Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 86 (2025) 104–112

Transportation Research Procedia

www.elsevier.com/locate/procedia

26th Euro Working Group on Transportation Meeting (EWGT 2024)

Joint Design of Conventional Public Transport Network and Mobility on Demand

Xiaoyi Wu[a,*], Nisrine Mouhrim[b], Andrea Araldo[b], Yves Molenbruch[c], Dominique Feillet[d], Kris Braekers[e]

[a] *Department of Management, Technical Univerisity of Denmark*
[b] *SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris*
[c] *Vrije Universiteit Brussel, Business Technology and Operations*
[d] *Mines Saint-Etienne, Univ. Clermont Auvergne, CNRS, UMR 6158 LIMOS, Centre CMP*
[e] *Hasselt University, Faculty of Business Economics*

Abstract

Conventional Public Transport (PT) is based on fixed lines, running with routes and schedules determined a-priori. In low-demand areas, conventional PT is inefficient. Therein, Mobility on Demand (MoD) could serve users more efficiently and with an improved quality of service (QoS). The idea of integrating MoD into PT is therefore abundantly discussed by researchers and practitioners, mainly in the form of adding MoD *on top of PT*. Efficiency can be instead gained if also conventional PT lines are redesigned after integrating MoD in the first or last mile. In this paper we focus on this re-design problem. We devise a bilevel optimization problem where, given a certain initial design, the upper level determines stop selection and frequency settings, while the lower level routes a fleet of MoD vehicles. We propose a solution method based on Particle Swarm Optimization (PSO) for the upper level, while we adopt Large Neighborhood Search (LNS) in the lower level. Our solution method is computationally efficient and we test it in simulations with up to 10k travel requests. Results show important operational cost savings obtained via appropriately reducing the conventional PT coverage after integrating MoD, while preserving QoS.

Keywords: Ride-sharing, Mobility-on-demand, Routing Algorithms, Public Transportation, Multi-modal Routes, Mobility as a Service (MaaS), Transport Network Design

1. Introduction

It is well known that conventional Public Transport (PT) is inadequate in the suburbs. The sparse demand density in such areas forces PT operators to provide a low-frequency low-coverage service, to prevent the operational cost per passenger from exploding. This leads to poor QoS and a chronic car-dependency in the suburbs. Several studies

* Corresponding author.
E-mail address: xiawu@dtu.dk

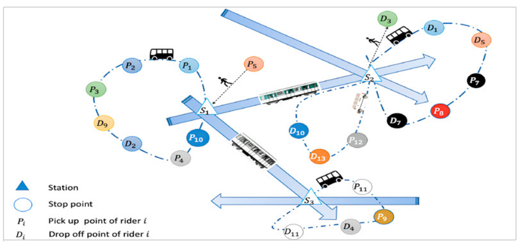
10.1016/j.trpro.2025.04.014

evaluated the benefits of MoD in low-density areas, often in mono-modal context (Hyland et al. , 2024; Linares et al. , 2016; Goncu et al. , 2022). The focus of this paper is however larger areas, which include both low and high demand density. For such areas, Mahmassani (2016) mentions (without studying it) a multimodal service, consisting of both conventional PT and MoD. Wang et al. (2024) assume that MoD is integrated with conventional PT and a single trip can be composed of conventional PT legs and MoD legs. However, conventional PT is still assumed to be unchanged, even after MoD is integrated. In Calabro et al. (2023), the joint design of conventional PT and MoD is proposed, at a strategic level, consisting of deciding in which regions MoD should operate and how conventional PT lines should change. The description of such design remains however a very high level, resorting to approximated density functions and geometrical abstractions.

This paper instead focuses on the tactical and operational aspects of multimodal PT (conventional PT + MoD). We in particular focus on redesigning conventional PT lines in order to more efficiently exploit the integration with MoD. To this aim, we present a bilevel optimization problem. In the **upper level**, we decide stop activation status and frequencies of conventional PT lines. In the **lower level**, we solve instead an Integrated Dial-A-Ride Problem (IDARP) (Posada et al. , 2017), in which MoD fleet sizing and routing decisions are taken, together with user trips. MoD vehicle routes are constructed so as to allow multimodal user trips, composed of conventional PT legs and MoD legs. We solve the upper level via Particle Swarm Optimization (PSO) metaheuristic and we use the Large Neighborhood Search (LNS) metaheuristic from Molenbruch et al. (2020) for solving the lower level. Table 1 summarizes the related work closest to ours. The main novelty of our work is that we jointly redesign conventional PT lines (via stop selection) and at the same time compute the exact routing of MoD vehicles to harmonize with conventional PT lines. We simulate our solution in a scenario representing a simplified version of the Paris region in a low demand period. We concentrate on this period, because that is when MoD becomes more adapted to the demand. Therefore, the output of our method illustrates how multimodal PT should operate in the considered time period and we assume that such structure should change over the day, according to the demand (leaving a larger and larger role to conventional PT during peak hours). We show that operational cost savings can be achieved if partially (and appropriately) replacing conventional PT with MoD, while still appropriately satisfying the considered demand. Overall, the designs we obtain imply introducing an appropriately sized MoD fleet and reducing the extent of conventional PT, by skipping certain stops (from 27% to 67% of stops, depending on the line) and reducing the bus frequency (from 64% to 94%, depending on the line) and also completely deactivating certain lines.

