Problem decomposition and approximation for shared mobility applications with endogenous congestion: integrated vehicle assignment and routing in capacitated transportation networks

Jiangtao Liu
School of Sustainable Engineering and the Built Environment,
Arizona State University, Tempe, AZ, 85281, USA
Email: jliu215@asu.edu

Pitu Mirchandani
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University, Tempe, AZ, 85281, USA
Email: pitu@asu.edu

Xuesong Zhou
School of Sustainable Engineering and the Built Environment,
Arizona State University, Tempe, AZ, 85281, USA
Email: xzhou74@asu.edu
Tel.: +1 480 9655827
(Corresponding Author)

Submitted for publication in Transportation Research Part C

Abstract

The ridesharing services have been growing in recent years with the start of network service companies, and will be further enhanced by the recently emerging trend of autonomous vehicles applications for future traveler mobility. One fundamental question for transportation managers to answer is how to capture the endogenous traffic patterns with those upcoming new and uncertain elements for future transportation planning and management. Therefore, this paper aims to integrate travel demand, vehicle supply and limited infrastructure supply by optimally assigning available rideshared and autonomous vehicles from different depots to satisfy passengers' trips with specific pickup and drop-off requests while considering the endogenous congestion in capacitated networks. A number of decomposition approaches are adopted in this research. Focusing on this primal problem, we propose an arc-based agent-based integer linear programming model in space-time-state (STS) networks, which is solved by Dantzig-Wolfe decomposition. From the perspective of dynamic traffic assignment, a space-time-state (STS) path-based flow-based linear programming model is also provided as an approximation according to the mapping information between vehicle and passenger and between vehicle and space-time arc in each STS path in our priori generated column pool. Later, we apply Alternating Direction Method of Multipliers (ADMM) to solve this linear programming model and compare its results with standard solver. Finally, a series of numerical experiments are performed to demonstrate our proposed modeling methodology and algorithms.

Keywords

Shared mobility; Endogenous congestion; Autonomous vehicle routing; Alternating Direction Method of Multipliers (ADMM); Dantzig-Wolfe decomposition; Column pool;

1. Introduction

The transportation sector is experiencing an unprecedented revolution with the emerging advanced sensing, telecommunications and vehicular technologies, which are driving a new wave of rich information and providing a great opportunity to better control and optimize transportation system operations. On the other hand, it results in great challenges to estimate and predict their impacts on existing congested roadway infrastructure and future transportation system design, because any changes in a complex interdependent system may invoke a series of hardly predictable reactions of endogenous variables. There has been a number of studies in transportation area to consider the challenges caused by endogenous factors. For example, the bottleneck model (Vickrey, 1969) of congestion with endogenous scheduling has been used to address a number of challenging transportation economics problems, such as, time pattern of congestion, optimal pricing, unpriced equilibria (Samll, 1982; Small and Verhoef, 2007; Small, 2015). In addition, Batarce and Ivaldi (2014) formulated a structural travel demand model with endogenous congestions, Chow and Recker (2016) proposed an estimation model with endogenous arrive time constraints for better calibrating the household activity pattern problem, and Liu et al. (2018) considered the endogenous road congestions for finding household activity patterns. Therefore, one fundamental question for transportation managers to answer is how to capture the endogenous traffic patterns with those upcoming new and uncertain elements for future transportation planning and management.

The uncertain elements in future transportation systems spread over travel demand, vehicle supply and infrastructures, ranging from specifically personal trip requirements (Rayle et al., 2016; Davidson and Spinoulas, 2016; Lim et al., 2018; Ma et al., 2018; Tong et al., 2019), vehicle driving modes (Chen et al., 2017; Wong et al., 2017), route behavior (Levin et al., 2017; Wong et al., 2017; Hyland and Mahmassani, 2018; Tong et al., 2019), vehicle ownership (Davidson and Spinoulas, 2016; Lavieri et al., 2017; Allahviranloo and Chow, 2019), to infrastructures capacity changes due to sensor and communication advancement (Varaiya 1993; Papadimitratos et al., 2009; Qu et al., 2010; Gentili and Mirchandani, 2012; Mahmassani, 2016; Dey et al., 2016; Din et al., 2019).

Various approaches to respond to these are being studied, most notably the recent Intelligent Transportation Systems (ITS) initiatives of USDOT (e.g., the recent Dynamic Mobility Applications and Active Transportation Demand Management (DMA-ATDM); and Connected Vehicles Programs). In addition, since 2014 in Hannover, Germany, the concept of Mobility as a service (MaaS) is presented to improve transportation system efficiency and enabled by combining transportation services from public and private transportation providers through a unified gateway that creates and manages the trip, which users can pay for with a single account (Heikkilä, 2014; Kamargianni et al., 2016; Hensher, 2017; Jittrapirom et al.,2017; Mulley, 2017; Djavadian and Chow, 2017; Bruun, 2018; Tong et al., 2019). What is largely missing in studies of implementing large fleets of rideshared vehicles, as autonomous vehicles potentially become increasingly popular among ride providers and freight operators, and perhaps as some personal vehicles, is addressing of how these vehicles can be operated to meet temporally and spatially distributed traveler mobility needs with a limited road expansion budget and constrained infrastructure capacity. In other words, it is crucial to study the integration of travel demand, vehicle supply and infrastructure capacity under one unified modeling framework.

The interplay of travel demand and vehicle supply is close to traditional vehicle routing problems (Toth and Vigo, 2002) which are modeled in virtual point-to-point networks without physical road traffics. The link travel time is either constant or time-dependent based on externally observed historical, real-time or predicted traffic congestions (Taniguchi and Shimamoto, 2004; Kok et al., 2012). Meanwhile, the interaction of vehicle supply and physical networks is usually used to model its internal road congestion as traditional dynamic traffic assignment problems (Peeta and Ziliaskopoulos, 2001), which treats vehicles equal to its carried passengers at origin zones. This paper aims to consider demand, vehicle and infrastructure simultaneously to capture the endogenous traffic pattern with a number of assumptions. (i)

All vehicles are autonomous or can be systematically guided by one management center. (ii) Vehicles have carrying capacity and depart from its origin depots to destination depots with specific working time windows. (iii) All passengers submit their trip requests with pickup and/or drop-off locations and time windows and accept the ridesharing service. (vi) The road congestion is endogenously incurred by those guided vehicles.

1.1 Challenges

The modeling approach for our problem with route coordination, ridesharing services and constrained road capacity is close to the integration of vehicle routing problem (VPR) and dynamic traffic assignment (DTA), so this section aims to summarize the challenges of each separated problems and their integrations at first.

Without considering the endogenous road congestions, the first set of problems in our research context can be simplified as the vehicle routing problem with pickup and delivery with time windows (VRPPDTW), which has been proved to be NP-hard (Baldacci, et al., 2011). The difficulty of this problem arises from the complex categories of constraints, (i) vehicle flow balance, (ii) the logic of passenger pickup and drop-off by the same vehicle within the required time windows, and (iii) dynamic vehicle carrying capacity under ridesharing choice. In particular, it is sometimes challenging to even find a feasible solution due to the complicated interaction of all constraints. Recently, Psaraftis et al. (2016) summarized the research of the last three decades and offered a systematic classification of dynamic vehicle routing problem according to 11 criteria.

Focusing on the road congestion incurred by those assigned autonomous vehicles, the tight link capacity limitation at each time point could greatly make a large number of side constraints. If the queue spillback and kinematic waves (Newell, 1993; Daganzo, 1994) are further considered, the complex interaction among vehicles makes the problem more challenging. A detailed discussion about the connection between different traffic flow models can be found in the paper by (Zhang et al., 2013). Even in traditional dynamic traffic assignment models without considering passenger pickup and drop-off requests, it is still difficult to calculate the path marginal cost in congested networks to reach the system optimal goals (Ghali and Simth, 1995; Peeta and Mahmassani, 1995; Shen et al., 2007; Qian et al., 2012; Lu et al., 2016), especially having overlapped paths in large scale networks. In addition, Kalifates (2010) proposed a graph theoretic modeling framework with cell transmission model for generalized transportation systems to reduce the problem complexity. The current mathematically tractable solutions (Arnott et al., 1990; Yang and Huang, 2005; Munoz and Laval, 2006) mainly apply in parallel networks with a single bottleneck originally studied in the paper (Vickrey, 1969).

From the perspective of practice, simulation approaches are usually selected in dynamic traffic assignment problems to capture the road congestion with queue spillback and first-in-first-out (FIFO) rule. However, those approaches cannot explicitly handle the personalized user requests without optimization techniques. Therefore, the integration of simulation and optimization would be the trend to serve the future urban mobility systems with connected autonomous vehicles and ridesharing services.

It has been a long time to study the autonomous control of vehicles (Reece and Shafer, 1993) and automated intelligent vehicle/highway system design (Varaiya, 1993) for increasing the system safety, efficiency and reliability by using simulated environment and optimization techniques (Hanebutte et al., 1998; Van Arem et al., 2006; Talebpour and Mahmassani, 2016; Chen et al., 2017; Sun et al., 2017; Ghiasi et al., 2017; Ye and Yamamoto, 2018; Stern et al., 2018). On the other hand, there are recently a large number of studies that focus on the impacts of shared-use mobility on future transportation systems with autonomous vehicles. Behrisch et al. (2011) developed an open-source traffic simulation package (SUMO) for the simulation of urban mobility with automated driving and flexible traffic management strategy evaluations. Fagnant and Kockelman (2014) developed an agent-based simulation model in a grid-based urban area where some strategies are provided to match passengers with vehicles and relocate vehicles to reduce traveler waiting time, but the endogenous road congestion and vehicle carrying capacity for ridesharing are not considered. Mahmoudi and Zhou (2016) proposed a space-time-state modeling framework to model vehicle routing problem with pickup and delivery with time windows and solved it by Lagrangian relaxation. Zhou et al., (2017) further developed an open-source engine (VRPLite), which can be extended to many general vehicle routing problems. Levin et al. (2017) proposed a modeling framework to (i) capture the traffic congestion by simulation-based network loading based on the updated flow-density diagram with autonomous vehicles (Levin and Boyles, 2016) and (ii) serve the ride-sharing services by some heuristic algorithms. Maciejewski and Bischoff (2017) analyzed the impact of autonomous taxi on traffic congestion based on their dynamic vehicle routing problem under an agent-based simulation environment. Alonso-Mora et al. (2017) studied the real-time ride-sharing problem with high-capacity vehicles and large number of trips by dynamically generating the optimal route for the online demand and available vehicles with high-quality solutions, but as the traditional vehicle routing problems, the road congestion is not embedded in the models. Hyland and Mahmassani (2018) focused on the on-demand shared-use autonomous vehicle mobility services (SAMS) without shared rides by proposing six vehicle-to-passenger assignment strategies tested in an agent-based simulation tool. Ma et al. (2017) proposed a linear programming model to assign available autonomous vehicles to satisfy those trip requests by constructing its feasible service network in advance, but the ride-sharing option and road congestion are not considered as well. In congested networks, Rossi et al. (2018) studied the autonomous vehicle routing and rebalancing, and Salazar et al. (2018) considered how to best assign travelers between autonomous vehicles in traffic systems and public transit vehicles, but the road congestion is simplified by flow-based travel cost function, which is typically used for long-term transportation planning rather than short-term traffic operations, so it could affect the accuracy of estimated congestions. Focusing on the household activity pattern problem, Liu et al. (2018) first formulated the endogenous road congestion in a point queue model, caused by household owned vehicles that are assigned perform different mandatory or optional household daily activities. Recently, Tong et al. (2019) offered a modeling framework in an open-source simulation engine (DTALite-S) to incorporate agent-based simulation and optimization in a multimodal transportation environment with different trip requests to capture complex traffic dynamics. Mourad et al. (2019) provided a survey on models and algorithms of shared mobility systems and it can be observed that the endogenous congestion is rarely considered in current optimization models.

1.2 Problem Decomposition approaches

With the development of computer hardware, the computation capabilities for solving mathematical programming models evolves very quickly. However, many large-scale problems still lead to formulation that greatly goes beyond the computation limit. One usual way is to find the special blocks in formulation to directly decompose models as relatively solvable subproblems connected by coupling constraints. The decomposition can be classified two categories, primal decomposition and dual decomposition (Boyd et al., 2007; Palomar and Chiang, 2006), where the dual price in primal decomposition or the Lagrangian multiplier in dual decomposition for the coupling constraints is used to update and control subproblems. In the primal decomposition, column generation and Dantzig-Wolfe decomposition is widely used for linear programming and mixed integer programming with branch-and-price methods (Barnhart et al., 1998). Lagrangian relaxation/decomposition (Fisher, 1981; Mahmoudi and Zhou, 2016; Wu et al., 2019) is usually used for integer programming from the dual perspective. Huisman et al. (2005) summarized that the dual price in linear programming (LP) relaxed restricted master problem in column generation can be replaced by the Lagrangian multiplier in its LP relaxed dual problem through Lagrangian relaxation without using branch and price. It should be noted that finding a feasible initial solution in primal decomposition or a good feasible final solution in dual decomposition sometimes are still challenging in large-scale complicated problems, especially having different categories of side constraints. Also, how to address the non-unique dual prices or Lagrangian multipliers is important for the quality of solutions in the iterative process. In addition, from the perspective of primal-dual algorithm, the alternating direction method of multipliers (ADMM) (Boyd, 2011) can be used to decompose the overall problem as a number of sequentially connected subproblems, which are controlled by Lagrangian multipliers. Therefore, ADMM is also viewed as an efficient way to break the symmetry issues compared with Lagrangian relaxation. Recently, to address the stochastic mixed-integer programming models, Boland et al. (2018) applied ADMM to this problem and use the Frank-Wolfe method based on simplicial decomposition to deal with the nonlinear objective functions of subproblems, and then showed their approach theoretically supported, computationally efficient, and parallelizable.

