Les Cahiers du GERAD ISSN: 0711–2440

Column generation and local search for the profit-oriented hub-line location problem with elastic demands

B. Cobeña, C. Contardo

G–2024–58

Septembre 2024
Revised: February 2025

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2024
– Bibliothèque et Archives Canada, 2024

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2024
– Library and Archives Canada, 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

Column generation and local search for the profit-oriented hub-line location problem with elastic demands

Brenda Cobeña [a, b]

Claudio Contardo [a, b]

[a] *Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, Montréal (Qc), Canada, H3G 1M8*

[b] *GERAD & CIRRELT, Montréal (Qc), Canada, H3T 1J4*

brenda.cobena@mail.concordia.ca
claudio.contardo@concordia.ca

Septembre 2024
Revised: February 2025
Les Cahiers du GERAD
G–2024–58

Abstract : Population growth and city sprawl have been driving increasing amounts of traffic congestion in multiple major cities worldwide. In this scenario, developing efficient public transportation networks becomes critical to ensure adequate mobility. Hub network location models address the problems of designing public transit networks to model —and to optimize— passenger mobility. More specifically, hub-line location problems (HLLP) play an essential role in the design of rapid transit corridors and subway lines. In this work we address the profit-oriented hub-line location problem (ED-HLLP) for which we introduce a column generation method to solve the linear relaxation of a mixed-integer model and matheuristic that combines column generation and local search. The proposed methodologies lead to the calculation of primal and dual bounds. We assess the performance of the proposed methods on some classic datasets from the HLLP literature. Furthermore, we conduct a more realistic study on a problem instance representing the metropolitan area of Montreal, Canada. Finally, we conduct a sensitivity analysis to assess the major attributes driving our results, both from an algorithmic point of view as well as from a planning perspective.

Keywords : Hub location, urban mobility hubs, gravity models, column generation, local search

Acknowledgements: We thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for its financial support through Grant no. 2020–06311. We would like to extend our gratitude to Dr. Ivan Contreras for their constructive remarks in the early stages of this project.

1 Introduction

Population growth, city sprawl and the urbanization of rural areas have been driving incremental levels of traffic in major cities worldwide —mainly due to an increase in the acquisition and use of private vehicles— significantly impacting the mobility of passengers on their daily trips (Nations, 2019; Aydin et al., 2022). In this context, hub location problems (HLPs) play an important role in the design of transportation networks. Specifically, the hub-line location problem (HLLP) addresses the problem of designing a corridor (such as a subway line, or a rapid transit bus corridor) in transportation planning to improve passenger mobility in their daily trips (Contreras and O'Kelly, 2019).

HLLP enables the integration of multiple modes of transportation, allowing for the allocation of passengers to multiple sub-systems, which in turn translates into direct interactions between non-hub nodes and multiple assignments for each origin and destination (OD) pair to more than one hub. A hub node may be a a metro, train, or tram station where two or more transportation modes interact. A non-hub node may be a bus, taxi, car/bike share station or urban district. The flows are ridership or users travelling between the multiple OD pairs in one or more modes such as train, metro, or subway (Martins de Sá et al., 2015).

In the classical HLLP, the demand is assumed to be inelastic and independent of the design of the resulting hub-line system. Recently, Cobeña et al. (2023) introduced the *profit-oriented hub line location problem with elastic demand* (ED-HLLP). They use gravity models to incorporate demand elasticity into an optimization model. ED-HLLP aims to maximize revenue that in turn depends of the time savings obtained when using the hub-line system with respect to the existing network. It is only natural to try to capture the fact that increased time savings will result in higher demands in the new system. Hence, considering the elasticity of demand in ED-HLLP makes the model more realistic.

The HLLP and the ED-HLLP give raise to difficult optimization models. In Cobeña et al. (2023) the authors model the ED-HLLP as a mixed-integer nonlinear optimization problem. A commercial off-the-shelf solver is shown to be able to scale and solve very small instances of the nonlinear model. To better cope with the nonlinear nature of the problem, the same authors reformulate the problem as a mixed-integer linear problem (MILP) using a very large number of variables, one for every possible OD-path in the network, including or not nodes in the new hub-line system. Via a smart enumeration mechanism, the authors can solve to proven optimality larger problems when compared to solving the nonlinear models. The combinatorial nature of their enumeration algorithm, however, only pushes but does not get rid of the combinatorial explosion. Small problems only (with up to 25 total nodes) remain tractable for their method.

We address the problem of solving the ED-HLLP for larger problem sizes. Since the number of feasible OD-paths grows exponentially with the problem size, an enumeration of all possible paths may quickly become prohibitively inefficient. In this article, we investigate the development of more scalable methods to address the ED-HLLP. A novel variable enumeration mechanism based on the column generation (CG) paradigm is used to generate promising paths dynamically. A local search procedure is then applied to generate high-quality feasible solutions quickly.

This paper presents several significant contributions to the field of urban planning and optimization. The key contributions are as follows:

1. We introduce a novel column generation method that uses dynamic programming for efficient path selection for the ED-HLLP. Its efficiency depends on the ability to identify promising OD-paths, incorporating extension and dominance rules to avoid the generation of non-promising ones. The strategy used in the path extension step is a label-setting algorithm. The proposed method allows us to compute primal and dual bounds efficiently.
2. We introduce a hybrid matheuristic that combines column generation and local search (LS) to enrich the pool of promising paths for the ED-HLLP. The proposed method allows us to compute primal bounds in short computing times.

3. We provide a thorough comparison between the path enumeration method of Cobeña et al. (2023) and the new methods introduced in this manuscript using the classical CAB dataset. In particular, our methods are compared against the sequential and parallel implementations of the path enumeration mechanism of Cobeña et al. (2023).
4. We conduct a case study using data from the metropolitan area of Montreal to show the applicability and relevance of the proposed heuristic on a real-world context. This study offers insights for city planners, urban planners and public transport managers on the design of urban mobility systems.

The remainder of this article is structured as follows. In Section 2 we provide a the literature review on hub-line location and related problems. In Section 3 we provide a formal definition of the problem. In Section 4 we describe in full detail the CG method to address the solution of the linear relaxation of ED-HLLP and the computation of primal and dual solutions. Section 5 describe the proposed matheuristic algorithm that combines the column generation and local search. Section 6 is dedicated to presenting the results of extensive computational experiments designed to evaluate the performance of the proposed methodologies. It includes a detailed examination of the application of the proposed method using the CAB dataset and a case study focused on the metropolitan area of Montreal, Canada. This case study demonstrates the application of the new method to tackle real instances. It quantifies the benefits of implementing mobility hubs, including the percentage reduction in travel time facilitated by the hub-line system and an assessment of spatial coverage. The paper culminates in Section 7, where we conclude our study, highlighting the significant contributions and potential avenues for future research.

2 Literature review

The first mathematical model for Hub location problems (HLPs) is introduced by O'kelly (1986). HLPs are pivotal in designing hub-and-spoke networks by locating a set of hub facilities and selecting a set of links to route flows between OD pairs. One main assumption of a classical hub location problem is that hubs are fully interconnected and that direct connections between non-hub nodes are not allowed; however, for applications in public transport planning, the hub-level network is an incomplete hub network (Alumur and Kara, 2008).

Nickel et al. (2001) made a notable contribution by introducing HLPs in urban public transportation networks. Their models introduced the concept of HLPs, where the hubs are not fully interconnected, and direct connections between pairs of non-hub nodes are allowed. Afterwards, Gelareh and Nickel (2011) proposed hub location problems in urban transportation and liner shipping network design. In this problem, the complete interconnection assumption is relaxed, but no specific topology is required; multiple allocations and direct connections between non-hub nodes are allowed.

Zhong et al. (2018) design a multi-level hub and spoke (H&S) network to determine the location of integration of rural and public transport hubs; Another concrete example of applying HLPs in public transportation planning is the Hub Line Location Problems (HLLPs). It fits in the multiple-allocation HLP with incomplete hub-level networks in which direct connections between pairs of non-hub nodes are allowed. Particularly, HLLP is applied in designing rapid transit systems and highway networks to enhance users' travel times.

HLLP was first introduced by Martins de Sá et al. (2015). The authors introduce mathematical models to address the problem of locating p special facilities known as hubs and $p-1$ hub edges to form a path network. The HLLP incorporates a service-based objective that minimizes the total travel time between OD pairs. The flows represent passengers traveling between OD pairs who wish to minimize their commute time. Users will use the hub-line whenever time savings are perceived, otherwise, they will use a direct link. The models introduced in that article capture other aspects relevant to model travel times, such as the access and exit times incurred when using the hub network. To solve the resulting model, the authors propose an exact algorithm based on Benders decomposition. They provide computational evidence of their method by considering two standard benchmark instances from the hub location literature: the data set of the U.S. Civil Aeronautics Board (CAB) (see, O'kelly, 1987) and the Australian Post data set (see, Ernst and Krishnamoorthy, 1996).

In the previous works in HLPs applied in public transportation, the demand is assumed to be static and independent from the location of the hub facility; however, this is not a reasonable assumption. According to Alumur et al. (2021), the nature of the demand and how it affects the resulting hub network is a key aspect of better modelling HLPs. The authors emphasize the importance of incorporating the elasticity of demand to price or quality of service, including the location of hubs and the opportunities to serve only some of the demand, especially within a profit maximization model.

Location problems with demand elasticities have already been studied in the past. In competitive facility location problems, a decision maker seeks to minimize lost demand or maximize the market share captured considering elastic demand. Solution algorithms and extensions of these can be found Marianov et al. (1999), Eiselt and Marianov (2009), Marianov et al. (2005), and Marianov et al. (2008). In network design problems, Aboolian et al. (2012) introduced the profit-maximizing service network design problem and Zetina et al. (2019) introduced profit-oriented multi-commodity network design, both incorporating elastic demands.

Furthermore, the concept of demand elasticity has been integrated into gravity-type models used in transportation planning models (see, De Dios Ortúzar and Willumsen, 1991; Tamin and Willumsen, 1989). Traffic assignment problems (TAPs) were among the earliest to incorporate elastic demands into transportation planning problems. The TAP is a sub-class of transit network design problems in which high-level decisions such as adding road capacity, deciding vehicle passing frequency (mostly for public transit), or vehicle capacities must be determined (Newell, 1979). The TAP with elastic demands induces a bi-level optimization structure that is very hard to address computationally. Because of this, the majority of solution algorithms for these problems have been heuristics (Cipriani et al., 2012).

