Transportation Research Part C 151 (2023) 104100

Contents lists available at ScienceDirect

Transportation Research Part C

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

Balancing convenience and sustainability in public transport through dynamic transit bus networks☆

Ayman Abdelwahed [a,*], Pieter L. van den Berg [b], Tobias Brandt [c,b], Wolfgang Ketter [d,b]

[a] *Department of Engineering Mathematics and Physics, Faculty of Engineering, Alexandria University, Alexandria, Egypt*
[b] *Department of Technology and Operations Management, Rotterdam School of Management, Erasmus University Rotterdam, Rotterdam, The Netherlands*
[c] *European Research Center for Information Systems, University of Muenster, Germany*
[d] *Faculty of Management, Economics and Social Sciences, University of Cologne, Cologne, Germany*

ARTICLE INFO

Keywords:
Sustainable urban mobility
Convenient public transport
Dynamic transit bus
Demand responsive transport
Ridesharing
Mathematical optimization
Vehicle routing problem

ABSTRACT

The rapid transformation of urban mobility fueled by the evolution of digital technologies has enabled smarter and more convenient modes of transportation. However, these new modes (e.g., ride-hailing and ridesharing) are not always more sustainable than traditional modes, such as public transport networks. In this work, we propose a dynamic transit bus system that aims at combining the higher convenience provided to passengers in on-demand systems with the sustainability of public transport. The goal is to attract more passengers to the transit service by offering reduced walk-to-station distance and total travel time compared to the fixed lines. Our proposed system plans the routes and intermediate stops of the transit lines based on the actual observed demand. Due to the complexity of the resulting optimization problem under investigation, we propose a solution method which splits the problem into four stages: clustering, initialization, optimization, and merging. We compare our proposed system in two settings: flexible (i.e., all intermediate stations are dynamic), and semi-flexible (i.e., original fixed lines intermediate stations are kept at the route and more dynamic stations are added) to the fixed static line. Our results show that our proposed system offers a more sustainable option and can reduce the system-wide travel distance by attracting more on-demand passengers to the transit service by offering them reduced walking distance and travel duration compared to the static lines. We also quantify the positive impact of passengers' willingness to act more sustainably by accepting to walk more or arrive later to their destinations on the system-wide sustainability.

1. Introduction

The rapid development in information and communication systems has reshaped our economy, society, and individual lives. Services leveraging these technologies have emerged and started to dominate a variety of markets, from hospitality to mobility, challenging their established competitors. These companies have mainly focused on using the new technologies to provide a customized, better, and more convenient service to their customers. However, the intensified focus on convenience and over-customized services may lead to less sustainable operations (Malhotra et al., 2013; Yigitcanlar et al., 2019). This aspect is

☆ This article belongs to the Virtual Special Issue on IG005586: VSI: On-Demand Transportation.

* Corresponding author.
E-mail addresses: ayman.moussa@alexu.edu.eg (A. Abdelwahed), vandenberg@rsm.nl (P.L. van den Berg), tobias.brandt@ercis.uni-muenster.de (T. Brandt), ketter@wiso.uni-koeln.de (W. Ketter).

https://doi.org/10.1016/j.trc.2023.104100
Received 29 September 2022; Received in revised form 27 January 2023; Accepted 11 March 2023
Available online 3 April 2023
0968-090X/

particularly relevant as new global challenges have emerged due to the massive increase in populations, urbanization, and resources consumption (Nejat et al., 2015). As a result, the global community started to react by establishing rules and regulations to mitigate these effects. In 2015, the United Nations have determined "making cities more sustainable and smarter" among its well-known sustainable development goals (United Nations, 2016). Accordingly, the smart city paradigm has started to gain global attention. Smart cities aim at utilizing information and communication systems alongside other technologies to tackle the challenges which urban areas face by offering sustainable and efficient operations to improve the quality of services provided to citizens (Silva et al., 2018). Among these challenges, and one of the most critical, is improving urban mobility.

Lyons (2018) defines smart urban mobility as a system that uses communication technologies to provide an affordable, effective, attractive, and sustainable service to passengers. However, an attractive and convenient system does not necessary imply sustainability, and vice versa. Sustainable urban mobility focuses on fulfilling the needs of individuals with improved efficiency, safety and security, and less pollution and energy consumption (Wefering et al., 2013). Therefore, the ultimate goal of smart urban mobility is to reach a system that balances these two aspects by providing a service that is convenient, yet sustainable. There are four main means of transportation in the current urban mobility systems: public transport, ride-hailing, vehicle-sharing and private travel (i.e., private vehicles). Each of these modes needs to make some concessions on either convenience or sustainability. For example, traditional public transport is the most sustainable option that is used for mass transport. However, it has been struggling in recent years to provide a competitive service level relative to the emerging ride-hailing services due to first and last mile issues (Shaheen and Chan, 2016). Moreover, it suffers from lower occupancy rates during off-peak hours, and overcrowdedness during peak hours. In contrast, ride-hailing services and private vehicles increases traffic congestion when more people use them instead of public transport (Agatz et al., 2021). Additionally, they increase the system-wide total travel distance and score lower in terms of sustainability.

In this work, we contribute to the literature on urban mobility by proposing a dynamic public transit bus network which can provide a convenient service and yet a sustainable system. Our proposed system resembles a demand-responsive transport system within the public transport sector. It aims for conserving the traditional public transit networks' sustainability and mass transport nature, while providing a better service to passengers using smart technologies. In our proposed system, we keep the original transit lines' terminal stations (i.e., at the beginning and the end of the lines) but with dynamic intermediate stations. Thus, the intermediate stations and the route of each trip are customized based on the actual ride requests that are submitted by the passengers. We evaluate our proposed system in two different settings: flexible and semi-flexible. In the flexible settings, only the terminal stations are kept as the fixed static line and all intermediate stations are dynamic. In the semi-flexible settings, we keep all the static line stations (i.e., terminal and intermediate) in the dynamic line, and add more stations to the line based on the demand. Thus, the semi-flexible settings would have a minimal impact on the existing riders of the static lines.

Our proposed system offers a good balance between convenience and sustainability by providing a better service (i.e., less walking distance, travel duration, and better coverage) than traditional transit networks to the on-demand riders. Moreover, it also saves a significant amount of system-wide travel distance by attracting more on-demand passengers to the transit service. Additionally, we investigate the impact of sustainable passenger behavior, reflected in being willing to walk more or having a more flexible arrival time, on the system-wide results in terms of the total system-wide travel distance to serve all passengers. Our study contributes to a growing research stream that envisions the future of public transit networks by leveraging digital technologies to restructure transit bus networks and improve their service, equity and sustainability levels. Finally, to solve the underlying optimization problem, we develop a heuristic optimization algorithm that can solve the problem efficiently and outperforms other benchmark techniques.

In the next section, we cover and summarize relevant studies in the literature. In Section 3, we describe the proposed dynamic system and explain its main characteristics. In Section 4, we introduce the resulting optimization problem and demonstrate the models we developed to solve it. Afterwards, in Section 5, we present our results and a case study using actual ride-hailing data from the city of Chicago. Finally, in the last section we summarize our findings and conclude on our work.

2. Literature review

2.1. The impact of ride-hailing on public transit ridership

The global adoption of smartphones and other mobile devices in the last decade has reshaped urban mobility around the world (Cohen-Blankshtain and Rotem-Mindali, 2016). This evolution opened the door to several new means of passenger transportation, such as ridesharing and ride-hailing services, which seized a substantial share of the urban transportation markets in cities around the world. These new urban mobility systems have been intensively researched in recent years, particularly with respect to potential benefits and optimizing their performance (Mourad et al., 2019; Agatz et al., 2012). However, the impact of these emerging systems on the ridership of public transit networks is not clear yet. There is an ongoing debate in the literature whether the relationship between ride-hailing and public transport is complementary or substitutional.

On the one hand, several studies claim that ride-hailing services are complementary to public transit networks. A simulation study by Stiglic et al. (2018) showed that integrating public transit and ridesharing can significantly improve urban mobility and increase the use of public transport. In the synthesis report prepared by Transportation Research Board and National Academies of Sciences, Engineering, and Medicine (2012), the first and last mile problem and the service area gaps are identified among the main reasons why public transit operators need to cooperate with ride-hailing operators to attract more passengers to their services. In another study, Zhang and Zhang (2018) examined the relationship between ride-hailing usage and public transit ridership at the individual level. They found that the two are significantly positively related, which again reaffirms the fact that people are willing to use public transit if the first and last mile issue is solved.

On the other hand, an opposite stream of research asserts that the substitutional effect might be stronger than the complementary effect (Tirachini, 2019). The study from Babar and Burtch (2020) shows that the entry of ride-hailing services has led to a significant reduction in the transit bus services ridership in the United States. Furthermore, various recent studies showed that the spread of ride-hailing in urban areas could have several negative effects. Beside the extra vehicle miles and pollution associated with the ride-hailing services, Gong et al. (2017) demonstrated how the entry of ride-hailing services in China between 2010 and 2015 led to an average increase in new vehicle ownership by 8%, as more people bought cars to join these services as drivers. Additionally, Agarwal et al. (2019) showed with their study in India that ride-hailing services can lead to an increase in traffic congestion of up to 41% during peak hours in busy regions. In summary, the increasing usage of ride-hailing and ridesharing services instead of public transit is potentially driving urban mobility in a less sustainable direction (Agatz et al., 2021). As an alternative to combining ride-hailing and public transit, we present a dynamic public transit solution, in which public transit bus networks are operated in a dynamic way such that the drawbacks of the first and last mile problem are reduced. We aim to reach a more flexible public transit system which can provide higher service levels to passengers with less need to integrate with ride-hailing services.

2.2. Factors attracting and deterring passengers to public transit

From a sustainability perspective, more effort is required to attract more passengers to use public transit services instead of private vehicles, taxis, or ride-hailing services. Therefore, determining the factors that attract passengers to transit networks is essential. Paulley et al. (2006) addressed this question and showed that fare elasticities tend to increase over time, and other factors such as wait and walk time are essential concerns for passengers. In line with that, Gao et al. (2016) analyzed online reviews to examine which factors affect passenger satisfaction in public transport. They deduce that accessibility, price, waiting and travel time, and transfers are among the most important factors that can lead to passengers' satisfaction. While there are many other external factors that are out of control of public transit operators, service quality has the most significant impact among all the internal factors (Taylor and Fink, 2003). In another study, Redman et al. (2013) qualitatively studied which aspects may attract private car users to public transport. They found that reliability and frequency are the most critical factors. However, they concluded that this is not enough, and more individual preferences and needs should be taken into account to attract more car users to use public transport. In line with this, our proposed system addresses these issues by aiming at providing higher service levels and taking into account individual preferences of passengers.

Beside last mile and station proximity problems, fairness and equity of the service provided to passengers represents one more essential issue which public transit networks are facing. Camporeale et al. (2019) proposed a model to investigate the equity of transit bus networks. Their results show the trade-off between the cost of the system and the realized level of equity. They also show that operators can improve the level of equity and reduce their costs with better planning. In line with this, Rosenberg (2016) presents an analysis from several cities in the United States, which indicates that housing prices positively correlate with proximity to public transit stations. Additionally, from a social perspective, Chetty et al. (2014) show that education level and income equality of several districts in the United States directly correlate with the transportation facilities in these areas. Thus, the constraints imposed by the inflexible traditional transit bus networks may further reinforce inequality already existing in society. Our proposed model offers a perspective to alleviate this challenge by allocating the stations and routing the lines dynamically based on actual demand. As a result, locations of stations and their proximity to passengers are determined based on the amount of observed demand instead of being fixed.

2.3. Flexible transit bus systems

Thereby, our work extends a stream of previous research aimed at increasing the ridership of transit bus networks by making them more flexible by either: (1) integrating and coordinating the static lines with on-demand services, and (2) dynamizing (i.e., adding flexibility to) the static lines. These two options have become more viable recently with the progress made in the development of autonomous vehicles (Narayanan et al., 2020). Leveraging the first option, Chen and Nie (2017) proposed a transit system that integrates static lines with dynamic lines that connect passengers to the fixed lines. Their results showed that the approach can reduce the total system costs (i.e., agency and user costs) compared to the static system. Pinto et al. (2020) introduced a system that jointly operates multi-modal transit networks and shared autonomous fleets. They reallocate vehicles from transit lines in areas where they are not efficient during certain times of the day to operate as a shared mobility fleet. In a similar approach, Sayarshad and Gao (2020) optimized the dynamic switching of vehicles between fixed and flexible routes and proposed re-positioning of idle vehicles to improve the social welfare and service levels provided to passengers.