To solve the traditional static traffic assignment problem (Wardrop, 1952; Beckmann et al., 1956), LeBlanc et al. (1975) offered a linearization algorithm to solve the classical model based on the Frank-Wolfe algorithm (Frank and Wolfe, 1956). An important improvement of Frank-Wolfe algorithm is simplicial decomposition, which is a special version of the Dantzig-Wolfe decomposition principle based on Carathéodory's theorem, where extreme points are usually generated by the solution of the linear Frank-Wolfe. Larsson and Pattrikson (1992) proposed a disaggregate simplicial decomposition (DSD) which treats each path of OD pairs as one extreme point rather than the network flow solution in simplicial decomposition (SD). Then Larsson et al. (2004) focused on the side constrained traffic equilibrium problem, which is solved by column generation based on their DSD approach. Moving forward to dynamic traffic assignment, a similar disaggregated simplicial decomposition is also used for gap-function-based user equilibrium (Lu et al., 2009) and eco-system optimum (Lu et al., 2016) with different traffic flow models. In addition, based on the cell transmission model, Dantzig-Wolfe decomposition was also used to solve system optimal (Li et al., 2003) and user optimal (Lin et al., 2010). Focusing on vehicle routing problems with ride-sharing services, Lagrangain relaxation is also used decompose the problem as a number of shortest path finding problems (Mahmoudi and Zhou, 2016; Liu et al., 2018). Recently, Yao et al. (2018) applied ADMM to solve the vehicle routing problem with drop-off requests only in the context of urban logistics to show the solution performance from the primal and dual aspects.

1.3 Potential contributions and structure of this paper

As stated by Gendreau et al. (2016), spatial and temporal behavior of traffic variations should be analyzed, but still is an enormous challenge requiring consolidating knowledge from various disciplines (traffic flow theory, statistics, etc.) for vehicle routing problems with stochastic travel time (VRPSTT). This kind of challenges indeed mainly arises from the endogenous congestions among moving vehicles in capacitated transportation systems. Focusing on the specific scenario stated before, the contributions of our research are listed as follows.

(1) Compared with the literature about vehicle routing problem and dynamic traffic assignment, this paper takes a further step to integrate travel demand, vehicle supply and infrastructure supply to explicitly capture the new traffic condition. Specifically, it aims to optimally assign vehicles from different depots to satisfy individuals' temporally and spatially distributed mobility requests under constrained road capacity and queue spillbacks.

(2) Due to the complexity of this problem, a primal decomposition approach, Dantzig-Wolfe decomposition, is used to decompose the proposed space-time-state (STS) arc-based agent-based integer linear programming model as a restricted master problem and a number of subproblems. The subproblems can be independently solved for each vehicle by time-dependent state-dependent shortest path algorithms. To our best knowledge, this is the first to use a three-dimensional STS path as an extreme point in Dantzig-Wolfe decomposition, compared with previous research using physical paths or space-time paths.

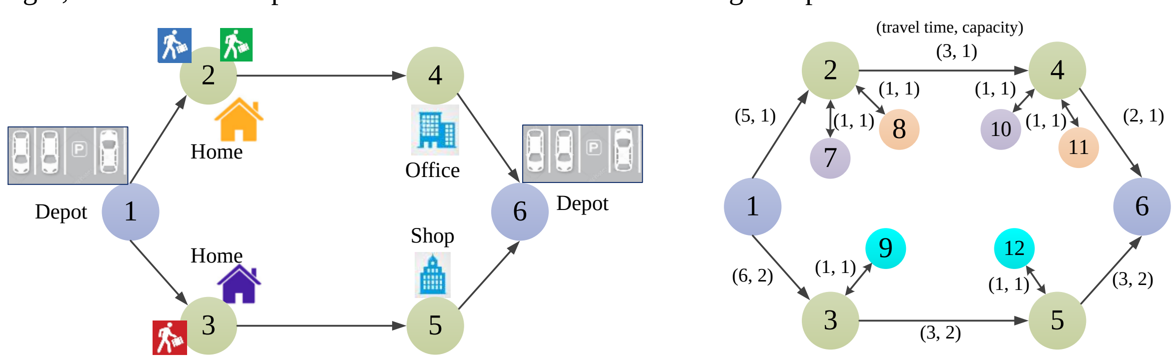
(3) As an approximation from the perspective of dynamic traffic assignment, a STS path-based flow-based linear programming model is proposed by building a column pool in advance. ADMM is then applied to decompose this model as a number of sequential quadratic programming subproblems solved by projected gradient method for each column. Specifically, each column represents a space-time-state path with the vehicle-to-passenger assignment and vehicle-to-arc assignment information through solving the primal arc-based agent-based model in a sampling dataset by ADMM, which can also decompose the primal problem as a number of sequential time-dependent state-dependent shortest path problems for each vehicle.

The remainder of this paper is organized in the following manner. Section 2 formally state our focused problem and conceptually illustrates our modeling approach in a space-time-state network. Section 3 provides an arc-based agent-based integer linear programming model (**Model 1**) and a STS path-based flow-based linear programming model (**Model 2**), which are decomposed and solved by Dantzig-Wolfe decomposition and column-pool-based approximation approach, respectively, in Section 4. Section 5 further discusses the modeling on the queue spillback. Numerical experiments are performed to demonstrate our proposed methodology and algorithms in Section 6. Finally, our future research is discussed in Section 7.

2. Problem statement and illustrative example

Consider a physical transportation network with a set of nodes N and a set of links L. Each link can be denoted as a directed link (i, j) from upstream node i to downstream node j with a given free-flow link travel time $TT_{i,j}$and link capacity $Cap_{i,j}$. Each vehicle a has an origin depot o_a and a destination depot d_a with a specific departure time window $[l_a, m_a]$ and an arrival time window $[l'_a, m_a{}']$. The number of seats in vehicle a is Cap_a and is also named vehicle carrying capacity. In addition, each passenger will submit his/her trip requests with origin o_p, destination d_p, departure time window $[l_p, m_p]$ and arrival time window $[l'_p, m_p{}']$. Our problem aims to optimally assign each vehicle to meet those passengers' requests while considering the road capacity constraint.

As shown in Fig. 1(a), assume that 2 travelers plan to go to office (node 4) from home (node 2) and 1 traveler wants to go shopping (node 5) from home (node 3). They all have a specific departure time window and arrival time window. There are a number of available autonomous vehicles at different depots waiting for being dispatched to serve those time-dependent travel requests. Since the vehicle fleet size is probably large enough to incur traffic congestion, the physical traffic network with specific road capacities should be not neglected. For the modeling needs, a pick-up virtual node and a drop-off virtual node will be added at each passenger's pick-up and drop-off locations, respectively, as shown in Fig. 1(b). As a result, only if the vehicle visits those virtual nodes, the passengers' served status and vehicle carrying state can have changes, which will be explained in detail later when constructing the space-time-state network.



(a) Physical transportation network with vehicles and trip requests

(b) Modified transportation network with virtual pickup and delivery nodes and links

Fig. 1 The physical and modified transportation networks

In order to take account into the time dimension, a space-time network is employed to explain how to model the time window and road capacity at first, and then we will focus on a space-time-state network for modeling the whole process of our problem. Taking the physical path $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$ in Fig.1 (b) as an example, its corresponding space-time network is built in Fig. 2. Each node i is extended as a set of vertices (i, t) at each time interval and each link (i, j) is extended as a set of arcs (i, j, t, s) from vertex (i, t) to vertex (j, s). The arc capacity is derived based on the hourly link capacity and the number of intervals in one hour. In addition, the arc $(i, i, t, t+1)$ from vertex (i, t) to vertex $(i, t+1)$ means that vehicles can wait at node i at time t for one time interval in case the downstream arcs don't have enough capacity to accommodate them. The capacity of waiting arcs is infinite, so the queuing process will be similar to the point queue traffic flow model. As shown in Fig. 2, passengers 1 and 2 request the same pickup and delivery time windows. Assume the carrying capacity of vehicles is always 1.

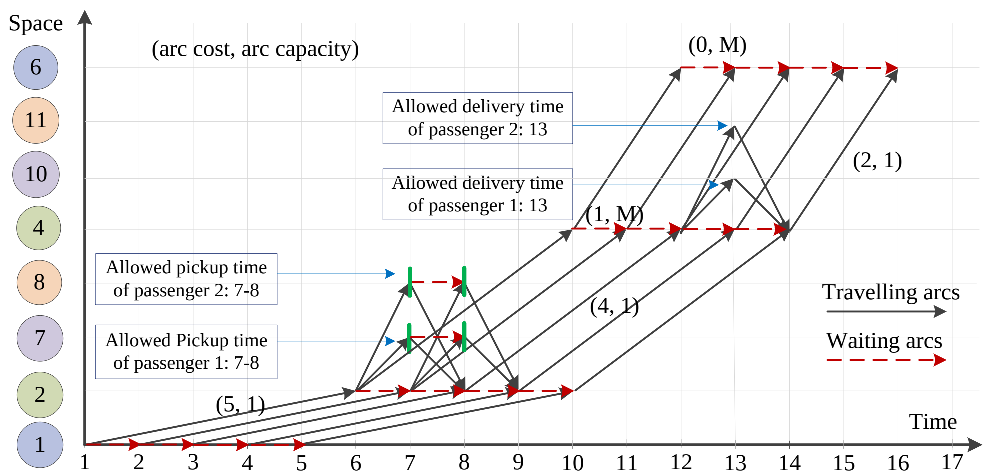


Fig. 2 The physical and modified transportation networks

We consider the following two cases without/with road capacity.

Case 1: the road capacity is not considered. Then it needs 2 vehicles departing at time 1 to satisfy those trip requests.

Case 2: the road capacity is strictly constrained. Then one vehicle has to wait at the depot until time 2 to depart. Moreover, one passenger cannot be served, because the vehicle waiting at depot can pick up one passenger but cannot delivery him/her at the allowed time due to tight road capacity constraints.

Therefore, it shows the difference and difficulty of finding the best vehicle assignment and routing solutions under tight physical facility limitations, compared with the traditional vehicle routing problems. This example in the space-time network only illustrates the general process of vehicle routing under road capacity constraints, but it doesn't consider (i) how to guarantee that once one passenger is picked up, he/she must be dropped off by the same vehicle and (ii) that the vehicle carrying capacity (the number of seats) cannot be violated due to the ridesharing options.

In order to model the process of passenger's pickup and delivery, the cumulative passenger served status is introduced and defined as follows, 0: the passenger is not served, 1: under being served (picked up but not delivered), 2: finished (delivered). In addition to the dimensions of space and time, we introduce one more dimension w as vehicle carrying state to record which passenger is being served during the vehicle routing process. If passenger p is picked up by vehicle v with the carrying capacity of 1, the carrying state of vehicle v is $p[1]$, where the first number is passenger number p and the number in square bracket mean the cumulative passenger served status of passengers. Still, focusing on the case in Fig. 2, if the vehicle capacity is 1, the possible vehicle carrying states include be ( ), (1[1]), (1[2]), (2[1]) or (2[2]). Similarly, if the vehicle capacity is 2, one possible vehicle carrying state example could be (1[1] _2[1]), which represents that passengers 1 and 2 are currently picked up but not dropped off in the vehicle. Therefore, we construct a three-dimension space-time-state network for vehicle routing, where each vertex is (i, t, w) and each arc is (i, j, t, s, w, w') from vertex (i, t, w) to vertex (j, s, w').

The vehicle carrying state transition process (state w to state w') is highly connected with the space (location) and time. Specifically, the vehicle carrying state will change when the vehicle picks up or drop off one passenger while ensuring that the vehicle carrying capacity is not violated. Note that once one passenger is served with a cumulative served status as 2, it doesn't allow to be served again, so there is no circle being selected in the state transition graph. The connection among state, space and time is the foundation of our constructed space-time-state networks. Section 4 will illustrate how to dynamically build this three-dimension network to find the time-dependent state-dependent shortest path for each vehicle. Actually, it is also possible to build the whole three-dimension network in advance based on the relation of space, time, and state, but the complexity will be explored in large-scale networks. Fig. 3 shows one feasible STS vehicle trajectory along the physical path $1 \rightarrow 2 \rightarrow 4 \rightarrow 6$. This STS path contains the vehicle-to-passenger mapping information when vehicle carrying states get changed at the pickup and drop-off locations within the required time windows, and also has the vehicle-to-arc mapping information in the space-time dimension. This kind of mapping information among vehicle, passenger and arc will be used in sections 3.2 and 4.2 for flow-based models.

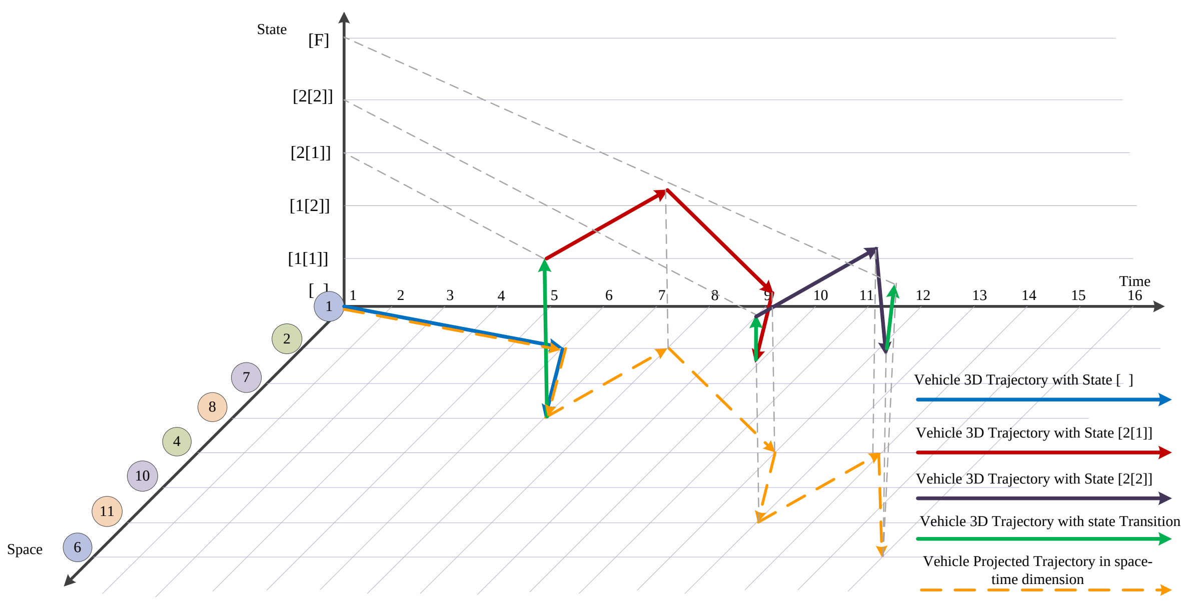


Fig. 3 A vehicle STS path with mapping information among vehicle, passenger and arc

As a remark, this modeling framework can be extended to the first/last mile problem where (i) passengers and vehicles have the same destination/origin and (ii) either pickup service or drop-off service is only considered. The difference is only about the state definition. We just need 0 and 1 for passenger service status to indicate if he/she is served by the vehicle or not.