These studies demonstrate the importance and impact of accounting for elastic demands in strategic hub network design problems in public transportation. To the best of our knowledge, ED-HLLP is the only problem that addresses the design of the hub line system using gravity models to incorporate demand elasticity within an optimization framework. In the ED-HLLP the authors Cobeña et al. (2023) present two mixed-integer nonlinear programming formulations (MINLP) using arc-based variables to model OD paths and capture the nonlinear components. Furthermore, given how difficult these nonlinear formulations are to optimize using state-of-the-art MINLP solvers, the authors also propose mixed-integer linear programming formulations (MILP) using path-based variables to model OD paths. They introduce an *a priori* enumeration algorithm to generate all candidate OD paths to used in the MILP formulations. In their computational study, they report that the MINLP formulations require very high solving times, often much higher than enumerating the paths and solving the MILP models.

Significant efforts have been made to develop algorithms to achieve superior solutions for a range of HLPs. The use of column generation approaches to address HLPs remains rather limited (see, Farahani et al., 2013; Alumur and Kara, 2008; Alumur et al., 2021; Contreras and O'Kelly, 2019). In Rothenbächer et al. (2016), the authors propose an exact branch-and-price-and-cut algorithm for the service network design and hub location problem. They consider a path-based formulation for the problem where the subproblems resort to shortest paths with resource constraints (SPPRC), and solved by means of a labeling algorithm.

3 Problem definition and mathematical formulation

Let us consider the linear model of Cobeña et al. (2023) for the ED-HLLP. Let $\widetilde{G} = (N, A)$ be a directed graph for the ED-HLLP model, derived from the undirected graph $G = (N, E)$ where N is the set of nodes and E the set of edges $e := [k, m]$ with $k < m$. Here $A = \{(k, m) \cup (m, k) : e = [k, m] \in E\}$ is the set of arcs induced by E. Furthermore, let C be the collection of OD pairs whose demands must be routed either through a hub line or directly from origin to destination. Each OD pair will be referred to as commodity $c \in C$ and its origin and destination nodes denoted o_c and d_c, respectively.

For each commodity $c \in C$, $t_{o_c d_c} \geq 0$ denotes the optimal (minimum) travel time required to travel from o_c to d_c in the absence of the hub line. Without loss of generality, $t_{o_c d_c}$ also incorporates any average transfer time required when changing modes of transportation from o_c to d_c. When a hub arc is located between hub nodes $(k, m) \in A$, the travel time between k and m is computed as $\alpha_{km} t_{ij}$, where α_{km} $(0 \leq \alpha_{ij} \leq 1)$ is a reduction factor that models the use of a faster transport technology to connect o_c and d_c. Also, the access and exit times to the hub line through node $i \in N$ respectively are incorporated, denoted as $\tilde{t}_i^a \geq 0$ and $\tilde{t}_i^e \geq 0$ respectively.

The demand of a commodity $c \in C$, denoted by w_c, is modeled with a gravity-like attraction that depends on the attraction between o_c and d_c, as well as the travel time. It satisfies the equation:

$$w_c = \frac{P_{o_c} P_{d_c}}{(t_c)^r}, \tag{1}$$

where P_{o_c} and P_{d_c} are weights associated to the populations of o_c and d_c, respectively. Moreover, we denote $R_c \geq 0$ the revenue for each unit of time reduction for $c \in C$ when using the hub line system.

Because of the triangle inequality property of the travel times $t_{o_c d_c}$, there exists a solution of the HLLP that routes the demands w_c either with a direct connection between OD or with a path containing at most two access arcs and at least two hub nodes and one hub arc. Thus, once a commodity leaves the hub-line, it cannot access the hub-line again.

Let $\mathcal{P}_c$ denote the set of all possible paths using a hub-line of p hubs with an associated travel time smaller than or equal to $t_{o_c d_c}$. Each path $\pi \in \mathcal{P}_c$ can be expressed as: $\pi = [o_c, h_1, \ldots, h_k, d_c]$, where h_m, for $m = 1, \ldots k$ with $k \leq p$, denote the hub nodes that the path π traverses in its correct order. In particular, h_1 and h_k represent the access-to and exit-from nodes in the hub-line, respectively. We can recognize four types of paths $\pi \in \mathcal{P}_c$:

(ODH$_c$)-paths. Corresponding to paths in which all the nodes are hubs. In particular, o_c and d_c must be hubs.

(DH$_c$)-paths. Corresponding to paths whose origin node (o_c) is not a hub node, but the destination node (d_c) is.

(OH$_c$)-paths. Correspnding to paths whose destination node (d_c) is not a hub node, but the origin node (o_c) is.

(ODNH$_c$)-paths. Corresponding to paths in which neither o_c nor d_c are hub nodes.

Then, the travel time for routing commodity $c \in C$ via a path $\pi \in \mathcal{P}_c$, and denoted $\tau_{\pi c}$ is:

$$\tau_{\pi c} = t_{o_c h_1} + \tilde{t}_{h_1}^a + \sum_{m=1}^{k-1} \alpha t_{h_m h_{m+1}} + \tilde{t}_{h_k}^e + t_{h_k d_c}.$$

Therefore, if all the links of a path $\pi \in \mathcal{P}_c$ are known and fixed, its associated travel time $\tau_{\pi c}$ is also known and, consequently, associated with a profit of:

$$g_{\pi c} = R_c \frac{P_{o_c} P_{d_c}}{(\tau_{\pi c})^r} (t_{o_c d_c} - \tau_{\pi c}).$$

The previous two observations regarding the travel times and profits of the paths give raise to the following MILP formulation of the ED-HLLP.

We use binary hub-line variables $v_{\pi c}$, $\pi \in \mathcal{P}_c, c \in C$, equal to 1 if and only if commodity c is delivered using path π. Also, the following binary variables are used: z_k, $k \in N$, equal to 1 if and only if a hub is located at node k and $y_{km}, (k,m) \in A$ equal to 1 if and only if a hub arc is located between hubs k and m, enabling flows to be routed in both directions. Finally, for every node $k \in N$ we consider integer variables l_k representing the order in which nodes are traversed in the hub-line. Besides the variable set, we also make use of parameters $h_e^{\pi c}$ equal to 1 if and only if path $\pi \in \mathcal{P}_c$ contains the arc (k,m) or (m,k) defined by edge $e = [k,m] \in E$. The ED-HLLP is stated as the following MILP:

$$
\begin{aligned}
(P) \quad \max \quad & \sum_{c \in C} \sum_{\pi \in \mathcal{P}_c} g_{\pi c} v_{\pi c} && \\
& \sum_{i \in N} z_k = p && && (2) \\
& \sum_{k \in N} \sum_{\substack{m \in N \\ (k,m) \in A}} y_{km} = p - 1, && && (3) \\
& \sum_{\substack{m \in N \\ (k,m) \in A}} y_{km} + \sum_{\substack{m \in N \\ (m,k) \in A}} y_{mk} \leq 2 z_k, && k \in N. && (4) \\
& \sum_{\substack{l \in N \\ (k,l) \in A}} y_{kl} \geq z_k + z_m - 1, && k \in N \setminus \{n\}, m = k+1, \ldots, n. && (5) \\
& l_k - l_m + n y_{km} \leq n - 1 && (k,m) \in A, && (6) \\
& \sum_{\pi \in \mathcal{P}_c} h_{[k,m]}^{\pi c} v_{\pi c} \leq y_{km} + y_{mk}, && [k,m] \in E, c \in C, && (7) \\
& \sum_{\pi \in \mathcal{P}_c} v_{\pi c} \leq 1, && c \in C, && (8) \\
& \sum_{\pi \in (\mathrm{ODH}_c)\text{-paths}} v_{\pi c} + \sum_{\pi \in (\mathrm{DH}_c)\text{-paths}} v_{\pi c} \leq z_{d_c}, && c \in C. && (9) \\
& \sum_{\pi \in (\mathrm{ODH}_c)\text{-paths}} v_{\pi c} + \sum_{\pi \in (\mathrm{OH}_c)\text{-paths}} v_{\pi c} \leq z_{o_c}, && c \in C. && (10) \\
& y_{km} \in \{0,1\}, && (k,m) \in A. && (11) \\
& z_k \in \{0,1\}, && k \in N. && (12) \\
& v_{\pi c} \in \{0,1\}, && \pi \in \mathcal{P}_c, c \in C. && (13)
\end{aligned}
$$

The objective function maximizes the total time savings obtained from using the hub-line system. Constraints (2) and (3) strictly define the number of hubs and inter-hub links to be installed. The series of constraints from (4) through (6) are crafted to uphold the design of the hub-line. Specifically, constraints (4) limit each hub to a maximum of two links to other hubs, while constraints (5) dictate that there must be at least one outgoing arc from each hub node on the path, excluding the hub node with the largest index to mitigate the appearance of symmetric solutions. The constraints (6), also referred to as Miller-Tucker-Zemlin (MTZ) constraints, act as sub-tour elimination constraints (SECs), ensuring uninterrupted connectivity of the hub-line. By enforcing these constraints, we seek to establish an oriented path where the highest indexed hub does not have an outgoing arc, hence removing feasible solution symmetries.

Furthermore, constraints (8) restrict that each commodity is transported using the hub line, and constraints (7) enforce the exclusive use of paths where hub arcs are opened. Constraints (9) and (10) delineate between different path types. In particular, constraints (9) apply when all candidate paths of types (ODH_c) and (DH_c) are identified for a specific commodity c, thus requiring d_c to be a hub. Conversely, constraints (10) apply when paths of types (ODH_c) and (OH_c) confirm that o_c is a hub for a specific commodity c. The decision variables' domain is defined by constraints (11)–(13).

The number of feasible hub line paths can be huge, so an enumeration of all possible paths is not practicable. Instead, we use column generation to dynamically generate promising paths, as described next.

4 Column generation

We now address the problem of solving the linear relaxation of problem ED-HLLP (see Section 3) via column generation. We describe next the different components of this method, namely the restricted master problem (RMP), the pricing sub-problem (SP), and a labeling algorithm for the solution of SP.

4.1 Restricted master problem

The restricted master problem (RMP) is obtained from the linear relaxation of the problem (P) by restricting it to a subset $\overline{\mathcal{P}_c}$ of paths for each commodity $c \in C$, which leads to a problem that we denote RMP($\overline{\mathcal{P}}$). Upon solving the linear programming relaxation of RMP($\overline{\mathcal{P}}$), its dual variables are employed to construct a subsequent problem, the *pricing subproblem*. The aim of this problem is to identify new paths with the potential to enhance the value of the objective function for the LP-relaxation, thereby bringing it closer to the optimal value of the master problem MP = RMP($\mathcal{P}$), which contains exponentially many variables.

