On-Demand, Transit: Designing Hybrid Transportation Systems

Report by
Meilame Tayebjee

In Partial Fulfillment of the Requirements for the
Degree of
Cycle ingénieur polytechnicien - X 2020

Supervised by Prof. Alexandre Jacquillat

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Boston, Massachusetts, USA

ECOLE POLYTECHNIQUE
Palaiseau, France

Defended Sept. 2023

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Prof. Alexandre Jacquillat for giving me this wonderful opportunity to work during four months with him and to benefit from the resources of MIT and ORC, institutions in which I have always felt at home during my appointment. His guidance, pedagogy and availability have been crucial throughout the duration of my internship.
I am also very grateful to Prof. Julia Yan and Prof. Arthur Delarue for their greatly helpful involvement and guidance during the realization of this project.

I would like to extend my sincere thanks to Prof. Stéphane Gaubert, Prof. Xavier Allamigeon and Prof. Frédéric Meunier for having introduced me to Operations Reasearch through one of the best courses I have taken during my time at Polytechnique.

Lastly, I would like to deeply thank my family for their unconditional support throughout my studies.

ABSTRACT

Microtransit as a hybrid system mixing on-demand vehicles and transit emerged as a major research area for transportation network design. This work focuses on two specific metrics that heavily influence the design of a transportation network: spread of customers around transit stations and distance between origins and destinations. It tries to answer to the question: when (i.e. in which spread-distance combination) should customers choose transit, when should they choose on-demand and when should they choose hybrid?

In order to provide an answer, this work uses time-space networks and proposes a Mixed-Integer Linear Programming model that enables to design and operate a multimodal transportation network - choosing between transit, on-demand vehicles or hybrid - given a set of origin-destination pairs. It then focuses on the operational stage (assuming the operated transit lines as exogenous) and presents a Benders decomposition formulation. Finally, several experiments are conducted on synthetic data and results yield an empirical answer to the question above.

This report paves the way for further research to resolve tractability issues and to hinge the design stage to the operational stage and eventually apply the model to real-world data.

TABLE OF CONTENTS

Number *Page*

1

INTRODUCTION

Current urban mobility is divided into two clear, distinct modes of transportation: either using public transit transports (buses, subways, tramways - we call *transit* all the transporation that follow a clear timetable and clear route with predefined stops) or taking an on-demand vehicle (Uber, Lyft... - we call *on-demand* vehicles the ones that go specifically pick up the customer when called or booked, at a location defined *on the spot*) from origin to destination. The problems arising from this duopoly are numerous:

- On the one hand, transit networks are less and less used, due to a weak level of service in expanding and sparsely populated large cities. Rebuilding these networks call for a data-driven holistic design approach to enhance service quality (Bertsimas, Ng, and Yan, 2021).
- On the other hand, on-demand vehicle services such as Uber and Lyft have revolutionized the single-occupancy door-to-door mobility but are now responsible for traffic congestion, greenhouse gas emissions, increasing costs and waiting times (Diao, Kong, and Zhao, 2021).

Therefore, *hybrid mobilities* also called **microtransit** have sparked interest to solve these issues. Basically, we call hybrid mobility any time of mobility that involves a request to an on-demand vehicle as well as a transit public transportation. Typically, in the context of sparsely populated urban areas or rural areas, a passenger who lives far away from the train station can request a ride with an on-demand vehicle to go to the station (first-mile) ; they then takes the train, arrives at a station and can request another on-demand ride to their final destination (last-mile). Moreover, all the on-demand rides are shared with other passengers that follow a similar route (2).

Microtransit offers the possibility to get rid of the individual car, reducing congestion and pollution (2). It strengthens urban mobility in at least two ways:

1. Population without a car that would have taken a direct (possibly long and polluting) individual on-demand Uber or Lyft drive to their destination,- or would have given up the trip due to its high cost - are now able to get there in an affordable and sustainable way

2. Population that would have otherwise taken their individual car are more interested in taking the transit transportation that is now reachable, reducing therefore congestion and pollution

Our contributions

In the following work, we put ourselves in a transportation network designer's shoes that has access to both large-capacitated transit vehicles (also referred to as buses, trains...) and smaller on-demand vehicles (also referred to as vans, taxis or vehicles...). Given a dataset of trips (origin and destination) and a set of transit transit stations, we want to design an optimal system involving transit, on-demand and hybrid mobilities, in *a holistic way* : the design of the network and the routing of passengers and vehicles at minimum cost are intertwined and are the two stages of a single bilevel optimization. More broadly, the general aim of this report is to answer to the following questions:

1. Can we quantify the contribution of this new hybrid possibility over clear Key Performance Indicators (KPIs), compared to classic full transit or full on-demand systems?
2. In which *configurations* (in terms of spread of customers around stations (*radius*) and distances between these stations (Δ)) do customers choose full transit? full on-demand? hybrid?

We start with an overview of the related work about the different themes that are tackled by the project, namely: First-Mile Last-Mile microtransit, ride-sharing, holistic transportation design (2).

We then propose a Mixed-Integer Linear Programming (MILP) model that aims at assigning in an optimal way the customers to a *type* of route (transit, on-demand, hybrid) and routes both vans and passengers through a **time-space network**. This model can be decomposed in two stages, **design** (which line to open) and **operational** (routing of both passengers and vans) (3).

4 and 5 have a special focus on the second stage (operational) considering the operated transit lines and the number of vans as exogenous. We decompose this second stage itself into two stages and managed to run a Benders decomposition. We then conduct experiments that enables us to validate the model, answer to the above questions and gain further insights on its behaviour.

We end up in 6 by suggesting some possible next steps for further research.

RELATED WORK

Our project involves these following topics that have been the interests of many research works.

Multi-modality to support public transport: first-mile, last-mile

The emergence of on-demand ride platforms such as Uber and Lyft churned out a lot of opportunities to strengthen public transportation systems, using a hybrid mode between on-demand and transit.

Steiner and Irnich (2020) designed a strategic network planning optimization model using on-demand mobility as a possible mode for access and egress leg (first-mile, last-mile). The paper starts with an existing transit network, and designs a model that aims at defining zones where to add on-demand vehicles and eventually decide which segment of the transit network should still be included in the future network. Hence the future network being an improvement of the previous one. In this paper, the first mile and the last mile have the same mode (walking or on-demand) ; our model enables more hybridness and versatility by choosing between full transit, full on-demand, first-mile, last-mile or both.

Other works include a generalization of the Line Planning Problem by Banerjee and al. (2021), which uses single-occupancy on-demand vehicle and high-capacity transit buses : it makes two assumptions (the need to access a set of feasible lines and no interbuses transfers) which necessity is theoretically justified to achieve a $1 - \frac{1}{e} - \epsilon$ approximation. Salazar et al. (2018) studies the interaction between self-driving cars and public system and aims at maximizing social welfare.

Hub-Arc Location Problem and holistic design

The design of transportation system provided a set of stations, or the problem of linking a set of hubs with arcs is named the Hub-Arc Location Problem, introduced by Campbell, Ernst, and Krishnamoorthy (2005). Its two-stage-structure (deciding which arcs to open first and then how to route the flow at minimum cost) makes it quite suitable for a Benders decomposition (Benders, 1962).

Several variants of the HALP have been studied over the past few years. Mahéo, Kilby, and Hentenryck (2019) designed a Hub and Shuttle Public Transit System (HSPTS) for the city of Canberra, introducing a hybrid type of mobility between *shuttles* (also called mutli-hire taxis) and high-frequency bus in response to the city's covering of a wide geographic area which makes public transportation design challenging. Given a whole set of bus stops across the city, the paper focuses on the design of the network (i.e. deciding which hubs to link via a bus to leverage economies of scale while on shuttles for the remaining "last mile") and then the routing of the passengers: both stages being intertwined. The problem is separable in terms of trips and enables the authors to use powerful acceleration techniques for the Benders decomposition.

Basciftci and Hentenryck (2022) push further the analysis of Mahéo, Kilby, and Hentenryck - renaming the HSPTS into On-Demand Multimodal Transit Systems (ODMTS) - by studying the latent demand (new riders integrating the system) and how to integrate it.

Our work keeps the holistic design approach, yet introduces a temporal dimension thanks to the use of time-space networks: while the previous papers focus on the spatial routing (*where* to assign the customer), our model enables to also choose *when* to pick up, *when* to board etc.

Multimodality and Ride Sharing

Sharif Azadeh, van der Zee, and Wagenvoort (2022) focus on the addition of Demand Responsive Transport (DRT) (our on-demand) to supplement the Fixed Line and Schedule (FLS) system (our transit) in sparsely populated area. Both modes would share the same existing station infrastructure. DRT vehicle are "booked" thanks to an app, pick up the passenger at a station (and not at their origin location) and drop them off at their destination station ; with some detours to pick up other passengers that want to follow a similar trip (**ride-sharing**). On the other-hand, FLS buses are classicly timed and stop at every station on their scheduled route. Moreover, the model considers that customers have the choice between the modes, following a multinomila logit choice model. Therefore, rather than a an assignment problem, the paper focuses on the robustness of the design. Our model makes furthermore the on-demand vehicles completely independent of the transit stops, which adds a vehicle routing dimension (see Zhang et al. (2022)).

MODEL FORMALISM

3.1 Data, Inputs and Preprocessing

We assume as given:

- Passengers $p \in \mathcal{P}$, their origins (denoted as $o(p)$ for a passenger p) and destinations (denoted as $d(p)$ for a passenger p), as well as their departure times denoted as dep_time_p.
- Set of transit lines $\mathcal{L}$, each associated with an ordered sequence of stops and corresponding departure times (we will rather use the notion of frequence (a vehicle leaves the departure station each *freq* time-steps)).
- Spatial locations $i \in \mathcal{I}$, corresponding to origins, destinations and transit stops. These spatial locations are partitioned into *zones*.
- Number of on-demand vehicles originally in each zone.

Each location is given with a longitude-latitude pair. We classicly use the Euclidean distance and we denote by d_{l_1,l_2} the distance between locations l_1 and l_2. As we consider that the vehicles have a speed of 1, it is also the time needed by vehicles to travel between those two locations.

We create a **time-space network** (TSN) for on-demand operations. Each node $n \in \mathcal{N}$ corresponds to a location-time pair, where locations include all origins, all destinations and all transit stations—that is, the node set is given by $\mathcal{N} = \mathcal{I} \times \mathcal{T}$. A non-idle arc connects two nodes (i,t) and (j,s) if a vehicle starting from location i at time t will reach location j between $(s-1)$ and s. An idle arc connects each time-space node (i,t) to the next one in the same location, $(i,t+1)$.

We also create a **transit time-space network** using the same process, but for transit vehicle operations. Here, the locations only include the transit stations. **We will call *transit arcs* the arcs of this transit time-space network**.

To these original transit arcs, we add *transfer arcs* using the following process. Given a threshold MAX_TRANSFER_WAITING, if there is a pair of transit arcs (a_1, a_2) such as:

- $a_1 = ((l_1, t_1), (l_2, t_2))$ and $a_2 = ((l_2, t_3), (l_3, t_4))$
- $t_3 \geq t_2$ and $t_3 - t_2 \leq$ MAX_TRANSFER_WAITING
- Station l_1 and station l_3 are not connected by any of the original arcs

Then we create a new transit arc linking (l_1, t_1) and (l_3, t_4). We compute the mapping between each of these transfer arcs and the vector of transit arcs (a_1, intermediary waiting arcs at l_2, a_2).