3. Mathematical Models

Table 1 lists the general indices, sets, parameters and variables used in our proposed arc-based agent-based model (**Model 1**) in Section 3.1, where each vehicle and each passenger are represented as vehicle a and passenger p, respectively. The notation of STS path-based flow-based model (**Model 2**) is listed in Table 2 where each vehicle group v has a number of vehicles with same origin depot, departure time window, destination depot and arrival time window, and each passenger group q has a number of passengers with same trip requests (pickup location and time window and/or drop-off location and time window). Some parameters in Table 1 can also be used for the flow-based model.

Table 1 Indices, sets, parameters and variables in **Model 1**

Symbols	Definition
Indices	
i, j	Index of nodes, $i, j \in N$
(i, j)	Index of physical link between two adjacent nodes, $(i, j) \in L$
a	Index of vehicles
p	Index of passengers
t, s	Index of time intervals in the space-time network
w, w'	Index of vehicle carrying state
Sets	
N	Set of nodes in the physical traffic network
L	Set of links in the physical traffic network
L_{inflow}	Set of inflow links of single links in the physical traffic network
$L_{outflow}$	Set of outflow links of single links in the physical traffic network
P	Set of passengers
A	Set of vertices in the space-time-state network
A_p	Set of vertices of passenger p's pickup location
A_m	Set of vertices at the merge point in the space-time-state network
E	Set of edges/arcs in the space-time-state network
Parameters	
$o(a)$	Index of origin depot of vehicle a
$d(a)$	Index of destination depot of vehicle a
$DT(a)$	Earliest departure time of vehicle a, equal to l_a
$[l_a, m_a]$	Departure time window of vehicle a at the origin depot
$[l'_a, m'_a]$	Arrival time window of vehicle a at the destination depot
$[l_p, m_p]$	Departure time window of passenger p at the origin
$[l'_p, m'_p]$	Arrival time window of passenger p at the destination
$VCap_a$	Carrying capacity of vehicle a
$Cap_{i,j,t,s}$	Capacity of arc (i, j, t, s) in the space-time network
$c^a_{i,j,t,s,w,w'}$	Travel cost of arc (i, j, t, s, w, w') of vehicle a
$\delta^{a,k}_{i,j,t,s,w,w'}$	Incidence between passenger pickup arc (i, j, t, s, w, w') and path k of vehicle $= 1$, matched; otherwise, $= 0$.
$\delta^a_{k,p}$	Incidence between passenger p and path k of vehicle $= 1$, matched; otherwise, $= 0$.
$\beta^{a,k}_{i,j,t,s}$	Incidence between arc (i, j, t, s) and path k of vehicle $= 1$, matched; otherwise, $= 0$.
$FFTT_{i,j}$	Free-flow travel time of link (i, j)
$n_{i,j}$	The number of lanes on link (i, j)
$KJam_{i,j}$	Jam density of link (i, j)
Variables	
$x^a_{i,j,t,s,w,w'}$	Binary variable, vehicle a will choose arc (i, j, t, s, w, w') or not in the space-time-state network
λ^a_k	Binary variable, vehicle a will choose path k or not in the space-time-state network
$y_{j',j,t,t+1}$	Positive integer variable, the outflow arc capacity on arc $(j', j, t, t+1)$
π_p	Lagrangian multiplier of passenger p's trip request

$\pi_{i,j,t,s}$	Lagrangian multiplier of capacity constraint of arc (i, j, t, s)

Table 2 Indices, sets, parameters and variables in **Model 2**

Symbols	Definition
Indices	
v	Index of vehicle groups
q	Index of passenger groups
Parameters	
$d(v)$	Total number of available vehicles of vehicle group v
$g(q)$	Total number of trip requests of passenger group q
c_v^k	Cost of path k of vehicle group v
$\delta_q^{v,k}$	Incidence between passenger gourp q and path k of vehicle group $=1$, matched; otherwise, $=0$.
$\delta_{i,j,t,s}^{v,k}$	Incidence between arc (i, j, t, s) and path k of vehicle group $=1$, matched; otherwise, $=0$.
Variables	
y_v^k	Positive continuous variable, the number of vehicles belonging to group v choosing path k in the space-time-state network, finally simplified as y_k
$y_{j',j,t,t+1}$	Positive continuous variable, the outflow arc capacity on arc $(j', j, t, t+1)$
λ_q	Lagrangian multiplier of passenger group q's trip requests
$\mu_{i,j,t,s}$	Lagrangian multiplier of capacity constraints of arc (i, j, t, s)

3.1 Arc-based agent-based integer linear programming model (Model 1)

Based on the space-time-state network constructed in Section 2, our mathematical programming model is proposed to minimize the total vehicle travel cost and satisfy passengers' trip requests and road capacity constraints.

Objective function:

$$Min\ Z = \sum_a \sum_{(i,j,t,s,w,w')} (c_{i,j,t,s,w,w'}^a \times x_{i,j,t,s,w,w'}^a) \tag{1}$$

Subject to,

(i) **Vehicle supply**: Arc-based flow balance constraint for each vehicle

$$\sum_{i,t,w:(i,j,t,s,w,w')} x_{i,j,t,s,w,w'}^a - \sum_{i,t,w:(j,i,s,t,w',w)} x_{j,i,s,t,w',w}^a = \begin{cases} -1 & j = O(a), s = DT(a), w = [0,0,\dots,0] \\ 1 & j = D(a), s = T, w = [0,0,\dots,0] \\ 0 & otherwise \end{cases}, \forall a \tag{2}$$

(ii) **Travel demand**: Passenger p's pick-up request constraint

$$\sum_a \sum_{i,t,s:(i,j,t,s,w,w') \in A(p)} x_{i,j,t,s,w,w'}^a = 1, \forall p \tag{3}$$

(iii) **Infrastructure supply**: Tight road capacity constraint (endogenous congestion)

$$\sum_a \sum_w x_{i,j,t,s,w,w'}^a \le cap_{i,j,t,s}, \forall (i, j, t, s) \tag{4}$$

(iv) Binary definitional constraint

$$x_{i,j,t,s,w,w'}^a \in \{0,1\} \tag{5}$$

Constraint (2) ensures that each vehicle follows the flow balance. By constraint (3), each passenger will be picked up only once. For the problem with both pickup and drop-off requests, the state transit graph with cumulative passenger served status can guarantee that the passenger will be dropped off once he/she is picked up, so the drop-off constraint is always satisfied in our model. For the problem with pickup or drop-off requests only, we just need to ensure that the passenger pickup arc is only visited once by all vehicles. To capture the endogenous congestions, Constraint (4) makes the number of vehicles entering arc (i, j, t, s) without exceeding the arc capacity, which can be addressed as a point queue model where the vehicle has to choose the waiting arc if the capacity of the downstream link is not available at current time interval. The modeling on queue spillback will be discussed in Section 5. Variable $x^{a}_{i,j,t,s,w,w'}$ is a binary variable, which indicates whether or not vehicle a will visit arc (i,j,t,s,w,w'). This proposed model is an integer linear programming model, so it can be directly solved by standard solvers. However, for the large-scale network applications, we will apply different decomposition approaches to decompose the problem as a number of relatively easy sub-problems in next sections.

3.2 Path-based flow-based linear programming model (Model 2)

From the perspective of flow-based dynamic traffic assignment problems, all vehicles are assigned to the network based on each origin-destination (OD) pair as continuous flows rather than each individual vehicle, which could greatly reduce the number of variables to improve the computational efficiency. Similarly, if (i) vehicles and passengers can be grouped by its origin, destination and required service time period, and (ii) if all possible space-time-state path information with vehicle-to-passenger and vehicle-to-arc assignment can be enumerated in advance for our overall problem, the remaining is just to assign vehicles from each vehicle group to the network to satisfy the passenger group trip requests and not to violate the road capacity limitations. The linear programming model is listed as follows.

$$\min \sum_{(v,k)} (c_v^k \times y_v^k) \tag{6}$$

Subject to

(i) **Vehicle supply:** Path-based vehicle group flow balance constraint:

$$\sum_k y_v^k = d(v), \forall v \tag{7}$$

(ii) **Travel demand**: Pickup requests on passenger group q:

$$\sum_{(v,k)} (y_v^k \times \delta_{v,k}^q) = g(q), \forall q \tag{8}$$

(iii) **Infrastructure supply**: Road capacity constraints (endogenous congestion):

$$\sum_{(v,k)} (y_v^k \times \delta_{v,k}^{i,j,t,s}) \leq cap_{i,j,t,s}, \forall (i,j,t,s) \tag{9}$$

(iv) Positive continuous variable:

$$y_v^k \geq 0 \tag{10}$$

Since each STS path of each vehicle group is provided in advance, the cost of path k of vehicle group v for one specific OD pair is given. Constraints (7) ensures the total number of vehicles in each vehicle group v is assigned to the network, which is consistent with constraint (2) for all vehicles in agent-based model. Eq.(8) makes the total demand of passenger group q completely satisfied and also corresponds Eq. (3) of passenger trip requests. Road capacity is considered in constraint (9), similar to constraint (4). Finally, we can generate a linear programming model, which is possible to be solved for large-scale transportation networks. Usually, it is difficult to enumerate all possible space-time-state paths, so a decomposition-based solution is helpful to generate a number of available paths, which can be viewed as the column pool construction used in Section 4.2.

4. Problem decomposition approaches

4.1 Dantzig-Wolfe decomposition for arc-based agent-based formulation (Model 1)

In our proposed mathematical models in section 3.1, the flow balance constraint for each vehicle is a special block and can be solved independently. Therefore, Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960) is applied to solve our models and the flow balance constraints are used to develop the sub-problems. In addition, as mentioned by Larsson and Patriksson (1992), the generated paths (extreme points) from this decomposition approach are helpful for re-optimization if the demand or network has any updates in the future. Based on the point queue model to capture the road congestion, the primal model is decomposed as a master problem and different sub-problems for each vehicle as follows.

The master problem:

$$Min \sum_a \sum_{(i,j,t,s,w,w')} [c^a_{i,j,t,s,w,w'} \times \sum_k (\lambda^k_a \times x^k_a \times \delta^{a,k}_{i,j,t,s,w,w'})] \quad (11)$$

$$\sum_a \sum_{(i,j,t,s,w,w')\in A(p)} \sum_k (\lambda^k_a \times x^k_a \times \delta^{a,k}_{i,j,t,s,w,w'}) = 1, \forall\, p \quad (12)$$

$$\sum_a \sum_w \sum_k (\lambda^k_a \times x^k_a \times \beta^{a,k}_{i,j,t,s,w,w'}) \le cap_{i,j,t,s}, \forall\,(i,j,t,s) \quad (13)$$

$$\sum_k \lambda^a_k = 1, \forall a \quad (14)$$

$$\lambda^a_k = \{0,1\} \quad (15)$$

The sub-problem for each vehicle a:

$$Min\ Z' = \sum_{(i,j,t,s,w,w')} (c^a_{i,j,t,s,w,w'} \times x^a_{i,j,t,s,w,w'}) - \sum_p \sum_{(i,j,t,s,w,w')\in A(p)} (\pi_p \times x^a_{i,j,t,s,w,w'}) - \sum_{(i,j,t,s)} (\mu_{i,j,t,s} \times \sum_w x^a_{i,j,t,s,w,w'}) - \omega_a \quad (16)$$

Subject to vehicle flow balance constraint. π_p, $\mu_{i,j,t,s}$ and ω_a are the duals of side constraints (12), (13) and (14), respectively, and $x^a_{i,j,t,s,w,w'}$ is a binary variable.

In the master problem, x^k_a is always equal to 1 for each vehicle, so it can be removed, and λ^a_k determines whether vehicle a will select path k or not. The sub-problem generates paths for each vehicle at each iteration, so it is convenient to use a path-based formulation for our master problem.

Objective function (11) can be reformulated as

$$Min \sum_a \sum_k (c^k_a \times \lambda^k_a) \quad (17)$$

Passenger pickup constraint (12) can be updated as

$$\sum_a \sum_k (\lambda^k_a \times \delta^a_{k,p}) = 1, \forall\, p \quad (18)$$

Space-time arc capacity constraint (13) is renewed as

$$\sum_a \sum_k (\lambda^k_a \times \beta^k_{a,(i,j,t,s)}) \le cap_{i,j,t,s},\ \forall (i,j,t,s) \quad (19)$$

The algorithm procedure is shown in Fig. 4.

At step 1, find an initial feasible solution. Kalvelagen (2003) provided a mathematical model by adding virtual variables to find the feasible solution. Another way is to sequentially load each vehicle by using the solution from Lagrangian relaxation (Zhou et al., 2018). Specifically, once one vehicle finds its best route, the passengers served by this vehicle will be given a flag so that the following vehicles cannot visit those passengers anymore. Meanwhile, the space-time trajectory of that vehicle is recorded, so the capacity of visited space-time arcs will be updated by reducing 1. Once there is no available capacity on that arc, its arc travel time will be infinity. The pseudo-code is shown in Fig. 5.

At step 2, the restricted master problem is an integer linear programming model, which can be solved by standard solver or a hybrid of solvers and branch-and bound, and then the model provides the dual prices of side constraints for the sub-problems.

At step 3, the sub-problem is a time-dependent state-dependent shortest path problem and offers new extreme points (paths) for each vehicle for the master problem at step 2, if all reduced cost is not non-negative; otherwise, the optimal solution is found. Note that path cost of c^k_a, passenger-vehicle assignment incidence of $\delta^a_{k,p}$ and path-arc incidence of $\beta^a_{k,(i,j,t,s)}$ are obtained once a new path (path k) is generated for vehicle a.