Second, several other studies focus on enhancing transit networks performance and ridership by adding more flexibility to their structure and operations (Errico et al., 2013), similar to what we propose in this study. Nourbakhsh and Ouyang (2012) proposed a flexible transit system in which buses are preassigned to specific areas to serve the demand within those regions. Their results showed that the flexible system is better and has lower costs in low-to-moderate demand areas. In line with that, Basu et al. (2018) had a similar conclusion in their study and their results showed that transit systems cannot be replaced by on-demand mobility to serve mass demand. In another study, Zhao and Dessouky (2008) proposed a model to determine the optimal service area, in which the route is allowed to deviate within, that maximizes the service capacity while guaranteeing arriving on time at terminal stations. Cao and Ceder (2019) incorporated a skip-stop tactic based on real-time passenger demand to reduce passenger travel time and used vehicles. Tong et al. (2017) proposed developing new, customized, temporary (i.e.,only active for a specific time period) transit lines based on passengers' ride requests. The new lines are developed based on the aggregated demand and passengers are required to book their seats ahead for the whole active period of the line (e.g., a week, a month, …).

Finally, a few previous studies proposed a flexible transit structure that is close to our proposed system. The structure of our proposed system is close to the mobility allowance shuttle transport in which vehicles have fixed lines, but routes are allowed to deviate within a certain range from the original route. An early study from Quadrifoglio et al. (2008) proposed a mixed integer linear programming formulation with logic cuts to solve the static scheduling problem of the mobility allowance shuttle transport efficiently. Their results showed that their formulation can reduce the solution time by more than 90% compared to the exact benchmark model in some cases. Qiu et al. (2014) used analytical models and simulation to show that using dynamic stations in mobility allowance shuttle can satisfy up to 30% more passengers without additional operating costs. In recent studies, Ng and Mahmassani (2022) did a simulation study to compare the performance of semi-flexible lines, where intermediate stations are selected based on demand, against the fixed static lines. However, they do not optimize the station selection or consider passengers' individual preferences. Errico et al. (2021) introduced a similar system that they optimized following a hierarchical decomposition in which at the first step they decide on the compulsory stations with mass demand, followed by adding additional stops based on the demand, then defining the time window by scheduling the rides. However, they do not consider passengers' individual preferences regarding the walk-to-station distance, and earliest pickup and latest drop-off time. Additionally, they did not consider grouping pickups and drop-offs at potential stations (i.e., meeting points) which was shown in previous studies to have an essential impact on reducing the system-wide travel distance (Stiglic et al., 2015).

3. The dynamic transit system

Traditional static transit bus networks consist of lines, routes, and schedules. A line is defined by its two terminal stations and its route is defined by its intermediate stations. On top of them, the schedules are planned with a frequency that should correlate to passenger demand during different times of the day. The main weakness of this static structure is its rigidity. These lines and schedules are planned once, and they remain unchanged although the actual demand might have seasonal spatial or temporal shifts. Moreover, these routes might not be able to serve the scattered demand around them which goes in the same direction of the line but is not within an acceptable walking distance to the nearest station. Thus, routes of the static transit networks might be losing potential passengers because of their inflexible design.

Attempting to solve these issues, we propose a dynamic public transport (DPT) system. Instead of having fixed routes and timetables, we introduce a dynamic network that is shaped based on the actual observed demand. On-demand passengers use an online platform to submit their ride requests (i.e., their origin, destination and time). Ride requests need to be registered with a minimum notice period (i.e., the duration between registering the request and the desired travel time) that is determined by the operator. Although passengers may prefer allowing shorter notice periods, operators would have more time to improve their planning as this period gets longer. For example, an operator may use a planning period of one-hour with a cut-off interval of 15 min. Thus, if a passengers wants a ride anytime in between 9AM and 10AM, they will need to submit their request by 8:45AM. Hence, operators would have 15 min to plan their routes and communicate them to the passengers. Operators plan their routes based on the preferences expressed in the ride requests and any required operational restrictions (e.g., minimum and maximum number of stops, specific vital stops that should be included in some routes, maximum trip duration). Finally, the resulting routes and schedules are communicated via an online platform, so passengers who did not submit their ride requests can still use the service.

The dynamic network would have the same terminal stations as the already existing static lines, and dynamically select the intermediate stations (from a predefined set of potential stations) to plan the routes based on the actual demand. Since this will normally result in longer trips, we put an upper bound on the duration of dynamic trips and the number of intermediate stations per trip. Thus, the driving duration from the initial to the final terminal stations must not exceed a certain limit. This ensures that the returning trips of the line in the opposite direction can be planned accordingly.

Our system aims at maximizing the number of served ride requests by fulfilling their requirements. We define requests by their origin, destination, earliest ready-to-go time, latest arrival time, and maximum walking distance. The ready-to-go time is the time at which a passenger is ready to walk from the origin location to the assigned pickup station. The latest arrival time is the time before which a passenger should arrive at their destination. For a successfully served request, the sum of the walking distances from origin to pickup station and from drop-off station to destination should be less than the specified maximum walking distance, and the departure and arrival times should be within the ready-to-go time and the latest arrival time. If we cannot find a route and a trip which satisfies all requirements of a given request, we mark it as an unserved request. In that case, the passenger should either relax their preferences to fit with the generated route or the request will not be served by the DPT.

Fig. 1 gives an illustrative example for our proposed system's process. The maximum walking distance requested by the passengers is 1.5 km for passenger 1 and 1 km for passenger 2. As a result, the potential pickup stations for passenger 1 are "a", "b", and "c", and they can be dropped off either at "f", "g" or "j". However, for instance, passenger 1 cannot be picked up at "c" and dropped off at "g" as this will result in 2 km walking distance which is beyond the requested maximum walking distance. As shown in the figure, the dynamic route that is able to serve these two requests would travel from "O" to "c" where the bus picks up the two passengers. Afterwards, it travels to "f" to drop-off passenger 1 then to "h" to drop-off passenger 2. In this case, the actual total walking distances for passengers 1 and 2 are 1.2 km and 1 km, respectively. As shown in our example, the static route would not be able to serve passenger 1 nor 2 as there are no stations within their maximum walking distance ranges, while the dynamic route is able to serve both successfully. Beside the passengers' restrictions on the walking distance, they also specify their earliest ready-to-go

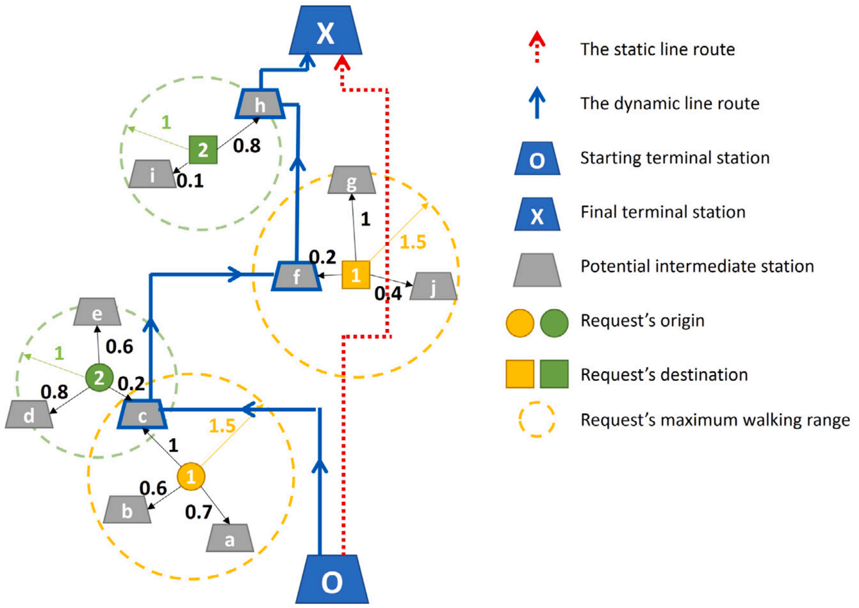

Fig. 1. Illustration of the proposed DPT's system.

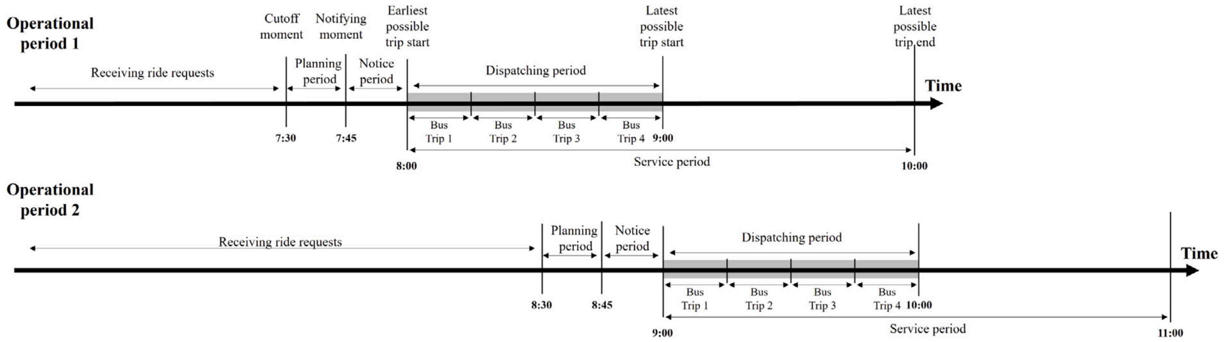

Fig. 2. Illustration of the proposed DPT's system.

time from their origin point and the latest arrival time to their destination. Hence, the dynamic route must satisfy the passengers' time constraints as well to successfully serve them. Finally, to enable the operators to plan for the return trip, the dynamic route's duration should not exceed a pre-specified limit which we model as a fixed multiplier of the static line's duration. For instance, if the planned duration of the static line is 60 min and we apply a factor of 1.25 for the dynamic route, then its duration from "O" to "X" must not exceed 75 min.

From a network planning perspective, planners will need to split the operations on each line to multiple operational periods per day. Each operational period consists of receiving requests, planning, notice, and service periods. The duration and arrangement of each period depends on the nature of the line (i.e., expected demand, route length,…,etc.). Fig. 2 shows a timeline and an example of how operational periods are structured. An operational period begins by starting to accept requests from passengers for rides during a specific time period. For example, the first operational period in the figure accepts ride requests from passengers till 7:30 for rides that start and end between 8:00 and 10:00 (i.e., the service period). The operator stops accepting ride requests for that period at the cutoff moment (i.e., at 7:30 for the first operational period), and starts planning and optimizing the routes for all the bus trips during this period. Operators would have a limited time to finish the planning (15 min in our example). Then, operators will notify on-demand passengers if their requests are accepted and publish the routes for this operational period (at the notifying moment shown in the figure). Operators should also add a notice period between the notification moment and the earliest possible trip start time to allow passengers to walk to their pickup stations. Afterwards, operators will start dispatching bus trips according to the planned optimized routes. Each trip has a dispatching window that is defined by its earliest and latest start time. For example, the first trip in the first operational period should start between 8:00 and 8:15. The whole duration between the earliest possible start time of the first trip and the latest possible start time of the last trip is defined as the dispatching period. Then, the next operational period can start anytime after the first such that dispatching periods do not overlap.

Finally, it is up to the operator to decide what to do with on-demand ride requests that could not be fulfilled with the planned dynamic routes. For example, the easiest approach would be to notify them that their exact requirements could not be met and suggest the most suitable planned trip to them. This can also probably be done after offering a fare discount based on how strict the ride request's requirements are and how far the most suitable trip meets them. For example, a passenger may require a maximum walking distance of 10 min but the best trip operators can offer would require them to walk for 15 min. Another for the public transport operator would be to integrate with ride-hailing services. Thus, they can transfer the unfulfilled requests to other providers. However, this is beyond the scope of our work and can be investigated in future work.