Let, $\overline{\mathcal{P}_c} \subseteq \mathcal{P}_c$, the current subset of feasible paths under consideration. RMP($\overline{\mathcal{P}}$) consists in the following linear program:

$$(RMP) \quad \max \sum_{c \in C} \sum_{\pi \in \overline{\mathcal{P}_c}} g_{\pi c} v_{\pi c}$$

$$\text{s.t.} \quad (2) - (6),$$

$$[\beta] \sum_{\pi \in \overline{\mathcal{P}_c}} h^{\pi c}_{[k,m]} v_{\pi c} \leq y_{km} + y_{mk}, \qquad [k,m] \in E, c \in C. \tag{14}$$

$$[\sigma] \sum_{\pi \in \overline{\mathcal{P}_c}} v_{\pi c} \leq 1, \qquad c \in C. \tag{15}$$

$$[\rho] \sum_{\pi \in (\mathrm{ODH}_c)\text{-paths}} v_{\pi c} + \sum_{\pi \in (\mathrm{DH}_c)\text{-paths}} v_{\pi c} \leq z_{d_c}, \qquad c \in C. \tag{16}$$

$$[\omega] \sum_{\pi \in (\mathrm{ODH}_c)\text{-paths}} v_{\pi c} + \sum_{\pi \in (\mathrm{OH}_c)\text{-paths}} v_{\pi c} \leq z_{o_c}, \qquad c \in C. \tag{17}$$

$$y_{km} \geq 0, \qquad (k,m) \in A. \tag{18}$$

$$z_k \geq 0, \qquad k \in N. \tag{19}$$

$$v_{\pi c} \geq 0, \qquad c \in C, \pi \in \overline{\mathcal{P}_c}. \tag{20}$$

Here β to ω are dual values from constraints (14) to (17).

4.2 The pricing sub-problem

In this section, we present the pricing sub-problem used to generate positive reduced-cost columns when solving the problem defined in Section 4.1. We first present the exact pricing algorithm. Then, we present a heuristic implementation of the algorithm to speed up the search for columns, especially useful on the more challenging problem instances.

4.2.1 Exact pricing sub-problem

Because of the different types of (ODH_c, DH_c, OH_c and ODNH_c)-paths, the pricing sub-problem must be performed for each $c \in C$ and type of path π. An initial set of columns must be provided to initialize the CG algorithm. In Appendix A we describe the heuristic applied to obtain an initial set of feasible paths. The pricing problem searches the variable's hub line paths with positive reduced cost to add them to the current set of columns of the RMP. The reduced cost for a path π_c can be computed as follows:

$$\bar{p}(v_{\pi_c}) = \begin{cases} g_{\pi c} - \sigma_c - \rho_c - \sum_{e \in E} \beta_e^c & \text{if } c \in C, \pi \in \text{(ODH}_c \text{ and DH}_c\text{)-paths} \\ g_{\pi c} - \sigma_c - \omega_c - \sum_{e \in E} \beta_e^c & \text{if } c \in C, \pi \in \text{(ODH}_c \text{ and OH}_c\text{)-paths} \\ g_{\pi c} - \sigma_c - \sum_{e \in E} \beta_e^c & \text{if } c \in C, \pi \in \text{(ODNH}_c\text{)-paths} \end{cases}$$

Finding a new path of positive reduced cost can be performed separately for every $c \in C$ and type of path via the solution of a shortest path problem with resource constraints (SPPRC). We address the solution of the SPPRC using dynamic programming. We refer the reader to Irnich and Desaulniers (2005), Ropke and Cordeau (2009), Costa et al. (2019) for overviews of constrained shortest-path problems and appropriate solution techniques. The dynamic programming algorithm can be explored according to different search strategies, and the order in which the labels are extended may be very important for the effectiveness of the overall algorithm.

In label-setting algorithms, labels become permanent as soon as they are deemed as not dominated by labels created previously. Once a label is set, it cannot change. On the other hand, label-correcting algorithms allow labels to be updated multiple times as new, potentially shorter paths are discovered. Label-setting algorithms are generally more efficient because they avoid re-processing labels multiple times. Label-correcting algorithms can be less efficient due to their iterative updating of labels. Label-setting is often preferred for problems such as the shortest path in network routing (where edge weights are non-negative). We refer the reader to Zhan and Noon (2000), Desrochers and Soumis (1988) for a more in-depth discussion on this subject.

In this work, we use a label-setting algorithm to solve the shortest path problem. The pricing sub-problem is performed independently for each $c \in C$ and the dual values in RMP are then used to update the profit of each commodity $c \in C$. The following section describes the labeling algorithm for identifying potential optimal paths and removing labels through the use of dominance rules.

Labeling algorithm. Labeling algorithms are employed to solve the SPPRC that commonly arise in CG approaches for routing problems. The objective is to find the shortest path for each origin-destination (OD) pair in the graph $G=(N,A)$. The paths must satisfy conditions on resources used between OD, for instance time and the number of hub arcs represent examples of resources consumed along the path.

Labeling algorithms build partial paths in the graph G; paths are built from the origin (o_c) to the destination (d_c) for each commodity c. Each path starts with an initial label that holds the information about the resource consumption, and the labels are updated as the forward partial paths are extended toward the destination of the commodity c.

Given the different types of paths between OD (see Section 3), which implies different varieties of arcs between those nodes, a path cannot be represented merely as a sequence of nodes. Therefore, a *path* is defined as $\pi = [o_c, h_1, \ldots, h_k, d_c]$, where h_1 to h_k with $k \leq p$ denote the hubs that the path π traverses in its correct order.

Irnich and Desaulniers, 2005 provide an overview of techniques for addressing the SPPRC and their potential solutions; however, in the SPPRC proposed by them, there are no constraints on the structure of the paths. Thus, all paths are feasible. This work introduces a path structure constraint to obtain the shortest paths using the hub line of p hubs and $p-1$ hub arcs while ensuring a travel time smaller than the direct times obtained by the labeling algorithm.

A *label* L is a tuple that associates the set of information for a partial path starting at the origin o_c and ending at the destination d_c: the final node η of the label, τ - the cumulative time at the node, the cumulative β dual value, Π - the set of the nodes visited, H_A - the accumulated number of hubs arcs. Our resources are τ, $\bar{\rho}$, Π and H_A. The notation $\tau(L)$ is used to refer to the cumulate time in label L, and a similar notation is used for the rest of the resources (e.g. $\eta(L)$, $\beta(L)$). Thus, the label of each forward partial path is denoted by $L = (\eta, \tau, \beta, \Pi, H_A)$.

The dynamic programming recursion starts from a label $L_0^i = (\{o_c\}, 0, 0, \emptyset, 0)$. It is based on a label extension rule to create paths π and a dominance rule to discard nonpromising labels.

Label extension. The *label extension* process is performed until the destination d_c is reached. Paths are extended, and resource usage is accumulated during path construction. Then, for each resource H_A, type of path, and commodity c, if extension along the arc (η, j) is feasible, a new label L' is created at node j. The information in label L' is set as follows:

$$\eta(L') = j \tag{21}$$

$$\tau(L') = \begin{cases} \tau(L) + (\alpha * \tau_{\eta(L),j}) & \text{if } (\eta(L), j) \in A \\ \tau(L) + \tau_{\eta(L),j} & \text{otherwise} \end{cases} \tag{22}$$

$$\beta(L') = \begin{cases} \beta(L) + \beta_{\eta(L),j} & \text{if } (\eta(L), j) \in A \\ \beta(L) & \text{otherwise} \end{cases} \tag{23}$$

$$\Pi(L') = \begin{cases} \Pi(L) \cup \{j\} & \text{if j is a node visited}, j \in N \\ \Pi(L) & \text{otherwise} \end{cases} \tag{24}$$

$$H_A(L') = \begin{cases} H_A(L) + \{j\} & 1 \text{ if j is selected as a hub node}, j \in N \\ H_A(L) & \text{otherwise .} \end{cases} \tag{25}$$

Equations (21)–(23) set the current node, the time, and the cumulative β dual value associated of the constraint whose hub edges are opened of the new label, respectively. Equation (24) updates the set of visited nodes, this ensures that the paths are simple, meaning they do not contain a cycle. Equation (25) updates the total of open hub edges.

In the path-searching process, to efficiently select the next adjacent nodes to explore, we employ the resource of the number of hub arcs (H_A) as a referential dimension in the partial paths process. To preserve feasibility, paths are checked for travel time when extending a label L along an arc (η, j), the extension is valid only if path times are less than direct times ($t_{o_c d_c}$).

Dominance criterion. In addition to the infeasible labels rejected by the *extending rule*, unpromising labels are also eliminated by the *dominance rule*. Let L_1 and L_2 be two labels sharing the same terminal node η. We say that L_1 dominates L_2 if:

$$\begin{aligned} &\eta(L_1) = \eta(L_2), \quad \tau(L_1) \le \tau(L_2), \\ &\beta(L_1) \ge \beta(L_2), \quad H_A(L_1) \le H_A(L_2), \\ &\Pi(L_1) \subseteq \Pi(L_2). \end{aligned} \tag{26}$$

The dominance rule is correct in the sense that it allows to discard the label L_2 when every feasible extension of it also is feasible for L_1, leading to a larger value of $\overline{p}$.

Acceleration of the dominance processes. To reduce computational time in the dominance processes, we store the label list as an ordered list, which means that the set of the non-dominated labels is sorted by descending order of cumulative β dual values. Furthermore, the rule of dominance is applied to labels sharing the same destination node and having less or equal hub arcs. Then, the time complexity of the dominance process is linear. Once the destination of the commodity d_c is reached, dominance and extension rules are not applied any further, and the labels are stored.

The pricing sub-problem is solved sequentially. Algorithm 1 summarizes the label-setting applied to get new columns (paths) for each commodity c and type of path. We omit the outer loop and concentrate on the labeling algorithm. There are two main restrictions on every path this procedure must ensure: Firstly, the travel time must not exceed the current travel time, and secondly, paths must not exceed the number of hub edges (p-1 hubs) until the destination (d_c) of the commodity is reached.

In general terms, Algorithm 1 returns new paths with a positively reduced cost for each OD pair and path type. Finally, to solve the ED-HLLP, we solve the RMP as an integer program to obtain a heuristic solution.