A beam search algorithm is proposed as an approximate dynamic programming approach to dynamically construct the space-time-state network and find the optimal routing with least generalized route cost for each vehicle. As an improved version of the previous beam-search algorithm (Zhou et al., 2018), we add one more loop on each node so that more possible vehicle states will be considered during the beam search process. The key part is to update vehicles' states based on the state transition rule considering the sequence and time windows of passenger pickup and delivery and vehicles' carrying capacity. In addition, this algorithm is also applicable to solve the problems with pickup or drop-off only when the vehicle's state definition is correspondingly changed based on this new problem.

To improve the computation efficiency, we also consider a tree-based path representation embedded in the beam search algorithms, which can provide the one-to-all shortest path tree during the search process. Therefore, when one vehicle finds its three-dimensional shortest path, the shortest path tree from its origin vertex (with its depot node and departure time) to all vertex is also found. Then if another vehicle also departs from that origin vertex but to different destination depot, it doesn't need to run the beam-search algorithm again. Instead, we can directly find its shortest path in the one-to-all shortest path tree, which will reduce the total computation time.

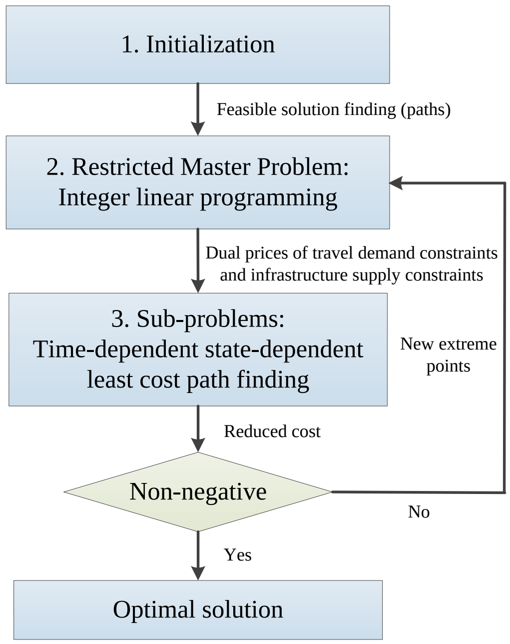


Fig. 4 The algorithm procedure of Dantzig-Wolfe decomposition

```
//initialization: passenger status, relation between vehicle and passenger
for passenger p=1 to total_number_of_passengers
    number_of_visits_of_passenger [p] = 0 // passenger p is not served
end// passenger
for vehicle a=1 to total_number_of_vehicles
     for passenger p=1 to total_number_of_passengers
          vehicle_passenger_visit_allowed_flag[a][p] = 1 // a is allowed to serve p
     end //passenger
end// vehicle
// sequentially loading each vehicle to find its own least cost path
for vehicle a=1 to total_number_of_vehicles
    run the beam search algorithm to find the best route for vehicle a and number_of_visits_of_passenger [p] = 1 if
    passenger p is served
    if (a < total_number_of_vehicles)
        for p = 1 to total_number_of_passengers
             if (number_of_visits_of_passenger [p] = 1)
                // the following vehicles cannot visit passenger p
                vehicle_passenger_visit_allowed_flag[a + 1][p] = 0
             end
         end // passenger
     end
     // update current available road capacity after loading vehicle a to obtain the visited link and time sequence
     for link_no = 1 to total_number_of_visited_links_of_vehicle a
          visit_time = visit_time_sequence[link_no] of vehicle a
          link_capacity[link_no][visit_time]= max (0, link_capacity[link_no][visit_time]- 1)
          if (link_capacity[link_no][visit_time]=0) // no available capacity
             link_time_dependent_travel_time[link_no][ visit_time] = infinity // arc cost is infinity
```

```
            end
        end// link
    end // vehicle
```

Fig. 5 Pseudo-code of proposed feasible solution finding algorithm

4.2 Column-pool-based approximation for STS path-based flow-based formulation (Model 2)

Each vehicle group has a number of vehicles with the same origin and destination depots and departure time, and each passenger group has a number of passengers with same trip requests. As stated in section 3.2, it is possible to generate the connection of vehicle-to-passenger and vehicle-to-arc relationship in advance and then assign vehicles as continuous flows to satisfy the requirements of passengers and road capacity, from the perspective of dynamic traffic assignment.

As an approximation, we decompose this complex primal problem as two stages as shown in Fig. 6. Stage A is to build a column pool by generating a number of columns (space-time-state paths) for each vehicle group by solving the arc-based agent-based formulation using ADMM. The other advantage of column pool generation is for re-optimization in case the demand, vehicle or network has any changes in the future, so we can use those available columns as a starting point instead of performing the optimization model from the beginning. Stage B is to solve the STS path-based flow-based linear programming model by ADMM to assign vehicles from different vehicle groups to serve passengers from different passenger groups while satisfying the time-dependent road capacity constraints.

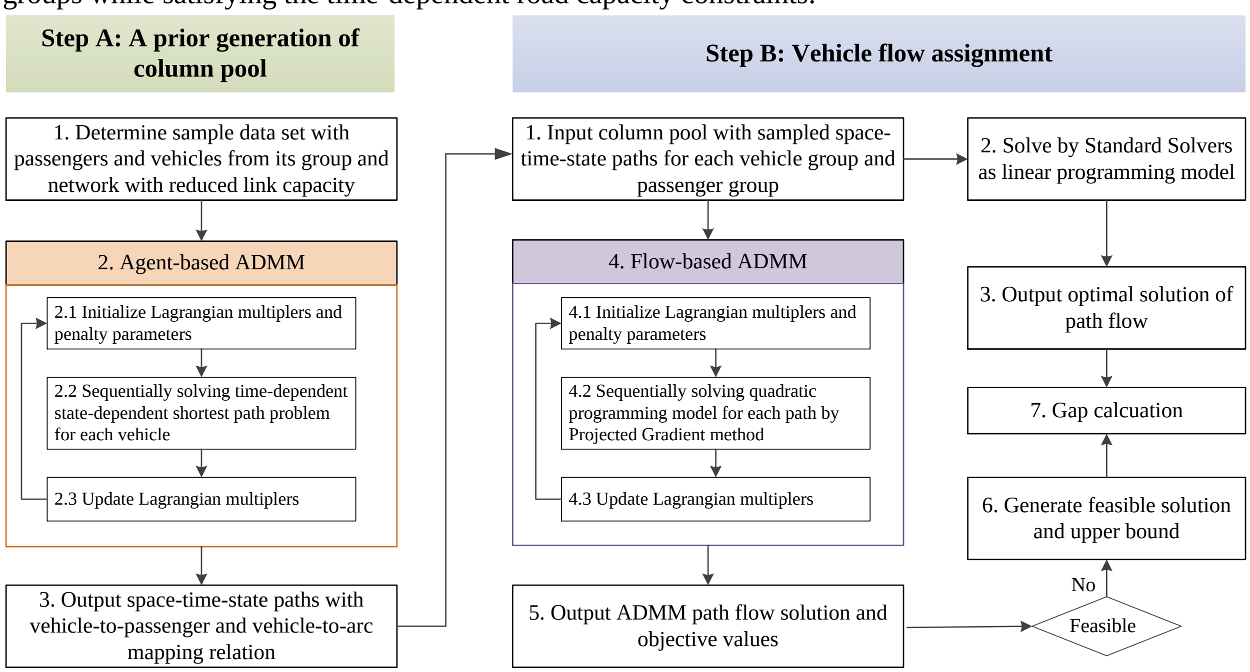


Fig. 6 The flow chart of column-pool-based approximation

4.2.1 A priori generation of column pool based on Model 1

Since it needs to obtain the possible relation between different vehicle groups, passenger groups and space-time arcs, we select a number of passengers and vehicles from their groups as a sample set. Also, the arc capacity is reduced correspondingly to produce the possible congestions. If the generated column pool is too small, it is possible to have an infeasible solution or a sub-optimal solution. Therefore, the key is to find the balance among column number, computation time, and solution quality. How to determine this sample set and how to dynamically manage the column pool (Barnhart et al., 1998) will be our future research.

Then an arc-based agent-based model in section 3.1 is used to generate the column (space-time-state path) for each vehicle. ADMM is selected to decompose this problem as a number of sequential time-dependent state-dependent shortest path problem without using any standard solvers. In addition, it is also possible to try other heuristic methods to generate different columns, such as, changing the benefits of serving different passengers, local search algorithms, and so on.

The objective function of ADMM:

$$Min\ Z = L(\boldsymbol{x^a}, \boldsymbol{\pi_p}, \boldsymbol{\pi_{(i,j,t,s)}}) = \sum_a \sum_{(i,j,t,s,w,w')}(c_{i,j,t,s,w,w'} \times x^a_{i,j,t,s,w,w'}) + \sum_p [\pi_p \times (\sum_a \sum_{(i,j,t,s,w,w') \in A(p)} (x^a_{i,j,t,s,w,w'} \times \delta^a_{i,j,t,s}) - 1)] + \frac{\rho_1}{2} \sum_p \left[\sum_a \sum_{(i,j,t,s,w,w') \in A(p)} (x^a_{i,j,t,s,w,w'} \times \delta^a_{i,j,t,s}) - 1\right]^2 + \sum_{(i,j,t,s)} [\pi_{(i,j,t,s)} \times (\sum_a \sum_w x^a_{i,j,t,s,w,w'} - cap_{i,j,t,s})] + \frac{\rho_2}{2} \sum_{(i,j,t,s)} \left[\sum_a \sum_w x^a_{i,j,t,s,w,w'} - cap_{i,j,t,s}\right]^2 \quad (20)$$

Subject to the flow-balance constraint for each vehicle at constraint (2). $x^2 = x$ if $x = \{0,1\}$, so the quadratic terms in objective function can be converted to be linear with binary variables, when the problem is solved for each vehicle sequentially based on the standard procedure of ADMM (Boyd, 2011). For the illustration purpose, we assume there is a model with objective function $(x_1 + x_2 - a)^2$and two binary variables x_1 and x_2. When solving x_1, we can have $(x_1 + x_2 - a)^2 = [x_1 + (x_2 - a)]^2 = x_1^2 + 2x_1(x_2 - a) + (x_2 - a)^2 = (2x_2 - 2a + 1)x_1 + (x_2 - a)^2$ where x_2 is fixed and a is a parameter.

The penalty parameters of $\rho_1$and ρ_2 for passenger service constraint and arc capacity constraints are given in this paper, but they can also be updated based on some rules used in previous augmented Lagrangian relaxation models. The iterative scheme of ADMM is shown in Fig.7.

```
// initialization
Set up initial values for all Lagrangian multipliers and penalty parameters
for n = 1 to n_max // total number of iterations
    for a = 1 to a_max //total number of vehicles
        Find the time-dependent state-dependent shortest path for vehicle a with the fixed solution of other vehicles
        Update the network-arc costs based on the new solution of vehicle a for vehicle a + 1
    end // vehicle
    Update Lagrangian multipliers of passenger pickup constraints and arc capacity constraints
end // iterations
```

Fig. 7 The iterative scheme of ADMM

At iteration $n+1$ of ADMM:

$$x_a^{n+1} = \arg\min\{L(x_1^{n+1}, x_2^{n+1}, \dots, x_a, x_{a+1}^n, \dots, x_{a_{\max}}^n, \boldsymbol{\pi_p^n}, \boldsymbol{\pi_{i,j,t,s}^n})\} \quad (21)$$

$$\pi_p^{n+1} = \pi_p^n - \rho_1[\sum_a \sum_{(i,j,t,s,w,w') \in A(p)} (x^{a,n+1}_{i,j,t,s,w,w'} \times \delta^a_{i,j,t,s}) - 1] \quad (22)$$

$$\pi^{n+1}_{i,j,t,s} = \max\{0, \pi^n_{i,j,t,s} - \rho_2[\sum_a \sum_w x^{a,n+1}_{i,j,t,s,w,w'} - cap_{i,j,t,s}] \quad (23)$$

The subproblem for each vehicle is a time-dependent state-dependent shortest path problem due to the linear relation in the objective function. Once one vehicle finds its best solution, the network arc cost will be updated for the next vehicle's subproblem solving. All Lagrangian multipliers are updated at the end of each iteration.

4.2.2 Dynamic vehicle flow assignment based on Model 2

Once the columns are generated for each vehicle group, the remaining is to assign vehicles to satisfy passengers' trip requests and network capacities. Assume that the total number of vehicles from each vehicle group is also unknown, then we can apply ADMM to convert the flow-based linear programming model as a quadratic programming model as follows.

Objective function:

$$\min \sum_k (c^k \times y^k) + \sum_q (\lambda_q \times [(\sum_k (y^k \times \delta_q^k) - g(q)]) + \frac{\rho_1}{2} \sum_q \left(\left(\sum_k (y^k \times \delta_q^k) - g(q)\right)^2 + \right.$$

$$\sum_{i,j,t,s} (\mu_{i,j,t,s} \times [\sum_k (y^k \times \delta_{i,j,t,s}^k) - cap_{i,j,t,s}]) + \frac{\rho_2}{2} \sum_{i,j,t,s} \left(\sum_k (y^k \times \delta_{i,j,t,s}^k) - cap_{i,j,t,s} \right)^2 \quad (24)$$

where y^k is the path flow of path k, and ρ_1, ρ_2 are parameters. For simplicity, y_v^k for each OD pair (o, d, τ) is replaced by y^k by resorting all path numbers.