For additional transfers, we can reiterate the process. We will denote as *n-transfer arcs* arcs that involve n transfers.

In the following, we will only deal with - at most - 1-transfer arcs. Unless clearly stated, 1-transfer arcs are included when we will refer to transit arcs.

Definition 1. *A **route** is a pair (transit arc, type) where type is a number between 1 and 5.*

The number *type* maps these following five cases:

1. **Full Transit type**: The passenger walks and uses transit lines exclusively. The passenger does not use any arcs in the on-demand time-space network: nothing appears on the on-demand time space network.
2. **Full On-Demand type**: The passenger takes a on-demand vehicle from origin to destination. No transit arcs are used. Therefore, for each passenger, the one and only route of type 2 is $(\emptyset, 2)$ where $\emptyset$ is a virtual *empty* transit arc. The passenger will have to be picked up by a on-demand vehicle on a certain time-space node **at his origin location**, and **dropped off at his destination location** on another time-space node.
3. **First-Mile On-Demand type**: The passenger route involves a first-mile on-demand trip from an origin time-space node to a boarding transit station time-space node, followed by a transit arc. The route includes one modal transfer and potentially, one or several transit transfers as specified by the transit arc.

4. **Last-Mile On-Demand type**: The passenger route involves a transit arc followed by a last-mile on-demand trip from the alighting transit station node to a destination node. The route includes one modal transfer and potentially, one or several transit transfers as specified by the transit arc.

5. **First-Mile & Last-Mile On-Demand type**: The passenger route involves a first-mile on-demand trip from the origin to the boarding transit station node, followed by a transit arc and then by a last-mile on-demand trip from the alighting transit station node to a destination node. The route includes two modal transfers plus, potentially, one or several transit transfers as specified by the transit arc.

Basically, a route encapsulates the transit arc taken (which can be the $\emptyset$ arc in the full on-demand case) and *how* you take it.

For each route, we can easily compute the implied walking distance, the number of modal and transit transfers and the transfer waiting time. We use all of these metrics to preprocess and discard all of the routes that do not meet the thresholds. For each passenger, the preprocessing also enables to discard some origin and destination nodes that are "too late" or "too early". For instance, for a given passenger, if we need at least t time steps to travel between their origin location and their destination location, we can discard all the destination nodes which times are below t. The preprocessing procedure is detailed in algorithm 1.

In the following, the time-space network will refer to the "on-demand" time-space network. Once the preprocessing achieved, the model only needs the on-demand time-space network.

Algorithm 1 Preprocessing

Require: On-Demand TSN, Transit TSN, max_walking, max_waiting, walking_speed

Generate all routes (transit_arc, type)

for $r = (transit\_arc, type) \in$ All routes **do**

$((s_1, t_1), (s_2, t_2))$ = transit_arc

for $p \in \mathcal{P}$ **do**

Compute distance from origin to station s_1: $d_{o(p),s_1}$

Compute walking time to station: $w_{o(p),s_1} = d_{o(p),s_1} * walking\_speed$

if $\mathbf{W}(r) \leq$ max_walking **then**

if type == 1 or type == 4 **then**

For these two types, we can discard the routes that board too early (we need to get to the station !) or too late (we don't want to wait too much)

if $dep\_time_p + w_{o(p),s_1} \leq t_1 \leq dep\_time_p + w_{o(p),s_1} + max\_waiting$ **then**

$\mathcal{R}_p^{type}$.add(r)

if type == 4 **then**

$\mathcal{N}_p^A$.add((s_2, t_2))

Here we force the fact that passengers do not wait for the vehicle at the alighting station ; there is only one single node where a vehicle can pick up the passenger, exactly when they alight. A way to relax this constraint is available in 6.

end if

end if

end if

if type == 3 or type == 5 **then**

if $dep\_time_p + d_{o(p),s_1} \leq t_1 \leq dep\_time_p + d_{o(p),s_1} + max\_waiting$ **then**

$\mathcal{R}_p^{type}$.add(r)

$\mathcal{N}_p^B$.add((s_1, t_1)) *// Same comment as above for the alighting node*

if type == 5 **then**

$\mathcal{N}_p^A$.add((s_2, t_2))

end if

end if

end if

end if

end for

end for

$\mathcal{R}_p^2 = (\emptyset, 2)\ \forall p$

$\mathcal{R}_p = \cup_{i \in [1,5]} \mathcal{R}_p^i\ \forall p$

$\forall p \in \mathcal{P}, \mathcal{N}_p^O = \{(o(p), t) : t \in [dep\_time_p, max\_waiting]\}$

$\forall p \in \mathcal{P}, \mathcal{N}_p^D = \{(d(p), t) : t \in [dep\_time_p + d_{o(p),d(p)}, dep\_time_p + d_{o(p),d(p)} * walking\_speed]\}$

return $\mathcal{R}_p, \mathcal{R}_p^1, \mathcal{R}_p^2, \mathcal{R}_p^3, \mathcal{R}_p^4, \mathcal{R}_p^5, \mathcal{N}_p^O, \mathcal{N}_p^B, \mathcal{N}_p^A, \mathcal{N}_p^D\ \forall p$

3.2 Notations

- **Preprocessing and routes**
  - For a given route $r = (a, type)$ where $a \neq \emptyset$, if a stems from station s_1 and ends in station s_2, we denote s_1 as $B(r)$ (the boarding station of the route) and s_2 as $A(r)$ (the alighting station).
  - For each route r, we can compute the walking distance implied, and we denote it by $\mathbf{W}(r)$. More precisely, for a route of type 1, we know that the passenger has to walk the distance from his origin location to the station $B(r)$ and from the alighting station $A(r)$ to the destination location. For *the* route of type 2 and routes of type 5, the walking distance is 0. For type 3, the walking distance is from the alighting station $A(r)$ to the destination. For type 4, it is the distance between the origin location and the boarding station $B(r)$.
  - $\forall p \in \mathcal{P}, \forall i \in [1,5], \mathcal{R}_p^i$ denotes the ensemble of routes of type i available for passenger p after preprocessing and $\mathcal{R}_p = \cup_{i\in[1,5]}\mathcal{R}_p^i$. As stated before, $\forall p \in \mathcal{P}, \mathcal{R}_p^2 = \{(\emptyset, 2)\}$.
  - $\forall p \in \mathcal{P}, \mathcal{N}_p^O$ denotes all the available nodes after preprocessing for being picked up by a on-demand vehicle **at origin location**. These nodes need to be chosen for route type 2, 3 and 5 (see (3.2)).
  - $\forall p \in \mathcal{P}, \mathcal{N}_p^B$ denotes all the available nodes after preprocessing for being dropped off at a departure station by a on-demand vehicle. These nodes need to be chosen for route type 3 and 5. For each passenger p, for each route $r \in \mathcal{R}_p^3 \cup \mathcal{R}_p^5$, we denote by $B(r) \in \mathcal{N}_p^B$ the boarding node associated.
  - $\forall p \in \mathcal{P}, \mathcal{N}_p^A$ denotes all the available nodes after preprocessing for being picked up at a alighting station by a on-demand vehicle. These nodes need to be chosen for route type 4 and 5. For each passenger p, for each route $r \in \mathcal{R}_p^4 \cup \mathcal{R}_p^5$, we denote by $A(r) \in \mathcal{N}_p^A$ the alighting node associated.
  - $\forall p \in \mathcal{P}, \mathcal{N}_p^D$ denotes all the available nodes - after preprocessing - for being dropped off **at a destination location** by a on-demand vehicle. These nodes need to be chosen for route type 2, 4 and 5 (see (3.3)).
- **Time-space network**
  - Time horizon T
  - $\mathcal{N}$: the ensemble of nodes. We will denote by $\mathcal{N}_0$ the ensemble $\{n = (i,t) \in \mathcal{N} : 0 < t < T\}$.

- $\mathcal{A}$: the ensemble of arcs. $\mathcal{A}^{tr}$ are the passenger travelling arcs i.e. moving arcs ($i_1 \neq i_2$) and idle arcs at stations. **Our model assumes that passengers do not use idle arcs at locations other than stations in the time-space network**. However, vehicles can wait: they can use those idle arcs (hence, the difference between the vehicle flow y that uses each a $\in \mathcal{A}$ and the passenger flow f that uses only $\mathcal{A}^{tr}$ - see below).
- For a given node n, we denote respectively the entering and the exiting arcs by $\mathcal{A}_n^+$ and $\mathcal{A}_n^-$. We also use $\mathcal{A}_n^{\pm,tr} = \mathcal{A}_n^{\pm} \cap \mathcal{A}^{tr}$.
- For a given arc $a = ((l_1, t_1), (l_2, t_2))$, we use $\|a\|$ to denote the Euclidean distance between locations l_1 and l_2.
- For a given node $n = (l, t)$, t_n denotes t.
- $\mathcal{I}_k$ is the k^{th} zone (a zone is an ensemble of locations). For each zone, one location is considered as the "center", denoted by center(k) ; it is typically a source for the on-demand vehicle flows.
- $\mathcal{I} = \{\mathcal{I}_1 ...\}$
- For an arc $a = (n_1, n_2)$, start(a) and end(a) denote respectively n_1 and n_2. If $n_1 = (l_1, t_1)$, $\mathcal{I}(a)$ and $\tau(a)$ denote respectively the zone of l_1 and t_1.

- **Model variables**
  - $y_a^{\text{OnD}} \in \mathbb{Z}_+$: flow of on-demand vehicles along arc $a \in \mathcal{A}$.
  - $f_{p,a} \in \{0, 1\}$: on-demand flow of passenger p along arc $a \in \mathcal{A}^{tr}$.
  - $x_{p,r} \in \{0, 1\}$: selects which route $r \in \mathcal{R}_p$ passenger p takes.
  - $z_{p,n}^O \in \{0, 1\}$: selects the node where $n \in \mathcal{N}_p^O$ passenger p is picked up by a on-demand vehicle (if necessary) **at the passenger's origin location**.
  - $z_{p,n}^D \in \{0, 1\}$: selects the node where $n \in \mathcal{N}_p^O$ passenger p is dropped off by a on-demand vehicle (if necessary) **at the passenger's destination location**.

- **Parameters**
  - $walking\_speed$, provided that on-demand and transit vehicles have a speed of 1.
  - K is the capacity of on-demand vehicles.
  - w_k is the number of initial vehicles in zone k.
  - ϵ is the stability coefficient for on-demand vehicles within a zone.

- Costs:
  - $C^{\text{pass}}_{p,a}$ is the cost for passenger p to use arc $a \in \mathcal{A}^{tr}$ in the time-space network. Typically, it could be a moving cost for non-idle arcs and waiting cost at station for idle arcs.
  - C^{OnD}_{a} is the cost for vehicles to use arc a in the time-space network. Typically, it could be gas and wear-and-tear costs for non-idle arcs and waiting cost for idle arcs.
  - $C^{\text{route}}_{p,r}$ is the cost for passenger p to take route r. Typically, it involves walking, transfer, waiting "out of the car" costs. For type 1 and 3 routes, it can also encapsulate time-related costs of arrival time.
  - $C^{\text{node}}_{p,n}$ is the cost for passenger p to be picked up at origin / dropped off at destination at node n. Typically, it is time-related and penalizes waiting before being picked up by a van or late-arrival for type 2, 4 and 5.