4. Models and formulations

To optimize our dynamic routes, we need to make various interrelated decisions regarding (1) the assignment of ride requests to vehicle/bus trips, (2) the selection of the active stations for each trip, and (3) the optimization of the route for each trip. Our decision is to determine the intermediate stations (thereby, the route) and the assignment of requests to vehicle trips. Our objective is to maximize the number of served requests by the transit bus network. Our problem resembles solving a facility allocation problem plus a multi-vehicle routing problem with time window (VRPTW), in which each vehicle has a given earliest start and latest end time at the terminal stations. For this, we develop a mixed-integer programming (MIP) formulation to optimize our decisions, which is presented in Section 4.1.

However, as is well-known, the vehicle routing problem is an NP-hard problem that is computationally intense to solve, especially in real-time or near-real-time settings. Although Alonso-Mora et al. (2017) introduced an efficient optimization algorithm that can be used to solve trip-vehicle assignment and routing in real-time ride-sharing networks, our system is more complicated as we consider meetings points (i.e., stations) and passengers' individual constraints on the maximum walking distance and time-windows. As a result, it is practically infeasible to solve the problem using a state-of-art MIP formulation, especially when the problem size increases. Accordingly, we propose a solution algorithm that decomposes our problem into smaller and easier problems to solve. Namely, we split our solution into four phases: Cluster-Initialize-Optimize-Merge (CIOM). The details of the algorithm and its four phases are presented in Section 4.2.

Nevertheless, it is not easy to compare our CIOM method to any existing algorithm as we are proposing a new problem and system that takes into consideration various factors that were not simultaneously considered in previous studies. Instead, we evaluate the performance of our solution method using two other benchmark techniques to evaluate and compare their performance to the CIOM algorithm. First, we solve the problem using the exact mathematical formulation that is introduced in the next section. Second, we develop a column generation technique, which is being widely used to solve similar vehicle routing problems, and solve our problem with it (see Appendix D for the column generation formulation).

4.1. The mathematical formulation of the problem

Before we introduce the CIOM algorithm, we first introduce the exact approach to solve the problem through an MIP formulation. Although it might not be feasible to use this exact algorithm to solve such a complicated problem in practice, we introduce it to aid in defining the problem precisely.

Our primary goal is to provide a transit service that contributes to the environmental sustainability of urban mobility while providing more convenient service than the traditional static lines. Hence, our objectives aim at increasing sustainability by attracting and serving on-demand passengers by the transit service with minimal additional travel distance. Our constraints can be split into two groups. First, constraints (3) through (12) ensure the correct flow and set the necessary configurations of the routes such as the maximum route duration and number of stations. Second, to ensure convenience, constraints (13) to (21) guarantee that the served on-demand passengers' preferences are met by the generated routes.

There are three main sets that form the input to our problem. The set R represents the ride requests in the system and is indexed by r. The set V contains all the vehicle (i.e. bus) trips in the planning window with index v, with each vehicle trip having its own earliest start and latest end times. The set S includes the potential stations which define the route for each vehicle with indices i and j, while the terminal stations (i.e., first and last stations) are the same for all the vehicle trips. We consider a hierarchical multi-objective function where we first maximize the number of served rides as shown in Eq. (1). Afterwards, we add a constraint on the minimum number of served requests, to keep them greater than or equal to the value obtained in the first step, and minimize the total travel distance as shown in Eq. (2). The two objectives are formulated in Eqs. (1) and (2) below:

$$\max \sum_{r\in R} a_r \tag{1}$$

$$\min \sum_{v\in V} \sum_{i\in S} \sum_{j\in S} x_{vij} \Delta_{ij} \tag{2}$$

where a_r is a binary variable which is set to 1 if request r is served by the DPT system and is zero otherwise. x_{vij} is a binary variable which is one if vehicle trip v travels directly from station i to station j, while the parameter Δ_{ij} gives the distance between them.

The network flow constraints are shown in (3) and (4). Constraint (3) ensures that if a vehicle visits a station, it must leave it. To avoid zigzagging in the routes, Constraint (4) restricts the number of visits to each station for each vehicle to one. Constraints (5) and (6) are used to ensure that each vehicle trip must visit the terminal stations, where o is the starting and f is the final terminal station.

$$\sum_{j\in S} x_{vij} = \sum_{j'\in S} x_{vj'i} \qquad \forall v \in V, \forall i \in S \tag{3}$$

$$\sum_{j\in S} x_{vij} \leq 1 \qquad \forall v \in V, \forall i \in S \tag{4}$$

$$\sum_{j\in S} x_{voj} = 1 \qquad \forall v \in V \tag{5}$$

$$\sum_{j\in S} x_{vjf} = 1 \qquad \forall v \in V \tag{6}$$

The variable t_{vi} is used to evaluate the arrival time of each vehicle at each station i, and is calculated using constraints (7) and (8). w_{vi} is a variable representing the waiting time at each station. Adding the waiting time variable is necessary to distinguish between the time at which passengers leave the vehicle and the latest time at which they can enter it at each stop. The parameter D_{ij} gives the travel duration between the stations. The big-M value that we use in the constraints is station-dependent and is equal to the latest possible arrival time at each station, which is defined as the latest arrival time at the final terminal station minus the direct travel duration between the two stations. Constraint (9) is used to set the arrival time at the first station for each vehicle trip to its specified starting time τ_v^s, while Constraint (10) restricts the total vehicle trip time to a maximum duration τ^m. Constraint (11) ensures that the terminal station f is the last visited station. Constraint (12) ensures that the number of active stations for each route is less than or equal to the maximum number of allowed stations Λ.

$$t_{vj} \geqslant t_{vi} + w_{vi} - M(1 - x_{ij}) + D_{ij}x_{ij} \qquad \forall v \in V, \forall j \in S, \forall i \in S \tag{7}$$

$$t_{vj} \leq t_{vi} + w_{vi} + M(1 - x_{ij}) + D_{ij}x_{ij} \qquad \forall v \in V, \forall j \in S, \forall i \in S \tag{8}$$

$$t_{vo} = \tau_v^s \qquad \forall v \in V \tag{9}$$

$$t_{vf} \leq \tau^m + t_{vo} \qquad \forall v \in V \tag{10}$$

$$t_{vf} \geqslant t_{vi} \qquad \forall v \in V, \forall i \in S \tag{11}$$

$$\sum_{i\in S}\sum_{j\in S} x_{vij} <= \Lambda + 1 \qquad \forall v \in V \tag{12}$$

To evaluate the successfully assigned variable a_r, we introduce two variables p_{vri} and q_{vri}. The former is set to one if request r is assigned to be picked up at station i by vehicle v, while the latter is set to one if it is assigned to be dropped off at station i. P_r and Q_r are the two sets of the potential pickup and drop-off stations, respectively, which are within the maximum allowed walking distance (k_r^m) for request r. Thus, constraints (13) and (14) are added to link satisfying passenger's preferences to the successfully assigned variable a_r.

$$a_r \leq \sum_{v\in V}\sum_{i\in P_r} p_{vri} \qquad \forall r \in R \tag{13}$$

$$a_r \leq \sum_{v\in V}\sum_{i\in Q_r} q_{vri} \qquad \forall r \in R \tag{14}$$

Constraint (15) is added to ensure that if a request is to be served, it must be picked up and dropped off by the same vehicle. Constraints (16) and (17) ensure that requests cannot be served by more than one vehicle, and cannot be assigned to more than one station for pickup and one station for drop-off.

$$\sum_{i\in P_r} p_{vri} = \sum_{i\in Q_r} q_{vri} \qquad \forall v \in V, r \in R \tag{15}$$

$$\sum_{v\in V}\sum_{i\in P_r} p_{vri} \leq 1 \qquad \forall r \in R \tag{16}$$

$$\sum_{v\in V}\sum_{i\in P_r} q_{vri} \leq 1 \qquad \forall r \in R \tag{17}$$

Constraint (18) is added to ensure that if a request is served, its passenger's total walking distance must not exceed the maximum allowed distance (k_r^m). k_{ri}^o and k_{ri}^d represent the walking distances from origin location to stations and from stations to destination location, respectively.

$$\sum_{v\in V}\sum_{i\in P_r} k_{ri}^o p_{vri} + \sum_{v\in V}\sum_{i\in Q_r} k_{ri}^d q_{vri} \leq k_r^m \qquad \forall r \in R \tag{18}$$

Additionally, for a request to be served successfully, it must be picked up after the earliest possible pickup time and arrive in time at its destination. Constraints (19) and (20) are added to guarantee that, respectively. e_{ri} is the earliest pickup time of request r at station i, it is equal to the earliest ready-to-go time from the origin plus the walking duration from the origin location to the

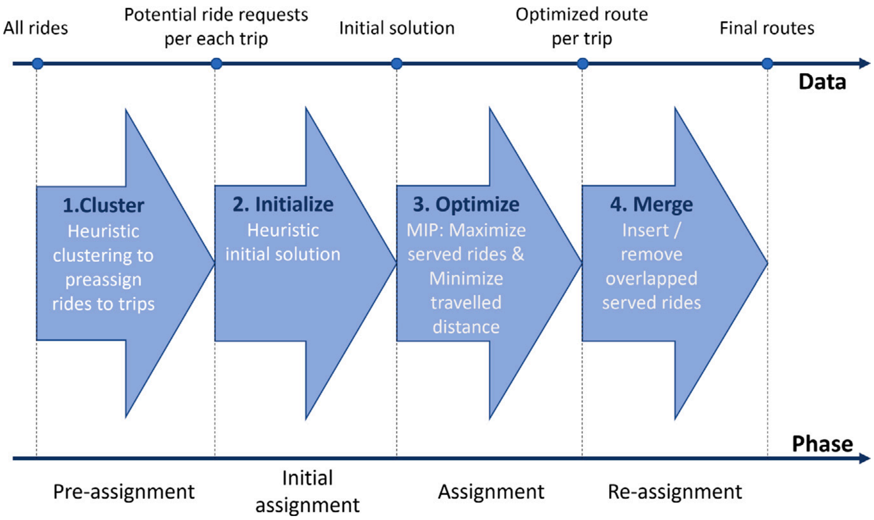

Fig. 3. The steps of the CIOM algorithm.

station. Similarly, l_{ri} is the latest arrival time at station i after taking into account the walking distance from the station to the final destination.

$$t_{vi} \geqslant e_{ri} p_{vri} - w_{vi} \qquad \forall v \in V, r \in R, i \in P_r \tag{19}$$

$$t_{vi} \leq l_{ri} q_{vri} + M(1 - q_{vri}) \qquad \forall v \in V, r \in R, i \in Q_r \tag{20}$$

Finally, successfully assigned requests must be picked up before being dropped off which is ensured by Constraint (21).

$$t_{vi} - M(1 - p_{vri}) \leq t_{vj} + M(1 - q_{vri}) \qquad \forall v \in V, r \in R, i \in P_r, j \in Q_r \tag{21}$$

4.2. The cluster-initialize-optimize-merge algorithm

Although the exact formulation developed and presented in the previous section can theoretically reach the global optimal solution of our problem, it is practically infeasible to use it. This is specifically true with larger instances with a large number of rides and many bus trips. Hence, we develop a heuristic algorithm which decomposes the original problem into four subproblems that aims at reaching a good solution within a practically feasible time span.

Our proposed algorithm decomposes the problem into four subproblems or stages as shown in Fig. 3. In the first stage, the clustering step, we determine which rides may potentially fit to each vehicle trip that is scheduled to start at a specific time from the starting terminal station. The output of the clustering phase is the set of potential ride requests for each vehicle trip. From this step onward (i.e., in the second and third steps) we solve the problem for each vehicle trip separately and in the last step we merge the solutions. In the initialization phase, we first solve the station allocation problem heuristically to select the set of intermediate stations. Then, given these stations, we solve the vehicle routing optimization problem. Given the initial solution from the initialization phase, in the third phase we resolve the vehicle routing optimization problem but with the whole set of intermediate stations. Finally, since the proposed algorithm may result in some ride requests being assigned to be served by more than one vehicle trip, we fix this in the merging stage. We check if there are ride requests that are successfully assigned to two or more vehicle trips. If that is the case, we attempt to remove these requests from these vehicle trips that can use space to add other ride requests after the removal, such that more requests would be served. We present the details of each step next.

