Autonomous Vehicle Assignment and Routing in Congested Networks

Jiangtao Liu (jliu215@asu.edu)
Xuesong Zhou (xzhou74@asu.edu)
School of Sustainable Engineering and the Built Environment

Pitu Mirchandani (pitu@asu.edu)
School of Computing, Informatics, and Decision Systems Engineering (CIDSE)

Arizona State University
Outline

1. Introduction
2. Problem Statement
3. Space-Time-State Network Flow Models
4. Dantzig-Wolfe Decomposition Algorithm
5. Preliminary Experiments
1. Introduction

Ride Sharing Companies

Information-sharing technology
1. Introduction

The “three revolutions” in the future transportation systems

- Automation
- Shared Use
- Electrification
Future urban transportation systems: **integrated multi-modal scheduled transportation system**

1. Mobility as a service via an integrated platform

2. Mass public transit forms the backbone of the mobility system

3. Feet of shared AVs fulfills first and last mile needs

How to optimize demand, **supply** and infrastructure?
1. Introduction

Key questions:

- How many autonomous vehicles do we need?
- How many passengers can we serve?
- How to capture the new traffic congestion?
- What is the best vehicle routing and vehicle-to-passenger assignment solution?
2. Problem Statement

Traffic Assignment Problem:

- **Network**: Physical traffic network
- **Objective function**: System Optimal or User Equilibrium
- **Road capacity**: considered to capture road congestions
- **Vehicle-to-passenger assignment**: given in advance
- **Passenger trip request**: has the same origin and destination as that of the assigned vehicles
- **Vehicle carrying capacity**: not explicitly considered
- **Variable**: usually continuous vehicle flow
2. Problem Statement

Vehicle Routing Problem:

- **Network**: Virtual point-to-point network
- **Objective function**: System Optimal
- **Traffic congestion**: not explicitly considered
- **Vehicle-to-passenger assignment**: will be found
- **Passenger trip request**: has specific pick-up and drop-off location with time windows
- **Vehicle carrying capacity**: considered
- **Variable**: discrete vehicle routing and scheduling
2. Problem Statement

Keywords:

- **Physical traffic network** to consider traffic congestion
- **Trip requests** with Pickup and delivery with time windows
- **Autonomous vehicles with carrying capacity** for ride sharing
- **Central control (System optimal)**
Link Capacity

Without link capacity:
Total cost is 6

With link capacity:
Total cost is 15
System Optimal Coordination

Without coordination (selfish routing):
Total cost is 15

With coordination:
Total cost is 9
The red vehicle can wait until time 3 to pick up passenger 2, so the blue vehicle can pick up passenger 1 at exact time 3.

The optimal result doesn’t only optimize the vehicle routing, but also the departure time of picked up passengers.
When the red vehicle’s carrying capacity is increased to 2, the total cost is reduced to 4 from 9;

Only the red vehicle is required to serve the trip requests.
The Challenge of solving system optimal vehicle routing with pickup and drop-off location and time windows in congested physical traffic networks:

System-impact of adjusting one vehicle routing:
- System marginal travel time
- System marginal passenger service benefit/cost

In this queuing system:
- Waiting time for individual: 4 min

After adding one more person in the queue:
- Societal travel time: additional 4 min for each person behind: +16 min, and the waiting time of added person is 4 min, so the system marginal waiting time is 20 min.
- Societal service benefit: some persons may not be served in their preferred time window and it decreases the service benefit.
Marginal cost calculation in system optimal dynamic traffic assignment (SODTA)

Link marginal delay equals this area, which equals this grey area

One unit of flow

Ghali and Smith (1995)

Our Approach 1: Marginal Cost Calculation

**Step 1:** Build **virtual pickup and drop-off links** in physical traffic networks, and its service pricing is converted to generalized link travel cost.

