Simplifying Modeling Complexity In Dynamic Transportation Systems: A State-space-time Network-based Framework

Xuesong Zhou (xzhou74@asu.edu; xzhou99@gmail.com)
Associate professor, School of Sustainable Engineering and the Built Environment
Arizona State University

Prepared for Workshop IV: Decision Support for Traffic
Long Program New Directions in Mathematical Approaches for Traffic Flow Management
Institute for Pure & Applied Mathematics, UCLA

Outline

1. Introduction

   large-scale dynamic traffic assignment and simulation

2. From simulation to optimization:
   - Modeling next-generation of transportation systems
   - Key Questions: modeling challenges
   - Problem Definition: VRPPDTW
   - Methodology: combination of dynamic programming and Lagrangian relaxation

3. Extensions

   Traffic flow state estimation, and traffic signal control and train timetabling…

Background

Xuesong Zhou

- Pronounced as “Su-song Joe”

Acronym for extending traffic user equilibrium and system optimum models to next generation

Research areas

Simulation-based mesoscopic dynamic traffic assignment

DYNASMART

DTALite (based on simplified kinematic wave model)

Traffic state estimation and prediction

Train routing and scheduling

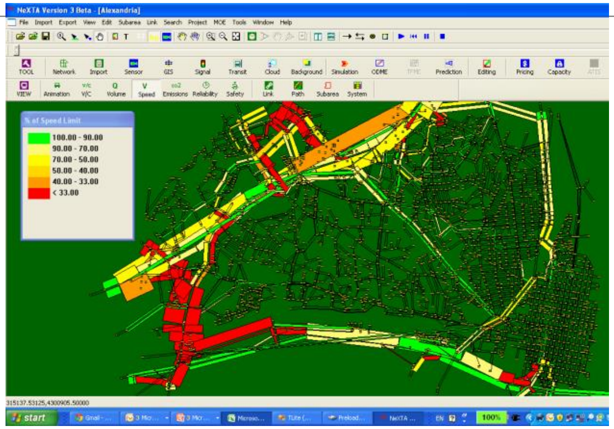
→ Vehicle routing and scheduling (new)

Topic 1:
Introduction: large-scale dynamic traffic assignment and simulation

Open-source Free Software Package

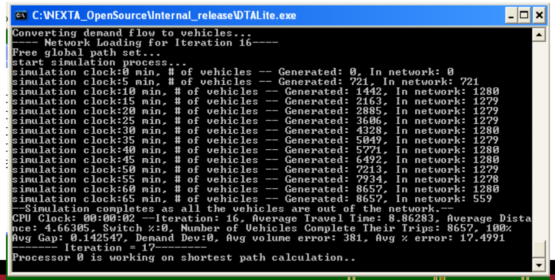
NEXTA: front-end Graphical User Interface GUI (C++)

- https://github.com/xzhou99/dtalite_beta_test



DTALite: Open-source computational engine (C++)

- Light-weight and agent-based DTA
- Simplified kinematic wave model (Newell)
- Built-in OD demand matrix estimation (ODME) program
- Emission prediction (light-weight MOVES interface)
  - Simplified car follow modeling (Newell)



Learning Traffic Network Modeling using Open-source tools

www.learning-transportation.org

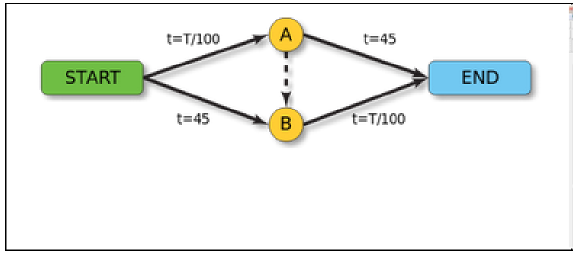
12 lessons:

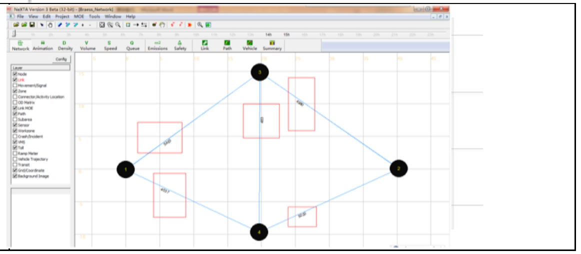
Lesson 1: Let Us Create a Transportation Network

Lesson 2: From Population to Driving Trips:

Lesson 3: Remove Roads to Speed Travel?

