EURO Journal on Transportation and Logistics 14 (2025) 100164

Contents lists available at ScienceDirect

EURO Journal on Transportation and Logistics

journal homepage: www.elsevier.com/locate/ejtl

Solving the line planning problem with service-levels using a column generation-based heuristic algorithm

Hector Gatt [a,b,1], Jean-Marie Freche [b], Fabien Lehuédé [a,*], Thomas G. Yeung [a,2]

[a] IMT Atlantique, 4 Rue Alfred Kastler, 44300, Nantes, France
[b] Lumiplan, 1 Impasse Augustin Fresnel, 44800, Saint-Herblain, France

ARTICLE INFO

Keywords:
Bus network design
Transit network design
Line planning
Column generation
Column enumeration
Service-levels

ABSTRACT

This research investigates the strategic design of bus line networks by incorporating the operational objectives and constraints relevant to network operators. Drawing on industry expertise from a public transportation software company, we formulate the problem as a line planning optimization problem aimed at minimizing network operating costs while simultaneously addressing two critical service quality metrics: the deviation between passengers' actual travel times and their respective shortest-path travel times, and the proportion of passengers completing trips without transfers. To address problem instances of realistic scale and complexity, we develop and implement a column generation heuristic framework wherein both bus lines and passenger routing paths are generated dynamically throughout the solution process. Our methodological approach combines column generation and column enumeration algorithms, demonstrating computational efficiency when evaluated against established benchmark instances. Furthermore, we present a comprehensive case study examining a medium-sized French municipality, through which we demonstrate the practical integration of real-world design principles within our optimization framework. This empirical analysis yields novel insights regarding the sensitivity of both our proposed parameters and commonly employed parameters in bus network design literature.

1. Introduction

This paper is motivated by the challenge faced by a software company seeking to provide network design tools to public transport operators. As part of this software, we explore how to propose bus lines that meet transportation demand while ensuring a high service-level, with the goal of minimizing the network's operating costs. In view of the various existing models and methods, the Lumiplan company is looking for an approach capable of minimizing the cost of the network while guaranteeing the quality of service that will ensure that the network is used by the interested populations. This quality of service should be based primarily on the number of transfers on a trip, or the time exceeding a trip perceived as direct for a given path. Given the complexity of the problem, implementation aspects also play a major role in the proposed method. Lumiplan is looking for a tool that is both maintainable and flexible. In particular, we are interested in the tool's ability to integrate business rules, for example, or to extend an existing network.

Providing a high-quality bus network at a reasonable cost is the primary goal of many public transportation companies. We consider the design of a set of bus lines served in both directions with an associated theoretical service frequency, so that passenger demands in a predefined area are satisfied. This problem has been defined as the line planning problem (LPP) (Schöbel and Scholl, 2006), bus network design problem (BNDP) (Ceder and Wilson, 1986), or transit network design problem (TNDP) (Magnanti and Wong, 1984). This problem is at a strategical level in the overall planning process for a public transport network, and is modified according to the type of involved city. For example, a city in a developed country, with an already established network of metro and bus rapid transit systems, does not require the same level of restructuring as a city in a developing country, where new lines are often added in response to emerging demand without consideration for the overall network. Most of cities' demands

* Corresponding author.
E-mail addresses: hector.gatt@imt-atlantique.org (H. Gatt), jean-marie.freche@lumiplan.com (J.-M. Freche), fabien.lehuede@imt-atlantique.fr (F. Lehuédé), thomas.yeung@imt-atlantique.fr (T.G. Yeung).
[1] Present address: MFP Michelin, Clermont-Ferrand, France.
[2] Present address: Amazon, Seattle, USA.

https://doi.org/10.1016/j.ejtl.2025.100164
Received 12 May 2023; Received in revised form 8 July 2025; Accepted 27 August 2025
Available online 24 September 2025
2192-4376/© 2025 The Author(s). Published by Elsevier B.V. on behalf of Association of European Operational Research Societies (EURO). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

can be summarized in three different cases: (1) Initial creation of lines, (2) Complete redesign of lines, (3) Redesign of some of the lines. In our work, we propose a heuristic process satisfying each of these three cases.

While capital and operating expenses of the network are the main challenge for the operator, the passenger is also looking for a fast, reliable and convenient network. Therefore, existing line planning models can be classified into two objective types: cost-oriented or passenger-oriented. Cost-oriented models focus primarily on minimizing the total length of bus lines, while passenger-oriented models aim to minimize total travel time for users. Numerous mathematical approaches have been developed for each of these objectives (see Schöbel, 2012 and Cancela et al., 2015). Additionally, Schöbel and Scholl (2006) demonstrate that the line planning problem with budget constraints is NP-hard when passenger path choices depend on the proposed lines. An additional challenge in modeling service-levels for this problem is evaluating the number of transfers on passenger paths based on the selected bus lines. In optimization models, the number of transfers can be assessed by representing the problem on a *change-and-go graph* (Schöbel and Scholl, 2006). Unfortunately, for large instances, the size of this graph makes it intractable for real-world bus network design problems. To address this, Borndörfer and Karbstein (2012) propose the *direct line connection model (DLC)*, which accounts for both the number of passengers with direct paths and those with at least one transfer (indirect paths). To address large-scale instances, the authors introduce the DC model—an approximation of the DLC model—which treats a path as direct if all its arcs are covered by a single line connecting the origin and destination (an rdc-path). This simplification enables a more tractable solution approach for large Line Planning Problem (LPP) instances. Notably, no subsequent studies seem to have leveraged or extended this model to improve performance or tackle other LPP variants.

According to Lumiplan's market analysis, while most bus operators adopt a cost-centric planning approach, they are increasingly required to incorporate service-level constraints into their models. From the passenger's perspective, two primary service-level drawbacks are the number of transfers per trip and the additional travel time incurred relative to a direct origin–destination trip. To address these service-level concerns, we propose an extension of the DC model that embeds a realistic operational cost function along with explicit service-level constraints. To ensure tractability on networks of practical size — typical of medium-scale urban areas — we design a column generation-based heuristic, which is evaluated on a real-world case study from a French city. In summary, our work makes the following contributions:

- We propose a LPP formulation that integrates the operational constraints and technical specifications commonly observed in practice, drawing on insights from a public transport software provider with extensive field experience.
- We extend the DC model of Borndörfer and Karbstein (2012) to solve the resulting LPP.
- We propose a heuristic process, combining column generation and column enumeration methods, to solve the line planning problem with service-level. This approach does not require to implement the column generation approach in a branch-and-price algorithm.
- The generation of bus lines is directly integrated into our program. It thus avoids the need to generate line pools using third-party methods upstream of our program as with most existing methodologies. To our knowledge, this is the first time a column generation algorithm is proposed for bus line generation in a DC model.
- In our experiments, we adapt our model to solve instances from the literature and show the benefits of the bus line generation approach.
- We use a real case study, based on a medium-sized French city, in order to: (i) show the relevance of our method, (ii) demonstrate its flexibility with regards to the integration of practical line design considerations, and (iii) analyze the impact that different input parameters have on the solution.

Interestingly, in the last part, we find that the bus capacity, which is never discussed in academic papers, is a rather fuzzy consideration which has a strong impact on the obtained solutions.

The remainder of the paper is organized as follows: Section 2 focuses on the description of the state of the art. Section 3 describes the line planning problem with service-level (LPP-SL) and formulates a model, extended from the literature and reinforced by service-level and operational constraints, to solve it. This model formulation is then relaxed in Section 4 to be integrated into a process to minimize the cost of a bus network with minimum service-level. For this purpose, this process will include column generation and column enumeration methods to dynamically generate bus lines and passenger paths. Finally, results based on two literature-based instances are presented in Section 5 and a case study is described in Section 6.

2. State of the art

The LPP consists of designing a set of lines on a given infrastructure network and associating a frequency for the vehicles on each line such that the passenger demand, modeled by an origin–destination matrix, is satisfied. The LPP is also referred to as Bus Network Design Problem (Ceder and Wilson, 1986) or Transit Network Design Problem (Magnanti and Wong, 1984) and has been thoroughly investigated over time as illustrated by Guihaire and Hao (2008), Schöbel (2012), Bussieck et al. (1997), Durán-Micco and Vansteenwegen (2021). This section begins with a review of how service-level objectives and constraints have been integrated into optimization models, followed by an overview of recent methodological advances for solving the LPP.

2.1. Integrating service-level evaluation in the LPP

The service-level of a bus network relates to its performance from the passengers point of view. This refers to *user behavior* and should be modeled in a bilevel formulation since passengers have their own strategy. Solving real-size LPP instances with a bilevel model is still a challenge. We note, though, that frequencies can be refined by bilevel optimization after solving the SPP (see Gatt et al. (2022) for the method that follows this contribution). To reduce the complexity, service-level indicators are defined and integrated in the objective or constraints of a mono-level optimization problem. We focus our literature review on papers that follow this approach.

Several models and criteria have been proposed to model service-levels in LPP. Suman et al. (2018) analyze the perceptions of potential passengers of existing bus services in Delhi, India, and conclude that people refrain from using buses due to factors such as indirect paths, overcrowding, time inefficiency compared to alternative modes of transportation, and concerns about punctuality. The European standard (EN-13816, 2002), enacted in September 2002 and still in effect, introduces two key performance measures at the network design scale. The first one is called "possibility to go wherever, whenever". It implies that it should be possible to travel between any pair of stops in the network within

predefined time or distance limits. The second measure, called "convenience", implies that a minimum number of passengers should be offered direct paths for their demand.

Regarding optimization methods, a first approach to assess the service-level of a network involves evaluating the travel time for each passenger on the network. Numerous studies propose to minimize passenger travel times, including in-vehicle travel times and penalties for transfers (Schöbel and Scholl, 2006; Mumford, 2013; Jin et al., 2016). This is modeled in Schöbel and Scholl (2006) by a so-called "change-and-go" graph, where the set of nodes considered is defined for each station and for each line serving that station, and where arcs are defined between two nodes if they are consecutive stations within the same line, or if they belong to the same station and link two different lines. As this graph is large as the number of lines increases, the resulting models can be solved only for small networks. Following a different approach, some papers integrate passenger travel times as a constraint. Thus, some studies enforce all passenger paths to have a duration constrained by a maximum deviation to the shortest in-vehicle travel time of passengers. In particular, Ceder and Wilson (1986) and Baaj and Mahmassani (1995) both force passengers not to use a bus line if the resulting path exceeds a maximum factor of its shortest feasible in-vehicle travel time. Ceder and Wilson (1986) set this parameter to 50% while Baaj and Mahmassani (1995) propose to set it to 40%. Yan (2020) and Bertsimas et al. (2021) propose two models whose objectives are to maximize passenger satisfaction with a maximum network cost limit. In the first model, passengers can only be assigned to a direct path, while in the second model, passengers can be assigned to a direct path or a path with one transfer. For direct paths, a maximum deviation is implemented by filtering rules on lines for passengers with direct paths: the path duration cannot be longer than a factor Γ times the shortest possible in-vehicle travel time. For passengers with indirect paths, this maximum deviation is implemented by filtering flow variables for transfer nodes that are too far from the direct path. The resulting constraint is that the duration of an indirect path cannot be longer than another factor Λ multiplied by Γ times the shortest possible in-vehicle travel time. A drawback of this model is that it considers that passengers with one transfer will accept longer deviations than passengers with a direct path, which does not seem a good approximation of user behavior. Second, the limit of a single transfer per passenger path is very restrictive if serving the whole demand is a constraint. Hence, the model is limited to problems where demand satisfaction is in the objective. Durán-Micco et al. (2022) also use the maximum deviation to the shortest path by using a passenger assignment problem based on Baaj and Mahmassani (1990). In their passenger assignment problem, passengers have a set of attractive paths to reach their destination at most 10% larger than the shortest option. Finally, some studies prefer to set a travel time limit to concurrence the private transport mode. Marín and Jaramillo (2009) propose a model for the urban rapid transit network design problem defined in Marín (2007). The authors introduce a constraint enforcing OD pairs to be assigned to the public transit mode only if the travel time by using public transport is less than the private mode travel time. Gutiérrez-Jarpa et al. (2017) present a discrete, multi-objective mathematical program for strategic planning of a rapid transit system. An origin–destination flow is considered captured only if the estimated time required for the associated passenger to use the network is at most equal to its travel time by car.

The second key aspect of service-level evaluation involves examining the directness it provides. It accounts for the fact that, for a similar travel time, a direct path is always preferred to a path that requires one or more transfers. Hence, the possibility for a passenger to take a direct path is a major asset in terms of attractiveness for a network. In terms of optimization models, not accounting for transfers may lead to assigning high frequencies to lines that will not be used by passengers. To model this, some studies set a constraint on the maximum number of transfers to one on a passenger path (Yu et al., 2012; Duran-Micco et al., 2020; Bertsimas et al., 2021; Durán-Micco et al., 2022). Nevertheless, this may be overly restrictive to connect all origins to destinations with demand, leading to an expensive network. Baaj and Mahmassani (1990, 1995), Mumford (2013) limit the maximum number of transfers to two. Another approach consists in considering all transfer possibilities and integrating a penalty per transfer in the model (Mandl, 1980; Ceder and Wilson, 1986; Baaj and Mahmassani, 1990, 1995; Yu et al., 2012; Mumford, 2013; Jin et al., 2016). For instance, Baaj and Mahmassani (1990, 1995), Mumford (2013) propose to set this transfer penalty to five minutes. Other studies focus on the directness of the network, accounting for the number of passengers with a direct path. Zhao and Ubaka (2004) minimize transfers and optimize line directness while maximizing service coverage. Yu et al. (2012) use an ant-colony optimization method to solve the LPP, maximizing directness and number of direct passengers carried by per unit length of a line under some resource constraints. In a similar way, Suman and Bolia (2019) maximize the number of kilometers traveled by passengers with a direct path. Finally, other papers propose to set a minimum percentage of OD pairs with a direct path. Among them, Baaj and Mahmassani (1990, 1995) propose to set a minimum of 50% of passengers to have a direct path.

2.2. Bus lines generation in methods for the LPP

Two main approaches can be distinguished in the literature to create or select bus lines in optimization models. The first approach consists of separating the LPP resolution in two distinct steps: first, creating a pool of potential lines or using a pool of potential lines created beforehand, and then selecting some interesting lines from this pool. We call this type of approach *a priori line pool generation approach*. The second approach consists of generating the bus lines while solving the LPP, which we call *integrated line generation approach*. The LPP literature is detailed below for each of these two approaches.

The first major challenge for the a priori line pool generation approach is the generation of a qualitative line pool. Among the investigated approaches, Ceder and Wilson (1986) generate a line pool by enumerating all lines with lengths that do not exceed a factor of the shortest path length between their terminus. Baaj and Mahmassani (1990) and Baaj and Mahmassani (1995) concentrate on generating bus lines connecting high-demand node pairs. They consider either the shortest path in terms of in-vehicle travel time or alternate shortest paths within a specific threshold. More recently, Gattermann et al. (2017) generate their line pool based on the solution of a minimum spanning tree problem on the transit network. Camporeale et al. (2019) presents a methodology for generating a line pool using a candidate route set generation procedure.

Given the generated line pool, the LPP becomes a line selection and frequency assignment problem in which passenger paths have to be designed or identified to evaluate service-level criteria. This problem is mainly solved by MIP (mixed-integer programming) approaches, many of which are based on column-generation methods to generate passenger paths. Borndörfer and Karbstein (2012), Karbstein (2013) introduce the direct line connection model (DLC), which distinguishes between direct passenger paths and indirect paths (ie. paths with at least one transfer). The objective function is a weighted sum of line operation costs and passenger travel times and integrates a penalty for the use of indirect paths. The authors introduce the DC model, which approximates DLC through the concept of relaxed direct connection passenger paths (rdc-passenger-paths). In this relaxation, on each arc, at least one line operates, connecting the path's origin to its destination. This approach is further developed in Borndörfer and Karbstein (2015) with the introduction of the complete direct connection model (DC-complete), which is proven equivalent to DLC and easier to solve. In this paper, the relaxed model (DC model), is called DC-basic. Through these papers, the authors show that, although it is a relaxation,

the number of indirect paths estimated by DC-basic is extremely close to the true estimation provided by DLC or DC-complete. On the other hand, the DC-basic formulation can be solved by branch-and-price and it outperforms the other models in terms of solving times and scalability. It is worth noting that Borndörfer and Karbstein (2015) generate all direct paths beforehand, while Borndörfer and Karbstein (2012) use column generation. Our paper uses the terminology (DC model) of Borndörfer and Karbstein (2012). Following this work, Suman and Bolia (2019) proposes a method to optimize existing bus lines, adjusting them based on alternative lines generated from the same terminus with lengths limited by the shortest path factor and maximizing the number of kilometers traveled by direct passengers while considering operational constraints. In all these contributions, a clear limitation of the a priori line pool generation approach is that the line pool cannot be overly large for the optimization problem to remain tractable. Consequently, there is generally no guarantee that an optimal set of lines is included in the line pools used to solve large instances.

