 de este documento, se estudia el Bus Synchronization Timetabling Problem (BST) incluyendo tiempos de espera de los vehículos en los paraderos, motivado por la operación nocturna del sistema de buses en Santiago de Chile. En la operación actual, en lugar de regirse a programaciones, el despacho de vehículos se decide en tiempo real de acuerdo con la flota disponible y otras condiciones. Este enfoque funciona adecuadamente durante la operación diurna, cuando la demanda y las frecuencias son altas. Sin embargo, durante la operación de la noche, cuando la demanda es baja y hay un número reducido de servicios operando con frecuencias más bajas, este tipo de política puede resultar en tiempos de espera largos (no solo para abordar la línea de servicio, sino también para transferirse de vehículo) deteriorando la calidad de servicio para los usuarios. Para manejar estas ineficiencias, autoridades definieron que todas sus líneas de servicios nocturnas operarán con programaciones de servicio fijas y coordinadas. Para ello, se propone un modelo de programación lineal mixta con el objetivo de definir programaciones de servicios específicas y los tiempos de espera de los vehículos en los paraderos definidos para tal efecto, analizando el desempeño del modelo bajo un conjunto de diferentes instancias de tamaño real.

Dedicado a la memoria de los que estuvieron en el proceso pero no lograron ver el final.
Abuelita, todos te extrañamos.

Agradecimientos

A mi familia, que desde la distancia siempre me alentó. Mamá no sabes cuantas veces me hizo falta tu chocolate en las mañanas y tu abrazo en las noches. A mis sobrinitos, una sorpresa en el camino, tío una palabra que nunca pensé ser pero que ahora entiendo, los amo, fueron mi motivo al final de este proceso.

A mis profesores, Cristián y Pablo, muchas gracias por su comprensión, paciencia, diligencia y conocimiento en este camino. Nunca encontré un no en sus palabras y fueron una guía apropiada a mi estilo, espero seguirlos encontrando en el futuro. También un agradecimiento para el profe Gendreau, me brindó de su experiencia y me permitió conocer lugares nuevos y muchos amigos más. Agradecimientos especiales a los profesores: Dominique Feillet y Michael Drexl por amablemente compartir los códigos que usan para sus respectivas clases.

A mis amigos y compañeros del DSI, en cada uno encontré aliento y ayuda en su debido momento, lástima que este proceso algún día iba a acabar y aunque no los volveré a ver tan seguido, por supuesto que los veré nuevamente. Y para Eduardo, Renny y Dana, agradecimientos especiales por su compañía: "chancheadas", cine y viajes juntos. Todo un placer compartir con ustedes, que se siga repitiendo en cualquier lado del mundo.

Esta investigación fue realizada bajo la Beca de Doctorado Nacional para Extranjeros sin permanencia definitiva 2014 (Folio: 63140028) de la Comisión Nacional de Investigación Científica y Tecnológica de Chile (CONICYT) y bajo la Beca de Doctorado en el Exterior (convocatoria 679) del Departamento Administrativo de Ciencia, Tecnología e Innovación de Colombia (COLCIENCIAS). Agradecimientos adicionales al Gobierno Canadiense y su Programa de Lideres Emergentes de America (ELAP) por permitirme realizar mi pasantia doctoral en Canada junto al profesor Gendreau.

Acknowledgements

To my family, from the distance they always encourage me. Mom, you don't know how many times I missed your morning chocolate and your hug before go to bed. To my niece and nephew, a surprise in the path, "uncle" is a word that I had never thought so valuable and now I understand all love that it implies, you were my motivation at the end of this process.

To my professors, Cristian and Pablo, thank you so much for your comprehension, patience, practicality and knowledge in this path. I never found a "no" in their words, they were a befitted guidance to my style, I hope to continue finding you in the world in the future. Special thanks to professor Gendreau, who share with me his experience and allowed me meet a lot of new places and friends. Special thanks to professors: Dominique Feillet and Michael Drexl for kindly providing the code that they use in their respective classes.

To my friends and partners from DSI, on each one of you, I found breath and support at due time, it is a shame this process ends and I will not see you so frequently as usual, but of course this is not the last time. Finally, special thanks to Eduardo, Renny y Dana for their companionship: "chancheadas", cinema and travels together. A whole pleasure share with you, we have to keep repeating our adventures anywhere in the world, anytime.

This research was made under Doctoral Scholarship in Chile for Foreigners without definitive permanence 2014 (Folio: 63140028) from the Chilean National Council for Science and Technology (CONICYT) and under Abroad Doctoral Scholarship (call 679) from Colombian Administrative Department of Science, Technology and Innovation (COLCIENCIAS). Special thanks to Canadian Government and their Emerging Leaders in the Americas Program (ELAP) that allowed me do my doctoral internship in Canada and work with professor Gendreau.

Contents

List of Tables

List of Figures

Introduction

Transportation of people and goods is relevant for the economy and quality of life. Efficient transportation systems are essential for promoting sustainable growth of economic activities and global competitiveness (Rais et al., 2014). An efficient transportation system is essential for meeting the transportation requirements within an urban area, in terms of spatial and temporal covering (Liu et al., 2007).

The next generation of transportation systems will aim to integrate various demand management strategies and traffic control measures to actively achieve mobility, environment, and sustainability goals. The possibility of including transfers (of either goods or people) at selected locations in the network is one feature that adds flexibility to the operation of such transportation systems. This operation occurs when a first vehicle drives the request (freight or passenger) to a predetermined location (transfer point). A second vehicle picks up the request at the transfer point and drives it to the final delivery point.

Performing a transfer is an operation that consumes time and may be very costly (Takoudjou et al., 2012) to users and transport managers, so it must be meticulously planned. The literature has shown the benefits of a transfer strategy for transportation systems with different degrees of route flexibility: from a fixed route, such as those typically related to traditional public transport systems, to those happening in the context of a flexible route scheme, such as the well known demand responsive transportation systems. The difference is basically whether the routing is a decision for the model or it becomes a parameter.

Conceptually, a transfer operation could occur in both, fixed and flexible route transit systems. In the case of fixed route services, the coordination of timetables in a low frequency system is a problem directly related to the synchronization of transfer operations, addressed in the specialized literature as the Bus Synchronization Timetabling Problem (BST). In the case of flexible route services, this depends exclusively on the option of performing efficient operations to move passengers or freight among vehicles to cover high-demand minimizing costs, and that is known as the Pickup and Delivery Problem with Transfers (PDP-T).

Transfers in a Bus Synchronization Timetabling problem involves two subproblems: synchronization and timed transfers. In pickup and delivery problems, if passengers are involved, this is usually known as the Dial a Ride Problem (DARP), and in such a case the synchronization became an issue as happens also in the fixed route timetabling described before. We can find many applications arising from observing the facts of transfers and synchronization as main issues, in contexts where route and scheduling have several degrees of flexibility. The resulting applications will be discussed in detail later in this document.

With regard to BST, proven benefits in academic works have reached up to 13% in shorter waiting times at transfers stops (Cevallos and Zhao, 2006), reductions up to 25% in number of operative vehicles (Fügenschuh, 2009) and increments up to 36% in number of transfers (Guihaire and Hao, 2010), in their own contexts.

In Pickup and Delivery Problems with Transfers and their derived problems, academic works have proven benefits in real applications (medium and large instances) reaching up to 37% in travel distances savings (Dessouky et al., 2003), with reductions up to 10.8% in number of operative vehicles (Liaw et al., 1996) and up to 36% of savings in daily operational vehicle costs (Shang and Cuff, 1996). Note that all of these mentioned works were solved employing heuristic methods. Despite these benefits, the combinatorial nature of problems states a computational burden for medium and large instances, where formulations become difficult to solve using exact methods in both kind of problems.

For example, in PDP-T, Fügenschuh (2009) just attained optimality in one case, obtaining gaps between 61% and 238% in other cases through a branch and cut algorithm, observing large gaps, frequent situation in NP-hard problems such as BST. Thus, authors have proposed heuristics and metaheuristics methods to solve large instances, i.e., Cevallos and Zhao (2006) with a genetic algorithm and Guihaire and Hao (2010) proposing a Tabu search algorithm, as illustrative examples.

Exact methods in the PDP-T literature were only employed for solving small instances with large computational times: the best is no more than 75 requests and 4 transfer points running up to 1 CPU time hours (an imposed limit) with average gaps of 33.84% (Masson et al., 2014), showing the difficulties one has to face in real applications when applying exact methodologies. In order to solve larger problems in the context of the PDP-T, heuristics, meta-heuristics or hybrid methods are the prevailing methodologies to find near-optimal (Braekers and Kovacs, 2016) or good solutions, which do not guarantee optimality.

It is worth mentioning two recent promising works, which found fundamental results to improve the gaps in reasonable computational times. One of these is Gschwind (2015) who developed four column generation approaches to synchronize routes in subproblems solving 91% of small and medium size instances (less than 80 request) and 66% of large size instances (greater than 80 request) to optimality. Cortés et al. (2010) proved the computational benefits of implementing a branch-and-cut algorithm (based on Benders decomposition) to solve PDP-T problems. They reported savings of around 90% in CPU time when compared to standard MIP solvers.

For Bus Synchronization Timetabling problems, large instances with exact methods had not been approached (Table 1) until Fouilhoux et al. (2016), who using proper valid inequalities obtained optimality gaps up to 3% for instances of 200 line services and 40 transfer points. Notwithstanding, their better gap was of 8% for instances of 200 line services and 150 transfer points and in many cases the optimal solution was not reached.

An exact methodology can be an appropriate tool to tackle these limitations and suboptimalities. However, exact methods have not been successfully used to address timed transfers in real contexts so far. Bookbinder and Desilets (1992) used an integer programming model into a simulation framework with no success for supporting their work. Moreover, there are no exact methodologies that prove the joint effectiveness of synchronization and timed transfer strategies.

It is clear that these authors (Fouilhoux et al., 2016; Gschwind, 2015 and Cortés et al., 2010) extended the frontier of knowledge. However, there is still much work to be done in both dimensions, mainly in the direction of developing exact methods that will be able to solve real instances to optimality, producing much better benchmarks of real applications, related to PDP-T and BST, highlighting that at present the development of efficient exact methods (through a type of decomposition for example) remain being a theoretical and practical challenge.

The complexity and limitations mentioned above are the main challenges of this dissertation. Hereinafter, we review the literature related to transfers in the context of the Pickup and Delivery problem with transfers (PDP-T, Chapter 1) and the Bus Synchronization Timetabling problem (BST, Chapter 2), we will propose new mathematical formulation and methodologies to deal with both problems and address them to achieve practical results in real applications that allow to contribute with knowledge to the current state of art of the fields.

Chapter 1

The Pickup and Delivery Problem with Transfers

Vehicle routing problems (VRP) in their different versions have been studied intensely from 1959; Dantzig and Ramser (1959) were probably the first authors to formalize models for VRP. In general terms, the decision is to determine the sequence of customers to be visited by each vehicle of the fleet pursuing the minimization of distance, time or cost involved in the operation. Specific real world applications give rise to numerous variants of VRP, specified by different sets of constraints. For example, capacitated VRP (CVRP) addresses the VRP under vehicle capacity constraints, VRP with time windows (VRPTW) requires that the customer must be served within a specific time-lapse, multi depot VRP (MDVRP) corresponds to situations in which vehicles can start their routes at different depots.

The vehicle routing problem with pickup and delivery (VRPPD or PDP) is a specific version of VRP, in which the service required by customers may be both the pickups or deliveries of commodities (Kachitvichyanukul et al., 2015). The PDPTW is a generalization of the vehicle routing problem with both time windows (VRPTW) (Baldacci et al., 2011) and capacity (CVRP) (Bettinelli et al., 2014) constraints. In addition to capacity and time windows constraints, each route must satisfy pairing (origin and destination must be visited by the same vehicle) as well as precedence constraints (the origin location must precede the destination in the route).

The PDPTW is a complex problem to be solved given all the aforementioned conditions (Berbeglia et al., 2011). Some of these constraints could be relaxed allowing origin and destination locations to be serviced by different routes and vehicles (possibility of transfer). However, the addition of the possibility to transfer adds new constraints to the problem, those which controls that the pickup vehicle and the delivery vehicle of each load synchronize (Cortés and Jayakrishnan, 2002). Then, the transfer in the context of a PDPTW establishes a challenge in terms of modeling and complexity. Hereinafter, the Pickup and Delivery Problem with Transfers will be identified as PDP-T.

When the operation involves passengers, the underlying vehicle routing problem is generally known as the Dial-a-Ride Problem (DARP). The DARP is a special case of the Pickup and Delivery Problem with Time Windows (PDPTW) where people are transported instead of goods. As a consequence, new constraints or objectives related to quality of service have to be addressed (Garaix et al., 2011).

In general, one may think of other constraints in connection with the quality of service. The DARP differs from the general PDPTW only in one additional constraint imposing a maximum travel time between the pickup location and the corresponding delivery location (Gschwind and Irnich, 2015). The Dial-a-Ride Problem involving transfers is identified as DARP-T. Since the VRPTW is also a generalization of the VRP in strong sense, then the PDPTW and DARP are NP-hard problems (Qu and Bard, 2013) and consequently, so are PDP-T and DARP-T.

The PDPTW can be stated as the problem of designing a set of routes and assigning a set of requests to a fleet of capacitated vehicles satisfying time windows at locations. A request consists of picking up goods at one location and delivering these goods to another location. Service times may be associated with each pickup and delivery location.

A vehicle is assigned to each request, each vehicle has a limited capacity, and it starts and ends its duty at given locations called start and end depots. The starting and ending location are not necessarily the same, and two vehicles can have different starting and ending depots. Two time windows are assigned to each request: a pickup time window that specifies when the goods can be picked up and a delivery time window that tells when the goods can be dropped off.

A time window is a lapse of time during which the service at a customer must start. If a vehicle arrives before the beginning of the time window, it must wait until its opening time in order to start the service. If a vehicle arrives after the closing of the time window, then the service can not be performed. In general, the objective function of the PDPTW is the minimization of a weighted sum consisting of one or more of the following: the total number of vehicles used or the total number of routes, the total distance traveled by the vehicles, the total time vehicles are busy (waiting and traveling), vehicle waiting times, etc.

Initially the PDPTW also was classified in two categories: single-vehicle and multi-vehicle. In general, pickup and delivery problems are classified into three groups according to origin-destination relation: many-to-many, one-to-many-to-one and one-to-one (Berbeglia et al., 2007) and can also be classified by specific features of the problem such as: static and dynamic (Berbeglia et al., 2011).

In static approaches, all requests are known in advance; typically one day before the realization of the request's service. In the dynamic approach, not all information is available in advance but is revealed during the planning process; associated with the issue of routing is the decision of vehicle positioning in anticipation to future arrivals (Chou et al., 2014).

1.1 The Dial a Ride problem with Transfers.

Traditional public transport systems are in most cases not sufficient to provide a suitable public transport alternative for everyone. For example, in case of elderly and handicapped citizens, the use of a regular system consisting of fixed-route, timetabled services could be impossible (Hall and Peterson, 2013). Then, Dial a Ride services, initially referenced as paratransit services, were born as a personalized mode of transportation offered to those citizens, mainly with low income, who cannot use fixed route bus services and cannot afford private transportation (Karabuk, 2009) to preserve their mobility (Braekers and Kovacs, 2016).

Traditional urban public transport is limited by their accessibility constraints; then, a personalized service could help to avoid the drawbacks of public transport, although keeping a costly and probably not ecological fleet of more personalized vehicles. Indeed, not only people with difficulty mobility were seeking ways to a more flexible transport that can meet their needs, but also that phenomenon could appear in case of general travelers (Zidi et al., 2012). Thus, the concept of paratransit evolved to a more universal definition of Dial a Ride problems (DARP).

The field of DARP has received considerable attention in the literature in the last 30 years (Cordeau and Laporte, 2004; Parragh et al., 2010). Many of the methods published were designed to solve real-world applications. Paquette et al. (2013) reported numerous applications of DARP in large cities: Copenhagen (Madsen et al., 1995), Bologna (Toth and Vigo, 1997), Berlin (Borndorfer et al., 1997), Los Angeles County (Diana and Dessouky, 2004), Brussels (Rekiek et al., 2006), Milan (Calvo and Colorni, 2007), and a not specified mid-size US city (Karabuk, 2009). Although dial-a-ride services are currently provided in many large cities, these services are expected to become even more widely spread in the future due to the aging population (Cordeau and Laporte, 2004; Paquette et al. 2013). Hence, a need for efficient planning tools to make those systems efficient in their operation arises (Braekers and Kovacs, 2016).

Most dial-a-ride services are characterized by the presence of two conflicting objectives: minimizing operating costs together with maximizing user level of service. One way to achieve a balance between these objectives is to impose service quality constraints and minimize cost as the primary objective (Cordeau, 2006). Note that minimization of transportation cost (usually time or distance) may not represent the real needs of users, and that is why it is necessary to evaluate the quality of the service (Rodrigues et al., 2014).

The quality of service for one customer depends on several factors such as user ride time, time window violation and waiting time. In DARP, these factors are either penalized or restricted. For psychological reasons, waiting time for a vehicle is perceived in a different way than travel time on board (except for loading or unloading operations). The latter affects the perceived service quality more than the former (Parragh, 2011). The weighted sum of these factors is commonly known in the literature as the dissatisfaction level (Coslovich et al., 2006).

With regard to time windows, users select a time window of a pre-specified width for the pickup and the drop-off times while the dispatcher determines the most suitable pick-up and(or) drop-off times, depending on the policy of the service provider and the customer request. This is employed by authors like Jaw et al. (1986) and Cordeau and Laporte (2003). Due to the complexity of finding feasible solutions, mainly in large instances, there are several works that consider soft time windows in their implementations. For example, Chen et al. (2016) propose a fuzzy time window approach for their instances.

In DARP-T, transfer scheduling constraints must be imposed in order to ensure that the comfort of the passengers during transfers remains at an acceptable level. This is achieved by avoiding excessive short or long transfer times and also limiting the total ride time between the pickup location and the final destination (Schonberger, 2017). According to Cortés et al. (2010), it is important to quantify the trade off between the advantage of adding flexibility and the extra disutility due to the addition of transfers to the operational scheme. This is one of the reasons why in the last years the literature has taken special interest in this problem.

In the present work, we will focus in the Pickup and Delivery Problem with Transfers (PDP-T) for the one-to-one request relation, multi-vehicle and static approach. In PDP-T, unlike PDPTW, the implicit constraint stating that the Pickup and Delivery points of a given request should be serviced by the same vehicle is relaxed. Next, we will list some of the main academic works related to PDP-T with their main results, findings and novelties.

1.2 Literature review

Given the complexity of DARP-T and PDP-T, most papers developed heuristic methods instead exact ones. For instance, Aldaihani et al. (2003) proposed a mixed offer of fixed routes and PDP, or an integrated DARP, with a three-step heuristic, using local search; the algorithm solves instances with up to 155 requests. In a similar way, Shang and Cuff (1996) proposed an approach taking into account time windows, advanced requests, multi-vehicle, and uncertain fleet size, by means of an insertion heuristic algorithm.

Other authors proposed insertion algorithms. Hickman and Blume (2000), developed a heuristic for scheduling integrated transit trips combining passenger and vehicle goals. The study by Horn (2004) applied Adaptive Large Neighborhood Search (ALNS) to the PDPT with a single transfer, which could benefit paratransit systems. Masson et al. (2017) used ALNS, to solve a two-level vehicle transportation problem, in which, the spare capacity of city buses is used to transport goods to the core of the city. The study solves instances up to 50 customers.