Lesson 4: Optimize Traffic Signal Lights

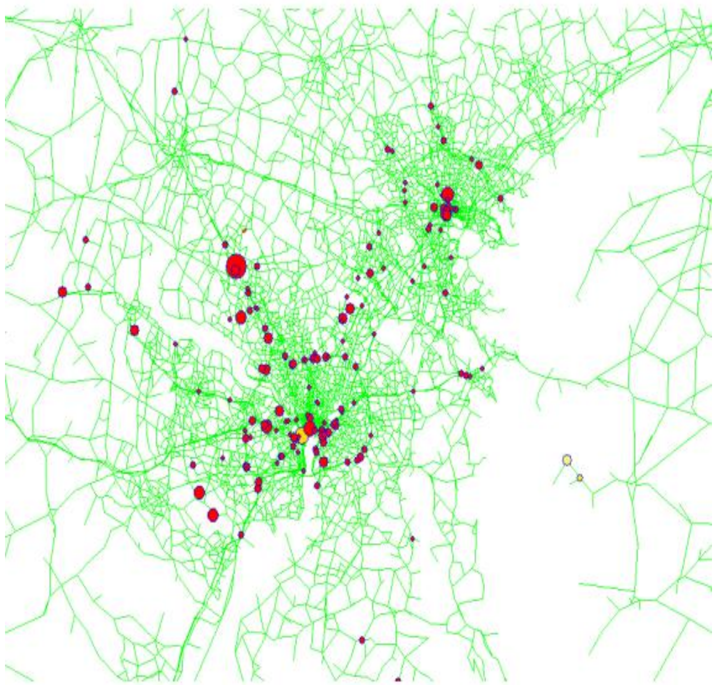




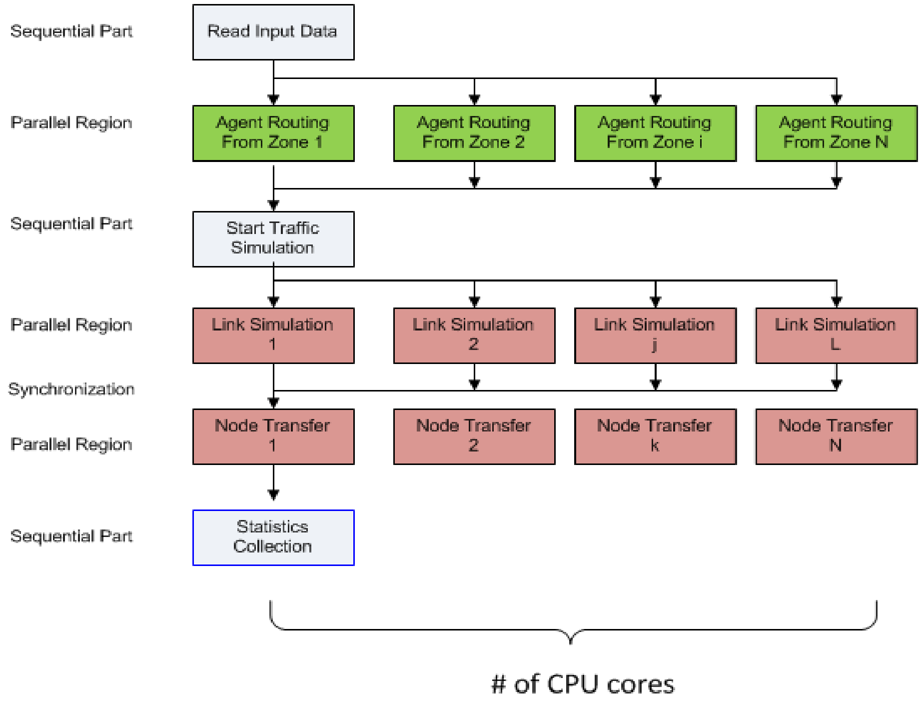
Learning Goals:

1. *Understand modeling principles of user equilibrium, and gap functions*
2. *Know how to setup BPR parameters for special link types*
3. *Understand the impact of random incidents and analyze traffic at link, path and network levels*
4. *Understand different network equilibrium method: method of successive average vs. day to day learning*
5. *Understand the impact of road pricing to resolve Braess's Paradox*

Computational Challenges

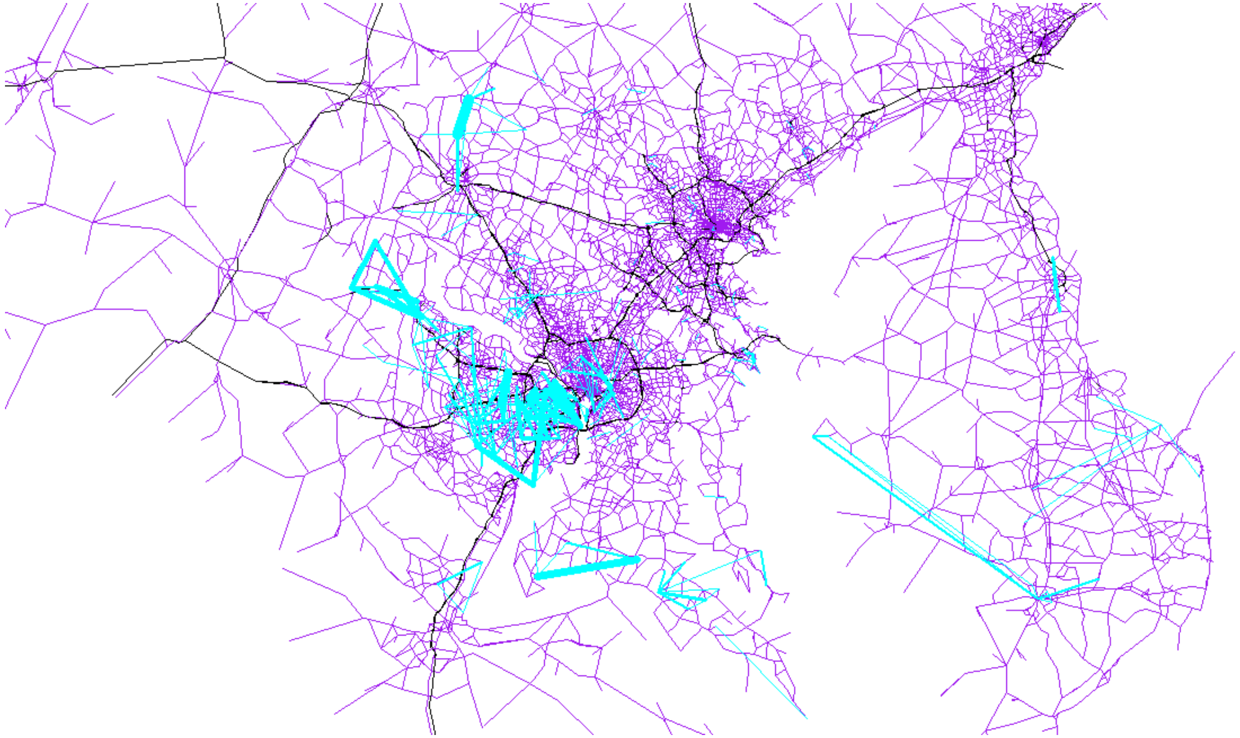


Shared memory-based parallel computing for agent-based path finding and mesoscopic traffic simulation (based on OpenMP)