Table 1: Closest related work and illustration of the multimodal PT

Properties	Stiglic et al. (2018)	Sun et al. (2018)	Posada et al. (2017)	Molenbruch et al. (2020)	Lee (2017)	Steiner and Irnich (2020)	**Our paper**	Chow et al. (2019)
Public transit	Yes	Yes	Yes	Yes[1]	Yes	Yes	Yes	Yes
Online or Offline algorithm (On/Off)	Off	Off	Off	Off	Off	Off	On	On
First mile	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Last mile	No	No	Yes	Yes	No	Yes	Yes	Yes
Time window constraint	Yes	Yes	Yes	Yes	Yes	No	Yes	No
Maximum ride duration constraint	Yes	Yes	Yes	Yes	No	Yes	No	No
Meeting points	No	Yes	Yes	No	No		No	
Walking possibility	Yes	No	Yes	No	No	Yes	Yes	Yes
Door-to-door possibility	Yes	No	No	Yes	No	Yes	Yes	Yes
Single or multiple transit line (S/M)	S	S	M	M	S		M	M
Zones	No	No	No	No	Yes	Yes	No	Yes
Decision variables: Pax (passenger trajectory), Veh (vehicle trajectory)	Veh	Veh	Pax+ Veh	Pax+ Veh	Pax+ Veh	Pax	Pax+ Veh	Pax+ Veh
Heuristic (H) or Exact (E) resolution	H	H	E	H	E+H	E	H	E+H



2. System model

The form of MoD we consider is Ride Sharing (RS). The objective of the upper level is to decide stop activation status and line frequencies to minimize a cost function, which depends on the number of RS cars and PT vehicles employed. In the lower level, the objective is to minimize the total kilometers traveled by RS cars. Therefore, RS cars can transport users from their origin to their destination directly, or from their origin to a PT stop at which users can board a PT vehicle, or from a PT stop to their final destination. Due to this possibility of transferring between RS and PT, any change in the upper level of the PT layout impacts RS routing in the lower level. On the other hand, the decisions in the lower level result in a certain fleet size, which contributes to the total cost that the higher level aims to minimize.

2.1. Graph Representation of Conventional Public Transport

Public Transport (PT) is defined by set $\mathcal{PS}$ of potential stops and set $\mathcal{L}$ of lines. Each line $l \in \mathcal{L}$ is characterized by a sequence of potential stops $\mathcal{P}_l \subseteq \mathcal{P}$. A decision variable of our optimization problem will determine subset $\mathcal{S}_l \subseteq \mathcal{P}_l$ of stops that are active (stops in $\mathcal{P}_l \setminus \mathcal{S}_l$ will be skipped). *PT graph* $\mathcal{G}$ is composed of active nodes $\mathcal{S} = \bigcup_{l\in\mathcal{L}} \mathcal{S}_l$ and arcs (u, v), where u and v are consecutive active stops on the same line. We define A as the set of arcs, related to all the lines. Any line l serves the sequence of stops $\mathcal{S}_l$ in forward and backward directions and has a frequency f_l, which is a decision variable. We consider a given time period during which PT graph $\mathcal{G}$ remains unchanged. For simplicity, we assume PT vehicles have enough capacity to serve all the demand, as in (Chow et al., 2019, §3).

Average in-vehicle travel time t^{TT}_{uv} on an arc $(u, v) \in \mathcal{A}$ is known and independent of the line. We compute it as $t^{TT}_{uv} = d(u, v)/v_{\text{PT}}$, where $d(u, v)$ is the Euclidean distance between stops u and and v and v_{PT} is the average speed of a PT vehicle. Let t^{PT}_u be the dwell time at a stop $u \in \mathcal{S}$, i.e., the time a PT vehicle stays in a stop for passenger boarding and alighting. We compute average time t^l_{uv} needed to go from stop $u \in \mathcal{S}_l$ to stop $v \in \mathcal{S}_l$ (not necessarily consecutive) along a single line l:

$$t^l_{uv} = \underbrace{\frac{1}{2f_l}}_{\substack{\text{waiting time} \\ \text{(Cascetta , 2009, (2.4.28))}}} + \underbrace{\sum_{\substack{i,j\in\mathcal{S}_l(u,v) \\ i,j \text{ consecutive}}} t^{PT}_{ij}}_{\text{in-moving vehicle travel time}} + \underbrace{\sum_{i\in\mathcal{S}_l(u,v)\setminus\{u\}} t^{PT}_i}_{\text{dwell time}} \tag{1}$$

where $S_l(u, v)$ indicates the sequence of active stops between stop u and v. To compute trips between any two stops of a certain PT graph $G = (\mathcal{S}, \mathcal{A})$, we need to define an additional graph $\mathcal{G}' = (\mathcal{S}, \mathcal{A}')$, constructed as follows. There is an arc $(u, v)^l$ in $\mathcal{A}'$, whenever there exists a line l, such that is possible to go from u to v along that line. The time associated with this arc is (1). In case multiple lines connect the same pair of stops, there is a connecting arc $(u, v)^l \in \mathcal{A}'$ for each line l. Therefore, $\mathcal{G}'$ is a multi-graph. Using graph $\mathcal{G}'$, the minimal average traveling and waiting time t_{uv} between two active stops $u, v \in \mathcal{S}$ can be computed by solving a shortest path problem in multigraph $\mathcal{G}'$. Let us consider a path $\mathcal{P} = [(u_1, v_1)^{l_1}, \ldots, (u_m, v_m)^{l_m}]$ of $\mathcal{G}'$, where $v_i = u_{i+1}$. Such path means entering PT via stop u_1, boarding line l_1 and going up to stop v_1, changing for line l_2 to go from stop $v_1 = u_2$ to stop v_2, etc. Denoting with t_{ingress} and t_{egress} the time to go from the road to the PT stop and vice-versa, and denoting with t_{change} the time to walk when changing from a line to another, the travel time is of such path is

$$t_{\mathcal{P}} = t_{\text{ingress}} + \underbrace{\sum_{i=1}^{m} t^{l_i}_{u_i,v_i}}_{\text{within-lines}} + (m - 1) \cdot t_{\text{change}} + t_{\text{egress}}. \tag{2}$$

For any pair of stops $u, v \in \mathcal{S}$, $\mathcal{P}^*(u, v)$ is the shortest path, i.e., the path that minimizes quantity (2). We assume that, whenever a user enters stop u at instant t, she will reach stop v at instant $t + t_{\mathcal{P}^*(u,v)}$.