Studies employing an integrated line generation approach are based on two distinct approaches. The first focuses on reinforcing a set of feasible lines generated beforehand during the solving of the LPP, while the second is dedicated to the generation of bus lines directly within the LPP-solving process. Among studies focusing on reinforcing a set of feasible lines already generated, Mandl (1980) proposes a local search heuristic to minimize an objective based on in-vehicle travel times and transfer times. It iteratively improves a set of feasible lines through three strategies: (1) exchanging line segments, (2) adding stations to lines, and (3) removing stations from lines. Mumford (2013) introduces an evolutionary bi-objective framework involving four steps: initial line pool creation, crossover step, repair step, and mutation step. Based on Duran-Micco et al. (2020), Durán-Micco et al. (2022) introduce a bi-objective memetic algorithm, employing a heuristic-based approach for initial line pool generation. Their strategy involves incorporating the k shortest paths between high-demand Origin-Destination (OD) pairs. Subsequently, they integrate an iterative process of population renewal utilizing crossover, mutation, and local search operators. Among studies concentrating on generating bus lines within the LPP solving process and close to ours, several methods are based on solving a restricted master problem and a column generation procedure to generate bus lines. In particular, Borndörfer et al. (2007) propose a model for the LPP. To our knowledge, this approach is the first exact method that dynamically generates bus lines and passenger paths. It uses a path-based formulation for passenger paths and two separate pricing problems to generate lines and paths. The main limitation of Borndörfer et al. (2007) is that it does not model transfers and the resulting passenger paths may be unrealistic. Similarly, Liang et al. (2019) use a column generation process that generates bus lines and passenger paths in a multimodal application where passengers can use bus and metro lines. The problem consists of designing bus lines while the metro lines are fixed. The authors consider only transfers between the two transportation modes, specifying a limited number of transfer possibilities on each passenger path but no overall service-level consideration. Bertsimas et al. (2021), Yan (2020) propose a column generation approach to generate bus lines while modeling the paths taken by passengers as flows on the network. Their model maximizes the satisfaction of the passenger demand, including constraints to limit the network cost. Despite the flow formulation for passenger paths, direct and indirect passenger paths can be distinguished, limiting the number of transfers to a maximum of one per passenger. Line frequencies are chosen from a finite set to avoid associating lines with too low frequencies. Heinrich et al. (2022) consider the complexity of generating lines and paths together in an integrated approach. They consider that each passenger path can be a line. They show that the problem is NP-hard and that no polynomial time resolution is possible except for trees graph when $f^{min} = f^{max}$.

2.3. Positioning in the literature

In this paper, we formulate a LPP that minimizes overall network operating costs while incorporating service-level considerations through two performance indicators: a bound on the deviation of each passenger's actual travel time from their individual shortest-path travel time, and the share of passengers traveling without transfers. The model adopts a path-based formulation to represent both passenger itineraries and bus lines, enabling the incorporation of realistic operational constraints on line and path structures, as detailed in Section 6. Our approach extends the framework proposed by Borndörfer and Karbstein (2012), which effectively captures directness in path-based models, adapting it to accommodate our specific objective function and service-level constraints. To address the computational complexity of solving large instances, we develop a column generation heuristic that independently generates candidate bus lines, direct passenger paths, and paths involving transfers.

3. The line planning problem with service-levels

The LPP consists of designing the optimal set of lines and their theoretical operating frequencies so that all passengers can travel along feasible paths while meeting operational and resource requirements, at a minimal estimated cost for the operator.

3.1. Problem formulation

The infrastructure network is modeled on a graph $G = (V, A)$ where V represents a set of station nodes and A a set of arcs. Each of the arcs $a \in A$ is associated with an in-vehicle travel time r_a, a kilometric distance d_a, and a maximum capacity s_a. The maximum capacity of an arc, representing the maximum number of buses allowed to run on that arc, is used to avoid saturation at the bus stations. A set of terminal stations is defined as a subset of the set of station nodes. This set must be chosen with care. By choosing many stations, we increase the ability to generate potentially interesting lines and thus increasing the possibility of obtaining a low-cost network. However, their generation takes more time during the execution of the algorithm.

We consider passenger demands defined in an aggregated origin–destination (OD) matrix noted D. The duration for which this matrix is defined is denoted Δ. Each coefficient in this OD matrix is called an OD pair and denoted q_{st}. It represents the number of passengers wishing to travel from station s to station t during the aggregated time period. The shortest in-vehicle travel time of an OD pair (s, t), meaning the time of the shortest path from s to t with respect to the kilometric distances of the arcs, is denoted by $r^{min}_{s,t}$.

To model this problem, we use equipment inputs. These inputs concern the cost per kilometer, the bus capacity, the fleet size, and the maximum number of operable lines. The cost per kilometer, noted c, is an overall cost that takes into account maintenance costs, fuel costs, and driver salary costs. The idea behind this cost is that the kilometric costs defined in the macro step of the line planning problem should be as close as possible to the actual operational costs of the operator. The bus capacity, noted κ, is the maximum number of passengers authorized in each bus. This parameter is determined by analyzing the maximum number of passengers present simultaneously on each bus currently operated and averaging these values. The fleet size, denoted by o, refers to the number of buses simultaneously available for operation, that is, the maximum total number

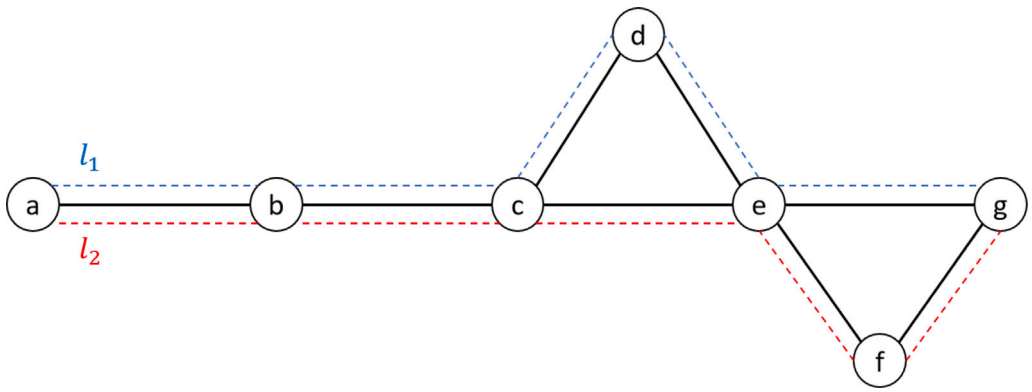


Fig. 1. A small network composed of seven nodes and two bus lines: $l_1 = (a, b, c, d, e, g)$ and $l_2 = (a, b, c, e, f, g)$.

of buses the operator can run at any given time, taking into account driver availability. Finally, the maximum number of operable lines is denoted by n.

To consider the service-level offered by a network, we retain two quality of service parameters: (1) A passenger must have a path to go from his/her origin to destination without deviating by a limit from the shortest path; (2) A minimum percentage of passengers must be able to go from their origin to their destination without transferring. We thus define two corresponding parameters:

- **Maximum-spt-deviation**: Defined as $\alpha \geq 1$, is the maximum deviation allowed for each passenger from its shortest path in time. It can be defined as $\frac{r_p}{r_{s,t}^{min}} \leq \alpha \quad \forall p \in \mathcal{P}_{st}$; where $p \in \mathcal{P}_{st}$ symbolizes the paths from s to t, r_p the in-vehicle travel time of the passenger path p.
- **Minimum-rdc-percentage**: Defined as $\beta \in [0, 1]$, is the minimum percentage of passengers using a rdc-passenger-path (defined below). It is based on the total number of passenger requests over the given time period associated with the OD matrix.

3.2. Problem modeling

This subsection presents the three sets of variables used to model the problem.

A bus line is defined as a sequence of arcs. We denote by $\mathcal{L}$ the set of all bus lines. It is assumed that the number of bus trips on a line is the same in both directions (outbound and inbound). The round-trip distance (in kilometers) of a bus line $l \in \mathcal{L}$ is denoted by d_l, and its round-trip in-vehicle travel time is denoted by r_l. The notation $\mathcal{L}_{st}^{\alpha}(a)$ is an extension of $\mathcal{L}_{st}(a)$, which was first introduced by Borndörfer and Karbstein (2012) to represent the set of lines connecting nodes s and t through arc $a \in A$. Thus, $\mathcal{L}_{st}^{\alpha}(a)$ denotes the set of lines connecting nodes s and t through arc $a \in A$, while also respecting the service-level parameter α.

Similar to a bus line, a passenger path is composed of a sequence of arcs. We define two types of passenger path sets: the rdc-passenger-paths set and the indirect-passenger-paths set.

The relaxed-direct-connection (rdc) Passenger Path Set, denoted by $\mathcal{P}^{0+}$, is used to approximate the set of direct passenger paths. We denote by $\mathcal{P}_{st}^{0+}$ the set of rdc-passenger paths for the OD pair (s, t).

As introduced by Borndörfer and Karbstein (2012), this set is constructed by collecting, for each OD pair $(s, t) \in D$, all paths contained in the graph $G_{(s,t)} = (V_{(s,t)}, A_{(s,t)})$, where $V_{(s,t)}$ and $A_{(s,t)}$ represent the nodes and arcs, respectively, that are served by at least one direct passenger path connecting s to t. In other words, a passenger path belongs to the rdc-passenger path set if, on each of its arcs, there exists at least one line $l \in \mathcal{L}$ that operates on the arc and provides a connection from the origin to the destination of the path.

In Fig. 1, the passenger path $p = (a, b, c, e, g)$ is an rdc-passenger path from a to g, but not a direct passenger path. In other words, there is no single line operating along the entire path $p = (a, b, c, e, g)$, but there exists at least one direct line from a to g that operates on each individual arc of the path. The kilometric distance of a rdc-passenger-path $p \in \mathcal{P}^{0+}$ is noted $d_p = \sum_{a \in p} d_a$ and its in-vehicle travel time is noted $r_p = \sum_{a \in p} r_a$. The main interest of creating the rdc-passenger paths set is that it enables approximating the number of transfers in a column generation approach and thus reduces the complexity of pricing the path variables of connecting passengers. By accepting an overestimation of the number of direct connections, Borndörfer and Karbstein (2012) propose a constraint, called DC (Direct Connection) constraint, to omit the specific coverage of paths by bus lines but to focus only on the arcs covered by bus lines. Thus, the use of rdc-passenger-paths instead of direct passenger paths avoids the need to define the passenger path in terms of individual lines iterating over the lines during the pricing problems. rdc-passenger-path pricing problems are then assimilated to shortest path sub-problems. Borndörfer and Karbstein (2012, 2015) show that this approach provides a very good approximation of the actual number of direct and indirect paths in the problem solutions.

The indirect-passenger-path Set is denoted by $\mathcal{P}^1$. It contains all passenger paths for which at least one bus line runs on each of its arcs, but none of which can make the path directly. On Fig. 1, we can see for example that path (d, e, f) is indirect because there is at least one line on each of its arcs but none of these lines allows a passenger to go directly from d to f. We note $\mathcal{P}_{st}^1$ the set of indirect-passenger-paths for the (s, t) OD pair. The kilometric distance of an indirect-passenger-path $p \in \mathcal{P}^1$ is noted $d_p = \sum_{a \in p} d_a$ and its in-vehicle travel time is noted $r_p = \sum_{a \in p} r_a$. Table 1 summarizes the notations used in *LPP-SL*.

3.3. The model

In this model, we use the following variables:

Table 1
Main notation used to modelize the LPP.

$G = (V, A)$	Graph representing the infrastructure
V	Set of main stations
A	Set of road sections
Δ	Time duration of the considered time period
D	OD matrix modeling passenger demand
q_{st}	Quantity of passengers requesting to go from s to t
$r^{min}_{s,t}$	Shortest in-vehicle travel time from s to t
$\mathcal{P}^{0+}$	Set of rdc-passenger paths
$\mathcal{P}^{0+}_{st}$	Set of rdc-passenger paths from s to t
$\mathcal{P}^{1}$	Set of indirect passenger paths
$\mathcal{P}^{1}_{st}$	Set of indirect passenger paths from s to t
$\mathcal{L}$	Set of bus lines
$\mathcal{L}^{\alpha}_{st}(a)$	Set of lines connecting s and t through arc a with a valid SPT deviation
f_{min}	Minimum frequency for an operated line
f_{max}	Maximum frequency for an operated line
n	Maximum number of lines to be operated
o	Number of operable buses
κ	Passenger capacity of buses
s_a	Arc a saturation threshold
d_a	Kilometric distance of the arc $a \in A$
d_p	Kilometric distance of the passenger path p
d_l	Round-trip kilometric distance of the line l
r_a	In-vehicle travel time associated to the arc $a \in A$
r_l	Round-trip travel time of a bus on the line $l \in L$
r_p	In-vehicle travel time of the passenger path $p \in P$
c	Cost per kilometer

- $z_l = \begin{cases} 1, & \text{if the bus line } l \in \mathcal{L} \text{ is opened} \\ 0, & \text{otherwise} \end{cases}$
- $f_l \in \mathbb{N}$: operating frequency of the bus line $l \in \mathcal{L}$ if opened. It defines the number of bus passages for each of the two directions of the line during a given period. In this study, we assume line frequencies to be symmetric in each of the two directions and to be between f_{min} and f_{max}
- $y^{0+}_p \in \mathbb{R}_+$: quantity of passengers assigned to the rdc-passenger-path $p \in \mathcal{P}^{0+}$
- $y^{1}_p \in \mathbb{R}_+$: quantity of passengers assigned to the indirect-passenger-path $p \in \mathcal{P}^{1}$

LPP-SL

$$\text{Min} \quad \lambda \sum_{l \in \mathcal{L}} c\, d_l\, f_l + (1-\lambda)\left(\sum_{p \in \mathcal{P}^{0+}} r_p\, y^{0+}_p + \sum_{p \in \mathcal{P}^{1}} r_p\, y^{1}_p \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}^{0+}_{st}} y^{0+}_p + \sum_{p \in \mathcal{P}^{1}_{st}} y^{1}_p = q_{st} \qquad \forall (s,t) \in D \tag{2}$$

$$\sum_{p \in \mathcal{P}^{0+}\,:\, a \in p} y^{0+}_p + \sum_{p \in \mathcal{P}^{1}\,:\, a \in p} y^{1}_p \le \sum_{l \in \mathcal{L}\,:\, a \in l} \kappa\, f_l \qquad \forall a \in A \tag{3}$$

$$\sum_{(u,v) \in D\,:\, \mathcal{L}^{\alpha}_{uv}(a) \subseteq \mathcal{L}^{\alpha}_{st}(a)} \; \sum_{p \in \mathcal{P}^{0+}_{uv}(a)} y^{0+}_p \le \sum_{l \in \mathcal{L}^{\alpha}_{st}(a)} \kappa\, f_l \qquad \forall a \in A,\ \forall (s,t) \in D \tag{4}$$

$$\sum_{l \in \mathcal{L}\,:\, a \in l} f_l \le s_a \qquad \forall a \in A \tag{5}$$

$$\sum_{p \in \mathcal{P}^{0+}} y^{0+}_p \ge \sum_{(s,t) \in D} \beta\, q_{st} \tag{6}$$

$$\sum_{l \in \mathcal{L}} r_l\, f_l \le o\, \Delta \tag{7}$$

$$f_l \ge f_{min}\, z_l \qquad \forall l \in \mathcal{L} \tag{8}$$

$$f_l \le f_{max}\, z_l \qquad \forall l \in \mathcal{L} \tag{9}$$

$$\sum_{l \in \mathcal{L}} z_l \le n \tag{10}$$

$$y^{0+}_p \in \mathbb{R}_+ \qquad \forall p \in \mathcal{P}^{0+}$$

$$y^{1}_p \in \mathbb{R}_+ \qquad \forall p \in \mathcal{P}^{1}$$

$$f_l \in \mathbb{N} \qquad \forall l \in \mathcal{L}$$

$$z_l \in \{0, 1\} \qquad \forall l \in \mathcal{L}$$

The *objective function* (1) minimizes the weighted sum of the total kilometric costs and total passenger in-vehicle travel times. A parameter $\lambda \in [0, 1]$ is used to model the weight of each of these two objectives. In this function, the kilometric cost of a line is defined as the total distance

traveled by buses on that line, multiplied by a per-unit distance cost specified by the operator. The total passenger in-vehicle travel times are defined by the sum of the in-vehicle travel times of the passenger paths multiplied by the number of passengers using these paths.

Constraints (2) force each OD pair to be satisfied. Constraints (3) ensure a sufficient line capacity on each arc to transport all passengers. Constraints (4) ensure an approximate sufficient line capacity on each arc to transport passengers using a rdc-passenger-path. This constraint eliminates the assignment of rdc-passenger-paths to particular lines and thus the need to enumerate all bus lines during rdc-passenger pricing. Constraints (5) ensure the number of buses using an arc does not exceed the maximum capacity of the arc. Constraint (6) enforces the ratio of passengers having a rdc-passenger-path to be bigger than β. Constraint (7) ensures the availability of the number of buses required for the frequency selected based on the round-trip travel time of the bus lines operated. Constraints (8) and (9) are used to enforce the frequency of each line to belong to a given range. Constraint (10) limits the number of open lines to a maximum threshold.

Constraints (4) ensure that each line operating on every arc maintains a sufficiently high frequency to satisfy passenger demand via an rdc-passenger-path. These constraints are constructed for each arc $a \in A$ based on the dominance relationship between pairs of OD pairs—that is, when the set of lines serving one OD pair is a subset of the lines serving another. A key advantage of this formulation is that it does not require specifying the exact passenger paths covered by bus lines. Instead, it focuses solely on ensuring coverage at the arc level. It should be noted, however, that this approach may lead to an overestimation of the number of direct passengers (see Appendix A for an illustrative example). Nonetheless, experimental results reported in Borndörfer and Karbstein (2015) indicate that the estimation error is typically very low in practice.

Compared to Borndörfer and Karbstein (2012), we remove the penalty on indirect-passenger-paths from the objective function, replacing it with our service-level parameter β. This parameter provides a precise and direct way to favor direct passenger paths. We fix almost all the weight on the total estimated kilometric costs, but keep a small percentage on the total passengers' in-vehicle travel time. Note that the overall passenger in-vehicle travel time is integrated at this step for comparison with the literature but that its weight is set to be negligible in our case study.