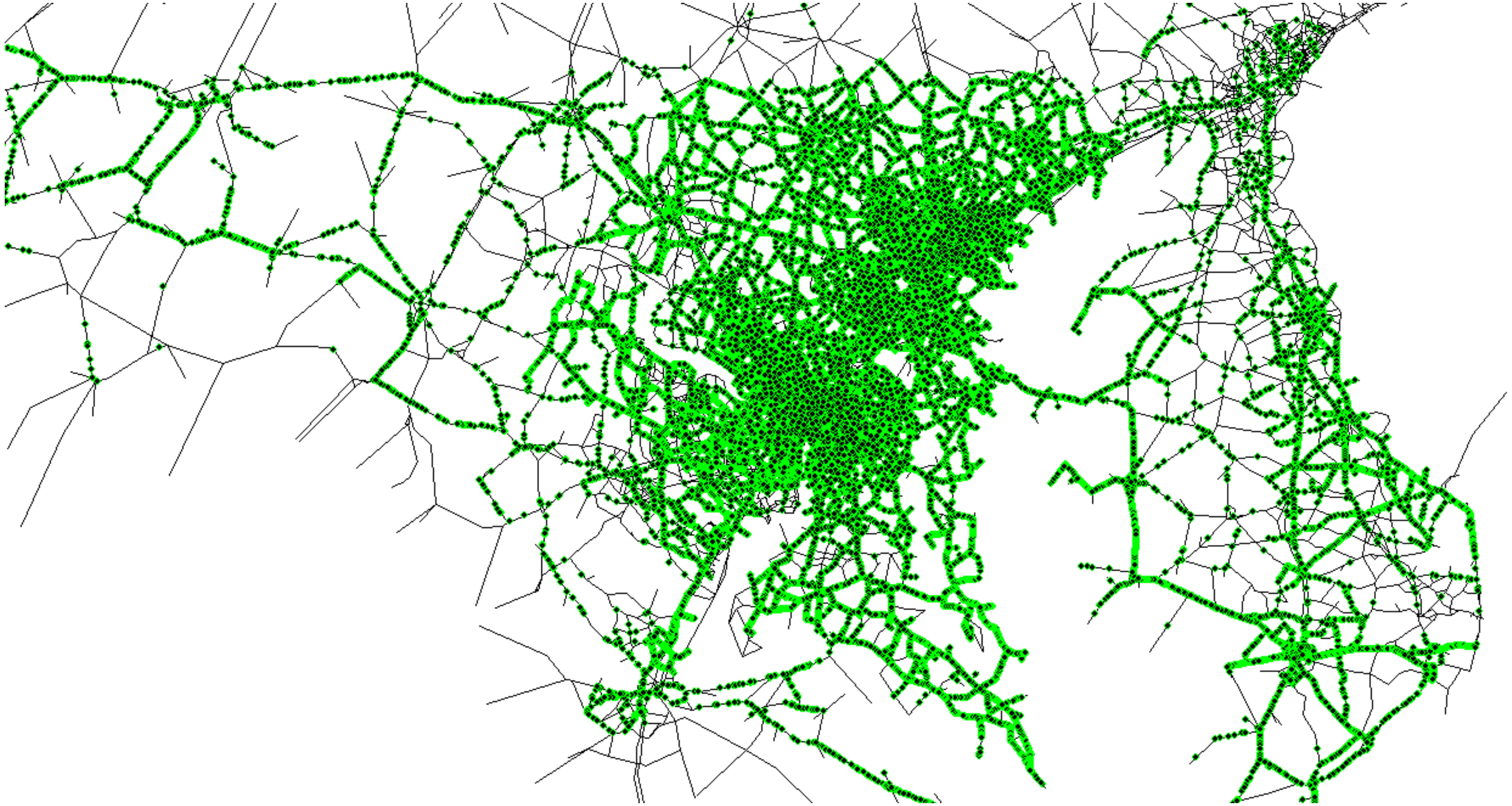
Maryland State-wide model:
20 K nodes, 47K links, 3,000 zones, 18 M agents
CPU time: 30 min per UE iteration on a 20-core workstation with 194 GB RAM

Origin-Destination Demand Spatial Distribution Pattern

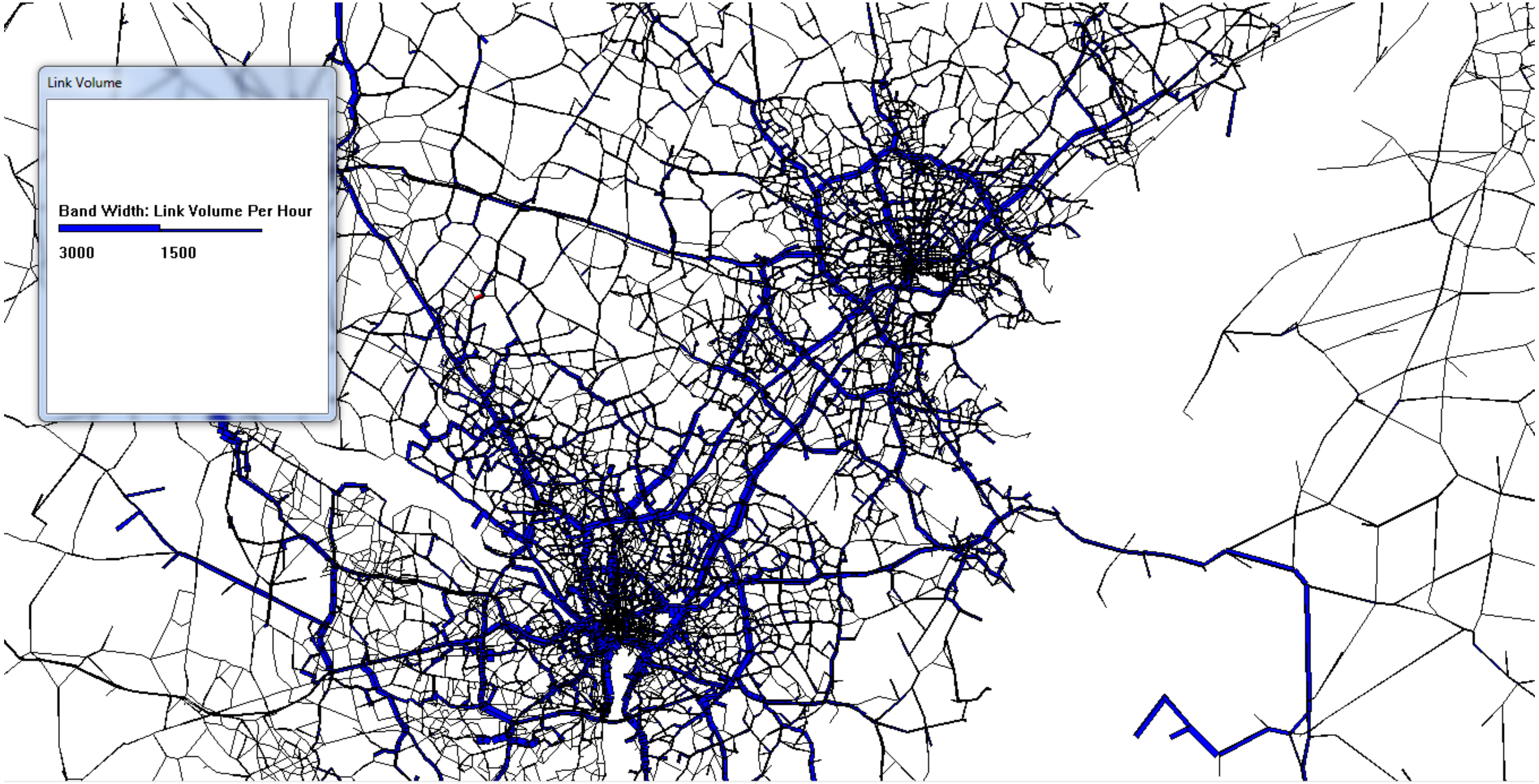


Collaboration with University of Maryland and Maryland State Highway Administration
Supported by TRB SHRP II Program