Its Hessian Matrix can be derived as,

$$H = \begin{vmatrix} \sigma_1 \sum_p \delta_p^1 + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^1 & \sigma_1 \sum_p \delta_p^1 \delta_p^2 + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^1 \delta_{i,j,t,s}^2 & \cdots & \sigma_1 \sum_p \delta_p^1 \delta_p^{k_{max}} + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^1 \delta_{i,j,t,s}^{k_{max}} \\ \sigma_1 \sum_p \delta_p^1 \delta_p^2 + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^1 \delta_{i,j,t,s}^2 & \sigma_1 \sum_p \delta_p^2 + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^2 & \cdots & \sigma_1 \sum_p \delta_p^2 \delta_p^{k_{max}} + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^2 \delta_{i,j,t,s}^{k_{max}} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_1 \sum_p \delta_p^1 \delta_p^{k_{max}} + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^1 \delta_{i,j,t,s}^{k_{max}} & \sigma_1 \sum_p \delta_p^2 \delta_p^{k_{max}} + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^2 \delta_{i,j,t,s}^{k_{max}} & \cdots & \sigma_1 \sum_p \delta_p^{k_{max}} + \sigma_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^{k_{max}} \end{vmatrix}$$

Since it is difficult to directly obtain its inverse matrix H^-, especially in large-scale networks, we apply ADMM to decompose the primal problem to sequentially solve the subproblem for each column as

$$y_k^{n+1} = \arg\min\{L(y_1^{n+1}, y_2^{n+1}, \ldots, y_k, y_{k+1}^n, \ldots, y_{k_{\max}}^n, \boldsymbol{\lambda_q^n}, \boldsymbol{\mu_{i,j,t,s}^n})\}$$

The subproblem for y^k is a quadratic programming model which could be solved by projected gradient method (Rosen, 1960) as follows:

$$y_k^{n+1} = \max\{0, y_k^n - \frac{1}{s} \times L(y_k^n)' \quad (25)$$

where $L(y_k^n)' = c^k + \sum_q \lambda_q \times \delta_q^k + \rho_1 \left(\sum_q \delta_q^k \left(\left(\sum_k (y_k^n \times \delta_q^k) - g(q) \right) \right) + \sum_{i,j,t,s} \mu_{i,j,t,s} \times \delta_{i,j,t,s}^k + \right.$ $\rho_2 (\sum_{i,j,t,s} \delta_{i,j,t,s}^k (\sum_k (y_k^n \times \delta_{i,j,t,s}^k) - cap_{i,j,t,s}))$, and $s = \frac{\partial^2 L(x)}{\partial x^2} = \rho_1 \sum_q \delta_q^k + \rho_2 \sum_{i,j,t,s} \delta_{i,j,t,s}^k$. In addition, projected gradient method also has been used in solving the path-based nonlinear programming models in equilibrium traffic assignment (Larsson and Patriksson, 1992; Jayakrishnan et al, 1994; Florian et al., 2009), and it is more efficient, compared with arc-based nonlinear programming models, but needs more memory use.

At each iteration of ADMM, the Lagrangian multipliers are updated as follows,

Passenger group trip requests: $\lambda_q^{n+1} = \lambda_q^n + \rho_1 ((\sum_k (y_k^n \times \delta_q^k) - g(q))$

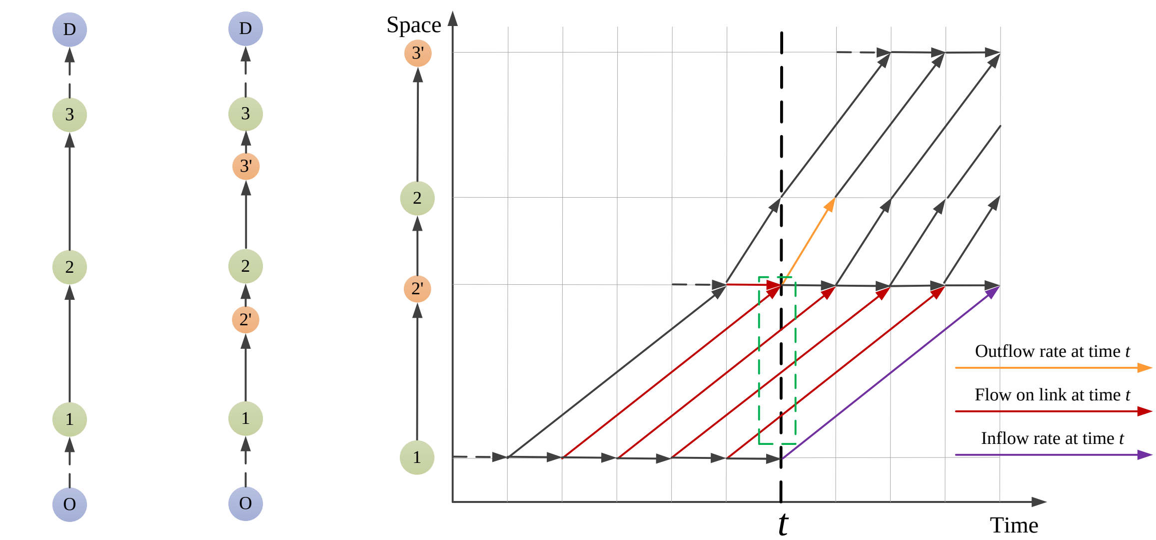
Arc capacity constraints: $\mu_{i,j,t,s}^{n+1} = \max\{0, \mu_{i,j,t,s}^n + \rho_2 (\sum_k (y_k^n \times \delta_{i,j,t,s}^k) - cap_{i,j,t,s})\}$

As a discussion, it is possible to assign different vehicles within different blocks, and each block can be sequentially solved in ADMM and vehicles within a same block can find the best solution with parallel computing techniques.

We need to note that the solution from ADMM cannot always guarantee its feasibility. In order to find a feasible solution and upper bound value, we can sequentially load each column flow from ADMM. Path flows that exceed the required passenger trip requests or arc capacity will be removed during the sequential loading process. Finally, if some passenger requests cannot be satisfied, virtual vehicles will be used to find a feasible solution as the upper bound.

5. Discussions on modeling queue spillback

In order to capture the queue spillback, we proposed a spatial queue model by improving the approach proposed by Drissi-Kaïtouni and Hameda-Benchekroun (1992) where the link storage capacity with jam density and backward wave speed are not considered in space-time networks. Take the simple network in Fig.8 (a) as an example. A virtual node as the waiting node is added for each link in the modified network in Fig.8 (b). The travel time of link $(2', 2)$ is assumed to be 1 time unit and its length is a small value as an approximation, so this link is used for discharging flows, and its capacity as the outflow capacity of link (1,2) as a variable will be determined by its downstream links. The inflow capacity of link (1,2) is the capacity of link (1,2′), equal to $Cap_{1,2}$. The link storage capacity of link (1,2) is $Len_{1,2} \times n_{1,2} \times Jam_{1,2}$ and will be represented on link (1,2′). The corresponding space-time network is constructed in Fig.8 (c). Specifically, at time t on link (1,2), (i) the inflow capacity constraint is $x_{1,2',t-FFTT_{1,2'},t} \leq Cap_{1,2',t-FFTT_{1,2},t}$ for arc $(1,2',t-FFTT_{1,2'},t)$ shown in purple; (ii) the outflow capacity constraint is $x_{2',2,t,t+1} \leq Cap_{2',2,t,t+1}$ for arc $(2',2,t,t+1)$ shown in orange; (iii) the link storage capacity constraint is $CA_{1,2,t} - CD_{1,2,t} = x_{2',2',t-1,t} + \sum_{s=t-FFTT_{1,2'}+1}^{t-1} x_{1,2',s,t} \leq Len_{1,2'} \times n_{1,2'} \times KJam_{1,2'}$. $CA_{1,2,t}$ and $CD_{1,2,t}$ are the cumulative arrival count and the cumulative departure count of link (1,2) at time t. The link outflow capacity is calculated and given by the capacities of its downstream links.



(a) Physical network (b) Modified physical network (c) Space-time network for modeling the queue on link (1, 2)

Fig. 8 Illustration for the spatial queue model in a space-time network

Focusing on the vehicle routing in our proposed space-time-state networks, the spatial queue model can be formulated as follows where $x_{i,j,t,s,w,w'} = \sum_a x^a_{i,j,t,s,w,w'}$ in Model 1,

Inflow arc capacity constraint:

$$\sum_w x_{i,j',t-FFTT_{i,j}+1,t,w,w'} \leq Cap_{i,j',t-FFTT_{i,j}+1,t}, \forall (i,j') \in L_{inflow}, \forall t \quad (26)$$

Outflow arc capacity constraint:

$$\sum_w x_{j',j,t,t+1,w,w'} \leq y_{j',j,t,t+1}, \forall (j',j) \in L_{outflow}, \forall t \quad (27)$$

Outflow arc capacity balance constraint at points without merger and diverge:

$$y_{j',j,t,t+1} \leq Cap_{j,i,t+1,s} \quad (28)$$

Outflow arc capacity balance constraint at merger points:

$$\sum_{(j',t)} y_{j',j,t,t+1} \leq Cap_{j,i,t+1,s}, \forall (j,t+1) \in A_m \quad (29)$$

Outflow arc capacity balance constraint at diverge points:

$$y_{j',j,t,t+1} \leq \sum_{(i,s)} Cap_{j,i,t+1,s} \; \forall (j,t+1) \in A_d \quad (30)$$

Link storage capacity constraint:

$$\sum_w x_{j',j',t-1,t,w,w'} + \sum_w \sum_{s=t-FFTT_{i,j'}}^{t-1} x_{i,j',s,t,w,w'} \leq Len_{i,j'} \times n_{i,j'} \times KJam_{i,j'}, \forall (i,j') \in L_{inflow}, \forall t (31)$$

Furthermore, to consider the backward wave speed under congested conditions, Newell's simplified kinematic wave model (Newell, 1993) considers the link storage capacity by $CA_{(i,j,t)} - CD_{i,j,t-BWTT(i,j)} \leq Len_{i,j} \times n_{i,j} \times Jam_{i,j}$. Similar to the derivation of the spatial queue model above, Newell's simplified kinematic wave model can have the following constrain for link storage capacity.

$$\sum_w \sum_{s=t-BWTT(i,j')}^{t} x_{j',j',s-1,s,w,w'} + \sum_w \sum_{s=t-FFTT_{i,j'}-BWTT_{i,j'}}^{t-1} x_{i,j',s,s+FFTT_{i,j'},w,w'} \le Len_{i,j'} \times n_{i,j'} \times KJam_{i,j'}, \forall (i,j') \in L_{inflow}, \forall t \quad (32)$$

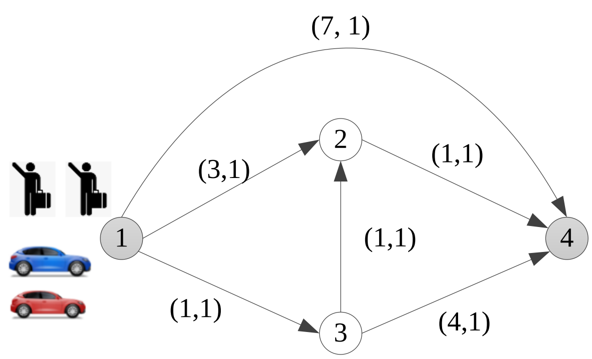
As a note, from the perspective of car-following models, the backward wave speed depends on drivers' average reaction time and minimal following safety distance, and Newell's simplified microscopic car-following model (Newell, 2002) is consistent with his macroscopic kinematic wave model. Wei et al. (2017) proposed a binary integer programming model to optimally control vehicle trajectory based on Newell's simplified car-following model in time-extended space-time networks, which can be incorporated in our modeling framework in state-space-time networks but will cause a huge number of variables and constraints.

6. Numerical examples

Section 6.1 shows how vehicles with passengers switch routes to reach the optimal solution in a capacitated network iteration by iteration using Dantzig-Wolfe decomposition. Section 6.2 focuses on both pickup and drop-off requests in the Sioux-Fall network. The restricted master problem is solved by CPLEX in GAMS, and the sub-problems are solved by a beam search algorithm. In section 6.3, we consider the pickup only in the Chicago sketch network with a large number of autonomous vehicles and passengers belonging to different groups. A column-pool-based approximation approach is used to solve this problem. All corresponding source codes are shared online at https://github.com/TonyLiu2015/AVRLite.

6.1 A simple case in a capacitated network

Fig. 9 shows a simple capacitated network with 4 nodes, 6 links and 4 possible paths. Assume that there are 2 vehicles and each vehicle picks up one passengers from origin node 1 to destination node 4. Our goal is to minimize the total vehicle travel cost by Dantzig-Wolfe decomposition approach.



Path ID	Node Sequence	Path Cost	Path Trajectory
Path 1	1→2→4	4	
Path 2	1→3→4	5	
Path 3	1→4	7	
Path 4	1→3→2→4	3	

(link cost, link capacity)

Fig. 9 A simple capacitated network

Table 3 lists the details of each iteration where λ_k is the weight of selecting path k of from node 1 to ndoe 4, $\mu_{i,j}$ is the dual price of tight capacity constraint of link (i,j), and π is the dual price of path weight constraint. Iterations 1 and 2 are used to generate a feasible solution by adopting the approach (Kalvelagen, 2003) in Dantzig-Wolfe decomposition. After 4 iterations, one vehicle chooses path 1and the other selects route 2. The reduced cost is $\sum_k(\lambda_k \times c_k \times 2) - \sum_{(i,j)\in L} \mu_{i,j} \times f_{i,j} - \pi_w$ where c_k is the cost of path k and $f_{i,j}$ is the flow on link (i,j). Finally, the reduced cost is $4 + 5 - (-2) - (-2) - (-1) - 14 = 0$ and reach the optimal solution.