Algorithm 1: Dynamic programming algorithm for the solution of the pricing sub-problem ($L_0^i = (\{o_c\}, 0, 0, \emptyset, 0)$)

Input: $L_0^i, \tilde{t}^a_{o_c}, \tilde{t}^e_{d_c}, t_{o_c d_c}$
Output: $\mathcal{L}$ // (ODH$_c$, DH$_c$, OH$_c$ and ODNH$_c$)-paths for $c \in C$
Initialize $Q \leftarrow L_0^i, \mathcal{L} \leftarrow \emptyset$
repeat
  Take label L from Q and set $Q \leftarrow Q \backslash \{L\}$
  for *all* $r \in \mathcal{R}$ *s.t.* $\tau(L) + \tilde{t}^a_{o_c} + \tilde{t}^e_{d_c} < t_{o_c d_c}$ **do**
    for *all* $j \in N \backslash \Pi(L)$ **do**
      Extend L to j to create a new label L' // 4.2.1
      Apply dominance rule L' // 4.2.1
      if L' *has not been discarded and* $\eta(L') \neq dc$ **then**
        $Q \leftarrow Q \cup \{L'\}$
      else
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{L'\}$ // set label by desc order
      end
    end
  end
until $Q = \emptyset$
return $\{L \in \mathcal{L} \,|\, \bar{p} > 0\}$

4.2.2 Heuristic pricing algorithm

The exact solution of the pricing sub-problem can be complex in the presence of a large set of paths. Therefore, before executing the exact pricing method described before, we consider a truncated labeling algorithm as a heuristic. In this truncated labeling method, only the 5 nearest neighbors of every node are considered for the extension step.

4.3 Computing primal and dual bounds

The CG described before naturally leads to the calculation of a dual bound. To compute a primal bound, we proceed by enforcing integrality on the path variables $v_{\pi c}$. This is a known technique in the scientific literature (see for instance Ceselli et al., 2009; Joncour et al., 2010; Yuan et al., 2021). A feasible solution to the resulting MILP provides a primal bound of the problem, however not necessarily an optimal solution, which could only achieved if the CG was repeated on every node of the branching tree, thus leading to a branch-and-price method (Barnhart et al., 1998).

5 A hybrid matheuristic

This section introduces a hybrid matheuristic that combines CG using the labeling algorithm described in the previous section with local search (LS). Local search is the most widely heuristic used in applications to large problems (Ribeiro et al., 2002; Gendreau et al., 2010). It has been extensively applied to multiple combinatorial optimization problems including hub-location problems (see for instance, Contreras et al., 2011; de Sá et al., 2015). These methods are particularly effective in exploring neighborhood structures to escape local optima. This section details the implementation of LS for the ED-HLLP, highlighting its operators and their integration with CG.

5.1 Initialization

Like all local search procedures, the proposed method requires an initial feasible solution to the ED-HLLP. We consider a simple strategy that involves using the columns generated by the CG applied to problem (P). The resulting RMP resulting from solving the linear relaxation of the problem is then solved upon adding the integrality constraints. Note that this procedure in general cannot guarantee the return of a feasible solution to the problem. In the ED-HLLP, however, commodities may always choose to use the current network (without the hub-line), and therefore such solution always exists.

5.2 Generation of new paths and hub-line configurations

The proposed local search relies on the fact that thanks to constraints (2)–(3), every solution to the problem contains exactly p hub nodes and $p-1$ hub-arcs. We have therefore designed two operators that maintain this structure at all times, as described below:

- `Swap` **operator**: this operator exchanges the positions of two hubs within the hub line network. Specifically, starting from the initial hub-line solution, each pair of hubs is swapped. Then, for each Swap, all possible combinations of paths are recalculated for every commodity c and path type (ODH, OH, DH, and ODNH). In Fig. 1, we illustrate this operator by means of an example for $n = 3$ and $p = 6$. Fig. 1-(a) shows the current hub-line configuration solution that traverses nodes 1, 2, and 3 and the different feasible paths using the hub nodes. Fig. 1 from (b) to (d) show the three different swaps of two nodes in this configuration and, after each Swap, the multiple paths that are added to Ω. Note that only the paths that result in time savings are added to Ω (see Algorithm 2).

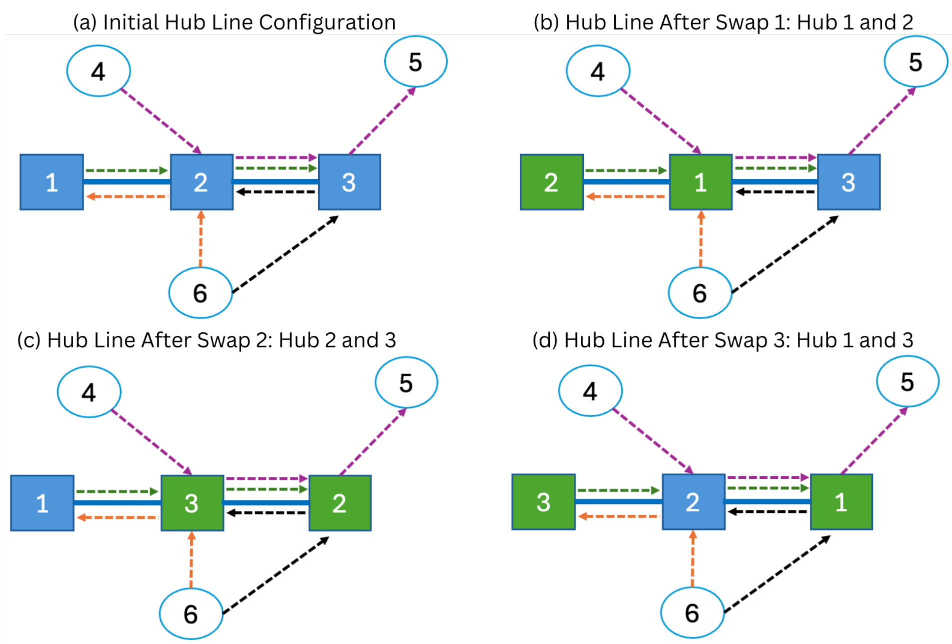


Figure 1: Example of the `Swap` **operator with** $n = 6, p = 3$

- `Replace` **operator**: this operator deactivates an existing hub and activates a non-hub node as a hub, at the same position in the current hub-line. Fig. 2 illustrates this operator with an example for $n = 3$ and $p = 6$. Paths are then generated and evaluated under the same criteria as for the `Swap` operator, making sure that only paths that provide time savings are generated (see Algorithm 3).

Algorithm 2: `Swap` **operator**

Input: Initial hub-line $h = (h_1, \dots, h_p)$
Output: List of feasible paths Ω, Final hub line $\eta = (\eta_1, \dots, \eta_p)$
Initialize: Paths $\Omega \leftarrow \emptyset$, hub-line $\eta \leftarrow h$
while *true* **do**
  Let $\tau \leftarrow \eta$
  foreach $1 \leq i < j \leq p$ **do**
    Evaluate swapping $\tau_i \leftrightarrow \tau_j$, let τ' be the resulting hub-line
    Generate valid paths in τ'
    foreach *path* π **do**
      Calculate $time(\pi)$
      if $time(\pi) < t_{o_c d_c}$ **then**
        Add π to Ω
      end
    end
    if $profit(\tau') > profit(\tau)$ **then**
      $\tau \leftarrow \tau'$
    end
  end
  if $profit(\tau) > profit(\eta)$ **then**
    $\eta \leftarrow \tau$
  else
    break
  end
end
return Ω, η

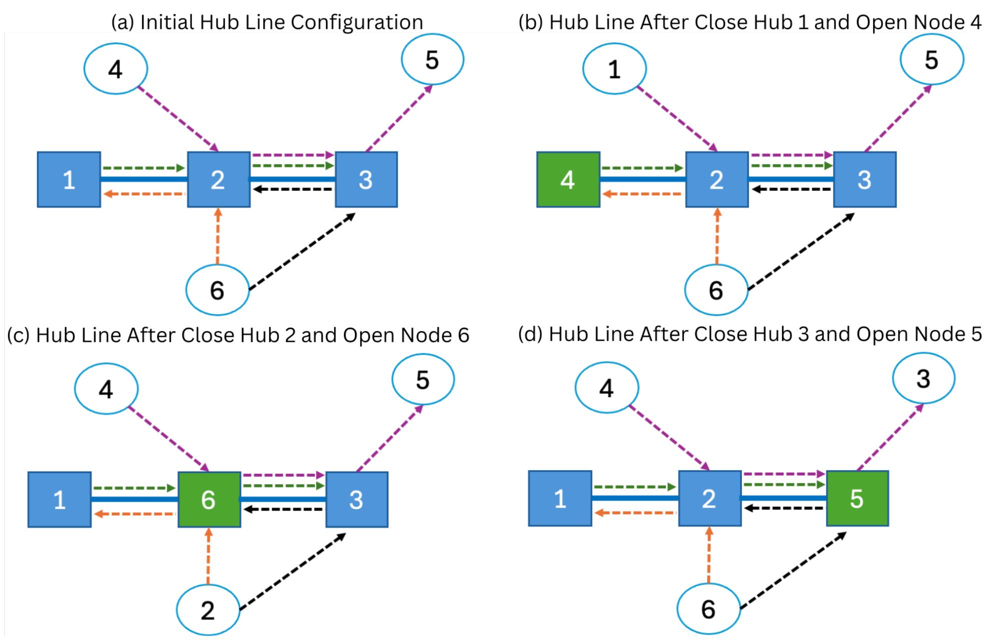


Figure 2: Example of the `Replace` **operator with** $n = 6, p = 3$

The proposed local search procedure iterates over the two operators defined above, achieving an appropriate balance between exploration and exploitation. The `Swap` operator is computationally efficient because it only changes two hub nodes at a time, limiting the combinatorial explosion of paths. The `Replace` operator allows for a broader exploration of the solution space compared to `Swap`, as it introduces previously non-hub nodes into the hub configuration. The LS effectively balances computational effort by prioritizing paths with shorter travel times. In the two pseudocodes, the profit associated with a line configuration corresponds to evaluating the objective in problem (P) for appropriate choices of paths π.

Algorithm 3: `Replace` **operator**

```
Input: Initial hub-line h = (h_1, ..., h_p)
Output: List of feasible paths Ω, Final hub line η = (η_1, ..., η_p)
Initialize: Paths Ω ← ∅, hub-line η ← h, non-hub nodes w = (w_1, ..., w_{n−p})
while true do
    Let τ ← η, ω ← w
    foreach 1 ≤ i ≤ p, 1 ≤ j ≤ n − p do
        Evaluate replacing τ_i by ω_j, let τ' be the resulting hub-line
        Generate valid paths in τ'
        foreach path π do
            Calculate time(π)
            if time(π) < t_{o_c d_c} then
                Add π to Ω
            end
        end
        if profit(τ') > profit(τ) then
            Swap ω_j ⟷ τ_i
        end
    end
    if profit(τ) > profit(η) then
        η ← τ
        w ← ω
    else
        break
    end
end
return Ω, η
```

An important attribute of every local search procedure is its capability to evaluate operators as efficiently as possible (in $O(1)$ ideally), both in terms of *feasibility* of the resulting configuration, as in terms of *economic benefit*. For the feasibility requirement, note that the two proposed operators always produce feasible hub-line configurations. To compute the economic benefit, one can compute by simple inspection all possible paths connecting o_c and d_c for a given commodity c in the resulting hub-line, simply by considering all pairs of entrance-exit nodes in the hub-line, and by discarding those with non-positive profits.