2.2. User trips

We represent n users as a set of nodes (note that they are not the nodes of PT graph $\mathcal{G}$) corresponding to their origin, destination, and transfer nodes, detailed as follows. Set $\mathcal{N}$ consists of an artificial depot node 0, a set of origin nodes $\mathcal{O} = \{1, ..., n\}$, a set of destination nodes $\mathcal{D} = \{n + 1, ..., 2n\}$. With slight abuse of notation, we will represent by i the origin of a user and the user itself, and $i + n$ referring to the corresponding destination. We also define set $\mathcal{T}_i$ of potential transfer nodes at which user i can switch from Ride Sharing (RS) to PT (or vice-versa). To define $\mathcal{T}_i$, we duplicate set $\mathcal{S}$ of active stops per each user: $\mathcal{T}_i = 2n + (i - 1) \cdot |\mathcal{S}| + 1, \quad 2n + (i - 1) \cdot |\mathcal{S}| + 2, \quad \ldots, 2n + (i - 1) \cdot |\mathcal{S}| + |\mathcal{S}|$. The entire set of possible transfer nodes is $\mathcal{T} = \bigcup_i \mathcal{T}_i$ and $\mathcal{N} = \mathcal{O} \cup \mathcal{D} \cup \mathcal{T}$.

Let $\mathcal{R}$ be the set of all trip requests. We partition it into $\mathcal{R} = \mathcal{R}_W \cup \mathcal{R}_{\text{PT}} \cup \mathcal{R}_{\text{RS}} \cup \mathcal{R}_{\text{W-PT-RS}} \cup \mathcal{R}_{\text{RS-PT-W}}$., i.e., requests of users who will just walk, or will be served by PT only, or will be served by RS only, or will walk in the first mile, enter PT in a certain stop, and then use RS in the last mile, or will use RS in the first mile, then enter PT to a certain stop and finally walk in the last mile, respectively. Therefore, we assume that RS cannot be arbitrarily chosen by a user. The transport operator will provide RS service only to users that could not perform their trip otherwise. This prevents competition between RS and conventional PT. For simplicity, we do not consider trips that have RS in both the first and last mile. This assumption can be later removed, following an approach similar to Chow et al. (2019).

The procedure to partition users is described in Alg. 1. More complex mode choice models are out of scope here and could be integrated later. In broad terms, our partitioning assumes that users would walk if possible, otherwise use PT if possible, otherwise use PT+RS if possible, otherwise use RS. It is reasonable to assume these preferences, as no monetary cost is associated with walking, some monetary cost is associated with PT and a larger monetary cost is associated with RS, so that a user would use RS only if no other feasible alternatives are available. In our case "if possible" means that the required walking time is less than a certain $d_{\text{walk}}^{\max}$ and the trip duration is less than a maximum duration M_i tolerated by user i. We assume that M_i is such that at least a direct trip via RS is shorter than M_i.

Algorithm 1: Partition of users

Input: Set $\mathcal{R}$ of users; Maximum walking distance $d_{\text{walk}}^{\max}$; Set $\mathcal{S}_i^k$, $\forall$ origin or destination i: the k closest stops to i; PT graph $\mathcal{G}$
Hyperparameters: M_i: maximum trip time user i can tolerate; τ_{RS}: threshold on RS feeder travel duration; $d_{\text{walk}}^{\max}$
Output: $\mathcal{R}', \mathcal{R}_W, \mathcal{R}_{PT}, \mathcal{R}_{\text{RS}}, \mathcal{R}_{\text{W-PT-RS}}, \mathcal{R}_{\text{RS-W-PT}}$