3.3 Operational Model

Constraint (3.1) enables each customer to be assigned to a single route. (3.2) and (3.3) guarantees respectively that for each passenger, if a route of type 2, 3 or 5 is chosen, we assign a origin node in the time-space network where a vehicle picks up the customer and if a route of type 2, 4 and 5 is chosen, we assign a destination node where a vehicle drops off the passenger.

(3.4) is a flow-balance consraint in the time-space network, for each passenger. If a route of type 3 or 5 is chosen, the boarding node of the route "absorbs" the passenger flow ; the same reasoning applies for the alighting node.

(3.5) is a capacity constraint that links f and y: on a given travelling arc in the on-demand time-space network, there can not be more passengers than the total on-demand capacity available on this arc.

(3.6) is a flow-balance constraint for vehicles, excluding *sources* at $t = 0$ and *sinks* at $t = T$. (3.7) and (3.8) state that for each zone, the exact chosen number of vehicles must start and end at the center. (3.9) is a *stability constraint*: we must maintain a certain proportion of the initial vehicles in each zone.

$$\min \sum_{p\in\mathcal{P}} \left\{ \sum_{r\in\mathcal{R}_p} C^{\text{route}}_{p,r} x_{p,r} + \sum_{n\in\mathcal{N}^O_p} C^{\text{node}}_{p,n} z^O_{p,n} + \sum_{n\in\mathcal{N}^D_p} C^{\text{node}}_{p,n} z^D_{p,n} + \sum_{a\in\mathcal{A}^{tr}} C^{\text{pass}}_{p,a} f_{p,a} + \right\} + \sum_{a\in\mathcal{A}} C^{\text{OnD}}_a y^{\text{OnD}}_a$$

s.t.

$$\sum_{r\in\mathcal{R}_p} x_{p,r} = 1,\ \forall p \in \mathcal{P} \tag{3.1}$$

$$\sum_{n\in\mathcal{N}^O_p} z^O_{p,n} = \sum_{r\in\mathcal{R}^2_p} x_{p,r} + \sum_{r\in\mathcal{R}^3_p} x_{p,r} + \sum_{r\in\mathcal{R}^5_p} x_{p,r},\ \forall p \in \mathcal{P} \tag{3.2}$$

$$\sum_{n\in\mathcal{N}^D_p} z^D_{p,n} = \sum_{r\in\mathcal{R}^2_p} x_{p,r} + \sum_{r\in\mathcal{R}^4_p} x_{p,r} + \sum_{r\in\mathcal{R}^5_p} x_{p,r},\ \forall p \in \mathcal{P} \tag{3.3}$$

$$\sum_{a\in\mathcal{A}^{+,tr}_n} f_{p,a} - \sum_{a\in\mathcal{A}^{-,tr}_n} f_{p,a} = \begin{cases} +z^O_{p,n} & \text{if } n \in \mathcal{N}^O_p \\ -\sum\limits_{\substack{r\in R^3_p\cup R^5_p \\ B(r)=n}} x_{p,r} & \text{if } n \in \mathcal{N}^B_p \\ \sum\limits_{\substack{r\in R^4_p\cup R^5_p \\ A(r)=n}} x_{p,r} & \text{if } n \in \mathcal{N}^A_p \\ -z^D_{p,n} & \text{if } n \in \mathcal{N}^D_p \\ 0 & \text{otherwise} \end{cases} \quad \forall p \in \mathcal{P},\ \forall n \in \mathcal{N} \tag{3.4}$$

$$\sum_{p\in\mathcal{P}} f_{p,a} \le K y^{\text{OnD}}_a,\ \forall a \in \mathcal{A}^{tr} \tag{3.5}$$

$$\sum_{a\in\mathcal{A}^+_n} y^{\text{OnD}}_a - \sum_{a\in\mathcal{A}^-_n} y^{\text{OnD}}_a = 0\ \forall n \in \mathcal{N}_0 \tag{3.6}$$

$$\sum_{a\in\mathcal{A}^+_{(\text{center}(k),0)}} y^{\text{OnD}}_a = w_k,\ \forall k \in \{1,...,|\mathcal{I}|\} \tag{3.7}$$

$$\sum_{a\in\mathcal{A}^-_{(\text{center}(k),T)}} y^{\text{OnD}}_a = w_k,\ \forall k \in \{1,...,|\mathcal{I}|\} \tag{3.8}$$

$$\sum_{i\in\mathcal{I}_k} \sum_{a\in\mathcal{A}^+_{(i,t)}} y^{\text{OnD}}_a \ge w_k(1-\varepsilon),\ \forall k \in \{1,...,|\mathcal{I}|\},\ \forall t > 0 \tag{3.9}$$

$$f_{p,a} \in \{0,1\},\ \forall p \in \mathcal{P},\ \forall a \in \mathcal{A}^{tr} \tag{3.10}$$

$$y^{\text{OnD}} \ge 0 \tag{3.11}$$

Once again, we remind that $\forall p,\ \sum\limits_{r\in\mathcal{R}^2_p} x_{p,r} = x_{p,(\emptyset,2)}$.

3.4 Two stages: Design-Operational model

Previously (3.1), we have considered the operated transit lines and the number of on-demand vehicles as totally exogenous.
We can also add *stage 0* variables, that we will refer to as the **network design stage**.

More precisely:

- We introduce the decision variable $\forall a \in \textit{transit\_arcs}, y_a^{\text{transit}} \in \{0, 1\}$ that indicates whether a transit arc (i.e. a given line between two stations at a given time) is "operated" or no
- The initial number of on-demand vehicles per zone w_k, which was previously an exogenous parameter, becomes here a variable

And we add the following constraint - a passenger can be assigned to a route only if the corresponding transit arc is operated:

$$x_{p,(a,\text{type})} \leq y_a^{\text{transit}} \; \forall p, \forall r = (a, \text{type}) \in \mathcal{R}_p \tag{3.12}$$

Adding to the cost function the costs of operating an arc and to input the vehicles, we obtain a natural setting for a Benders decomposition - as Mahéo, Kilby, and Hentenryck (2019) - to completely design and then operate a microtransit network. **In this work, we will only focus on the initial model, the operational stage**. Nonetheless, we propose in the next section a Benders decomposition dividing this operational stage into two substages.

OPERATIONAL STAGE: BENDERS DECOMPOSITION AND ACCELERATION

4.1 Outline: subproblem, dual and master problem

We decompose our operational model into two stages. x, z^O and z^D become *stage 1* (**assignment**) variables; y and f are the *stage 2* variables (**flow**).

Subproblem

Here, we consider x, z^O and z^D as exogenous. The subproblem boils down to a min-cost multiflow problem on f (passengers) and y (vehicles). Thanks to the integrality of the first-stage variables, the subproblem is unimodular and the solution is integral.

$$\textbf{(SP):} \qquad \min \sum_{p\in\mathcal{P}} \sum_{a\in\mathcal{A}^{tr}} C^{\text{pass}}_{p,a} f_{p,a} + \sum_{a\in\mathcal{A}} C^{\text{OnD}}_{a} y^{\text{OnD}}_{a}$$

$$\textit{s.t.} \qquad \sum_{a\in\mathcal{A}^{+,tr}_n} f_{p,a} - \sum_{a\in\mathcal{A}^{-,tr}_n} f_{p,a} = \begin{cases} +z^O_{p,n} & \text{if } n \in \mathcal{N}^O_p \\ -\sum\limits_{\substack{r\in R^3_p\cup R^5_p \\ B(r)=n}} x_{p,r} & \text{if } n \in \mathcal{N}^B_p \\ \sum\limits_{\substack{r\in R^4_p\cup R^5_p \\ A(r)=n}} x_{p,r} & \text{if } n \in \mathcal{N}^A_p \\ -z^D_{p,n} & \text{if } n \in \mathcal{N}^D_p \\ 0 & \text{otherwise} \end{cases} \qquad \forall p \in \mathcal{P},\ \forall n \in \mathcal{N} \quad (4.1)$$

$$K y_a^{\text{OnD}} - \sum_{p \in \mathcal{P}} f_{p,a} \geq 0, \ \forall a \in \mathcal{A}^{tr} \tag{4.2}$$

$$\sum_{a \in \mathcal{A}_n^+} y_a^{\text{OnD}} - \sum_{a \in \mathcal{A}_n^-} y_a^{\text{OnD}} = 0 \, , \forall n \in \mathcal{N}_0 \tag{4.3}$$

$$\sum_{a \in \mathcal{A}^+_{(center(k),0)}} y_a^{\text{OnD}} = w_k, \ \forall k \in \{1, ..., |\mathcal{I}|\} \tag{4.4}$$

$$\sum_{i \in \mathcal{I}_k} \sum_{a \in \mathcal{A}^+_{(i,t)}} y_a^{\text{OnD}} \geq w_k(1 - \varepsilon), \ \forall k \in \{1, ..., |\mathcal{I}|\} \, , \ \forall t > 0 \tag{4.5}$$

$$- f_{p,a} \geq -1, \ \forall p \in \mathcal{P}, \ \forall a \in \mathcal{A}^{tr} \tag{4.6}$$

$$f, y \geq 0 \tag{4.7}$$

Dual of the subproblem

We introduce the following Lagrangian variables:

- $\forall p \in \mathcal{P}, \forall n \in \mathcal{N}, \ \alpha_{p,n} \in \mathbb{R}$ (constraint (4.1))
- $\forall a \in \mathcal{A}^{tr}, \ \beta_a \in \mathbb{R}_+$ (constraint (4.2))
- $\forall n \in \mathcal{N}_0, \lambda_n \in \mathbb{R}$ (constraint (4.3))
- $\forall k \in \{1, .., |\mathcal{I}|\}, \ \gamma_{k,0} \in \mathbb{R}$ (constraint (4.4))
- $\forall k \in \{1, .., |\mathcal{I}|\}, \forall 1 \leq t \leq T - 1, \ \gamma_{k,t} \in \mathbb{R}_+$ (constraint (4.5))
- $\forall p \in \mathcal{P}, \forall a \in \mathcal{A}^{tr}, \ \xi_{p,a} \in \mathbb{R}_+$ (constraint (4.6))

We obtain the dual of the subproblem:

(DualSP):

$$\max_{\alpha,\beta,\lambda,\gamma,\xi} \sum_{p\in\mathcal{P}} \left\{ \sum_{n\in\mathcal{N}_p^O} \alpha_{p,n} z_{p,n}^O - \sum_{n\in\mathcal{N}_p^B} \alpha_{p,n} \sum_{\substack{r\in R_p^3\cup R_p^5 \\ B(r)=n}} x_{p,r} + \sum_{n\in\mathcal{N}_p^A} \alpha_{p,n} \sum_{\substack{r\in R_p^4\cup R_p^5 \\ A(r)=n}} x_{p,r} - \sum_{n\in\mathcal{N}_p^D} \alpha_{p,n} z_{p,n}^D - \sum_{a\in\mathcal{A}^{tr}} \xi_{p,a} \right\} + \sum_{k=1}^{|\mathcal{I}|} w_k \left[\gamma_{k,0} + \sum_{t=1}^{T-1} \gamma_{k,t}(1-\epsilon) \right]$$

$$\begin{aligned}
s.t. \quad & \alpha_{p,\text{start}(a)} - \alpha_{p,\text{end}(a)} - \beta_a - \xi_{p,a} \le C_{p,a}^{\text{pass}}, \quad \forall p \in \mathcal{P}, \forall a \in \mathcal{A}^{tr} && (4.8)\\
& K\beta_a \mathbb{1}_{a\in\mathcal{A}^{tr}} + \lambda_{\text{start}(a)} \mathbb{1}_{\text{start}(a)\in\mathcal{N}_0} - \lambda_{\text{end}(a)} \mathbb{1}_{\text{end}(a)\in\mathcal{N}_0} + \gamma_{\mathcal{I}(a)\tau(a)} \le C_a^{\text{OnD}}, \ \forall a \in \mathcal{A} && (4.9)\\
& \gamma_{k,t} \ge 0 \ \ \forall k, \forall 1 \le t \le T-1 && (4.10)\\
& \beta, \xi \ge 0 && (4.11)
\end{aligned}$$