3.4. Classes

Equivalence classes were introduced by Borndörfer and Karbstein (2012) to reduce the complexity of the DC constraint and decoupling passenger paths from bus lines in DC constraints. In this section we summarize their idea and provide an example that illustrates how it is used in the model. We presume a graph $G = (V, A)$ where V represents the main stations and A the arcs. Equivalence classes are defined according to the existing rdc-passenger-paths $p \in \mathcal{P}^{0+}$ and bus lines $l \in \mathcal{L}$.

dc-equivalence classes

For each arc $a \in A$, two OD-pairs (u, v) and (s, t) are said dc-equivalent (direct connection) with respect to a, if $\mathcal{L}_{uv}(a) = \mathcal{L}_{st}(a)$ (these OD have the same set of direct lines that include arc a). Then, (s, t) and (u, v) are said to belong to the same equivalence class for arc a. The set of equivalence classes for dc-equivalent OD-pairs with respect to a is denoted by $\tilde{D}(a) = \{[s, t]_a\}$, where (s, t) is one of the OD pairs chosen arbitrarily among all the OD pairs of the class. We note $(u, v) \in [s, t]_a$ if the two OD-pairs (u, v) and (s, t) are said dc-equivalent with respect to a.

dc-dominated classes

Furthermore, based on the equivalence classes definition, an OD pair (u, v) is said dc-dominated with respect to arc a by an OD pair (s, t) if $\mathcal{L}_{uv}(a) \subseteq \mathcal{L}_{st}(a)$. We note $(u, v) \in [s, t]_a^{\leq}$ if the OD-pair (u, v) is dc-dominated by the OD pair (s, t) with respect to a.

dc-dominated classes wrt maximum-spt-deviation

An OD pair (u, v) is said dc-dominated with respect to arc a and with respect to the service-level parameter α by an OD pair (s, t) if $\mathcal{L}^{\alpha}_{uv}(a) \subseteq \mathcal{L}^{\alpha}_{st}(a)$. $\mathcal{L}^{\alpha}_{st}(a)$ symbolizing the set of lines relying s and t through a in a travel time at most equal to $\alpha\, r^{min}_{st}$. We note $(u, v) \in [s, t]_a^{\leq,\alpha}$ if the OD-pair (u, v) is dc-dominated by the OD pair (s, t) with respect to a and with respect to the service-level parameter α.

Considering the LPP-SL model, for each equivalence class $[s, t]_a$, one constraint (4) is defined for each OD pair in the class. Using the dc-domination principle, constraints (4) can be reformulated as follows:

$$\sum_{(u,v)\in[s,t]_a^{\leq,\alpha}} \sum_{p\in\mathcal{P}^{0+}_{uv}(a)} y_p^{0+} \leq \sum_{l\in\mathcal{L}^{\alpha}_{st}(a)} \kappa\, f_l, \qquad \forall a \in A,\ \forall [s, t]_a \in \tilde{D}(a), \tag{11}$$

Example 3.1. Let us illustrate the concept of equivalence classes on the example of Fig. 1. Assuming the service-level α is satisfied for all paths on this network, we observe that lines l_1 and l_2 connect a to c and a to g through the arc (b, c). We thus have $\mathcal{L}^{\alpha}_{ac}(b, c) = \{l_1, l_2\}$ and $\mathcal{L}^{\alpha}_{ag}(b, c) = \{l_1, l_2\}$. As $\mathcal{L}^{\alpha}_{ac}(b, c) = \mathcal{L}^{\alpha}_{ag}(b, c)$, we thus can say that OD-pairs (a, c) and (a, g) are dc-equivalent with respect to arc (b, c) and we denote this equivalence class by $[a, c]_{(b,c)}$.

Supposing equivalence classes are not used and defining constraints (4) for both OD, the following two constraints are derived:

$$\sum_{(u,v)\in D\,:\,\mathcal{L}^{\alpha}_{uv}(b,c)\subseteq\mathcal{L}^{\alpha}_{ac}(b,c)} \sum_{p\in\mathcal{P}^{0+}_{uv}(b,c)} y_p^{0+} \leq \sum_{l\in\mathcal{L}^{\alpha}_{ac}(b,c)} \kappa\, f_l$$
$$\sum_{(u,v)\in D\,:\,\mathcal{L}^{\alpha}_{uv}(b,c)\subseteq\mathcal{L}^{\alpha}_{ag}(b,c)} \sum_{p\in\mathcal{P}^{0+}_{uv}(b,c)} y_p^{0+} \leq \sum_{l\in\mathcal{L}^{\alpha}_{ag}(b,c)} \kappa\, f_l$$

Since $\mathcal{L}^{\alpha}_{ac}(b, c)$ and $\mathcal{L}^{\alpha}_{ag}(b, c)$ refer to the same set of lines, these two constraints are identical. Hence, equivalence classes avoid duplicating these constraints.

4. Column-generation for the LPP-SL

The LPP-SL model supposes that all feasible bus lines, direct and indirect passenger paths, are enumerated before solving the problem. This assumption reduces the direct application of the model to transportation networks of limited sizes. Accordingly, we propose a column generation approach to generate the associated variables while solving the problem.

The column generation master problem is introduced in Section 4.1. The proposed method is composed of a first phase for column generation followed by a second phase for column enumeration. The general process is described in Section 4.2. The dual problem of the master problem is then defined with three pricing problems for the generation of bus lines and passenger paths. The enumeration step is then described in Section 4.4.

It should be noted that, because the lines and paths are generated independently in this approach, while their variables are in the same constraints, the resulting solution is not proven optimal. This is further discussed in Section 4.5.

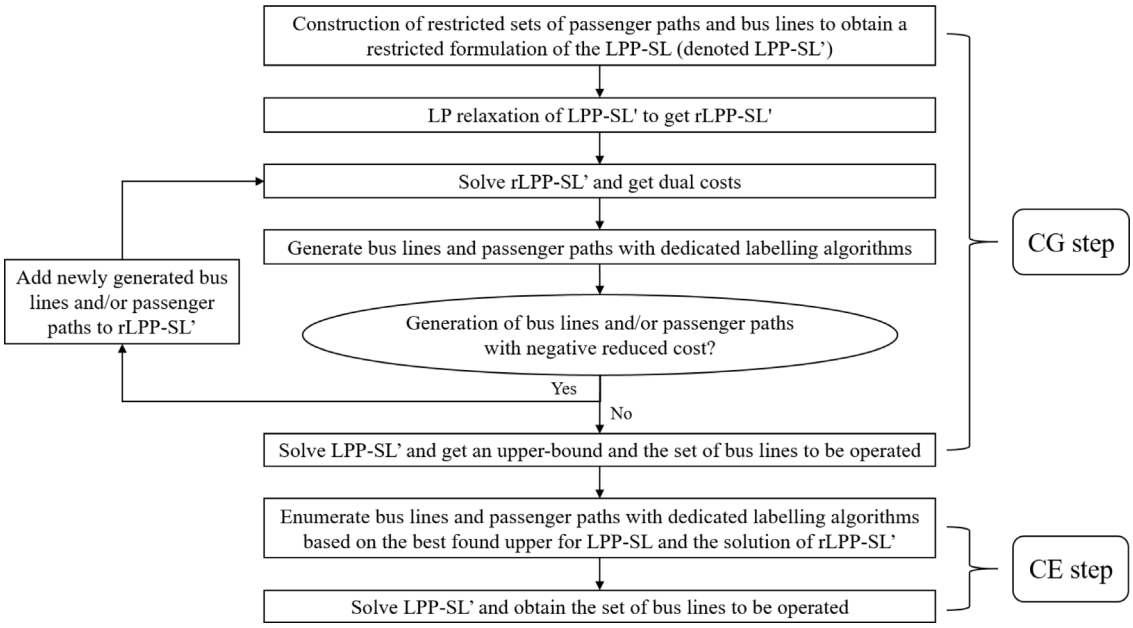


Fig. 2. General overview of the column generation and enumeration process (CGEP) implemented to solve the LPP-SL.

4.1. Column generation master problem

The master problem solved in the LP relaxation of LPP-SL is denoted by rLPP-SL. Similarly to Borndörfer et al. (2007), the model can be simplified in this LP relaxation by substituting z_l by f_l/f_{max} and simplifying constraints (8)–(10) accordingly. The resulting formulation is equivalent with a reduced number of variables and constraints. It is provided below:

rLPP-SL

$$\text{Min} \quad \lambda \sum_{l\in\mathcal{L}} c\, d_l\, f_l + (1-\lambda)\left(\sum_{p\in\mathcal{P}^{0+}} r_p\, y_p^{0+} + \sum_{p\in\mathcal{P}^{1}} r_p\, y_p^{1}\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{p\in\mathcal{P}^{0+}_{st}} y_p^{0+} + \sum_{p\in\mathcal{P}^{1}_{st}} y_p^{1} = q_{st} \qquad \forall (s,t)\in D \qquad (\pi_{st}) \tag{2}$$

$$\sum_{p\in\mathcal{P}^{0+}\,:\,a\in p} y_p^{0+} + \sum_{p\in\mathcal{P}^{1}\,:\,a\in p} y_p^{1} \le \sum_{l\in\mathcal{L}\,:\,a\in l} \kappa\, f_l \qquad \forall a\in A \qquad (\mu_a) \tag{3}$$

$$\sum_{(u,v)\in[s,t]_a^{\le,\alpha}} \sum_{p\in\mathcal{P}^{0+}_{uv}(a)} y_p^{0+} \le \sum_{l\in\mathcal{L}^{\alpha}_{st}(a)} \kappa\, f_l \qquad \forall a\in A,\ \forall [s,t]_a\in\tilde{D}(a) \qquad (\nu_{a,[s,t]_a}) \tag{11}$$

$$\sum_{l\in\mathcal{L}\,:\,a\in l} f_l \le s_a \qquad \forall a\in A \qquad (\eta_a) \tag{5}$$

$$\sum_{p\in\mathcal{P}^{0+}} y_p^{0+} \ge \sum_{(s,t)\in D} \beta\, q_{st} \qquad (\psi_1) \tag{6}$$

$$\sum_{l\in\mathcal{L}} r_l\, f_l \le o\,\Delta \qquad (\theta) \tag{7}$$

$$\sum_{l\in\mathcal{L}} f_l \le f_{max}\, n \qquad (\omega) \tag{12}$$

$$y_p^{0+} \in \mathbb{R}_+ \qquad \forall p\in\mathcal{P}^{0+}$$

$$y_p^{1} \in \mathbb{R}_+ \qquad \forall p\in\mathcal{P}^{1}$$

$$f_l \in [0,\ f_{max}] \qquad \forall l\in\mathcal{L}$$

Constraints (1)–(3), (5)–(7) are repeated from the LPP-SL model in Section 3.3. Constraints (11) replace constraints (4) using the time-class concept (Section 3.4). Constraints (12) result from the aforementioned variable substitution and subsequent simplification. They replace constraints (8)–(10) in the LPP-SL model. To each constraint, we associate a dual variable which is used to formulate the dual problem (presented in Appendix B) and the corresponding pricing problems.

4.2. Solution methodology

Our general solution methodology is outlined in Fig. 2. This method is based on the approach of Baldacci et al. (2008). Its general principle has been applied by Glize et al. (2020) to solve multi-vehicle covering tour problems. We denote it *Column Generation and Enumeration Process (CGEP)*. A strength of the approach is that it does not require to implement a branch-and-price algorithm and remains exact for classical VRPs,

while relying on column generation. The first step is called the column generation step (CG). Some initial passenger paths are generated based on the existing lines of the network. In the case of a new network, a module based on the so-called *route generation algorithm* of Ceder and Wilson (1986) generates bus lines until we obtain a set of lines satisfying the service-level parameters chosen by the user. Then, a set of initial passenger paths is generated based on these bus lines. Hence, for each generated bus line, all direct and one-transfer passenger paths are added to an initial set of passenger paths. For OD pairs for which no direct nor one-transfer path are generated for this initial line set, some two-transfer passenger paths are generated and added to the initial passenger paths set. This gives rise to a restricted formulation of LPP-SL and rLPP-SL, denoted LPP-SL' and rLPP-SL', respectively. Then, bus lines, RDC-passenger-paths, and indirect-passenger-paths are generated in the column generation step using two dedicated labeling algorithms: one for bus lines and one for passenger paths. After each iteration of the column generation step, the new lines and paths are added to rLPP-SL' and the equivalence classes are updated according to the definition given in Subsection *3.4*. This process terminates when the labeling algorithm cannot find any line or path with a negative reduced cost. LPP-SL' which is composed of all generated columns is then solved with a commercial solver, thus providing an upper bound to the problem. A column enumeration (CE) step is launched to create a larger subset of lines and paths, generating columns with positive reduced costs. This process is presented in Section 4.4. The resulting LPP-SL', including all generated and enumerated columns, is then solved with a commercial solver, thus providing an upper bound to the problem.

4.3. Pricing problems

We propose two labeling algorithms to generate bus lines and passenger paths for our line planning problem with service-level. These two labeling algorithms both present new contributions: (i) This is the first time a bus line generation algorithm is proposed for the LPP with the DC constraint to model transfers. (ii) For the bus line labeling algorithm, the storage of passengers potentially interested by a bus line label is introduced and an update of the cost of the line label is performed each time a passenger can use the line in a direct way. (iii) For the passenger path labeling algorithm, a verification that the passenger path label can be continued within the service-level parameter α is performed each time a new node is added to the label.

4.3.1. Bus lines

The bus line pricing problem is used to generate bus lines to integrate into the bus line set $\mathcal{L}$. Each of these bus lines has an associated reduced cost $\bar{c}_l$, defined by referring to the first constraint (B.3) of the dual problem (detailed in Appendix B).

$$\bar{c}_l = \lambda\, c\, d_l - \kappa\, (\sum_{a\in l} \mu_a + \sum_{a\in l} \sum_{[s,t]_a \in \bar{D}(a)\,:\, l\in\mathcal{L}^{\alpha}_{st}(a)} \nu_{a,[s,t]_a}) + \sum_{a\in l} \eta_a + r_l\, \theta + \omega \qquad \forall l \in \mathcal{L} \tag{13}$$

Constraints (13) are based on the μ, ν, η, θ and ω dual variables respectively referring to the constraints (3), (11), (5), (7) and (12) of the master problem.

A bus line label ϵ is defined by the tuple (v, c, δ, N, R, S) where v denotes the last visited node, c the reduced cost of the label, δ the distance of label, N the set of nodes visited by the label, S the set of OD pairs satisfied by the label because they have a direct path on its bus line and R the set of OD pairs for which no direct path exist on the label's line, whatever its future extension. The purpose of the sets R and S is to identify which OD pairs can still have a direct path on the line for further extensions. As a reminder, each acceptable bus line must have a length between l_{min} and l_{max}. The algorithm is executed by attempting to generate lines from a node src to a node dst. The initial label is thus defined as: $E_0 = (src, 0, 0, \{src\}, \{\}, \{\})$. The structure of the bus line labeling algorithm is based on four actions: (1) selection of an incomplete label, (2) extension of this label, (3) update of passengers that can have a direct path on the label's line, (4) update of passenger that can no longer have a direct path on the label's line, and (5) dominance update of this label. This third step performs an update of the S attribute of the label while the fourth step performs an update of the R attribute. See Appendices B and C for more details on dual problem and bus line labeling algorithm.

Label extension step. The label extension step is based on extension rules that must be respected when extending a line label ϵ along arc (v_ϵ, j) to create a new label ϵ'. These rules are defined as follows:

- $v_{\epsilon'} \leftarrow j$
- $c_{\epsilon'} \leftarrow c_\epsilon + \lambda\, c\, (d_{v_\epsilon,j} + d_{j,v_\epsilon}) - (\mu_{(v_\epsilon,j)} + \mu_{(j,v_\epsilon)})\, \kappa + \eta_{(v_\epsilon,j)} + \eta_{(j,v_\epsilon)} + \theta\, (r_{v_\epsilon,j} + r_{j,v_\epsilon})$
- $\delta_{\epsilon'} \leftarrow \delta_\epsilon + (d_{v_\epsilon,j} + d_{j,v_\epsilon})$
- $N_{\epsilon'} \leftarrow N_\epsilon \bigcup \{j\}$
- `cost_Line_Update`$(c_{\epsilon'}, S_{\epsilon'})$
- `R_Update`$(R_{\epsilon'})$

`cost_Line_Update` function

`cost_Line_Update` updates the cost $c_{\epsilon'}$ and the set $S_{\epsilon'}$ of a label ϵ' based on the equivalence classes defined on arc $a = (v_\epsilon, j)$. More precisely: for an equivalence class $[s,t]_a$ with $a = (v_\epsilon, j)$, if s belongs to N_ϵ and t is equal to j or the opposite, if the duration of the path from s to t on the created line respects the maximum-spt-deviation, the cost $c_{\epsilon'}$ is updated as follows:

$$c_{\epsilon'} \leftarrow c_{\epsilon'} - \sum_{a\in A_{N^{(s,t)}_{\epsilon'}}} (\nu_{a,[s,t]^{\leq,\alpha}_a} + \nu_{\bar{a},[s,t]^{\leq,\alpha}_{\bar{a}}})\, \kappa \tag{14}$$

where $A_{N^{(s,t)}_{\epsilon'}}$ represents all the arcs on the created line between s and t and $\bar{a}$ represents the reversed arc of a. Finally, the OD pairs that are dc-equivalent to (s,t) with respect to a, and thus belong to the equivalence class $[s,t]_a$, are added to the set $S_{\epsilon'}$.

Note however that the reduced cost corresponding to an OD pair that becomes directly connected by a newly generated line, and does not belong to an existing equivalence class, cannot be integrated before rLPP-SL' is solved with the new lines and paths of the current CG step.