This study is related to Trentini and Mahléné (2010) which makes recommendations to apply this innovative good transportation system into crowded cities. The research of Gørtz et al. (2009) presented a heuristic for both capacitated and uncapacited vehicles. The objective is to minimize the maximum makespan. Simulations and mixed-integer programming are used in the application of DARP-T.

Cortés and Jayakrishnan (2002) simulate a single transfer system, of a demand-responsive transit in order to test its benefits. Posada et al. (2017) explore an exact solution via a mixed-integer program; their objective is to solve an integrated DARP, including timetables. Although the authors come up to an exact solution, they highlight the difficulty of the problem; and the necessity of using heuristics to solve more complex instances.

Masson et al. (2014) solve instances up to 100 requests, using multiple transfers with timetabling constraints but only at fixed points. In this work, an ALNS is developed; comparing the service time with that obtained with no transfers; they find an improvement of 8% in costs with the proposed solutions, noting that the customers can experiment a lower levels of service. They suggest to study the trade-off between transfers and service level in this context.

In contrast, Deleplanque and Quilliot (2013) present a solution with dynamic transfers, where they are allowed to happen in any point, using an insertion heuristic. Kerivin et al. (2008) consider an exact approach, by proposing two mathematical models based on a space-time graph, with discrete periods but no time windows; the problem is solved with a branch and cut algorithm solving instances up to 15 requests. Cortés et al. (2010) present a branch and cut algorithm for a PDP-T; the authors show better results with respect to PDP. They infer that under high demand conditions, transfer operations become more profitable. They could solve instances with up to six requests.

In order to solve larger instances, heuristics, meta-heuristics or hybrid methods are proposed to find near-optimal or good solutions. In many cases, the solutions can be far from optimal. Petersen and Ropke (2011) present a ANLS heuristic solving instances with 982 requests. Ghilas et al. (2016b) also present an ALNS method for a PDP with synchronization and time-windows constraints, implemented for up to 100 requests.

The above literature suggest that the use of mathematical models to solve the PDP-T problem could be a research opportunity: from both a theoretical and a practical standpoint. Up to date, PDP-T can be solved exactly within an acceptable time frame in case of relatively small problem instances.

Table 2 summarizes our literature review for the PDP-T including DARP-T; the academic works are arranged chronologically. The third column provide information regarding the flexibility of the transport system considered; fourth and fifth columns show the solution approach employed and the objective function. Columns six, seven and eight are related to features: vehicle capacities, time windows and if it considers split loads or nor, which give complexity to problem. The next three columns: real (R), benchmark (B) and example (E) explain the nature of the data instances to test the proposed algorithms. Moreover, the number of requests and number of transfer points are shown to give an idea of the size of the treated instances.

Table 1.1: Literature review for the Pickup and Delivery Problem with Transfers.

Authors	PDP-T	DARP-T	Transport system	Solution method	Objective Function	Vehicle capacity	Time windows	Split	Number of requests	Transfer points	R	B	E	Data instances based on:
Liaw et al. (1996)		x	Hybrid	Simulated annealing	max number of requests serviced per unit time	Homogeneous	Hard		[50,120]	fixed routes ∗	x		x	Ann Arbor Transportation Authority, Michigan
Shang and Cuff (1996)	x		Pure flexible	Look ahead heuristic	min vehicle expense, tardiness and travel time	Uncapacitated	Hard		300	all ∗∗	x			Health Maintenance Organization
Hickman and Blume (2000)		x	Hybrid	Scheduling Heuristic	max passenger utility	Heterogeneous	Hard		3500	fixed routes ∗	x			Transit service in Houston, Texas.
Oertel (2001)	x		Pure flexible	Tabu-search	min travel distance, tour length and handling cost	Homogeneous	Without		[9,40]	2			x	
Aldaihani et al. (2003)		x	Hybrid	Tabu-search	min travel distance and travel time of passengers	Heterogeneous	Hard		114	fixed routes ∗	x			Lancaster County, California
Mitrović-Minić, S. and Laporte (2006)	x		Pure flexible	Random multi-start and Insertion heuristic	min total route lengths	Uncapacitated	Hard	x	[50,100]	[1,4]			x	
Thangiah et al. (2007)	x		Pure flexible	Insertion heuristic	min number of required and vehicles, number of shipments to be rescheduled travel distance by vehicles	Uncapacitated	Hard	x	159	9		x	x	Shang and Cuff (1996)
Lin (2008)	x		Pure flexible	CPLEX	min vehicle fixed cost and variable travelling cost	Uncapacitated	Hard		[27,100]	1 ∗∗	x		x	Hong Kong, China
Kerivin et al. (2008)	x		Pure flexible	Branch-and-cut	min vehicle-related total cost	Homogeneous	Without	x	[5,15]	all ∗∗			x	
Dondo et al,. (2009)	x		Pure flexible	CPLEX	min routing cost of vehicles	Heterogeneous	Hard		20	2			x	
Hall et al. (2009)		x	Hybrid	CPLEX	min routing cost of vehicles	Heterogeneous	Hard		4	3			x	City of Norrköping, Sweden
Cortés et al. (2010)	x		Pure flexible	Branch-and-cut	min the total travel time of vehicles	Heterogeneous	Hard		6	1			x	
Petersen and Ropke (2011)	x		Pure flexible	ALNS	min fixed vehicle cost, km cost road toll, cross-dock handling cost hourly vehicle cost.	Homogeneous	Hard		[500-1000]	1	x			Danish transporter of flowers
Qu and Bard (2012)	x		Pure flexible	GRASP and ALNS	min number of vehicles and travel distance	Heterogeneous	Hard		25	1			x	
Masson et al. (2013)	x		Pure flexible	ALNS	min travel distance	Homogeneous	Hard		[50,193]	all ∗∗	x	x		Mitrović-Minić, S. and Laporte (2006)
Deleplanque and Quilliot (2013)		x	Pure flexible	Insertion heuristic and Constraint propagation	min travel distance	Homogeneous	Without	x	[20,65]	all ∗∗			x	
Chou et al. (2014)	x		Pure flexible	Simulation	min courier workload and customer waiting time.	Uncapacitated	Soft		24	3	x			Courier services in a large hospital
Masson et al. (2014)		x	Pure flexible	ALNS	min travel distance	Homogeneous	Hard		[55,193]	all ∗∗	x			Services to disabled or elderly individuals
Masson et al. (2014)		x	Hybrid	Branch-and-cut-and-price	min travel distance	Homogeneous	Hard		[25,75]	[2,4]			x	
Ghilas et al. (2016)	x		Hybrid	CPLEX	min routing cost of vehicles and transfer cost	Heterogeneous	Hard		12	[1,3]			x	
Ghilas et al. (2016b)	x		Hybrid	ALNS	min routing cost of vehicles and transfer cost	Heterogeneous	Hard		100	[1,3]			x	
Ghilas et al. (2016c)	x		Hybrid	ALNS and Sample average approximation	min routing cost of vehicles and recourse expected costs	Heterogeneous	Hard		≤ 40	[1,3]			x	
Schonberger (2017)		x	Pure flexible	Memetic	min arrival times	Homogeneous	Without		[6,100]	1			x	
Posadaet al. (2017)		x	Hybrid	CPLEX	min operational cost of used vehicles and fixed cost of routes	Heterogeneous	Hard		[1,6]	4			x	City of Norrköping, Sweden

∗∗ In column: Transfer point, the word "all" means that all arigin and destination points of requests are considered as possible transfer points.
∗ In column: Transfer point, the words "fixed routes" means that a subset of bus stops in all service lines considered as possible transfer points.

The remainder of the chapter is organized as follows. Section 1.3 formally defines the PDP-T and introduces a the formulation of the problem. Section 1.4 describes the Branch and price method for the PDP-T including multiple column definitions; the pricing problem is an elementary shortest path problem with resource constraints. In section 1.5, we explain the branching strategies applied to the PDP-T; in Section 1.6, we show some computational experiments and results. Finally, in the last section, we state relevant conclusions and challenges in future work.

1.3 Branch and Price methodology for PDP-T

The Pickup and Delivery Problem with Transfers addressed in this work is stated as follows: for a given fleet of capacitated vehicles and a set of requests (each one with associated pickup and delivery locations), find a set of static vehicle routes to satisfy such customer requests allowing to perform freight or passenger transfer between vehicles at minimum total cost, where costs are related to route distances and fixed cost. The Transfer operation occurs when a first vehicle drives the request to a predetermined location (transfer point). A second vehicle picks up the request at the transfer point and drives it to the delivery point.

To formulate the PDP-T problem, we define a directed graph $G(N, A)$ with the set of nodes N (including depots, transfer points, pickup and delivery nodes) and the arc-set A. For $i, j \in N$, $(i, j) \in A$ denotes the arc from i to j. $T \subset N$ is the subset of transfer points where it is allowed to transfer freight or passengers between vehicles.

The set of pickup-and-delivery requests is denoted as R and indexed by $r = 1, \ldots, |R|$; every element of R has associated a quantity q_r (could be freight or passengers) to be picked up from $p(r) \in N$ and delivered to $d(r) \in N$ respecting time windows for each type of node: namely $\left[a_{p(r)}; b_{p(r)}\right]$ and $\left[a_{d(r)}; b_{d(r)}\right]$. V is the set of vehicles indexed by $v = 1, \ldots, |V|$; for each vehicle v, Q_v is the vehicle capacity, f^v is the fixed costs of use, τ_{ij}^v is the time required to go from node i to node j, C_{ij}^v are the arc transit costs, $o(v)$ and $d(v)$ are the initial and final depot, all of them belonging to set N.

Rais et al. (2014) introduced a three-index formulation for P-DPT in a very general form involving not only heterogeneous vehicles, but also multiple transfer points. This MIP model has a polynomial number of variables and a polynomial number of constraints based on the size of the problem instance. For the worst-case scenario, Rais et al. (2014) showed that the total number of variables is bounded by $|A| \cdot |V^4|$, i.e., the number of variables is $O(n^6)$, where $n = |N|$ and the total number of constraints is bounded by $O(n^5)$. This shows that the three-index formulation for the PDP-T is a very NP-hard combinatorial optimization problem.

Rais et al. (2014) solve only small instances (10-node and 14-node) taken from Li and Lim (2001) as a benchmark, even adjusting the time windows to only 50% of the originals and changing the load quantity required to serve each request. They found that, compared with non-transfer models, these transfers models typically require more CPU-time for solving benchmark instances, which should be expected due to the larger number of constraints and variables than the non-transfer model, but have the potential to yield better optimal values and more flexible solutions.

In subsection 1.3.1 we define the concept of columns specifically defined for our problem: the Pickup and Delivery Problem with Transfers. In Subsection 1.3.2, is the set partitioning problem formulation for the PDP-T is presented. In section 1.3.3, we describe the Pricing problem, then in subsection 1.3.4, we explain the associated Elementary Shortest Path Problem with Resource Constraints and finally describe the branching strategies in subsection 1.3.5.

1.3.1 Column definitions

To formulate the problem, we define three types of feasible routes to address the Pickup and Delivery Problem with Transfers: columns from depot to depot, columns from depot to transfer point and columns from transfer point to depot. Next, we define these columns.

The set K_1 are routes from depot to depot that do not contain the transfer point at any case. Then if a request is picked up, it is mandatory to be dropped before arriving to the depot (Figure 1.1). The set K_2 are routes from depot to transfer point; unlike columns of type K_1, in these columns some of the requests are served partially, requiring a complement provided by columns of type 3 defined next (see Figure 1.2). Finally, set K_3 contains routes from transfer point to depot for delivery of requests served previously by a route of set K_2 (Figure 1.3).

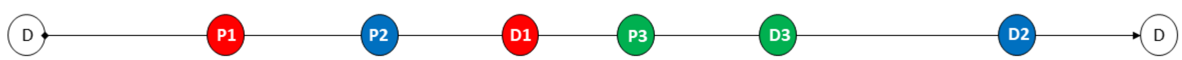

Figure 1.1: A column of type K_1

Figure 1.2: A column of type K_2

Figure 1.3: A column of type K_3

These three column types allow us to model the Pickup and Delivery Problem with Transfers. The main idea is that those requests that are served through the transfer point may be picked up by one columns of type K_2 and then delivered by one columns of type K_3. As shown in Figure 1.4, a vehicle picks up requests 1, 2 and 3. The freight (passenger) are then dropped at transfer point; note that they do not have to be delivered together by the same vehicle to the final destination; in this case, they are split in two vehicles: request 2 in a first route and requests 1 and 3 in a second route. Eventually, they could be dropped by the same vehicle or delivered by three different vehicles depending on the actual setting.

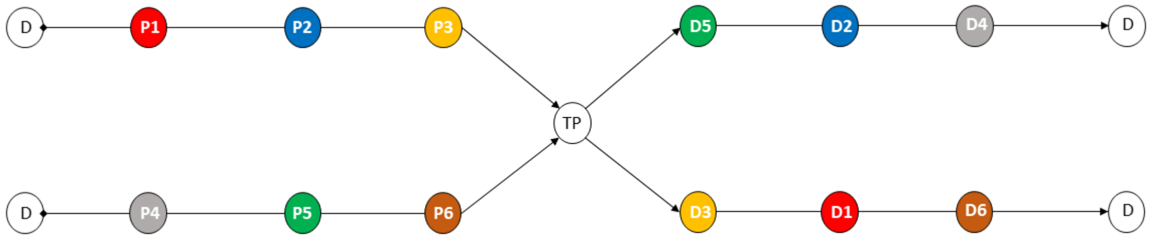

Figure 1.4: PDP-T solution example.

This option to drop freight or passengers through the transfer point has also a significant advantage, which is to create partial routes from depot to transfer point and viceversa, making the master problem smaller in number of variables. This is because it is not necessary to create entire routes from depot to depot passing through the transfer point. It is enough to pair a couple of columns (a K_2 with a K_3) reducing processing times due to a fewer number of combinations.

It is relevant to highlight that columns of type K_2 and K_3 can travel directly from or to the transfer point, respectively. For example, in Figure 1.5 one of the columns of type K_2 is not paired with a column of type K_3, then it goes empty until the depot from the transfer point. In Figure 1.6 we present a kind of mirrored case, in which the column of type K_3 that is not paired with a column of type K_2 starts the route empty from the depot to the transfer point and then picks up some freight or passengers after the transfer point going to both pickup and delivery points.

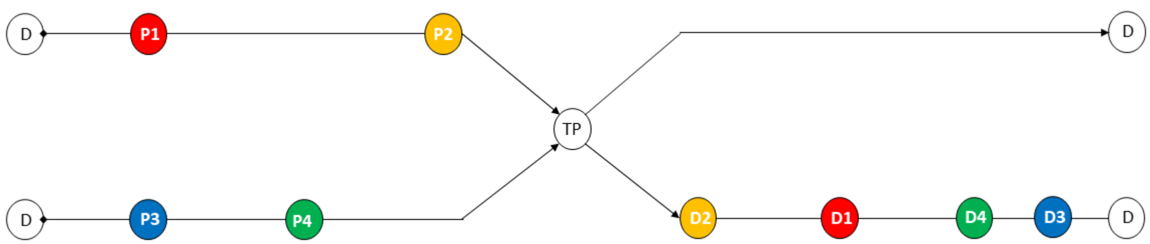

Figure 1.5: Not paired column of type K_2 example.

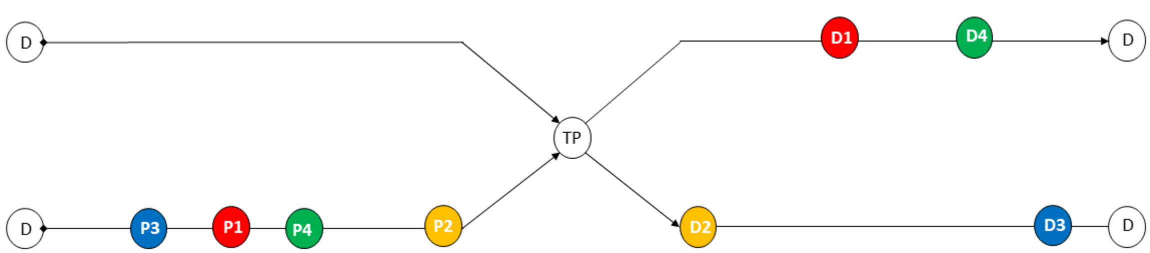

Figure 1.6: Not paired column of type K_3 example.

1.3.2 Formulation for the PDP-T

Let $K = K_1 \cup K_2 \cup K_3$ be the set of all routes. For each route $k \in K$, let C_k be the cost of the route, let C^z be the cost of a direct trip from depot to transfer point and viceversa and let a_i^k be the value that indicates if a node $i \in P \cup D$ belongs to route k or not, B_k denotes the latest (earliest) time that a vehicle serving route K_2 (K_3) can reach (leave) the transfer point, respectively and the service time at transfer node d_r.

Let X_k be a binary variable equal to 1 if and only if route $k \in K$ is used in the solution. The integer variable Z counts the number of direct trips between the depot and the transfer point. Let $Y_{k\hat{k}}$ be a binary variable equal to 1 if feasible paths $k \in K_2$ and $\hat{k} \in K_3$ are served by the same vehicle and finally, η_i be a binary variable equal to 1 if request i is served through the transfer point (changing vehicle), 0 if not.

Thus, the PDP-T can then be formulated as the following set partitioning program:

$$\min \quad F_{PDP-T} = \sum_{k \in K} C_k \cdot X_k + C^z \cdot Z \tag{1.1}$$

Subject to:

$$(\alpha_i) \quad \sum_{k \in K} a_i^k \cdot X_k = 1 \quad \forall i \in P \cup D \tag{1.2}$$

$$(\omega_i) \quad \sum_{k \in K_2} X_k \cdot (a_i^k - a_{i+n}^k) - \eta_i = 0 \quad \forall i \in R \tag{1.3}$$

$$(\theta_i) \quad \sum_{k \in K_3} X_k \cdot (a_i^k - a_{i+n}^k) + \eta_i = 0 \quad \forall i \in R \tag{1.4}$$

$$(\gamma_k) \quad \sum_{\hat{k} \in K_3} Y_{k\hat{k}} \leq X_k \quad \forall k \in K_2 \tag{1.5}$$

$$(\lambda_{\hat{k}}) \quad \sum_{k \in K_2} Y_{k\hat{k}} \leq X_{\hat{k}} \quad \forall \hat{k} \in K_3 \tag{1.6}$$

$$(\pi) \quad \sum_{k \in K_2} X_k + \sum_{\hat{k} \in K_3} X_{\hat{k}} - 2 \cdot \sum_{k \in K_2} \sum_{\hat{k} \in K_3} Y_{k\hat{k}} = Z \tag{1.7}$$

$$(\psi_i) \quad \sum_{\hat{k} \in \tilde{K}_3(i)} B_{\hat{k}} \cdot X_{\hat{k}} - \sum_{k \in K_2(i)} (d_r + \tilde{B}_k) \cdot X_k \geq 0 \quad \forall i \in R \tag{1.8}$$

$$(\sigma_{k\hat{k}}) \quad Y_{k\hat{k}} = 0 \quad \forall \hat{k} \in K_3 \ k \in K_2 \quad if \ B_{\hat{k}} - B_k < d_r \tag{1.9}$$

The objective function (1.1) minimizes the cost of the chosen routes, while constraints (1.2) ensure that every request is served once. Constraints (1.3)-(1.4) strengthen the algorithm obtaining useful information for branching through the η_i variables. Constraints (1.5)-(1.6) control that routes belonging to K_2 can be paired at most with a route of type K_3 and viceversa. Constraint (1.7) computes the value of variable Z. Constraint (1.8) cuts solutions where a request picked up by a route from set K_2 cannot be delivered by a route from set K_3 (route K_2 does not arrive before than a route from set K_3 that starts from the transfer point),to compute this constraint we define the sets: $\tilde{K}_2(i)$ and $\tilde{K}_3(i)$ which include columns serving request i through the transfer point. Constraint (1.9) forbid using the same vehicle for a route of type K_2 and a route of type K_3 which do not synchronize.