Master problem

Eventually, the Master Problem is as follows:

$$\textbf{(MP)} \qquad \min_{x,z^O,z^D} \sum_{p\in\mathcal{P}} \left\{ \sum_{r\in\mathcal{R}_p} C_{p,r}^{route} x_{p,r} + \sum_{n\in\mathcal{N}_p^O} C_{p,n}^{node} z_{p,n}^O + \sum_{n\in\mathcal{N}_p^D} C_{p,n}^{node} z_{p,n}^D \right\} + \Theta$$

$$\begin{aligned}
s.t. \quad & \sum_{r\in\mathcal{R}_p} x_{p,r} = 1, \ \forall p \in \mathcal{P} && (4.12)\\
& \sum_{n\in\mathcal{N}_p^O} z_{p,n}^O = \sum_{r\in\mathcal{R}_p^2} x_{p,r} + \sum_{r\in\mathcal{R}_p^3} x_{p,r} + \sum_{r\in\mathcal{R}_p^5} x_{p,r}, \ \forall p \in \mathcal{P} && (4.13)\\
& \sum_{n\in\mathcal{N}_p^D} z_{p,n}^D = \sum_{r\in\mathcal{R}_p^2} x_{p,r} + \sum_{r\in\mathcal{R}_p^4} x_{p,r} + \sum_{r\in\mathcal{R}_p^5} x_{p,r}, \ \forall p \in \mathcal{P} && (4.14)\\
& z_{p,n}^O x_{p,r} (t_{B(r)} - t_n - d_{o(p),s(r)}) \ge 0, \ \ \forall p \in \mathcal{P}, \forall n \in \mathcal{N}_p^O, \forall r \in \mathcal{R}_p^3 \cup \mathcal{R}_p^5 && (4.15)\\
& z_{p,n}^D x_{p,r} (t_n - t_{A(r)} - d_{a(r),d(p)}) \ge 0, \ \ \forall p \in \mathcal{P}, \forall n \in \mathcal{N}_p^D, \forall r \in \mathcal{R}_p^4 \cup \mathcal{R}_p^5 && (4.16)\\
& \Theta \ge \text{DualSP}(x, z^O, z^D, \alpha^{(i)}, \beta^{(i)}, \lambda^{(i)}, \gamma^{(i)}, \xi^{(i)}), \forall i \in I && (4.17)\\
& \text{DualSP}(x, z^O, z^D, \alpha^{(j)}, \beta^{(j)}, \lambda^{(j)}, \gamma^{(j)}, \xi^{(j)}) \le 0, \forall j \in J && (4.18)\\
& x, z^O, z^D \in \{0,1\} && (4.19)
\end{aligned}$$

where:

- I and J index respectively the extreme points and the extreme rays of the DualSP polyedra
- $(x, z^O, z^D) \mapsto \text{DualSP}(x, z^O, z^D, \alpha, \beta, \lambda, \gamma)$ is a linear function of (x, z^O, z^D)) which value is the solution of DualSP for the corresponding $\alpha, \beta, \lambda, \gamma$

Note that constraints (4.15) and (4.16) were not needed in the global problem - as they are satisfied by the optimal solution - and are specifically added here, to avoid trivial subproblem infeasibilities. Basically, we say that, for each passenger, when we choose a type 3 (First-Mile) or type 5 (Last-Mile) route, and a certain origin node (we recall that the origin node is the pair location-time where the vehicle picks the passenger up at their origin), we must put enough time between this origin node and the boarding node involved by the route to drive between the origin and the involved station. Similarly, we say that for each passenger, when we choose a route of type 4 (Last-Mile) or type 5, we also need to allow enough time to travel between the alighting station involved and the destination node chosen. These basic considerations enable to improve the feasibility of the subproblem.

4.2 Benders algorithm (Benders, 1962)

In algorithm 2, we will denote as $(MP)_{I,J}$ the Master Problem (4.1) with given sets I and J for the constraints (4.17) and (4.18) respectively.

For $\epsilon = 0$, or very small, we obtain at the last iteration the exact optimal solution for the Master Problem, and we can easily deduce the global optimal solution (if we used the primal subproblem, we already have it using the solutions of the last iteration; otherwise we solve the subproblem using the last (optimal) value of $x, z^O \text{and} z^D$).

We note that as an ILP and with reasonable parameters (especially, enough vehicles), the model is feasible, so we can safely say that the Master Problem is feasible at each iteration, and that the primal subproblem is never unbounded. If the MP happens to be infeasible at a given iteration, we conclude that the model is infeasible. Convergence is guaranteed in a finite number of iterations as the number of possible cuts is finite (Gaubert and Bonnans, 2020).

Algorithm 2 Benders Algorithm

Require: $\epsilon \geq 0$
  upper_bound $\leftarrow +\infty$
  lower_bound $\leftarrow 0$
  $I \leftarrow \emptyset$
  $J \leftarrow \emptyset$
  $i \leftarrow 0$
  while upper_bound − lower_bound $> \epsilon$ **do**
    Solve $(MP)_{I,J}$ and compute solutions $x^i, z^{O,i}, z^{D,i}, \theta^i$
    lower_bound $\leftarrow$ value($(MP)_{I,J}$)
    Solve the subproblem (primal or dual) - using $x^i, z^{O,i}, z^{D,i}$ as input
    if Subproblem is feasible **then**
      Compute dual solutions $\alpha^{(i)}, \beta^{(i)}, \lambda^{(i)}, \gamma^{(i)}, \xi^{(i)}$
      Add **optimality cut**: I.add(i)
      upper_bound $\leftarrow$ lower_bound + value(DualSP) - θ
    else if Primal Subproblem is infeasible / Dual Subproblem is unbounded **then**
      Compute an unbounded dual extreme ray $\alpha^{(i)}, \beta^{(i)}, \lambda^{(i)}, \gamma^{(i)}, \xi^{(i)}$
      Add **feasibility cut**: J.add(i)
      upper_bound $\leftarrow +\infty$
    end if
    $i \leftarrow i + 1$
  end while

4.3 Pareto-optimal cuts

One acceleration technique mainly used for the Benders decomposition is using Pareto-optimal cuts (Magnanti and Wong, 1981).

Algorithm 3 Benders Algorithm with Pareto-optimal cuts

Require: $\epsilon \geq 0$
  upper_bound $\leftarrow +\infty$
  lower_bound $\leftarrow 0$
  $I \leftarrow \emptyset$
  $J \leftarrow \emptyset$
  $i \leftarrow 0$
  while upper_bound − lower_bound $> \epsilon$ **do**
    Solve $(MP)_{I,J}$ and compute solutions $x^i, z^{O,i}, z^{D,i}, \theta^i$
    lower_bound $\leftarrow$ value($(MP)_{I,J}$)
    Solve the subproblem (primal or dual) - using $x^i, z^{O,i}, z^{D,i}$ as input
    if Subproblem is feasible **then**
      Compute SP_value^i, the solution value of the subproblem
      Solve **(Pareto)**$_i$ and compute solutions $\alpha^{(i)}, \beta^{(i)}, \lambda^{(i)}, \gamma^{(i)}, \xi^{(i)}$
      Add **optimality cut**: I.add(i)
      upper_bound $\leftarrow$ lower_bound + value(DualSP) - θ
    else if Primal Subproblem is infeasible / Dual Subproblem is unbounded **then**
      Compute an unbounded dual extreme ray $\alpha^{(i)}, \beta^{(i)}, \lambda^{(i)}, \gamma^{(i)}, \xi^{(i)}$
      Add **feasibility cut**: J.add(i)
      upper_bound $\leftarrow +\infty$
    end if
    $i \leftarrow i + 1$
  end while

where $(\textbf{Pareto})_i$:

$$\max_{\alpha,\beta,\lambda,\gamma,\xi} \sum_{p\in\mathcal{P}} \left\{ \sum_{n\in\mathcal{N}_p^O} \alpha_{p,n}\frac{1}{2} - \sum_{n\in\mathcal{N}_p^B} \alpha_{p,n} \sum_{\substack{r\in R_p^3\cup R_p^5 \\ B(r)=n}} \frac{1}{2} + \sum_{n\in\mathcal{N}_p^A} \alpha_{p,n} \sum_{\substack{r\in R_p^4\cup R_p^5 \\ A(r)=n}} \frac{1}{2} - \sum_{n\in\mathcal{N}_p^D} \alpha_{p,n}\frac{1}{2} - \sum_{a\in\mathcal{A}^{tr}} \xi_{p,a} \right\}$$
$$+ \sum_{k=1}^{|\mathcal{I}|} w_k \left[\gamma_{k,0} + \sum_{t=1}^{T-1} \gamma_{k,t}(1-\epsilon) \right]$$

$$\text{s.t.} \quad \alpha_{p,\text{start}(a)} - \alpha_{p,\text{end}(a)} - \beta_a - \xi_{p,a} \le C_{p,a}^{\text{pass}}, \quad \forall p \in \mathcal{P}, \forall a \in \mathcal{A}^{tr}$$

$$K\beta_a \mathbb{1}_{a\in\mathcal{A}^{tr}} + \lambda_{\text{start}(a)}\mathbb{1}_{\text{start}(a)\in\mathcal{N}_0} - \lambda_{\text{end}(a)}\mathbb{1}_{\text{end}(a)\in\mathcal{N}_0} + \gamma_{\mathcal{I}(a)\tau(a)} \le C_a^{\text{OnD}}, \quad \forall a \in \mathcal{A}$$

$$\gamma_{k,t} \ge 0, \quad \forall k, \forall 1 \le t \le T-1$$

$$\beta, \xi \ge 0$$

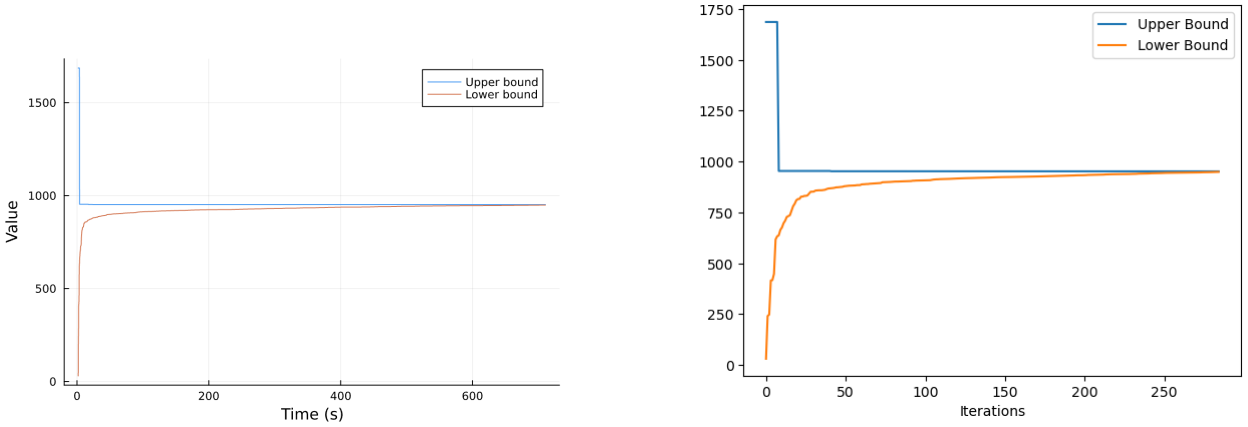
$$\sum_{p\in\mathcal{P}} \left\{ \sum_{n\in\mathcal{N}_p^O} \alpha_{p,n} z_{p,n}^{O,i} - \sum_{n\in\mathcal{N}_p^B} \alpha_{p,n} \sum_{\substack{r\in R_p^3\cup R_p^5 \\ B(r)=n}} x_{p,r}^i + \sum_{n\in\mathcal{N}_p^A} \alpha_{p,n} \sum_{\substack{r\in R_p^4\cup R_p^5 \\ A(r)=n}} x_{p,r}^i - \sum_{n\in\mathcal{N}_p^D} \alpha_{p,n} z_{p,n}^{D,i} - \sum_{a\in\mathcal{A}^{tr}} \xi_{p,a} \right\}$$
$$+ \sum_{k=1}^{|\mathcal{I}|} w_k \left[\gamma_{k,0} + \sum_{t=1}^{T-1} \gamma_{k,t}(1-\epsilon) \right] = \text{SP\_value}^i$$