`R_Update` function

The `R_Update` function identifies OD pairs that can no longer have a direct passenger path on the label ϵ' to add them to the $R_{\epsilon'}$ set. Suppose that a label ϵ' is obtained by extending a label ϵ to a node j. For each OD pair $(s,t) \notin R_{\epsilon'} \bigcup S_{\epsilon'}$, the sum of the travel time on the label ϵ' from s to j and the shortest in-vehicle time from j to t is computed. If it exceeds the maximum-spt-deviation, then the OD pair can no longer have a direct passenger path on the label, regardless of future extensions to other nodes. These OD pairs are then added to the set $R_{\epsilon'}$.

Dominance check. The dominance check consists of comparing labels with one another to eliminate those that are proven not to lead to an optimal path (Marques, 2020). A label ϵ_1 is said to be dominated by a label ϵ_2 if the following four conditions are satisfied:

1. $v_{\epsilon_1} = v_{\epsilon_2}$
2. $c_{\epsilon_1} \leq c_{\epsilon_1}$
3. $\delta_{\epsilon_1} \leq \delta_{\epsilon_2}$
4. $R_{\epsilon_1} \bigcup S_{\epsilon_1} \subseteq R_{\epsilon_2} \bigcup S_{\epsilon_2}$

See Appendix C for a simple example illustrating the reduced cost update and the dominance step.

4.3.2. Passenger paths

The passenger path pricing problem is used to generate passenger paths to integrate into the passenger path sets. This sub-problem can be used to generate both rdc-passenger-paths and indirect-passenger-paths. For this, rdc-passenger-paths are associated to the reduced cost $\bar{c}_p^{0+}$ (15) while indirect-passenger-paths are associated to the reduced cost $\bar{c}_p^{1}$ (16).

$$\bar{c}_p^{0+} = -\pi_{uv} + \sum_{a \in p} \mu_a + \sum_{a \in p} \sum_{[s,t]_a \in \bar{D}(a)\,:\,(u,v) \in [s,t]_a^{\leq,\alpha}} v_{a,[s,t]_a} + (1-\lambda)\, r_p - \psi_1 \qquad \forall (u,v) \in D,\ \forall p \in \mathcal{P}_{uv}^{0+} \tag{15}$$

$$\bar{c}_p^{1} = -\pi_{uv} + \sum_{a \in p} \mu_a + (1-\lambda)\, r_p \qquad \forall (u,v) \in D,\ \forall p \in \mathcal{P}_{uv}^{1} \tag{16}$$

(15) and (16) are based on the π, μ, v and ψ_1 dual variables respectively referring to the constraints (2), (3), (11) and (6) of the master problem. A passenger path label is defined by the tuple (v, c, τ, N) where v represents the last node visited by the label, $\bar{c}$ the reduced cost of the label, τ the time of the label and N the set of nodes visited by the label. Recall that every passenger path must respect the service-level parameter α. The structure of the common path generation pricing problem is based on three actions : (1) selection of an incomplete label, (2) extension of this label, (3) dominance checking of this label. See Appendices B and D for more details on dual problem and on the passenger path labeling algorithm.

The **label extension step** is based on extension rules that must be respected when a path label ϵ is extended along an arc (v_ϵ, j) to create a new label ϵ'. They are defined as follows:

- $v_{\epsilon'} \leftarrow j$
- $c_{\epsilon'} \leftarrow c_\epsilon + (1-\lambda)\, r_{v_\epsilon, j} + \mu_{(v_\epsilon, j)}$
- $\tau_{\epsilon'} \leftarrow \tau_\epsilon + r_{v_\epsilon, j}$
- $N_{\epsilon'} \leftarrow N_\epsilon \bigcup \{j\}$

Each time a label is extended along an arc (i,j), it must be checked whether the label meets certain conditions. In the case of a rdc-passenger-path generation, a search is made in the bus lines pool $\mathcal{L}$ to determine whether there is at least one line that includes the arc (i,j) and covers the origin and destination of the path with a in-vehicle travel time respecting the service-level parameter α. In the case of an indirect-passenger-path, a search is made in the bus lines pool $\mathcal{L}$ to determine the existence of at least one line including the arc (i,j). See Appendix D for a small example on this update of the cost.

Finally, a function `cost_rdcPath_Update`, dedicated to the rdc-passenger-paths generation, is executed each time the label is extended to a new node. Let src and dst be the origin and the destination of a rdc-passenger-path and let a denote the arc that extends the label ϵ to generate ϵ'. Let $[s,t]_a$ be an equivalence class that dc-dominates the OD pair (src, dst) with respect to a. The set of all such equivalence class is $\{[s,t]_a \in \bar{D}(a) \ :\ (src, dst) \in [s,t]_a^{\leq,\alpha}\}$. For each equivalence class in this set, the reduced cost of the label $c_{\epsilon'}$ is updated according to the values of the dual variables $v_{a,[s,t]_a}$.

$$c_\epsilon \leftarrow c_\epsilon + \sum_{[s,t]_a \in \bar{D}(a)\,:\,(src,dst) \in [s,t]_a^{\leq,\alpha}} v_{a,[s,t]_a} \tag{17}$$

Following this, a **dominance check** is performed between the extended label and the others. To do this, a label ϵ_1 is said to be dominated by a label ϵ_2 if the following conditions are met:

- $v_{\epsilon_1} = v_{\epsilon_2}$
- $c_{\epsilon_1} \leq c_{\epsilon_1}$
- $\tau_{\epsilon_1} \leq \tau_{\epsilon_2}$
- $N_{\epsilon_1} \subseteq N_{\epsilon_2}$

4.4. Column enumeration step

It is well known in column generation that solving the linear relaxation may overlook columns that are crucial for the optimal solution of the mixed-integer version of the problem. To address this limitation, a common strategy is to generate columns at each node of the branch-and-bound tree, which leads to the implementation of a full branch-and-price algorithm. As an alternative, Baldacci et al. (2008) propose a column enumeration approach for the Vehicle Routing Problem. We adopt a similar strategy, which requires only minor adjustments to the column generation process. This allows us to generate additional bus lines and paths, thereby enhancing the performance of our heuristic.

Table 2
Network details of the two instances considered.

Instance	Nb of nodes	Nb of arcs	Nb of non-zero OD pairs	Total nb of passengers
Dutch	23	60	420	183 582
Sioux-Falls	24	76	528	360 600

Table 3
Borndörfer and Karbstein (2015) parameters for Sioux-Falls and Dutch. The columns list the weight λ in the objective function, the kilometric cost c, the fixed cost for opening a bus line, the maximum load per bus K, and the transfer penalty σ (in minutes) for each indirect-passenger-path used.

Instance	Weight λ	Kilometric cost	Fixed cost	Max Load Per Bus	Transfer penalty
Dutch	0.8	100	100	600	15
Sioux-Falls	0.8	1.96	100	57	15

This column enumeration step is based on the enumeration of all columns whose reduced cost is less than the difference between the best-known upper bound z_{IP} of the problem and its lower bound z_{MP} determined at the output of the column generation step. Each bus line and passenger path having a reduced cost, as defined in (13), (15) and (16) smaller or equal than $z_{IP} - z_{MP}$ is added to the problem which is then solved with the MIP solver, using the previous upper bound as an initial solution. In our heuristic process, we take the solution of the CG step as the upper bound. As the lower bound, we take the solution of rLPP-SL' (although it may not be optimal in our case).

4.5. On the exactness of the approach

Similar to the branch-and-price algorithm, the approach of Baldacci et al. (2008) has been proven to yield an optimal solution for the VRP. Nonetheless, our algorithm remains heuristic, not because of the search process, but because of constraints in the formulation, coupling bus lines and passenger paths variables. Indeed, when a line is generated, some passenger paths are missing in constraints (3) and (11) where paths and lines are linked. Hence, the simplex multipliers that result from these constraints may be inexact and the reduced costs are not correct. In addition, with each addition of new bus lines into the model, the equivalence classes are updated, with an impact in Constraint (11). More precisely, in Eq. (14), an OD pair that becomes directly connected when creating a line, and that requires to create a new equivalence class, cannot be integrated into the reduced cost of the line because the corresponding dual variable does not exist at this moment. Hence, some negative reduced cost lines may not be found because some paths are missing and the evaluation is biased. The same is true when generating passenger paths. Nevertheless, in this paper we use these biased reduced costs to guide the search for lines and paths but keep in mind that the approach is heuristic.

5. Computational results on instances from the literature

In this section, we evaluate CGEP on benchmarks from the literature. We take Borndörfer and Karbstein (2015) as a reference, since these authors also use the DC relaxation. The main difference between the two approaches is the generation of bus lines performed in CGEP, compared to the a priori enumeration of a line pool of Borndörfer and Karbstein (2015). Their computations were done on computers with an Intel(R) Xeon(R) CPU X5672 with 3.20 GHz, 12 MB cache, and 48 GB of RAM and solved by CPLEX 12.4. Borndörfer and Karbstein (2015) implement their own branch-and-price algorithm using the SCIP framework (Achterberg, 2009) to solve their models to optimality. Our models are implemented with Julia and solved by CPLEX 20.1 through the JuMP interface on a DELL R440 1U server with a 2.1 GHz Intel Xeon 6230 CPU and 192 GB of RAM.

5.1. Presentation of instances

To compare our results to Borndörfer and Karbstein (2015), we use Dutch https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_lop.html and Sioux-Falls instances (see Table 2). More details on the Sioux-Falls instance can be found in Appendix F. Concerning the Dutch instance, similar to Borndörfer and Karbstein (2015), we consider that each arc has a length of one kilometer.

We compare our results to those of Borndörfer and Karbstein (2015). Therefore, for both instances, all nodes are set as potential termini. Furthermore, operational constraints concerning the maximum number of lines and the buses fleet availability are relaxed. In the same way, the two service-level constraints concerning parameters α and β are both relaxed. In addition, the definition set of line frequencies is modified so that it can be defined in the same way as Borndörfer and Karbstein (2015). Thus, line frequencies can be set to 3, 6, 9 or 18. Furthermore, the objective function and thus the bus line generation sub-problem are both modified by adding a fixed opening cost to each bus line. For its part, as mentioned before, the overall passenger travel time is integrated in the objective function for comparison with the literature. With these modifications, our model becomes comparable to Borndörfer and Karbstein (2015).

The parameters presented below are those used by Borndörfer and Karbstein (2015) for their benchmarks. They are summarized in Table 3. In addition, we require that bus lines generated for the Sioux-Falls instance have a length at most equal to 0.15.

For both instances, Dutch or Sioux-Falls, Borndörfer and Karbstein (2015) base their experiments on six cases: DC-basic-1, DC-complete-1, DC-basic-2, DC-complete-2, DC-basic-3, DC-complete-3. The value 1, 2 or 3 in the file name refers to the number of lines present in the upstream generated line pool. For each of these cases, the size of the line pool is presented in Tables 4 and 6 in order to compare the number of lines considered by Borndörfer and Karbstein (2015) with the number of lines generated by our program. Moreover, the file name DC-basic or DC-complete refers to the model used by Borndörfer and Karbstein (2015) for their experimentation. For our experiment, we use our process, first using the CG step alone to generate bus lines and passenger paths, and then, using the CGEP. For the CG step, at each iteration, all pricing problems are resolved prior to solving rLPP-SL'. Regarding bus lines generated with a negative reduced cost, the 50 lines with the lowest reduced costs are integrated into the base. For passenger paths generated with a negative reduced cost, all of them are included in the base.

Table 4
Statistics on the Dutch instance for the first experiment. The columns list the number of lines considered, the objective value of the LPP-SL, the in-vehicle travel times (in minutes) of passengers, the kilometric costs, the number of direct passengers expected by the model and the solving time. The CG and CGEP rows present our approach with CG step alone or with CGEP.

Method	# lines considered	Obj value	Travel times	Kilometric costs	Dir. Pass predicted	Solving time
DC-complete-1	402	2 613 305	1.279×10^7	69 500	179 536	15 s
DC-basic-1	402	2 613 305	1.279×10^7	68 900	179 376	15 s
DC-complete-2	2 679	2 609 841	1.279×10^7	64 400	179 796	910 s
DC-basic-2	2 679	2 609 841	1.279×10^7	64 400	179 796	1 668 s
DC-complete-3	7 302	2 609 271	1.278×10^7	65 600	180 234	1 766 s
DC-basic-3	7 302	2 609 271	1.278×10^7	65 600	180 234	17 765 s
CG	349	2 619 250	1.278×10^7	74 700	178 720	26 s
CGEP	2 889	2 608 912	1.277×10^7	66 800	180 138	90 s

Table 5
Solving times (in seconds) for computing the restricted sets of passenger paths and bus lines, computing the DC classes, solving the rLPP-SL' linear problem in the CG step, solving the pricing problems in the CG and CE steps and solving the LPP-SL' MIP at the end of each step for the Dutch instance.

		Restricted sets construction	DC classes computation	rLPP-SL'	Pricing problems		LPP-SL'
					Bus lines	Passenger paths	
CG		0.88	–	–	–	–	–
	Iteration 1	–	0.04	0.07	0.57	0.10	–
	Iteration 2	–	0.28	0.07	0.42	0.19	–
	Iteration 3	–	0.52	0.08	0.81	0.26	–
	Iteration 4	–	0.76	0.09	0.69	0.37	–
	Iteration 5	–	0.83	0.09	0.20	0.46	–
	Iteration 6	–	0.95	0.12	0.27	0.50	–
	Iteration 7	–	1.12	0.09	2.34	0.53	–
	Iteration 8	–	1.33	0.10	1.88	0.64	–
	Iteration 9	–	1.31	0.09	1.33	0.63	–
	Iteration 10	–	1.35	–	–	–	2.13
CE		–	–	–	1.31	2.74	–
		–	12.70	–	–	–	47.59

5.1.1. Dutch instance

Table 4 summarizes our results compared to the results of Borndörfer and Karbstein (2015) on the Dutch instance; see Appendix F for more details on this instance. For this experiment, we notice that the CE step is necessary to obtain a better solution than Borndörfer and Karbstein (2015). Indeed, by not using this step, we obtain a solution 0.382% worse than the best of their solutions, while by using CE we obtain a solution 0.014% better. This 0.014% better solution is obtained in less than two minutes with a dynamically generated pool of 2889 lines while the best solution of Borndörfer and Karbstein (2015) is obtained in half an hour with an a priori generated line pool composed of 7302 lines. All problems in Borndörfer and Karbstein (2015) are solved to optimality. In addition to the difference in computers or CPLEX versions, we see two major reasons for the difference in solving times between the two approaches: First, CGEP does not require the implementation of a branch-and-price algorithm and makes full use of the powerful dynamic search of the solver. Second, CGEP solves sub-problems of increasing sizes and implements a warm start procedure in which the solution of one sub-problem is used as a lower bound for the next model.

Table 5 describes the solving times (in seconds) for solving the rLPP-SL' linear problem in the CG step, solving the pricing problems in the CG and CE steps, and solving the LPP-SL' MIP at the end of each step for the Dutch instance. We observe that the solving times of pricing problems for bus lines are on average 0.91 s and passenger paths are on average 0.63 s for this Dutch instance. These solving times increase throughout the execution of the CG step. However, we observe that the solving time of rLPP-SL' remains constant throughout the execution of the CG step. Furthermore, we observe that the time duration dedicated to the computation of DC classes increases throughout the execution of the CG step (from 0.04 s at the start to 1.35 s at the end) and even rises to 227.44 s in the CE step once the enumeration of columns has been completed. Finally, we observe that the solving time of LPP-SL' rises from 2.13 s at the end of CG to 47.59 s at the end of CE.

5.1.2. Sioux-Falls instance

Table 6 summarizes our results compared to the results of Borndörfer and Karbstein (2015) on the Sioux-Falls instance; see F for more details on this instance. We see that the column enumeration (CE) step has little effect on the obtained solution. Indeed, a solution 0.929% better than Borndörfer and Karbstein (2015) is obtained in 13 minutes without CE, while the best solution of Borndörfer and Karbstein (2015) takes 5 hours. And we get a solution 0.930% better than Borndörfer and Karbstein (2015) with the complete CGEP in 45 minutes. Concerning the solution obtained with the only CG step, it is obtained with only 400 lines generated during the column generation process. This number is quite smaller than the number of lines considered by Borndörfer and Karbstein (2015) in their sub instances DC-complete-1, DC-basic-1, DC-complete-2, DC-basic-2, DC-complete-3 and DC-basic-3. Hence, CG generates high-quality bus lines for this network.

Concerning the travel time of the passengers, CGEP provides a solution that is 0.657% better than the best result of Borndörfer and Karbstein (2015). Regarding the kilometric cost, CGEP improves the *best* result of Borndörfer and Karbstein (2015) by 29.636%. The direct passengers predicted by the model are the same as those predicted by Borndörfer and Karbstein (2015) with their sub instances DC-basic-1, DC-basic-2 and DC-basic-3.

Table 6
Statistics on the Sioux-Falls instance for the first experiment. The columns list the number of lines considered, the objective value of the LPP-SL, the in-vehicle travel times (in minutes) of passengers, the kilometric costs, the number of direct passengers predicted by the model, the solving time, and the gap to optimality. The CG and CGEP lines present our approach with the single CG step or with CGEP.

Method	# Lines Considered	Obj Value	Travel Times	Kilometric Costs	Dir. Pass Predicted	Solving Time (s)
DC-complete-1	866	649 162	3.198×10^7	12 007	360 585	5 h
DC-basic-1	866	652 238	3.226×10^7	8 801	360 600	5 h
DC-complete-2	9 397	644 432	3.181×10^7	10 312	360 578	5 h
DC-basic-2	9 397	644 365	3.197×10^7	6 094	360 600	5 h
DC-complete-3	15 365	644 432	3.181×10^7	10 312	360 578	5 h
DC-basic-3	15 365	646 392	3.211×10^7	5 200	360 600	5 h
CG	400	638 648	3.176×10^7	4 288	360 600	13 min
CGEP	17 400	638 642	3.176×10^7	4 288	360 600	45 min

Table 7
Solving times (in seconds) for computing the restricted sets of passenger paths and bus lines, computing the DC classes, solving the rLPP-SL' linear problem in the CG step, solving the pricing problems in the CG and CE steps and solving the LPP-SL' MIP at the end of each step for the Sioux-Falls instance.