To compute the dual variables and to run the pricing problem in the first iteration, it is necessary to have an initial set of columns for the set partitioning problem. In this case, we build a set of simple columns: for example, columns of type K_1 for every request, in which the route starts from the origin depot and contains the pickup point, the delivery point and finally ends at the destination depot. Columns of type K_2 contain the pickup point between the origin depot and the transfer point and Columns of type K_3 contain the delivery point between the origin depot and the transfer point.

1.3.3 Pricing problems

The pricing problems for each set of routes, K_1, K_2 and K_3, are obtained from model (1.1)-(1.9). The pricing problem for routes of type K_1 is similar to those found in the state of the art literature in PDP-T applications like Feillet (2010) and Ropke & Cordeau (2009). Nonetheless, constraint (1.2) applies not only on pickup nodes also on delivery nodes, due to the existence of routes of type K_2 where delivery points associated with requests are not necessarily part of the route. The price of a column of type K_1 is written in expression (1.10), where the pricing associated with the arcs of these columns is assigned according to expression (1.11).

$$\bar{C}_k = C_k - \sum_{i \in R} \alpha_i \cdot a_i^k \tag{1.10}$$

$$\bar{C}_{ij} = \begin{cases} C_{ij} - \alpha_i & \forall i \in P \cup D, \quad \forall j \in N \\ C_{ij} & \forall i = depot, \quad \forall j \in N \end{cases} \tag{1.11}$$

Pricing problems associated with routes of type K_2 and K_3 cannot be expressed completely through the pricing of arcs as developed in the pricing problem for routes of type K_1. The reason is that the dual costs of constraints (1.5), (1.6) and (1.9) depend on the new route to be created, in both types of routes: K_2 and K_3. To complete the dual costs of constraints (1.5) and (1.6), we use the fundamental definition of dual cost: it quantifies the improvement in the objective function due to an additional unit of the resource in the constraint. In these cases, we must think if the new constraint will be either active or not. For instance, a new column K_2 requires a new constraint (1.5); that will be active only if there is a column K_3 not paired and able to synchronize properly.

A relevant issue is to quantify the savings in the objective function due to the addition of the synchronization variable $Y_{k\hat{k}}$ into the model. Note that $Y_{k\hat{k}}$ will be equal to 1, and there is a column that belongs to K_3, namely $\hat{k}$, available to pair, the vehicle associated to this column started empty its route until reaching the transfer point, which increases variable Z in one unit, objective function is penalized in C_z, which is an estimate of the savings of adding the new columns of type K_2. For new columns of type K_3, an analogous procedure is conducted but related to constraint (1.6). Analytically:

$$\forall \text{ new column } k \in K_2 \ \ if \begin{cases} \exists \hat{k} \in K_3 : \ \sum_{k \in K_2} Y_{k\hat{k}} < 1 \\ B_k + d_r < B_{\hat{k}} \end{cases} \quad then : \gamma_k = -C_z, \ otherwise : \gamma_k = 0$$

$$\forall \text{ new column } k \in K_3 \ \ if \begin{cases} \exists k \in K_2 : \ \sum_{\hat{k} \in K_3} Y_{k\hat{k}} < 1 \\ B_k + d_r < B_{\hat{k}} \end{cases} \quad then : \lambda_k = -C_z, \ otherwise : \lambda_k = 0$$

Note that for constraint (1.8) its dual cost (terms $+\psi_i \cdot (B_k + d_r)$ and $-\delta_i \cdot B_k$ for routes of types K_2 and K_3, respectively) requires a value for parameter B_k which is not known until the end of the dynamic programming algorithm that need those values to be performed. We compute these terms at the end of the process of creating new routes, we use these values along with the dual cost of constraints (1.5), (1.6) and (1.7), to determine if the new route must be added into the model or not. The objective function of pricing problem for routes of type K_2 is as follows:

$$\bar{C}_k = C_k - \sum_{i \in R} \alpha_i \cdot a_i^k - \sum_{i \in R} \omega_i \cdot (a_i^k - a_{i+n}^k) + \sum_{i \in R} \mathbb{1}_{[a_i^k - a_{i+n}^k = 1]} \psi_i \cdot (B_k + d_r) - \pi - \gamma_k \quad (1.12)$$

Where:

$$\bar{C}_{ij} = \begin{cases} C_{ij} - \alpha_i - \omega_i + \mathbb{1}_{[a_i^k - a_{i+n}^k = 1]} \psi_i \cdot (B_k + d_r) & \forall i \in P, \quad \forall j \in N \\ C_{ij} - \alpha_i + \omega_i & \forall i \in D, \quad \forall j \in N \\ C_{ij} - \pi - \gamma_k & \forall i = TP, \quad \forall j \in N \end{cases} \quad (1.13)$$

In the pricing problem for routes of type K_3, new routes are created similar to routes of type K_2 but "reversing" the time and the interpretation of pickups and delivery nodes. In this case, we start from depot backwards handling delivery points as pickup points and viceversa. The price of a route of type K_3 and its respective version in terms of arcs is shown in expressions (1.14) and (1.15).

$$\bar{C}_k = C_k - \sum_{i \in R} \alpha_i \cdot a_{i+n}^k - \sum_{i \in R} \theta_i \cdot (a_i^k - a_{i+n}^k) - \sum_{i \in R} \mathbb{1}_{[a_i^k - a_{i+n}^k = -1]} \psi_i \cdot B_k - \pi - \lambda_k \quad (1.14)$$

$$\bar{C}_{ij} = \begin{cases} C_{ij} - \alpha_i - \theta_i & \forall i \in P, \quad \forall j \in N \\ C_{ij} - \alpha_i + \theta_i - \mathbb{1}_{[a_i^k - a_{i+n}^k = -1]} \psi_i \cdot B_k & \forall i \in D, \quad \forall j \in N \\ C_{ij} - \pi - \lambda_k & \forall i = TP, \quad \forall j \in N \end{cases} \quad (1.15)$$

1.3.4 Elementary Shortest Path Problem with Resource Constraints

Subproblems are the shortest path problems with precedence, capacity and time windows constraints. We solve an elementary shortest path problem with resource constraints (ESPPRC) through a dynamic programming algorithm, in the form of a label correcting algorithm. For routes of type K_2 we assume these routes are *left justified*, i.e., each route performs the sequence as early as possible. Similarly, for routes of type K_3 we assume that these routes are *right justified*, i.e., each route performs the sequence as late as possible.

A label can be viewed as a vector of data states (features and resources) of a partial route. In a label L, we store the following information: the current node (last extended) $\eta(L)$, the arrival time at the node $t(L)$, the load of the vehicle after visiting current node $l(L)$, the accumulated cost $C(L)$, the opened set which is the set of requests that started but have not been completed $O(L)$, and the set of unreachable requests $U(L)$. In this contest a request is unreachable: if its pickup node has already been visited on the partial path or if we go straight from $\eta(L)$ to the pickup node i, the time window at that node would be violated.

Labels are built to extend the route through the arcs that starts at the end node of the partial path, the algorithm constructs all feasible paths in the graph by starting with the partial path containing only the origin depot until reaching the destination node (destination depot or transfer point). When a label is extended, it is necessary to check all the resources: vehicle capacity, precedence and time windows for feasibility. Moreover, its features for comparison have to be updated. In order to accelerate the dynamic programming algorithm we can discard dominated labels. In fact, to speed up the process of label creation, in columns of types K_2 or K_3 if value of variable η_i is negative and request i belongs to opened set $O(L)$ we do not allow to extend the label. In this way we assure to add columns really useful for the solution process.

1.3.4.1 Label extension

For columns of types K_1 and K_2, the extension of a label L along arc $(\eta(L),\ j)$ is evaluated only if constraints: $t(L) + \tau_{\eta(L),j} \leq b_j$ (late arrival) and $l(L) + q_j \leq Q$ (vehicle capacity) can be satisfied. After the extension, we have to check that every delivery node associated with the pickup nodes belonging to opened set $O(L)$ satisfy time windows constraints (namely $t(L') + \tau_{\eta(L'),i+n} \leq b_{i+n} \quad \forall i \in O(L)$), otherwise, such an extension is dismissed and we continue with the next candidate node. As columns of type K_3 are built in an opposite way: from transfer point to depot, these constraints are modified in the following way: $l(L) - q_j \leq Q$ and $t(L') + \tau_{\eta(L'),i} \leq b_{2n+2} - a_i \quad \forall i \in N$, this allows that pickup nodes to be treated as delivery nodes, and viceversa.

To guarantee the feasible extension of columns of type K_1, one of the three conditions (1.16) have to be fulfilled: (a) j cannot belong to the unreachable set if j is a pickup node, (b) if j is a delivery node its respective pickup node has to belong to the opened set and (c) if j is the destination depot, the opened set must be empty. Analytically,

$$\begin{aligned} 0 < j \leq n \quad &\wedge \quad j \notin U(L) \\ n < j \leq 2n \quad &\wedge \quad j - n \in O(L) \\ j = 2n+1 \quad &\wedge \quad O(L) = \varnothing \end{aligned} \tag{1.16}$$

For columns of type K_2, the conditions to extend nodes (1.17) are similar to that applied to columns of type K_1, except with condition (c). As in columns of types K_2 and K_3 it is not mandatory to have both nodes related to the request (namely pickup and delivery nodes) in the same route (column), the opened set could be not empty. Here the condition (c) can be stated as: if j is the transfer point, the respective delivery node of every pickup node belonging to the opened set must be reachable.

$$\begin{gathered} 0 < j \leq n \quad \wedge \quad j \notin U(L) \\ n < j \leq 2n \quad \wedge \quad j - n \in O(L) \\ j = 2n+2 \quad \wedge \quad \forall i \in O(L), \quad i + n \notin U(L) \end{gathered} \tag{1.17}$$

The conditions for columns of type K_3 (1.18) that have to be satisfied (one of them) are : (a) j cannot belong to the unreachable set if j is a delivery node, (b) if j is a pickup node its respective delivery node has to belong to the opened set and (c) if j is the origin depot, the respective pickup node of every delivery node belonging to the opened set must be reachable.

$$\begin{gathered} n < j \leq 2n \quad \wedge \quad j \notin U(L) \\ 0 < j \leq n \quad \wedge \quad j + n \in O(L) \\ j = 2n+2 \quad \wedge \quad i \notin U(L) \quad \forall i + n \in O(L) \end{gathered} \tag{1.18}$$

After extending a node j in a label L, a new label L' must be created. For columns of types K_1 and K_2, the new label is set as follows: (a) set j as extended node $\eta(L')$, (b) compute arrival time $t(L')$, if j is reached before time window service it is set equal to its early arrival (a_j), (c) calculate the vehicle load ($l(L')$), (d) totalize travel length ($c(L')$), (e) update the unreachable set of the label ($U(L')$), j is added if it is a pickup node, otherwise remains unchanged. And (f) update the opened set of the label ($O(L')$), j is added if it is a pickup node and it is removed if it is a delivery node. Analytically,

a. $\quad \eta(L') = j;$

b. $\quad t(L') = \max\left\{a_j,\ t(L) + \tau_{\eta(L),j}\right\}$

c. $\quad l(L') = l(L) + q_j$

d. $\quad c(L') = c(L) + \tau_{\eta(L),j}$

e. $\quad U(L') = \begin{cases} U(L) \cup \{j\} & if\ j \in P \\ U(L) & if\ j \in D \end{cases}$

f. $\quad O(L') = \begin{cases} O(L) \cup \{j\} & if\ j \in P \\ O(L) \backslash \{j - n\} & if\ j \in D \end{cases}$

For columns of type K_3, remember that they are built in an opposite way: from transfer point to depot, the new label is set as follows: (a) set j as extended node $\eta(L')$, (b) compute arrival time $t(L')$, if j is reached before temporary time window service it is set equal to its early arrival $(b_{2n+2} - b_j)$, (c) calculate the vehicle load $l(L')$, load q_j is subtracted because we are treating pickups as deliveries and viceversa, (d) totalize travel lenght $(c(L'))$, (e) update the unreachable set of the label $(U(L'))$, if j is a delivery node its respective pickup node is attached, otherwise remains unchanged. For (f) update the opened set of the label $(O(L'))$, if j is a delivery node its respective pickup node is attached and it is removed if it is a pickup node. Analytically,

$a.\quad \eta(L') = j;$

$b.\quad t(L') = \max\left\{b_{2n+2} - b_j,\ t(L) + \tau_{\eta(L),j}\right\}$

$c.\quad l(L') = l(L) - q_j$

$d.\quad c(L') = c(L) + \tau_{\eta(L),j}$

$$e.\quad U(L') = \begin{cases} U(L) \cup \{j-n\} & if\ j \in D \\ U(L) \quad if\ j \in P \end{cases}$$

$$f.\quad O(L') = \begin{cases} O(L) \cup \{j-n\} & if\ j \in D \\ O(L) \setminus \{j\} \quad if\ j \in P \end{cases}$$

1.3.4.2 Dominance rules

Ropke and Cordeau (2009) propose dominance rules for the extended labels of the Pickup and Delivery Problem with Time Windows. The main objective of these rules is to avoid enumeration, in every iteration of all possible combinations to create columns with negative dual costs. With these rules, if label (L_1) dominates label (L_2), the latest has to be discarded and consequently, no more extension will be performed from L_2. These rules make the algorithm to be computationally efficient. Analytically, label (L_1) dominates label (L_2) if:

$$n(L_1) = n(L_2) \quad \wedge \quad t(L_1) \leq t(L_2) \quad \wedge \quad c(L_1) \leq c(L_2)$$
$$U(L_1) \subseteq U(L_2) \quad \wedge \quad O(L_1) \subseteq O(L_2)$$

However, these dominance rules are only valid if the dual cost matrix satisfies the triangular inequality. According to Ropke and Cordeau (2009): $C_{ij} \geq C_{ik} + C_{kj} \quad \forall i \in P, \quad \forall j \in D, \quad \forall k \in N$. In our case, the triangular inequality does not hold: $C_{ij} \geq C_{ik} + C_{kj} - \alpha_k \quad \forall i \in P, \quad \forall j \in D, \quad \forall k \in N$, because dual variable α_k is free given that constraint (1.2) is an equality. As we described in section 1.3.3, the pricing problem, in constraint (1.2) applies not only on pickup nodes, but also on delivery nodes, due to the existence of routes of type K_2 where is not mandatory to serve the delivery points related to the pickups of requests observed in these columns.

Then, our dominance rules change the condition to $O(L_1) = O(L_2)$ instead of $O(L_1) \subseteq O(L_2)$, which implies that we cannot achieve the computational efficiency of the original algorithm (without transfers):

$$n(L_1) = n(L_2) \quad \wedge \quad t(L_1) \leq t(L_2) \quad \wedge \quad c(L_1) \leq c(L_2)$$
$$U(L_1) \subseteq U(L_2) \quad \wedge \quad O(L_1) = O(L_2)$$

1.3.5 Branching strategies

Three branching strategies where implemented. These branching rules are applied hierarchically: the methodology branches on the number of vehicles first. Then, we branch on variables η_i and finally we branch on arcs. Next, we detail each branching procedure:

1. Branching on the number of vehicles: if the solution of the current node is fractional, we verify the total number of vehicles employed. If this quantity is fractional, the branching process is performed on the number of vehicles $V = \sum_{k \in K_1} X_k + \sum_{\hat{k} \in K_3} \sum_{k \in K_2} Y_{k\hat{k}} + Z$. If V is fractional, then the next two new branches are added to the tree adding one of the following equations directly to the master problem (CMP) at the node:

$$\sum_{k \in K_1} X_k + \sum_{\hat{k} \in K_3} \sum_{k \in K_2} Y_{k\hat{k}} + Z \geq \lceil V \rceil \tag{1.19}$$

$$\sum_{k \in K_1} X_k + \sum_{\hat{k} \in K_3} \sum_{k \in K_2} Y_{k\hat{k}} + Z \leq \lfloor V \rfloor \tag{1.20}$$

2. Branching on η_i: constraints (1.3) and (1.4) are strengthening the problem as they were created to give valid information to the model through the variable η_i, recalling that η_i is a binary variable that let us know if request i is served through the transfer point or not. If η_i is fractional, we create two branches: $\eta_i \leq 0$ and $\eta_i \geq 1$ are imposed in every branch. The variable to be branched is chosen in order of the request set.

3. Branching on arcs: this is a rule that has become a standard in the context of the VRPTW (Feillet, 2010). This strategy selects an arc (i, j) creating two branches: first, a branch where the arc cannot belong to the solution; to implement this, we have to remove columns using arc (i, j) from the set partitioning problem, making sure that in the pricing problem no column using this arc is generated. Second, we create a branch where the arc is forced into the solution; to implement this, we have to remove columns using ingoing arcs to j and outgoing arcs from i, leaving the unique possibility that node i goes directly to node j. Here, the arcs are explored based in their ordinality until we find and arc with fractional solution to be branched on.

The tree is traveled in the order in which the branching nodes were created. No time or gap condition to stop was used. The framework provided by Professor Dominique Feillet was employed for the implementation of these branching rules.

1.4 Computational Results

The performance of our proposed methodology was evaluated by means of the generation of new instances, created by selecting randomly subsets of requests from available instances of Qu and Bard (2012). We solve our B&P approach using an Intel core i7-8550U computer with 4 GB of RAM memory and 16 GB of Intel memory Optane. The algorithm was implemented in C++ and the linear programs solved using CPLEX 12.4. To start the column generation at the root node we use as feasible solution the following: For each request, we generate three routes; a route in K_1 that starts from the origin depot, visits the request's pickup point, its delivery point and the destination depot, a route in K_2 that starts from the origin depot, visits the request's pickup point and the transfer point and a route in K_3 that starts from the transfer point, visits the request's delivery point and the destination depot.

1.4.1 Effect of branching on eta

With respect to B&P with and without the η_i branching rule, we notice that their performance were similar in all features but time. In tables 1.2 and 1.3 we can observe the following: optimality gaps are zero for most instances. In 6 out of 10 instances, less than 10 iterations were needed and the optimal solution was found at the root node. In all instances, the root LB solutions were very close to final lower bound (LB), which shows that modeling approach was able to find valuable columns (belonging to optimal base) at root node.