4.2.1. The clustering phase

The goal of this step is to select the rides which may potentially fit each vehicle trip. Fig. 4 gives an illustrative example of how this selection process is executed. First, we determine the covered area around the bus line, which is illustrated by the dotted area in the figure. We only consider the rides whose origins and destinations lie within this area for this line. Next, we check if rides can potentially be assigned to a vehicle trip based on its direction and required time. First, we check for direction. Namely, we only include rides for which the angle between their origin–destination vector and the vector connecting the trip's two terminal stations is smaller than a pre-specified threshold (e.g., 90 degrees). Excluded rides can potentially be assigned to the return trip of the line going in the opposite direction. For example, ride 1 cannot be assigned to the line going from O to X, but can be assigned to the line going from X to O.

Afterwards, we check if the ride's time window corresponds with this vehicle trip time window. Namely, we check if the time window of the ride request (i.e., based on its earliest ready-to-go time) and the trip schedule (i.e., based on its scheduled start and

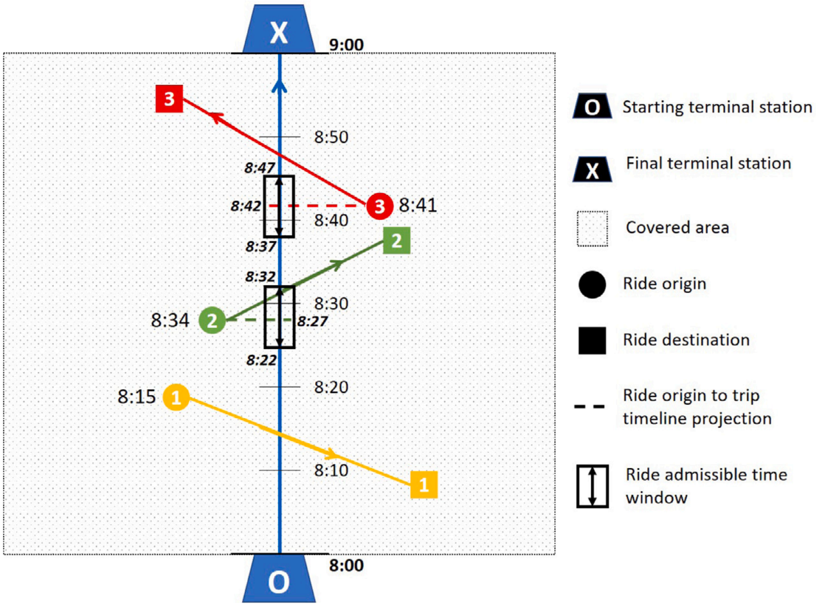

Fig. 4. Illustrative example of the clustering step.

end time) overlap. For example, in Fig. 4 we show a vehicle trip which starts at 8:00 at O and should end at 9:00 at X. Based on that, we plot the timeline as shown in the figure with each point having a specific time, which corresponds to a fixed speed and straight line. Then, we plot the projection line from the ride's origin to the vehicle trip timeline as shown by the dotted lines in the figure. Based on the projection point, we plot the ride admissible time window around it. For example, for ride 2 in Fig. 4, the projection point from the origin to the trip timeline is 8:27. We then draw the time window around this point ranging from plus or minus a pre-specified time threshold (e.g., 5 min). As a result, the admissible time window for ride 2 is between 8:22 and 8:32. If the ride's earliest pickup time is within that time window, ride 2 would be added as a potential ride request for this vehicle trip. In our example, the earliest pickup time for ride 2 is 8:34, so it is not within the admissible time window and it will not be considered as a potential ride for this vehicle trip. However, it might be considered for the next vehicle trip after this one if it starts a few minutes later. In contrast, ride 3's earliest pickup time is 8:41 which falls within its admissible time window that ranges from 8:37 to 8:47. As a result, ride 3 will be added as a potential ride for this vehicle trip as it also passes the direction check.

It is important to note that there are a few parameters which need to be tuned in this model to select the best potentially fitting rides for each vehicle trip. First, the covered area needs to be determined and can be based on either a specific distance range or on neighborhoods around the line. Second, the range of the time window needs to be specified. In our illustrative example, we used a simple interval of plus or minus five minutes to the projection point. However, in practice we tune it according to a directly proportional relation to the maximum permissible additional trip duration compared to the static trip. Accordingly, we can have wider admissible time windows for ride requests with more flexible time windows. Additionally, we adjust this time window according to how far the pickup point is situated from the start of the line. We narrow down the admissible time window for ride requests which are nearer to the starting points, as near the start the bus has probably not picked up many requests yet and its aggregate delay compared to the static line is still small. Additionally, this might also be adjusted according to the frequency of the line. Finally, it is important to point out that ride requests may get linked to multiple vehicle trips as a result of this approach. As a consequence, we may get a ride request that is served by two or more vehicle trips — an issue that is resolved in the merging step.

4.2.2. *The initialization phase*

After the clustering stage, we know the potential ride requests that might be feasibly added to each vehicle trip in our planning horizon. Before optimizing the stations' selection and vehicle routing using the exact MIP formulation for each vehicle trip, we go through the initialization step. This is an optional step and its aim is to improve the efficiency and the computational performance of the next step by providing an initial solution.

For extracting a reasonable initial feasible solution, we developed a heuristic algorithm that solves the problem by decomposing it into a sequence of smaller subproblems as shown in Fig. 5. These subproblems are classified into two main phases: (1) solving a coverage and (2) solving a routing problem. The procedures of the algorithm are:

1. *The coverage problem*: The main objective of this stage is to determine the intermediate stations which can potentially cover the maximum number of requests. It consists of the following three steps:

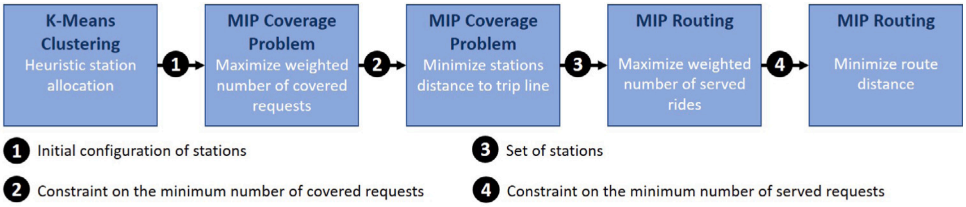

Fig. 5. The steps of the initialization phase.

(a) As a first step, we determine the busiest demand spots around the line, i.e. those with many pickup and/or drop-off requests, by solving a k-means clustering problem. We set the number of clusters to the maximum number of stations which we want to allow along the line. Then, we select the intermediate station which is nearest to the centroid of each cluster and provide this configuration as an initial solution.
(b) Second, given the set of stations from the previous step as initial solution, we solve the MIP coverage problem (see Appendix A for details). We define a request to be covered if it has a station for pickup near the origin and another for drop-off near the destination such that the total walking distance is less than the maximum determined by the passenger. We solve the coverage problem with the objective of maximizing the weighted number of requests covered given the maximum number of stations. Each request is given a weight based on its occurrence in the assignment resulting from the clustering step. This implies that if a request is assigned only to one trip, its weight would be 1, while a request that matches two trips would have a weight of 0.5 and 0.33 in case it matches three trips. This way, we direct the model towards minimizing the overlap in the served rides among the trips.
(c) Finally, before solving the routing problem, we attempt to bring the stations closer to each other. We do so by minimizing the orthogonal distance between the stations and the original static line's route. We re-solve the same MIP coverage problem but with the objective of minimizing the intra-distances while putting a constraint on the minimum number of successfully covered requests resulting from the previous step. Thus, we cover the same number of requests but with stations that are nearer to the original line's route.

2. *The routing problem*: Nevertheless, covering a request by providing a pair of pickup and drop-off stations that would result in a walking distance shorter than the maximum distance preferred by the passenger does not guarantee successfully serving the request if the time constraints are not satisfied as well. Hence, the main objective in the routing step is to maximize the number of successfully served requests by satisfying the arrival/departure time constraints.

(a) First, we maximize the weighted number of successfully served requests. Once more, the request weights are calculated as previously explained.
(b) Then, we put a constraint on the minimum number of successfully served requests resulting from the previous step and we re-solve the MIP problem to minimize the route distance.

For the routing problem, we solve the MIP problem formulated before in Section 4.1 but with only the set of stations selected from the coverage problem.

4.2.3. *The optimization phase*

After the initialization step, we move to the third phase in which the routes are optimized for each vehicle trip. Given the initial solution, once more we solve the exact station selection and routing MIP formulation presented in Section 4.1 but for one vehicle trip at a time. Accordingly, the v index is removed from the problem. We use the CPLEX default branch and cut algorithm to solve the problem.

Given the initial solution with specific stations which were selected heuristically in the initialization step, now we solve the exact problem with all the possible intermediate stations. Thus, the model can move or select another station to serve more rides or serve the same number of rides but with shorter travel distance. Once more, the problem is solved in two steps. In the first step, we maximize the number of served rides. Then, we put a constraint to serve at least the same number of rides we gained from the first step and minimize the travel distance.

4.2.4. *The merging phase*

After the third step, we have an optimized route serving the maximum possible rides at the minimum distance per each vehicle trip. We can now classify the potential requests per vehicle trip, which were assigned in the clustering phase, into three groups: exclusively served requests, overlapping served requests, and unassigned requests. The exclusively served requests of a vehicle trip are those which are served by it and are not served by any other vehicle trip. The overlapping served requests of a vehicle trip are those served by it and at least one other vehicle trip. The unassigned rides are those which belong to the potential requests of the vehicle trip but were not selected to be served by it.

Fig. 6 shows a simple example with two vehicle trips and ten ride requests. The first column in each matrix is the initial solution that we received from the preceding optimization step. In the initial solution, Trip 1 is serving requests [1,2,3,4,5] and Trip 2 is

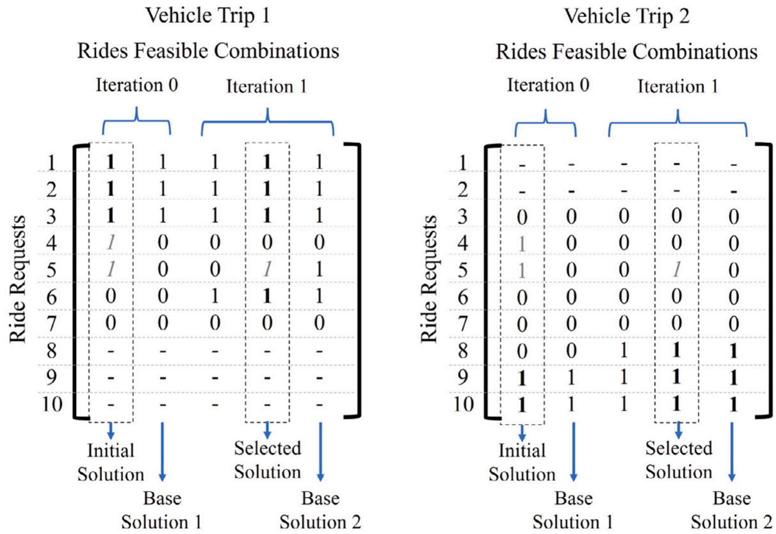

Fig. 6. Illustrative example of the merging phase.

serving requests [4,5,9,10]. Thus, in total there are seven requests being served. It is clear that there might be a space for further improvement and to successfully serve more ride requests by removing the overlapping served ride requests and potentially serving other unassigned requests. Hence, in this phase we check if we can serve more rides by developing a removal-insertion algorithm. Our algorithm follows a logic which is similar to the column-generation algorithm. We generate the columns, with each column representing a different combination of served requests. We solve sub- & master MIP optimization problems to add new columns, which can improve the solution, to each matrix as demonstrated in the pseudocode in Appendix B. As in the column-generation algorithm, we solve the subproblem for each vehicle trip separately while the master problem is solved for the whole set of vehicle trips.

In Fig. 6, the bold ones in the initial solution represent the exclusively served requests, which will be fixed in all the solutions of the next iteration. The italic grey ones are the overlapping served requests, while the zeros are the unassigned ride requests. Finally, the blocked requests are those which do not belong to the potential requests set of this vehicle trip. The second column represents the base solution of the initial solution which contains only the exclusively served requests.