We use here as a core point the vector that is valued 0.5 for each coordinate.

4.4 Computational results

In the figure 4.1, we present an example of a Benders decomposition convergence in a Cross Setting (see 5.1) with $\Delta = 20$ and $R = 5$, with 1 customer per zone, one on-demand vehicle per zone initially, transit frrequence of 10, and medium reluctance to walk. Other parameters are similar to what is done in the next section.

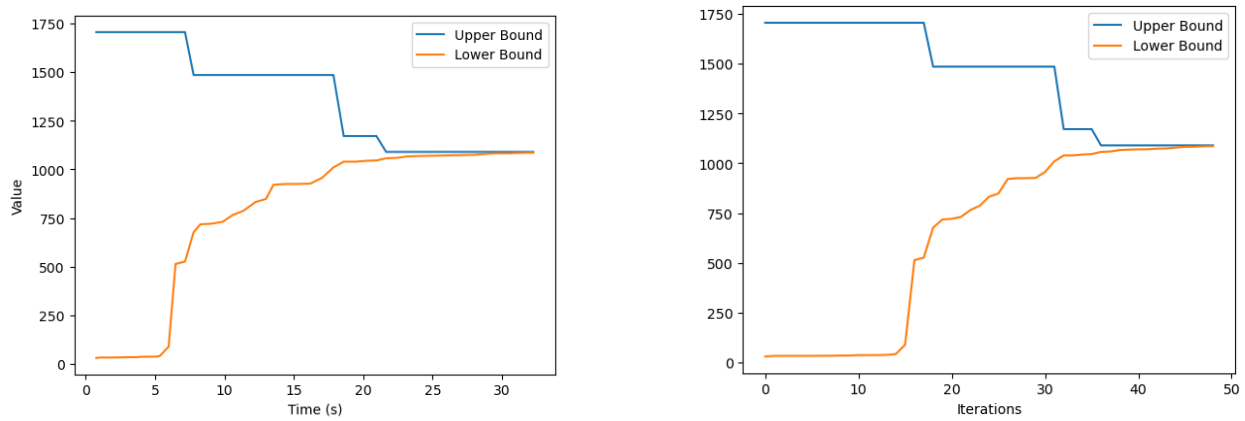
The convergence happens, with a solution that equals the one given by the Mixed-Integer Linear Programming optimizer. However, it is very long: more than 250 iterations and more than 10 minutes, when the MILP optimizer gives it in less than 1 minute. Both feasibility and optimality cuts



(a) Bounds evolution versus time

(b) Bounds evolution versus iterations

Figure 4.1: Benders decomposition



(a) Bounds evolution versus time

(b) Bounds evolution versus iterations

Figure 4.2: Benders decomposition with Pareto optimal cuts

are weak: this motivates the implementation of Pareto-optimal cuts to strengthen the optimality cuts.

The figure 4.2 shows how efficient are the Pareto-optimal cuts, enabling us to converge in approximately 50 seconds and less than 50 iterations. We have incorporated the time needed to solve $(Pareto)_i$.

Table 4.1 summarizes our computational results. For each total number of customers, we generate a setting and we run the methods on the same setting. We set a *Time Limit* of 2000s. **X** refers to an **infeasibility deadlock** where infeasibility cuts do not have any effect on the Master Problem's

Total num. cust.	Gurobi - MIP	Benders	Pareto-optimal Benders
$n = 4$	9s	696s	34s
$n = 12$	148s	**X**	532s
$n = 20$	1265s	*Time Limit*	**X**

Table 4.1: Summary of computational results

solution value (the solutions change, however) and the subproblem stays infeasible *ad eternam* - at least until the time limit is reached.

All in all, the Benders decomposition, even accelerated, has not proven fruitful at this stage: we kept using the Gurobi MIP optimizer (Gurobi Optimization, LLC, 2023). We are not able as for now to scale this exponential-sized problem.

OPERATIONAL STAGE: EXPERIMENTAL RESULTS ON BASIC SETTINGS

5.1 Settings

In order to test the behavior of our model and conduct experiments, we designed "settings": typical ways of distributing origins, destinations and transit lines. These are described below.

Linear Expansion

Given a total number of customers num_cust, the process to generate a Linear Expansion setting is as follow:

- We set three transit stations on a straight line (*West*, *Central* and *East*). The distance between West and Central and between Central and East are parameters, respectively denoted as Δ_1 and Δ_2.
- Around each of the three stations, we consider a circle, which radiuses R_1, R_2, R_3 are parameters
  - Within zone 1 (around West), we **simulate uniformly** $\lfloor \frac{2}{3} num\_cust \rfloor$ **origin points**
  - Within zone 2 (around Central), we simulate uniformly $\lceil \frac{1}{3} num\_cust \rceil$ origin points and $\lceil \frac{1}{3} num\_cust \rceil$ **destination points**
  - Finally, within zone 3 (around East), we simulated uniformly $\lfloor \frac{2}{3} num\_cust \rfloor$ destination points
- Eventually, we **randomly** assign each origin to a destination: we then have num_cust origin-destination pairs, as wished.

To simplify the experiments we will make the following assumptions. We assume that the transit line goes only in one direction, from West to East, and makes a stop at Central. There is one transit vehicle leaving each *freq* time steps, *freq* being a parameter. We also assume that $\Delta_1 = \Delta_2$. The radiuses of each zone are equal. The center of each zone (starting point for each vehicle) are the stations.

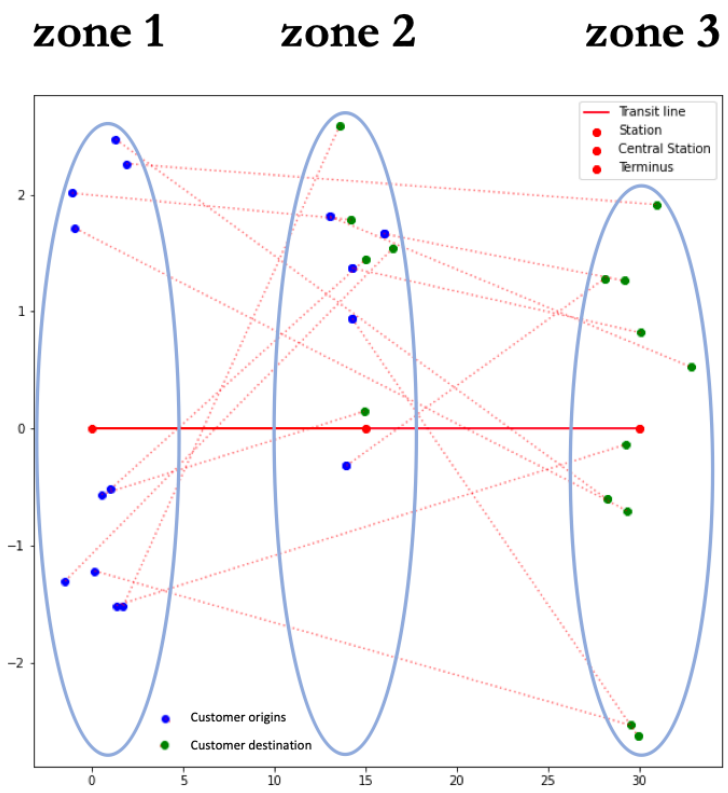


Figure 5.1: The Linear Expansion setting

Cross Setting

In that setting, we generate five stations: *North*, *South*, *West*, *East* and *Central*.

One transit line joins North and South (distance Δ_1) and another one joins West and East (distance Δ_2). **Both lines operate in both directions**. These two lines cross at Central, where **transfers are available**.

Each station is the center of a zone. The radiuses are respectively denoted as R_1, R_2, R_3, R_4, R_5 (1: North, 2: South, 3: East, 4: West, 5: Central).

The number of customers initially present in each zone are denoted as S_1, S_2, S_3, S_4 - we assume that there is no customer in the Central zone. For each zone i:

- We generate S_i origin points uniformly

- For each origin point:
  - We choose randomly another zone $j \neq i, j \in \{1, ..., 4\}$
  - We generate a destination point uniformly within zone j

In our experiments, we will assume that all the radiuses are equal to R, that $\Delta_1 = \Delta_2 (= \Delta)$ and that we generate the same number S of origin points in each zone. With these assumptions, we do not have any transfer waiting (both trains from each of the end arrive at Central exactly at the same time).

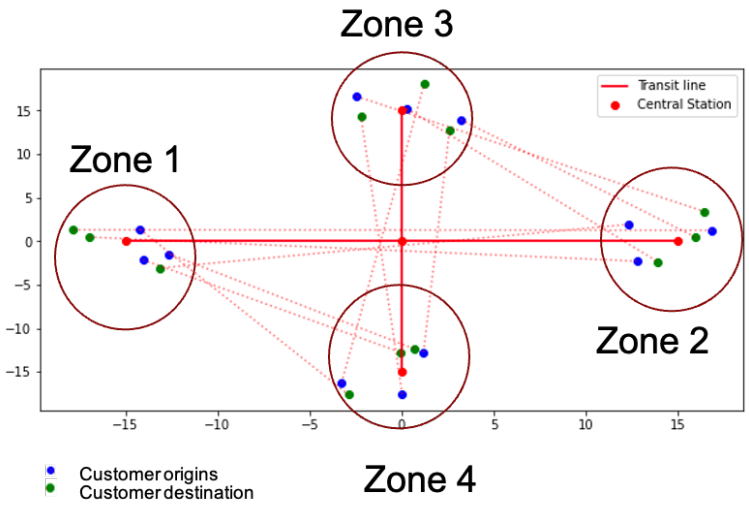


Figure 5.2: The Cross setting

5.2 Experimental process

For different sets of parameters, we will make vary Δ - the length of the transit lines / the distance between zones - and R, the dispersion of the customers within these zones.