**Step 2:** find one **initial solution** as the input.

**Step 3:** Perform **network loading** within a **space-time-state network**
   3.1 use **cumulative arrival and departure counts** to derive the link marginal travel cost.
   3.2 update the marginal service link benefit of passengers (not served or served by multiple vehicles)

**Step 4:** find the **new least-marginal-cost route** for each vehicles, and go to step 3; otherwise, stop.
Our Approach 2: Dantzig-Wolfe Decomposition

Restricted master problem
- Road capacity constraint
- Passenger service constraint

Solved by standard solvers

Marginal costs from the solver

Subproblems
(Time-dependent State dependent shortest path problem for each vehicle)

Vehicle routing solution

Solved by the beam-search algorithm in space-time-state networks
3. Space-time-state network and model formulation

(a) Physical transportation network with vehicles and trip requests

(b) Modified transportation network with virtual pickup and delivery nodes and links
3. Space-time-state network and model formulation

Time-extended Space-time network construction for physical path
1 → 2 → 4 → 6

Arc \((i,j,t,s)\) with capacity

Vertex \((i,t), (j,s)\)

Passenger pickup and drop-off
time windows and locations are
embedded in this network
3. Space-time-state network and model formulation

**Challenge:** how to model the logit of passenger pickup and delivery with vehicle carrying capacity

One more dimension-> Vehicle Carrying States:
which passengers are being carried by this vehicle:

To record the passenger service status:

0: the passenger is not served;
1: under served (picked up but not delivered);
2: finished (delivered)

**Example:** In the case: if vehicle capacity is 1 and 2 passengers trip requests,

All possible states: [ ], [1[1]], [1[2]], [2[1]] or [2[2]]
3. Space-time-state network and model formulation

Vehicle carrying state transition logit: focusing on specific passenger 1

- 0->0: vehicle departs at the origin depot
- 0->1[1]: passenger 1 is picked up at his location within time window.
- 1[1]->1[2]: passenger 1 is dropped off at his location within time window
- 1[2]->0: vehicle arrives at the destination depot.
- Once one passenger is picked up, he/she will always be dropped off, because 1[1]->0 is not a feasible state transition.
- No passenger will be served if the vehicle carrying capacity is fully used.
3. Space-time-state network and model formulation

Vehicle State Transition with specific locations and time intervals

Before serving any passengers

After passenger 1 is picked up at node 7 at the allowed time, this state keeps unchanged until passenger 1 is delivered

After passenger 1 is delivered at node 10 at the allowed time, the state keeps unchanged and the vehicle capacity is reduced by 1

Virtual ending state connecting those feasible ending states at the destination depot

After passenger 2 is picked up at node 8 at the allowed time, this state keeps unchanged until passenger 2 is delivered

After passenger 2 is delivered at node 10 at the allowed time, the state keeps unchanged and the vehicle capacity is reduced by 1

Vehicle carrying state transition graph
3. Space-time-state network and model formulation

One possible vehicle trajectory in a space-time-state network

Vertex: \((i, t, w)\)

Arc:
\((i, j, t, s, w, w')\)
3. Space-time-state network and model formulation

Mathematical formulation:

\[ \begin{align*}
\text{Min } Z &= \sum_v \sum_{i,j,t,s,w,w'r}(c_{i,j,t,s} \times x_{i,j,t,s,w,w'}) \\
\text{(1) Flow balance constraint for each vehicle:} \\
\sum_{i,j,t,s} x_{i,j,t,s,w,w'} - \sum_{j,i,t,s} x_{i,j,t,s,w,w'} &= \begin{cases} 
-1 & j = O(v), s = DT(v), w = [0,0, \ldots] \\
1 & j = D(v), s = T, w = [0,0, \ldots] \\
0 & \text{otherwise}
\end{cases}, \forall \ v \\
\text{(2) Passenger } p \text{ pick-up request at } (o, d, \tau): \\
\sum_v \sum_{i,j,t,s,w,w'r} x_{i,j,t,s,w,w'} &= 1, \forall p \\
\text{(3) Tight road capacity constraint (point queue model)} \\
\sum_v \sum_w x_{i,j,t,s,w,w'} &\leq cap_{i,j,t,s}, \forall (i,j,t,s) \\
\text{(4) Binary variables} \\
x_{i,j,t,s,w,w'} &= \{0,1\}
\end{align*} \]
3. Space-time-state network and model formulation

Capture queue spillback:

Inflow arc capacity constraint:
\[
\sum_w x_{i,j',t-FFTT_{i,j}+1,t,w,wr} \leq Cap_{i,j',t-FFTT_{i,j}+1,t}, \forall (i,j') \in L_{inf low}, \forall t \quad (6)
\]

Outflow arc capacity constraint:
\[
\sum_w x_{j',j,t,t+1,w,wr} \leq Cap_{j',j,t,t+1}, \forall (j',j) \in L_{out flow}, \forall t \quad (7)
\]