Vehicle Animation at Network Level

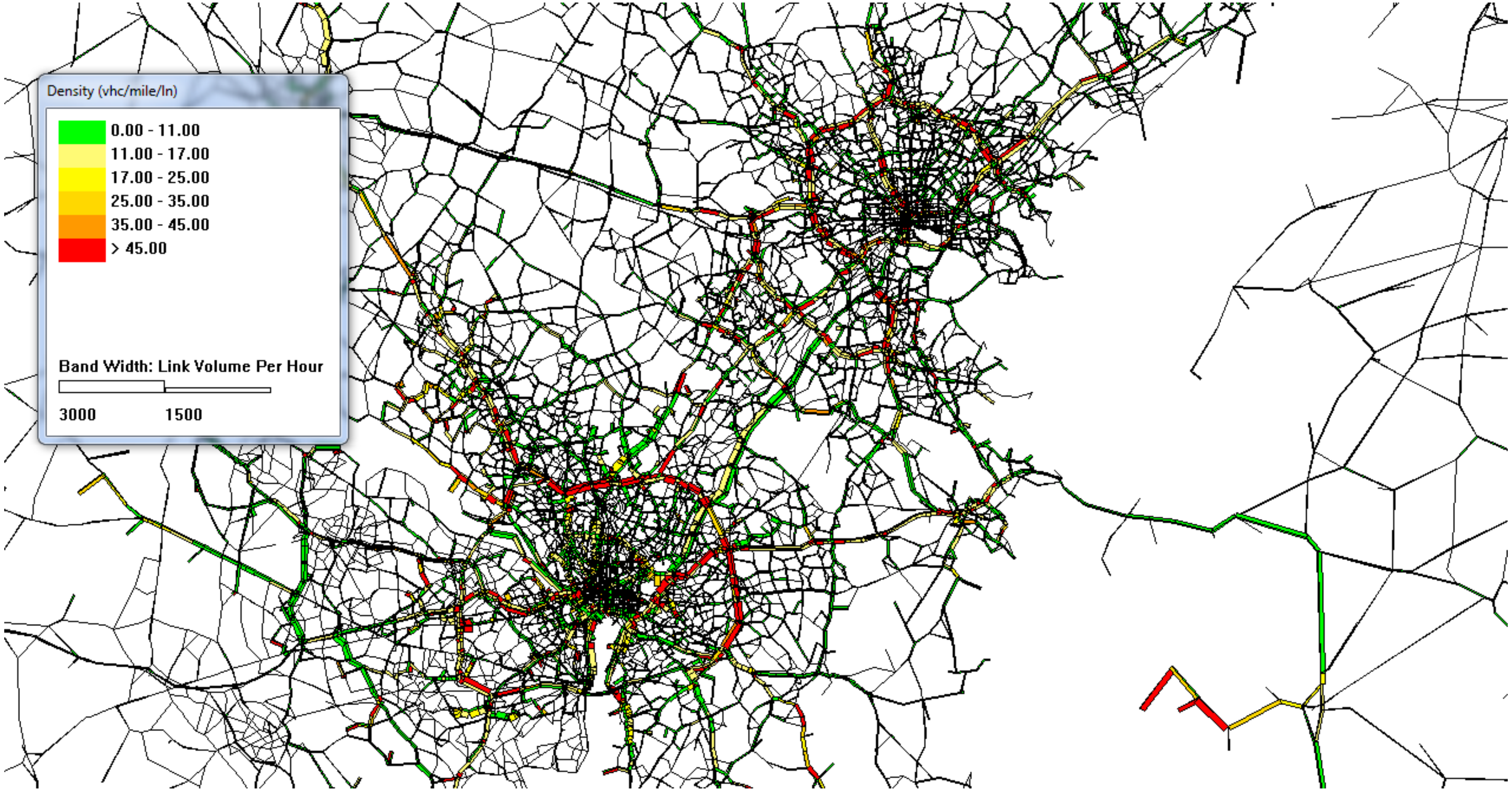


Volume at Network Level



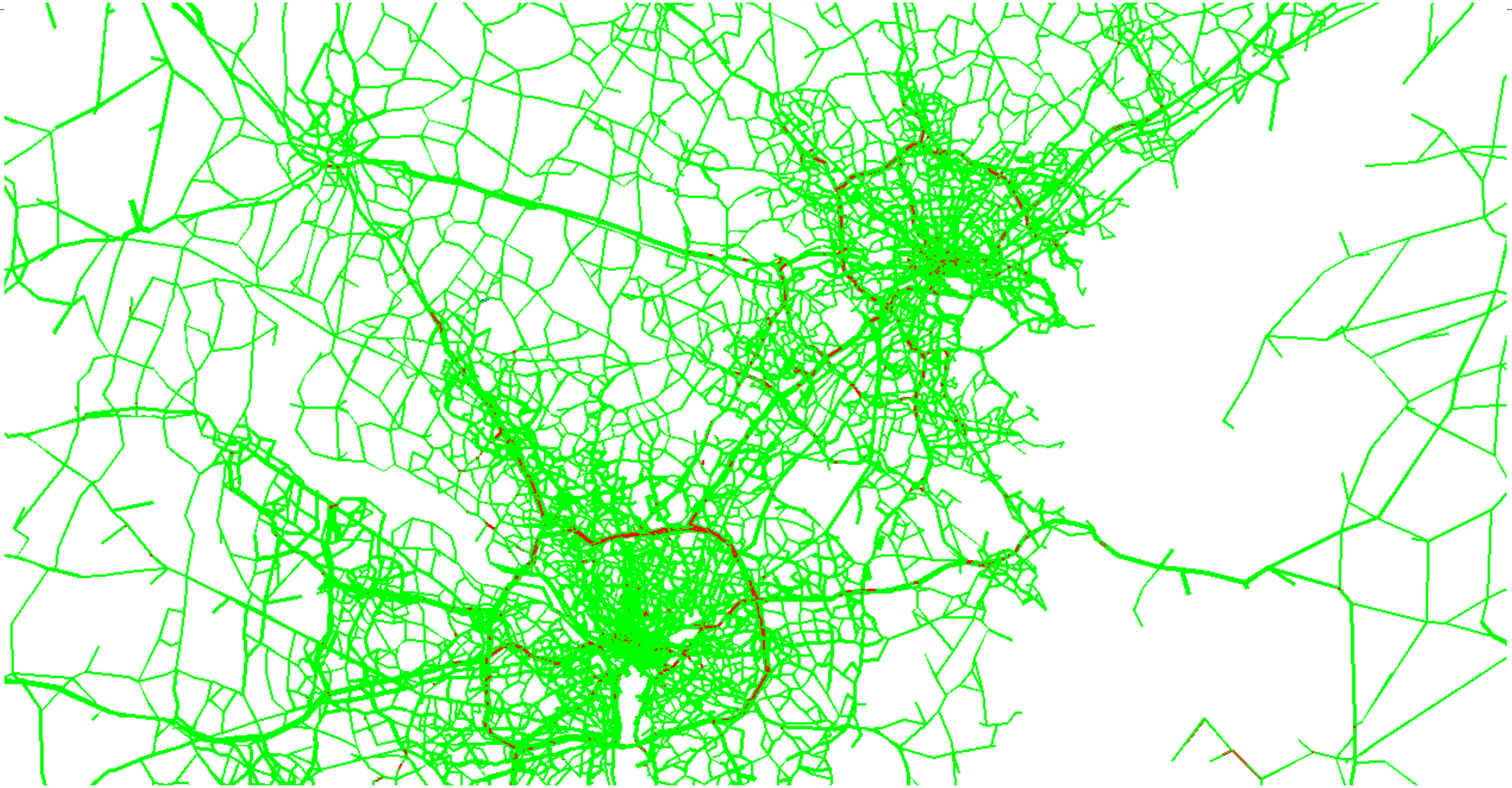
Band width of a link is proportional to link volume