Table 3 The process of vehicle routing with endogenous congestions in Dantzig-Wolfe decomposition

Iteration NO.	Decomposed problem	Solution of different subproblems
Iteration 1	Subproblem	New column: **path 4**
	Restricted master problem	$\lambda_4 = 1$, $\mu_{1,3} = -1$, $\mu_{1,2} = 0$, $\mu_{3,2} = 0$, $\mu_{2,4} = 0$, $\mu_{3,4} = 0$, $\mu_{1,4} = 0$, $\pi = 2$

	Subproblem	New column: **path 1**
Iteration 2	Restricted master problem	$\lambda_1 = 1$, $\mu_{2,4} = -1$, $\mu_{1,2} = 0$, $\mu_{3,2} = 0$, $\mu_{1,3} = 0$, $\mu_{3,4} = 0$, $\mu_{1,4} = 0$, $\pi = 2$
	Subproblem	New column: **path 3**
Iteration 3	Restricted master problem	$\lambda_4 = 0.5$, $\lambda_3 = 0.5$, $\mu_{1,3} = -1$, $\mu_{2,4} = -3$, $\mu_{1,2} = 0$, $\mu_{3,2} = 0$, $\mu_{3,4} = 0$, $\mu_{1,4} = 0$, $\pi = 14$
	Subproblem	New column: **path 2**
Iteration 4	Restricted master problem	$\lambda_1 = 0.5$, $\lambda_2 = 0.5$, $\mu_{1,3} = -2$, $\mu_{2,4} = -2$, $\mu_{1,2} = -1$, $\mu_{3,2} = 0$, $\mu_{3,4} = 0$, $\mu_{1,4} = 0$, $\pi = 14$

6.2 Trips with pickup and drop-off requests in the Sioux-Fall test network

As shown in Fig. 10, the Sioux-Fall network has 24 nodes, 84 links with hourly capacity, and 5 vehicle depots. We assume that there are 30 trip requests with specific pickup and drop-off location and time windows, which are not listed due the space limit of this paper. In addition, 30 candidate vehicles departs from different origin depots at different departure time to its corresponding destination depots to serve those trip requests. The optimization time horizon is 110 time units to cover those time windows and possible trip time. The generalized benefit of serving one trip request is -20 time units, and the waiting generalized cost of vehicles is the half of its waiting time.

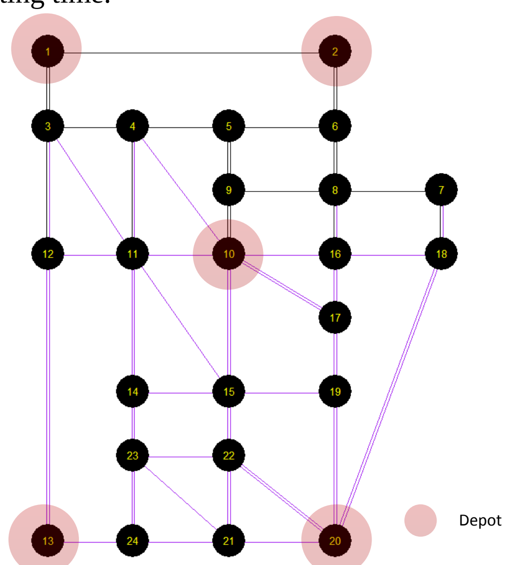


Fig. 10 Sioux-Fall network with five vehicle depots

At the beginning, we assume that the carrying capacity of all vehicles is just 1.Based on the algorithm for the initial feasible solution generation in section 4.1, 30 vehicles can serve 30 passengers in the physical network. Further, Dantzig-Wolfe decomposition is used to improve the initial solution. After 4 iterations, the final result is that only 27 vehicles are required to satisfy those requests. In addition, when the vehicle carrying capacity is increased to 2, the new solution just require 25 vehicles. The specific comparison is listed in Table 4.

Table 4 Solution comparison under different settings

Veh_No	Passenger_No			Veh_No	Passenger_No		
	Ini_solu	veh_cap=1	veh_cap=2		Ini_solu	veh_cap=1	veh_cap=2
1	[15]	[15][25]	[15][25]	16	[10]	[10]	[10]
2	[8]	[8]	[8]	17	[3]	[3]	[3]

3	[1]	[1]	[1]	18	[12]	[12]	[12]
4	[7]	[7]	[7][9]	19	[27]	[27]	[27]
5	[9]	[9]		20	[30]	[30]	
6	[11]	[11]		21	[23]	[23]	[23]
7	[29]			22	[25]	[20][21]	[11][20]
8	[28]	[28]	[28]	23	[22]	[22]	[22]
9	[17]	[17][29]	[17][29][30]	24	[19]	[19]	[19]
10	[21]		[21]	25	[4]	[4]	[4]
11	[20]			26	[5]	[5]	[5]
12	[26]	[26]	[26]	27	[24]	[24]	[24]
13	[16]	[16]	[16]	28	[14]	[14]	[14]
14	[18]	[18]	[18]	29	[13]	[13]	[13]
15	[2]	[2]	[2]	30	[6]	[6]	[6]

Focusing on vehicle 9, in the initial solution, it picks up and then drops off passenger 17, but in the improved solution by Dantzig-Wolfe decomposition it serves passenger 17 and then continues to serve passenger 29. In addition, when the vehicle carrying capacity is 2, vehicle 9 serves passenger 17, and then picks up passenger 30 and goes to picks up passenger 29, and finally drops off passenger 29 first and then drops off passenger 30. The required number of vehicles and total vehicle travel cost are shown in Fig. 11.

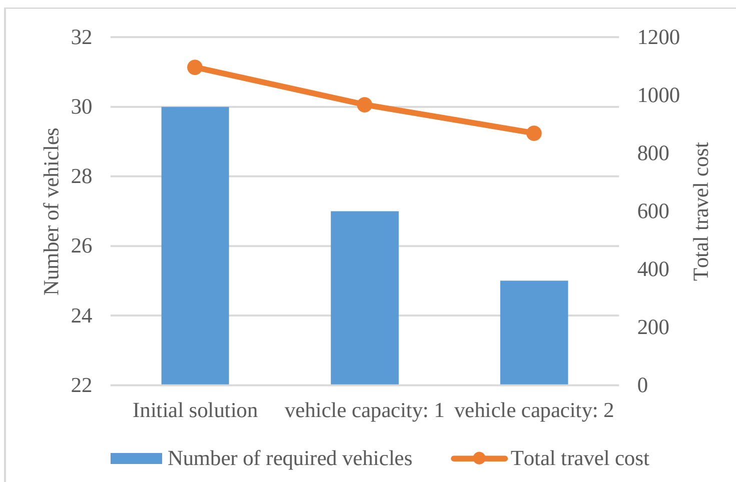


Fig. 11 Comparison on vehicle use and total travel cost under different conditions

6.3 Trip with pickup only request in the Chicago sketch test network

Step A: A priori generation of column pool in Model 1

The Chicago sketch network has 1320 nodes and 5431 links in Fig. 12. We assume that all passengers have the pickup only trip requests as the first mile problem, which indicates that each passenger group with a number of passengers will have a same destination with a vehicle group. We treat them as one pair of vehicle group and passenger group. To generate the column pool, two scenarios are designed:

Scenario 1: 10 pairs of vehicle groups and passenger groups. In each pair, as a sample set, we assume that (i) 243 vehicles departs from different origins to one destination with different working time windows and (ii) 387 passengers submit trip requests with different pickup locations and time windows. The time horizon is 60 min (time unit). Since this is a simple set, the space-time arc capacity in each minute is assumed to be 5 vehicles. Vehicle carrying capacity is given as 1, so each vehicle aims to pick up one passenger from the origin depot to their same destination. It has 2430 binary variables and 332,160 constraints.

Scenario 2: 20 pair of vehicle groups and passengers groups. For each pair, similar to scenario 1, we also assume the same number of vehicles and passenger trip request but with different vehicle inputs (origin, destination, working time windows) and passenger inputs (pickup locations and time windows). It has 4860 binary variables and 338,460 constraints

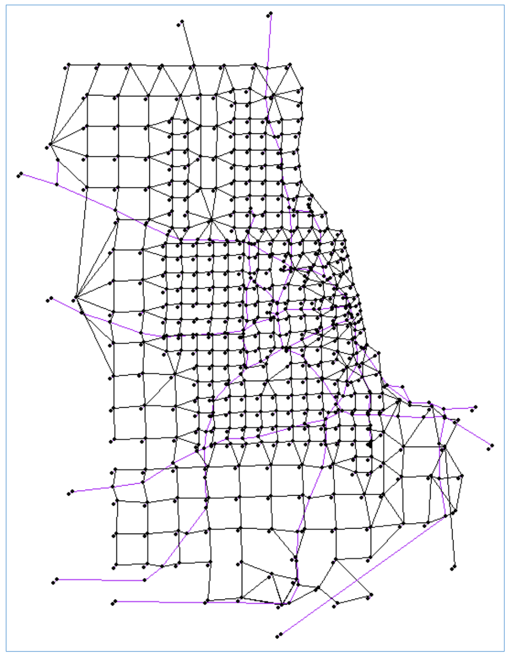


Fig. 12 Chicago sketch network for passenger pickup only

Then the agent-based ADMM in section 4.2.1 is used to find the vehicle-to-passenger and vehicle-to-arc assignment. Since the input data is randomly generated, some passengers may not be served and some vehicles may not serve any passengers. The final results are that (i) in scenario 1, 1789 vehicles find their paths/columns to serve 1084 passengers, and 23357 space-time arcs are generated based on vehicles' paths, and (ii) in scenario 2, 3686 vehicles find their paths/columns to serve 2226 passengers, and 36454 space-time arcs are generated based on vehicles' paths. The computation times for scenarios 1 and 2 are about 70 seconds and 140 seconds, respectively, at the laptop with 2.80GHz.

Step B: customized algorithm for flow-based ADMM

Note that each passenger has a specific pickup location, time window and destination, and vehicle can only pick up passengers within a same pair of groups. Therefore, we can build a column pool where each path of vehicles represent one column and each passenger represents one passenger group from Model 1. The question becomes how many vehicles from different vehicle groups are required to satisfy those trip request from different passenger groups under tight road capacity constraints. Based on the last two scenarios, we design two experiments:

Experiment 1: there are 1084 passenger groups and each passenger group has 4 passenger trip requests. **Experiment 2**: there are 2226 passenger groups and each passenger group has 2 passenger trip requests. In this physical network, we assume that all space-time arc capacity in each minute is 35 vehicles, equal to 2100 vehicles per hour. To solve this problem, we try three approaches: (i) flow-based ADMM, (ii) upper bound generation by sequentially loading the column flow solution from ADMM, (iii) optimal solution from standard solver CPLEX in GAMS.

In **experiment 1**: three cases with different penalty parameters of $\rho_1$for passenger trip constraints and ρ_2 for arc capacity constraints in ADMM are performed. Case 1: $\rho_1 = 3$ and $\rho_2 = 1$; Case 2: $\rho_1 = 3$ and $\rho_2 = 3$; Case 3: $\rho_1 = 3$ and $\rho_2 = 5$. The results from ADMM by running 250 iterations and the optimal solution from CPLEX are shown in Fig. 13. The ADMM can converge to different objective values in three cases with different penalty parameters. Since capacity values and the number of capacity constraints are much higher than that of passenger trip requests, it is better to assign a smaller penalty value for ρ_2 in the objective function of ADMM.

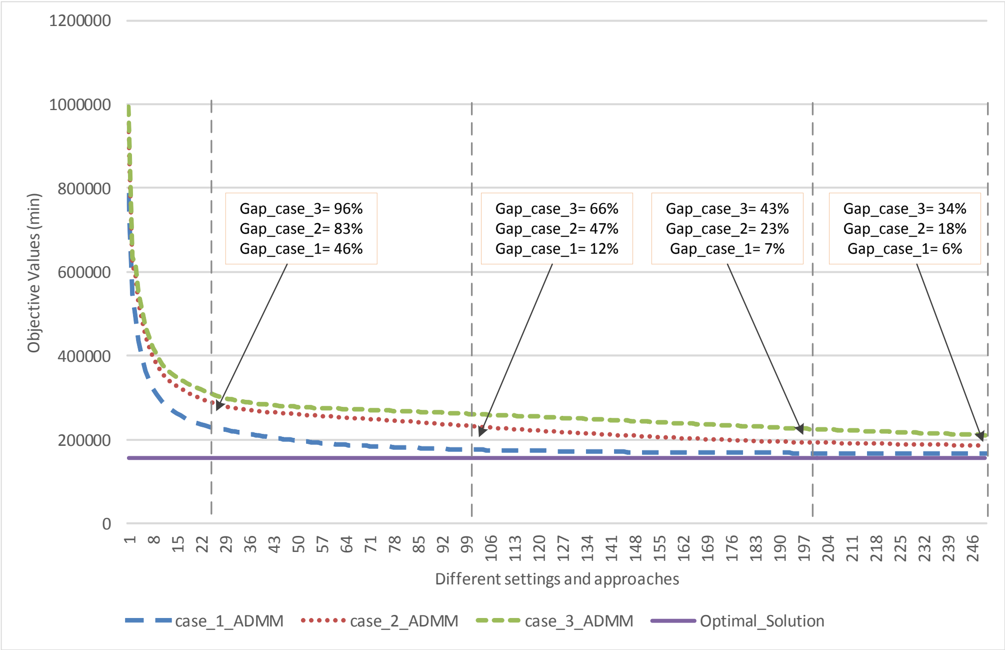


Fig. 13 Solution of each iteration of ADMM in three cases and CPLEX in experiment 1

Then the upper bound generation algorithm is also implemented to find a feasible solution based on the results of ADMM. Fig. 14 shows the objective values of upper bound in three cases and the optimal solution. The Gap values of three cases compared with the optimal solution are 4.3%, 3.4% and 3.1%, respectively. In addition, the computation time is 700 CPU seconds for running 250 iterations in ADMM. It is observed that the three cases can finally reach good solutions with very small gap values. Further, the ADMM result of case 3 has the biggest gap value, but its upper bound solution can still have a small gap value. The possible reason is that the total path flow is a variable, so the upper bound generation can reduce the total path flow from ADMM to have a better feasible solution to satisfy those passenger trip demands and not violate the arc capacity constraints.