Table 1.2: B&P results for 10-requests instances including eta branching

Ins.	Z_{IP}	LB	GAP(%)	Root LB	Nodes B&B	Iterations	Time		Columns		Column type		
							Root	Total	Root	Tree	1	2	3
B1	2601.50	2601.50	0.00	2601.50	2	3	3285	3872	458	0	10	278	170
B2	2518.30	2445.72	2.97	2317.63	17	26	1391	14609	360	495	15	392	448
B3	2510.10	2510.10	0.00	2510.10	2	9	3556	8908	770	249	10	745	264
B4	2358.40	2358.40	0.00	2358.40	2	7	35680	44083	972	0	10	590	372
B5	2358.20	2358.20	0.00	2358.20	2	6	12711	14751	445	0	10	250	185
B6	2487.00	2485.77	0.05	2432.27	35	16	35418	55861	1733	14	10	1,420	317
B7	2450.90	2450.90	0.00	2450.90	2	6	11830	14795	802	0	10	602	190
B8	2459.50	2459.50	0.00	2459.50	2	9	20567	28531	1643	45	10	1,400	278

Table 1.3: B&P results for 10-requests instances without include eta branching

Ins.	Z_{IP}	LB	GAP(%)	Root LB	Nodes B&B	Iterations	Time		Columns		Column type		
							Root	Total	Root	Tree	1	2	3
B1	2601.50	2601.50	0.00	2601.50	2	3	3872	7061	458	0	10	278	170
B2	2518.30	2344.17	7.43	2337.12	36	54	2461	33497	356	137	10	226	257
B3	2510.10	2453.62	2.30	2453.62	20	57	2560	15663	443	139	11	383	188
B4	2358.40	2358.40	0.00	2358.40	2	7	37215	95503	971	1	10	590	372
B5	2358.20	2358.20	0.00	2358.20	2	6	19481	25801	445	0	10	250	185
B6	2487.00	2485.77	0.05	2432.27	35	58	35159	119210	1367	94	10	1,036	415
B7	2450.90	2450.90	0.00	2450.90	2	6	24344	25052	802	0	10	602	190
B8	2459.50	2459.50	0.00	2459.50	2	37	35272	75023	1643	130	10	1,371	392

Here, the main finding is that the η_i branch rule decreased the CPU times by up to 61.97% for all instances and on average 49.68%. Clearly, saved time occurs in the tree where the η_i branching rule has a direct impact. Regarding execution times, comparing instances with 10 requests against instances with 12 requests, although there is not enough evidence to conclude about differences, we observed that the number of nodes in the search tree and the number of iterations for the column generation procedure are naturally higher for the instances with 12 requests as expected.

1.4.2 Characteristics of optimal solutions

In general, when analyzing instances, we note that columns of type K_1 they were hardly created in column generation procedure, therefore they do not appear in optimal solutions. We believe that this is due to the fact that columns of type K_2 contain similar columns to those of type K_1 (with the difference that they end at the transfer point instead of the depot) and a column K_2 can be synchronized with a column K_3 saving costs eventually because of returning to the depot from the transfer point (C_z). Thus, columns of type K_2 are preferred to columns of type K_1. The algorithm, in every pricing problem mostly creates columns of type K_2 in the initial iterations and subsequently creates some columns of type K_3, at the current node to complement previous created columns of type K_2.

Table 1.4 presents the column configuration for 12-requests instances in PDP-T. The information shown in table 1.4 is: the instance name (Ins.), the number of vehicles used (Veh.), the number of columns generated by the branching procedure at the root and through the tree (Columns), the number of each type of columns generated in the complete algorithm (Column type) and the number of each type of columns in the solution (Column Sol.).

Table 1.4: Columns configuration for 12-requests instances in PDP-T

Ins.	Veh.	Columns		Column type			Column Sol.		
		Root	Tree	1	2	3	1	2	3
A1	3	989	0	12	650	327	1	2	2
A2	3	240	123	13	231	119	0	3	3
A3	2	148	260	12	190	206	0	2	2
A4	2	233	116	12	97	240	0	2	2
A5	2	170	126	12	74	210	1	1	1
A6	2	156	31	12	77	98	0	2	2

Solutions used the same number of columns for both columns of types K_2 and K_3, pairing between them to minimize the number of vehicles used. Also, we observe that columns of types K_2 and K_3 where no load was released or picked up at transfer point (similar to columns of type K_1) were present in optimal solutions, meaning that going through transfer point was performed to synchronize with other columns and generate savings in objective function.

Table 1.5 presents the optimal solution of one of these instances, specifically instance A1. The information shown in table 1.5 is: the column number (Col.), the type of the column (Type), the length of the route that the column represents (Length), the load carried when the vehicle that serves the column goes through the transfer point (Load), the moment that the column arrives to transfer point, in case of columns of type K_1 and K_2, or departs from transfers point, in case of column type K_3 (Time at depot) and the nodes are included in the route that the column represents (Nodes).

Table 1.5: Optimal solution for instance A1 in PDP-T

Col.	Type	Length	Load	Time at depot	Nodes
1	K_1	394.0	0	394.0	0 P_1 D_1 0
2	K_2	528.5	42	528.5	0 P_9 P_{12} P_6 D_{12} D_9 D_6 P_2 TP
3	K_3	804.5	0	594.3	TP P_3 D_3 P_5 D_5 0
4	K_2	524.3	55	524.3	0 P_4 P_8 P_7 P_{10} D_7 P_{11} D_8 TP
5	K_3	385.6	-97	613.2	TP D_{10} D_2 D_{11} D_4 0

The optimal solution for instance A1 is composed of five columns: one column of type K_1, two columns of type K_2 and two columns of type K_3. Columns 2 and 4 arrive to transfer point at minutes 528.5 and 524.3, respectively the earliest possible time, allowing them to synchronize with columns 3 and 5 that depart at minutes 594.3 and 613.2, respectively the latest possible time. Three vehicles are employed to fulfill properly the requests: the first one operates the route of column 1, the second one operates the routes of columns 2 and 3 and the third one operates the routes of columns 4 and 5.

First vehicle serves just column 1 with request 1, that is a column of type K_1. Second vehicle picks up 42 load units along the path of column 2 that will release at transfer point (pickup point of request 2) and continue serving requests of column 3. Unlike, third vehicle, it seems to pick up 55 unit loads (pickup points of requests 4 and 10) but it will not release them at transfer point because this vehicle is same that serves respective delivery points of these requests, actually this vehicle is going through transfer point to pick up the load released from second vehicle and fulfill the request 2.

1.4.3 Benefits of transfers

Tables 1.6, 1.7 and 1.8 show the results of solving 12-requests instances considering Pickup and Delivery Problems with and without Transfers. The information shown in these tables is: the instance name (Ins.), the value of the best integer solution (Z_{IP}), the best lower bound (LB) and relative gap between Z_{IP} and LB (GAP), the number of nodes in the search tree (Nodes B&B), the number of iterations for the column generation procedure (Iterations), the number of columns generated for the branching procedure at the root and through the tree (Columns), the execution times (in seconds) for the root node (Root time) and the complete algorithm (Total time), the number of each type of columns generated through the complete algorithm (Column type) and number of vehicles used.

Table 1.6: B&P results for 12-requests instances in PDP-T

Ins.	Z_{IP}	LB	GAP(%)	Nodes B&B	Iterations	Time		Columns		Vehicles
						Root	Total	Root	Total	
A1	2636.9	2636.9	0	2	5	22855	23338	989	0	3
A2	2796.9	2550.5	9.66	20	56	3779	14003	240	123	3
A3	2531.2	2531.2	0	25	67	1945	29869	148	260	2
A4	2459.7	2459.7	0	37	69	5372	74157	233	116	2
A5	2316.5	2316.5	0	63	57	2951	14373	170	126	2
A6	2475.5	2475.5	0	13	108	8717	13745	156	31	2

Table 1.7: B&P results for 12-requests instances in PDP

Ins.	Z_{IP}	LB	GAP(%)	Nodes B&B	Iterations	Time		Columns		Vehicles
						Root	Total	Root	Tree	
A1	3290.20	3290.20	0.00	2	8	1315	1573	18	67	4
A2	2848.60	2844.60	0.14	49	77	4178	12345	170	146	3
A3	2749.20	2746.30	0.11	11	24	12878	14837	421	38	3
A4	2681.70	2681.70	0.00	23	23	7545	9800	343	31	3
A5	2668.40	2668.40	0.00	2	5	12216	12487	787	12	2
A6	2686.60	2686.60	0.00	2	4	54134	54353	264	12	3

From tables 1.6 and 1.7, we observe the following: the PDPT optimality gaps are zero for all instances but instance 2, while the PDP optimality gaps are almost zero for all instances, noting that only instances 2 and 3 report gaps of 0.14% and 0.11%, respectively. We require a smaller number of nodes in the search tree and number of iterations for the column generation procedure for the PDP than for the PDPT. This is probably due to the flexibility granted by the transfer point that is reflected in several new columns that require time in the pricing problem. However, the number of columns generated for the branching procedure in the root and the tree is not significantly different. In general, fewer vehicles were used in PDPT for satisfying the same demand.

Table 1.8: Comparison of results for the PDP and the PDPT 12-requests instances

Ins.	Z_{IP}			CPU Time (s)		
	PDP-T	PDP	Δ.(%)	PDP-T	PDP	Δ.(%)
A1	2636.9	3290.2	-19.86	23338	1573	1383.66
A2	2796.9	2848.6	-1.81	14003	12345	13.43
A3	2531.2	2749.2	-7.93	29869	14837	101.31
A4	2459.7	2681.7	-8.28	74157	9800	656.70
A5	2316.5	2668.4	-13.19	14373	12487	15.10
A6	2475.5	2686.6	-7.86	13745	54353	-74.71

Table 1.8 shows a comparison between the Pickup and Delivery Problem and Pickup and Delivery Problem with Transfers solved with the B&P technique. The best integer solution (Z_{IP}) for the PDPT is lower than the PDPT best integer solution, going from 1.81% and up to 19.86% of difference among instances. The CPU times to solve the PDP outperforms the CPU time to solve the PDPT model for all but instance 6, spending up to 14 times more seconds to solve the problem.

1.5 Conclusions and future work

We have introduced a branch-and-price algorithm for the Pickup and Delivery Problem with Transfers. Up to date, we solve exactly instances with up to 24 nodes (12 requests), which is larger than Rais et al. (2014) who addressed the problem through a MIP and the benchmark of Cortés et al. (2010). Given the current state of the art for the exact solution of vehicle routing problems with time windows instances solved to optimality are larger. Nevertheless our work lays relevant basis for further research on this challenging problem.

Our solution approach essentially exploits the fact that routes passing through a transfer point could be built in two parts: from and to the depot (this corresponds to columns of types K_2 and K_3). Solving the problem with a proper pairing of these two types of columns, we can save a great number of variables from being introduced in the master problem reducing the complexity of the problem. The presented set partitioning formulation of the PDP-T is exponentially bounded in the size of the problem, initially by constraints (1.5), (1.6), (1.8) and (1.9) that depend on the number of columns of type K_2 or K_3.

The CPU time required for solving the model increases sharply when the size of the problem increases; this is evidenced when we try to solve instances with more than 12 requests. Constraints (1.5) and (1.6) continue being an issue. It is relevant to think if it is possible implement new strategies in the way to turn the exponential bound to a polynomial bound. Regardless of that, the current research in both software, and solution techniques for models, is quite promising.

Research should focus on the development of refined procedures to improve the performance of our algorithm in the Elementary Shortest Path Problem with Resource Constraints: first, to involve innovative and faster methods to build columns that skip, at least partially, the usual combinatorial difficulty of ESPPRC. Second, to create more valuable columns to reduce the degeneracy typical in vehicle routing problems (Baldacci et al., 2011). Third, to add strategies to explore nodes in the branch and price method and to parallelize the algorithm. Additionally, we should develop heuristics to get better initial solutions and new valid inequalities for this problem.

Chapter 2

The Bus Synchronization Timetabling Problem with Dwelling Times

The planning process behind a fixed route transit system operated with buses is complex. Ceder (2007) framed such a scheme into four subproblems: network route design, timetable development, vehicle scheduling, and crew scheduling. The efforts in the literature generally treat each problem separately to simplify the challenges of each stage. In particular, setting timetables is a key process in order to coordinate and feed the other stages of a transit planning process.

The Bus Timetabling Synchronization problem (BST) consist of finding an optimal bus schedule for a transit system that minimizes transfer time for passengers (Ceder et al., 2001); the objective is to set departure times for all trips associated with a set of predefined lines over the planning periods (day or night). BST is associated with the satisfaction of passengers as well as with the efficiency of transit agencies operation (Wu et al., 2012). Usually, problems involving synchronization are complicated due to the large set of binary and discrete variables involved in the formulations. In fact, Ibarra-Rojas and Rios-Solis (2012) proved that BST is NP-hard.

According to Ibarra-Rojas and Muñoz (2016), in a transit system operated by buses, there are different lines that share overlapping route segments or common stops. In these cases, passengers could benefit from a proper synchronization between the different bus lines at common stops/zones as they travel using more than one line through the implementation of transfers. A set of passengers require to transfer from one transit vehicle to another in order to fulfill their trips. In some cases passengers need to travel using different modes (e.g. bus, train, metro, shuttle services).

Transfers improve the transit network connectivity by increasing the possible number of travel paths, enhancing operational flexibility and efficiency, which could help the system become cost effective, guaranteeing high service quality and efficient resource utilization (Eranki, 2004). However, it is relevant to synchronize timetables carefully, since uncoordinated transfers can produce magnified opposite effects.

Unfortunately, transfers involve various drawbacks related to the discomfort of alighting a bus and boarding a new vehicle (passenger orientation and walking between stops to transfer), negative perception due to extra waiting for transferring from a vehicle plus other sources of delay (Eranki, 2004). In addition to this, there are the inconveniences generated by transferring itself, known as “transfer penalty” and which corresponds to the inconveniences related to transfers (not associated with walking or waiting time), i.e. the inconvenience of the interrupted trip, the uncertainty of the conditions that will be found in the next stage, etc. (Currie, 2005; Raveau et al., 2014). Suitable transfer policies could benefit users and transport companies, since concentration of passengers on major routes enables higher speeds, smaller investments in equipment (Lee and Schonfeld, 1991) and a better city coverage.

However, many transit agencies do not have efficient ways to create timetables that provide an optimal synchronized service (Cevallos and Zhao, 2006) allowing users to transfer softly, mainly due to a lack of tools and methodologies to formulate and solve the mathematical problems. Timetable synchronization can reduce transfer waiting times, which improves the service quality. A poor transfer coordination, on the other hand, can reduce the number of passengers using transit as a result of switching to competitor modes (Teodorović and Lucić, 2005).

The remainder of the paper is organized as follows: section 2.1 reviews related studies, section 2.2 describes the Bus Rapid Transit system in Santiago of Chile: Metropolitan Mobility Network (MMN). In section 2.3, we reformulate the BST model proposed by Eranki (2004) including dwelling time decisions (BST-DT), enhancing the model with service inequalities agreed with the authorities. Additionally, in Section 2.3 new valid inequalities to strengthen the new model are stated and a preprocessing operation to reduce the size of the instances is described. We present experimental results in Section 2.4, showing the impact of transfer coordination, whether dwelling times are included or not compared with current system performance. Finally, conclusions and future research are addressed in Section 2.5.

2.1 Literature review

The time that passengers spend waiting at transfer stations is a relevant indicator of the service quality provided by transit service (Wu et al., 2012). No waiting at transfers is considered to be optimal for passengers, but in practice passengers experiment waiting times at transfers different from zero due to the uncertainties involved in the functioning of transit systems. There are two main ways of operating a transit system: by frequency or interval (when demand and frequencies are high; without published timetable), and by fixed schedules (when demand and frequencies are low; with publication of a timetable). And there are basically two alternatives for reducing waiting times on transfers in a transit network for these ways of operation (Bookbinder and Desilets, 1992).

The first one, associated with frequency operation, consists of reducing headways, which involves an increase in the number of buses and drivers required to operate the routes. This alternative tends to fail: headways on very frequent lines are inherently unstable; when a bus falls slightly behind schedule, it tends to pick up more passengers than what is planned, which slow down that bus further, until it eventually bunches with the trailing bus (Zhao et al., 2006); this phenomenon also increases the variability of passengers travel times.

In the same way, the variability of passenger's travel times between stops along with variations in passenger demand, lead to an increase in bus headway variance and a consequent deterioration of both the magnitude and variability of average waiting times (Delgado et al., 2012). Since the subjective valuation of waiting time tends to be higher than the valuation of time of any other trip component (access time, in-vehicle time) (Boardman et al., 2001), transfer coordination becomes more important and pertinent.

The second alternative, associated with fixed schedule operation, is to perform transfer coordination. This means to conduct a joint synchronization and timing transfers, including synchronization plus a holding policy. The average waiting times experienced when transferring are a direct consequence of schedule synchronization. To maintain the transfer reliability at a reasonable level, safety margins called slack times are built into the schedule in order to absorb the randomness of the system (Sun and Schonfeld, 2016). For these reasons, it may be beneficial to perform timing transfers in order to improve the synchronization of bus arrivals at transfer stations (Kim and Schonfeld, 2014).

2.1.1 Timetabling synchronization

With regards to the problem of synchronization of vehicles, in the literature we find different approaches. Ceder and Tal (2001) stated that synchronization is the most difficult task of transit schedulers defining synchronization as the simultaneous arrival of two vehicles (different services) at a transfer node. Later, Eranki (2004) extended the concept of synchronization as the arrival of two vehicles (different line services) considering a maximum allowable waiting time range associated with the passengers at the transfer zone.

Domschke (1989) explores the combination of regret heuristics, improvement algorithms and simulated annealing, for coordination of the schedule by minimizing the sum of transfer waiting times. The author tested a branch and bound algorithm finding that large problems (more than 20 routes) cannot be solved by means of exact methods. Similarly, Bookbinder and Desilets (1992) propose scheduled time departures based on an optimization model, with fixed routes and headways, pursuing the minimization of total waiting time of passengers.

Ceder and Tal (2001) and Ceder et al. (2001) aim to maximize the number of synchronization events with a heuristic method, which is a common methodological aspect found in the approaches when a large network is studied. Eranki (2004) proposed a similar approach, also solved heuristically, but considering time windows at the transfer zone, in order to find the maximum number of simultaneous arrivals.

Cevallos and Zhao (2006) focus on an evolutionary algorithm to minimize transfer times of the whole system by modifying the existing timetable, taking into account no-holding policies, random travel times and fixed headways. Liu et al. (2007) proposed a bus timetabling, based on Ceder and Tal (2001) applied to a regional operation, by using a taboo search algorithm. Teodorović and Lucić (2005) developed a fuzzy procedure, a combination of Ant Colony and Fuzzy logic, where transferring passengers were treated like fuzzy numbers. Wu et al. (2012) minimized the transfer waiting times, including weights for transfer stations. In a first stage, they proposed a mixed integer nonlinear programming model, and developed a genetic algorithm for its practical application.

Aksu et al. (2014) studied the trade-off between transfer times and the waiting time to start the trip considering heterogeneous headways. They showed a reduction of total waiting time also showing some drawbacks, such as the difficulty to remember the different schedules in a heterogeneous setting and the cost of implementation and information is also an issue.

Other works related of synchronization of transit vehicles, are described in Shrivastava and Dhingra (2002), Verma and Dhingra (2006), Chakroborty et al. (1995) who addressed the problem of synchronization between train lines and bus feeders. Wong et al. (2008), Kang et al. (2015), Kang et al. (2015b), Niu et al. (2015), Wang et al. (2015) and Wu et al. (2015) addressed the problem of synchronization between different train and metro lines.

2.1.2 Timed transfers

A timed transfer is an event in which two vehicles from different service lines coincides at a transfer station, due to holding action performed by at least one of the vehicles for a predefined amount of time, which triggers an intentional synchronization allowing transferring of passengers between vehicles. The main problem here is to determine which vehicles should be held, where, when and for how long they should be retained (Eberlein et al., 2001). The objectives behind dwelling are two: to prevent bus bunching by stabilizing frequency services (in a frequency based system for example) and to provide service connectivity between origins and destinations at transfers points.

With regard to the stabilization of frequency services, some strategies are tested along with dwelling to regularize headways like: boarding limits (Delgado et al., 2009; Delgado et al., 2012), rolling horizon approaches (Sánchez-Martínez, 2012), speed adjustment (Daganzo and Pilachowski, 2011), station skipping (Cortés et al., 2011). To improve service connectivity, many transit agencies consider significant dwelling times in the timetables, which increases ridetimes, consequently increasing cycle times and waiting times of non-transferring passengers. To keep the service level standard, more vehicles and drivers are required implying increasing operational costs.