We optimize and compute these following metrics to measure the impact of dispersion and distance:

- Total walking distance ($\sum_{p \in \mathcal{P}} \sum_{r \in \mathcal{R}_p} x_{p,r} \mathbf{W}(r)$)
- Total waiting time (sum of the waiting times for all the customers).
- Last Arrival Time
- Total distance covered by vehicles

The waiting time is computed as follow, for a given customer:

- If the route chosen is of type 1 (Full Transit), it is the transfer waiting time if applicable (i.e. if the transit arc chosen is a transfer arc and involves transfer waiting) and the difference between the boarding time and the walking time to station of the customer (the waiting on the platform).
- If the route chosen is of type 2 (Full On-Demand), we consider as "waiting" the delay between the departure time and the time customer is being picked up (basically, it is t_{n_o} where n_o is *the* node such as $z^O_{p,n_o} = 1$. We do not consider detours due to ride-sharing.
- If the route chosen is of type 3 or 5, we have again the delay between departure and pick-up, and we add transfer waiting if applicable.
- If the route chosen is of type 4, we only add potential transfer waiting to the delay.

We remind what we said in algorithm 1: when taking a hybrid mode, we forced that passengers are dropped off to stations just in time for boarding and are picked up at stations just exactly when they alight, without waiting. A way to relax this with a simple tweak in the model is available in 6.

As we want to approach a sort of *natural* ground truth, we want to mitigate the variability caused by the stochastic nature of the setting generation and decided to use Monte-Carlo approximations. For each couple (Δ, R), we generate **10** settings, run the optimizations and take the average to have the final metrics. The number of 10 Monte-Carlo iterations has been arbitrarily set due to the high computation time required.

5.3 Results and insights

For our experiments we use the following parameters:

- $\forall p, \forall r,\ C^{\text{route}}_{p,r} = walkingCost * \mathbf{W}(r)$. We only penalize walking: there is no costs for modal transfers (between on-demand vehicles and transit vehicles) or transit transfers.
- $\forall p, \forall n,\ C^{\text{node}}_{p,n} = t_n$
- $\forall p, \forall a, C^{\text{pass}}_{p,a} = \|a\|$
- $\forall a, C^{\text{OnD}}_{a} = \|a\|$
- $K = 4$

- walking_speed = 2
- $\epsilon = 1$: for our experiments, we did not include the stability of vehicles within a zone ; the vehicles can shift between zones freely.
- Departure times are randomly simulated between 1 and 5
- MAX_TRANSFER_WAITING = 1

We set the parameter walkingCost to 6 (medium reluctance to walk) ; other possible values are 3 (low reluctance) or 12 (high reluctance). The number of customer per zone is chosen between 1 and 3.

Linear Expansion

For the Linear Expansion setting, we made a little detour and tried to see the benefits of hybrid operation when comparing to the *statu quo* (the duopoly full transit or full on-demand).

The following table summarizes our results, for 15 total customers and medium reluctance to walk, $\Delta = 15$, $R = 5$:

	Full transit	**25% On-Demand**	**50% On-Demand**	**75% On-Demand**	**Full On-Demand**
Walk	**100**	55	40	19	**0**
Wait	**0**	**100**	57	57	25
On-demand distance	**0**	33	66	66	**100**
Last arrival time	55	41	41	38	36

Figure 5.3: Results of our experiments with Linear Expansion setting, in base 100

where:

- Full transit: Transit frequence 2, 0 On-Demand vehicle
- 25% On-Demand: Transit frequence 4, 1 On-Demand vehicle
- 50% On-Demand: Transit frequence 6, 2 On-Demand vehicles

- 75% On-Demand: Transit frequence 10, 4 On-Demand vehicles
- Full On-Demand: Transit frequence 100, 6 On-Demand vehicles

We verify that hybrid operations strike a balance between level of service (walking, waiting, LAT) and cost of operations, which give us an answer to the first question (1).

The above mentioned delta-radius analysis for the Linear Expansion can be found in Appendix (6).

Cross Setting

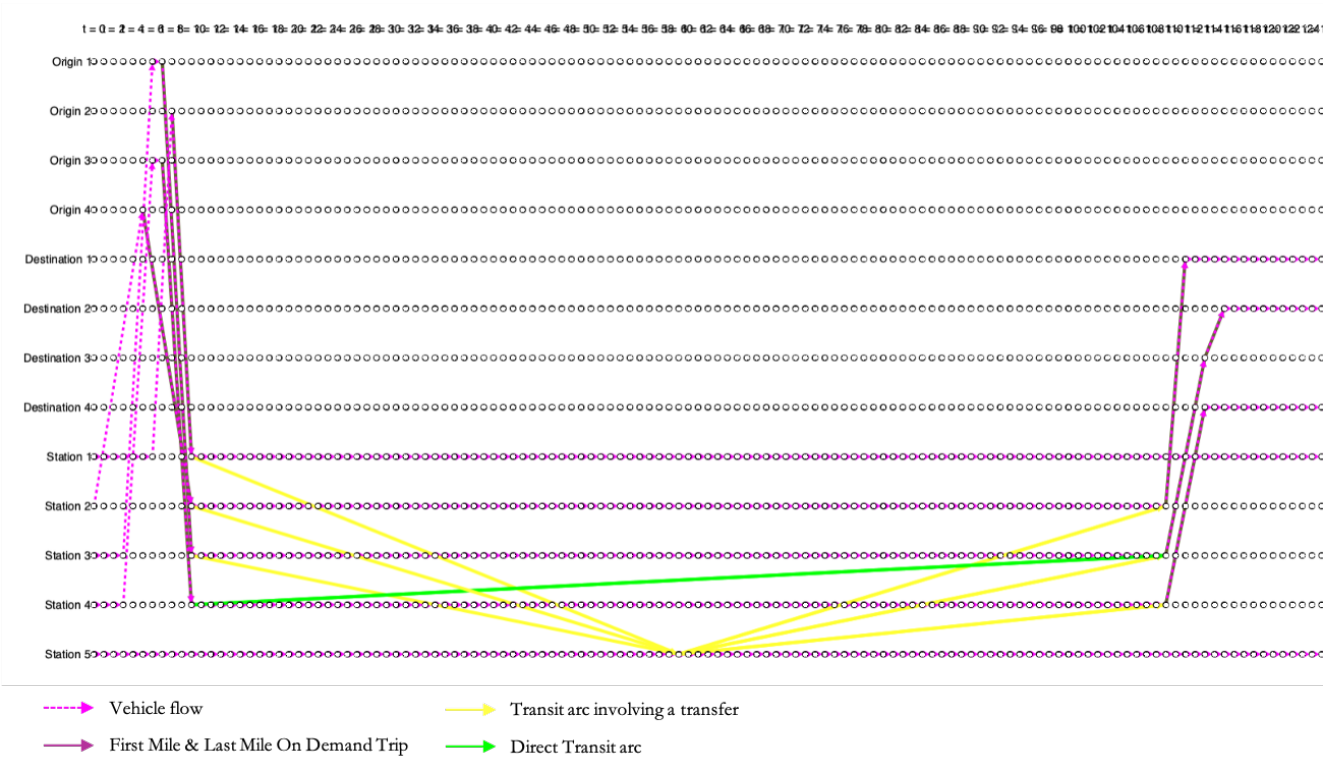


Figure 5.4: Example of results with the Cross Setting

The figure 5.4 shows us an example of results given by the optimization model, illustrated on the on-demand time-space network. There is 1 customer per zone (4 in total) and the 5 stations appear. The transit arcs (green for direct and yellow for those involving a transfer) have been reported here to simplify the visualization (instead of having two separate time-space networks). In this example, there is one vehicle starting in each station at $t = 0$.

In the following, we tried to evaluate the impact of progressively adding on-demand vehicles to a full transit system and to see how our KPIs fare.

Here, *freq* is kept constant at 10 (a transit vehicle leaves each station every 10 time steps), and we try to compare three situations:

- *Low capacity*: we add 1 on-demand vehicle
- *Medium capacity*: we add 2 on-demand vehicles
- *High capacity*: we add 4 on-demand vehicles

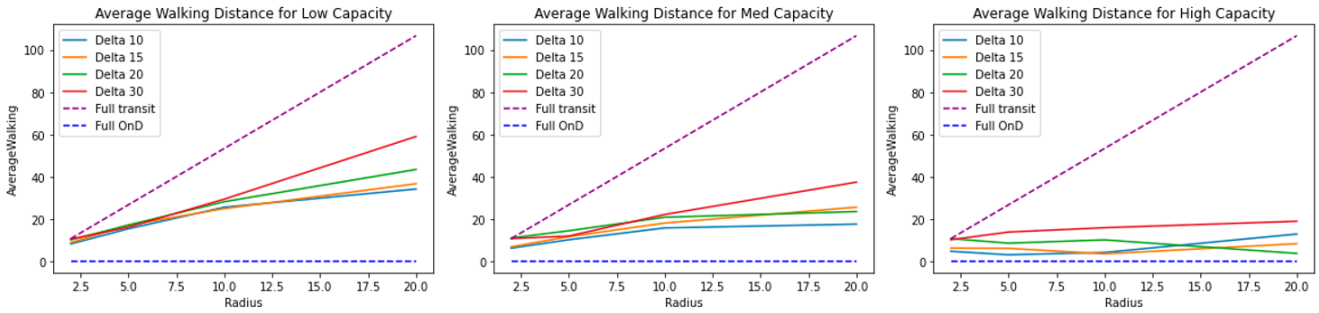


Figure 5.5: Walking distance analysis (1 customer per zone)

We note that in figure 5.5 the Full Transit curve is a theoretical bound. Given our simple setting where all the stations are connected and where customers do not discriminate between direct routes or routes involving transfers, in a full transit system, customers have to walk between their origin and the closest boarding station; they alight at the station closest to their destination and then have to walk from the alighting station to their destination. We recall that origins and destinations are uniformly simulated within a circle of radius R: the expectation of their distance to the centre is $\frac{2R}{3}$, i.e. each customer walks a distance of $\frac{4R}{3}$, and as we have 4 customers, the total walking distance is $\frac{16R}{3}$.

We observe that walking is drastically reduced, even when adding a single on-demand vehicle. In fact, figure 5.6 even shows that the **first vehicle** has the best marginal benefit: a single vehicle can suffice to achieve great performance.

The benefits of the model are even clearer when radiuses get bigger. For a high on-demand capacity, the curves are almost flat: even when customers are largely spread around stations, hybrid operations enable to drastically contain total walking.

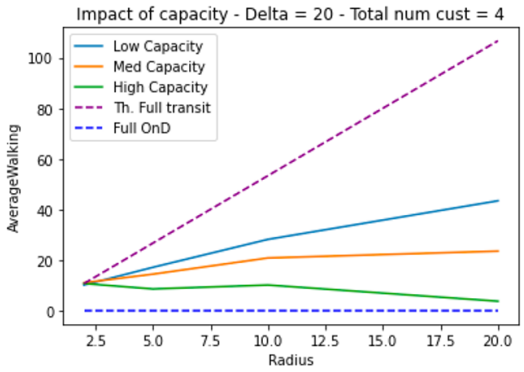


Figure 5.6: Walking distance analysis: marginal benefits of capacity

We now analyse the Latest Arrival Time metric.