Initialize $\mathcal{R} = \mathcal{R}' = \mathcal{R}_W = \mathcal{R}_{PT} = \mathcal{R}_{\text{RS}} = \mathcal{R}_{\text{W-PT-RS}} = \mathcal{R}_{\text{RS-W-PT}} = \emptyset$
for *User* $i \in \mathcal{R}$ **do**
  Take origin i and destination $i+n$
  if $d(i, i+n) \leq d_{walk}^{max}$ **then**
    $\mathcal{R}_W = \mathcal{R}_W \cup \{i\}$
  else
    if $\exists$ *active stops* $s, s' \in \mathcal{S} : d(i, s) + d(s', i+n) \leq d_{walk}^{max}$ **then**
      Compute shortest path $\mathcal{P}^*(s, s')$ within PT network
      if $t_{walk}(i, s) + t_{\mathcal{P}^*(s,s')} + t_{walk}(s', i+n) \leq M_i$ **then**
        $\mathcal{R}_{\text{PT}} = \mathcal{R}_{\text{PT}} \cup \{i\}$
      else
        $\mathcal{R}' = \mathcal{R}' \cup \{i\}$
      end
    else
      $\mathcal{R}' = \mathcal{R}' \cup \{i\}$
    end
  end
end
// We have now partitioned $\mathcal{R} = \mathcal{R}' \cup \mathcal{R}_{\text{PT}} \cup \mathcal{R}_{\text{W}}$
// In the next lines we are going to partition $\mathcal{R}'$
for *User* $i \in \mathcal{R}'$ **do**
  Take origin i and destination $i+n$
  if *For any pair of active stops* $s, s' \in \mathcal{S}$*, we have* $d(i, s) > d_{walk}^{max}$ *and* $d(s', i+n) > d_{walk}^{max}$ **then**
    // User i cannot reach any close stop, neither in the first nor the last mile. User i will need a door-to-door RS trip
    $\mathcal{R}_{\text{RS}} = \mathcal{R}_{\text{RS}} \cup \{i\}$
  else
    if $\exists s, s' \in \mathcal{S}$ *s.t.* $d(i, s) \leq d_{walk}^{max}$ ***and*** $t_{walk}(i, s) + t_{\mathcal{P}^*(s,s')} + \tau_{RS} \leq M_i$ **then**
      $\mathcal{R}_{\text{W-PT-RS}} = \mathcal{R}_{\text{W-PT-RS}} \cup \{i\}$
    else if $\exists s, s' \in \mathcal{S}$ *s.t.* $d(s', i+n) \leq d_{walk}^{max}$ ***and*** $\tau_{RS} + t_{\mathcal{P}^*(s,s')} + t_{walk}(s', i+n) \leq M_i$ **then**
      $\mathcal{R}_{\text{RS-PT-W}} = \mathcal{R}_{\text{RS-PT-W}} \cup \{i\}$
    else
      $\mathcal{R}_{\text{RS}} = \mathcal{R}_{\text{RS}} \cup \{i\}$
  end
end

2.3. Ride Sharing

The Mobility in Demand service we consider is Ride Sharing (RS): a fleet of cars that can pickup and dropoff passengers. We use the model of RS and the calculation of routing for RS cars from Molenbruch et al. (2020).

In order to compute the time window that RS needs to ensure, we have to consider some constraints related to the quality of service demanded by users. Let y be any location (it can be a stop or not) and suppose we want to ensure a user i arrives at y no later than instant l_i. We can compute the latest time at which a user can depart from a stop s to arrive at location y no later than instant l_i as

$$t_{\max}(s, y, l_i) = \sup\{t | t + t_{\mathcal{P}^*(s,v)} + t_{\text{walk}}(v, y) \leq l_i, \forall v \in \mathcal{S}\} = \sup\{t = l_i - t_{\mathcal{P}^*(s,v)} - t_{\text{walk}}(v, y) | \forall v \in \mathcal{S}\} \tag{3}$$

Similarly, we focus now on a user staying at location y and willing to reach stop s as early as possible and willing to depart from y no earlier than instant e_i. The earliest arrival time at s is:

$$t_{\min}(y, s, e_i) = \inf\{t = e_i + t_{\text{walk}}(y, v)) + t_{\mathcal{P}^*(v,s)} | v \in \mathcal{S}\} \tag{4}$$

Let set $\mathcal{R}' = \mathcal{R}_{\text{RS}} \cup \mathcal{R}_{\text{W-PT-RS}} \cup \mathcal{R}_{\text{RS-PT-W}}$ contain all requests that must be handled by RS, either entirely or partially. By construction (Alg. 1), in the absence of RS, such users would need to walk more than $d_{\text{walk}}^{\max}$ (summing the first and last mile walk) or would take longer than the total tolerated trip time M_i. We assume all requests in $\mathcal{R}'$ need to be served and that the demand is inelastic, i.e., $\mathcal{R}$ does not change when changing the PT network layout or RS routing. However, $\mathcal{R}'$ changes every time we change the PT network layout, after running Alg. 1 with a new PT graph $\mathcal{G}$.

Every user i is associated with an origin node i and corresponding destination node $i+n$. RS users will need appropriate time constraints to be respected. To calculate such constraints, we treat $\mathcal{R}_{\text{RS}}, \mathcal{R}_{\text{W-PT-RS}}$ and $\mathcal{R}_{\text{RS-PT-W}}$ differently:

- For every user $i \in \mathcal{R}_{\text{RS-PT-W}}$, we assume latest arrival time l_i at destination is exogenously determined. We compute the earliest departure time at the origin as $e_i = l_i - M_i$, where M_i is the maximum tolerable trip time of user i (including waiting). Let $\mathcal{T}_i \subset \mathcal{T}$ be the set of potential transfer nodes available to user i. This set may be a subset of all eligible PT stops, e.g. the k closest stops to the user's origin. If the user is dropped off by RS in a certain transfer node, then they will traverse a PT path. If a user uses transfer node $s \in \mathcal{T}_i$, RS should drop them at s at time l_s such that, departing from s the user can reach destination $i+n$ before l_i. Via (3), the latest possible arrival time l_s at stop s is $l_s = t_{\max}(s, i+n, l_i)$.
- For every user $i \in \mathcal{R}_{\text{W-PT-RS}}$, earliest possible departure time e_i at origin i is assumed to be exogenously determined. The latest arrival time l_i at the destination is computed as $e_i + M_i$. For every transfer node $s \in \mathcal{T}_i$, the earliest possible RS departure time e_s is computed via (4) such that the user, leaving their origin after e_i, reaches stop s (via walking and conventional PT) no earlier than e_s, i.e., $e_s = t_{\min}(i, s, e_i)$.
- Users $i \in \mathcal{R}_{\text{RS}}$ declare the latest arrival time at destination, such as $t + M_i$, where t is the time instant at which user i submits their trip request.