		Restricted sets construction	DC classes computation	rLPP-SL'	Pricing problems		LPP-SL'
					Bus lines	Passenger paths	
CG		6.15	–	–	–	–	
	Iteration 1	–	0.28	0.11	339.36	4.73	–
	Iteration 2	–	1.06	0.12	129.07	6.52	–
	Iteration 3	–	1.74	0.19	34.19	9.80	–
	Iteration 4	–	2.58	0.16	43.21	14.14	–
	Iteration 5	–	3.17	0.19	51.83	20.84	–
	Iteration 6	–	4.08	0.21	18.00	22.56	–
	Iteration 7	–	4.54	0.26	33.08	28.47	–
	Iteration 8	–	4.86	–	–	–	5.13
CE		–	–	–	15.50	285.72	–
		–	227.44		–	–	1346.97

Similar to Tables 5, 7 provides some details on the time used by each component of the CGEP for the Sioux-Falls instance. We observe that while the processing time for passenger path generation initially exceeds that of bus lines at the start of the column generation stage (CG), there is a gradual convergence and reversal of this trend throughout the execution of the CG stage. Finally, it becomes clear that the processing time for passenger path generation exceeds that of bus lines during the CE stage. This considerable increase in processing time for passenger path generation is due to the extreme number of symmetries in the Sioux-Falls graph, which leads to an exponential number of passenger paths available to passengers as soon as the number of bus lines increases. We also observe that the solving time of rLPP-SL' increases slightly throughout the execution of the CG step (from 0.1 s at the start to 0.26 s at the end). Moreover, we observe that the time duration dedicated to the computation of DC classes increases throughout the execution of the CG step (from 0.28 s at the start to 4.86 s at the end). It becomes significant in the CE step once the enumeration of columns has been completed (227.44 s). Finally, we observe that the solving time of LPP-SL' increases considerably between its execution at the end of the CG step and its execution at the end of CE step. It rises from 5.13 s at the end of CG to 1346.97 s at the end of CE.

6. Case study

This section presents a case study conducted in a medium-sized French city. The objective is to discuss the applicability of the proposed method when applied to real-world data, specifically in generating bus lines that simultaneously satisfy operational and service-level constraints. During the implementation of the models and algorithms in this context, it was observed that the practical relevance of the resulting network could be further enhanced by integrating the typological guidelines for bus lines as defined in Faivre d'Arcier (2014). This is detailed in Section 6.1. Then, the influence of varying key input parameters is examined, with particular attention to how such modifications affect the structural characteristics of the generated network. Finally, we discuss how CGEP performs in solving the LPP-SL for the case study compared to the LPP that were solved in Section 5.

This case study is based on the agglomeration of Poitiers, France, which has more than 130,000 inhabitants. The current bus network operating in Poitiers is managed by the transport authority of Poitiers called RTP (Régie des Transports Poitevin) and operated under the commercial name Vitalis. For our experiment, we analyzed the administrative division of the Poitiers agglomeration and carried out an analysis of passenger movement between the different zones of this division. By linking the aforementioned movement data with boarding and alighting data by stations, we selected 80 major stations among the 322 existing on the network to determine the transportation demands. To connect these major stations, we also analyzed existing bus lines to determine the street sections that could be used by buses. This led to the creation of 218 arcs.

Our solution procedure is shaped by input characteristics of the existing network (illustrated on Fig. 3): 22 lines, with a distance ranging $5-50$ kilometers, and a frequency of between 5 and 80 bus passages during a commercial day. The objective function for this study is defined by a weight $\lambda = 0.99$ for the total kilometric cost in the objective function (1), a cost per kilometer $c = 4.36e$. There is no penalty for transfers, which are integrated in the service-level constraints, and no fixed cost for the number of bus lines, which is limited to 22.

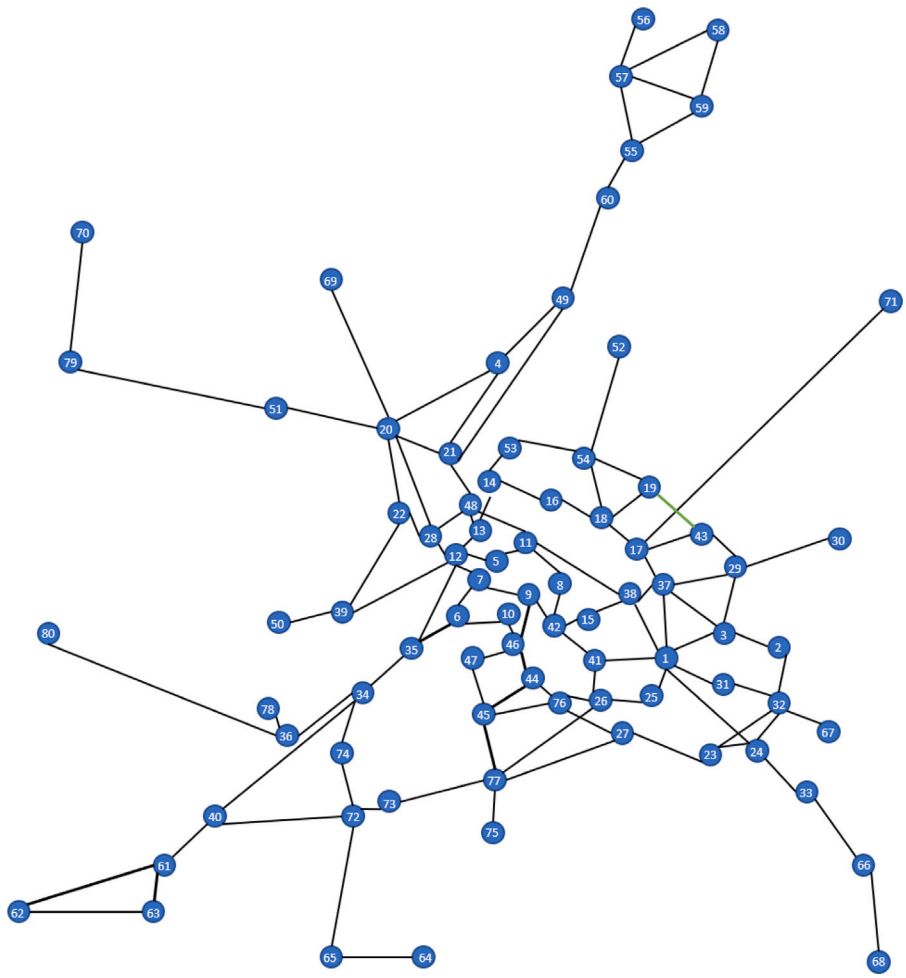


Fig. 3. General overview of the Bus Network of the Poitiers agglomeration.

Table 8
Localization characteristics according to bus line functions according to Faivre d'Arcier (2014).

	Connection		Diffusion	
	Major	Meshing	Major	Secondary
Localization	Inner city and city center	Inner city (excluding city center)	Outskirts/peri-urban to the inner center	Outskirts/peri-urban

6.1. Algorithm adaptation

To treat this case study we performed two adaptations to our column generation algorithm. The first adaptation concerns the initialization of the line and path sets. In order to find feasible solutions for all of our experiments, we start with the set of currently operated network lines and manually modify eight of them so that the α and β level of service parameters can be met with this set of lines. CGEP is then launched and the MILP is solved by using CPLEX with a time limit set to 20 h for CG step and to 10 h for CE step. The longer time associated with CG reduces the gap between the problem lower bound and its upper bound, that determines which columns are enumerated in CE. The second modification is the restriction of the set of generated lines to verify topological criteria and filter unrealistic lines. We consider two types of filtering rules: (1) rules based on lines topology and (2) rules based on lines lengths.

6.1.1. Topology-based filtering rule

The first filtering rule aims at generating more realistic lines with respect to city policies. The applied rules are taken from Faivre d'Arcier (2014), which propose a classification according to the function of each line and its localization (see Table 8). Hence, each function defines a line class: major connection, meshing connection, major diffusion and secondary diffusion. Accordingly, in the bus line labeling algorithm, we filter all lines that do not belong to one of the four classes.

To apply the rules, we separate the Poitiers agglomeration in three areas as illustrated on Fig. 4. Each of them refer either to the city center, inner city and peri-urban area.

According to this definition, the four line classes are illustrated on Fig. 5.

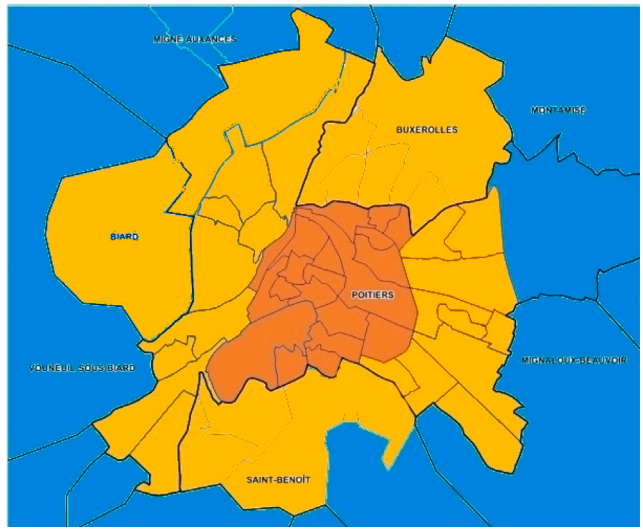


Fig. 4. Overview of the three areas in Poitiers.

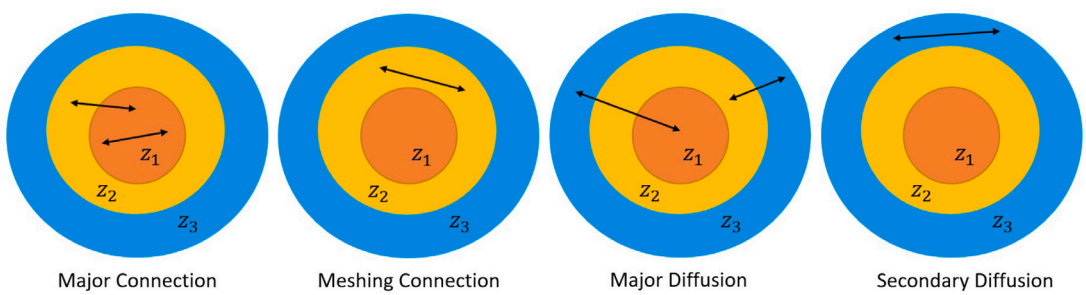


Fig. 5. Illustration of the four line classes of Faivre d'Arcier (2014) on the areas defined for Poitiers.

Table 9
Sinuosity ratios set for the Poitiers instance.

	Connection		Diffusion	
	Major	Meshing	Major	Secondary
Sinuosity ratio	2	1.75	1.5	2

The filtering rules are implemented as sets of possible sequences of zones defined by the line bus stops locations. For example, a line with a sequence of bus stop locations $(Z_3, Z_2, Z_2, Z_2, Z_1, Z_1)$ is a feasible major diffusion line. On the other hand, a line with a sequence of bus stop locations $(Z_2, Z_3, Z_3, Z_2, Z_1, Z_1)$ would be filtered.

6.1.2. *Length-based filtering rules*

We define two rules that verify constraints on line lengths. First, we specify a minimum and a maximum length for a line. These two values are fixed in our case study according to the shortest and longest lines of the existing network of Poitiers: 5 and 50 kilometers, as described above. The second rule limits the line sinuosity taking into account its class. Here, we apply the same rule as Karbstein (2013): we limit the length of a line starting from terminal s and ending at t to be less-than-or-equal-to a ratio of the distance of the shortest path between s and t. We call this ratio, the *sinuosity ratio*. It depends on the topological class of the line defined in Section 6.1.1. The applied values are given in Table 9.

6.2. *Influence of service-level parameters*

The first experiment focuses on the influence of the service-level parameters described in Section 3. The actual service-level of the real bus network of Poitiers were calculated and used as inputs for our re-optimization experiments. The service-level parameter α is 1.75 and β is 72%. We modify these two service-level parameters to analyze the impact on the optimized network obtained.

6.2.1. *Impact of variations of the maximum-spt-deviation (α) service-level parameter for a fixed minimum-rdc-percentage (β) service-level*

Table 10 presents three different scenarios of network design by varying the maximum-spt-deviation service-level parameter (α). The last column shows the estimated commercial kilometers, kilometric costs, and quantity of CO_2 obtained at the output of LPP-SL for each scenario.

From these results, it is clear that adjusting the maximum-spt-deviation (α) service-level parameter has a limited impact on the estimated kilometric costs obtained through CGEP. By tightening this parameter from its original value of 1.75 to 1.5, the total estimated kilometric costs

Table 10
Perspectives of network redesign with fixed minimum-rdc-percentage parameter.

α	β		Business day (6 h–21 h) Proposed network
2	74%	Estimated commercial kms	9 567 kms
		Estimated kilometric costs (base 4.36 €/km)	41 711 €
		In-vehicle passenger travel times	456 602 min
		Estimated qty of CO_2 (base 1.2 kg/km)	11 480 kgs
1.75	74%	Estimated commercial kms	9 584 kms
		Estimated kilometric costs (base 4.36 €/km)	41 788 €
		In-vehicle passenger travel times	448 206 min
		Estimated qty of CO_2 (base 1.2 kg/km)	11 500 kgs
1.50	74%	Estimated commercial kms	9 702 kms
		Estimated kilometric costs (base 4.36 €/km)	42 302 €
		In-vehicle passenger travel times	442 353 min
		Estimated qty of CO_2 (base 1.2 kg/km)	11 642 kgs

Table 11
Perspectives of network redesign with fixed maximum-spt-deviation parameter.

α	β		Business day (6 h–21 h) Proposed network
1.75	68%	Estimated commercial kms	9 584 kms
		Estimated kilometric costs (base 4.36 €/km)	41 788 €
		In-vehicle passenger travel times	428 896 min
		Estimated qty of CO_2 (base 1.2 kg/km)	11 500 kgs
1.75	74%	Estimated commercial kms	9 584 kms
		Estimated kilometric costs (base 4.36 €/km)	41 788 €
		In-vehicle passenger travel times	448 206 min
		Estimated qty of CO_2 (base 1.2 kg/km)	11 500 kgs
1.75	80%	Estimated commercial kms	10 364 kms
		Estimated kilometric costs (base 4.36 €/km)	45 186 €
		In-vehicle passenger travel times	461 685 min
		Estimated qty of CO_2 (base 1.2 kg/km)	12 437 kgs

increases by 1.23%. On the other hand, loosening the maximum-spt-deviation (α) parameter to 2 results in a decrease of 0.18% of this cost. One explanation for this result is that, in all proposed networks, the majority of passengers experience in-vehicle travel times that are less than 50% longer than their shortest-path alternatives. The maximum-spt deviation constraint primarily affects a small subset of passengers with higher deviations. Notably, improving travel times for these less advantaged passengers is achievable at a relatively modest additional cost. Indeed, when relaxing α from 1.75 to 2, the total travel time estimation is increased by 2.3% while it is reduced by 1.31% when tightening α to 1.5.

6.2.2. *Impact of variations of the minimum-rdc-percentage (β) service-level parameter for a fixed maximum-spt-deviation (α) service-level*

Table 11 presents three different scenarios of network design with several values for the minimum-rdc-percentage service-level parameter (β). The last column shows the estimated commercial kilometers, kilometric costs, and quantity of CO_2 obtained at the output of LPP-SL for each scenario.

From these results, we observe that the minimum-rdc-percentage (β) service-level parameter has varying impacts on the estimated kilometric costs. When the minimum-rdc-percentage is reduced from 74% to 68%, there is no change in the estimated kilometric costs. This indicates that in the minimum cost network with a service-level α of 1.75, a rather high percentage of passengers can be served with an rdc-passenger-path. However, when the minimum-rdc-percentage is increased from 74% to 80%, there is a noticeable increase of 8.14% in the estimated kilometric costs. In this case, the minimum-cost network is no longer sufficient to satisfy the expected service-level, and some lines need to be modified or extended to increase the proportion of passengers with a direct path. Interestingly, increasing the percentage of direct travelers also increases the total estimated in-vehicle passenger travel time (+3.01% for $\beta = 80\%$, −4.31% for $\beta = 68\%$). Since transfer times are not accounted for, some indirect paths may actually be shorter than the direct paths. Furthermore, the more complex network reduces the number of transfers for a minority of passengers, but many others experience an increase in their individual travel times due to these modifications. Nevertheless, the analysis of passenger travel times should be interpreted with caution in the LPP, as the passenger behavior model used here is only a rough approximation. These travel times are more accurately estimated during the frequency refinement step (as presented in Gatt et al. (2022) in our case).

6.2.3. *Influence of average bus load*

The setting of average bus load is a topic not widely discussed in the literature. However, it is a factor that significantly influences the result of the network design algorithm. Thus, this subsection focuses on the influence of the average bus load on the resulting network. An analysis of the three European bus manufacturers Iveco, Mercedes-Benz and Scania showed that standard city buses generally have a capacity of 90 to 100 passengers, of which about 30 are seated, while articulated urban buses generally have a capacity of 150 to 160 passengers, including about 45 seats. However, a bus is not always full throughout the day. Thus, we analyze the data received within the framework of the partnership between Lumiplan and the RTP in order to determine the value that we could give to the average bus load.