5.3 Post-optimization

Our local search concludes when no operator can be applied to construct a configuration achieving a better total profit. Note that our local search not only allows for the finding of a better solution than that constructed by the initialization procedure, but also constructs paths π, many of which may have failed to be enumerated by the CG. We re-solve the RMP as a MILP powered with the new paths generated using our local search. Algorithm 4 summarizes the proposed hybrid matheuristic. In lines 1 and 2, we obtain the initial feasible paths using the heuristic pricing algorithm and solve the RMP as MILP to obtain the initial lower bound, respectively. In line 3, we execute the LS procedure using the initial hub line obtained in the previous step to add new columns. Finally, we resolve the RMP with the paths generated using the CG method and LS.

Algorithm 4: Hybrid matheuristic for the ED-HLLP

```
1 Solve the LP relaxation of the restricted master by CG;                                // see Section 4
2 Solve the RMP with integrality constraints and obtain a primal solution;               // see Section 5.1
3 Execute the local search method and generate new paths and solutions;                  // see Section 5.2
4 Update the subsets P̄_c, c ∈ C of paths and resolve the RMP with integrality constraints; // see Section 5.3
```

6 Computational experiments

In order to assess the performance of the methodology described in Sections 4 and 5, we have conducted a series of tests for which we present computational results in this section. The algorithm has been coded in Python v3.9.14. IBM CPLEX 22.1.1 has been used to solve the linear and integer programs associated with our method. All experiments has been executed on an Intel Xeon Gold 6258R CPU @ 2.70GHz with a 512GB RAM computer and given a time limit of 48 hours.

In this campaign we consider the following configurations for the two proposed methods. The baseline CG (denoted "CG") solves a heuristic pricing subroutine whenever possible to generate paths of positive reduced costs, and an exact pricing routine upon failing to generate paths with the heuristic. Our hybrid matheuristic (denoted "CG+LS"), on the other hand, considers a simplified CG phase where the exact pricing subroutine is entirely ignored.

The remainder of this section is divided in two, as follows. In Section 6.1 we compare the performance of our proposed methods in this paper against Cobeña et al. (2023), using the same CAB data instances that the authors used. Then, in Section 6.2 we demonstrate the effectiveness of the proposed methodology employing a real data set from the metropolitan area of Montreal.

6.1 Computational analysis and comparison

We assess the computational efficiency of the proposed methodologies to solve ED-HLLP and solution algorithm using the well-known Civil Aeronautics Board (CAB) (see, O'kelly, 1987) as the basis of our testbed. In the next two sections we first describe the experimental setup used through this experiment, and next we present detailed computational results where we compare the proposed methods against the enumeration based of Cobeña et al. (2023) for the same problem.

6.1.1 Experimental setup

The population weights of each node P_i are calculated as the sum of all inbound and outbound demand in each node. Moreover, the results applying the methodologies proposed in this paper are compared with the methodology applied for the authors Cobeña et al. (2023) to solve the linear formulation of ED-HLLP (For the parallel processes, the maximum number of simultaneous sub-processes is fixed to 4 CPUs).

We considered instances with $n \in \{10, 15, 20, 25\}$. These instances have symmetric OD demand matrices. In addition, regarding the parameters of the problem, we use $R_c = (1 + \gamma_c)t_{o_c d_c}$ where γ_c is drawn uniformly in $\in [0, 1]$. Parameters r, α, and p are as follows: $r = 1.7$, $\alpha \in \{0.2, 0.5, 0.8\}$ and $p \in \{3, 5, 7\}$ (see, Fotheringham and O'Kelly, 1989; Zetina et al., 2019). Similarly to Martins de Sá et al. (2015), the access and exit times do not depend on the node, i.e., $\tilde{t}^a_k = \tilde{t}^a$ and $\tilde{t}^e_k = \tilde{t}^e$ for all $k \in N$. Additionally, the access and exit times are defined as a proportion of the average travel time:

$$\tilde{t}^a = \tilde{t}^e = \vartheta \frac{\sum_{(i,j)\in A} t_{ij}}{n \cdot (n-1)},$$

where ϑ is fixed to 0.1 for these computational results.

For the presentation of our results, we sometimes report mean values. Please note that in all cases, the averages are computed using a geometric mean. For a collection $X = \{x_1, \dots, x_n\}$ of n real numbers (not necessarily unique), the geometric mean is defined as:

$$gm(X) = \sqrt[n]{\prod_{i=1}^{n} x_i}.$$

The geometric mean is less sensitive to outliers when compared to the arithmetic mean. However, it may be biased when the set X contains numbers close to zero. In those cases, it may be convenient to consider the shifted geometric mean that, for X and a shift value $t \in \mathbb{R}$, is defined as

$$gm(X,t) = \sqrt[n]{\prod_{i=1}^{n}(x_i + t)} - t.$$

In this manuscript we consider geometric means for all the averages reported, except for the average computing times that are reported using a shifted geometric mean with shift value $t = 1.0$, and for the average gaps that are computed using a shift value of $t = 0.01$. Since all problems are given a time limit of two days, timeouts are given a time of $86,400 \times 7 = 604,800$ seconds for the effect of computing the means of the CPU times. Memory limits are not given any specific time, but instead are ignored from the computation of the means across all methods. In this way we make sure that all the means reported are comparable.

6.1.2 Performance analysis of proposed methods

This section shows the results of the two proposed methodologies to solve the ED-HLLP. In Tables 1 and 2, the first column presents the number of nodes (n), the second column reports the number of open hubs (p), then the used values of the parameters ϑ, r and α are specified.

The number of paths is displayed in column $\mid \mathcal{P} \mid$, the total time in seconds to generate paths and time to solve the integer model are reported in columns t_{paths}, t_{MIP} respectively. Columns z_{LB^1} and z_{UB} represent the primal and dual bound, respectively, the percentage of paths that the proposed CG does not enumerate is reported under column "%del". Column z_{LB^2} represent the primal bound applying the second methodology (CG+LS). The column labeled t_{tot} represents the total CPU time (in seconds) spent by the exact method proposed by Cobeña et al. (2023); this time is the time to generate all paths using the parallel process that the authors used plus time to solve the MILP. Column labeled "Opt" represent the optimal solutions within the time limit under the methodology proposed by Cobeña et al. (2023).

Let us now describe the three types of relative gaps reported. The value gap^1 is calculated $(z_{UB} - Opt)/Opt \times 100$, gap^2 is calculated as $(Opt - z_{LB^1})/z_{LB^1} \times 100$, and gap^3 is calculated as $(Opt - z_{LB^2})/z_{LB^2} \times 100$. In simple words, the value gap^1 represents the relative deviation between the dual bound computed by our CG and the optimal solution; the value gap^2 represents the relative deviation from the optimal solution of the primal bound achieved upon solving the RMP with integrality constraints immediately after solving the linear relaxation, this is without considering our local search; the value gap^3 corresponds to the relative deviation from the optimal solution of the primal bound achieved by our hybrid matheuristic.

Tables 1 and 2 show the performance of our proposed methodologies against the method used in Cobeña et al. (2023). We can see in Table 1 that for small instances with $n \in \{10, 15\}$ our first method described in Section 4 is effective in generating good primal solutions (z_{LB^1}), with most instances achieving the optimal or near-optimal solutions and a significant percentage of deleted paths (% del). For example, we can see that for the instances with $n = 15$ and for different values of p and α that our method can solve them in less than 1 hour in a sequential way, often faster to Cobena's method that solves them in a parallel way to find all paths in usually longer computing times. Furthermore, on average, this method in these instances solves them in less than 1 minute with a gap^1 of 1.05% and a gap^2 of 0.87%, demonstrating that our method remains competitive for these small problems.

Table 1 shows that the quality of the primal bounds z_{LB^1} are improved with the second method proposed (z_{LB^2}), combining the heuristic pricing algorithm and local search. We can see in the "gap^3" column, most of the gaps are 0.00%. Furthermore, on average, this method solves them is less than 10 seconds and on average the "gap^3" is 0.24%.

Table 1: Comparison of performance and solution for $n \in \{10, 15\}$ with CAB instances

n	p	ϑ	r	α	This paper (CG)					% del	This paper (CG+LS)				gap^1	ED-HLLP Cobeña et al. (2023)			gap^2	gap^3
					$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^1}	z_{UB}		$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^2}		$\mid\mathcal{P}\mid$	t_{tot}	**Opt**		
				0.2	3556	0.1	0.1	190104	190788	9.05%	2398	1.6	0.1	190104	0.36%	3910	0.4	190104	0.00%	0.00%
10	3	0.1	1.7	0.5	1098	2.0	0.0	35001	35001	17.07%	1042	1.3	0.0	35001	0.00%	1324	0.1	35001	0.00%	0.00%
				0.8	188	1.4	0.0	1767	1767	4.08%	172	0.5	0.0	1767	0.00%	196	0.1	1767	0.00%	0.00%
				0.2	20134	108.2	1.5	314993	327751	67.36%	6272	5.8	13.5	318197	3.00%	61692	16.0	318197	1.02%	0.00%
10	5	0.1	1.7	0.5	3296	30.0	1.4	51761	55083	43.91%	1850	9.6	0.3	52411	5.10%	5876	4.6	52411	1.26%	0.00%
				0.8	210	2.0	0.0	3032	3071	31.82%	244	1.6	0.0	3032	1.29%	308	0.5	3032	0.00%	0.00%
				0.2	26766	249.4	41.5	361535	401746	93.78%	16202	28.5	64.5	373933	3.91%	430314	1668.6		6.95%	3.40%
10	7	0.1	1.7	0.5	3052	52.4	9.5	63031	70739	72.76%	2396	22.8	7.6	64946	8.92%	11206	34.9		3.04%	0.00%
				0.8	208	4.0	0.1	3985	4163	32.90%	254	3.2	0.1	3985	4.47%	310	5.6	3985	0.00%	0.00%
				0.2	8246	14.1	0.1	1364045	1364045	59.12%	6168	4.8	0.9	1364046	0.00%	20172	1.3	1364050	0.00%	0.00%
15	3	0.1	1.7	0.5	3402	7.4	0.1	250647	250647	42.22%	2624	2.9	0.3	250647	0.00%	5888	0.8	250647	0.00%	0.00%
				0.8	670	2.7	0.0	27510	27510	16.87%	504	1.8	0.0	27510	0.00%	806	0.3	27510	0.00%	0.00%
				0.2	61048	318.8	26.4	1798981	1908211	93.27%	16706	14.4	29.1	1868014	0.10%	906590	4158.6		5.96%	2.04%
15	5	0.1	1.7	0.5	13932	194.6	0.2	341080	351007	77.28%	4578	11.9	0.6	345665	1.20%	61330	107.4	346829	1.69%	0.34%
				0.8	774	27.9	0.1	34012	34012	65.93%	782	7.2	0.0	34012	0.00%	2272	12.3	34012	0.00%	0.00%
				0.2	110148	2615.1	91.8	1848060	2106849	99.62%	42068	58.0	331.0	1927969	-	28999594	mem		-	-
15	7	0.1	1.7	0.5	19608	2002.5	81.7	417624	445284	93.81%	7016	114.0	1.8	432351	1.66%		316966		4.89%	1.31%
				0.8	774	39.3	0.3	37400	38185	72.08%	1046	22.2	0.2	37964	0.58%		2772		1.51%	0.00%
Mean					**2767**	**23.3**	**1.7**			**40.71%**	**1873**	**6.8**	**1.5**		**1.05%**	**7895**	**18.8**		**0.87%**	**0.24%**

We next analyze in Table 2 the results for instances with $n \in \{20, 25\}$. While our CG method provides good dual bounds (with an average deviation of 3.85%), it fails as a primal heuristic as it achieves an average gap^2 of 5.33%. Our hybrid matheuristic on these problems proves much more efficient to find feasible configurations of high-quality, with an average gap^3 of 0.2%, within computing times often orders of magnitude lower than for Cobeña et al. (2023).