Density at Network Level



Speed at Network Level

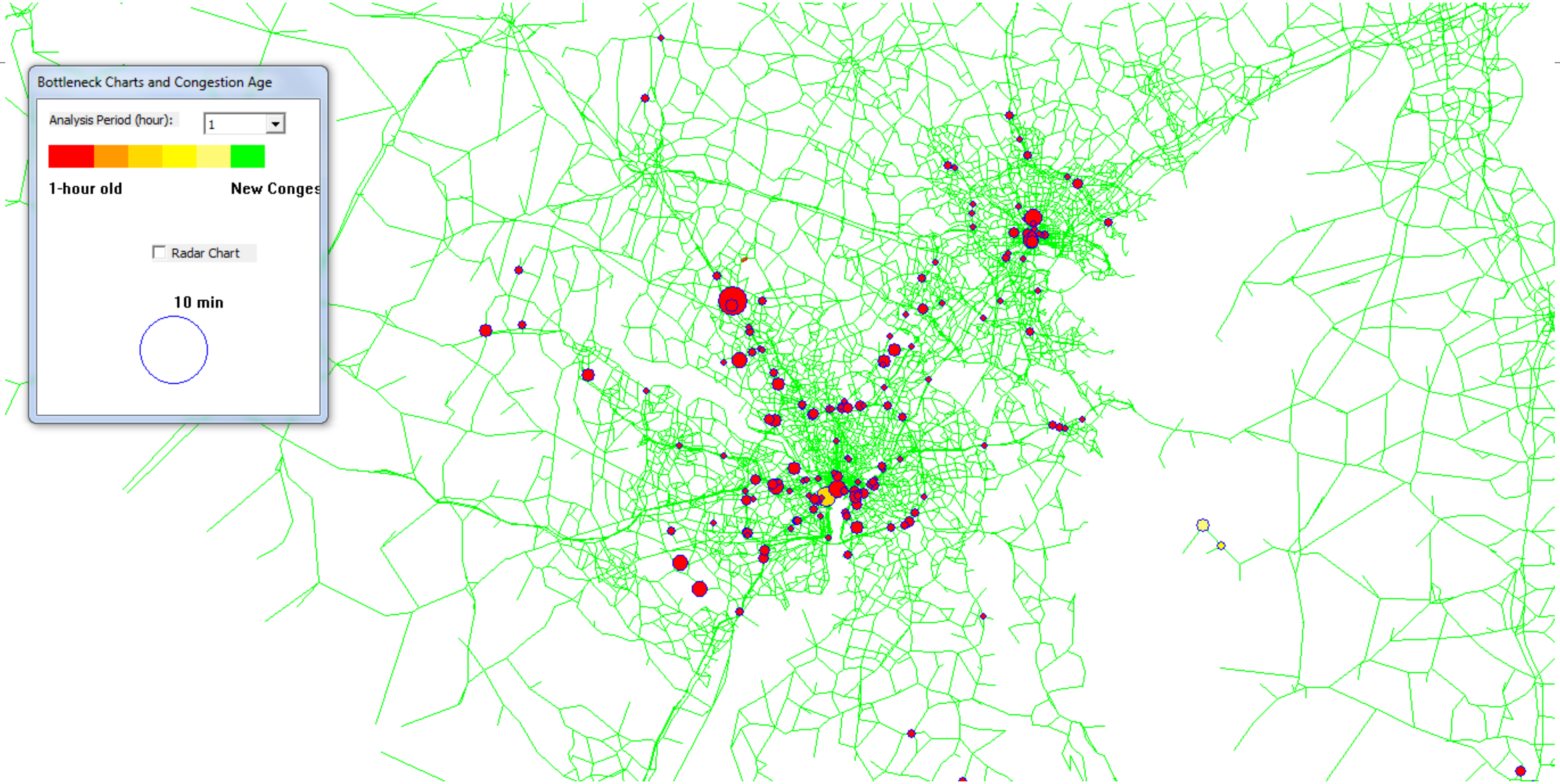
Queue at Network Level



Queue Duration at Network Level

Link width represents duration of congestion (e.g., 60 min vs. 120 min)