Let us denote $\mathcal{V}$ the set of RS cars, each with capacity Q. We assume they all start and end their activity at a depot 0. For any pair of nodes i, j, the travel time by RS is $t_{i,j} = d(i,j) \cdot circ/v_{\text{car}}$, where v_{car} is the average speed of a car and $circ \geq 1$ is the circuity (Boeing, 2019), which accounts for the fact that a real world topology implies that any movement from i to j is longer than the Euclidean distance.

3. Optimization Problem

3.1. Bilevel optimization problem definition

We solve a bilevel optimization problem. In the **upper level**, we fix the following decision variables:

- Number N_l of PT vehicles for each line l and, as a consequence, average frequency $f_l = N_l/(2t_l)$, (Cascetta , 2009, (2.4.28)), where t_l is the time for a PT vehicle to go from the beginning to the end of line l and f_l is the frequency of that line. Consequently, the minimum and maximum vehicle numbers N_{min} and N_{max} are dependent on the maximum and minimum frequency, which are 0.25 and 0.06 vehicles/min.
- Set $\mathcal{S}_l \subseteq \mathcal{P}_l$ of stops to activate in each line l (represented as a vector of binary variables indicating whether a stop in $\mathcal{P}_l$ is activated or not).

A *PT layout* $\mathbf{y}$ (or *solution*) is a vector of values for the decision variables above. $\mathcal{G}(\mathbf{y})$ is the resulting PT graph (§2.1). Fixing a PT layout $\mathcal{G}(\mathbf{y})$ also induces a certain partition of users, calculated with Alg. 1, indicating whether each user walks, use PT, use RS or a combination of such modes.

The **lower level** gets set $\mathcal{R}'$ of user using RS either entirely or partially (§2.3) as input. The lower level decides:

- The fleet size $N^{\text{RS}} = |\mathcal{V}|$ of active cars in the RS service.
- The route of all ride-sharing cars, i.e., the sequence of pickups and dropoffs
- The precise trip of each user, i.e., (i) the instant and the location in which they will be picked up and dropped off (either at the origin, destination, or some transfer stop), (ii) the exact path traveled within conventional PT (if any), including changes from a line to another (if any), (iii) the trajectory traveled within a RS car (if any), including possible stopovers to serve other passengers, (iv) the walking legs.

In the lower level, decisions are calculated as in Molenbruch et al. (2020), with the objective to minimize kilometers traveled by RS cars, subject to serving all users $\mathcal{R}'$ (users using RS either for the entire trip or for a part of it) and respecting the time constraints specified in § 2.3. The resolution method is Large Neighborhood Search (LNS).

To compute the cost of a solution $\mathbf{y}$, we assume that a PT vehicle has an operating cost that is β times the one of a RS car. We wish to minimize the following expression, which is a proxy of the operating cost of the multimodal system composed of conventional PT and a RS service:

$$f(\mathbf{y}) = N^{\text{RS}} + \beta \cdot \sum_{l \in \mathcal{L}} N_l \tag{5}$$

3.2. Particle Swarm Optimization (PSO)

We adapt Particle Swarm Optimization (PSO) to solve the upper level (Alg. 2). A particle p corresponds to a sequence of PT layouts, evolving along epochs. The set of particles is called *swarm*. Saying that a particle evolves means that the corresponding layout changes from $\mathbf{y}$ in an epoch to $\mathbf{y}'$ in the following epoch. Such changes are the activation/deactivation of some stops and the number of PT and RS vehicles (§3.1). For any particle p, in addition to its layout $\mathbf{y}_p$ at the current epoch, we also keep $\mathbf{y}_p^{\text{ibest}}$ the best "version" of particle p across all previous epochs (the one with the best performance (5)). $\mathbf{y}^{\text{gbest}}$ is the best particle among the whole swarm and all epochs, up to the current epoch. The evolution of each particle p at each epoch is obtained by two perturbations: Binary PSO (BPSO) (Alg. 3 - inspired by (Khanesar et al., 2007)) activates/deactivates conventional PT stops and Discrete PSO (DPSO) (Alg. 4 - inspired by (Cipriani et al., 2020)) changes the number of PT vehicles per line.

Algorithm 2: Particle Swarm Optimization (PSO) for the upper level problem.

Input: Initial PT layout $\mathbf{y}^0$
Output: The best PT layout $\mathbf{y}^{\text{best}}$
//Initialize the swarm with the initial version of P particles
for *particle index* $p = 1, \dots, P$ **do**
  // Generate particle $\mathbf{y}_p$ *as follows*
  for *Every stop* $s \in \mathcal{P}$ *of every line* $l \in \mathcal{L}$ **do**
    Set s to active with probability 0.5
    Generate number N_l of PT vehicles in l, unif. at rnd in $[N_{\min}, N_{\max}]$
  end
  $\mathbf{y}_p^{\text{ibest}} = \mathbf{y}_p$
end
$\mathbf{y}^{\text{gbest}} = \arg \max_{p=0,1,\dots,P} f(\mathbf{y}_p)$
//Associate a "velocity" to each particle, line and stop
for *each particle index* $p = 1, \dots, P$*, line* $l \in \mathcal{L}$*, stop* $s \in \mathcal{P}$ **do**
  $v_{p,0}(l,s) = 0; v_{p,1}(l,s) = 0$
end
repeat
  for *particle index* $p = 1, \dots, P$ **do**
    Evaluate cost $f(\mathbf{y}_p)$ via (5).
    //Compare performance $f(\mathbf{y}_p)$ *to the best version particle* p*:*
    if $f(\mathbf{y}_p) < f(\mathbf{y}_p^{ibest})$ **then**
      $\mathbf{y}_p^{\text{ibest}} = \mathbf{y}_p$
    end
    //Compare performance $f(\mathbf{y}_p)$ *to the globally best particle* $\mathbf{y}^{gbest}$*:*
    if $f(\mathbf{y}_p) < f(\mathbf{y}^{gbest})$ **then**
      $\mathbf{y}^{\text{gbest}} = \mathbf{y}_p$
    end
    //Perturb the particle
    $\mathbf{y}_p = \text{BPSO}(p)$ *//Alg. 3*
    $\mathbf{y}_p = \text{DPSO}(p)$ *//Alg. 4*
    Compute the number of RS cars via the low level optimization
  end
until *number of epochs*;