Table 2: Comparison of performance and solution for $n \in \{20, 25\}$ with CAB instances

n	p	ϑ	r	α	This paper (CG)					% del	This paper (CG+LS)				gap^1	ED-HLLP Cobeña et al. (2023)			gap^2	gap^3
					$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^1}	z_{UB}		$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^2}		$\mid\mathcal{P}\mid$	t_{tot}	**Opt**		
				0.2	65184	201.5	3.1	7520167	8681283	35.84%	13856	14.1	2.5	8215958	5.7%	101600	6.1	8215970	9.25%	0.00%
20	3	0.1	1.7	0.5	24406	47.1	0.4	1378229	1590003	17.54%	7208	9.7	1.4	1456083	9.2%	29596	3.8	1456080	5.65%	0.00%
				0.8	3530	13.8	0.1	144485	144485	9.16%	1360	4.7	0.2	144485	0.0%	3886	1.1	144485	0.00%	0.00%
				0.2	318222	3910.7	673.2	11063604	12309935	96.71%	85022	1807.0	237.6	11888094	2.3%	9683748	15335.3	12032900	8.76%	1.22%
20	5	0.1	1.7	0.5	151562	2789.0	124.7	1892184	2207708	78.11%	21178	230.5	185.1	2047280	6.7%	692466	651.1	2069720	9.38%	1.10%
				0.8	4680	91.7	0.3	174727	189417	80.16%	2344	23.0	0.7	183535	3.2%	23592	126.1	183535	5.04%	0.00%
				0.2	337616	6475.0	1144.7	8764050	9955766	-	112988	96.0	731.8	9076959	-	-	time		-	
20	7	0.1	1.7	0.5	103380	7973.6	311.1	1991589	2084199	98.70%	18210	129.2	70.8	2050411	-	7977830	mem		-	
				0.8	8464	783.5	36.5	197314	212704	82.59%	3300	73.0	4.3	207659	2.4%	48622	21909.8	207659	5.24%	0.00%
				0.2	213792	473.3	9.9	15821095	18346406	31.99%	24490	28.4	0.8	17066439	7.5%	314344	20.4	17066500	7.87%	0.00%
25	3	0.1	1.7	0.5	53826	104.0	0.5	2800596	3213313	34.59%	12160	16.7	0.5	3018559	6.5%	82296	11.2	3018560	7.78%	0.00%
				0.8	3842	24.8	0.3	269573	298545	53.13%	2606	9.5	0.2	279629	6.8%	8198	2.9	279628	3.73%	0.00%
				0.2	422908	1674.5	860.0	18310039	20265806	99.20%	72166	44.6	421.9	19132368	-	53008052	mem		-	-
25	5	0.1	1.7	0.5	164652	5001.9	75.5	3705786	4195470	94.78%	27776	163.6	38.1	3985180	4.3%	3154792	32648.8	4021640	8.52%	0.91%
				0.8	12496	999.5	16.2	377178	396779	80.98%	4374	38.6	26.8	390592	1.6%	65684	853.6	390592	3.56%	0.00%
				0.2	785430	16929.8	6044.8	17163561	19750300	-	200656	311.5	33571.5	19750063	-		time		-	-
25	7	0.1	1.7	0.5	39900	9272.3	187.5	3340542	3549913	-	48204	307.9	749.8	3504538	-		time		-	-
				0.8	13040	12255.2	52.5	437827	466640	-	5946	200.4	43.2	453527	-		time		-	-
Mean					**42963**	**744.7**	**25.2**			**47.78%**	**13460**	**60.9**	**23.0**		**3.85%**	**180076**	**1133.9**		**5.33%**	**0.20%**

We are now in a position to highlight the following remark. The baseline CG consistently constructs a larger pool of paths when compared to our hybrid matheuristic. The latter, however, consistently achieves better primal solutions (by comparing gap^2 and gap^3). While this observation may seem counterintuitive at first, it is often the case when the optimal solution of the linear relaxation of a problem is not entirely capable of capturing the structure of an integer-feasible optimal solution. This has been observed in the past in problems that present with high degrees of fractionality (see, for instance, Joncour et al., 2010; Michel and Vanderbeck, 2012). Again, this highlights the efficiency of the proposed matheuristic which consistently constructs near-optimal solutions to the ED-HLLP.

6.2 Case study: Montreal's metropolitan area

We now apply our proposed methodology to a case study involving data from the City of Montreal, Canada. More specifically, we consider data from Montreal's 2018 Origin-Destination (OD) survey, as utilized by Cobeña et al. (2023), building on the analysis conducted with the CAB instances in the previous section. This analysis uses the methodology CG+LS described in Section 5.

6.2.1 Comparison against Cobeña et al. (2023)

In this section, we conduct a detailed comparative analysis using a test-bed of 36 instances derived from the Montreal OD survey, ranging from small- to large-scale. These instances are divided into three sets based on size: the first set includes 12 small instances with $n \in \{10, 15\}$; the second set comprises medium-sized instances with $n \in \{20, 25\}$; and the third set contains large instances with $n \in \{30, 39\}$. The classification into small, medium, and large scales follows the criteria established in the previous analysis. For these experiments, we adopted the parameters $r = 2.68$ (see, Goh et al., 2012), $\vartheta = 0.1$, and discount factors $\alpha \in \{0.2, 0.5\}$. The time limit was set to 48 hours for all runs. Tables 3-5 report the same type of data also reported in Tables 1-2, except that we omit the computation of the dual bounds, to focus explicitly in the performance of the proposed matheuristic.

In Table 3, we observe that the proposed matheuristic remains competitive with the method of Cobeña et al. (2023), taking about the same time and achieving the optimal solutions in all these instances. Moreover, it is also more robust as it never runs out of resources, as opposed to Cobeña et al.'s method that runs out of memory in one instance.

Table 3: Comparison between the proposed matheuristic and Cobeña et al. (2023) for $n \in \{10, 15\}$ on MTL instances

n	p	ϑ	r	α	CG+LS				ED-HLLP Cobeña et al. (2023)			gap^3
					$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^2}	$\mid\mathcal{P}\mid$	t_{tot}	**Opt**	
10	3	0.1	2.68	0.2	902	2.0	0.3	15270	1252	0.3	15270	0.00%
				0.5	272	0.8	0.0	1480	292	0.1	1480	0.00%
10	5	0.1	2.68	0.2	6948	9.0	3.0	27096	21288	5.7	27096	0.00%
				0.5	330	2.4	0.3	2442	358	0.5	2442	0.00%
10	7	0.1	2.68	0.2	18440	33.9	57.7	35106	86956	127.9	35106	0.00%
				0.5	334	3.9	0.3	3161	358	4.9	3161	0.00%
15	3	0.1	2.68	0.2	4528	8.4	3.0	18960	7692	9.2	18960	0.00%
				0.5	1002	4.1	0.2	1881	1226	0.3	1881	0.00%
15	5	0.1	2.68	0.2	15902	35.1	23.6	32729	266882	169.5	32729	0.00%
				0.5	1192	10.7	1.3	3016	2012	10.1	3016	0.00%
15	7	0.1	2.68	0.2	43310	175.9	209.1	43584	4367192	mem		-
				0.5	1372	30.8	6.9	3879	2018	608.2	3879	0.00%
Mean					**1729**	**7.6**	**2.5**		**3476**	**9.5**		**0.00%**

For the medium-sized instances with $n \in \{20, 25\}$, the results reported in Table 4 show a clear edge in favor of our method, leading to computing times one order of magnitude lower than Cobeña et al.'s on average (120 seconds vs 19 minutes), while ensuring solutions that are no farther from the optimal ones than 0.13% on average. The robustness of the proposed method is confirmed, as it is efficient to address all problems, while Cobeña et al. (2023)'s method runs out of resources in four of them.

For the instances with $n \in \{30, 39\}$ reported in Table 5, our method again shows robustness; in spite of the considerable computational challenges posed by these instances, our method efficiently generates promising paths which lead to the optimal solutions in all cases for which such certificate is available, which showcases the robustness of our method across all instance sizes.