Hence, in our LPP-SL model, as for most LPP models in the literature, setting κ to the real bus capacity means that all buses are considered to be operating at full load for the entire planning period. This is of course not realistic and κ should be seen as a target for the average bus load over this period. Accordingly, this value can be seen as a coarse indicator of the congestion levels in buses, which is considerably refined in the

Table 12
Influence of the average bus load value for Poitiers instance.

κ	Kilometric costs	Kilometric costs value evolution	CPLEX gap at timeout
14.2 (−25%)	55 751 €	+33.41%	8.20%
18.9	41 788 €	–	8.25%
23.6 (+25%)	33 471 €	−19.90%	7.77%

Table 13
Insights on the algorithm.

Network	Dutch	Sioux-Falls	Poitiers, $\alpha = 1.75$, $\beta = 74\%$, $\kappa = 18.9$
Problem features			
# OD pairs	420	528	2 797
# passengers	183 582	360 600	38 732
# Nodes	23	24	80
# Arcs	30	38	218
Solution features			
# lines	20	41	22
Total Kilometers	2648	4195	9584
# of buses	260	524	53
Direct percentage	98%	100%	74%
Generated columns and solving times			
# bus lines	2 889	17 400	639
# rdc paths	1 072	21 203	7 874
# indirect paths	1 107	21 304	65 210
Solving time	90 s	45 min	30 h
LPP-SL' optimality gap	0%	0%	8%

frequency setting step. However, while revising a bus network it makes sense to investigate the sensitivity of the proposed network to changes on this target.

In our case study, the average number of passengers in a bus that has been calculated from the operator data on an aggregated day period is equal to 18.9. Note that since the flow on paths is a real value in our model, it makes sense to keep it a real value. We therefore carry out a sensitivity analysis on the impact that this average bus load can have on the optimized network obtained. For this experiment, we consider lower and higher fill-rates of 14.2 and 23.6. An average bus load of 14.2 corresponds to a 25% decrease in the average bus load from the estimated 18.9 while 23.6 corresponds to a 25% increase. In Table 12, we find that increasing the average bus load by 25% from 18.9 to 23.6 generates a decrease of 19.88% of the kilometric costs with very small variations of the CPLEX gap at timeout. On the contrary, decreasing the average load by 25% from 18.9 to 14.2 yields a kilometric costs value increase of 33.38%. From these experiments, it seems that the objective value is highly sensitive to bus load and that the latter must therefore be chosen with care.

6.3. Insights on CGEP

Table 13 presents a synthesis of the three main instances solved using CGEP. This table aims to compare CGEP's performance on the case study with its performance on benchmark instances from the literature. In addition to the various problem characteristics listed, the case study introduces service-level constraints, a maximum number of lines, and a different objective function, primarily focused on minimizing total kilometric cost. We observe that the network in the case study is slightly larger; in particular, the number of arcs is nearly six times higher than in the Sioux-Falls instance. Another major difference lies in the demand: the Dutch and Sioux-Falls cases result in a significantly higher number of required buses compared to Poitiers.

Regarding algorithm performance, the most striking result is how challenging it is for CPLEX to find an optimal solution for the LPP-SL model, compared to the classical LPP solved in the Dutch and Sioux-Falls instances. Despite the relatively small number of generated lines (likely due to the initialization with the existing network and the use of topology-based filtering rules), we allowed CGEP 30 h to solve the problem — most of which was spent solving the LPP-SL', as with the other instances — yet the final MIP still had an optimality gap of 8 This can be attributed to both the additional constraints introduced in the LPP-SL model and the revised objective function.

7. Conclusion

In this paper, we propose a formulation of the LPP that is grounded in the practical experience of a company that supplies equipment and software solutions to public transportation operators. The proposed formulation focuses on cost minimization by explicitly incorporating service-level constraints. These include passenger-oriented criteria such as maximum allowable path deviation and minimum direct connection coverage, enabling a more realistic balance between operational efficiency and service quality. To solve this problem, we develop a heuristic approach combining column generation with column enumeration. This allows us to dynamically generate cost-effective bus lines and assign their daily frequencies, minimizing total kilometric costs for a representative operating day while satisfying both operational and service-level constraints. Computational experiments on two benchmark instances from the literature confirm the effectiveness of our method. Specifically, our Column Generation and Enumeration Procedure (CGEP) consistently produces better solutions than approaches relying on precomputed pools of candidate lines, demonstrating the value of dynamic line generation within the optimization loop. We also apply our methodology to a real-world case

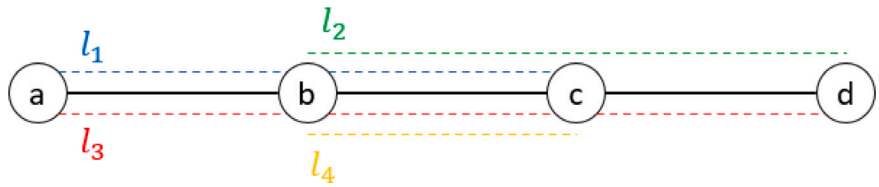


Fig. A.6. A bus network with four nodes : $\{a, b, c, d\}$ and four bus lines : $l_1 = (a, b, c)$, $l_2 = (b, c, d)$, $l_3 = (a, b, c, d)$ and $l_4 = (b, c)$.

study based on a medium-sized French city. This application validates the practical applicability of the approach and provides insights into how service-level parameters influence network design. Notably, we observe that reducing the maximum deviation allowed for each passenger from its shortest path has minimal impact on costs, whereas increasing the minimum percentage of direct passengers significantly raises operational costs. Additionally, we incorporate the city's average bus load into the model as the available bus capacity. Our results show that this parameter is highly sensitive and substantially impacts network structure and cost, highlighting the importance of careful calibration.

As a direction for future work, we plan to extend the model by refining frequency assignments across different time periods, enabling the generation of time-dependent line plans that better match temporal variations in demand. Several directions of research could be followed to improve the scalability of the approach, including generating a better set of initial bus lines, addressing the challenge of feasibility with respect to the new constraints for this initial set, and studying the impact of this initialization. Further theoretical investigations would also be useful to better characterize the solutions obtained with our approach, and evaluate how often an optimal solution is found or not found, and which type of lines and paths are potentially not generated in the process.

CRediT authorship contribution statement

Hector Gatt: Validation, Methodology, Data curation, Writing – review & editing, Software, Investigation, Conceptualization, Writing – original draft, Resources, Formal analysis. **Jean-Marie Freche:** Validation, Investigation, Conceptualization, Resources, Funding acquisition, Project administration, Formal analysis. **Fabien Lehuédé:** Writing – original draft, Project administration, Funding acquisition, Validation, Methodology, Formal analysis, Writing – review & editing, Supervision, Investigation, Conceptualization. **Thomas G. Yeung:** Writing – original draft, Methodology, Validation, Investigation, Supervision, Formal analysis.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly, DeepL, ChatGPT, and Claude.ai to improve the English in some paragraphs. After using these services, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jean-Marie FRECHE reports a relationship with Lumiplan that includes: employment. Hector GATT reports a relationship with Lumiplan that includes: employment. Fabien LEHUEDE reports a relationship with IMT Atlantique Bretagne-Pays de la Loire - Nantes Campus that includes: consulting or advisory. Thomas Yeung reports a relationship with IMT Atlantique Bretagne-Pays de la Loire - Nantes Campus that includes: consulting or advisory.

Acknowledgments

We would like to acknowledge Mrs Karbstein for kindly answering our questions during this research and providing some data for the LPP instances.

Appendix A. Overestimation of the number of direct passengers with DC constraints

This section describes a small example concerning the overestimation of the number of direct passengers with DC constraints.

On Fig. A.6, suppose we have a (a, c) OD pair with 100 passengers, a (b, d) OD pair with 100 passengers and a (a, d) OD pair with 100 passengers. Furthermore, we assume the capacity of a bus to be set at 100 passengers.

By looking at Fig. A.6 we can see that passengers of the (a, c) OD pair can be served with a direct path through the arc $[b, c]$ by the lines l_1 and l_3. Passengers of the (b, d) OD pairs can be served by the lines l_2 and l_3, and passengers of the (a, d) OD pairs can be served by the line l_3. Hence, we have :

- For the arc (a, b), $\mathcal{L}_{ac}(a, b) = \{l_1, l_3\}$, $\mathcal{L}_{bd}(a, b) = \emptyset$, $\mathcal{L}_{ad}(a, b) = \{l_3\}$.
- For the arc (b, c), $\mathcal{L}_{ac}(b, c) = \{l_1, l_3\}$, $\mathcal{L}_{bd}(b, c) = \{l_2, l_3\}$, $\mathcal{L}_{ad}(b, c) = \{l_3\}$.
- For the arc (c, d), $\mathcal{L}_{ac}(c, d) = \emptyset$, $\mathcal{L}_{bd}(c, d) = \{l_2, l_3\}$, $\mathcal{L}_{ad}(c, d) = \{l_3\}$.

The instinctive choice consists to set a frequency of 1 to the lines l_1, l_2 and l_3 and a frequency of 0 to the line l_4. However, setting l_3 to 2 and l_4 to 1 suffices to satisfy the DC constraint even if the (a, c) and the (b, d) passengers cannot use this path without transferring in reality. Indeed, the equivalence classes are defined as follows:

- For the arc (a, b): (a, c) and (a, d) OD pairs have at least one line from their origin to their destination through it. (a, c) and (a, d) do not have the same allowed lines pool, thus we have two equivalence classes, each of them composed of one of the two OD pairs.

- For the arc (b,c): (a,c), (b,d) and (a,d) OD pairs have at least one line from their origin to their destination through it. (a,c), (b,d) and (a,d) do not have the same allowed lines pool, thus we have three equivalence classes, each of them composed of one of the three OD pairs.
- For the arc (c,d): (a,d) and (b,d) OD pairs have at least one line from their origin to their destination through it. (a,d) and (b,d) do not have the same allowed lines pool, thus we have three equivalence classes, each of them composed of one of the three OD pairs.

Thus, we obtain 7 different DC constraints, two for the arc (a,b), three for the arc (b,c) and two for the arc (c,d). Suppose the $((a,b),(b,c))$, $((a,b),(b,c),(c,d))$ and $((b,c),(c,d))$ rdc-passenger-paths are known and present in the rdc-paths pool $\mathcal{P}^{0+}$. Furthermore, suppose all these passenger paths respect the maximum-spt-deviation. These 7 constraints are as follows :

- *For the arc* (a,b) *and the* (a,c) *equivalence class :* both (a,c) and (a,d) OD pairs $\in [a,c]^{\leq,\alpha}_{(a,b)}$ as $L_{ac}(a,b) \subseteq L_{ac}(a,b)$ and $L_{ad}(a,b) \subseteq L_{ac}(a,b)$.
  Thus we have $y^{0+}_{((a,b),(b,c))} + y^{0+}_{((a,b),(b,c),(c,d))} \leq 100 * f_{l_1} + 100 * f_{l_3}$.
- *For the arc* (a,b) *and the* (a,d) *equivalence class :* (a,d) OD pair $\in [a,d]^{\leq,\alpha}_{(a,b)}$ as $L_{ad}(a,b) \subseteq L_{ad}(a,b)$.
  Thus we have $y^{0+}_{((a,b),(b,c),(c,d))} \leq 100 * f_{l_3}$.
- *For the arc* (b,c) *and the* (a,c) *equivalence class :* both (a,c) and (a,d) OD pairs $\in [a,c]^{\leq,\alpha}_{(b,c)}$ as $L_{ac}(b,c) \subseteq L_{ac}(b,c)$ and $L_{ad}(b,c) \subseteq L_{ac}(b,c)$.
  Thus we have $y^{0+}_{((a,b),(b,c))} + y^{0+}_{((a,b),(b,c),(c,d))} \leq 100 * f_{l_1} + 100 * f_{l_3}$
- *For the arc* (b,c) *and the* (b,d) *equivalence class :* both (a,d) and (b,d) OD pairs $\in [b,d]^{\leq,\alpha}_{(b,c)}$ as $L_{ad}(b,c) \subseteq L_{bd}(b,c)$ and $L_{bd}(b,c) \subseteq L_{bd}(b,c)$.
  Thus we have $y^{0+}_{((a,b),(b,c),(c,d))} + y^{0+}_{((b,c),(c,d))} \leq 100 * freq\ l_2 + 100 * freq\ l_3$
- *For the arc* (b,c) *and the* (a,d) *equivalence class :* (a,d) OD pair $\in [a,d]^{\leq,\alpha}_{(b,c)}$ as $L_{ad}(b,c) \subseteq L_{ad}(b,c)$.
  Thus we have $y^{0+}_{(a,b),(b,c),(c,d)} \leq 100 * f_{l_3}$
- *For the arc* (c,d) *and the* (b,d) *equivalence class :* both (a,d) and (b,d) OD pairs $\in [b,d]^{\leq,\alpha}_{(c,d)}$ as $L_{ad}(c,d) \subseteq L_{bd}(c,d)$ and $L_{bd}(c,d) \subseteq L_{bd}(c,d)$.
  Thus we have $y^{0+}_{((a,b),(b,c)(c,d))} + y^{0+}_{((b,c),(c,d))} \leq 100 * f_{l_2} + 100 * f_{l_3}$
- *For the arc* (c,d) *and the* (a,d) *equivalence class :* (a,d) OD pair $\in [a,d]^{\leq,\alpha}_{(c,d)}$ as $L_{ad}(c,d) \subseteq L_{ad}(c,d)$
  Thus we have $y^{0+}_{((a,b),(b,c),(c,d))} \leq 100 * f_{l_3}$

$y^{0+}_{((a,b),(b,c))}$, $y^{0+}_{((a,b),(b,c),(c,d))}$, and $y^{0+}_{((b,c),(c,d))}$ have to be set to 100 to satisfy the associated passengers. These 7 DC constraints can be respected by setting a frequency of 2 to the line l_3 and a frequency of 0 to the lines l_1, l_2 and l_4.

Going back to model 3.3, constraint (3) sets the line frequencies according to the passengers on its respective arcs. As it is not respected on the arc (b,c), a solution consists to put a frequency of 1 to the line l_4. We finally get a frequency of 2 for the line l_3, a frequency of 1 for the line l_4 and a frequency of 0 for lines l_1 and l_2. Nevertheless, one of the three passenger demands has to transfer from line $l3$ to line l_4 even if these three passenger demands are considered as direct in the model. This is the limit of this DC constraint.

Appendix B. Dual problem

$$\max \sum_{(s,t)\in D} q_{st}\ \pi_{st} - \sum_{a\in A} s_a\ \eta_a + \psi_1 \sum_{(s,t)\in D} q_{st}\ \beta - o\ \Delta\ \theta - f_{max}\ n\ \omega \tag{B.1}$$

s.t.

$$\sum_{a\in l} \kappa\ \mu_a + \sum_{a\in l} \sum_{[s,t]_a \in \bar{D}(a)\ :\ l\in\mathcal{L}^{\alpha}_{st}(a)} \kappa\ \nu_{a,[s,t]_a} - \sum_{a\in l} \eta_a - r_l\ \theta - \omega \leq \lambda\ c\ d_l \qquad \forall l \in \mathcal{L} \tag{B.2}$$

$$\pi_{uv} - \sum_{a\in\mathcal{P}} \mu_a - \sum_{a\in\mathcal{P}} \sum_{[s,t]_a \in \bar{D}(a)\ :\ (u,v)\in[s,t]^{\leq,\alpha}_a} \nu_{a,[s,t]_a} + \psi_1 \leq (1-\lambda)\ r_p \qquad \forall (u,v)\in D, \forall p \in \mathcal{P}^{0+}_{uv} \tag{B.3}$$

$$\pi_{uv} - \sum_{a\in\mathcal{P}} \mu_a \leq (1-\lambda)\ r_p \qquad \forall (u,v)\in D,\ \forall p \in \mathcal{P}^{1}_{uv} \tag{B.4}$$

$$\pi_{st} \quad \text{unrestricted} \qquad \forall\ (s,t)\in D \tag{B.5}$$

$$\mu_a, \eta_a \geq 0 \qquad \forall\ a \in A \tag{B.6}$$

$$\nu_{a,[s,t]_a} \geq 0 \qquad \forall\ a\in A,\ \forall\ [s,t]_a \in D_a \tag{B.7}$$

$$\psi_1,\ \theta,\ \omega \geq 0 \tag{B.8}$$

Appendix C. Bus line labeling algorithm

As a remind, a bus line label ϵ is defined by the tuple (v, c, δ, N, R, S) where v denotes the last node visited, c the reduced cost of the label, δ the distance of label, N the set of nodes visited by the label and S the set of OD pairs satisfied by the label because they have a direct path on it and R the set of OD pairs not directly satisfied by the label whatever its future extension.

Examples C.1 and C.2 show how the **cost_Update** and **R_Update** functions work on a miniature network.