Link storage capacity constraint:
\[
\sum_w x_{j',j',t-1,t,w,wr} + \sum_w \sum_{s=t-FFTT_{i,j}}^{s=t-1} x_{i,j',s,t,w,wr} \leq Len_{i,j}, \times n_{i,j}, \times Jam_{i,j}, \forall (i,j') \in L_{inf low}, \forall t \quad (8)
\]

Newell’s simplified kinematic wave model (Newell, 1993)
\[
\sum_w \sum_{s=t-BWTT(i,j)}^{s=t} x_{j',j',s-1,s,w,wr} + \sum_w \sum_{s=t-FFTT_{i,j}}^{s=t-1} x_{i,j',s,s+FFTT_{i,j},w,wr} \leq Len_{i,j}, \times n_{i,j}, \times \text{Jam}_{i,j}, \forall (i,j') \in L_{inf low}, \forall t \quad (9)
\]
4. Dantzig-Wolfe Decomposition Algorithm

Objective function

\[ \text{Min } Z = \sum_v \sum_{(i,j,t,s,w,w')} (c_{i,j,t,s,w,w'}^v \times x_{i,j,t,s,w,w'}^v) \]

Subject to,

(1) Flow balance constraint for each vehicle:

\[ \sum_{i,t,w:(i,j,t,s,w,w')} x_{i,j,t,s,w,w'}^v - \sum_{i,t,w:(i,j,t,s,w',w')} x_{i,j,t,s,w',w'}^v = \begin{cases} -1 & j = O(v), s = DT(v), w = [0,0,...,0] \\ 1 & j = D(v), s = T, w = [F] \\ 0 & \text{otherwise} \end{cases}, \forall v \]

(2) Passenger \( p \)'s pick-up request constraint

\[ \sum_v \sum_{i,t,s:(i,j,t,s,w,w')} x_{i,j,t,s,w,w'}^v = 1, \forall p \]

(3) Tight road capacity constraint (point queue model)

\[ \sum_v \sum_w x_{i,j,t,s,w,w'}^v \leq c \alpha p_{i,j,t,s}, \forall (i,j,t,s) \]

(4) Binary definitional constraint

\[ x_{i,j,t,s,w,w'}^v \in \{0,1\} \]

Special block: time-dependent state-dependent shortest path problem

- A special block can be solved by our VRP solution engine
- can be the subproblem in Dantzig-Wolfe decomposition
4. Dantzig-Wolfe Decomposition Algorithm

Restricted master problem

\[
\text{Min } \sum_v \sum_{(i,j,t,s,w,w')} [c_{i,j,t,s,w,w'} \times \sum_k (\lambda^v_k \times x^p_k \times \delta_{i,j,t,s,w,w'})]
\]  \hspace{1cm} (9)

Pick-up constraint:

\[
\sum_v \sum_{(i,j,t,s,w,w')} \sum_k (\lambda^v_k \times x^p_k \times \delta^{v,k}_{i,j,t,s,w,w'}) = 1, \forall p
\]  \hspace{1cm} (10)

Capacity constraint:

\[
\sum_v \sum_w \sum_k (\lambda^v_k \times x^p_k \times \beta^{v,k}_{i,j,t,s,w,w'}) \leq \text{cap}_{i,j,t,s}, \forall (i,j,t,s)
\]  \hspace{1cm} (11)

Column selection:

\[
\sum_k \lambda^v_k = 1, \forall v
\]  \hspace{1cm} (12)

\[
\lambda^v_k = \{0,1\}
\]  \hspace{1cm} (13)
4. Dantzig-Wolfe Decomposition Algorithm

Subproblems (TDSDSP)

\[ \text{Min } Z' = \sum_{i,j,t,s,w,w'} (c_{i,j,t,s,w,w'}^\nu \times x_{i,j,t,s,w,w'}^\nu) - \sum_p \sum_{(i,j,t,s,w,w') \in A(p)} (\pi_p \times x_{i,j,t,s,w,w'}^\nu) - \sum_{i,j,t,s} (\mu_{i,j,t,s} \times \sum_w x_{i,j,t,s,w,w'}^\nu) - \omega_\nu \]  \hspace{1cm} (17)