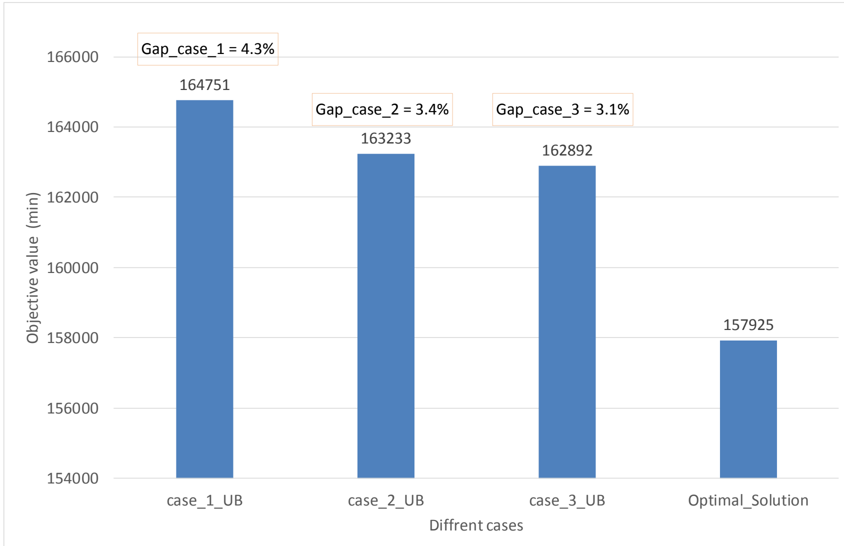
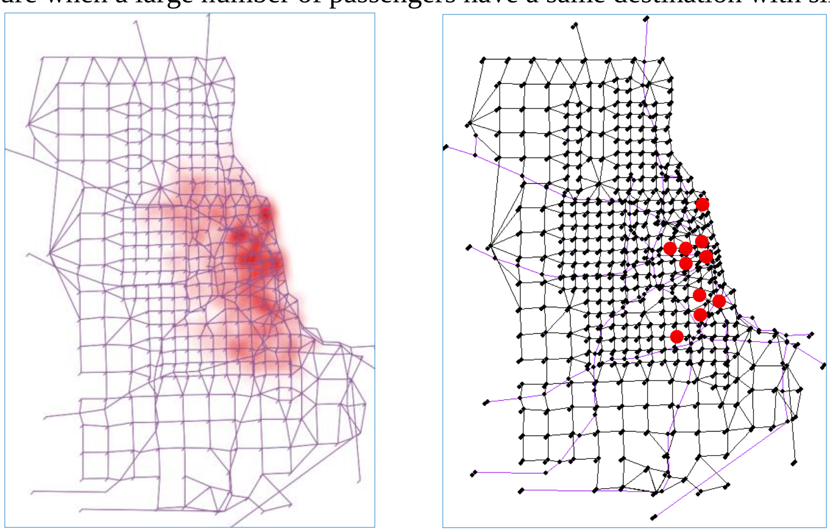


Fig. 14 Upper bound in three cases and the optimal value in CPLEX in Experiment 1

From the upper bound solution, 4737 space-time waiting arcs $(i, i, t, t+1)$ at 405 nodes have assigned vehicle flows, which indicates that the waiting happens at those nodes. By calculating the total waiting flow at those congested nodes during 60 mins, its heat map and the top10 of the most congested nodes are shown in Fig. 15(a) and (b), respectively. It can be observed that the destination areas of different passengers with pickup request only become congested, so it also raises one question about how to design the drop-off location in the future when a large number of passengers have a same destination with similar arrival time.



(a) The heat map on waiting flows in experiment 1 (b) Top 10 of congested nodes in experiment 1

Fig. 15 Visualization of congested nodes in experiment 1

Then the same procedure is also applied to **experiment 2**. The solution process and comparison among different cases are shown in Fig. 16 and Fig. 17, respectively. The Gap values of three cases compared with the optimal solution are 3.9%, 2.9% and 2.5%, respectively. It needs 2735 CPU seconds to finish 250 iterations in ADMM. Although CPLEX solver can provide an optimal solution with a short time in those tests, our approaches implemented by C++ is probably more flexible to manage the large-scale networks with more than 10 million columns in the future, especially when being design with parallel computing. From the upper bound solution, 7173 space-time waiting arcs $(i, i, t, t+1)$ at 448 nodes have assigned vehicle flows. Its heat map and the top10 of the most congested nodes are shown in Fig. 18(a) and (b), respectively.

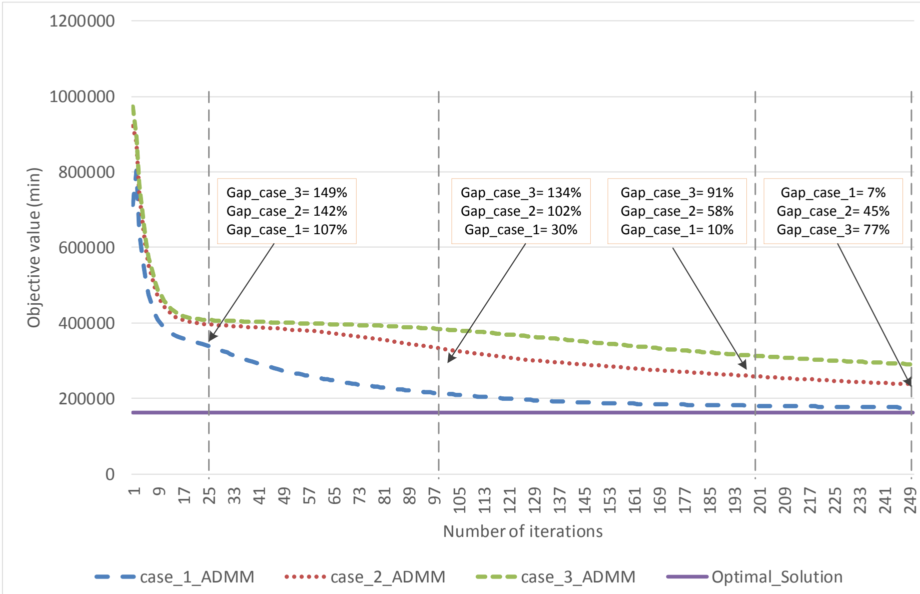


Fig. 16 Solution of each iteration of ADMM in three cases and CPLEX in experiment 2

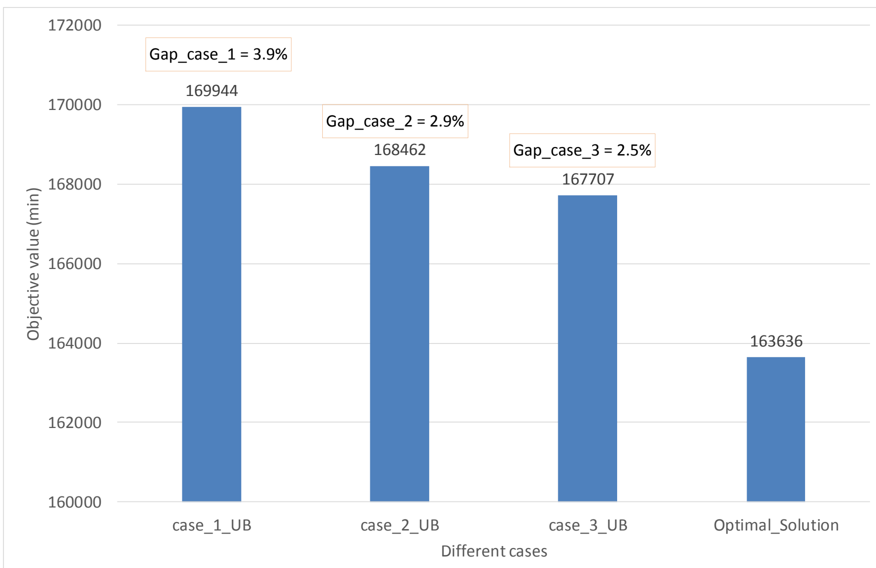
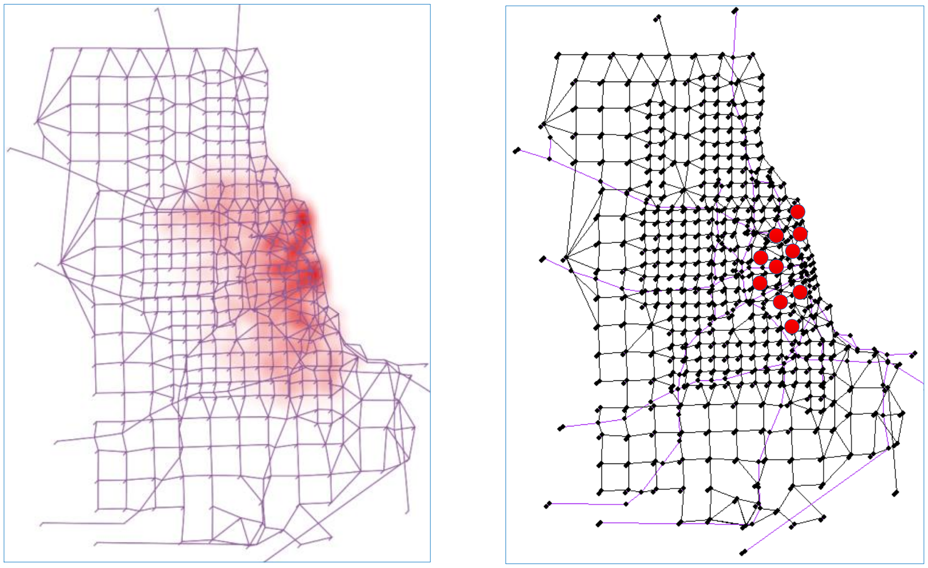


Fig. 17 Upper bound in three cases and the optimal value in CPLEX in Experiment 2



(a) The heat map on waiting flows in experiment 2 (b) Top 10 of congested nodes in experiment 2

Fig. 18 Visualization of congested nodes in two experiments

7. Future research

In the future research, we will improve the beam search algorithm (approximate dynamic programming) to address larger cases for finding the shortest path for pickup and drop-off requests. In addition, it is better to apply branch and bound to solve the relaxed problem in the master problem to obtain the dual prices in the Dantzig-Wolfe decomposition algorithm. Also, (i) how to determine the sample size and what algorithms can be used to generate the column pool and (ii) how to dynamically manage column pool are also important for the future application in large-scale networks. Since our proposed models are close to dynamic system optimal traffic assignment models, it is also possible to calculate the path marginal cost to solve our model under the space-time-state framework.

In a boarder sense, we will consider different kinds of trip requests and travel behaviors in a unified model to capture more complicated traffic conditions. Liu and Zhou (2016) showed how the tight road/vehicle capacity constraints could invoke travelers' bounded rationality based on their day-to-day learning due to the inner system uncertainty, which is incurred when identical travelers are competing the limited resources without an assignment rule. This bound on trip cost is similar to (i) the concept of travel time budget in travel activity analysis, (ii) the changeable departure time of dynamic traffic assignment, and (iii) the required time windows for pickup and drop-off in vehicle routing problem. Therefore, it could be a proper way to use time windows to model travel behavior in future congested multi-modal transportation systems. In addition, the participants from different parties, such as, traffic regulator, mobility service providers, human drivers, and passengers, could lead a more complex leader-follower game in the future research.

Acknowledgements

This paper is supported by National Science Foundation – United States under Grant No. CMMI 1663657 "Collaborative Research: Real-time Management of Large Fleets of Self-Driving Vehicles Using Virtual Cyber Tracks". The work presented in this paper remains the sole responsibility of the authors.

References

Allahviranloo, M., Chow, J., 2019. A fractionally owned autonomous vehicle fleet sizing problem with time slot demand substitution effects. Transportation Research Part C, 98, 37-53

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., Rus, D., 2017. On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. Proceedings of the National Academy of Sciences, 114(3), 462-467.

Arnott, R., de Palma, A., Lindsey, R., 1990. Departure time and route choice for the morning commute. Transportation Research Part B: Methodological, 24(3), 209-228.

Baldacci, R., Bartolini, E., Mingozzi, A., 2011. An exact algorithm for the pickup and delivery problem with time windows. Operations research, 59(2), 414-426.

Beckmann, M.J., McGuire, C.B., Winsten, C.B., 1956. Studies in the Economics of Transportation. Yale University Press, New Haven, CT.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., Vance, P.H., 1998. Branch-and-price: Column generation for solving huge integer programs. *Operations research*, *46*(3), 316-329.

Batarce, M., Ivaldi, M., 2014. Urban travel demand model with endogenous congestion. Transportation Research Part A: Policy and Practice, 59, 331-345.

Behrisch, M., Bieker, L., Erdmann, J. and Krajzewicz, D., 2011. SUMO–simulation of urban mobility: an overview. In Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation. ThinkMind.

Boland, N., Christiansen, J., Dandurand, B., Eberhard, A., Linderoth, J., Luedtke, J., Oliveira, F., 2018. Combining Progressive Hedging with a Frank--Wolfe Method to Compute Lagrangian Dual Bounds in Stochastic Mixed-Integer Programming. SIAM Journal on Optimization, 28(2), 1312-1336.

Boyd, S., 2011, December. Alternating direction method of multipliers. In Talk at NIPS workshop on optimization and machine learning.

Boyd, S., Xiao, L., Mutapcic, A., Mattingley, J., 2007. Notes on decomposition methods. Notes for EE364B, Stanford University, 1-36.

Bruun, E. C., 2018. A research agenda and proposed research approaches to enable MaaS to bring maximal benefits. Paper presented at the 97th Transportation Research Board (TRB) Annual Meeting, Washington, D.C., United States.

Chen, D., Ahn, S., Chitturi, M., Noyce, D.A., 2017. Towards vehicle automation: Roadway capacity formulation for traffic mixed with regular and automated vehicles. Transportation research part B: methodological, 100, 196-221.

Chow, J.Y., Recker, W.W., 2012. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. Transportation Research Part B: Methodological, 46(3), 463-479.

Daganzo, C.F., 1994. The cell transmission model: a simple dynamic representation of highway traffic. Transp. Res. Part B 28 (4), 269–287.

Dantzig, G.B., Wolfe, P., 1960. Decomposition principle for linear programs. Operations research, 8(1), 101-111.

Davidson, P., Spinoulas, A., 2016. Driving alone versus riding together—How shared autonomous vehicles can change the way we drive. Road & Transport Research: A Journal of Australian and New Zealand Research and Practice, 25(3), 51-65.

Dey, K.C., Rayamajhi, A., Chowdhury, M., Bhavsar, P., Martin, J., 2016. Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network–Performance evaluation. Transportation Research Part C: Emerging Technologies, 68, 168-184.

Djavadian, S. and Chow, J.Y., 2017. An agent-based day-to-day adjustment process for modeling 'Mobility as a Service' with a two-sided flexible transport market. Transportation research part B: methodological, 104, 36-57.