The next iteration starts from the base solution of the last iteration. We exclude all the overlapping served requests and maximize the number of remaining served requests by solving the sub-problem, with the same formulation presented in Section 4.1 being used. If more requests can be served (as shown in the third column of each matrix), we add this column to the matrix and we check if any more requests from the previously overlapping ones can be added. If one or more from the previously overlapping requests could be served, then we add this column to the matrix (as shown in the fourth column of each matrix). Then, we solve the master problem to select the columns that will maximize the total number of requests served by the two vehicles. In that case, the fourth column of each matrix will be selected which will result in serving requests [1,2,3,5,6] by Trip 1 and requests [5,8,9,10] by Trip 2. Thus the total number of served requests increased to eight compared to seven in the initial solution. Given the selected solution, we re-evaluate the base solution and go to the next iteration. The algorithm terminates once the master problem cannot find a new solution or when the maximum number of iterations is reached.

5. Case study

To test and assess our proposed system, we use the ride-hailing open-source data from Chicago in 2019 (Chicago Data Portal, 2020), which we will introduce in detail below. We carry out two different numerical analyses which will be presented in Section 5.2 to show the benefits of the proposed dynamic transit network when compared to the static one in serving the on-demand passengers and fulfilling their travel requirements. Moreover, to evaluate the computational performance of the proposed heuristic approach, we compare it to a set of benchmark algorithms based on the exact formulation in Appendix E. The results show that the heuristic is able to find better solutions, even if the exact algorithm is given twice as much time.

5.1. Data description

The city of Chicago provides an extensive open-source and periodically updated database containing all the reported rides provided by ride-hailing companies in the city starting from November 2018. We focus on the data of the year 2019 to eliminate the abnormal effect on the ridership caused by the Covid-19 pandemic in 2020. The data from 2019 contains 105 million rides operated by the different hailing service providers in the city.

Table 1
Assumed passengers' maximum permissible walking distance and delay.

Original ride-hailing distance	Possible maximum walking distance	Original ride-hailing duration	Possible maximum delay
>1 km & ⩽ 1.5 km	0.5 km	⩽ 15 min	10, 15 min
⩽ 2 km	0.5, 0.75 km	⩽ 30 min	10, 15, 20 min
⩽ 3 km	0.5, 0.75, 1 km	⩽ 60 min	10, 15, 20, 25, 30 min
⩽ 4 km	0.5, 0.75, 1, 1.25 km	>60 min	10, 15, 20, 25, 30, 35 min
>4 km	0.5, 0.75, 1, 1.25, 1.5 km		

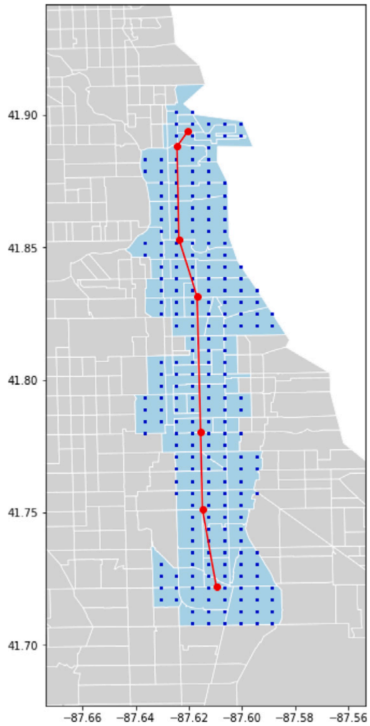

Fig. 7. The corresponding tracts and potential set of stations.

Due to privacy concerns, the exact pickup and drop-off coordinates are not provided. Instead, rides are defined by their pickup and drop-off census tracts' centroids and IDs. Similarly, the pickup and drop-off times are rounded in the data to the nearest 15 min. Since we want to avoid having all the rides within a census tract to be starting and ending at the exact same location, we redistribute the rides' origins and destinations throughout the tract's area uniformly at random. Similarly, we redistribute the pickup and drop-off times within the nearest 15 min time interval. Finally, we exclude all the rides that have a ride-hailing distance that is less than 1 km. We assume that passengers who used ride-hailing services for such short rides will not use public transport even if the quality of the service would be improved.

To incorporate passengers' preferences, we add the attributes *maximum walking distance* and *maximum delay duration* to each ride request. The maximum walking distance is the distance that must not be exceeded by the sum of the walking distances from the origin and destination of the passenger to the pickup and drop-off stations, respectively. The maximum delay is the extra arrival delay to the destination that a passenger would accept when using the DPT compared to ride-hailing. The values of the two attributes are randomly selected according to a uniform distribution based on the original ride-hailing trip's distance and duration as shown in Table 1. Our main assumption is that when the ride gets longer, passengers might be willing to walk or be delayed a bit more when they use the DPT service. Given that, we assume in all our analyses and results that a passenger would use the public transport service instead of ride-hailing if their walking and delay preferences are satisfied. Otherwise, if the public transport service does not satisfy a passenger's preferences, they stick to using ride-hailing.

We consider a single transit bus line for our analyses, namely line 3 King Drive. We start from the currently existing transit bus line, shown in Fig. 7. We classify all the tracts which the transit line crosses and their neighboring tracts as corresponding tracts which are highlighted in light-blue in Fig. 7. Accordingly, all the rides in the data that start and end at any of the corresponding tracts are selected from the data and included in our analysis. The 2019 data contains around 8 million rides starting and ending within these tracts which represents almost 7.5% of the total rides. The solid red line and points on it represent the current static public transit line and its stations, respectively. As previously explained, our objective in the dynamic public transport network is to reroute the line based on demand. Therefore, we generate a set of potential locations of stations that are uniformly distributed across the corresponding tracts in a grid structure with 0.5 km intra-distances as shown in Fig. 7 (the blue square-shaped scattered points). Although we assume an evenly distributed synthetic set of potential stations in our case study, feasible and optimal actual potential locations of intermediate stations will depend on several factors in a real-life implementation such as road blockages, road directions, and street width. Our algorithm can directly handle any input set of potential stations defined by the operator. Additionally, in the case study, we assume and set values for our main routing configuration and some other parameters as follows:

- Each operational period has four trips that are dispatched within one hour.
- We allow an extra duration factor of 1.25 for the DPT network. Thus, each DPT trip has a maximum duration of 75 min since the duration of the static trips is 60 min.
- We assume that each bus will do a round trip in the opposite inbound direction after finishing its outbound trip. The same planning process needs to be carried out for the inbound trips as well. Operators would need to plan the inbound trips based on the maximum duration of the outbound trips (i.e., 75 min in our case study).
- The maximum number of allowed stations in the DPT is 12 compared to seven in the static lines.
- We assume a walking speed of 5 km per hour for passengers which is used to calculate their walking duration to/from the pick-up/drop-off stations.
- We assume an average driving speed for the buses of approximately 20 km per hour, including its stops, based on the actual line's duration and distance.
- We do not consider the bus capacity in our study as we assume that operators can dispatch bigger or smaller buses based on their knowledge about the demand for each line at different times of the day.

Although we consider only a single bus line in our analysis, our proposed system and algorithm can be easily applied at a full network scale. Each bus line in the network represents a separate planning/optimization problem. Hence, the CIOM algorithm can be independently applied in parallel to several bus lines as long as there is no overlap in the set of potential rides they can serve. However, in practice there might be more than one bus line traveling on the same route that can serve the same potential requests. Yet, it is not a problem if that is the case as our CIOM algorithm can consider that overlap in the merging phase. Namely, ride requests originating and heading to locations within the overlapped areas can be considered by and assigned to trips from different lines. Afterwards, trips from the overlapped lines need to be resolved in the merging step to generate the final assignment of requests to trips. Finally, we assume that on-demand passengers are not willing to accept transfers (i.e., by taking more than one bus trip to go from their origin to destination) in their rides. Thus, we do not consider ride requests that can only be served through transfers.

5.2. Results

We simulate and evaluate the transit bus network performance with different settings for three consecutive days (from 1-May-2019 to 3-May-2019). We simulate six hours at different times of the day: 5AM, 8AM, 11AM, 2PM, 5PM, and 8PM. We assume that there will be four trips per hour (every 15 min) for the public transit network (i.e., first trip at Hour:00 and last at Hour:45), which results in 72 trips in total.

5.2.1. Comparing the static and dynamic public transport models

Our goal in the first comparison is to compare the different transit network modes in terms of the total system-wide travel distance (i.e., transit service travel distance plus ride-hailing travel distance required to serve unsatisfied demand by the transit service) and the provided quality of service as measured by walking distance and trip duration. We compare four different scenarios:

- Static public transit (SPT): the currently existing transit bus network that serves the area under study.
- Optimized static public transit (OSPT): a hypothetical scenario, in which we re-plan a static line based on the historical ride-hailing data (i.e., data from the period before our test horizon). We provide more details about this system and how we optimized it below.
- Dynamic public transit (DPT): the proposed system, in which we fix the two terminal stations of the line in the SPT, but we allow all intermediate stations to be assigned to the trips based on the actual observed demand (i.e., on-demand passengers submitting their requests to the DPT system instead of ride-hailing).
- Semi-dynamic public transit (SDPT): similar to the DPT, but we add a constraint to keep all the original SPT line's intermediate stations in the route.

In all scenarios, we assume that passengers will use the transit service if they can find a trip which satisfies their required arrival time and maximum walking distance. If they do not find a trip fulfilling their preference constraints, they will use ride-hailing services to perform their rides. For the DPT, we will use our proposed CIOM algorithm to plan the route of each trip. For the SDPT, we use the same model but with an additional constraint to ensure that all the original static line's intermediate station will be included in the dynamic route. We evaluate the SDPT as it is closer to the static system and will have a minimal impact on the current riders who use the SPT line and do not want to submit their ride requests to the system. We limit the total route duration in the dynamic scenarios to around 75 min, compared to around 60 min in the SPT. Thus, the maximum possible delay for a passenger taking a ride from the first to the last terminal station would be 15 min compared to the status quo. This also allows operators to easily plan the return trips accordingly as they know the latest arrival time of the dynamic trips at the terminal stations. As we

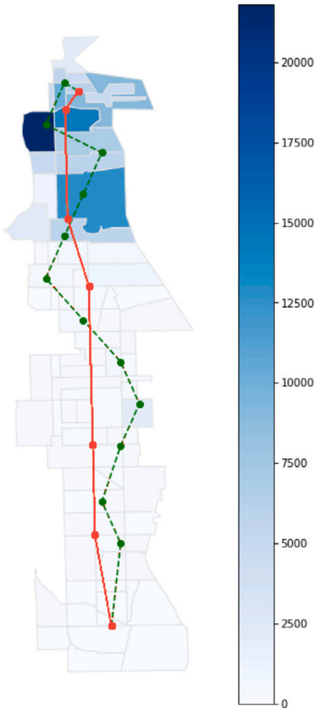

Fig. 8. The static route (solid red line) and the optimized static route (dotted green line) based on historical demand.

restrict the maximum number of additional intermediate stations that can be added to the dynamic routes to five, this limits the negative impact on the passenger experience from having too frequent stops.

For the OSPT, we re-plan the static line structure based on demand data throughout the period from January to April 2019. We run this scenario because the dynamic scenarios will be optimized for the on-demand passengers while the existing real-world static line is not. OSPT provides a fair benchmark to evaluate the dynamic scenarios against as it is also optimized for the on-demand passengers. To optimize the static route, we develop a MIP formulation (see Appendix C) that is based on solving a coverage problem. Thus, we add stations such that the demand coverage is maximized. We add constraints to limit the number of stations per route, control the minimum and maximum distances between the stations, and avoid zigzagging. Fig. 8 shows the resulting OSPT (the green dotted) line in comparison to the SPT (the solid red) line and the heatmap of the demand over all the corresponding tracts. The figure shows that the OSPT is better in covering areas with higher on-demand ride-hailing demand compared to the SPT at the cost of a longer route. Naturally, for the SPT and OSPT all the trips' routes are identical, respectively, while the DPT and SDPT customize the route and stations based on the on-demand riders' requests. Finally, the two terminal stations are kept the same in all four scenarios.