Subject to

\[ \sum_{i,j,t,s,w,w'} x_{i,j,t,s,w,w'}^\nu - \sum_{j,i,s,t,w',w} x_{j,i,s,t,w',w}^\nu = \begin{cases} -1 & j = O(v), s = DT(v), w = [0,0,...,0] \\ 1 & j = D(v), s = T, w = [F] \\ 0 & otherwise \end{cases} \]  \hspace{1cm} (18)

Dual variable/ marginal cost of passenger pickup constraints

Dual variable/ marginal cost of congestion constraints

Dual variable/ marginal cost of path weight constraints
4. Dantzig-Wolfe Decomposition Algorithm

1. Initialization

   Feasible solution finding (paths)

2. Restricted Master Problem: Integer linear programming

   Dual price of side constraints

3. Sub-problems:
   Time-dependent state-dependent least cost path finding

   Reduced cost

   Non-negative

   No

   Yes

   Optimal solution
4. Dantzig-Wolfe Decomposition Algorithm

A tree-based path representation

- One-to-all shortest path tree is generated for one vehicle
- Directly obtain the shortest path for vehicles with a same depot and departure time but different destination depot
- Does not need to run the shortest path algorithm again to improve the computation efficiency.
5. Preliminary Experiments

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>24</td>
</tr>
<tr>
<td># of links</td>
<td>84</td>
</tr>
<tr>
<td># of trip requests</td>
<td>30</td>
</tr>
<tr>
<td>(pickup and dropoff</td>
<td></td>
</tr>
<tr>
<td>with time windows)</td>
<td></td>
</tr>
<tr>
<td># of available</td>
<td>30</td>
</tr>
<tr>
<td>autonomous vehicles</td>
<td></td>
</tr>
<tr>
<td># of depots</td>
<td>5</td>
</tr>
<tr>
<td>optimization time</td>
<td>110</td>
</tr>
<tr>
<td>horizon (time unit)</td>
<td></td>
</tr>
<tr>
<td>Vehicle capacity (person)</td>
<td>1</td>
</tr>
</tbody>
</table>
5. Preliminary Experiments

<table>
<thead>
<tr>
<th>Vehicle_No</th>
<th>Passenger_No</th>
<th>Vehicle_No</th>
<th>Passenger_No</th>
<th>Vehicle_No</th>
<th>Passenger_No</th>
</tr>
</thead>
</table>
5. Preliminary Experiments

Dantzig-Wolfe decomposition algorithm solution:

- Each passenger has specific pickup and drop-off location and time windows
- The **vehicle benefit** of serving one passenger is 20
- The **vehicle waiting cost** is the half of the waiting time

<table>
<thead>
<tr>
<th></th>
<th>Number of required vehicles</th>
<th>Total travel cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial solution</td>
<td>30</td>
<td>1096</td>
</tr>
<tr>
<td>vehicle carrying capacity is 1</td>
<td>27</td>
<td>967.5</td>
</tr>
<tr>
<td>vehicle carrying capacity is 2</td>
<td>25</td>
<td>869.5</td>
</tr>
</tbody>
</table>

Take **vehicle 9** as an example:

- In initial solution: picks up passenger 17 -> drops off passenger 17;
- Vehicle carrying capacity is 1: picks up passenger 17 -> drops off passenger 17-> picks up passenger 29 -> drops off passenger 29
- Vehicle carrying capacity is 2: picks up passenger 17 -> drops off passenger 17-> picks up passenger 30-> picks up passenger 29-> drops off passenger 29-> drops off passenger 30
6. Summary

Our goals:

- Minimize the **system-level** travel cost, including vehicle travel time and service benefits
- Satisfy **passengers’ trip requests** with pickup and drop-off location and time windows
- Consider the **road congestion** incurred by assigned vehicles

**Future Extension**: multi-modal scheduled transportation system: human-driven vehicles, autonomous vehicles, and public transit systems.
6. Summary

Required knowledge:

- Dynamic Traffic Assignment and Vehicle Routing Problem
- Dantzig-Wolfe decomposition algorithm
- Time-dependent State-dependent shortest path problem
- Space-time-state network construction and network flow model
Space-time-state network flow models and vehicle routing problem:


Dynamic Traffic Assignment and Traffic flow model:


Dantzig-Wolfe Decomposition algorithm

THANK YOU