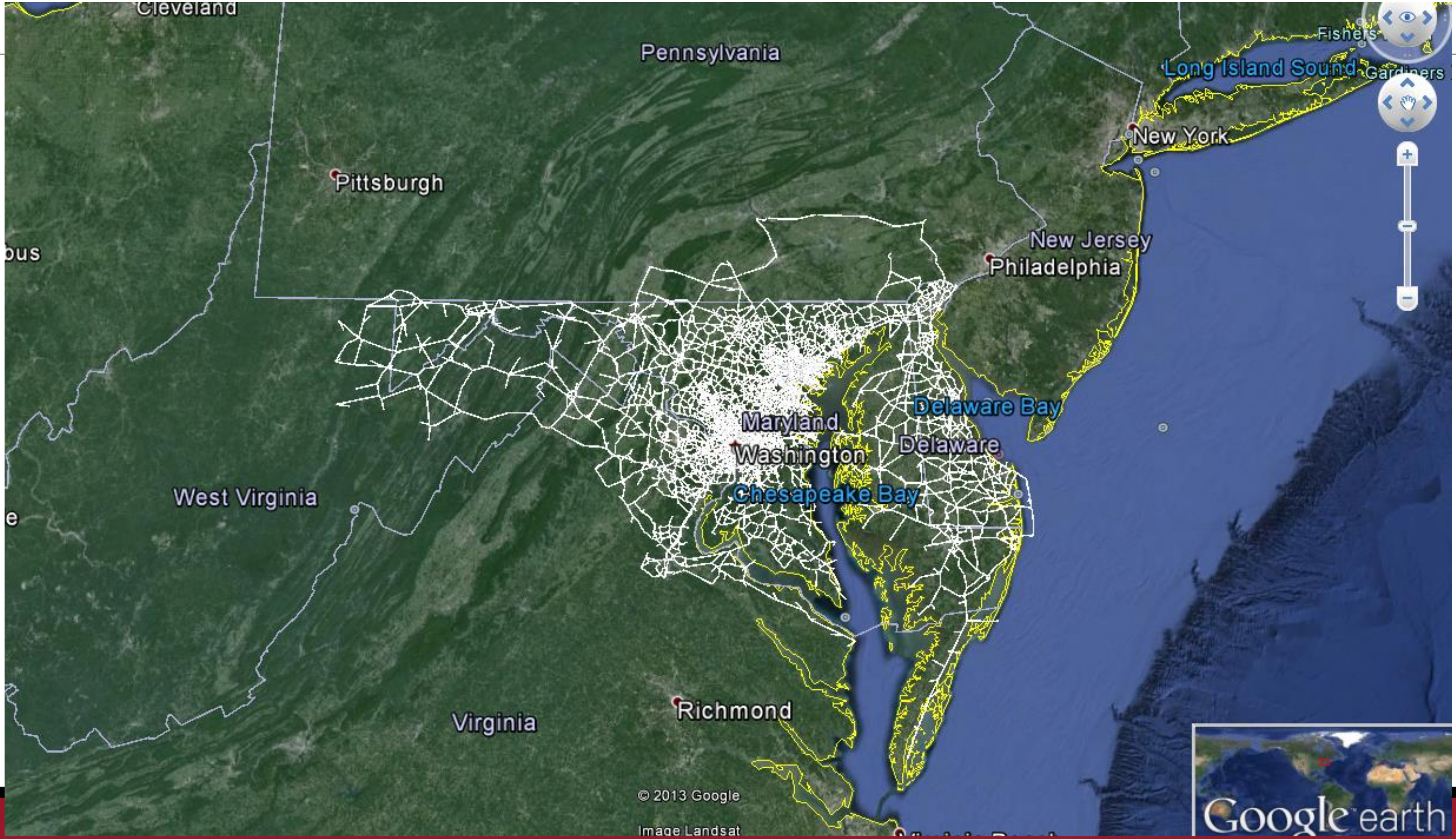
Time-dependent Bottleneck Locations



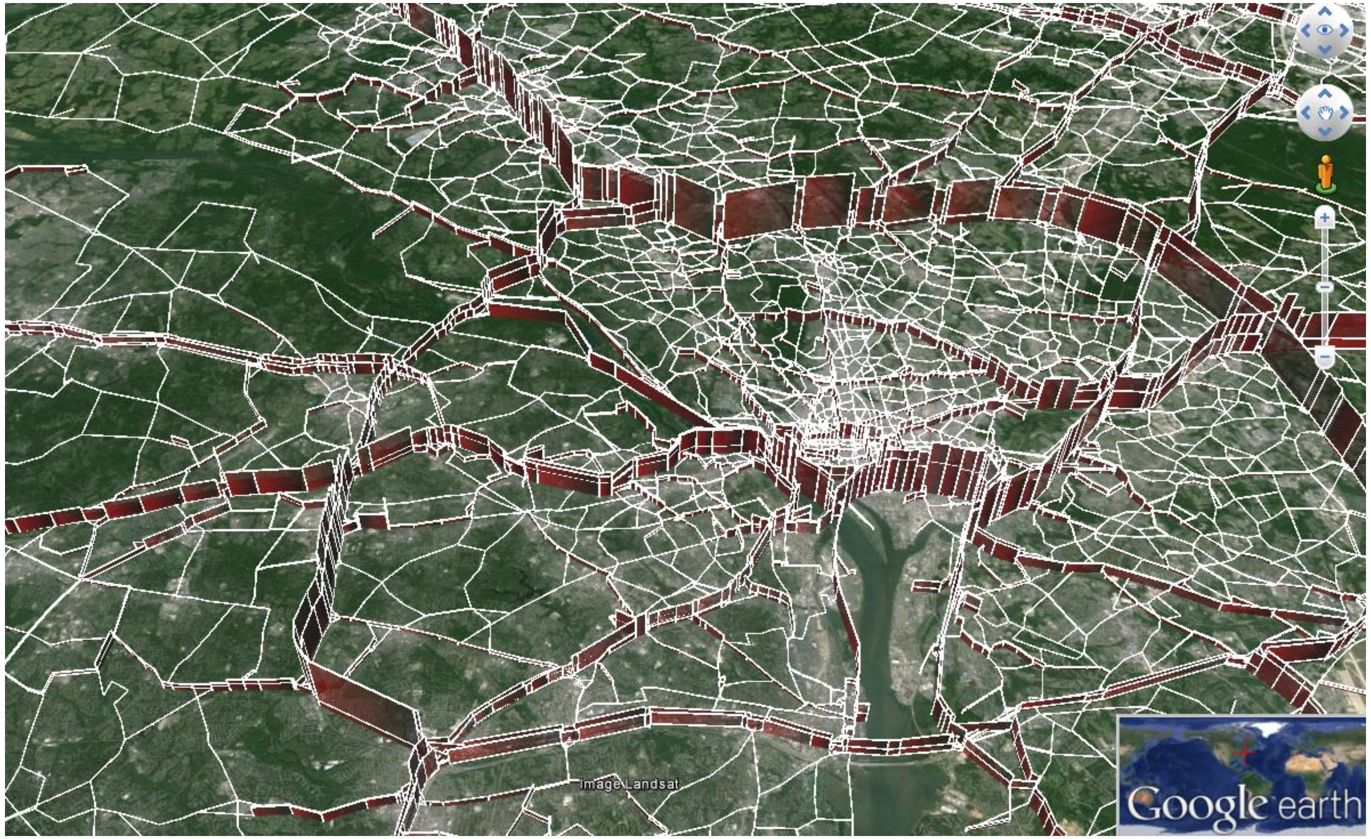
Size of a circle represents the total delay at one node