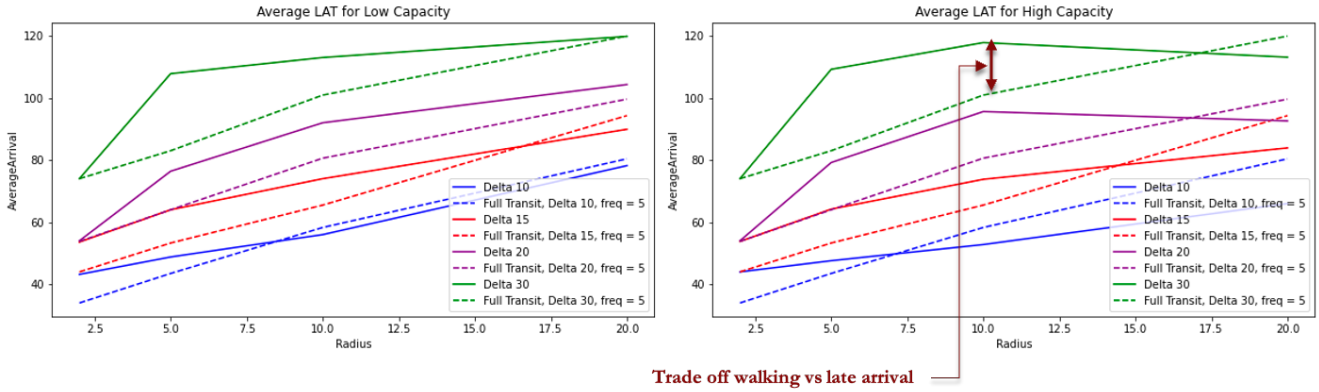


Figure 5.7: LAT analysis - 1 customer per zone

In figure 5.7, the full lines still represent our previous setting (frequence of 10 and respectively 1 and 4 on-demand vehicles added (low and high capacity)). However, we wanted to compare to a full transit model with **a doubled frequence** (frequence of 5 time steps): here, the full transit curves are empirical and are represented by the dotted lines.

We observe that hybrid operations start to beat a double-frequenced full transit model when radiuses are getting bigger i.e. when customers are spread around stations. The "beating point" is earlier when the on-demand capacity is higher.

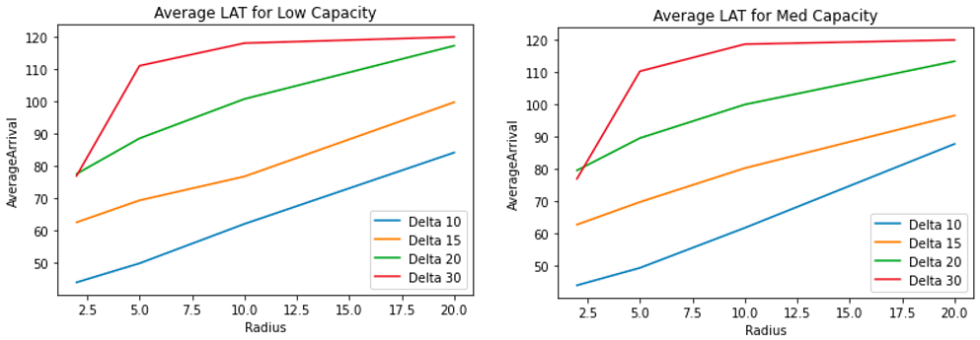


Figure 5.8: LAT analysis - 3 customers per zone

In figure 5.8, we increase the number of customers per zone to 3 (12 customers in total). We observe that thanks to **ride-sharing**, hybrid operations maintain the same level of service in terms of LAT. To be noted that in this setting, as radiuses and distances (Δ) are getting bigger, a full On-Demand system becomes very quickly infeasible (for a reasonable number of vehicles): hybrid operations enable to better handle customers' spread and large distances.

To summarize, figures 5.5, 5.6, 5.7 and 5.8 show us that hybrid operations enable to strike a great balance between the current Full Transit and Full On-Demand dichotomy:

- Full Transit systems do not have any infeasibility issues, even when customers are more spread, more numerous or when distances between stations are higher. But **hybrid operations drastically improve KPIs** such as walking or LAT, with limited additional costs, as we have seen that the first added on-demand vehicle has the best marginal benefit. Even better: when customers are largely spread, hybrid operations can do better than double-frequenced transit systems.
- Full On-Demand systems enable not to walk, and can achieve great LATs with limited waiting times when it makes no sense to take the transit lines (i.e. when customers are largely spread around stations). However, they are very costly: if one wants to curb the number of vehicles, the situations becomes quickly infeasible. Hybrid operations manage to limit the costs (problems are feasible even with reasonable number of vehicles) while maintaining great KPIs when customers are more numerous.

Insights

In the following, we put aside the metrics and leverage the versatility of the model that chooses, for a given situation, the best outcome between full transit, hybrid and on-demand. In a realistic

setting, with realistic parameters, **we want to answer the question: when do customers prefer transit, when do they prefer on-demand, and when do they prefer hybrid modes ?**

	Delta	Radius	AverageWaiting	AverageWalking	AverageLAT	Transit/OnD/Hybrid	PreprAvgTime	OptAvgTime
0	10	2	59.0	9.4	44.0	1.3/0.0/2.7	12.38	1.28
1	10	5	51.3	6.1	47.6	0.1/2.0/1.9	11.49	1.02
2	10	10	54.0	8.3	52.8	0.2/3.1/0.7	11.37	0.84
3	10	20	55.3	25.7	66.0	0.6/2.9/0.5	10.86	0.71
4	15	2	54.0	12.4	53.8	1.6/0.0/2.4	10.29	0.85
5	15	5	69.8	12.2	64.2	0.1/0.6/3.3	10.17	0.79
6	15	10	75.3	7.0	73.8	0.0/2.5/1.5	9.96	0.84
7	15	20	75.7	16.6	83.9	0.3/3.3/0.4	8.76	0.83
8	20	2	25.4	21.4	54.0	4.0/0.0/0.0	7.67	0.70
9	20	5	83.5	17.1	79.2	0.9/0.2/2.9	7.80	0.83
10	20	10	100.3	20.2	95.6	0.1/1.0/2.9	7.32	0.88
11	20	20	98.1	7.4	92.6	0.0/2.7/1.3	7.27	0.99
12	30	2	26.2	20.3	74.0	4.0/0.0/0.0	4.51	0.41
13	30	5	81.4	27.6	109.2	1.9/0.0/2.1	4.37	0.46
14	30	10	118.0	31.7	117.8	0.4/0.1/3.5	4.45	0.51
15	30	20	92.8	37.8	113.1	0.4/2.5/1.1	4.03	0.53

Figure 5.9: Result table: 1 customer per zone

	Delta	Radius	AverageWaiting	AverageWalking	AverageLAT	Transit/OnD/Hybrid	PreprAvgTime	OptAvgTime
0	10	2	196.2	20.6	44.0	0.0/0.0/12.0	198.11	22.59
1	10	5	156.4	41.6	49.4	1.9/4.3/5.8	193.24	33.56
2	10	10	148.8	93.9	61.8	4.3/5.4/2.3	185.11	35.91
3	10	20	168.0	153.5	87.8	5.4/6.3/0.3	172.84	32.19
4	15	2	227.4	22.0	62.8	0.3/0.0/11.7	182.00	17.28
5	15	5	238.1	43.8	69.8	0.1/0.3/11.6	177.64	30.52
6	15	10	207.5	89.7	80.3	2.9/4.8/4.3	159.51	101.60
7	15	20	221.0	123.2	96.6	3.3/6.6/2.1	145.89	184.98
8	20	2	273.7	28.3	79.6	2.9/0.0/9.1	127.60	27.21
9	20	5	330.5	45.3	89.6	0.2/0.2/11.6	128.80	20.04
10	20	10	278.9	91.8	100.0	2.0/2.1/7.9	128.55	88.35
11	20	20	227.5	176.6	113.4	3.9/5.6/2.5	118.26	89.60
12	30	2	87.5	61.5	77.0	11.7/0.0/0.3	76.94	9.10
13	30	5	337.1	58.2	110.3	2.1/0.0/9.9	76.65	13.51
14	30	10	278.8	119.2	118.7	3.1/1.8/7.1	69.86	30.81
15	30	20	236.8	251.4	120.0	3.8/3.3/4.9	67.15	16.03

Figure 5.10: Result table: 3 customers per zone

In figures 5.9 and 6.2, *PreprAvgTime* is the average (over the 10 Monte-Carlo iterations) time needed to do all the computation *before* the optimization itself (reading inputs, creating time-space

networks, rearranging nodes and arcs, preprocessing the routes...) and *OptAvgTime* is the average optimization time. The *Hybrid* category gathers Type 3, 4 and 5 routes. We have a transit frequence of 10 and medium On-Demand capacity (2 vehicles).

The following numbers directly refer to the red pointers on the figures.

1. Unsurprisingly, small radius and high Δ lead to a full-transit choice
2. Model prefers to avoid transit when radius is bigger than Δ
3. Higher radius leads to a better repartition between the types (i.e. «hybridness») (provided that we have enough vehicles)
4. Increasing the number of customers pushes the model to choose hybrid operations: thanks to ride-sharing, one vehicle riding cost is better than the addition of walking costs

We conclude by saying that the parameters do have an impact: a higher reluctance to walk will lead to the choice of a Type 5 (First-Mile Last-Mile) route rather than a full transit route, **provided that we have enough on-demand capacity**. The following figure 5.11 summarize the typical behaviours of the model: its versatility enables to choose the best outcome based on the setting, capacities and assumptions (reluctance to walk, eagerness to arrive...).

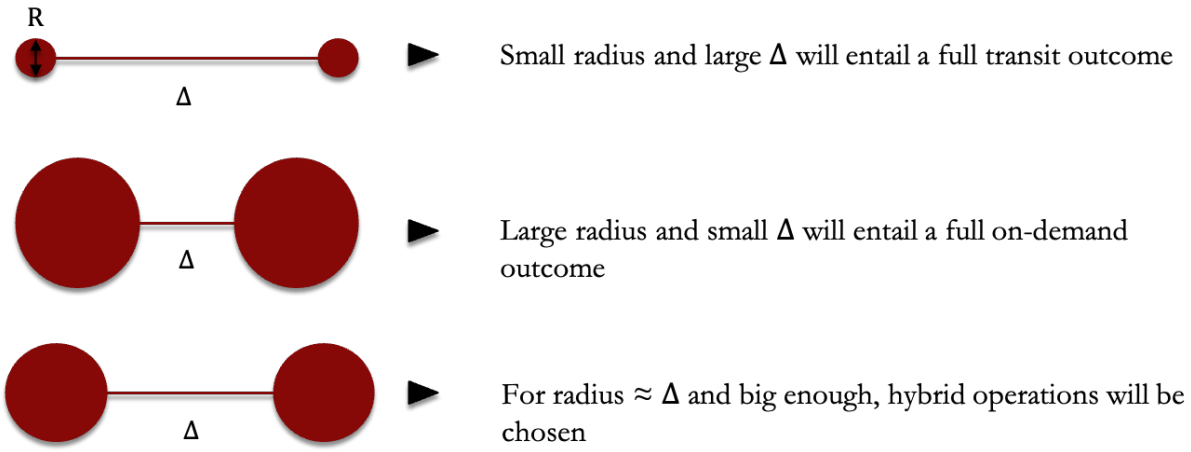


Figure 5.11: Typical behaviours of the model

CONCLUSION - NEXT STEPS

We have been able, thanks to the experiments conducted on our model, to answer to the two questions raised in 1.