Din, S., Paul, A. and Rehman, A., 2019. 5G-enabled Hierarchical architecture for software-defined intelligent transportation system. Computer Networks, 150, 81-89.

Drissi-Kaïtouni, O., Hameda-Benchekroun, A., 1992. A dynamic traffic assignment model and a solution algorithm. Transportation Science 26, 119–128.

Fagnant, D.J., Kockelman, K.M., 2014. The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios. Transportation Research Part C: Emerging Technologies, 40, 1-13.

Fisher, M.L., 1981. The Lagrangian relaxation method for solving integer programming problems. *Management science*, *27*(1), 1-18.

Florian, M., Constantin, I., Florian, D. 2009. A New Look at Projected Gradient Method for Equilibrium Assignment. Transportation Research Record, 2090(1), 10–16.

Frank, M., Wolfe, P., 1956. An algorithm for quadratic programming. Naval Research Logistics (NRL), 3(1-2), 95-110.

Gendreau, M., Jabali, O., Rei, W., 2016. 50th anniversary invited article—future research directions in stochastic vehicle routing. Transportation Science, 50(4), 1163-1173.

Gentili, M. and Mirchandani, P.B., 2012. Locating sensors on traffic networks: Models, challenges and research opportunities. Transportation research part C: emerging technologies, 24, 227-255.

Ghali, M.O., Smith, M.J., 1995. A model for the dynamic system optimum traffic assignment problem. Transportation Research Part B: Methodological, 29(3), 155-170.

Ghiasi, A., Hussain, O., Qian, Z.S., Li, X., 2017. A mixed traffic capacity analysis and lane management model for connected automated vehicles: A Markov chain method. Transportation Research Part B: Methodological, 106, 266-292.

Hanebutte, U., Doss, E., Ewing, T., Tentner, A., 1998. Simulation of vehicle traffic on an automated highway system. Math. Comput. Model. 27 (9–11), 129–141

Heikkilä, S., 2014. Mobility as a service—A proposal for action for the public administration, Case Helsinki. (Master of Science in Technology), Aalto University, Helsinki, Finland.

Hensher, D. A., 2017. Future bus transport contracts under a mobility as a service (MaaS) regime in the digital age: Are they likely to change? Transportation Research Part A: Policy and Practice, 98, 86-96.

Hyland, M., Mahmassani, H.S., 2018. Dynamic autonomous vehicle fleet operations: Optimization-based strategies to assign AVs to immediate traveler demand requests. Transportation Research Part C: Emerging Technologies, 92, 278-297.

Huisman, D., Jans, R., Peeters, M. and Wagelmans, A.P., 2005. Combining column generation and Lagrangian relaxation. In Column generation (pp. 247-270). Springer, Boston, MA.

Jayakrishnan, R., Tsai, W.T., Prashker, J.N., Rajadhyaksha, S., 1994. A faster path-based algorithm for traffic assignment.

Jittrapirom, P., Caiati, V., Feneri, A.-M., Ebrahimigharehbaghi, S., González, M. J. A., Narayan, J., 2017. Mobility as a service: A critical review of definitions, assessments of schemes, and key challenges. Urban Planning, 2(2), 13-25.

Kalafatas, G., 2010. A graph theoretic modeling framework for generalized transportation systems with congestion phenomena. Purdue University.

Kalvelagen, E., 2003. Dantzig-Wolfe Decomposition with GAMS. Amsterdam Optim. Model. Group LLC, Washington, DC, USA.

Kamargianni, M., Li, W., Matyas, M., Schäfer, A., 2016. A critical review of new mobility services for urban transport. Transportation Research Procedia, 14, 3294-3303.

Kok, A.L., Hans, E.W. and Schutten, J.M., 2012. Vehicle routing under time-dependent travel times: the impact of congestion avoidance. Computers & operations research, 39(5), 910-918.
Larsson, T., Patriksson, M., 1992. Simplicial decomposition with disaggregated representation for the traffic assignment problem. Transportation Science, 26(1), 4-17.
Larsson, T., Patriksson, M. and Rydergren, C., 2004. A column generation procedure for the side constrained traffic equilibrium problem. *Transportation Research Part B: Methodological*, *38*(1), 17-38.
Lavieri, P.S., Garikapati, V.M., Bhat, C.R., Pendyala, R.M., Astroza, S., Dias, F.F., 2017. Modeling individual preferences for ownership and sharing of autonomous vehicle technologies. Transportation Research Record: Journal of the Transportation Research Board, (2665), 1-10.
LeBlanc LJ, Morlok EK, Pierskalla WP, 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research 9: 309–318
Levin, M.W., Boyles, S.D., 2016. A multiclass cell transmission model for shared human and autonomous vehicle roads. Transportation Research Part C: Emerging Technologies, 62, 103-116.
Levin, M.W., Kockelman, K.M., Boyles, S.D., Li, T., 2017. A general framework for modeling shared autonomous vehicles with dynamic network-loading and dynamic ride-sharing application. Computers, Environment and Urban Systems, 64, 373-383.
Li, Y., Waller, S.T., Ziliaskopoulos, T., 2003. A decomposition scheme for system optimal dynamic traffic assignment models. *Networks and Spatial Economics*, *3*(4), 441-455.
Lim, H., Taeihagh, A., 2018. Autonomous vehicles for smart and sustainable cities: An in-depth exploration of privacy and cybersecurity implications. Energies, 11(5), 1062.
Lin, D.Y., Valsaraj, V., Waller, S.T., 2011. A Dantzig-Wolfe decomposition-based heuristic for off-line capacity calibration of dynamic traffic assignment. *Computer‐Aided Civil and Infrastructure Engineering*, *26*(1), 1-15.
Liu, J., Zhou, X., 2016. Capacitated transit service network design with boundedly rational agents. Transportation Research Part B: Methodological, 93, 225-250.
Liu, J., Kang, J.E., Zhou, X., Pendyala, R., 2018. Network-oriented household activity pattern problem for system optimization. Transportation Research Part C: Emerging Technologies, 94, 250-269.
Lu, C.C., Liu, J., Qu, Y., Peeta, S., Rouphail, N.M., Zhou, X., 2016. Eco-system optimal time-dependent flow assignment in a congested network. Transportation Research Part B: Methodological, 94, 217-239.
Lu, C.C., Mahmassani, H.S., Zhou, X., 2009. Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. Transportation Research Part B: Methodological, 43(3), 345-364.
Ma, J., Li, X., Zhou, F., Hao, W., 2017. Designing optimal autonomous vehicle sharing and reservation systems: A linear programming approach. Transportation Research Part C: Emerging Technologies, 84, 124-141.
Ma, T., Rasulkhani, S., Chow, J., Klein, S., 2018. A dynamic ridesharing dispatch and idle vehicle repositioning strategy with integrated transit transfers. https://arxiv.org/abs/1901.00760
Maciejewski, M., Bischoff, J., 2016. Congestion effects of autonomous taxi fleets. Transport. 1-10
Mahmassani, H.S., 2016. 50th anniversary invited article—autonomous vehicles and connected vehicle systems: flow and operations considerations. Transportation Science, 50(4), 1140-1162.
Mahmoudi, M., Zhou, X., 2016. Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. Transportation Research Part B: Methodological, 89, 19-42.
Mourad, A., Puchinger, J., Chu, C., 2019. A survey of models and algorithms for optimizing shared mobility. Transportation Research Part B, in press.

Mulley, C., 2017. Mobility as a services (MaaS)—Does it have critical mass? Transport Reviews, 37(3), 247-251.
Munoz, J.C., Laval, J.A., 2006. System optimum dynamic traffic assignment graphical solution method for a congested freeway and one destination. Transportation Research Part B: Methodological, 40(1), 1-15.
Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic, part I: general theory. Transp. Res. Part B: Methodological. 27 (4), 281–287.
Newell, G.F., 2002. A simplified car-following theory: a lower order model. Transp. Res. Part B: Methodol. 36 (3), 195–205
Palomar, D.P., Chiang, M., 2006. A tutorial on decomposition methods for network utility maximization. IEEE Journal on Selected Areas in Communications, 24(8), 1439-1451.
Papadimitratos, P., Fortelle, A.L., Evenssen, K., Brignolo, R. and Cosenza, S., 2009. Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation. IEEE communications magazine, 47(11), 84-95.
Peeta, S., Mahmassani, H.S., 1995. System optimal and user equilibrium time-dependent traffic assignment in congested networks. Annals of Operations Research, 60(1), 81-113.
Peeta, S., Ziliaskopoulos, A.K., 2001. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and spatial economics*, *1*(3-4), 233-265.
Psaraftis, Harilaos N., Min Wen, and Christos A. Kontovas. "Dynamic vehicle routing problems: Three decades and counting." Networks 67, no. 1 (2016): 3-31.
Qian, Z.S., Shen, W., Zhang, H.M., 2012. System-optimal dynamic traffic assignment with and without queue spillback: Its path-based formulation and solution via approximate path marginal cost. Transportation research part B: methodological, 46(7), 874-893.
Qu, F., Wang, F.Y. and Yang, L., 2010. Intelligent transportation spaces: vehicles, traffic, communications, and beyond. IEEE Communications Magazine, 48(11), 136-142.
Rayle, L., Dai, D., Chan, N., Cervero, R., & Shaheen, S. (2016). Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco. Transport Policy, 45, 168-178.
Reece, D.A., Shafer, S.A., 1993. A computational model of driving for autonomous vehicles. *Transportation Research Part A: Policy and Practice*, *27*(1), 23-50.
Rossi, F., Zhang, R., Hindy, Y., Pavone, M., 2018. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. Autonomous Robots 42, 1427-1442
Salazar, M., Rossi, F., Schiffer, M., Onder, C.H., Pavone, M., 2018. On the Interaction between Autonomous Mobility-on-Demand and Public Transportation Systems. arXiv preprint arXiv:1804.11278.
Shen, W, Nie, Y, Zhang, HM., 2007. On path marginal cost analysis and its relation to dynamic system-optimal traffic assignment. in Proceedings of the 17th International Symposium on Transportation and Traffic Theory.
Small, K.A., 1982. The scheduling of consumer activities: Work trips. American Economic Review 72, 467-479.
Small, K.A., 2015. The bottleneck model: An assessment and interpretation. Economics of Transportation, 4(1-2), 110-117.
Small, K.A., Verhoef, E.T., 2007. The Economics of Urban Transportation. Routledge (Taylor & Francis), London.
Stern, R.E., Cui, S., Delle Monache, M.L., Bhadani, R., Bunting, M., Churchill, M., Hamilton, N., Pohlmann, H., Wu, F., Piccoli, B., Seibold, B., 2018. Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments. *Transportation Research Part C: Emerging Technologies*, *89*, 205-221.

Sun, W., Zheng, J. and Liu, H.X., 2017. A capacity maximization scheme for intersection management with automated vehicles. Transportation research procedia, 23, 121-136.

Talebpour, A., Mahmassani, H.S., 2016. Influence of connected and autonomous vehicles on traffic flow stability and throughput. *Transportation Research Part C: Emerging Technologies*, *71*, 143-163.

Taniguchi, E. and Shimamoto, H., 2004. Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times. Transportation Research Part C: Emerging Technologies, 12(3-4), 235-250.

Tong, L., Pan, Y., Shang, P., Guo, J., Xian, K., Zhou, X., 2019. Open-Source Public Transportation Mobility Simulation Engine DTALite-S: A Discretized Space–Time Network-Based Modeling Framework for Bridging Multi-agent Simulation and Optimization. Urban Rail Transit, 5, 1-16.

Toth, P., Vigo, D. eds., 2002. The vehicle routing problem. Society for Industrial and Applied Mathematics.

Van Arem, B., van Driel, C.J.G., Visser, R., 2006. The impact of cooperative adaptive cruise control on traffic-flow characteristics. IEEE Trans. Intell. Transport. Syst. 7 (4), 429–436.

Varaiya, P., 1993. Smart cars on smart roads: problems of control. *IEEE Transactions on automatic control*, *38*(2), 195-207.

Vickrey, W.S., 1969. Congestion theory and transport investment. The American Economic Review, 59(2), 251-260.

Wardrop, JG., 1952. Some theoretical aspects of road traffic research. Proceedings of the Institution of Civil Engineers, Part II 1: 325–378.

Wei, Y., Avcı, C., Liu, J., Belezamo, B., Aydın, N., Li, P.T., Zhou, X., 2017. Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models. Transportation research part B: methodological, 106, 102-129.

Wong, Y. Z., Hensher, D. A., 2017. The Thredbo story: A journey of competition and ownership in land passenger transport. Paper presented at the 15th International Conference on Competition and Ownership in Land Passenger Transport (Thredbo 15), Stockholm, Sweden.

Wu, X., Nie, L., Xu, M., Zhao, L., 2019. Distribution planning problem for a high-speed rail catering service considering time-varying demands and pedestrian congestion: A lot-sizing-based model and decomposition algorithm. Transportation Research Part E, 123, 61-89.

Yang, H., Huang, H.J., 2005. Mathematical and economic theory of road pricing.

Yao, Y., Zhu, X., Dong, H., Wu, S., Wu, H., Zhou, X., 2018. An Alternating Direction Method of Multiplier Based Problem Decomposition Scheme for Iteratively Improving Primal and Dual Solution Quality in Vehicle Routing Problem, submitted to Transportation Research Part B

Ye, L., Yamamoto, T., 2018. Modeling connected and autonomous vehicles in heterogeneous traffic flow. Physica A: Statistical Mechanics and its Applications, 490, 269-277.

Zhang, H.M., Nie, Y., Qian, Z., 2013. Modelling network flow with and without link interactions: the cases of point queue, spatial queue and cell transmission model. Transportmetrica B: Transport Dynamics, 1(1), 33-51.

Zhou, X., Tong, L., Mahmoudi, M., Zhuge, L., Yao, Y., Zhang, Y., Shang, P., Liu, J., Shi, T., 2018. Open-source VRPLite Package for Vehicle Routing with Pickup and Delivery: A Path Finding Engine for Scheduled Transportation Systems. Urban Rail Transit, 4(2), 68-85.