Table 4: Comparison between the proposed matheuristic and Cobeña et al. (2023) for $n \in \{20, 25\}$ on MTL instances

n	p	ϑ	r	α	CG+LS				ED-HLLP Cobeña et al. (2023)			gap^3
					$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^2}	$\mid\mathcal{P}\mid$	t_{tot}	**Opt**	
20	3	0.1	2.68	0.2	9110	18.8	79.9	27078	24276	354.0	27078	0.00%
				0.5	2070	10.6	0.3	2596	2962	0.9	2596	0.00%
20	5	0.1	2.68	0.2	33070	56.9	1167.4	47036	1404644	2170.6	47871	1.74%
				0.5	2486	30.5	12.0	4259	6708	101.7	4259	0.00%
20	7	0.1	2.68	0.2	93202	265.6	9758.8	63672	51308490	mem		-
				0.5	2986	104.6	29.6	5724	6756	14410.2	5724	0.00%
25	3	0.1	2.68	0.2	16208	32.3	117.3	28486	83198	2087.1	28486	0.00%
				0.5	3874	24.2	0.7	2754	8920	2.6	2754	0.00%
25	5	0.1	2.68	0.2	62018	112.3	11007.1	49762	8014748	mem		-
				0.5	5214	68.0	38.3	4609	40146	719.7	4609	0.00%
25	7	0.1	2.68	0.2	370664	488.0	69806.6	67519		time		-
				0.5	6594	260.5	61.4	6111		time		-
Mean					**9055**	**55.4**	**65.2**		**24653**	**1154.6**		**0.13%**

Table 5: Comparison between the proposed matheuristic and Cobeña et al. (2023) for $n \in \{30, 39\}$ on MTL instances

n	p	ϑ	r	α	CG+LS				ED-HLLP Cobeña et al. (2023)			gap^3
					$\mid\mathcal{P}\mid$	t_{paths}	t_{MIP}	z_{LB^2}	$\mid\mathcal{P}\mid$	t_{tot}	**Opt**	
30	3	0.1	2.68	0.2	24336	63.3	293.0	30066	175142	4156.4	30066	0.00%
				0.5	5922	46.8	1.2	2935	16000	6.2	2935	0.00%
30	5	0.1	2.68	0.2	96834	172.5	10910.6	52113	23985418	mem		-
				0.5	7676	132.3	14.2	4864	84798	2907.0	4864	0.00%
30	7	0.1	2.68	0.2	295274	644.0	11080.6	71217		time		-
				0.5	10410	516.9	136.6	6332		time		-
39	3	0.1	2.68	0.2	42432	152.1	967.2	30451	539280	86511.1	30451	0.00%
				0.5	10920	176.5	2.8	3036	41652	31.2	3036	0.00%
39	5	0.1	2.68	0.2	188454	424.2	66609.3	52199		time		-
				0.5	15048	374.9	86.9	4907	350788	time		-
39	7	0.1	2.68	0.2	1562846	1935.5	86404.7	72054		time		-
				0.5	22902	1304.3	391.7	6496		time		-
Mean					**36644**	**294.5**	**353.1**		**111020**	**28932.3**		**0.00%**

We can conclude that the proposed method is efficient at consistently identifying near-optimal solutions which highlights its practical applicability to address large-sized hub-line location problems, as opposed to Cobeña et al. (2023)'s method which fails to scale past medium-sized problems.

6.2.2 Sensitivity analysis of r, p and α

In this section we analyze how parameters r, p and α impact the topology of the resulting hub network in Montreal using the methodology proposed in Section 5.2 on large-sized problem instances, more specifically on those with $n \in \{25, 30, 39\}$. The metrics used to perform the sensitivity analysis are similar to those used in Cobeña et al. (2023), namely:

Spatial Distributions. The % of served demand measures the proportion of total demand that benefits from the hub network, highlighting the spatial impact of different parameter configurations.

Total Travel Time. The % of saved time assesses the average reduction in travel time resulting from the hub network's implementation, providing insights into the efficiency gains achievable through an optimal hub placement.

Figures 3 and 4 show the percentage of demand served and the average time saved after the establishment of the hub line and different number of hubs for $n = 39 \mid \vartheta = 0.1 \mid r = 2.68 \mid \alpha = 0.2, 0.5$. The demand increases with the installation of more hubs in the city, and the average saved time after establishing the hub line decreases as the values of hubs increase and the values of α decrease. Figures 3 and 4 show the impact between the number of hubs (p) and the discount factor (α) on the percentage of demand served and the average time saved after establishing the hub line for $n = 39$, $\vartheta = 0.1$, and $r = 2.68$.

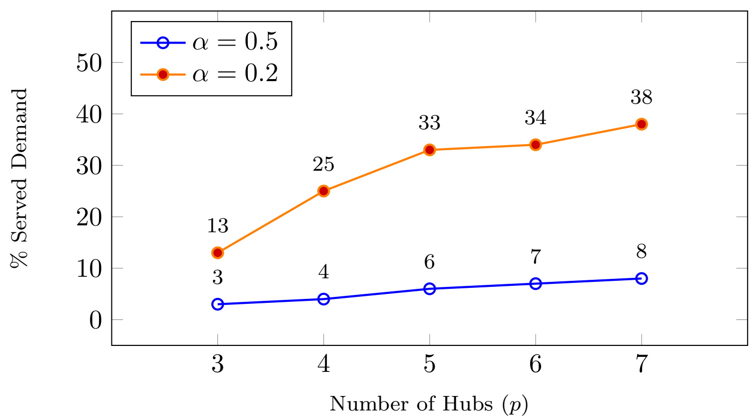


Figure 3: Impact of parameter α on % Served Demand for Different Numbers of Hubs. $n = 39 \mid \vartheta = 0.1 \mid r = 2.68$

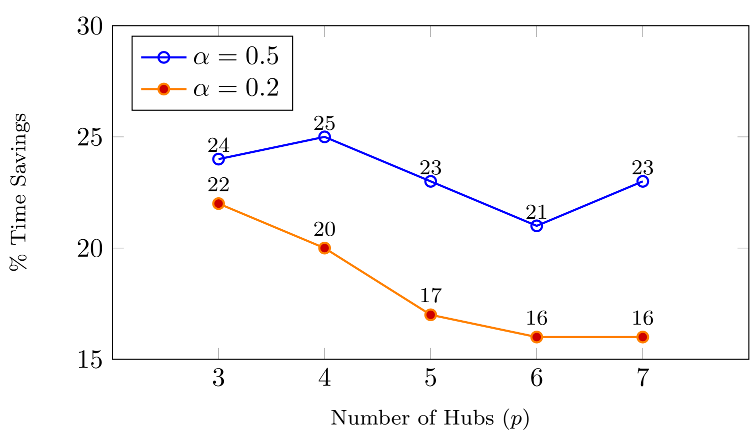


Figure 4: Impact of of parameter α on % Time Savings for Different Numbers of Hubs. $n = 39 \mid \vartheta = 0.1 \mid r = 2.68$

We observe that an increase in the number of hubs correlates positively with an increase in the percentage of demand served, particularly at a high discount factor ($\alpha = 0.2$). This shows that a high discount factor (1 - α), improves the utilization of the hub line system (see Figure 3). We can also see in Figure 4 that the average time saved decreases when the number of hubs increases and the discount factor decreases. The increase in hubs may enhance demand coverage but may also increase travel times.

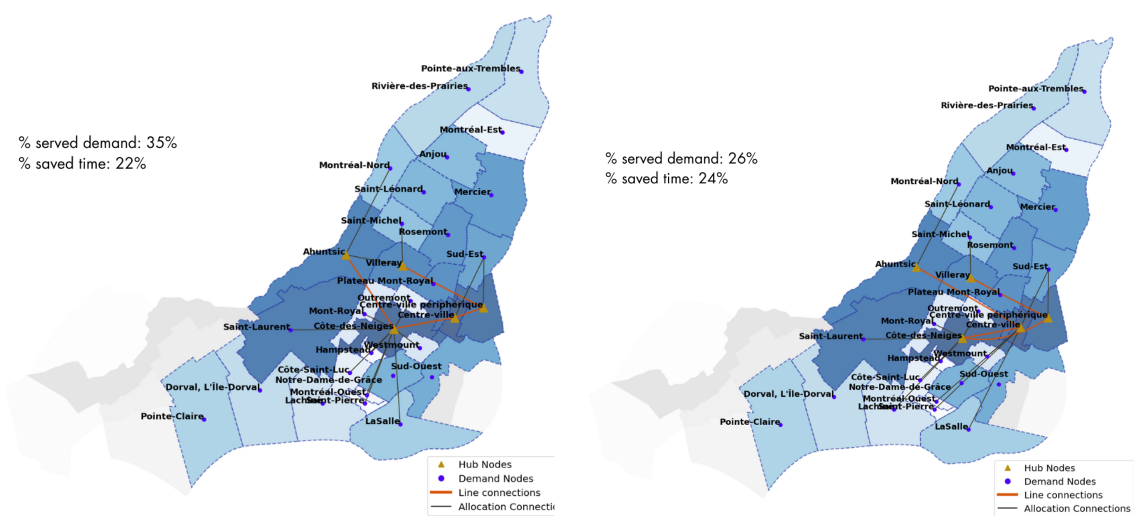
Furthermore, Table 6 shows the % of served demand and % of saved time for different values of n. We can see that as the values of α decrease and p increases, the spatial distribution tend to increases too; moreover, the average time saved is higher for a small number of hubs (p).

Table 6: Sensitivity analysis of # hubs(p) and discount factor(α)

n	p	r	ϑ	α	% served demand	% saved time
25	3	2.68	0.1	0.2	18%	31%
				0.5	4%	26%
	5	2.68	0.1	0.2	42%	20%
				0.5	8%	22%
	7	2.68	0.1	0.2	55%	20%
				0.5	12%	21%
30	3	2.68	0.1	0.2	16%	25%
				0.5	4%	25%
	5	2.68	0.1	0.2	35%	22%
				0.5	7%	21%
	7	2.68	0.1	0.2	56%	19%
				0.5	10%	20%
39	3	2.68	0.1	0.2	13%	22%
				0.5	3%	24%
	5	2.68	0.1	0.2	33%	17%
				0.5	6%	23%
	7	2.68	0.1	0.2	44%	16%
				0.5	8%	13%

6.2.3 Analysis of the hub-line configurations

We now analyze the different hub-line configurations and the metrics presented above for different values of n and r for large-sized problems. Figures 5a and 5b show the hub lines obtained for $n = 30$, $p = 5$, $\alpha = 0.2$, and $r \in \{2.68, 1.7\}$. Note that although the selected hub nodes are similar in both configurations, the resulting hub lines differ, which brings to light the sensitivity of the resulting hub line network to changes in the parameter r. This variation shows the importance of fine-tuning r to balance efficiency and connectivity.



(a) $n = 30, p = 5, \alpha = 0.2, r = 2.68$

(b) $n = 30, p = 5, \alpha = 0.5, r = 1.7$

Figure 5: Hub line configuration for the study case with $n = 30, p = 5, \vartheta = 0.1$

In Figure 6, for $n = 39$, $p = 5$, $r = 2.68$ and $\alpha = 0.2$, one of the hubs chosen is *Villeray*, a neighbourhood centrally located in the metropolitan area of Montreal. Its selection is likely driven by the high density around it, its proximity to important Points of Interest (POI) such as the Jean-Talon Market and Little Italy, and its closeness to shopping areas like Rockland Center and Marché Central. Another important node selected as a hub is *Saint Laurent*, close to Hotels and Montréal-Pierre Elliott Trudeau International Airport. This hub line configuration aligns with the model's focus on maximizing profit by reducing travel times and capturing high demand, as outlined in the ED-HLLP framework.