However, several pain points remain pending and invite for further research. We enumerate at least:

- The Benders decomposition of the operational does not enable to scale the model: the very weak infeasibility cuts need further research. The acceleration of this Benders decomposition is crucial to the eventual scalability of the model - in its current form - as the MILP solver unavoidably reach quickly its limits in this exponential setting.
- As of now, we have not implemented the design stage: the final implementation would lead to a *three-stage* optimization, which would require further algorithmic research.

The resolution of these obstacles is necessary to resolve the tractability issues, to scale the model and eventually apply it to real-world data.

Banerjee, Siddhartha and al. (2021). "Real-Time Approximate Routing for Smart Transit Systems". In: *Association for Computing Machinery*.

Basciftci, Beste and Pascal Van Hentenryck (2022). "Capturing Travel Mode Adoption in Designing On-demand Multimodal Transit Systems". In: *Transportation Science*.

Benders, J.F. (1962). "Partitioning procedures for solving mixed-variables programming problems." In: *Numerische Mathematik* 4, pp. 238–252. URL: http://eudml.org/doc/131533.

Bertsimas, Dimitris, Yee Sian Ng, and Julia Yan (Feb. 2021). "Data-Driven Transit Network Design at Scale". In: *Operations Research* 69. DOI: 10.1287/opre.2020.2057.

Campbell, James, Andreas Ernst, and Mohan Krishnamoorthy (Oct. 2005). "Hub Arc Location Problems: Part I—Introduction and Results". In: *Management Science* 51, pp. 1540–1555. DOI: 10.1287/mnsc.1050.0406.

Diao, Mi, Hui Kong, and Jinhua Zhao (June 2021). "Impacts of transportation network companies on urban mobility". In: *Nature Sustainability* 4.6, pp. 494–500. URL: https://doi.org/10.1038/s41893-020-00678-z.

Gaubert, Stéphane and J. Frédéric Bonnans (2020). *Recherche Opérationnelle : aspects mathématiques et applications*. Editions Ecole Polytechnique.

Gurobi Optimization, LLC (2023). *Gurobi Optimizer Reference Manual*. URL: https://www.gurobi.com.

Magnanti, T.L. and Richard Wong (June 1981). "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria". In: *Operations Research* 29. DOI: 10.1287/opre.29.3.464.

Mahéo, Arthur, Philip Kilby, and Pascal Van Hentenryck (Feb. 2019). "Benders Decomposition for the Design of a Hub and Shuttle Public Transit System". In: *Transportation Science* 53.1.

Salazar, Mauro et al. (2018). *On the Interaction between Autonomous Mobility-on-Demand and Public Transportation Systems*. arXiv: 1804.11278 [cs.SY].

Sharif Azadeh, Shadi, J. van der Zee, and M. Wagenvoort (2022). "Choice-driven service network design for an integrated fixed line and demand responsive mobility system". In: *Transportation Research Part A: Policy and Practice* 166, pp. 557–574. URL: https://www.sciencedirect.com/science/article/pii/S0965856422002828.

Steiner, Konrad and Stefan Irnich (Sept. 2020). "Strategic Planning for Integrated Mobility-on-Demand and Urban Public Bus Networks". In: *Transportation Science* 54.

Zhang, Wei et al. (2022). *Routing Optimization with Vehicle-Customer Coordination*. eng. Tech. rep. 10.2139/ssrn.4208397; 10.2139/ssrn.4208397. URL: http://hdl.handle.net/11159/520611.

Linear Expansion: Delta-Radius analysis with very high reluctance to walk

	Delta	Radius	AverageWaiting	AverageWalking	AverageArrival	Transit/OnD/Hybrid	PreprAvgTime	OptAvgTime
0	10	2	206.3	15.0	58.6	0.0/0.0/9.0	15.50	29.98
1	10	5	199.4	33.6	66.2	0.0/0.1/8.9	13.94	38.00
2	10	10	188.3	57.0	78.2	0.4/0.6/8.0	13.99	90.66
3	10	15	180.0	120.9	86.8	2.6/0.6/5.8	13.99	53.66
4	10	20	169.3	149.8	99.6	3.4/1.5/4.1	13.29	110.42
5	15	2	217.9	15.9	67.6	0.0/0.0/9.0	13.66	38.61
6	15	5	215.0	37.2	74.0	0.4/0.0/8.6	13.68	31.36
7	15	10	176.3	79.1	86.2	1.6/0.5/6.9	13.62	51.71
8	15	15	179.9	131.8	99.4	2.5/0.5/6.0	12.93	63.43
9	15	20	173.0	153.2	110.4	3.3/1.2/4.5	12.89	86.86
10	20	2	287.3	15.1	82.0	0.0/0.0/9.0	13.29	19.33
11	20	5	239.9	39.6	85.8	0.8/0.0/8.2	13.50	44.57
12	20	10	187.7	87.8	95.6	1.8/0.7/6.5	12.19	30.06
13	20	15	176.3	143.5	111.8	3.0/0.4/5.6	11.76	51.47
14	20	20	180.1	185.3	117.0	3.0/0.7/5.3	11.41	37.83
15	30	2	152.9	37.5	74.0	6.0/0.0/3.0	9.48	12.21
16	30	5	239.7	51.3	109.4	2.3/0.0/6.7	9.47	22.42
17	30	10	189.8	114.3	113.7	3.7/0.1/5.2	9.25	23.29
18	30	15	175.0	163.5	117.7	3.2/0.6/5.2	8.92	16.72
19	30	20	182.8	209.6	119.6	3.7/0.5/4.8	8.43	9.89

Figure 6.1: Result table: 9 customers, *freq* = 10

Key takeaways:

- Setting enables clear ride-sharing opportunities (everyone travels in the same direction) and thus fosters hybrid operations
- High reluctance to walk favors hybridness

Linear Expansion: Sample ridesharing-oriented results

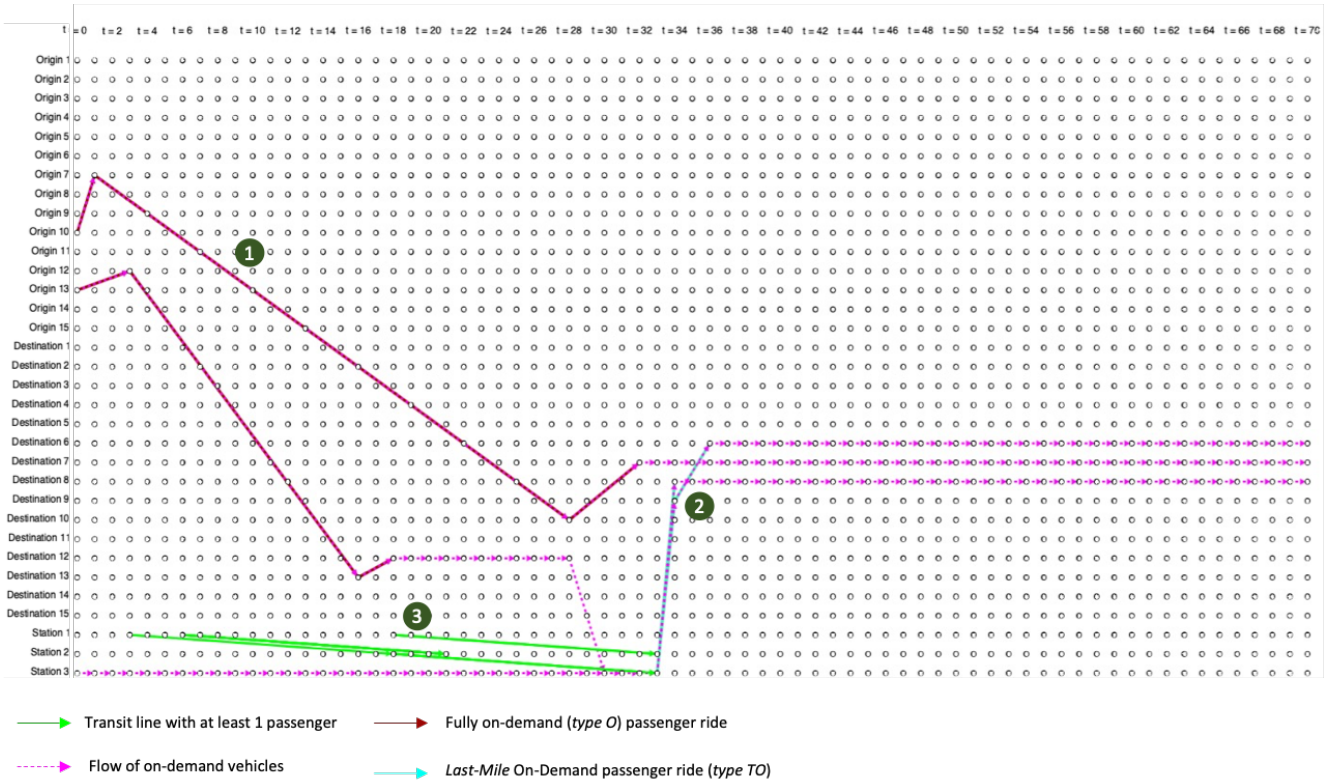


Figure 6.2: Result TSN: 15 customers, *freq* = 3

Relaxation of the *just-in-time* constraint

As we discussed, in case of a hybrid mode transportation, the current model forces the vans:

- to drop off the customer to a station *exactly* when a train departs
- to pick up the customer at a station *exactly* when the train arrives

Basically, passengers do not wait at the station after being dropped off (all the waiting is relocated to the origin node) nor they wait at the station for being picked up.

If we want to relax this, we have to *choose* the node where the van drop off and/or pick up at the station. Thus, we have to complexify the model a bit and add two new variables:

- $z^B_{p,n}$ that exists for all p $\in \mathcal{P}$ and $n \in \mathcal{N}^B_p$ and that will choose where the van will drop the customer at a boarding station for routes of type 3 and 5

- $z^A_{p,n}$ that exists for all p $\in \mathcal{P}$ and $n \in \mathcal{N}^A_p$ and that will choose where the van will pick the customer up at a alighting station for routes of type 4 and 5

Constraint (3.4) becomes:

$$\sum_{a \in \mathcal{A}^{+,tr}_n} f_{p,a} - \sum_{a \in \mathcal{A}^{-,tr}_n} f_{p,a} = \begin{cases} +z^O_{p,n} & \text{if } n \in \mathcal{N}^O_p \\ -z^B_{p,n} & \text{if } n \in \mathcal{N}^B_p \\ +z^A_{p,n} & \text{if } n \in \mathcal{N}^A_p \\ -z^D_{p,n} & \text{if } n \in \mathcal{N}^D_p \\ 0 & \text{otherwise} \end{cases} \qquad \forall p \in \mathcal{P},\ \forall n \in \mathcal{N}$$

In the preprocessing (algorithm 1), we then have to introduce a new parameter δ, that will allow for some waiting at the boarding station before the train arrives or at the alighting station before the van arrives.

We now have respectively:

- $\mathcal{N}^A_p$.add($\{(s_2, t) : t \in [t_2; t_2 + \delta]\}$)
- $\mathcal{N}^B_p$.add($\{(s_1, t) : t \in [t_1; t_1 - \delta]\}$)

In the Benders decomposition, $z^B_{p,n}$ and $z^A_{p,n}$ would be part of the first stage (assignment) to keep the unimodularity of the subproblem.