Algorithm 3: BPSO for stop (de)activation

Hyperparameters: Constants C_1, C_2
Input: Index p of a particle,
Previous velocities $v_{p,0}(l,s), v_{p,1}(l,s)$, $\forall$line $l \in \mathcal{L}$ and stop $s \in \mathcal{P}_l$
Output: Updated particle $\mathbf{y}_p$
for *each line* $l \in \mathcal{L}$ *and stop* $s \in \mathcal{P}_l$ **do**
  Generate random values: r_1, r_2 uniformly at random in [0, 1]
  if *Stop s is active in* $\mathbf{y}_p^{ibest}$ **then**
    $d_1^1 = C_1 \cdot r_1$
    $d_0^1 = -C_1 \cdot r_1$
  else
    $d_0^1 = C_1 \cdot r_1$
    $d_1^1 = -C_1 \cdot r_1$;
  end
  if *Stop s is active in* $\mathbf{y}^{gbest}$ **then**
    $d_1^2 = C_2 \cdot r_2$
    $d_0^2 = -C_2 \cdot r_2$
  else
    $d_0^2 = C_2 \cdot r_2$
    $d_1^2 = -C_2 \cdot r_2$;
  end
  Generate value *inertia* uniformly at random in [−1, 1]
  //Update velocities:
  $v_{p,1}(l,s) = inertia \cdot v_{p,1}(l,s) + d_1^1 + d_1^2$
  $v_{p,0}(l,s) = inertia \cdot v_{p,0}(l,s) + d_0^1 + d_0^2$
  $v = \begin{cases} v_{p,1}(l,s) & \text{Stop } s \text{ is active in } \mathbf{y} \\ v_{p,0}(l,s) & \text{Otherwise} \end{cases}$
  With probability *sigmoid*(v), activate stop s, else deactivate it
end

Algorithm 4: DPSO for number of PT

Hyperparameters: Constants CR_1, CR_2, CR_3. By increasing CR_1 we tend to guide particles closer to $\mathbf{y}_p^{\text{ibest}}$. Higher values of CR_2 force particles to resemble $\mathbf{y}^{\text{gbest}}$. CR_3 allows for tuning the level of randomness of the particles.
Input: Particle index p
Output: Modified particle $\mathbf{y}_p$
for *each line* $l \in \mathcal{L}$ **do**
  Generate r uniformly at random in [0, 1]
  $ac = CR_1 \cdot r$
  if $ac \leq 0.5$ **then**
    $N_l^{\text{aux1}} = y_z(N_l)$
  else
    $N_l^{\text{aux1}} = y_z^{\text{ibest}}(N_l)$
  end
  Generate another r uniformly at random in [0, 1]
  $ac = CR_2 \cdot r$
  if $ac \leq 0.5$ **then**
    $N_l^{\text{aux2}} = N_l^{\text{aux1}}$
  else
    $N_l^{\text{aux2}} = y^{\text{gbest}}(N_l)$
  end
  Generate another r uniformly at random in [0, 1]
  $ac = CR_3 \cdot r$
  if $ac \leq 0.5$ **then**
    $y_z(N_l) = N_l^{\text{aux2}}$
  else
    Generate $y_z(N_l)$ uniformly at random in $[N_{\min}, N_{\max}]$.
  end
end

4. Simulation results

We conduct numerical experiments simulated in an area with the same size as the Paris region, with 7 conventional PT lines, approximately corresponding to part of the PT lines in the Paris region (see figure inside Table 3 and parameters in Table 2). For simplicity, travel time t_{ij}^{RS} on all arcs ending and starting with the depot are set to 0.

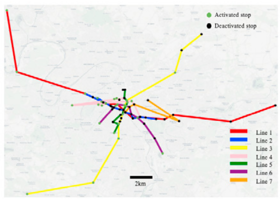
To simulate the transportation demand, we distribute user requests over a three-hour time window. The spatial distribution of these requests mirrors the geographic characteristics of the Paris region, which is divided into three