Example C.1. Example of cost_Line_Update execution

On Fig. C.7, suppose we are in presence of a label ϵ with $N_\epsilon = a, b, d, f$ and being extended to node g among the arc (f,g). All equivalence classes defined on the arc (f,g) are then analyzed to know if their origin is in the set N_ϵ and their destination is the node lastly added to extend the label

Algorithm 1 Function to generate lines from a source to a destination

function **GenerateLines**(src: source, dst: destination, g: graph, d_{max}: maximum Bus Line distance allowed, $duals$: dual values of the constraints of rLPP-SL')
L: Set of non-extended line labels
E: Set of extended line labels
$L := \{(src,\ 0,\ 0,\ \{src\},\ \{\},\ \{\})\}$
$E := \emptyset$
while $\sum_{\epsilon \in L : v_\epsilon \neq dst} |L| \neq 0$ **do**
  pick a label ϵ in L with $v_\epsilon \neq dst$
  $L \leftarrow L \setminus \epsilon$
  $E \leftarrow E \cup \epsilon$
  for all node $j \in V \setminus N_\epsilon$, linked to v_ϵ by an arc in the graph G **do**
    if $\delta_\epsilon + d_{(v_\epsilon, j)} + d_{(j, v_\epsilon)} \leq d_{max}$ **then**
      Extension of the label ϵ along arc (v_ϵ, j) with dual values to create label ϵ'
      for all equivalence class $[s,t]_a$ where $a = (v_\epsilon, j)$ **do**
        $cost_Line_Update([s,t]_a,\ \epsilon,\ \epsilon',\ j,\ duals)$
      end for
      for all OD pairs $(s,t) \notin R_{\epsilon'} \bigcup S_{\epsilon'}$ **do**
        $R_Update((s,t),\ \epsilon')$
      end for
      if ϵ' is not dominated by a label in $L_j \cup E_j$ **then**
        $L \leftarrow L \cup \epsilon'$
        Removing all labels dominated by ϵ' from $L_j \cup E_j$
      end if
    end if
  end for
end while
return all labels ϵ in L_{dst} with $\bar{c}_\epsilon < 0$
end function

Algorithm 2 Function that extends the reduced cost of the label according to the equivalence classes that have a direct path by using the new label ϵ'

function **cost_Line_Update**(Equivalence class $[s,t]_a$, label ϵ, label ϵ', j: node used to extend label ϵ , $duals$: duals values of the constraints of rLPP-SL)
if (the OD pair $(s,t) \notin R_{\epsilon'} \bigcup S_{\epsilon'}$) $\bigcap$ (s belongs to N_ϵ and t is equal to j, or the opposite) **then**
  if the duration of the path from s to t on the created line respects the maximum-spt-deviation **then**
    Update of the cost $\bar{c}_{\epsilon'}$ on all the arcs relying the nodes of the label ϵ' between u and w: $\bar{c}_{\epsilon'} \leftarrow \bar{c}_{\epsilon'} - \sum_{a \in A_{N_{\epsilon'},(u,w)}} (\nu_{a,(u,w)} + \nu_{\bar{a},(u,w)})\ \kappa.$
    Add all the OD pairs being dc-equivalent to (s,t) wrt a to the $S_{\epsilon'}$ set
  end if
end if

Algorithm 3 Function that determines the OD pairs that can no longer have a direct path by using the new label ϵ'

function **R_Update**(OD pair (s,t), label ϵ')
if the duration of the path from s to t on the created line exceeds the maximum-spt-deviation **then**
  Add the OD pair (s,t) to the $R_{\epsilon'}$ set
end if

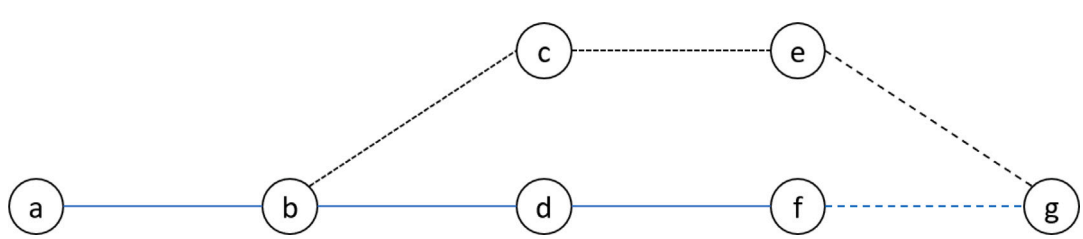


Fig. C.7. A small network represented with a graph composed of 7 nodes.

ϵ or the opposite. Suppose the existence of a (b, g) equivalence class with respect to the arc (f, g). $[b, g]_a$ respects the conditions described above as $b \in N_\epsilon$ and g is the lastly added node. We check if the duration of the path from b to g on the created line respects the maximum-spt-deviation, if it does, then it means that the passengers of the (b, g) OD pair have a direct passenger path by using the line associated to this label. The cost $c_{\epsilon'}$ is then updated according to the dual values of constraints (11) on the arcs of the created line between b and g. Finally, the OD pairs that are dc-equivalent to (b, g) with respect to the arc (f, g), and thus belong to the equivalence class $[s, t]_a$, are added to the set $S_{\epsilon'}$.

Example C.2. Example of R_Update execution

On Fig. C.7, suppose a label ϵ with $N_\epsilon = a, b, d, f$, being extended to the node g among the arc (f, g). The new label created is called ϵ'. Suppose there exists a (b, e) OD pair and that this OD pair is not present in $R_{\epsilon'}$ nor in $S_{\epsilon'}$. Let us call $r_{(\epsilon', bg)}$ the duration of the path from b to g on the created line and $r^{min}_{s,t}$ the minimum in-vehicle travel time to go from s to t on the graph considered. If the ratio between $r_{(\epsilon', bg)} + r^{min}_{g,e}$ and t^{min}_{be} exceeds the maximum-spt-deviation enforced by the service-level parameter α, the (b, e) OD pair can no longer have a direct passenger path on the label, regardless of future extensions to other nodes. This OD pair is then added to the set $R_{\epsilon'}$.

Appendix D. Passenger path labeling algorithm

As a reminder, a passenger path label is defined by the tuple (v, c, τ, N) where v represents the last node visited by the label, c the reduced cost of the label, τ the time of the label and N the set of nodes visited by the label.

Algorithm 4 Function for generating attractive paths for a passenger from its source node to its destination node

function **GeneratePath**(src: source, dst: destination, G: graph, $duals$: duals values of the constraints of rLPP-SL, $pathDesired$: type of path desired to be generated, $\mathcal{L}$: line pool)
L: Set of non-extended path labels
E: Set of extended path labels
if pathDesired = *rdc* **then**
  $L := \{(src,\ -\pi_{(src,dst)} - \psi_1,\ 0,\ \{src\},\ \{\})\}$
else if pathDesired = *indirect* **then**
  $L := \{(src,\ -\pi_{(src,dst)},\ 0,\ \{src\},\ \{\})\}$
end if
$E := \emptyset$
while $\sum_{\epsilon \in L : v_\epsilon \neq dst} |L| \neq 0$ **do**
  pick a label ϵ in L with $v_\epsilon \neq dst$
  $L \leftarrow L \setminus \epsilon$
  $E \leftarrow E \cup \epsilon$
  for all nodes j of the graph G such that $j \notin N_\epsilon$ and the arc (v_ϵ, j) is an arc of G **do**
    $\epsilon' = path_Extension(\epsilon,\ src,\ dst,\ G,\ duals,\ pathDesired,\ \mathcal{L})$
  end for
  if ϵ' is not dominated by a label in $L_j \cup E_j$ **then**
    $L \leftarrow L \cup \epsilon'$
    Removing all labels dominated by ϵ' from $L_j \cup E_j$
  end if
end while
return all labels ϵ in L_{dst} with $\bar{c}_\epsilon < 0$
end function

Algorithm 5 Function to extend the label of a passenger path

function **path_Extension**(ϵ : label to extend, src: source, dst: destination, G: graph, $duals$: duals values of the constraints of rLPP-SL', $pathDesired$: type of path desired to be generated, $\mathcal{L}$: line pool)
if $\tau_\epsilon + r_{v_\epsilon, j} + r^{min}_{j,dst} \leq \alpha\ r^{min}_{src,dst}$ **then**
  if (pathDesired = *rdc* **and** $\exists\ l \in \mathcal{L}$ to go from src to j wrt the maximum-spt-deviation) **or** (pathDesired = *indirect* **and** $\exists\ l \in \mathcal{L}$ circulating on the arc (v_ϵ, j)) **then**
    Extension of the label ϵ along arc (v_ϵ, j) to create label ϵ'
    if pathDesired = *rdc* **then**
      $cost_rdcPath_Update(\epsilon', (v_\epsilon, j),\ src,\ dst,\ duals)$
    end if
  end if
end if
return label ϵ'

Example D.1 shows how the **cost_Path_Update** function works on a miniature network.

Example D.1. Example of cost_rdcPath_Update execution

On Fig. C.7, we consider a label visiting nodes (a, b, d, f) and being extended to node g. Whether the passenger path we try to generate is a rdc or an indirect-passenger-path, it must be checked if the time of the label newly extended to the node g respects the service-level α, meaning has a in-vehicle travel time smaller or equal than α the minimum in-vehicle travel time from a to g noted $r^{min}_{a,g}$.

Algorithm 6 Function that extends the reduced cost of the label in case of a rdc-passenger path researched

```
function cost_rdcPath_Update(label ε', arc a extending the label ε, src: source, dst: destination, duals: duals values of the constraints of rLPP-SL')
for all equivalence classes [s,t]_a do
    if (src, dst) ∈ [s,t]_a^{≤,α} then
        c̄_ε' ← c̄_ε' + v_{a,[s,t]_a}
    end if
end for
```

Table E.14
Columns generated for the different variants of the service-level parameter α.

α	β	CG			CE		
		# bus lines	# rdc paths	# indirect paths	# bus lines	# rdc paths	# indirect paths
2	74%	547	3378	49 642	559	8811	121 570
1.75	74%	618	3434	31 650	639	7874	65 210
1.5	74%	577	3125	1884	932	7237	33 459

Table E.15
Columns generated for the variation of the service-level parameter β.

α	β	CG			CE		
		# bus lines	# rdc paths	# indirect paths	# bus lines	# rdc paths	# indirect paths
1.75	68%	573	3031	38 197	1095	9173	72 788
1.75	74%	618	3434	31 650	639	7874	65 210
1.75	80%	466	3506	35 168	491	7423	69 940

Furthermore, in the case of a rdc-passenger-path generation from a to e, it must be checked if there is at least one line in the bus lines pool $\mathcal{L}$ that allows going from a to e with the arc (f, g) in it and by respecting the maximum-spt-deviation enforced by the service-level parameter α. In the case of an indirect-passenger-path generation from a to e, it must checked if there it least one line in the bus lines pool $\mathcal{L}$ that drives on the newly added arc (f, g).

Appendix E. CGEP insights for the case study

Impact of variations of the maximum-spt-deviation (α) service-level parameter for a fixed minimum-rdc-percentage (β) service-level

Table E.14 highlights the results obtained for variations of the maximum-spt-deviation (α) service-level parameter on the execution of CGEP. The first three columns show the number of bus lines, rdc- passenger paths, and indirect passenger paths generated during the column generation step (CG) and the next three columns show the same information for the column enumeration step (CE).

It can be observed that by making the maximum-spt-deviation (α) service-level parameter more restrictive, i.e. decreasing it from 1.75 to 1.5, results in a decrease in the number of generated passenger paths. This is because the passenger path labeling algorithm filters out paths based on their duration. As a result, the number of rdc-passenger paths decreases from 7874 to 7237 (−8.09%) and the number of indirect passenger paths decreases from 65210 to 33459 (−48.69%). On the other hand, the number of bus lines increases from 639 to 932 (+45.85%).

Conversely, making the maximum-spt-deviation (α) service-level parameter less restrictive, i.e. increasing it from 1.75 to 2, leads to an increase in the number of generated passenger paths. The number of rdc-passenger paths increases from 7874 to 8811 (+11.90%), the number of indirect passenger paths increases from 56210 to 121570 (+116.28%), and the number of bus lines decreases from 639 to 559 (−12.52%).

It can be noted that reducing coefficient α implies finding a feasible set of passenger paths and lines to initialize the CGEP. This was done by modifying the existing lines manually for $\alpha = 1.5$. We did not reduce this parameter further for this reason.

Impact of variations of the minimum-rdc-percentage (β) service-level parameter for a fixed maximum-spt-deviation (α) service-level

In Table E.15, we observe that, as the minimum-rdc-percentage (β) service-level parameter is decreased from 74% to 68%, the number of generated bus lines increases from 639 to 1095, a growth of 71.36%. Additionally, the number of rdc-passenger paths generated also increases from 7874 to 9173, a rise of 16.50%. On the other hand, when the minimum-rdc-percentage (β) service-level parameter is increased from 74% to 80%, the number of bus lines generated decreases from 639 to 491, a decrease of 23.16%. Additionally, the number of rdc-passenger paths generated decreases from 7874 to 7423, a decline of 5.73%.

It is also noted that there is no significant change in the number of indirect passenger paths generated, regardless of whether the minimum-rdc-percentage (β) service-level parameter is decreased or increased.

Table E.16
Detail of the columns generated for Poitiers instance by varying the average bus load parameter.

	CG Step			CE Step		
K	# bus lines	# rdc paths	# indirect paths	# bus lines	# rdc paths	# indirect paths
14.2	550	3294	32 238	577	7937	66 236
18.9	618	3434	31 650	639	7874	65 210
23.6	515	3480	38 776	527	7303	66 742

Table F.17
Numerotation of the stations of the Dutch instance.

Node number	Code	Name
1	Ah	Arnhem
2	Apd	Apeldoorn
3	Asd	Amsterdam CS
4	Asdz	Amsterdam Zuid WTC
5	Asn	Assen
6	Bd	Breda
7	Ehv	Eindhoven
8	Gn	Groningen
9	Gv	Den Haag HS
10	Gvc	Den Haag CS
11	Hgl	Hengelo
12	Hr	Heerenveen
13	Lls	Lelystad Centrum
14	Lw	Leeuwarden
15	Mt	Maastricht
16	Odzg	Oldenzaal Grens
17	Rsdg	Rosendaal Grens
18	Rtd	Rotterdam CS
19	Shl	Schiphol
20	Std	Sittard
21	Ut	Utrecht CS
22	Zl	Zwolle
23	Zvg	Zevenaar Grens

Influence of average bus load

In Table E.16, we can observe that the variation of the average bus load parameter does not have a great impact on the number of bus lines and passenger paths generated. Hence, we find that reducing the average bus load from 18.9 to 14.2 has a small impact on the number of generated bus lines from 639 to 577 (−9.70%) but has almost no impact on the number of generated rdc and indirect passenger paths. On the other hand, we find that increasing the average bus load from 18.9 to 23.6 leads to a small decrease too in the number of generated lines from 639 to 527 (−17.53%) and has almost no impact too on the number of generated rdc and indirect passenger paths.

Appendix F. Literature instances

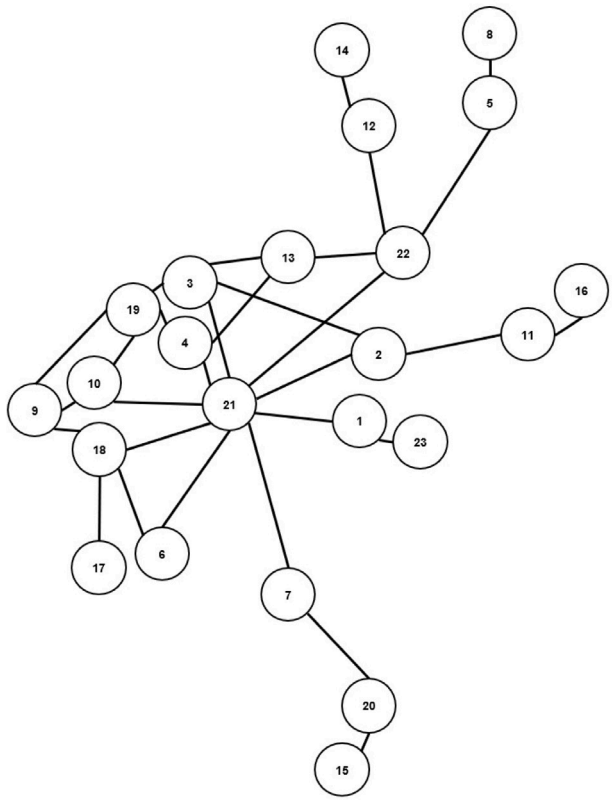
F.1. Dutch instance

The Dutch instance details can be found at https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_lop.html.
Table F.17 describes the names of the main station associated to the graph nodes.
Similar to Borndörfer and Karbstein (2015), we consider that each edge has a length of one kilometer (see Fig. F.8).

F.2. Sioux-Falls instance

The Sioux-Falls network is described in Bar-Gera (2021). It is based on a graph composed of 24 nodes and 38 arcs and the associated OD matrix is composed of 172 non-zero OD pairs (see Fig. F.9).

For this instance, Fig. F.10, obtained from a private communication with M. Karbstein, provides the data used in Borndörfer and Karbstein (2015).