Color of a circle represents the age of congestion (to identify the congestion propagation sequence)

Statewide Network Coverage in Google Earth



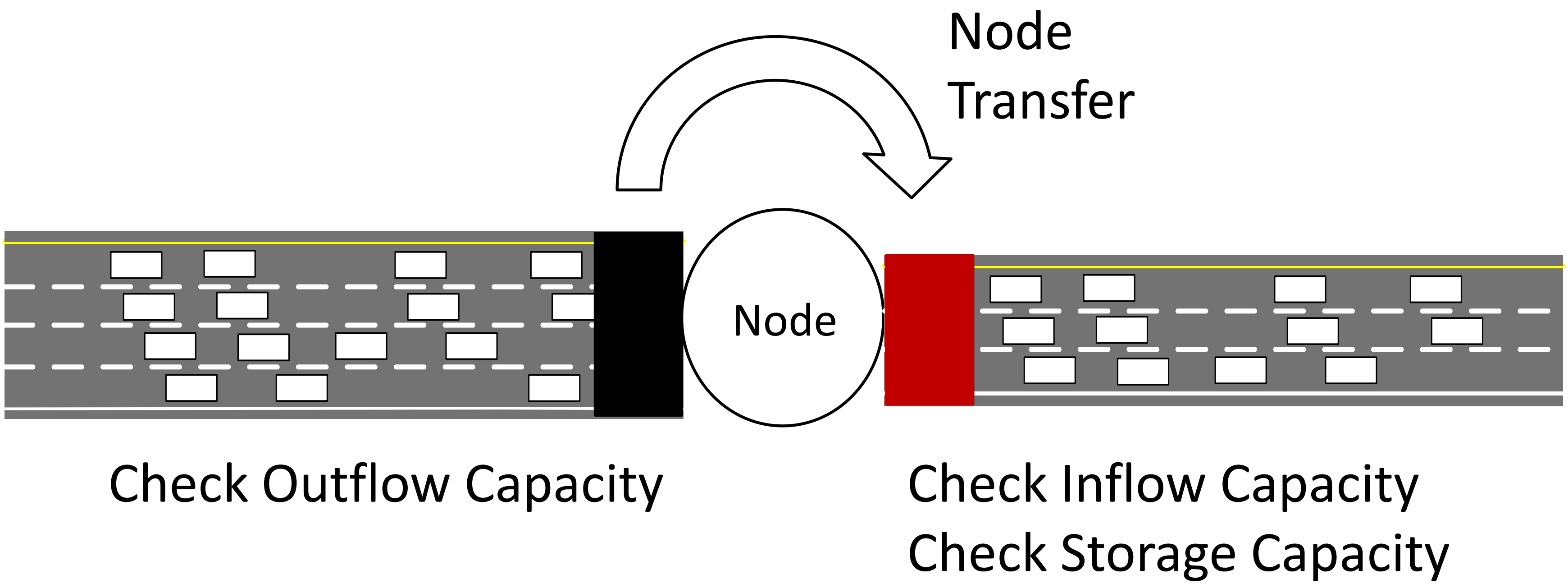
Volume Display in Google Earth



Height as volume

Inside: Simplified Event-based Traffic Simulator

Node transfer



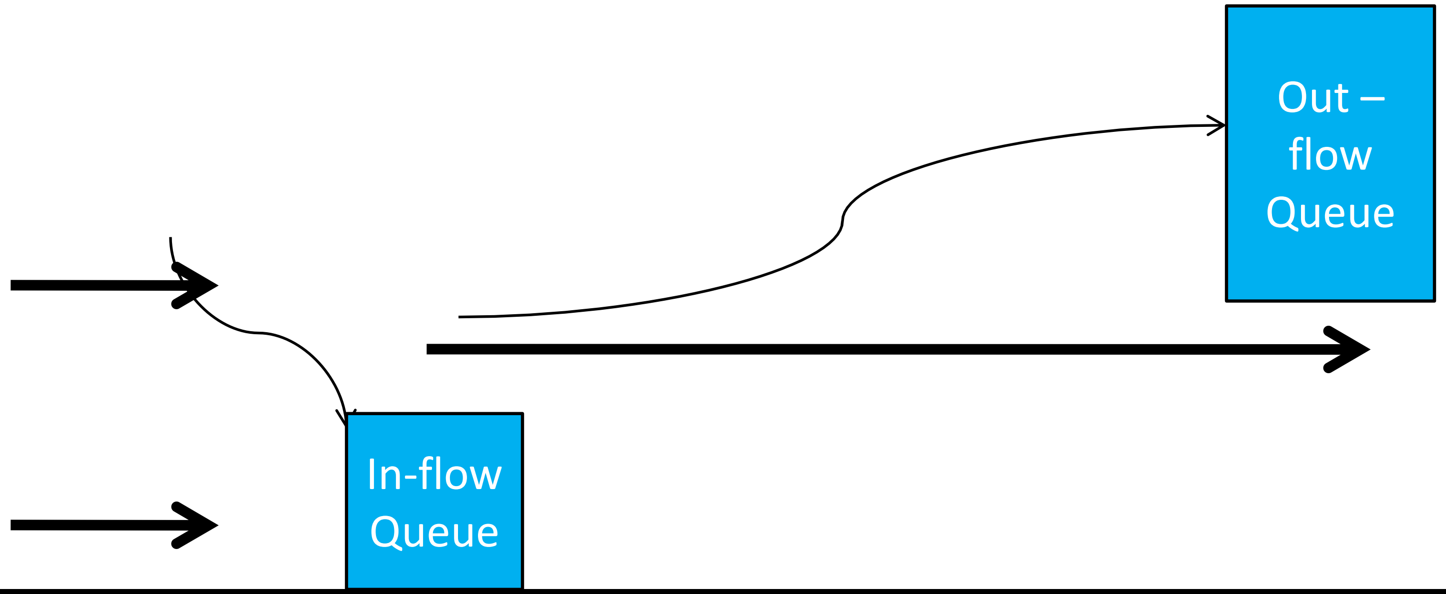
Multiple Traffic Flow Models

Point queue (relaxed storage constraints)