Hyperparameters of the Discrete Particle Swarm Optimization (DPSO) algorithm		
CR_1, CR_2, CR_3	0.55, 0.65, 0.52 (respectively)	
Evaluation scenario (default values in underlined bold)		
RS Car speed v_{car}	30 Kmh	Normal traffic Yong-chuan et al. (2011)
PT vehicle speed v_{PT}	60 Kmh	Metro line Domínguez et al. (2014)
Walking speed v_{walk}	1.4 m/s (5.04 Kmh)	Google Maps
Car circuity circ	1.255	Giacomin and Levinson (2015)
Walk circuity $\text{circ}_{\text{walk}}$	1.391	Zhao and Deng (2013)
$t_{\text{ingress}}, t_{\text{egress}}$	Both are 0	
Max walk distance $d^{\max}_{\text{walk}}$	2.52 Km (~30 min)	
Ingress, change and egress times $t_{\text{ingress}} = t_{\text{change}} = t_{\text{egress}}$ (§2.1)	0	
Dwell time t_i^{PT} of PT vehicle at a stop (includes time for acc(dec)elerating)	3 minutes	
RS car pickup and dropoff time (includes time for acc(dec)elerating)	1 minute	
Minimum bus headway (to avoid bus bunching)	2 minutes	Sadrani et al. (2022)
Maximum lengthening γ	$\gamma \in \{1, 1.25, \underline{\mathbf{1.5}}, 1.75, 2, 2.5, 3\}$	
Number of users (i.e., of trips)	$\{100, 500, \underline{\mathbf{1000}}, 5000, 10000\}$	
Maximum trip time M_i tolerated by user i	$M_i = \gamma \times$ direct trip time by a private car	
Cost of operating a bus	$\beta = 2\times$ cost of operating a RS car	Bosch et al. (2018), (Cats et al., 2021, Tab 3, γ_C)
Number of epochs of PSO for every scenario	50	
Processor and RAM of the PC used get our results	Threadripper 3970X, 128GB RAM	
The maximum time needed to run a single simulation	2318.77s (38.65 minutes)	

Table 2: Parameters considered

Table 3: PT layout changes in the default scenario

Line	**Skipped stops fraction**	**Reduction of num of buses**	**Num of users**
1	42%	76%	105
2	67%	80%	15
3	50%	64%	131
4	27%	88%	6
5	41%	84%	34
6	38%	94%	17
7	100%	100%	0



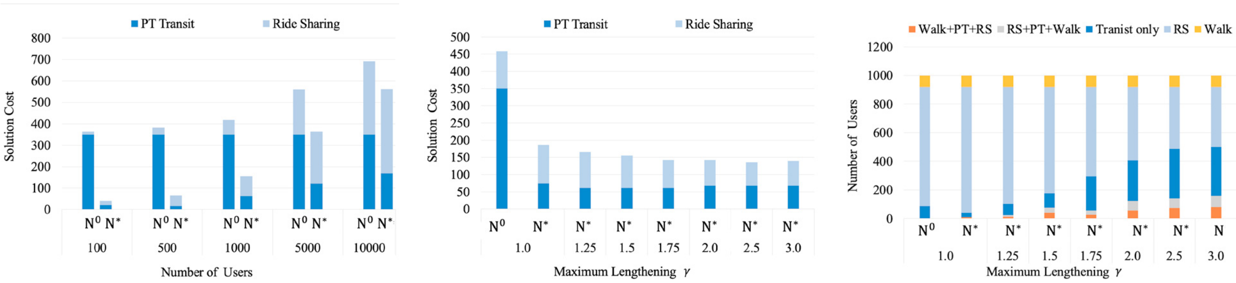
main zones: Paris (Central Zone), the Inner Suburbs, and the Outer Suburbs (see (Omnil , 2019, page 12)). We consider inter-zone and intra-zone travel requests to capture the diversity of transportation needs across the region.

Under different scenarios, we compare $N^0 = f(\mathbf{y}^0)$ and $N^* = f(\mathbf{y}^*)$, i.e., the cost of the initial and optimized solution, respectively. In all scenarios, solution $\mathbf{y}^0$ is initialized with $\sum_{l\in\mathcal{L}} N_l = 25$ buses (N_{RS} is then determined in the lower level - §3.1). For every scenario, to obtain optimized solution $\mathbf{y}^*$, we start from $\mathbf{y}^0$ and we first perform our optimization (§3.2) setting maximum lengthening parameter $\gamma = 1$ (Table 2) to obtain $\mathbf{y}^*$ and the corresponding cost $N^*_{\gamma=1}$. Then, for $\gamma = 1.25$, we start the optimization with the previously found $\mathbf{y}^*$ and we perform our optimization to obtain a new optimal solution $\mathbf{y}^*$ and the corresponding cost $N^*_{\gamma=1.25}$. We continue up to $\gamma = 3$.

The optimized layout in the default scenario (with parameters in Table 2) is shown in Table 3: buses and stops of conventional PT are considerably reduced (and so the operational cost - Fig. 1a). Line 7 is completely removed. Fig. 1a shows that cost reduction is consistent across different levels of demand. However, with few users there is more margin for cost reduction, as conventional PT becomes inefficient and can be well replaced by a relatively small fleet of vehicles. Fig. 1b and 1c show that the higher the maximum travel times M_i tolerated by users (i.e., larger γ - Table 2), the less RS is used in favor of PT.

5. Conclusion

We present a modeling and metaheuristic-based approach to design a multimodal PT, composed of conventional PT and Ride Sharing (RS). We show in simulation that the PT designs issued by our approach can reduce operational costs while respecting users' time constraints. In future work, we will apply this method to a detailed representation of a metropolitan area (current simulations are on a simplified version of the Paris Region) and increase even more the number of considered users to find up to which demand density it is convenient to integrate RS into conventional PT.



(a) (b) (c)

Fig. 1: Performance of the initial layout $\mathbf{y}^0$ and optimized layout $\mathbf{y}^*$, under different scenarios.