The results of our comparative study between the four scenarios are shown in Fig. 9a and 9b. The six hours included in the study over the three days involve 8238 ride requests. The static scenario is able to serve only 2.4% of these requests, which gives a clear indication that the majority of these passengers might have not used the transit service in reality because of their limited accessibility to its fixed route. In contrast, The DPT is able to serve almost 25% of the on-demand ride requests which also results in saving almost 25% of the system-wide travel distance compared to the SPT scenario (reduction from 25,857 to 19,424 km). The SDPT serves around 21% of the on-demand requests and reduces the system-wide distance by almost 19%. Thus, the operators may opt to sacrifice the additional 4% of served on-demand requests in the SDPT compared to DPT to keep their original static line intermediate stations and have a minimal impact on their existing ridership. Finally, the OSPT line can only serve 10.7% of the on-demand ride requests, which is significantly lower than the two dynamic systems. Thus, even if operators decide to optimize and launch a custom line to serve the on-demand riders, it would not be able to serve as many passengers as the dynamic system.

Fig. 10 shows the relationship between the percentage of on-demand requests that the DPT system can serve against the total number of submitted requests per planning period (i.e., hour). The results show that as the number of requests per planning period increases, the ability of the DPT system to serve more requests decreases. This is normal due to the fact that more requests would result in requiring having more distinct pickup and drop-off stations which cannot be fulfilled under the restrictions of the dynamic routes' maximum duration and number of stations. Thus, operators may need to increase the frequency of their service to be able to serve more on-demand passengers.

Finally, we assess the performance of the different systems from a passenger perspective. Table 2 shows the average walking distance and ride duration among the successfully served rides for each scenario. We calculate the ride duration as the duration of

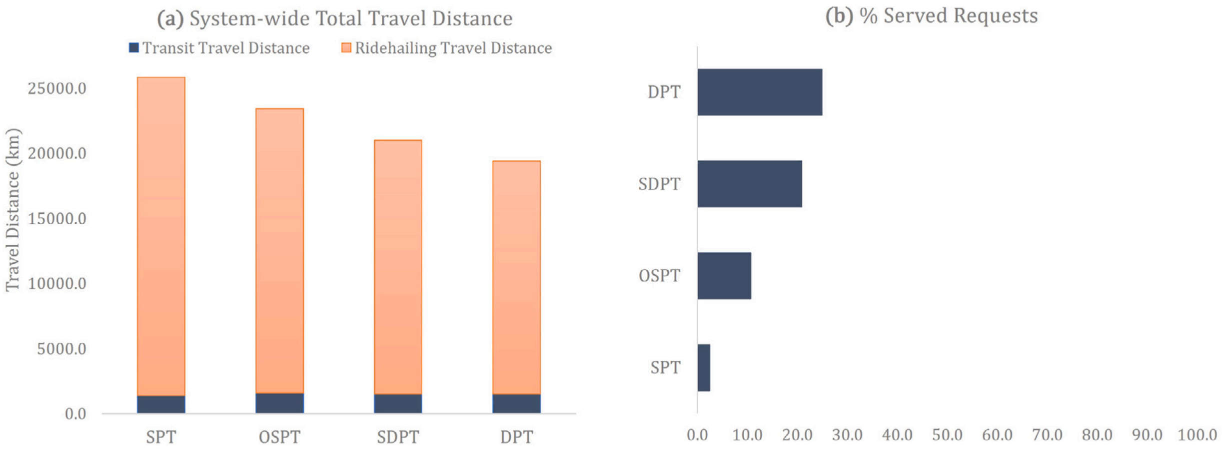

Fig. 9. Comparison between the four systems regarding: (a) transit and system-wide travel distances, (b) percentage of served on-demand ride requests.

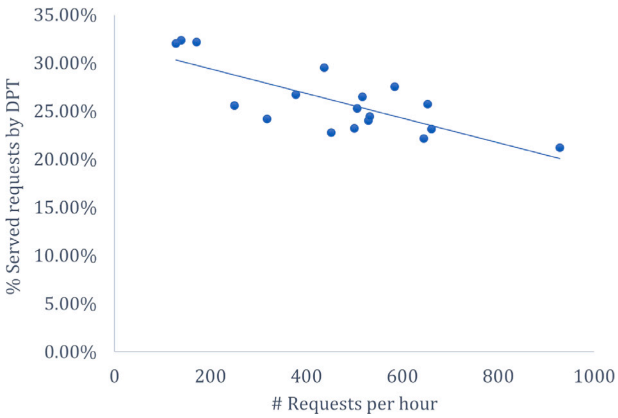

Fig. 10. Average numbers of served of ride requests per time slot during the day with the different systems.

Table 2
Performance metrics from passenger perspective.

	SPT	OSPT	SDPT	DPT
Average walking distance of successful rides (km)	0.95	0.75	0.72	0.7
Average ride duration of successful rides (min)	27.27	24.71	21.61	22.06

the ride plus the walking duration at origin and destination. We assume that passengers know the schedule ahead, so that there is no waiting time. In addition to their ability to serve more rides, the dynamic systems also serve the rides with less walking distance and ride duration compared to the static systems. Naturally, the OSPT routes reduce the walking distance compared to the SPT scenario as they are customized and optimized to the ride-hailing demand. The DPT results in a slightly longer average travel duration compared to the SDPT as it serves more passengers.

5.2.2. Sensitivity analysis of the impact of passengers' flexibility

In the second analysis, we investigate how passengers' willingness to act in a more sustainable manner and being more flexible can improve the system-wide performance. To this end, we compare the base case results of the DPT presented in the previous section to three new scenarios: (1) if every passenger is willing to walk an extra 0.5 km, (2) if every passenger is willing to arrive five minutes later, and (3) if every passenger is willing to walk an extra 0.5 km and arrive five minutes later. For the first scenario, we fix the latest arrival time. Thus, passengers are willing to walk more but are not willing to arrive later. The opposite is true in the second scenario, while the third scenario combines the two cases.

Fig. 11b shows the impact of each scenario on the percentage of requests that the DPT can serve. Compared to the base case scenario, the DPT can serve an extra 23.3% of the total requests when each passenger is willing to walk for an extra 0.5 km, and an

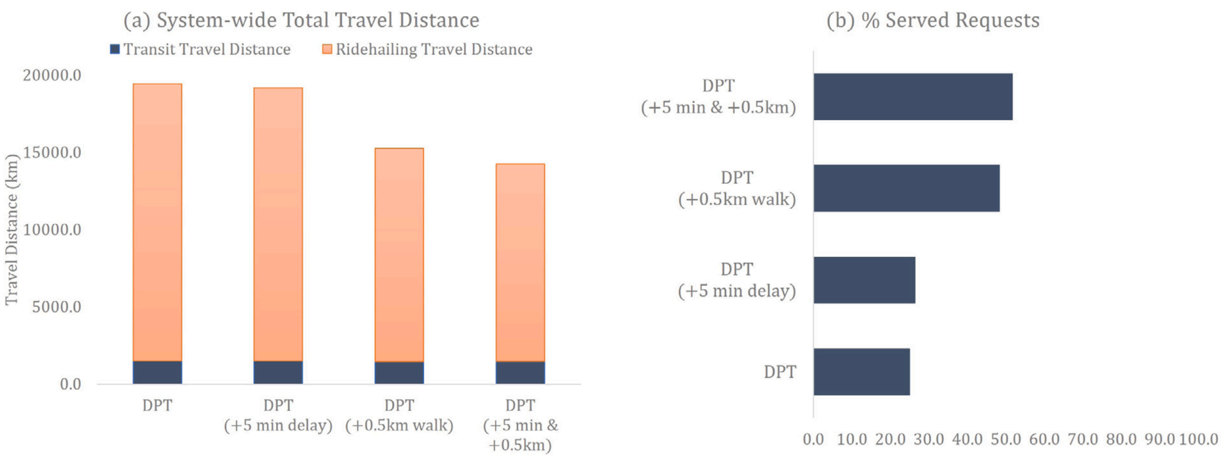

Fig. 11. Impact of passenger flexibility on the system's main performance measures: (a) transit and system-wide travel distances, (b) percentage of served on-demand ride requests.

extra 1.4% when they are willing to arrive five minutes later. In the most flexible scenario, the DPT can serve in total almost 52% of the requests compared to 25% in the base case. This increase in the number of served requests by the DPT is directly reflected in the system-wide travel distance as shown in Fig. 11a. Compared to the base instance, the system-wide travel distance can be reduced by 1%, 16%, and 20% in the three scenarios, respectively. Thus, passengers' willingness to walk more to stations has a larger impact than their flexibility with respect to the arrival time.

6. Conclusion and future work

The growing challenges facing urban mobility systems, especially in densely populated cities, require novel solutions. In this work, we provide directions for revised transit bus networks in order to contribute to the future of smart and sustainable urban mobility. We contribute to the concept of dynamic public transit system, which can provide a better service to attract more passengers to public transport instead of using private vehicles or ride-hailing services.

Our results show that the proposed dynamic network can meet and fulfill more passengers' requirements compared to the static settings. Accordingly, more passengers may choose using public transit instead of ride-hailing services which have higher fares. As a result, we showed in our case study that the DPT can reduce the system-wide travel distance by up to around 50% compared to the static routes scenario. Moreover, we show that passenger's individual behavior and preferences can have a critical impact on the system-wide performance. Namely, if all passengers are willing to be more flexible with respect to arrival time and walking distance, more passengers can be served. As a result, the system-wide travel distance can be also reduced significantly.

Our analyses emphasize several managerial and practical implications of our proposed dynamic system. It successfully tackles the first and last mile problem of the public transport services by adjusting the routes according to the observed demand. Thus, it has the potential to attract a large share of ride-hailing or private travel passengers to the transit services by taking into account and fulfilling their individual preferences. As a result, it can significantly reduce the system-wide travel distance. Additionally, the dynamic system can improve equity aspects of the static public transport services by allocating stations according to the observed demand instead of being fixed at specific locations.

Besides the practical implications, we proposed the CIOM model to solve our complicated optimization problem. We decompose our complex and large problem into smaller subproblems that are easier to be solved. Due to the complexity of the problem, further investigation and research need to be done to reach the globally optimal solution of the problem or assess the optimality gap.

In future research, we aim at studying how to allocate the cost and optimize the fare structure. The fare that each passenger pays should be a function of their ride distance and their flexibility regarding the walking distance and arrival time. In an ideal fully integrated urban mobility system, the fare of the transit services would be closely coordinated with that of ride-hailing. For example, passengers that are flexible but cannot be served by the transit service due to their distant location from any transit trip should not pay more to use ride-hailing service than those who can be served with the DPT and have the same flexibility and ride distance. Additionally, machine learning and demand prediction techniques have proven their ability in improving the performance of urban mobility systems in previous studies (Schroer et al., 2022). Thus, applying demand prediction methods could enhance the dynamic network's performance when not all passengers submit their ride requests to the system.

CRediT authorship contribution statement

Ayman Abdelwahed: Conceptualization, Methodology, Software, Development, Writing – original draft, Preparing results, Visualization. **Pieter L. van den Berg:** Conceptualization, Methodology, Writing – review & editing. **Tobias Brandt:** Conceptualization, Methodology, Writing – review & editing. **Wolfgang Ketter:** Conceptualization, Supervision, Writing – review & editing.