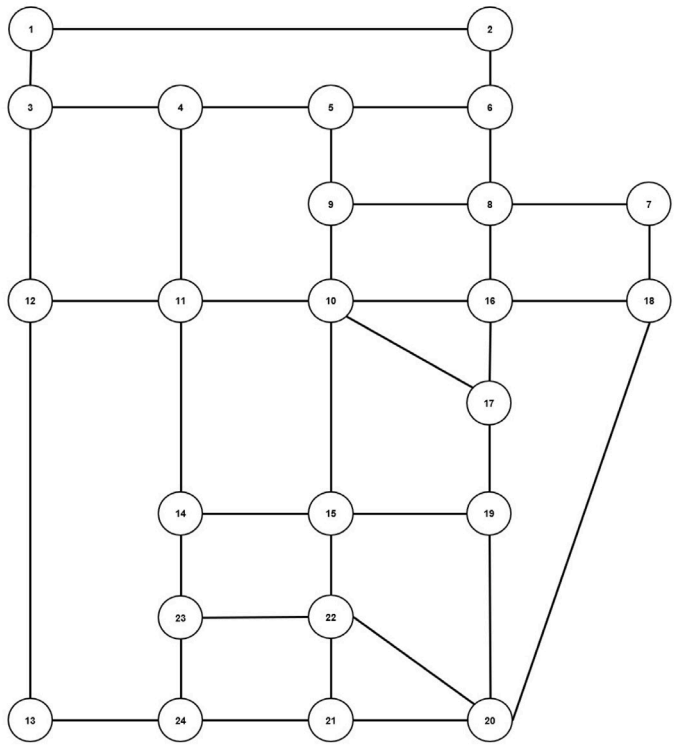


(a) The Dutch graph

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	726	721	16	136	101	0	109	741	0	11	8	13	0	0	320	180	60	0	4244	56	602
2	0	0	917	76	0	57	62	0	21	202	468	0	0	0	5	71	47	143	32	10	1160	0	83
3	726	917	0	0	235	154	437	205	730	2540	287	115	2783	90	37	22	2258	1756	6469	155	4919	400	489
4	721	76	0	0	58	138	542	88	461	207	24	16	819	21	149	0	6	369	1456	203	5826	171	155
5	16	0	235	58	0	42	33	1720	13	117	0	0	152	0	14	0	48	125	32	28	502	854	19
6	136	57	154	138	42	0	950	34	531	228	39	16	6	14	79	5	329	1829	7	157	1165	79	14
7	101	62	437	542	33	950	0	28	35	335	28	11	8	9	404	3	75	569	99	936	3109	47	11
8	0	0	205	88	1720	34	28	0	12	73	0	0	200	0	13	0	33	75	48	29	331	720	14
9	109	21	730	461	13	531	35	12	0	785	20	10	29	6	8	0	890	4586	1339	22	225	33	0
10	741	202	2540	207	117	228	335	73	785	0	81	48	31	26	41	7	3	2829	1503	104	3138	163	229
11	0	468	287	24	0	39	28	0	20	81	0	0	0	0	12	75	24	52	11	20	422	0	0
12	11	0	115	16	0	16	11	0	10	48	0	0	77	478	4	0	10	58	20	8	209	511	19
13	8	0	2783	819	152	6	8	200	29	31	0	77	0	77	1	0	15	46	103	2	89	390	0
14	13	0	90	21	0	14	9	0	6	26	0	478	77	0	4	0	7	36	20	9	145	380	14
15	0	5	37	149	14	79	404	13	8	41	12	4	1	4	0	0	22	73	29	863	359	21	0
16	0	71	22	0	0	5	3	0	0	7	75	0	0	0	0	0	0	11	0	21	0	0	0
17	320	47	2258	6	48	329	75	33	890	3	24	10	15	7	22	0	0	1077	164	19	325	100	0
18	180	143	1756	369	125	1829	569	75	4586	2829	52	58	46	36	73	11	1077	0	509	179	2260	182	157
19	60	32	6469	1456	32	7	99	48	1339	1503	11	20	103	20	29	0	164	509	0	44	278	64	0
20	0	10	155	203	28	157	936	29	22	104	20	8	2	9	863	21	19	179	44	0	720	46	0
21	4244	1160	4919	5826	502	1165	3109	331	225	3138	422	209	89	145	359	0	325	2260	278	720	0	1112	996
22	56	0	400	171	854	79	47	720	33	163	0	511	390	380	21	0	100	182	64	46	1112	0	32
23	602	83	489	155	19	14	11	14	0	229	0	19	0	14	0	0	0	157	0	0	996	32	0

(b) The symmetrical OD matrix associated to the Dutch graph

Fig. F.8. Dutch network and associated OD matrix.



(a) The Sioux-Falls graph

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	100	100	500	200	300	500	800	500	1300	500	200	500	300	500	500	400	100	300	300	100	400	300	100
2	100	0	100	200	100	400	200	400	200	600	200	100	300	100	100	400	200	0	100	100	0	100	0	0
3	100	100	0	200	100	300	100	200	100	300	300	200	100	100	100	200	100	0	0	0	0	100	100	0
4	500	200	200	0	500	400	400	700	700	1200	1400	600	600	500	500	800	500	100	200	300	200	400	500	200
5	200	100	100	500	0	200	200	500	800	1000	500	200	200	100	200	500	200	0	100	100	100	200	100	0
6	300	400	300	400	200	0	400	800	400	800	400	200	200	100	200	900	500	100	200	300	100	200	100	100
7	500	200	100	400	200	400	0	1000	600	1900	500	700	400	200	500	1400	1000	200	400	500	200	500	200	100
8	800	400	200	700	500	800	1000	0	800	1600	800	600	600	400	600	2200	1400	300	700	900	400	500	300	200
9	500	200	100	700	800	400	600	800	0	2800	1400	600	600	600	900	1400	900	200	400	600	300	700	500	200
10	1300	600	300	1200	1000	800	1900	1600	2800	0	4000	2000	1900	2100	4000	4400	3900	700	1800	2500	1200	2600	1800	800
11	500	200	300	1500	500	400	500	800	1400	3900	0	1400	1000	1600	1400	1400	1000	100	400	600	400	1100	1300	600
12	200	100	200	600	200	200	700	600	600	2000	1400	0	1300	700	700	700	600	200	300	400	300	700	700	500
13	500	300	100	600	200	200	400	600	600	1900	1000	1300	0	600	700	600	500	100	300	600	600	1300	800	800
14	300	100	100	500	100	100	200	400	600	2100	1600	700	600	0	1300	700	700	100	300	500	400	1200	1100	400
15	500	100	100	500	200	200	500	600	1000	4000	1400	700	700	1300	0	1200	1500	200	800	1100	800	2600	1000	400
16	500	400	200	800	500	900	1400	2200	1400	4400	1400	700	600	700	1200	0	2800	500	1300	1600	600	1200	500	300
17	400	200	100	500	200	500	1000	1400	900	3900	1000	600	500	700	1500	2800	0	600	1700	1700	600	1700	600	300
18	100	0	0	100	0	100	200	300	200	700	200	200	100	100	200	500	600	0	300	400	100	300	100	0
19	300	100	0	200	100	200	400	700	400	1800	400	300	300	300	800	1300	1700	300	0	1200	400	1200	300	100
20	300	100	0	300	100	300	500	900	600	2500	600	500	600	500	1100	1600	1700	400	1200	0	1200	2400	700	400
21	100	0	0	200	100	100	200	400	300	1200	400	300	600	400	800	600	600	100	400	1200	0	1800	700	500
22	400	100	100	400	200	200	500	500	700	2600	1100	700	1300	1200	2600	1200	1700	300	1200	2400	1800	0	2100	1100
23	300	0	100	500	100	100	200	300	500	1800	1300	700	800	1100	1000	500	600	100	300	700	700	2100	0	700
24	100	0	0	200	0	100	100	200	200	800	600	500	700	400	400	300	300	0	100	400	500	1100	700	0

(b) The symmetrical OD matrix associated to the Sioux-Falls graph

Fig. F.9. Sioux-Falls network and associated OD matrix.

Orig	Dest	Distance	Time
1	2	0.01	6
1	3	0.006	4
2	1	0.01	6
2	6	0.008	5
3	1	0.006	4
3	4	0.006	4
3	12	0.006	4
4	3	0.006	4
4	5	0.003	2
4	11	0.01	6
5	4	0.003	2
5	6	0.006	4
5	9	0.008	5
6	2	0.008	5
6	5	0.006	4
6	8	0.003	2
7	8	0.005	3
7	18	0.003	2
8	6	0.003	2
8	7	0.005	3
8	9	0.016	10
8	16	0.008	5
9	5	0.008	5
9	8	0.016	10
9	10	0.005	3
10	9	0.005	3
10	11	0.008	5
10	15	0.01	6
10	16	0.006	4
10	17	0.013	8
11	4	0.01	6
11	10	0.008	5
11	12	0.01	6
11	14	0.006	4
12	3	0.006	4
12	11	0.01	6
12	13	0.005	3
13	12	0.005	3

Orig	Dest	Distance	Time
13	24	0.006	4
14	11	0.006	4
14	15	0.008	5
14	23	0.006	4
15	10	0.01	6
15	14	0.008	5
15	19	0.005	3
15	22	0.005	3
16	8	0.008	5
16	10	0.006	4
16	17	0.003	2
16	18	0.005	3
17	10	0.013	8
17	16	0.003	2
17	19	0.003	2
18	7	0.003	2
18	16	0.005	3
18	20	0.006	4
19	15	0.005	3
19	17	0.003	2
19	20	0.006	4
20	18	0.006	4
20	19	0.006	4
20	21	0.01	6
20	22	0.008	5
21	20	0.01	6
21	22	0.003	2
21	24	0.005	3
22	15	0.005	3
22	20	0.008	5
22	21	0.003	2
22	23	0.006	4
23	14	0.006	4
23	22	0.006	4
23	24	0.003	2
24	13	0.006	4
24	21	0.005	3
24	23	0.003	2

Fig. F.10. Arcs characteristics for the Sioux Falls instance.

References

Achterberg, Tobias, 2009. SCIP: solving constraint integer programs. Math. Program. Comput. 1, 1–41. http://dx.doi.org/10.1007/s12532-008-0001-1.

Baaj, Mohamed Hadi, Mahmassani, Hani S., 1990. TRUST: A LISP program for the analysis of transit route configurations. Transp. Res. Rec. 11.

Baaj, Mohamed Hadi, Mahmassani, Hani S., 1995. Hybrid route generation heuristic algorithm for the design of transit networks. Transp. Res. Part C: Emerg. Technol. 3, 31–50. http://dx.doi.org/10.1016/0968-090X(94)00011-S.

Baldacci, Roberto, Christofides, Nicos, Mingozzi, Aristide, 2008. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. Math. Program. 115, 351–385. http://dx.doi.org/10.1007/s10107-007-0178-5.

Bar-Gera, Hillel, 2021. Transportation network test problems. URL https://github.com/bstabler/TransportationNetworks.

Bertsimas, Dimitris, Ng, Yee Sian, Yan, Julia, 2021. Data-driven transit network design at scale. Oper. Res. 69 (4), 1118–1133. http://dx.doi.org/10.1287/opre.2020.2057.

Borndörfer, Ralf, Grötschel, Martin, Pfetsch, Marc E., 2007. A column-generation approach to line planning in public transport. Transp. Sci. 41, 123–132. http://dx.doi.org/10.1287/trsc.1060.0161.

Borndörfer, Ralf, Karbstein, Marika, 2012. A direct connection approach to integrated line planning and passenger routing. In: 12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems. http://dx.doi.org/10.4230/OASIcs.ATMOS.2012.47.

Borndörfer, Ralf, Karbstein, Marika, 2015. Metric inequalities for routings on direct connections with application to line planning. Discrete Optim. 18, 56–73. http://dx.doi.org/10.1016/j.disopt.2015.07.004, URL https://www.sciencedirect.com/science/article/pii/S1572528615000432.

Bussieck, Michael R., Kreuzer, Peter, Zimmermann, Uwe T., 1997. Optimal lines for railway systems. European J. Oper. Res. 96, 54–63. http://dx.doi.org/10.1016/0377-2217(95)00367-3.

Camporeale, Rosalia, Caggiani, Leonardo, Ottomanelli, Michele, 2019. Modeling horizontal and vertical equity in the public transport design problem: A case study. Transp. Res. Part A: Policy Pr. 125, 184–206. http://dx.doi.org/10.1016/j.tra.2018.04.006.

Cancela, Héctor, Mauttone, Antonio, Urquhart, María E., 2015. Mathematical programming formulations for transit network design. Transp. Res. Part B: Methodol. 77, 17–37. http://dx.doi.org/10.1016/j.trb.2015.03.006.

Ceder, Avishai, Wilson, Nigel H.M., 1986. Bus network design. Transp. Res. Part B: Methodol. 20, 331–344. http://dx.doi.org/10.1016/0191-2615(86)90047-0.

Durán-Micco, Javier, van Kooten Niekerk, Marcel, Vansteenwegen, Pieter, 2022. Designing bus line plans for realistic cases - the Utrecht case study. Expert Syst. Appl. 187, http://dx.doi.org/10.1016/j.eswa.2021.115918.

Durán-Micco, Javier, Vansteenwegen, Pieter, 2021. A survey on the transit network design and frequency setting problem. Public Transp. http://dx.doi.org/10.1007/s12469-021-00284-y.

Duran-Micco, Javier, Vermeir, Evert, Vansteenwegen, Pieter, 2020. Considering emissions in the transit network design and frequency setting problem with a heterogeneous fleet. European J. Oper. Res. 282, 580–592. http://dx.doi.org/10.1016/j.ejor.2019.09.050.

EN-13816, 2002. Transportation. Logistics and services. Public passenger transport. Service quality definition, targeting and measurement. Eur. Stand. URL https://www.en-standard.eu/din-en-13816.

Faivre d'Arcier, Bruno, 2014. Measuring the performance of urban public transport in relation to public policy objectives. Res. Transp. Econ. 48, 67–76. http://dx.doi.org/10.1016/j.retrec.2014.09.033.

Gatt, Hector, Freche, Jean-Marie, Lehuédé, Fabien, Laurent, Arnaud, 2022. A bilevel model for the frequency setting problem. In: ATMOS 2022 : International Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems. Potsdam, Germany, http://dx.doi.org/10.4230/OASIcs.ATMOS.2022.5.

Gattermann, Philine, Harbering, Jonas, Schöbel, Anita, 2017. Line pool generation. Public Transp. 9, 7–32. http://dx.doi.org/10.1007/s12469-016-0127-x.

Glize, Estèle, Roberti, Roberto, Jozefowiez, Nicolas, Ngueveu, Sandra Ulrich, 2020. Exact methods for mono-objective and Bi-objective multi-vehicle covering tour problems. European J. Oper. Res. 283, 812–824. http://dx.doi.org/10.1016/j.ejor.2019.11.045.

Guihaire, Valérie, Hao, Jin-Kao, 2008. Transit network design and scheduling: A global review. Transp. Res. Part A: Policy Pr. 42, 1251–1273. http://dx.doi.org/10.1016/j.tra.2008.03.011.

Gutiérrez-Jarpa, Gabriel, Laporte, Gilbert, Marianov, Vladimir, Moccia, Luigi, 2017. Multi-objective rapid transit network design with modal competition: The case of concepción, Chile. Comput. Oper. Res. 78, 27–43. http://dx.doi.org/10.1016/j.cor.2016.08.009.

Heinrich, Irene, Schiewe, Philine, Seebach, Constantin, 2022. Line planning in public transport: Bypassing line pool generation. http://dx.doi.org/10.48550/arXiv.2205.09580, arXiv preprint arXiv:2205.09580.

Jin, Jian Gang, Teo, Kwong Meng, Odoni, Amedeo R., 2016. Optimizing bus bridging services in response to disruptions of urban transit rail networks. Transp. Sci. 50, 790–804. http://dx.doi.org/10.1287/trsc.2014.0577.

Karbstein, Marika, 2013. Line Planning and Connectivity (Ph.D. thesis). Mathematik und Naturwissenschaften der Technischen Universität Berlin.

Liang, Jinpeng, Wu, Jianjun, Gao, Ziyou, Sun, Huijun, Yang, Xin, Lo, Hong K., 2019. Bus transit network design with uncertainties on the basis of a metro network: A two-step model framework. Transp. Res. Part B: Methodol. 126, 115–138. http://dx.doi.org/10.1016/j.trb.2019.05.011.

Magnanti, Thomas L., Wong, Richard T., 1984. Network design and transportation planning: Models and algorithms. Transp. Sci. 18 (1), 1–55. http://dx.doi.org/10.1287/trsc.18.1.1.

Mandl, Christoph E., 1980. Evaluation and optimization of urban public transportation networks. European J. Oper. Res. 5, 396–404. http://dx.doi.org/10.1016/0377-2217(80)90126-5.

Marín, Ángel G., 2007. An extension to rapid transit network design problem. TOP 15, 231–241. http://dx.doi.org/10.1007/s11750-007-0017-0.

Marín, Ángel G., Jaramillo, Patricia, 2009. Urban rapid transit network design: accelerated Benders decomposition. Ann. Oper. Res. 169, 35–53. http://dx.doi.org/10.1007/s10479-008-0388-0.

Marques, Guillaume, 2020. Two-Echelon Vehicle Routing Problems in City Logistics: Approaches Based on Exact Methods of Mathematical Optimization (Ph.D. thesis). Université de Bordeaux.

Mumford, Christine L., 2013. New heuristic and evolutionary operators for the multi-objective urban transit routing problem. In: 2013 IEEE Congress on Evolutionary Computation. IEEE, pp. 939–946. http://dx.doi.org/10.1109/CEC.2013.6557668.

Schöbel, Anita, 2012. Line planning in public transportation: models and methods. OR Spectrum 34 (3), 491–510. http://dx.doi.org/10.1007/s00291-011-0251-6.

Schöbel, Anita, Scholl, Susanne, 2006. Line planning with minimal traveling time. In: 5th Workshop on Algorithmic Methods and Models for Optimization of Railways. ATMOS'05, http://dx.doi.org/10.4230/OASIcs.ATMOS.2005.660.

Suman, Hemant K., Bolia, Nomesh B., 2019. Improvement in direct bus services through route planning. Transp. Policy 81, 263–274. http://dx.doi.org/10.1016/j.tranpol.2019.07.001.

Suman, Hemant Kumar, Bolia, Nomesh B., Tiwari, Geetam, 2018. Perception of potential bus users and impact of feasible interventions to improve quality of bus services in Delhi. Case Stud. Transp. Policy 6, 591–602. http://dx.doi.org/10.1016/j.cstp.2018.07.009.

Yan, Julia, 2020. From Data to Decisions in Urban Transit and Logistics (Ph.D. thesis). Massachusetts Institute of Technology.

Yu, Bin, Yang, Zhong-Zhen, Jin, Peng-Huan, Wu, Shan-Hua, Yao, Bao-Zhen, 2012. Transit route network design-maximizing direct and transfer demand density. Transp. Res. Part C: Emerg. Technol. 22, 58–75. http://dx.doi.org/10.1016/j.trc.2011.12.003.

Zhao, Fang, Ubaka, Ike, 2004. Transit network optimization - minimizing transfers and optimizing route directness. J. Public Transp. 7, 63–82. http://dx.doi.org/10.5038/2375-0901.7.1.4.