Optimal dwelling time minimizes expected waiting times, involving a tradeoff between service connectivity and frequency (Zhao et al., 2006). In airports the tradeoff is sensitive; it affects competitively the costs due to the high investments and operational costs required for an airplane and the staff. In bus systems, the passengers quality of service is affected, and that is why many researchers have contributed in this area. Some works related to timed transfer strategy in airports are Odoni (1987), Richetta and Odoni (1993), Vranas et al. (1994), Hoffman and Ball (2000) and Ball et al. (2003).

Real applications of timed transfer strategy in the context of transit systems in medium and large cities have been found in the past: Kyte et al. (1982) in Portland-Oregon, Bakker et al. (1988) in Austin-Texas. Clever (1997) outlines the increased integration of timed transfer terminals with other transit services such as rail in European countries of Germany, Austria and Switzerland. Nevertheless, Rapp and Gehner (1967) were the first authors who deal with this aspect of the problem; they described a coordinated four-stage interactive graphic process for operational transit planning of Basel Transit System in Switzerland.

Abkowitz et al. (1986) simulated delivery strategies at a transfer station. The authors found that holding strategies depend on the consistency of headways; thus, if headways have a high variability, a no holding strategy is recommended. Lee and Schonfeld (1991) considered an optimization problem for the holding time in a transfer point, in the formulation they accounted for a delay cost that included a missed connection cost and a waiting cost for passengers.

Other studies considered as an objective the cost for coordination of the system: Chowdhury and Chien (2001), Chowdhury and Chien (2002) and Ting and Schonfeld (2007). They estimate the impact on passengers, resources and operation of the network. Dessouky et al. (1999) explore the impact of using technology for communication and tracking buses on a network; they found through simulations that the use of technology is most advantageous when holding time is close to zero and headways are large.

Other methods include the use of continuous estimates, such as a probabilistic model for synchronizing a bus network (Kim and Schonfeld, 2014) considering several decisions regarding the type of headway, relaxed schedules, vehicle size, frequency, flexibility to demand, and slack times. The study showed that optimal synchronizations were reached with a common factor of headways or integer-ratio headways.

Table 1 summarizes our literature review for BST, both for synchronization and timed transfers. The academic works are arranged chronologically. Fourth and fifth columns exhibit the number of line services and number of transfer points used in each study to give an idea of the size of the instances tested. Columns six and seven are related to some problem features: if holding times and headways are decision variables in the model or not. Columns R, B, E are the initials of the words: real, benchmark and example respectively, to explain the nature of the data instances where the algorithms introduced were tested (Column 11).

Table 2.1: Literature review for Bus Synchronization Timetabling problem.

Authors	Solution method	Objective Function	Number of line services	Number of Transfer points	Nature of holding	Nature of headways	R	B	E	Data instances based on:
Abkowitz et al. (1986)	Simulation	min waiting times of users at transfer nodes	2 one-way	1	Variable	Variable			x	
Domschke (1989)	Branch and bound Heuristics and Simulated annealing	min waiting times of users at transfer nodes	6 two-way	20*		Parameter			x	
Lee and Schonfeld (1991)	Numerical analysis	min total transfer costs	2 one-way	1	Variable	Variable			x	
Bookbinder and Desilets, (1992)	Integer programming and simulation	min disutility users for waiting time	NA	9 or 20	Variable	Parameter	x		x	Winnipeg, Canada
Dessouky et al. (1999)	Simulation	min transfer delays	2, 5 or 10 one-way	2; 5, or 10 respectivetly	Variable	Parameter	x			Los Angeles County, Transit Agency
Ceder et al. (2001)	Heuristic method	max number of synchronizations	7 one-way	3		Variable	x			An Israel bus company
Chowdhury and Chien. (2001)	Numerical analysis	min cost of users (transfer, waiting and travel times) and vehicles (operation)	1 train 3 feeder bus (one-way)	4	Variable	Variable			x	
Chowdhury and Chien. (2002)	Four stages heuristic	min waiting times of users at current transfer nodes and downstreams stops	1 train 3 feeder bus (one-way)	4	Variable	Variable			x	
Dessouky et al. (2003)	Simulation	min waiting times of users at current transfer nodes and downstreams stops	5 one-way	1	Variable	Parameter			x	
Eranki (2004)	Heuristic method	max number of synchronizations	7 one-way	3		Variable		x		Ceder and Tal, (2001)
Teodorović and Lucić (2005)	Ant Colony and Fuzzy Logic	min waiting times of users at transfer nodes	[20,150] one-way	[4,15]*		Parameter			x	
Cevallos and Zhao (2006)	Genetic Algorithm	min waiting times of users at transfer nodes	40 two-way	255		Parameter	x			Browar county, Florida.
Ting and Schonfeld (2007)	Numerical analysis	min costs of waiting on-board , missed connection for late arrival and transfer downstream at stops.	10 one-way	3	Variable	Variable			x	
Liu et al. (2007)	Nested tabu search	ratio between the number of routes arriving simultaneously respect all routes passed at each transfer zone.	8 one-way	3		Variable			x	A regional bus system
Fügenschuh (2009)	Branch-and-cut	min number of deployed vehicles and driving times of all trips	[25,490] one-way	[165,708]*		Parameter	x		x	Rural areas in Germany
Guihaire and Hao (2010)	Tabu search	max number of synchronizations min waiting times of transfer users	50 one-way	282*		Variable			x	Loiret, France
Ibarra-Rojas and Rios-Solis (2012)	Multi-start heuristic	max number of synchronizations	[15,200]	[3,40]		Variable	x			Bus network of Monterrey, Mexico
Wu et al. (2012)	Genetic algorithm	min weighted waiting times of users at transfer nodes	4 one-way	2	Parameter	Parameter			x	
Wu et al. (2013)	Simulated annealing	min waiting times of all users at system	8 one-way	3	Variable	Parameter			x	
Saharidis et al. (2014)	CPLEX	min waiting times of users at transfer nodes	25 one-way	13		NA	x			Bus network in Greece
Aksu et al. (2014)	Genetic algorithm	min transfer times, waiting times and penalties failed transfered users	4 and 14 one-way	2 and 3 respectively		Variable		x		Shafah and Khani (2010) Ceder et al. (2001)
Muñoz et al. (2014)	Iterative heuristic with numerical analysis	min fleet size cost, in-vehicle costs for users, transfer cost, waiting time cost in transfer and costs for users by holding	60 one-way	84**	Variable	Variable	x			Night services in Santiago, Chile
Wu et al. (2016)	Genetic algorithm	max number of transfered passengers	10 one-way	3		Variable			x	
Ibarra-Rojas and Muñoz (2016)	CPLEX and Genetic algorithm	min weighted sum of the penalties for violating the expected arrivals time at each stop.	2 or 5 one-way	4 or 20 respectively		Parameter	x		x	Santiago, Chile
Ibarra-Rojas et al. (2016)	Multistart heuristic, VNS and Population-Based Algorithm	max number of synchronizations	5	3		Variable			x	Bus network of Monterrey, Mexico
Fouilhoux et al. (2016)	CPLEX	max number of transfered users	[15,200] one-way	[3,150]		Variable		x		Ibarra-Rojas and Rios-Solis (2012)

∗ Total number of sychronization possibilities among all service lines, ∗∗ Maximum number of sychronization possibilities between two service lines.
NA - no available information.

2.2 Motivation: case study

The Metropolitan Mobility Network of Santiago de Chile connects physically all public transport buses of the city, the Metro of Santiago and the train line to the locality of Nos, in the peripheral commune of San Bernardo. System users can move in any of the three modes making up to two transfers for the value of a single fare. These connections can be performed within a period of two hours (120 minutes) after the first entry into the system. The system is inserted in an area of 2,353 km^2, which covers the 32 communes of the Province of Santiago plus the communes of San Bernardo and Puente Alto and operates in the urban areas of these communes covering an area of about 680 km^2 performing on average 2.92 million of transactions per day (Directorio de transporte publico metropolitano, 2018).

By 2017, the population in these 34 communes was estimated to 7 million inhabitants, who are serviced with: 7 underground train lines and 136 stations of Santiago's Metro and the bus system that is operated by 6 bus concession companies, with 6681 vehicles performing 377 services during a normal working day. Since 2015, the public administration has made an effort to improve the night services, passing from 7 operative services operating with timetables in may 2015 to 48 operative services at present, increasing coverage and attending 130,000 users every week during the night period corresponding to all the communes of the city (see Figure 2.1). In spite of these efforts, the system resolve the travel requirements at night with direct trips and low frequencies. That protocol results in a high number of transfers causing high uncertainty. At the same time, waiting times are also high and uncertain giving high sense of insecurity related to night travel.

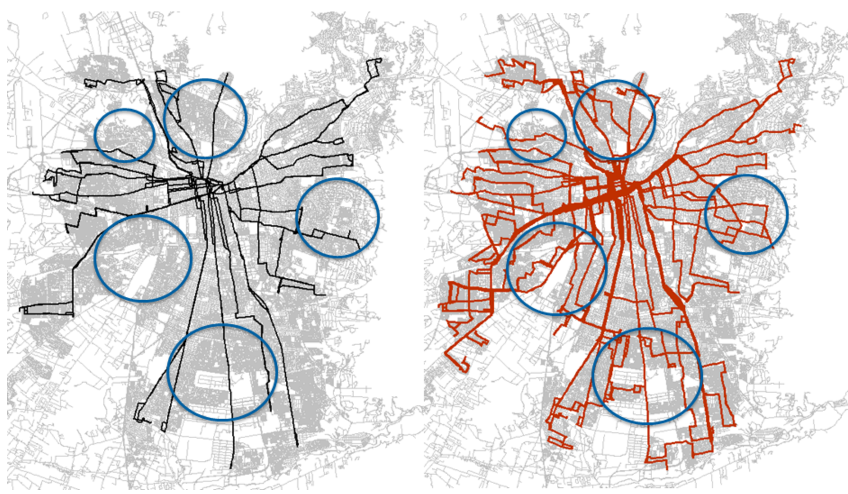

Figure 2.1: Increase in night coverage with service lines crossing from downtown.

Our research is focused on timed transfers on a real network of bus services with fixed schedule corresponding to the night urban bus system of Santiago de Chile. Buses that synchronize arrive within a time window of allowable waiting time and may dwell at certain bus stops where the holding operation is possible. The holding operation is a feature not addressed before in mixed integer programming models to solve problems of this kind. Passengers will benefit from a proper synchronization between the different bus lines at common transfer zones as they are served by more than one line to complete the journey, increasing the coverage of the city, its safety and decreasing users uncertainty.

2.3 Problem formulation

In this section, we present our integer programming formulation for the Bus Synchronization Timetabling Problem with Dwelling Times (BST-DT). The problem addressed in this work is the following: given a network of night urban bus services, the goal is to maximize the number of encounters of buses belonging to different lines that are able to perform a synchronized operation of passengers' transshipments at transfer zones, under a fixed schedule operation, satisfying systems conditions. Trips that synchronize arrive within a time window of allowable waiting time and may hold at certain bus stops where such an operation is allowed. Maximum dwelling times per trip and bus stop capacity constraints are considered in the model. A border limit constraint is added to deal with the transition between day shift and night shift. For the formulation, we use the following notation.

The set of bus lines is denoted by I. For each line i, we define J_i as the set of lines that may have synchronization nodes with line i. Set B^{ij} represents common transfer zones between lines i and $j \in J_i$. Ω_i is the set of nodes where the trips of line i could dwell and $STOP_b$ is the set of bus stops belonging to transfer zone b. $\overline{T}$ is the length of the planning horizon in minutes, f^i denotes the number of trips of line i during the planning horizon and is computed as $f^i = \lceil \overline{T}/h_i \rceil$, h_i is the duration in minutes of the headway of line i in the existing timetable, t_b^i is the travel time from the depot of line i to transfer zone b during the planning horizon. $\underline{W}_b$ $\overline{W}_b$ are the minimum and maximum allowable waiting time between synchronized trip arrivals at transfer zone b, respectively. CAP_r is the capacity of bus stop r. L_b^i is the maximum dwelling time that can perform a trip of line i in transfer zone b (it could be zero if the line does not perform holding in the transfer zone b), L_b^i must be bounded as the minimum between the maximum dwelling time policy and the headway of the line i. If this dwelling time is not bounded, two consecutive trips of the same line i could perform dwelling times at same time in a node, which is not reasonable.

The decisions are represented by three types of variables. The first type identifies a synchronization between two vehicles of two different lines, the second one determines the departure time for every trip of each line and the third one indicates the dwelling time for every trip of a line i in transfer zone b. S_b^i is an auxiliary variable employed in constraint (2.6) to keep track of the accumulated dwelling times of a vehicle. These variables are represented as follows.

- $Y_{pqb}^{ij} : \begin{cases} 1, \text{ if an user of trip } p \text{ of line } i \text{ could take the trip } q \text{ of line } j \text{ at transfer zone } b. \\ 0, \text{ otherwise.} \end{cases}$
- $X^i :$ departure time of the first trip of line i, $X^i \in [0, h^i]$
- $Z_b^i :$ dwelling time of line i at transfer zone b, $Z_b^i \in [0, L_b^i]$
- $S_b^i :$ accumulated dwelling times of a trip of line i before its arrival at transfer zone b.
- $V_{pqb}^{ij} : \begin{cases} 1, \text{ if trip } p \text{ of line } i \text{ coincides with trip } q \text{ of line } j \text{ at transfer zone } b. \\ 0, \text{ otherwise.} \end{cases}$

To simplify notation, we define another auxiliary variable $T_b^{ip} = X^i + t_b^i + (p-1) \cdot h^i + S_b^i \quad \forall i \in I, p = 1, \ldots, f^i, b \in B^{ij}$, that represents the arrival time of the trip p of line i to transfer zone b. We noticed that the next proposed model for timetabling night bus services is an extension of previous approaches by Ceder and Tal (2001), Eranki (2004) and Ibarra-Rojas and Rios-Solis (2012) considering fixed headways and potential dwelling times at transfer stations as new features.

$$F_{BTP} = max \sum_{i \in I} \sum_{j \in J_i} \sum_{b \in B_{ij}} \sum_{p=1}^{f_i} \sum_{q=1}^{f_j} IMP_{ij}^{b} \cdot Y_{pqb}^{ij} \tag{2.1}$$

Subject to:

$$X^i \leq h^i \quad \forall i \in I \tag{2.2}$$

$$\overline{T} - h^i \leq X^i + \left(f^i - 1\right) \cdot h^i \leq \overline{T} \quad \forall i \in I \tag{2.3}$$

$$\begin{gathered}(T_b^{ip} + \underline{W}_{ij}^b) - (T_b^{jq} + Z_b^j) \geq \underline{W}_{ij}^b \cdot Y_{pqb}^{ij} - \underline{M}_{pqb}^{ij} \cdot (1 - Y_{pqb}^{ij}) \\ \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij}\end{gathered} \tag{2.4}$$

$$\begin{gathered}(T_b^{jq} + Z_b^j) - (T_b^{ip} + \overline{W}_{ij}^b) \leq \overline{W}_{ij}^b \cdot Y_{pqb}^{ij} + \overline{M}_{pqb}^{ij} \cdot (1 - Y_{pqb}^{ij}) \\ \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij}\end{gathered} \tag{2.5}$$

$$S_{b'}^i = \sum_{b \in E^i : t_{b'}^i > t_b^i} Z_b^i \quad \forall i \in I, b' \in B^{ij} \tag{2.6}$$

$$Y_{pqb}^{ij} \in \{0, 1\} \quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} \tag{2.7}$$

$$X_i \in R^+ \quad \forall i \in I \tag{2.8}$$

$$Z_b^i \in [0, L_b^i] \quad \forall i \in I, b \in B^{ij} \tag{2.9}$$

With the objective function the weighted number of synchronizations is maximized. The parameter IMP_{ij}^b weighs each line combination, representing a pair of lines that synchronize in a transfer zone. We are aware that not all line combination have the same relevance and that is the purpose of creating IMP_{ij}^b. In the implementation we consider $IMP_{ij}^b = 1$ that means we are maximizing the straightforward number of synchronizations.

Constraints (2.2) and (2.3) force the first trip of each line to depart at the beginning of the planning period $\overline{T}$, and the last trip of each line to depart at the end of the planning period $\overline{T}$, respectively. Constraints (2.4) and (2.5) allow the synchronization variables Y_{pqb}^{ij} to be activated if the difference in time between arrivals of trip q of line j and trip p of line i at transfer zone b falls between $\underline{W}_{ij}^b$ and $\overline{W}_{ij}^b$. In this model, $\underline{M}_{pqb}^{ij}$ and $\overline{M}_{pqb}^{ij}$ are big-M parameters, that can be computed as the maximum difference of arrival times for every pair of vehicles (p, q) of lines (i, j) that synchronize at every transfer zone b and that will shape a better polyhedra for this problem. Constraints (2.6) accumulates in auxiliary variable T_b^{ip} the dwelling time performed for every vehicle before arriving to transfer zone b. Finally, constraints (2.7), (2.8) and (2.9) represent the domain of the decision variables.

2.3.1 Bus rapid transit service inequalities

For synchronizing urban bus services in general, not only night shift, the efforts to make it properly may to need some operational controls. We call these constraints Bus rapid transit service inequalities. They are related to bus stop capacity, maximum dwelling time per trip and period border condition.

2.3.1.1 Bus stop capacity

Here, bus stops have limited capacity at boarding aisles. To address this feature, we include sets of constraints: (2.10) and (2.11), to see if the trip p of line i synchronizes with the trip p of line i at the beginning or at the end of its dwelling time period. If it happens, the new variable $V_{pqb}^{ij} \in \{0,1\} \quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} : \exists r \in STOP_b \cap \Gamma^i \cap \Gamma^j$ will be activated. Let E^i be the set of transfer zones where line i can eventually synchronize $b \in E^i$. Thus,

$$T_b^{ip} - T_b^{jq} \leq \overline{M}_{pqb}^{ij} \cdot (1 - V_{pqb}^{ij})$$
$$\forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} : \exists r \in STOP_b \cap \Gamma^i \cap \Gamma^j \tag{2.10}$$

$$(T_b^{jq} + Z_b^j) - T_b^{ip} \leq \underline{M}_{pqb}^{ij} \cdot (1 - V_{pqb}^{ij})$$
$$\forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} : \exists r \in STOP_b \cap \Gamma^i \cap \Gamma^j \tag{2.11}$$

$$\sum_{\substack{j \in \Lambda_i: \\ r \subset \Gamma^j}} \sum_{q=1}^{f_j} V_{pqb}^{ij} \leq CAP_r - 1 \quad \forall i \in I, p = 1, \ldots, f^i, b \in E^i, r \in STOP_b \cap \Gamma^i \tag{2.12}$$

Constraint (2.12) counts all trips that physically coincide with trip q of line j in bus stop r located at transfer zone b (even if dwelling time is equal to zero). As it is observed from the perspective of a specific trip p of line i, bus stop capacity is reduced by one unit representing the spot in the boarding aisle that it is already using. $STOP_b$ are the bus stops that belong to transfer zone b. Γ_i are the bus stops that belong to the route of line i. And Λ_i is the set of lines that could synchronize in any bus stop in the route of line i.