Appendix A. The coverage problem

In this appendix, we present the coverage problem formulation that is used in the initialization step in the CIOM algorithm. As previously mentioned, the MIP coverage problem has two different objectives for the two steps in our algorithm. This problem is based on two sets, set S of all the potential stations and set R of the requests. For the first step, the objective is to maximize the weighted number of served requests as shown in Eq. (22), where c_r is a binary variable set to 1 if request r is successfully covered and ω_r is its weight. The second objective is to minimize the distance between the active stations and the trip line which is shown in Eq. (23), v_s is a binary variable that is set to 1 if station s is active in the solution, and d_s is the orthogonal distance between station s and the trip line.

$$\max \sum_{r\in R} c_r \times \omega_r \tag{22}$$

$$\min \sum_{s\in S} v_s \times d_s \tag{23}$$

The two big-M constraints (24) and (25) are added to calculate the binary variable for coverage c_r. The left hand side of the two constraints represents the resulting walking distance where the two binary variables p_{rs} and q_{rs} are set to 1 if station s is selected to cover ride r for pickup and drop-off, respectively. k^o_{rs} and k^d_{rs} are the walking distance between the origin and the destination of request r and station s, respectively. k^m_r is the maximum walking distance specified by the passenger and M is the big-M value that is specified based on the maximum route length. Constraints (26) and (27) ensure that at maximum only one station is assigned for pickup and drop-off of each request.

$$\sum_{s\in S} p_{rs}k^o_{rs} + \sum_{s\in S} q_{rs}k^d_{rs} \geqslant k^m_r - M \times c_r \qquad \forall r \in R \tag{24}$$

$$\sum_{s\in S} p_{rs}k^o_{rs} + \sum_{s\in S} q_{rs}k^d_{rs} \leq k^m_r - M \times (1 - c_r) \qquad \forall r \in R \tag{25}$$

$$\sum_{s\in S} p_{rs} \leq 1 \qquad \forall r \in R \tag{26}$$

$$\sum_{s\in S} q_{rs} \leq 1 \qquad \forall r \in R \tag{27}$$

Finally, constraints (28) and (29) are added to evaluate the active station binary variable v_s. Constraint (30) restricts the number of active stations to the maximum allowed number of stations N.

$$v_s \geqslant p_{rs} \qquad \forall s \in S, r \in R \tag{28}$$

$$v_s \geqslant q_{rs} \qquad \forall s \in S, r \in R \tag{29}$$

$$\sum_{s\in S} v_s \leq N \tag{30}$$

Appendix B. Pseudocode of the merging phase

See Fig. 12

Appendix C. Static routes optimization

In this appendix, we present the formulation that is used to generate the optimized static routes. The static line is optimized based on the historical demand data and stations are allocated to the areas of the highest demand. The problem is formulated as a MIP problem which resembles a facility location–allocation optimization problem.

Thus, the formulation only selects the locations of stations, without routing, to maximize the captured demand. We count all the pickups and drop-offs per each tract and we assume that the demand is uniformly distributed throughout the tract. Then, for each potential station, as previously shown in Fig. 7, we assume that its coverage range is 0.6 km around it. Thus, a potential station covers a ratio of the demand of a tract that is equal to the ratio between the intersection area between the coverage range of this station and the total area of the tract. Additionally, we introduce constraints on the maximum number of stations to be included and the minimum distance between any two selected stations not to be less than 1.5 km. Furthermore, we add a constraint to avoid zigzagging (i.e., traveling in a direction orthogonal to the line direction) by disallowing the model from selecting two stations whose connecting line would make an angle that is greater than 65° with the line connecting the two terminal stations of the line.

The problem has two sets: the tracts set T with index t and the stations set S with index s. We assume that each station covers a range around it with a specific radius and that demand in each tract is uniformly distributed over its area. Thus, we can calculate the ratio of the demand per area across each tract t that is covered by each station s, which we define by C_{st}.

Our decision is to select which stations to add to the route. Thus, the binary decision variable a_s is set to one if station s is selected to be added to the route. Our objective is to maximize the demand coverage as shown in Eq. (31), where D_t is the total historical demand in tract t.

$$\sum_{t\in T} \left(\sum_{s\in S} a_s . C_{st}\right) \times D_t \tag{31}$$

- For (t=0; t≤#AllTrips; t++):
  - Initialize *Most_Recent_Solution[t]* = Solution from previous step.
  - Add *Most_Recent_Solution[t]* column to *Columns_Matrix[t].*
  - Evaluate *Base_Solution = Most_Recent_Solution[t] – All overlapped served requests of Most_Recent_Solution[t]*, and add the *Base_Solution* column to *Columns_Matrix[t].*
- For (i=0; i≤Max_Num_Iterations ; i++):
  - For (t=0; t≤#AllTrips; t++):
    - Re-evaluate *Base_Solution = Most_Recent_Solution[t] – All overlapped served requests of Most_Recent_Solution[t]*, and add the *Base_Solution* column to *Columns_Matrix[t].*
    - **Solve** the subproblem (add constraint to serve all the requests in the *Base_Solution* and maximize the number of served requests while excluding all overlapped served requests).
    - IF(new solution didn't serve any additional requests): Continue.
    - Else: #the new solution serve more requests than *Base_Solution*
      - Add the new solution as a column to the *Columns_Matrix[t].*
      - **Solve** the subproblem again while fixing the served requests at the new solution while considering the overlapped served requests to check if any of them can be added.
      - IF(more requests added): Add the solution as a column to the *Columns_Matrix[t].*
  - IF(Solutions of all the trips didn't add any new columns): Break.
  - Else: #one or more new columns were added in this iteration
    - **Solve** the master problem to select columns that maximize the number of served requests.
    - Check and extract the overlapped served requests in this iteration.
    - IF(all overlapped served requests were overlapped in the previous solution): Break
    - Else: update *Most_Recent_Solution* for each *t*.
- For (t=0; t≤#AllTrips; t++):
  - Solve the subproblem with the objective of minimizing route distance with constraint on serving all the requests in the Most_Recent_Solution.

Fig. 12. Pseudo-code of the merging phase.

Constraint (32) is added to avoid having two stations in the route if the distance between them is less than a minimum threshold distance d^{min}. Similarly, Constraint (33) is added to avoid having two stations in the route if the angle between the line connecting them and the route line is greater than a maximum threshold g^{max}. Thus, we avoid zigzagging in the route (i.e., traveling orthogonally to the route line). Constraint (34) restricts the maximum number of stations to N^{max}

$$a_s + a_s' \le 1 \qquad \forall s \in S, s' \in S, \text{if distance}(s, s') \le d^{min} \tag{32}$$

$$a_s + a_s' \le 1 \qquad \forall s \in S, s' \in S, \text{if angle(route line, line connecting}(s, s')) \geqslant g^{max} \tag{33}$$

$$\sum_{s \in S} a_s \le N^{max} \tag{34}$$

Appendix D. The column generation algorithm

In this appendix, we present the column generation problem that is used to solve our vehicle routing with time window problem. Given the sets of ride requests Q and potential routes R, the two-dimensional matrix $\boldsymbol{S_{qr}}$ represents the potential routing solutions, where each column represents a route (r) while each row represents a ride request (q), and s_{qr} is set to one if request q is served by route r. The set B represents the bus trips in the problem, and the matrix $\boldsymbol{S_{qr}}$ is initialized by the initial route solution for each bus trip b. Each column is also added to the columns' subset R^b, which contains all the columns that belong to each bus. Given the set of requests Q with total number of ride requests N^q in the problem, additional N^q single-ride trip columns are added, where each column represents a request being served by ride-hailing. Thus, $\boldsymbol{S_{qr}}$ would have initially $N_q + N_b$ columns (see Fig. 13 for example). We define the travel distance (d_r) to be the route distance in case of the bus trips and the ride-hailing trip distance in case of the single-ride trips. Thus, the objective of the master problem is to minimize the total travel distance as follows:

$$min(\sum_{r \in R} d_r x_r) \tag{35}$$

In the integer programming form of the problem, the binary decision variable x_r is set to one if route r is active in the solution. However, we solve the linear relaxation of the problem to get the dual values for each request node π_r, which will be used to

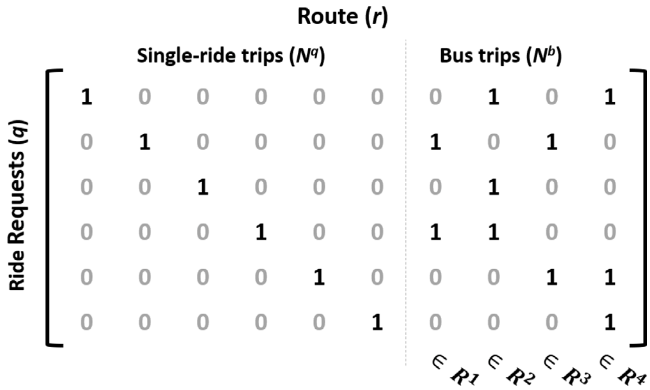

Fig. 13. Example of the column generation matrix.

calculate the modified reduced costs in the subproblem's objective function, given the following constraint:

$$\sum_{r\in R} s_{qr} x_r \qquad \forall q \in Q \tag{36}$$

The previous constraint is added to ensure that all ride requests are served. Finally, the following constraint is added to ensure that each bus trip has at most one column that is activated in the solution:

$$\sum_{r\in R^b} x_r \le 1 \qquad \forall b \in B \tag{37}$$

As the bus trips are not identical, since each trip has its own start time, the subproblem has to be solved for each bus trip separately per each iteration. The formulation of the subproblem is like the formulation presented in Section 4.1 but with a single bus trip (i.e., without the vehicle index). However, the nodes of the network (N) are now expanded such that each node is defined by three values: request, station and type (i.e., pickup or drop-off). For example, if request 1 can be picked up at stations A or B and can be dropped off at stations C or D, this results in four nodes in the formulation: (1, A, pickup), (1, B, pickup), (1, C, drop-off), (1, D, drop-off). Additionally, the objective function is changed to minimize the reduced cost as follows:

$$min(\sum_{i\in N}\sum_{j\in N} x_{ij} * (d_{ij} - \pi_i)) \tag{38}$$

where x_{ij} is the binary decision variable that is set to one if the arc i-j is active, d_{ij} is the distance between the two nodes, and π_i are the dual values extracted from the master problem. π_i is added only at the pickup nodes for each request and is set to zero at all the drop-off nodes.

The master problem and the subproblems are solved iteratively until the objective value of all the subproblems becomes positive, which means that there is no room for further improvement. Then, the master problem is solved with the integrality constraint to get the final solution.

Appendix E. Evaluating the performance of the CIOM algorithm

The goal of this analysis is to assess the performance and the added value of each step of the CIOM algorithm. Therefore, we compare its performance to two other benchmark methods: the exact MIP model and a column generation model which is widely used for the VRPTW. Although our problem cannot be exactly classified as a VRPTW, it is closest to it in the literature. However, our problem has a couple of extra layers of complexity due to considering the walking distance and the multiple potential stations for pickup and drop-off per request. Since we are the first to propose such a problem structure, there is no reference dataset to use for our comparison.

We generate 20 random instances with a planning period of one hour and four vehicle trips that are 15 min apart (i.e., with a frequency of 4 trips per hour). The number of ride requests in the instances varies from 28 to 143, with an average of 75. For a fair comparison, we predetermine the solution time limit (i.e., based on the instance size) and fix it for the three models for each instance.

We compare the CIOM algorithm to the exact MIP algorithm we presented in Section 4.1, and a column-generation algorithm. The details of the column-generation algorithm that we developed are presented in Appendix D. However, both the exact MIP and the column-generation algorithm need a warm-start to result in reasonable results. Therefore, we use the output of the initialization step of our CIOM algorithm as an initial solution for the two other models. Thus, we are comparing the CIOM to CIE (cluster-initialize-exact MIP) and CICG (cluster-initialize-column generation). We use the default CPLEX solver to solve all the MIP problems in the different algorithms.

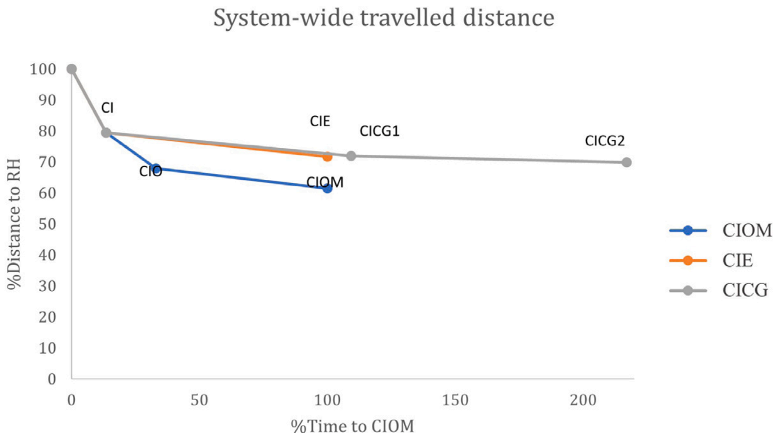

Fig. 14. The reduction in the system-wide travel distance relative to the all-ride-hailing network for the different algorithms.