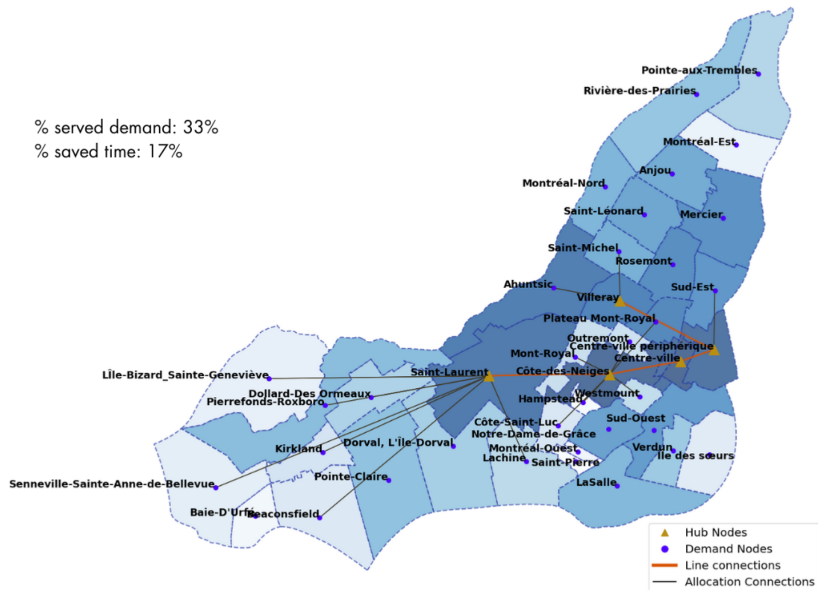


Figure 6: Hub line configuration for the study case with $n = 39, p = 5, \vartheta = 0.1, \alpha = 0.2, r = 2.68$

7 Conclusions

We have introduced a column generation-based algorithm and a hybrid matheuristic that combines column generation with local search to address the ED-HLLP for large-sized problems. The proposed methods are based upon the solution of a linear program with a very large number of variables (columns), from which only a subset of promising ones is identified. The proposed matheuristic is shown to be more robust and to provide optimal and near-optimal solutions on all problems considered when compared to the method introduced in Cobeña et al. (2023), providing valuable insights into the modeling of profit-oriented hub line location problems with elastic demands. Moreover, the proposed CG is also capable of providing dual bounds when powered by an exact pricing subroutine, but remains practical for small- to medium-sized problems only.

We have also conducted an analysis on a real network, namely that of the metropolitan area of Montreal, and performed a thorough sensitivity analysis to assess the behavior of the resulting solutions to different parameters. The proposed method can address the ED-HLLP effectively for problems with up to $n = 39$ nodes, higher than the reach of Cobeña et al.'s method. This shows our approach's practical applicability in real-world instances of the ED-HLLP, particularly in complex urban settings such as those of a city like Montreal.

Finally, our work provides a valuable tool for decision-makers such as city or transport planners, allowing them to design more effective and efficient transit networks, improving the accessibility of the citizens and reducing their travel times through the use of the hub line system, primarily to address the continuous increase of the demand of urban mobility, ensuring that urban public transportation systems can meet the needs of expanding urban areas and their surroundings.

Future research would incorporate additional service levels at the hub stations, particularly at the hub line's access and exit hubs, to enhance their attractiveness by integrating services such as car/bike share stations or POIs.

Appendix

A Starting feasible solution

Our procedure to obtain an initial feasible solution to be used in the RMP computes the shortest paths in a hub line composed of p hubs, by commodity and type of path. As mentioned in Section 3, the objective of the problem is to maximize the total revenue for the time saving using the hub line. Here, $f_c(t'_c)$ corresponds to the profit obtained for each commodity $c \in C$, for each OD pair.

$$f_c(t'_c) = R_c \frac{P_{o_c} P_{d_c}}{(t'_c)^r} \left(t_{o_c d_c} - t'_c\right).$$

To obtain an initial solution, we look for the smallest value of t'_c, i.e., the shortest path by type between o_c and d_c using the hub line. If the shortest (in terms of time utilization) path is such that $t'_c < t_{o_c d_c}$ and its length between 2 and p, it is added to $\overline{\mathcal{P}}_c$.

The search for the shortest paths is performed in a weighted network G_c as described in Cobeña et al. (2023). This procedure allows us to build paths for every commodity c between its origin node o_c and its destination node d_c using the hub line with an associated travel time smaller than or equal to $t_{o_c d_c}$.

References

U. Nations, World Urbanization Prospects: The 2018 Revision, Statistical Papers - United Nations (Ser. A), Population and Vital Statistics Report, 2019.

N. Aydin, S. Seker, B. Özkan, Planning location of mobility hub for sustainable urban mobility, Sustainable Cities and Society 81 (2022) 103843.

I. Contreras, M. O'Kelly, Hub location problems, Location science (2019) 327–363.

E. Martins de Sá, I. Contreras, J.-F. Cordeau, R. De Camargo, G. De Miranda, The hub line location problem, Transportation Science 49 (2015) 500–518.

B. Cobeña, I. Contreras, L. I. M. Merino, A. M. Rodríguez-Chía, The profit-oriented hub line location problem with elastic demand, Computers & Operations Research (2023) 106335.

M. E. O'kelly, The location of interacting hub facilities, Transportation science 20 (1986) 92–106.

S. Alumur, B. Y. Kara, Network hub location problems: The state of the art, European journal of operational research 190 (2008) 1–21.

S. Nickel, A. Schöbel, T. Sonneborn, Hub location problems in urban traffic networks, Mathematical methods on optimization in transportation systems (2001) 95–107.

S. Gelareh, S. Nickel, Hub location problems in transportation networks, Transportation Research Part E: Logistics and Transportation Review 47 (2011) 1092–1111.

W. Zhong, Z. Juan, F. Zong, H. Su, Hierarchical hub location model and hybrid algorithm for integration of urban and rural public transport, International Journal of Distributed Sensor Networks 14 (2018) 1550147718773263.

M. E. O'kelly, A quadratic integer program for the location of interacting hub facilities, European Journal of Operational Research 32 (1987) 393–404.

A. T. Ernst, M. Krishnamoorthy, Efficient algorithms for the uncapacitated single allocation p-hub median problem, Location science 4 (1996) 139–154.

S. A. Alumur, J. F. Campbell, I. Contreras, B. Y. Kara, V. Marianov, M. E. O'Kelly, Perspectives on modeling hub location problems, European Journal of Operational Research 291 (2021) 1–17.

V. Marianov, D. Serra, C. Revelle, Location of hubs in a competitive environment, European Journal of Operational Research 114 (1999) 363–371.

H. Eiselt, V. Marianov, A conditional p-hub location problem with attraction functions, Computers and Operations Research 36 (2009) 3128–3135.

V. Marianov, M. Rios, F. J. Barros, Allocating servers to facilities, when demand is elastic to travel and waiting times, RAIRO - Operations Research 39 (2005) 143–162.

V. Marianov, M. Ríos, M. J. Icaza, Facility location for market capture when users rank facilities by shorter travel and waiting times, European Journal of Operational Research 191 (2008) 32–44.

R. Aboolian, O. Berman, D. Krass, Profit maximizing distributed service system design with congestion and elastic demand, Transportation Science 46 (2012) 247–261.

C. A. Zetina, I. Contreras, J. Cordeau, Profit-oriented fixed-charge network design with elastic demand, Transportation Research Part B: Methodological 127 (2019) 1–19.

J. De Dios Ortúzar, L. G. Willumsen, Flexible long range planning using low cost information, Transportation 18 (1991) 151–173.

O. Tamin, L. Willumsen, Transport demand model estimation from traffic counts, Transportation 16 (1989) 3–26.

G. F. Newell, Some issues relating to the optimal design of bus routes, Transportation Science 13 (1979) 20–35.

E. Cipriani, S. Gori, M. Petrelli, Transit network design: A procedure and an application to a large urban area, Transportation Research Part C: Emerging Technologies 20 (2012) 3–14.

R. Z. Farahani, M. Hekmatfar, A. B. Arabani, E. Nikbakhsh, Hub location problems: A review of models, classification, solution techniques, and applications, Computers & industrial engineering 64 (2013) 1096–1109.

A.-K. Rothenbächer, M. Drexl, S. Irnich, Branch-and-price-and-cut for a service network design and hub location problem, European Journal of Operational Research 255 (2016) 935–947.

S. Irnich, G. Desaulniers, Shortest path problems with resource constraints, in: Column generation, Springer, 2005, pp. 33–65.

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

L. Costa, C. Contardo, G. Desaulniers, Exact branch-price-and-cut algorithms for vehicle routing, Transportation Science 53 (2019) 946–985.

F. B. Zhan, C. E. Noon, A comparison between label-setting and label-correcting algorithms for computing one-to-one shortest paths, Journal of Geographic information and decision analysis 4 (2000) 1–11.

M. Desrochers, F. Soumis, A generalized permanent labelling algorithm for the shortest path problem with time windows, INFOR: Information Systems and Operational Research 26 (1988) 191–212.

A. Ceselli, G. Righini, M. Salani, A column generation algorithm for a rich vehicle-routing problem, Transportation Science 43 (2009) 56–69.

C. Joncour, S. Michel, R. Sadykov, D. Sverdlov, F. Vanderbeck, Column generation based primal heuristics, Electronic Notes in Discrete Mathematics 36 (2010) 695–702.

Y. Yuan, D. Cattaruzza, M. Ogier, F. Semet, D. Vigo, A column generation based heuristic for the generalized vehicle routing problem with time windows, Transportation Research Part E: Logistics and Transportation Review 152 (2021) 102391.

C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, P. H. Vance, Branch-and-price: Column generation for solving huge integer programs, Operations research 46 (1998) 316–329.

C. C. Ribeiro, P. Hansen, G. Desaulniers, J. Desrosiers, M. M. Solomon, Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems, Springer, 2002.

M. Gendreau, J.-Y. Potvin, et al., Handbook of metaheuristics, volume 2, Springer, 2010.

I. Contreras, J.-F. Cordeau, G. Laporte, Benders decomposition for large-scale uncapacitated hub location, Operations research 59 (2011) 1477–1490.

E. M. de Sá, I. Contreras, J.-F. Cordeau, Exact and heuristic algorithms for the design of hub networks with multiple lines, European Journal of Operational Research 246 (2015) 186–198.

A. Fotheringham, M. O'Kelly, Spatial interaction models: Formulations and applications, 1 ed., Kluwer Academic Publishers, Dordrecht, 1989.

S. Michel, F. Vanderbeck, A column-generation based tactical planning method for inventory routing, Operations research 60 (2012) 382–397.

S. Goh, K. Lee, J. S. Park, M. Choi, Modification of the gravity model and application to the metropolitan seoul subway system, Physical Review E 86 (2012) 026102.