2.3.1.2 Maximum dwelling time per trip

Lines involving dwelling times increase their cycle time, this may imply longer travels for passengers. We established a policy to limit such increment in time. It controls that in every pair of transfer zones belonging to the route of a line, the total travel time cannot increase more than 10% (policy established by the authorities) due to dwelling times accumulated through the route.

$$\sum_{b \in E^i : t_{b'}^i > t_b^i} Z_b^i \leq \left\lfloor \left(\frac{1}{10}\right) \cdot t_{b'}^i \right\rfloor \quad \forall i \in I, b' \in E^i \tag{2.13}$$

2.3.1.3 Period border condition

There is a transition between the night and the day shifts that has to be considered (12:01am to 1:00am). During this transition period, the service must be guaranteed to night passengers; then, for each night service we check the difference between the departure time from the depot of the last trip of the day shift and the starting time of the night shift. If this difference is greater than the line headway, one or more trips are programmed in order to at least keep the headway. It means that the departure time upper bound of the first trip (CB_i) will be lower to keep the level of service.

$$X^i \leq CB_i \quad \forall i \in I \tag{2.14}$$

2.3.2 Formulation strengthening

Besides bus rapid transit service inequalities, to strengthen the formulation of BST-DT we add some sets of valid inequalities. These new inequalities were defined from the case of a real application (Stop capacity and passenger synchronization relation), also considering some features of the addressed problem (the least common multiple rule constraint), polyhedra characteristics (Zero start constraint) and adaptations from current scientific literature (valid inequalities of Fouilhoux et al. 2016).

2.3.2.1 Zero start constraint

In the first runs of the BST-DT model, the gaps obtained after a run limit time still remained large (more than 40%). We realized that the variables X_i for all service lines $i \in I$ were strictly greater than zero, we can get a new solution of the model with at least the same number of synchronizations if we move all departure times some minutes before. In fact, a greater number of synchronizations may be achieved due to the possibility of taking advantage of these minutes at the end of the planning horizon. This is because there is no constraint forcing some specific schedule. Then the schedules of all service lines can be advanced in the same magnitude, namely $\min_{i \in I} X_i$. In practice, we subtract this magnitude from X_i $i \in I$. In our MILP it is obtained creating a binary variable that controls at least one of the departure times of service lines (X_i) is equal to zero.

$$\exists i \in I, \quad X_i = 0 \tag{2.15}$$

2.3.2.2 Adaptation of constraints from the work of Fouilhoux et al. (2016)

Fouilhoux et al. (2016) proposed two set of valid constraints for the Bus Synchronization Timetabling Problem (BST), which they call: the Synchronization inequalities, constraints 2.18, and the Headway inequalities, constraints 2.17. These sets of inequalities can be explained with the following statement: Suppose that a passenger of a trip j is waiting at the boarding aisle of his respective bus stop in transfer zone b. The passenger will observe every h_i minutes a trip of line j passing before he decides what trip of line i to board; consequently, we can determine the possible number of synchronizations between the passenger of line j and the trips of line i. We also consider the case of the passengers of the line i synchronizing with trips of line j.

$$\sum_{p=1}^{f_i} Y_{pqb}^{ij} \leq 1 + \left\lfloor \frac{\overline{W}_{pqb}^{ij} - \underline{W}_{pqb}^{ij}}{h_i} \right\rfloor \quad \forall i \in I, j \in J_i, q = 1, \ldots, f^j, b \in B^{ij} \tag{2.16}$$

$$Y_{pqb}^{ij} + \sum_{q'=q+1}^{f_j} Y_{pq'b}^{ij} + \sum_{p'=p+1}^{f_i} Y_{p'qb}^{ij} \leq 1 + \left\lfloor \frac{\overline{W}_{pqb}^{ij} - \underline{W}_{pqb}^{ij}}{min(h_j, h_i)} \right\rfloor \tag{2.17}$$
$$\forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij}$$

We have adapted these set the constraints but changing the perspective of their original design. We consider that it is not the passenger who decides which trip to take, even if he has the possibility to do it, usually he wants to arrive to his destination in the shortest possible time. Then, a better insight is if we suppose that a vehicle associated with a trip of service line j is dwelling in its respective bus stop in transfer zone b. The vehicle could receive passengers transferring every h_i minutes. In this way, we can determine the possible number of synchronizations between passengers of line i, that effective can synchronize, through the headway of their service line i and the dwelling time of a vehicle of line j. Then, the adapted constraints are shown in expressions (2.19) and (2.20).

$$\sum_{p=1}^{f_i} Y_{pqb}^{ij} \leq 1 + \frac{Z_b^j}{h_i} \quad \forall i \in I, j \in J_i, q = 1, \ldots, f^j, b \in B^{ij} \tag{2.18}$$

$$Y_{pqb}^{ij} + \sum_{q'=q+1}^{f_j} Y_{pq'b}^{ij} + \sum_{p'=p+1}^{f_i} Y_{p'qb}^{ij} \leq 1 + \frac{Z_b^j}{h_i} \tag{2.19}$$
$$\forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij}$$

2.3.2.3 Least common multiple rule constraint

It is possible to know further synchronizations of both lines due to the least common multiple of their headways as long as that occurs within the planning horizon from the first possible synchronization. This constraint arises from the fact that headways of service lines are not decision variables in the model, and we consider them as fixed. The generalization of this constraint is expressed in equation (2.21). Basically, if trip p of line i synchronizes with trip q of line j then trip $p+k$ of line i synchronizes with trip $q+m$ of line j. This is accomplished if m and k are such that the next conditions are satisfied:

- $m, \ k \in \mathbb{N}$
- $m \cdot h_j \ = \ k \cdot h_i \ = LCM(h^i, h^j) \leq T$

$$Y_{pqb}^{ij} = Y_{p+k,q+m,b}^{ij} \quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i - k, q = 1, \ldots, f^j - m, b \in B^{ij} \tag{2.20}$$

2.3.2.4 Stop capacity and passenger synchronization relation

The variable V^{ij}_{pqb} reflects that if the vehicles associated to the trip p of line i and the trip q of line j synchronize in the boarding aisle of transfer zone b, then passengers from the trip p of line i can transfer perfectly to the trip q of line j and variable Y^{ij}_{pqb} is activated. It is clear that this constraint is useful only if the set of capacity constraints is employed.

$$Y^{ij}_{pqb} \geq V^{ij}_{pqb} \quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} : \exists r \in STOP_b \cap \Gamma^i \cap \Gamma^j \tag{2.21}$$

2.3.2.5 Time windows synchronization constraint

We could know if passengers from trip p of line i can synchronize with trip q of line j at transfer zone b', if arrival time of that vehicle to transfer zone b' falls into the expected passengers transfer time windows after the vehicle visited transfer zone b (immediately before). Parameter O^b_i means the position of transfer zone b in set E_i ordered by travel time t^i_b.

$$\begin{aligned} &T^{jq}_{b'} \geq T^{ip}_b + Z^i_b + (t^i_{b'} - t^i_b) + \underline{W}^{ij}_{pqb} - \underline{M}^{ij}_{pqb'} \cdot (1 - Y^{ij}_{pqb'}) \\ &\quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in E^i, b' \in B^{ij} : \quad O[i, b'] = O[i, b] + 1 \end{aligned} \tag{2.22}$$

$$\begin{aligned} &T^{jq}_{b'} \leq T^{ip}_b + Z^i_b + (t^i_{b'} - t^i_b) + \overline{W}^{ij}_{pqb} + \overline{M}^{ij}_{pqb'} \cdot (1 - Y^{ij}_{pqb'}) \\ &\quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in E^i, b' \in B^{ij} : \quad O[i, b'] = O[i, b] + 1 \end{aligned} \tag{2.23}$$

2.3.3 Preprocessing

A large number of synchronization variables Y^{ij}_{pqb} can be set to zero when the earliest arrival time of trip p belonging to line i arrives to transfer zone b later than the latest possible time that trip q of line j departs from transfer zone b. Then the following preprocessing is performed:

$$\begin{aligned} &if \quad (0 + t^j_b + (q-1) \cdot h^j + 0) - (h^i + t^i_b + (p-1) \cdot h^i + L^b_i \cdot O^i_b) > 0 \\ &or \quad (0 + t^i_b + (p-1) \cdot h^i + 0) - (h^j + t^j_b + (q-1) \cdot h^j + L^b_j \cdot O^j_b) > 0 \\ &then \quad Y^{ij}_{pqb} = 0 \quad \forall i \in I, j \in J_i, p = 1, \ldots, f^i, q = 1, \ldots, f^j, b \in B^{ij} \end{aligned} \tag{2.24}$$

In table 2.2, we observe the impact of this preprocessing on the instance size in terms of variables and constraints. The instances of 20 and 50 services here were created to test the potentiality of the preprocessing proposed. For this reason, the objective function that pursues the maximization of all possible synchronization and the first set of constraints, from constraint (2.2) to constraint (2.9), can be considerably improved by adding all inequalities and strengthening constraints discussed above. Both in number of variables and in number of constraints, the preprocessing reduces in a hundredth the size of the problem, a promising result for a formulation considered NP-hard.

Instance (Services)	20		50	
Preprocessing	No	Yes	No	Yes
Variables	$2.62 \cdot 10^6$	$1.09 \cdot 10^4$	$6.73 \cdot 10^6$	$9.79 \cdot 10^4$
Constraints	$0.83 \cdot 10^6$	$2.02 \cdot 10^4$	$3.86 \cdot 10^6$	$8.37 \cdot 10^4$

Table 2.2: Size impact of preprocessing.

2.3.4 Improving solution routine

Reaching optimal solutions in medium and large size instances is not easy, given the NP-hard nature of the BST problem. The formulation strengthening and the preprocessing helped to close initial gaps in the real instance, but in our best effort we could not get gaps lower than 18% for 48 service lines and 14 transfer zones, as we will see in the next section of application case. It was necessary to complement the model with other strategies. We explore new valid inequalities to the problem focused mainly in the direct impact of the synchronization variables (Y^{ij}_{pqb}).

Algorithm 1: Solution improving algorithm.

1 Set run time limit τ ;
2 Set improved gap limit γ ;
3 Set iteration $= 0$;
4 Solve the BST-DT model ;
5 **if** *run time* $> \tau$ **then**
6   **if** *improved gap* $> \gamma$ **then**
7   Save current solution of variables $\hat{X}_i$;
8   **if** *iteration* $>$ *0* **then**
9   Update bandwidth cuts: $\hat{X}_i + 1 \geq X_i \geq \hat{X}_i - 1 \quad \forall i \in I$;
10   **else**
11   Create bandwidth cuts: $\hat{X}_i + 1 \geq X_i \geq \hat{X}_i - 1 \quad \forall i \in I$;
12   **end if**
13   Return to step 3 ;
14   iteration ++ ;
15   **else**
16   Stop;
17   **end if**
18 **else**
19   Stop;
20 **end if**
21 Return current solution: Y^{ij}_{pqb}, X_i and Z^i_b ;

The algorithm works as follows: after a run time limit where a unsatisfactory gap has been reached, we add a couple of constraints to the model that we called bandwidth cuts. These inequalities use the current solution of service departure time variables, allowing them just to change in a bandwidth of one minute (up or down). This algorithm is a local search strategy as the proposals by Fischetti and Lodi (2003) and Danna et al. (2005) to improved solutions given a feasible solution to a Mixed Integer Programming model.

We tested several sizes of bandwidth, but we discovered that it improves quickly the general objective function (synchronizations) after few iterations. The insight of the idea was obtained by observing the values of the variable X_i; in small test instances, it did not change much comparing optimal solutions with respect to other solutions. In some way, the algorithm permits to find better values of dwelling variables Z_b^i when variables X_i are very restricted.

2.4 Numerical results

In this section, we describe the experiments performed and describe and analyze the results obtained. First, we describe the instances solved and the computational environment used. Then, we analyze and report on the effect of including or not the possibility of holding and the impact of using the valid inequalities proposed in subsections 2.3.1 Bus rapid transit service inequalities and 2.3.2 Formulation strengthening. Finally, we illustrate in more detail what the proposed solution looks like based on the analysis of one of the most relevant transfer zone of the system.

2.4.1 Case study and construction of instances

The case study includes 48 services considering both directions (96 lines) and 14 transfer zones (8 of them with dwelling time allowed). The length of the planning horizon (T) is 239 minutes, between 1:00 am and 5:00 am, which corresponds to the current night shift. Travel times from depots of lines to nodes (t_b^i) are in the range of 1 to 70 minutes. The minimum and maximum allowable waiting times between synchronized trips arrivals at bus stops ($\underline{W}_{ij}^b$, $\overline{W}_{ij}^b$) are between 0 and 12 minutes, respectively. The maximum dwelling time allowed (L) at any node is 3 minutes. The headway (h_i) of services 210, 401N, 301 were set to 10 minutes. Headway of others services were set to 30 minutes. Each bus stop has a capacity (CAP_r) of 2 vehicles. For this instance, the border condition for each line was established as half of its headway.

From the data of the case study, six instances were created to prove the benefits of different sets of inequalities in the BST-DT problem. Instance 1 is the proposed model for timetabling night bus services considering fixed headways and potential dwelling times at bus stops. Instance 2 includes the zero start constraint; in instance 3, we adapted the valid inequalities presented by Fouilhoux et al. (2016) for the BST problem; least common multiple rule constraint, stop capacity and passenger synchronization relation constraints and time windows synchronization constraint were tested in Instances 4, 5 and 6, respectively. Finally, whole sets of valid inequalities were incorporated in instance 7. In all of them preprocessing was previously performed.

2.4.2 Computational experiments

The model is implemented in AMPL and MIP are solved with CPLEX 12.4. We solve our instance using an Intel core i7-8550U computer with 4 GB of RAM memory and 16 GB of Intel memory Optane. If the models are not solved to optimality, a time limit of 7200 seconds is used. In subsection 2.4.2.3, we explain how this time limit value was determined.

Three series of experiments were performed: (i) we solve the equation system 2.1-2.9 including the Bus rapid transit inequalities (2.10-2.14) directly through CPLEX afterwards adding different sets of valid inequalities to evaluate their impact. Then, (ii) we introduce four instances that combines decisions of include (or not) fixed lines and to allow (or not) dwelling times at bus stops in Bus Synchronization Timetable model to evaluate the yield of these decisions and (iii) by last, the improving solution routine is applied to this instance the instance with the best solution gap found in (i).

2.4.2.1 Valid inequalities performance.

Table 2.3 summarizes the impact on results of incorporating different combinations of valid inequalities developed for the BST-DT. We show the instance number (Ins.), the set of constraints that compose the instance (Constraints), the number of explored nodes to get the best integer solutions achieved (# Nodes), the initial relaxed solution for the objective function ($Fobj_r$), the best integer solution obtained for the objective function ($Fobj_f$), the objective function percentage change with respect to Instance 1 ($\Delta\ Fobj_f$), its gap after 7200 cpu seconds runtime (Gap) and the gap from the best upper bound reached in all instances (BGap).

Ins.	Constraints	# Nodes	$Fobj_r$	$Fobj_f$	$\Delta\ Fobj_f$	Gap (%)	BGap (%)
1	(1-14)	299	19099	9450	-	34.55	12.20
2	(1-14)+(15)	61	18359	9722	272	37.29	9.06
3	(1-14)+(18-19)	1	17586	9532	82	55.49	11.23
4	(1-14)+(20)	1	16288	9329	-111	34.47	13.66
5	(1-14)+(21)	1	16480	9383	-67	38.76	13.00
6	(1-14)+(22-23)	1	15921	7857	-1593	34.95	34.95
7	(1-23)	1178	18353	9943	493	34.35	6.64

Table 2.3: Results impact for different combinations of valid inequalities.

The best solution found among these instances was the instance 7, probably an expected result since we are considering the model involving all valid inequalities designed to strengthen the problem. Although instance 6 reports the worst solution found, this experiment provides the best known upper bound (10603.02) among all comparable models that we tested in this work. By incorporating the zero start constraint (instance 2), the model showed the second best solution after instance 7, but without involving a large number of constraints as required in instances 4, 5 and 6, where we got worse values for the objective functions. None of the instances got a good initial relaxed objective function when compared with the solution found for the BST problem.

2.4.2.2 MMN instances results.

At beginning of the project, authorities of MMN desired to make decisions with regards dwelling and departure times just over 30 out of 50 lines services that compose the system, letting the remaining service lines static. From instance 7, we constructed these instances that combine these decisions of including (or not) fixed lines and allowing (or not) dwelling times at bus stops in Bus Synchronization Timetable model to evaluate the impact of these decisions, see table 2.4.

Ins.	Fixed lines	Dwell	Variable type			Constraints
			Integer	Binary	Continuous	
A	20	No	74	11548	0	27089
B	20	Yes	74	21198	75	62594
C	0	No	94	26554	0	84181
D	0	Yes	94	37282	268	134198

Table 2.4: Configuration and features of MMN proposed instances

Table 2.5 shows the instance number (Ins.), number of lines with fixed departure time (Fixed lines), the number of explored nodes to get the best integer solutions achieved (# Nodes), the initial relaxed solution objective function ($Fobj_r$), the first integer solution objective function ($Fobj_0$), the best integer solution obtained for the objective function after time limit ($Fobj_f$), its final gap (Gap), the objective function percentage change respect to the instance A (Δ $Fobj_f$) and the percentage change of the gap with respect to the instance A (Δ Gap).

Ins.	# Nodes	$Fobj_r$	$Fobj_0$	$Fobj_f$	Gap (%)	Δ $Fobj_f$	Δ Gap
A	617	10526	1759	6393	8.9	-	-
B	10270	7125	3828	6419	18.3	0.41%	106%
C	5103	13820	1759	8009	26.6	25.28%	199%
D	1178	18353	3531	9943	34.4	55.53%	287%

Table 2.5: Results of MMN proposed instances.

Here, we can observe that variables X_i, departure time of first trip of line i, reinforce the flexibility of the model. The number of synchronizations of the system increased when no fixed lines are included in the model, and also if holding is allowed at some bus stops (Z_i^b), the number of potential synchronizations increase in 2000 in comparison to the base model (instance A). The synchronizations that are achieved in the different solutions are maintained, in general, and new ones are achieved. This is not because the Y value variables are the same as in the previous model, but the number of synchronizations is kept if we look at it in aggregate at pairs of services.

We note a direct correlation between the final gap and the flexibility of the model, where flexibility is associated to: dwelling time at bus stops and departure time of lines. A larger number of decision variables implies, not only greater objective function, a greater final gap (up to almost 3 times from instance A, 287% more gap for 55% more synchronizations compared with instance D). This results probably from the combinatorial nature of the model, where the new variables are integers. None of our instances were solved to optimality (final gap equal to zero) after a very long runtime.

2.4.2.3 Improving solution routine results

Table 2.6 presents the results of the improving solution routine proposed for BST-DT made from the instance 7. Its columns show the iteration number (Iter.), the number of explored nodes to get the best integer solution achieved (# Nodes), the best integer solution obtained for the objective function ($Fobj_f$), the percentage change with respect to Instance 7 (Δ $Fobj_f$) and the gap from the best upper bound reached in all instances (BGap).

Every additional iteration implies a lower value for the objective function, meaning that the proposed solution routine improves quickly the current solution. For instance, in iterations one and two, 95 and 29 extra synchronization were achieved. Despite the fact that, it represents less than 1% of improvement, if we think in terms of passengers level of service it becomes significant since a number of users will have lower waiting times and uncertainty, both relevant issues specially for night urban transport services.

For all iterations, gaps of models were around 5% even employing bandwidth cuts, which reinforces the idea of complexity of the BST-DT problem. To make a fair comparison, we ran the model during 43200 sec (6x7200sec) directly to verify that the improvements were effectively achieved by the routine and not by the execution time. We obtained that the gap only improved 0.2% proven from 7200 seconds to 43200 seconds. This gap improvement was due to a better lower bound, not to a greater number of synchronizations. Due to this reason, we set the parameter Run time limit (τ) to 7200 seconds and parameter Improved gap limit (γ) to 0.1%. We identified this value, in general, is the maximum improving reached in the running gap in proven instances between seconds 6000 and 7200.