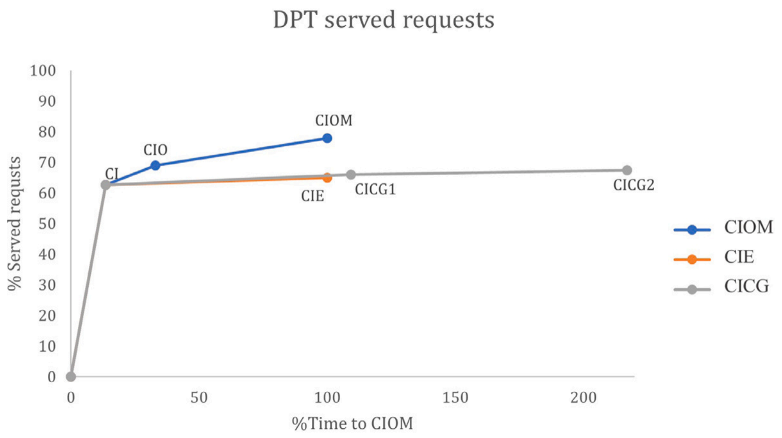

Fig. 15. The percentage of the served requests by DPT with the different algorithms.

We set a time-limit per instance that is proportional to its size (i.e., its number of requests and vehicles). In an attempt to reach the globally optimal solution, we run the CICG for a longer duration than the CIOM and the CIE. We run the CICG in two stages: in the first stage (CICG1), we run it for the same amount of time that is applied to the other two models. In the second stage (CICG2), we allow it to run for an extra period with the same amount of time (i.e., total run-time of the two stages is equal to two times the time-limit) in an attempt to reach the globally optimal solution.

Our two main metrics in the comparison are the number of served requests by the DPT and the system-wide total travel distance. The system-wide travel distance is the distance of the DPT's trips plus the distance of the ride-hailing trips for those requests which the DPT could not serve. Additionally, it is important to note that the objectives are not identical for the different models. For the CIOM and CIE, in the first step we prioritize serving more requests by the DPT. Then we minimize the DPT trips' distances while putting a constraint on the minimum number of served requests we received from the first step. On the other hand, the CICG objective is to minimize the system-wide travel distance without putting a weight on the number of served requests by the DPT. This is because the column-generation formulation optimizes the whole system considering that ride requests can be served by DPT or RH, while the CIOM and CIE optimizes for the DPT operations only.

Figs. 14 and 15 give an overview of the average performance of the three models over all the instances. The horizontal axes represent the percentage of time referenced to the total CIOM run time. Thus, the time of the CIOM is 100%. We attempt to keep the time of the other two models to the same time. This holds true for the CIE. However, for the CICG's first stage it exceeds the 100% as we do not terminate the algorithm while solving when the time-limit is reached. Instead, we allow the ongoing iteration to finish first.

Fig. 14 shows the reduction in the system-wide distance compared to the all ride-hailing (RH) scenario, in which all passengers use the ride-hailing service. For the CIOM algorithm, the initialization step (CI) takes on average around 14% of the runtime, which is equivalent to 13 min, and can reduce the system-wide distance by 20%. After the optimization (CIO) step, the system-wide distance is reduced on average by an additional 12% to reach 68%. The optimization step accounts for around 18% of the time on average, which is equivalent to 17 min. Finally, the merging step accounts on average for around 68% of the runtime and can reduce the system wide distance from 68% to 61%. The results also show that the CIOM algorithm outperforms the exact and column generation models. Within almost the same time-limit, the CIOM reduces the system-wide distance to 61% while both the exact and column-generation reduce it to 72%. Even when we run the CICG for an extra period, it can improve the solution only by an additional 2%.

Fig. 15 shows our second metric; the ratio of the requests which the DPT can serve. Once more, the results show that the CIOM outperforms its contenders. The CIOM can serve on average 78% of the total requests while the CIE and CICG averages are 65% and 66%, respectively. The CICG can serve additional 1.5% when it runs for an extra period.

References

Agarwal, S., Mani, D., Telang, R., 2019. The Impact of Ride-Hailing Services on Congestion: Evidence from Indian Cities. Indian School of Business.

Agatz, N., Erera, A., Savelsbergh, M., Wang, X., 2012. Optimization for dynamic ride-sharing: A review. European J. Oper. Res. 223 (2), 295–303.

Agatz, N., Hewitt, M., Thomas, B.W., 2021. "Make no little plans": Impactful research to solve the next generation of transportation problems. Networks 77 (2), 269–286.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proc. Natl. Acad. Sci. 114 (3), 462–467.

Babar, Y., Burtch, G., 2020. Examining the heterogeneous impact of ride-hailing services on public transit use. Inf. Syst. Res. 31 (3), 820–834.

Basu, R., Araldo, A., Akkinepally, A.P., Nahmias Biran, B.H., Basak, K., Seshadri, R., Deshmukh, N., Kumar, N., Azevedo, C.L., Ben-Akiva, M., 2018. Automated mobility-on-demand vs. mass transit: a multi-modal activity-driven agent-based simulation approach. Transp. Res. Rec. 2672 (8), 608–618.

Camporeale, R., Caggiani, L., Ottomanelli, M., 2019. Modeling horizontal and vertical equity in the public transport design problem: A case study. Transp. Res. A 125, 184–206.

Cao, Z., Ceder, A.A., 2019. Autonomous shuttle bus service timetabling and vehicle scheduling using skip-stop tactic. Transp. Res. C 102, 370–395.

Chen, P.W., Nie, Y.M., 2017. Analysis of an idealized system of demand adaptive paired-line hybrid transit. Transp. Res. B 102, 38–54.

Chetty, R., Hendren, N., Kline, P., Saez, E., 2014. Where is the land of opportunity? The geography of intergenerational mobility in the United States. Q. J. Econ. 129 (4), 1553–1623.

Chicago Data Portal, 2020. Chicago transportation network providers - Trips. https://data.cityofchicago.org/Transportation/Transportation-Network-Providers-Trips/m6dm-c72p. (Accessed 01 May 2020).

Cohen-Blankshtain, G., Rotem-Mindali, O., 2016. Key research themes on ICT and sustainable urban mobility. Int. J. Sustain. Transp. 10 (1), 9–17.

Errico, F., Crainic, T.G., Malucelli, F., Nonato, M., 2013. A survey on planning semi-flexible transit systems: Methodological issues and a unifying framework. Transp. Res. C 36, 324–338.

Errico, F., Crainic, T.G., Malucelli, F., Nonato, M., 2021. The single-line design problem for demand-adaptive transit systems: a modeling framework and decomposition approach for the stationary-demand case. Transp. Sci. 55 (6), 1300–1321.

Gao, L., Yu, Y., Liang, W., 2016. Public transit customer satisfaction dimensions discovery from online reviews. Urban Rail Transit 2 (3–4), 146–152.

Gong, J., Greenwood, B.N., Song, Y.A., 2017. Uber might buy me a mercedes benz: An empirical investigation of the sharing economy and durable goods purchase. Available at SSRN 2971072.

Lyons, G., 2018. Getting smart about urban mobility–aligning the paradigms of smart and sustainable. Transp. Res. A 115, 4–14.

Malhotra, A., Melville, N.P., Watson, R.T., 2013. Spurring impactful research on information systems for environmental sustainability. MIS Q. 37 (4), 1265–1274.

Mourad, A., Puchinger, J., Chu, C., 2019. A survey of models and algorithms for optimizing shared mobility. Transp. Res. B 123, 323–346.

Narayanan, S., Chaniotakis, E., Antoniou, C., 2020. Shared autonomous vehicle services: A comprehensive review. Transp. Res. C 111, 255–293.

Nejat, P., Jomehzadeh, F., Taheri, M.M., Gohari, M., Majid, M.Z.A., 2015. A global review of energy consumption, CO2 emissions and policy in the residential sector (with an overview of the top ten CO2 emitting countries). Renew. Sustain. Energy Rev. 43, 843–862.

Ng, M.T., Mahmassani, H.S., 2022. Autonomous minibus service with semi-on-demand routes in grid networks. Transp. Res. Rec. 03611981221098660.

Nourbakhsh, S.M., Ouyang, Y., 2012. A structured flexible transit system for low demand areas. Transp. Res. B 46 (1), 204–216.

Paulley, N., Balcombe, R., Mackett, R., Titheridge, H., Preston, J., Wardman, M., Shires, J., White, P., 2006. The demand for public transport: The effects of fares, quality of service, income and car ownership. Transp. Policy 13 (4), 295–306.

Pinto, H.K., Hyland, M.F., Mahmassani, H.S., Verbas, I.Ö., 2020. Joint design of multimodal transit networks and shared autonomous mobility fleets. Transp. Res. C 113, 2–20.

Qiu, F., Li, W., Zhang, J., 2014. A dynamic station strategy to improve the performance of flex-route transit services. Transp. Res. C 48, 229–240.

Quadrifoglio, L., Dessouky, M.M., Ordóñez, F., 2008. Mobility allowance shuttle transit (MAST) services: MIP formulation and strengthening with logic constraints. European J. Oper. Res. 185 (2), 481–494.

Redman, L., Friman, M., Gärling, T., Hartig, T., 2013. Quality attributes of public transport that attract car users: A research review. Transp. Policy 25, 119–127.

Rosenberg, M., 2016. This is how much more you'd have to pay for a home near light rail. The Settle Times, https://www.seattletimes.com/business/study-to-live-near-transit-in-seattle-youll-have-to-pay-up/. (Accessed 26 June 2021).

Sayarshad, H.R., Gao, H.O., 2020. Optimizing dynamic switching between fixed and flexible transit services with an idle-vehicle relocation strategy and reductions in emissions. Transp. Res. A 135, 198–214.

Schroer, K., Ketter, W., Lee, T., Gupta, A., Kahlen, M., 2022. Data-driven competitor-aware positioning in on-demand vehicle rental networks. Transp. Sci. 56 (1), 182–200.

Shaheen, S., Chan, N., 2016. Mobility and the sharing economy: Potential to facilitate the first-and last-mile public transit connections. Built Environ. 42 (4), 573–588.

Silva, B.N., Khan, M., Han, K., 2018. Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. Sustainable Cities Soc. 38, 697–713.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2015. The benefits of meeting points in ride-sharing systems. Transp. Res. B 82, 36–53.

Stiglic, M., Agatz, N., Savelsbergh, M., Gradisar, M., 2018. Enhancing urban mobility: Integrating ride-sharing and public transit. Comput. Oper. Res. 90, 12–21.

Taylor, B.D., Fink, C.N., 2003. The Factors Influencing Transit Ridership: a Review and Analysis of the Ridership Literature. UC Berkeley: University of California Transportation Center.

Tirachini, A., 2019. Ride-hailing, travel behaviour and sustainable mobility: an international review. Transportation 1–37.

Tong, L.C., Zhou, L., Liu, J., Zhou, X., 2017. Customized bus service design for jointly optimizing passenger-to-vehicle assignment and vehicle routing. Transp. Res. C 85, 451–475.

Transportation Research Board and National Academies of Sciences, Engineering, and Medicine, 2012. Ridesharing as a Complement to Transit. The National Academies Press, Washington, DC, http://dx.doi.org/10.17226/14655.

United Nations, 2016. The 17 sustainable development goals. URL https://sdgs.un.org/goals.

Wefering, F., Rupprecht, S., Bührmann, S., Böhler-Baedeker, S., 2013. Guidelines. developing and implementing a sustainable urban mobility plan. In: Workshop, Mar. p. 117.

Yigitcanlar, T., Kamruzzaman, M., Foth, M., Sabatini-Marques, J., da Costa, E., Ioppolo, G., 2019. Can cities become smart without being sustainable? A systematic review of the literature. Sustainable Cities Soc. 45, 348–365.
Zhang, Y., Zhang, Y., 2018. Exploring the relationship between ridesharing and public transit use in the United States. Int. J. Environ. Res. Public Health 15 (8), 1763.
Zhao, J., Dessouky, M., 2008. Service capacity design problems for mobility allowance shuttle transit systems. Transp. Res. B 42 (2), 135–146.