Acknowledgement

This study was supported by the Research Foundation - Flanders, Belgium (FWO junior research project G020222N) and, partially, by the French ANR research project MuTAS (ANR-21-CE22-0025-01).

References

Hyland, M., & Mahmassani, H. S. (2020). Operational benefits and challenges of shared-ride automated mobility-on-demand services. Transportation Research Part A: Policy and Practice, 134, 251-270.
Linares, M. P., Montero, L., Barceló, J., & Carmona, C. (2016). A simulation framework for real-time assessment of dynamic ride sharing demand responsive transportation models. In IEEE Winter Simulation Conference (WSC)
Goncu, S., Silgu, M. A., & Celikoglu, H. B. (2022). Evaluation of vehicle assignment algorithms for autonomous mobility on demand. In IEEE Intelligent Vehicles Symposium (IV)
Mahmassani, Hani S. 50th anniversary invited article—Autonomous vehicles and connected vehicle systems: Flow and operations considerations. Transportation Science 50.4 (2016): 1140-1162.
Wang, Duo, Andrea Araldo, and Mounim A. El Yacoubi. "AccEq-DRT: Planning Demand-Responsive Transit to reduce inequality of accessibility." Transportation Research Board (TRB) 103rd Annual Meeting. 2024.
Calabro, Giovanni, Araldo, Andrea, Oh, Simon, Seshadri, Ravi, Inturri, Giuseppe, Ben-Akiva, Moshe, 2023. Adaptive transit design: Optimizing fixed and demand responsive multi-modal transportation via continuous approximation. Transportation Research Part A
Posada, M., Andersson, H., Häll, C.H., The integrated dial-a-ride problem with timetabled fixed route service. Public Transport 9 (2017)
Molenbruch, Y., Braekers, K., Hirsch, P., Oberscheider, M., 2020. Analyzing the benefits of an integrated mobility system using a matheuristic routing algorithm. European Journal of Operational Research.
Stiglic, Mitja, Agatz, Niels, Savelsbergh, Martin, Gradisar, Mirko, 2018. "Enhancing urban mobility: Integrating ride-sharing and public transit." Computers & Operations Research, 90, 12–21, Elsevier.
Sun, Bo, Wei, Ming, Yang, Chunfeng, Xu, Zhihuo, Wang, Han, 2018. "Personalised and Coordinated Demand-Responsive Feeder Transit Service Design: A Genetic Algorithms Approach." Future Internet, 10(7), Article 61.
Posada, M., Andersson, H., 2017. The integrated dial-a-ride problem with timetabled fixed route service. Public Transport 9(1-2), 217-241.
Lee, Cassey, 2017. Dynamics of ride sharing competition. ISEAS Yusof Ishak Institute.
Chow, T.-Y. Ma, S. Rasulkhani, J. Y.J. Chow, S. Klein, 2019. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. Transportation Research Part E: Logistics and Transportation Review 128, 417-442.
Boeing, Geoff, 2019. Urban spatial order: street network orientation, configuration, and entropy. Applied Network Science, 4(1).
Cascetta, E. (2009). Transportation systems analysis: models and applications. Springer Science & Business Media.
Khanesar, Mojtaba Ahmadieh and Teshnehlab, Mohammad and Aliyari Shoorehdeli, Mahdi, 2007. A novel binary particle swarm optimization. 2007 Mediterranean Conference on Control & Automation, 1-6.
Cipriani, E., Fusco, G., Patella, S. M., & Petrelli, M., 2020. A Particle Swarm Optimization Algorithm for the Solution of the Transit Network Design Problem. Smart Cities, 3(2), 541–555.
Yong-chuan, Zhang and Xiao-qing, Zuo and Zhen-ting, Chen and others, 2011. Traffic congestion detection based on GPS floating-car data. Procedia Engineering 15, 5541–5546.
Domínguez, M. and Fernández-Cardador, A. and Cucala, A.P. and Gonsalves, T. and Fernández, A., 2014. Multi objective particle swarm optimization algorithm for the design of efficient ATO speed profiles in metro lines. Engineering Applications of Art. Intelligence 29, 43–53.
Giacomin, D and Levinson, D, 2015. Road network circuity in metropolitan areas. Environment and Planning B: Planning and Design 42.6
Zhao, Jinbao and Deng, Wei, 2013. Relationship of walk access distance to rapid rail transit stations with personal characteristics and station context. Journal of urban planning and development 139.4, 311–321.
Sadrani, M. and Tirachini, A. and Antoniou, C., 2022. Vehicle dispatching plan for minimizing passenger waiting time in a corridor with buses of different sizes: Model formulation and solution approaches. European Journal of Operational Research 299.1, 263–282.
Bösch, P. M., Becker, F., Becker, H., Axhausen, K. W., 2018. Cost-based analysis of autonomous mobility services. Transport Policy 64, 76-91.
Zhao, Jing, Sun, Sicheng, Cats, Oded, 2021. "Joint optimisation of regular and demand-responsive transit services." Transportmetrica A

Steiner, K., Irnich, S. (2020). Strategic planning for integrated mobility-on-demand and urban public bus networks. Transp. Sci. 54(6)
Zhao, Jinbao and Deng, Wei, 2013. Relationship of walk access distance to rapid rail transit stations with personal characteristics and station context. Journal of urban planning and development 139.4, 311–321.
Bösch, P. M., Becker, F., Becker, H., Axhausen, K. W., 2018. Cost-based analysis of autonomous mobility services. Transport Policy 64, 76-91
Omnil-Ile de France Mobilités. La nouvelle enquête globale transport Présentation des premiers resultats 2018. Technical report, 2019.