Iter.	# **Nodes**	$Fobj_f$	Δ $Fobj_f$	**BGap** (%)
0	1178	9943	-	6.64
1	3022	10038	95	5.63
2	7777	10067	29	5.32
3	9527	10073	6	5.26
4	6282	10074	1	5.25
5	4665	10082	8	5.17

Table 2.6: Results of the improving solution routine

2.4.3 Detailed analysis of proposed timetable

In the present subsection, we analyze the results of the model in the context of the real application. For this purpose, we illustrate the results with one of the fourteen transfer zones addressed in our work: Plaza Italia. This location, is not only one of the most representative locations at Santiago de Chile city, but also a very confluent point of transferring passengers between bus services and metro: it is composed of 42 service lines (see Figure 2.2) and two metro lines (line 1 and line 5) picking up and dropping passengers in six bus stops and a metro station (Baquedano) in daytime shift.

	Service lines					
PA393-1	508R					
PA343-10	119I	401R	405I	432NI	541NI	
PA384-11	418I	515NI	D09NI			
PA383-5	418R	508R	515NR	541NR		
PC86-4	210I	210vR	403R	516I	B02NI	F30NR

Table 2.7: Service lines that belong to each bus stop of Plaza Italia transfer zone.

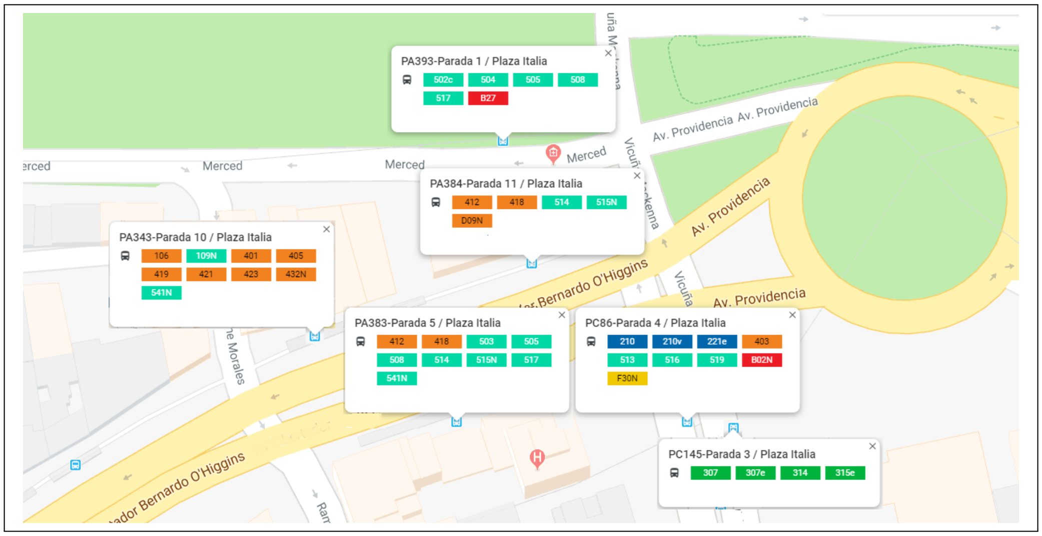

Figure 2.2: Service lines and bus stops of Plaza Italia transfer zone in Santiago de Chile.

At this important activity pole, 19 out of the 42 service lines in 5 bus stops work at MMN night shift block (see Table 2.7). Table 2.8 presents walking times (minutes) between these bus stops belonging to Plaza Italia transfer zone. Figure 2.4 shows a cycle (30 minutes) for service lines belonging to Plaza Italia transfer zone with their arrivals, dwelling and departure times. To exemplify, we choose one service line stopping at each of these 5 bus stops in Plaza Italia transfer zone: 508R, 541NI, F30NR, 418R and 515NI. We analyze the trips arriving between minutes 197 and 203 at Plaza Italia transfer zone. For the sake of illustration, we show the sequence of actions of a passenger of service line 508R alighting at bus stop PA393-1.

	PA393-1	**PA393-10**	**PC86-4**	**PA393-5**	**PA393-11**
PA393-1	-	3	5	5	1
PA343-10	3	-	4	2	1
PC86-4	5	4	-	2	3
PA383-5	5	2	2	-	2
PA384-11	1	1	3	2	-

Table 2.8: Walking time (minutes) between bus stops belonging to Plaza Italia transfer zone.

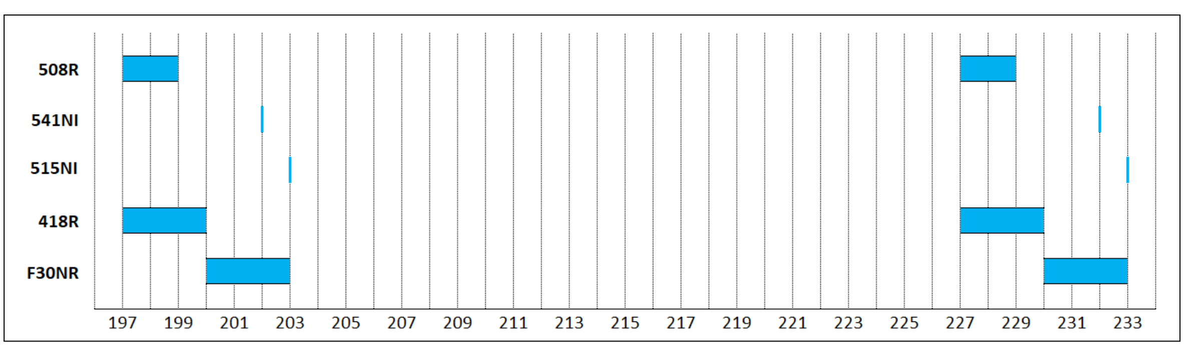

Figure 2.3: Trips arriving between minutes 197 and 233 at Plaza Italia transfer zone.

Figure 2.4 shows the transfer actions of this passenger: blue areas, black lines and red lines represent dwelling times, walking times between and waiting times at bus stops, respectively. A passenger who alights at 3:17 am (minute 197) from service line 508R at bus stop PA393-1 could synchronize with service lines: 541NI, F30NR and 515NI, respectively. For instance, a passenger who starts walking from bus stop PA393-1 at 3:17am will arrive to bus stop PC86-4 at 3:22am (a five minutes walk) to board the 5_{th} trip of service line F30NR that arrived to bus stop PC86-4 at 3:20am and will departure at 3:23am (it means the vehicle is hold 3 minutes).

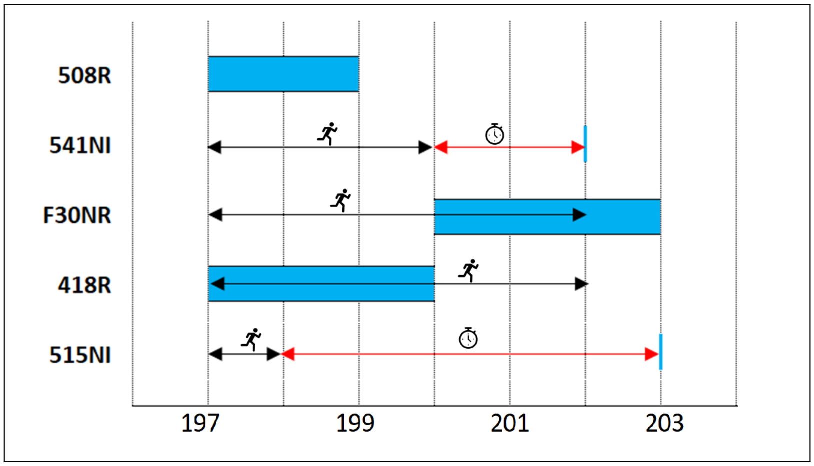

Figure 2.4: Transfer actions between minute 197 and 203 at Plaza Italia transfer zone.

If the same passenger has the intention of transferring to service line 541NI, he will arrive to bus stop PA343-10 at 3:20 am (a three minutes walk) to board this service line (not holding observed) at 3:22am (after 2 minutes wait at the bus stop). It is similar to board line 515NI, in that case the passenger will walk 1 minute to bus stop PA384-11 and will wait 5 minutes to board it at 3:23 am. But in case of service line 418R, even if the trip dwells from 3:17 am to 3:20 am, the passenger will never board the vehicle because he cannot arrive to bus stop PA383-5 before 3:22am. In case of service lines 541NI, F30NR and 515NI, the synchronization variable (Y^{ij}_{pqb}) is equal to 1. Otherwise, the synchronization variable with line 418R is equal to 0.

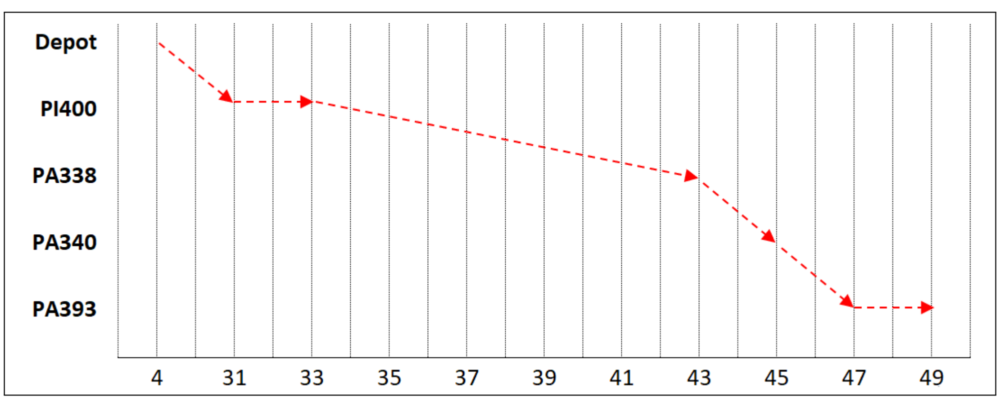

Figure 2.5: Route path of Service line 418R.

For all service lines, timetables can be traced and plotted. I.e. Figure 2.5 is a "space-time" graphic showing the trajectory of the service line 418R through bus stops that belong to its route path (on vertical axis), time is represented on horizontal axis. If trip arrives at bus stop b at moment t then the point (T_{ibp}, t) is on the path. If trip dwells at bus stop b, the point (T_{ibp}, $t + Z_{ib}$) is also in the trajectory. Vehicle speed is seen on the slope and it is shown the departure time of the trip from depot.

2.5 Conclusions and future work

We have proposed a model for the timetabling of night bus services in the context of MMN. The model is an extension of previous approaches by Ceder and Tal (2001), Eranki (2004), Ibarra-Rojas and Rios-Solis (2012) considering fixed headways and potential dwelling times at transfer stations as new features. The model is strengthened with additional constraints that increase the computational performance. The model is applied to a real case scenario of the Bus Rapid Transit night system in Santiago de Chile: Metropolitan Mobility Network, including new system constraints. Results are promising, showing the advantages of performing proper synchronization in the context of setting timetabling.

From the methodology proposed in this chapter, we cannot prove optimality of the model. However, we are solving a real instance at least 6.8 times larger than what is found in the literature (Fouilhoux et al., 2016), measured through interactions between trips of different lines at transfer zones that finally determine synchronizations, giving an idea of the complexity of the studied bus network. The number of interactions that we report is 2510 (against 369 interactions, the maximum reported by Fouilhoux et al., 2016), involving 96 lines (48 in both directions) and 14 transfer zones and reaching gaps around 6%, a good result but showing that there is still to research.

The approach is strengthened using new valid inequalities and Bus rapid transit service inequalities given by the authority. The gaps remains large and according to Ibarra-Rojas and Rios-Solis (2012), they have to be carefully analyzed since Bus Synchronization Timetabling problems and related problems, like ours involving dwelling times, are very complex and it is quite difficult to obtain an accurate measure of the solution quality. A simple but effective routine to improve the computational effectiveness of the model is proposed; numerical results show that quality solutions can be found for large instances in a relatively short time.

Further research will focus on two branches: first, from the real application, we will work on the design of a proper system: potentially defining new routes, involving vehicle and staff scheduling stages, determining real time control strategies for buses and integrating passenger demand behavior. Second, from the modeling and algorithm performance standpoints, despite our promising results, the integer nature of the problem needs to be addressed with more advanced exact techniques that could guarantee optimality.

Conclusions

Performing a transfer is an operation that consumes time and may be very costly. Conceptually, a transfer operation could occur in both, fixed and flexible route transit systems to users and transport managers, so it must be meticulously planned. The literature has shown the benefits of a transfer strategy for transport systems with different degrees of route flexibility: from a fixed route, such as those typically related to traditional public transport systems, to those happening in the context of a flexible route scheme, such as the well known demand responsive transportation systems. The difference is basically whether the routing is a decision for the model or it becomes a parameter.

In the case of fixed route services, the coordination of timetables in a low frequency system is a problem directly related to the synchronization of transfer operations, known in the specialized literature as the Bus Synchronization Timetabling Problem (BST); when timed transfers are involved we denoted this problem as the Bus Synchronization Timetabling Problem with Dwelling Times (BST-DT). We have proposed a model for timetabling of night bus services in the context of Transantiago. The model is an extension of previous approaches by Ceder and Tal, (2001), Eranki (2004), Ibarra-Rojas and Rios-Solis (2012) considering fixed headways and potential dwelling times at transfer stations as new features. The model is strengthened with additional constraints that increase the computational performance. The model is applied to a real case scenario of the Bus Rapid Transit night system in Santiago de Chile: Transantiago, including new system constraints, and results are promising showing the advantages of performing proper synchronization in the context of defining timetabling. The reader is recommended to review the section 2.5 to deepen on our results and findings on BST-DT problems.

In the case of flexible route services, the flexibility of a point to point transit system in case of high-demand and high coverage systems depend on exclusively to the option of performing transit operations, and that is known as the Pickup and Delivery Problem with Transfers (PDP-T). We have introduced a branch-and-price algorithm for the Pickup and Delivery Problem with Transfers. The methodology proposed in this chapter cannot be used to solve large-scale instances containing hundreds of request. However, we are solving now networks of size up to 30 nodes (15 requests), twice larger than Rais et al. (2014) who addressed the problem through a MIP and five times larger than the benchmark of Cortés et al. (2010). Nevertheless, given the current state of the art for the exact solution of VPRTW, we know it does not seem fair to say that these are large instances. This is something that will always be a challenge in the literature specially, in the context of real size problems. The reader is recommended to review the section 1.5 to deepen on our results and findings on PDP-T problems.

Bibliography

[1] Abkowitz, M., Eiger, A. & Engelstein, I. Optimal control of headway variation on transit routes. (1986) Journal of Advanced Transportation, 20, pp. 73-88.

[2] Aksu, D. & Yilmaz, S. Transit coordination with heterogeneous headways. (2014) Transportation Planning and Technology, 37(5), pp. 450-465.

[3] Aldaihani, M., & Dessouky, M. Hybrid scheduling methods for paratransit operations. Computers and Industrial Engineering, 45(1), 75–96. https://doi.org/10.1016/S0360-8352(03)00032-9

[4] Bakker, J., Calkin, J. & Sylvester, S. A Multi-Centered Timed Transfer System for Capital Metro, Austin, Texas. (1988) In: Proceedings of Transportation Research Board 67th Annual Meeting.

[5] Baldacci, R., Bartolini, E. & Mingozzi, A. An exact algorithm for the pickup and delivery problem with time windows. (2011) Operations Research, 59(2), pp. 414-426.

[6] Ball, M.O., Hoffman, R., Odoni, A. & Rifkin, R. A stochastic integer program with dual network structure and its application to the ground-holding problem. (2003) Operations Research, 51(1), pp. 167-171.

[7] Berbeglia, G., Cordeau, J., Gribkovskaia & Laporte, G. Static pickup and delivery problems: a classification scheme and survey. (2007) TOP, 15(1), pp. 1-31.

[8] Berbeglia, G., Pesant, G. & Rousseau, L.M. Checking the feasibility of dial-a-ride instances using constraint programming. (2011) Transportation Science, 45(3), pp. 399-412.

[9] Bettinelli, A., Ceselli, A. & Righini, G. A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows. (2014) Mathematical Programming Computation, 6(2), pp. 171-197.

[10] Boardman, A., Greenberg, D., Vining, A. & Weimer, D. Cost-Benefit Analysis. (2001) Prentice Hall, EEUU.

[11] Bookbinder, J. & and Desilets, A. Transfer optimization in a transit network. (1992) Transportation Science, 26(2), pp. 106-118.

[12] Borndorfer, R., Grotschel, M., Klostermeiner, F. & Kuttner, C. Telebus Berlin: vehicle routing scheduling in a dial a ride system. (1997) Konrad Zuse Zentrum for Information Technik Berlin, 471, pp. 391-422.

[13] Boussier, S., Feillet, D., Gendreau, M. An exact algorithm for team orienteering problems. (2007) 4OR, pp. 211-230.

[14] Braekers, K., Caris, A. & Janssens, G. Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. (2014) Transportation Research Part B: Methodological, 67, pp. 166-186.

[15] Braekers, K. & Kovacs, A. A multi-period dial-a-ride problem with driver consistency. (2016) Transportation Research Part B: Methodological, 94, pp. 355-377.

[16] Calvo, R.W. & Colorni, A. An effective and fast heuristic for the Dial-a-Ride problem. (2007) 4OR, 5(1), pp. 61-73.

[17] Ceder, A., Golany, B. & Tal, O. Creating bus timetables with maximal synchronization. (2001) Transportation Research Part A: Policy and Practice, 35(10), pp. 913-928.

[18] Ceder, A. & Tal, O. Designing synchronization into bus timetables. (2001) Transportation Research Record, (1760), pp. 28-33.

[19] Ceder, A. (2007) Public Transit Planning and Operation: Theory Modeling and Practice. Elsevier, Butterworth-Heinemann.

[20] Cevallos, F. & Zhao, F. A genetic algorithm for bus schedule synchronization. (2006) Proceedings of the Ninth International Conference on Applications of Advanced Technology in Transportation, pp. 737-742.

[21] Chakroborty, P., Deb, K. & Subrahmanyam, P.S. Optimal scheduling of urban transit systems using genetic algorithms. (1995) Journal of Transportation Engineering, 121(6), pp. 544-553.

[22] Chen, D., Cao, E. & Lai, W. A differential evolution algorithm for pickups and deliveries problem with fuzzy time windows. (2016) Journal of Intelligent & Fuzzy Systems, 30(1), pp. 267-277.

[23] Chou, Y., Chen, Y., & Chen, H. Pickup and delivery routing with hub transshipment across flexible time periods for improving dual objectives on workload and waiting time. (2014) Transportation Research Part E: Logistics and Transportation Review, 61, pp. 98-114.

[24] Chowdhury, M. & Chien, S. Optimization of transfer coordination for intermodal transit networks. (2001) In: Proceedings of the 80th Annual Meeting, Transportation Research Board. Washington, DC.

[25] Chowdhury, M. & Chien, S. Intermodal transit system coordination. (2002) Transportation Planning and Technology, 25, pp. 257-287.

[26] Clever, R. Integrated timed transfer: A European perspective. (1997) In: Proceedings of 76th Annual Meeting of the Transportation Board.

[27] Cordeau, J., & Laporte, G. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. (2003) Transportation Research Part B: Methodological, 37(6), pp. 579-594.

[28] Cordeau, J., & Laporte, G. The dial-a-ride problem: models and algorithms. (2007) Annals of Operations Research, 153, 29–46.

[29] Cordeau, J. A branch-and-cut algorithm for the dial-a-ride problem. (2006) Operations Research, 54(3), pp. 573-586.

[30] Cortés, C., & Jayakrishnan, R. (2002). Design and Operational Concepts of High-Coverage Point-to-Point Transit System. Transportation Research Record: Journal of the Transportation Research Board, 1783(02), 178–187. https://doi.org/10.3141/1783-22.

[31] Cortés, C., Matamala, M., & Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. European Journal of Operational Research, 200(3), 711–724. https://doi.org/10.1016/j.ejor.2009.01.022

[32] Cortés, C., Jara-Díaz, S. & Tirachini, A. Integrating short turning and deadheading in the optimization of transit services. (2011) Transportation Research Part A: Policy and Practice, 45(5), pp. 419–434.

[33] Coslovich, L., Pesenti, R. & Ukovich, W. A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. (2006) European Journal of Operation Research, 175, pp. 1605-1615.

[34] Currie, G. The Demand Performance of Bus Rapid Transit. (2005). Journal of Public Transportation, 8(1), 41-55.

[35] Daganzo, C. & Pilachowski, J. Reducing bunching with bus-to-bus cooperation. (2011) Transportation Research Part B, 45(1), pp. 267-277.

[36] Dantzig, G. & Ramser, J. The truck dispatching problem. (1959) Management Science, 6(1), pp. 80-91.

[37] Danna, E., Rothberg, E. & LePape, C. Exploring relaxation induced neighborhoods to improve MIP solutions . (2005) Mathematical Programming, 102(A), pp. 71-90.

[38] Deleplanque, S., & Quilliot, A. (2013). Constraint propagation for the Dial-a-Ride Problem with Split Loads. Studies in Computational Intelligence, 470, 31–50.

[39] Delgado, F., Muñoz, J., Giesen, R. & Cipriano, A. Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. (2009) Transportation Research Record, 2090, pp. 59-67.

[40] Delgado, F., Muñoz, J. & Giesen, R. How much can holding and/or limiting boarding improve transit performance?. (2012) Transportation Research: Part B, 46(9), pp. 1202-1217.

[41] Dessouky, M., Häll, R., Nowroozi, A. & Mourikas, K. Bus dispatching at timed transfer transit stations using bus tracking technology. (1999) Transportation Research: Part C, 7(4), pp. 187-208 .

[42] Dessouky, M., Häll, R., Zhang, L. & Singh, A. Real-time control of buses for schedule coordination at a terminal. (2003) Transportation Research: Part A, 37(2), pp. 145-164.

[43] Diana, M., Dessouky, M. A new regret insertion heuristic for solving large-scale diala-ride problems with time windows. (2004) Transportation Research Part B, 38(6), pp.539-57.

[44] Directorio de transporte publico metropolitano. (2018). Sistema de transporte publico de Santiago: Buses. Santiago de Chile, Chile. Retrieved from http://www.dtpm.cl/index.php/sistema-transporte-publico-santiago/buses

[45] Domschke, W. Schedule synchronization for public transit networks. (1989) ORSpektrum, 11(1), pp. 17-24.

[46] Dondo, R., Méndeza, C. & Cerdá, J. The supply-chain pick-up and delivery problem with transshipments. (2009) In: Proceedings of 19th European symposium on computer aided process engineering, pp. 1009-1014.

[47] Eberlein, X., Wilson, N. & Bernstein, D. The holding problem with realtiem information available. (2001) Transportation Science, 35(1), pp. 1-18 .

[48] Eranki, A. A Model to Create Bus Timetable to Attain Maximum Synchronization Considering Waiting Times at Transfer Stops. (2004) Master Thesis, Department of Industrial and Management Systems Engineering, University of South Florida.

[49] Feillet, D. A tutorial on column generation and branch-and-price for vehicle routing problems. (2010) 4OR, 8(4), pp. 407-424.

[50] Fischetti, M., Lodi, A. Local Branching. (2003) Mathematical Programming, 98(B), pp. 23-47.

[51] Fouilhoux, P., Ibarra-Rojas, O., Kedad-Sidhoum, S. & Rios-Solis, Y. Valid inequalities for the synchronization bus timetabling problem. (2016) European Journal of Operational Research, 251(2), pp. 442-450.

[52] Fügenschuh, A. Solving a school bus scheduling problem with integer programming. (2009) European Journal of Operational Research, 193(3), pp. 867-84.

[53] Garaix, T., Artigues, C., Feillet, D. & Josselin, D. Optimization of occupancy rate in dial-a-ride problems via linear fractional column generation. (2011) Computers and Operations Research, 38(10), pp. 1435-1442.

[54] Ghilas, V., Demir, E., & Van Woensel, T. The pickup and delivery problem with time windows and scheduled lines. (2016) Information Systems and Operational Research, 54(2), pp. 147-167.

[55] Ghilas, V., Demir, E., & Woensel, T. Van. A scenario-based planning for the pickup and delivery problem with time windows, scheduled lines and stochastic demands. (2016c) Transportation Research Part B: Methodological, 91, pp. 34-51.

[56] Ghilas, V., Demir, E., & Van Woensel, T. An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. (2016b) Computers and Operations Research, 72, 12–30. https://doi.org/10.1016/j.cor.2016.01.018

[57] Gørtz, I., Nagarajan, V., & Ravi, R. Minimum Makespan Multi-vehicle Dial-a-Ride. (2009) LNCS (Vol. 5757). Retrieved from http://repository.cmu.edu/tepper

[58] Gschwind, T. A comparison of column-generation approaches to the Synchronized Pickup and Delivery Problem. (2015) European Journal of Operational Research, 247(1), pp. 60-71.

[59] Gschwind, T., & Irnich, S. Effective Handling of Dynamic Time Windows and Its Application to Solving the Dial-a-Ride Problem. (2015) Transportation Science, 49(2), pp. 335-354.

[60] Guihaire, V. & Hao, J. Improving timetable quality in scheduled transit networks. (2010). Lecture Notes in Computer Science , 6096, pp. 21-30.

[61] Häll, C., Andersson, H., Lundgren, J. & Varbrand, P. The Integrated Dial-a-Ride Problem. (2009) Public Transport, 1(1), pp. 39-54.

[62] Häll, C. & Peterson, A. Improving paratransit scheduling using ruin and recreate methods. (2013) Transportation Planning and Technology, 36(4), pp. 377-393.

[63] Hickman, M., & Blume, K. A Method for Scheduling Integrated Transit Service. (2000) In 8th International Conference on Computer-Aided Scheduling of Public Transport (CASPT).

[64] Hoffman, R. & Ball, M. A comparison of formulations for the single-airport ground-holding problem with banking constraints. (2000) Operations Research, 48(4), pp. 578-590.

[65] Horn, M. Procedures for planning multi-leg journeys with fixed-route and demand-responsive passenger transport services. (2004) Transportation Research Part C: Emerging Technologies, 12(1), 33–55. https://doi.org/10.1016/j.trc.2002.08.001

[66] Ibarra-Rojas, O. & Rios-Solis, Y. Synchronization of bus timetabling. (2012) Transportation Research Part B: Methodological, 46(5), pp. 599-614.

[67] Ibarra-Rojas, O., Lopez-Irarragorri, F. & Rios-Solis, Y. Multiperiod bus timetabling. (2016) Transportation Science, 50(3), pp. 805-822.

[68] Ibarra-Rojas, O. & Muñoz, J. Synchronizing different transit lines at common stops considering travel time variability along the day. (2016) Transportmetrica A: Transport Science, 12(8), pp. 751-769.

[69] Jaw, J., Odoni, A., Psaraftis, H. & Wilson, N. A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. (1986) Transportation Research Part B, 20(3), pp. 243-257.

[70] Kachitvichyanukul, V., Sombuntham, P. & Kunnapapdeelert, S. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via PSO. (2015) Computers and Industrial Engineering, 89, pp. 125-136.

[71] Karabuk, S. A nested decomposition approach for solving the paratransit vehicle scheduling problem. (2009) Transportation Research Part B: Methodological, 43(4), pp. 448-465.

[72] Kang, L., Wu. J., Sun, H., Zhu, X. & Gao, Z. A case study on the coordination of last trains for the Beijing subway network. (2015) Transportation Research Part B: Methodological, 72, pp. 112-127.

[73] Kang, L., Wu, J., Sun, H., Zhu, X. & Wang, B. A practical model for last train rescheduling with train delay in urban railway transit networks. (2015b) Omega, 50, pp. 29-42.

[74] Kerivin, H., Lacroix, M., Mahjoub, A. & Quilliot, A. The splittable pickup and delivery problem with reloads. (2008) European Journal of Industrial Engineering, 2(2), 112–133. https://doi.org/10.1504/EJIE.2008.017347

[75] Kim, M. & Schonfeld, P. Integration of conventional and flexible bus services with timed transfers. (2014) Transportation Research Part B, 68, pp. 76-97.

[76] Kyte, M., Stanley, K. & Gleason, E. Planning, implementing, and evaluating a timed-transfer system in Portland, Oregon. (1982) Transportation Research, 877, pp. 23-29.

[77] Lee, K. & Schonfeld, P. Optimal slack time for timed transfers at a transit terminal. (1991) Journal of Advanced Transportation, 25(3), pp. 281-308 .

[78] Li, H. & Lim, A. A metaheuristic for the pickup and delivery problem with time windows. (2001) In: 13th IEEE International Conference on Tools with Artificial Intelligence, pp. 333-340.

[79] Liaw, C., White, C. & Bander, J. A Decision Support System for the Bimodal Dial-a -Ride Problem. (1996) IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 26(5), pp. 552-565.

[80] Lin, C. A cooperative strategy for a vehicle routing problem with pickup and delivery time windows. (2008) Computers and Industrial Engineering, 55(4), 766–782. https://doi.org/10.1016/j.cie.2008.03.001

[81] Liu, Z., Shen, J., Wnag, H. & Yang, W. Regional Bus Timetabling Model with Synchronization. (2007) Journal of Transportation Systems Engineering and Information Technology, 7(2), pp. 109-112.

[82] Madsen, O., Ravn, H. & Rygaard, J. A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. (1995) Annals of Operations Research, 60(1), pp. 193-208.

[83] Masson, R., Lehuédé, F., & Péton, O. The dial-a-ride problem with transfers. (2014) Computers and Operations Research, 41(1), 12–23.

[84] Masson, R., Trentini, A., Lehuédé, F., Malhéné, N., Péton, O., & Tlahig, H. (2017). Optimization of a city logistics transportation system with mixed passengers and goods. EURO Journal on Transportation and Logistics, 6(1), 81–109. https://doi.org/10.1007/s13676-015-0085-5

[85] Masson, R., Lehuédé, F., & Péton, O. An adaptive large neighborhood search for the pickup and delivery problem with transfers. (2013) Transportation Science, 47, 344-355.

[86] Masson, R., Ropke, S., Lehuédé, F., & Péton, O. A branch-and-cut-and-price approach for the pickup and delivery problem with shuttle routes. (2014) European Journal of Operational Research, 236(1), pp. 849-862.

[87] Mitrović-Minic, S. & Laporte, G. The Pickup and Delivery Problem With Time Windows and Transshipment. (2006) Information Systems and Operational Research , 44(3), pp. 217-228.

[88] Muñoz, J., Giesen, R., Ibarra-Rojas, O. & Knapp, P. Scheduling synchronized public transport systems considering holding times and demand assignment. (2014) Working paper. Retrieved in: 17th October 2017 of http://bit.ly/2yw7zWh.

[89] Niu, H., Tian, X. & Zhou, X. Demand-driven train schedule synchronization for high-speed rail lines. (2015) IEEE Transactions on Intelligent Transportation Systems, 16(5), pp. 2642-2652.

[90] Odoni, A. The flow management problem in air traffic control. (1987) In: Proceedings of Flow Control of Congested Networks. Springer, Berlin Heidelberg, pp. 269-288.

[91] Oertel, P. Routing with reloads. (2001) Ph.D. thesis, Faculty of Mathematics and Natural Sciences, University of Cologne, Germany.

[92] Paquette, J., Cordeau, J., Laporte, G. & Pascoal, M. Combining multicriteria analysis and tabu search for dial-a-ride problems. (2013) Transportation Research Part B: Methodological, 52, pp. 1-16.

[93] Parragh, S., Doerner, K. & Hartl, R. Variable neighborhood search for the dial-a-ride problem. (2010) Computers and Operations Research, 37(6), pp. 1129-1138.

[94] Parragh, S. Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. (2011) Transportation Research Part C, 19, pp. 912-930.

[95] Petersen, H. L., & Ropke, S. The Pickup and Delivery Problem with Cross-Docking Opportunity. (2011) In LNCS (Vol. 6971, pp. 101–113). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-24264-9_8

[96] Posada, M., Andersson, H., & Häll, C. The integrated dial-a-ride problem with timetabled fixed route service. (2017) Public Transport. https://doi.org/10.1007/s12469-016-0128-9

[97] Qu, Y. & Bard, J. A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. (2012) Computers and Operations Research, 39(10), pp. 2439-2456.

[98] Qu, Y., & Bard, J. The heterogeneous pickup and delivery problem with configurable vehicle capacity. (2013) Transportation Research Part C: Emerging Technologies, 32, pp. 1-20.

[99] Rais, A., Alvelos, F. & Carvalho, M. New mixed integer-programming model for the pickup-and-delivery problem with transshipment. (2014) European Journal of Operational Research, 235(3), pp. 530-539.

[100] Rapp, M. & Gehner, C. Transfer Optimization in an Interactive Graphic System for Transit Planning. (1967) Transportation Research Record, 619, pp. 22-29.

[101] Raveau, S., Guo, Z., Muñoz, J.C. & Wilson, N. A behavioural comparison of route choice on metro networks: Time, transfers, crowding, topology and socio-demographics. (2014) Transportation Research Part A: Policy and Practice, 66, pp. 185-195.

[102] Rekiek, B., Delchambre, A. & Saleh, H. A. Handicapped person transportation: An application of the grouping genetic algorithm. (2006) Engineering Applications of Artificial Intelligence, 19(5), pp. 511-520.

[103] Righini, G. & Salani, M. New dynamic programming algorithms for the resource constrained elementary shortest path problem. (2007) Networks, 51(3), pp. 155-170.

[104] Richetta, O. & Odoni, A. Solving optimally the static ground-holding policy problem in air traffic control. (1993) Transportation Science, 27(3), pp. 228-238.

[105] Rodrigues, P., Rosa, R., Resendo, L., Mauri, G. & Ribeiro, G. Resolucao de um caso real do problema dial-a-ride multicriterio via clustering search. (2014) Production, 24(3), 572-582.

[106] Ropke, S. & Cordeau, J.F. Branch and cut and price for the pickup and delivery problem with time windows. (2009) Transportation Science, 43(3), pp. 267-286.

[107] Saharidis, G., Dimitropoulos, C. & Skordilis, E. Minimizing waiting times at transitional nodes for public bus transportation in Greece. (2014) Operational Research, 14(3), pp. 341-359.

[108] Sánchez-Martínez, G. Running Time Variability and Resource Allocation: A Data-Driven Analysis of High-Frequency Bus Operations. (2012) Master thesis, Massachusetts Institute of Technology.

[109] Schonberger, J. Scheduling constraints in dial-a-ride problems with transfers: a metaheuristic approach incorporating a cross-route scheduling procedure with postponement opportunities. (2017) Public Transport, 9(1-2), pp. 243-272.

[110] Shafahi, Y., & A. Khani. A Practical Model for Transfer Optimization in a Transit Network: Model Formulations and Solutions. (2010) Transportation Research Part A, 44, 377–389.

[111] Shang, J. & Cuff, C. Multicriteria pickup and delivery problem with transfer opportunity. (1996) Computers and Industrial Engineering, 30(4), 631–645. https://doi.org/10.1016/0360-8352(95)00181-6

[112] Shrivastava, P. & Dhingra, S. Development of Coordinated Schedules Using Genetic Algorithms. (2002) Journal of Transportation Engineering, 128(1), pp. 89-96.

[113] Sun, Y. & Schonfeld, P. Holding decisions for correlated vehicle arrivals at intermodal freight transfer terminals. (2016) Transportation Research Part B, 90, pp. 218-240.

[114] Takoudjou, R.T., Deschamps, J.C. & Dupas, R. A hybrid multistart heuristic for the pickup and delivery problem with and without transshipment. (2012) In: Proceedings of 9th International conference of modeling, optimization and simulation.

[115] Teodorović, D. & Lucić, P. Schedule synchronization in public transit using the fuzzy ant system. (2005) Transportation Planning and Technology, 28(1), pp. 47-76.

[116] Thangiah, S., Fergany, a, & Awan, S. Real-time split-delivery pickup and delivery time window problems with transfers. (2007) Central European Journal of Operations Research. Central European Journal of Operations Research, 15(4), pp. 329-349.

[117] Ting, C.J. & Schonfeld, P. Dispatching control at transfer stations in multi-hub transit networks. (2007) Journal of Advanced Transportation, 41(3), pp. 217-243.

[118] Toth, P., & Vigo, D. Heuristic Algorithms for the Handicapped Persons Transportation Problem. (1997) Transportation Science, 31(1), pp. 60-71.

[119] Trentini, A., & Mahléné, N. Toward a shared urban transport system ensuring passengers & goods cohabitation. (2010) Journal of Land Use, Mobility and Enviroment, 3(2), 37–44. Retrieved from http://www.serena.unina.it/index.php/tema/article/view/165

[120] Verma, A. & Dhingra, S. Developing Integrated Schedules for Urban Rail and Feeder Bus Operation. (2006) Journal of Urban Planning and Development, 132(3), pp. 138-146.

[121] Vranas, P., Bertsimas, D. & Odoni, A. The multi-airport ground-holding problem in air traffic control. (1994) Operations Research, 42(2), pp. 249-261.

[122] Wang, Y., Tang, T., Ning, B., Van den Boom, T. & De Schutter, B. Passenger-demands-oriented train scheduling for an urban rail transit network. (2015) Transportation Research Part C: Emerging Technologies, 60, pp. 1-23.

[123] Wong, R., Yuen, T., Fung, K. & Leung, J. Optimizing timetable synchronization for rail mass transit. (2008) Transportation Science, 42(1), pp. 57-69.

[124] Wu, J., Liu, M., Sun, H., Li, T., Gao, Z. & Wang, D. Equity based timetable synchronization optimization in urban subway network. (2015) Transportation Research Part C: Emerging Technologies, 51, pp. 1-18.

[125] Wu, Y. & Tang, J. Optimizing timetable synchronization for regional public transit with minimum transfer waiting times. (2012) Proceedings of the 24th Chinese Control and Decision Conference, CCDC 2012, art. no. 6244608, pp. 3782-3786.

[126] Wu, Y., Tang, J. & Zhang, Y. Optimizing reliable timetable for bus transit network: Model formulation and solution. (2013) 25th Chinese Control and Decision Conference, CCDC 2013, art. no. 6561231, pp. 1835-1840.

[127] Wu, Y., Yang, H., Tang, J. & Yu, Y. Multi-objective re-synchronizing of bus timetable: Model, complexity and solution. (2016) Transportation Research Part C: Emerging Technologies, 67, pp. 149-168.

[128] Zhao, J., Dessouky, M. & Bukkapatnam, S. Optimal Slack Time for Schedule-Based Transit Operations. (2006) Transportation Science, 40(4), pp. 529-539.

[129] Zidi, I., Mesghouni, K., Zidi, K. & Ghedira, K. multi-objective simulated annealing for the multi-criteria dial a ride problem. (2012) Engineering Applications of Artificial Intelligence, 25(6), pp. 1121-1131.