Spatial queue

Newell's model (i.e. Link Transmission Model)

- Shockwave propagation



Available capacity at every simulation interval

Traffic Flow Model (on the Link)

Queue propagation

- Inflow capacity = outflow capacity

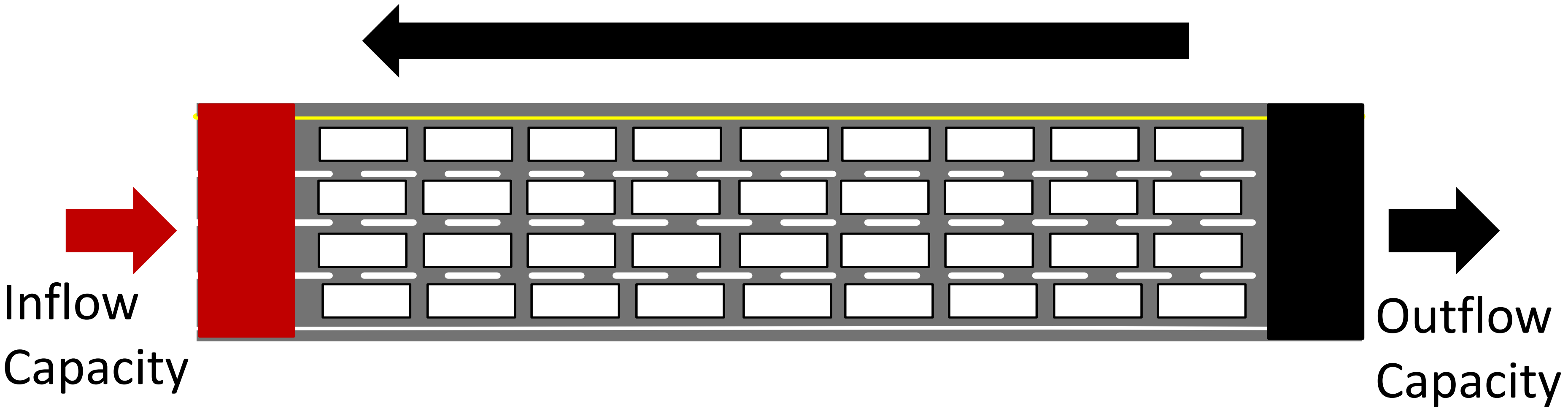
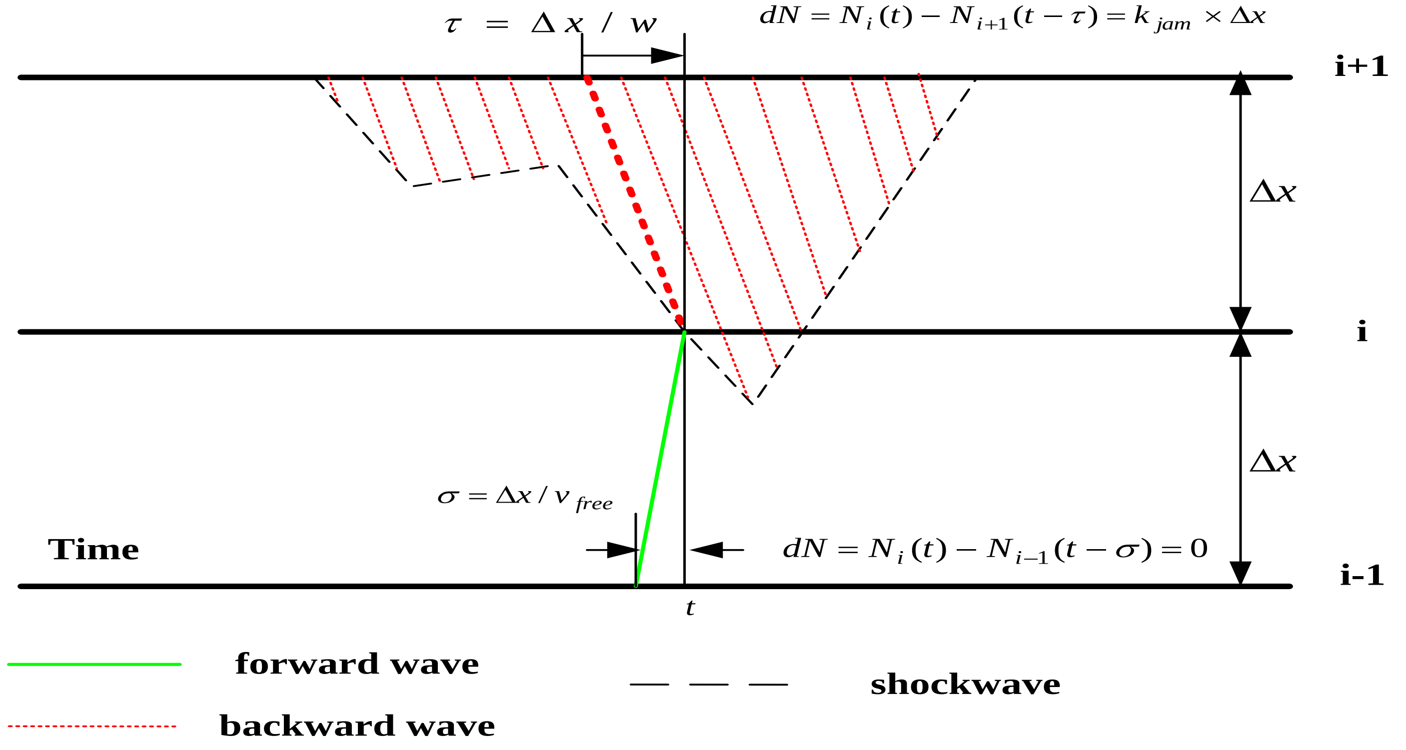
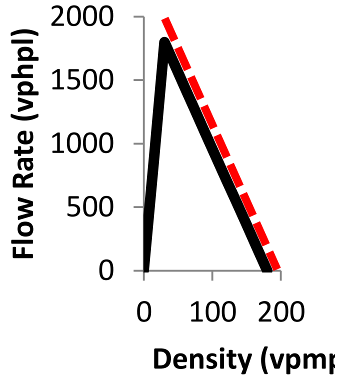
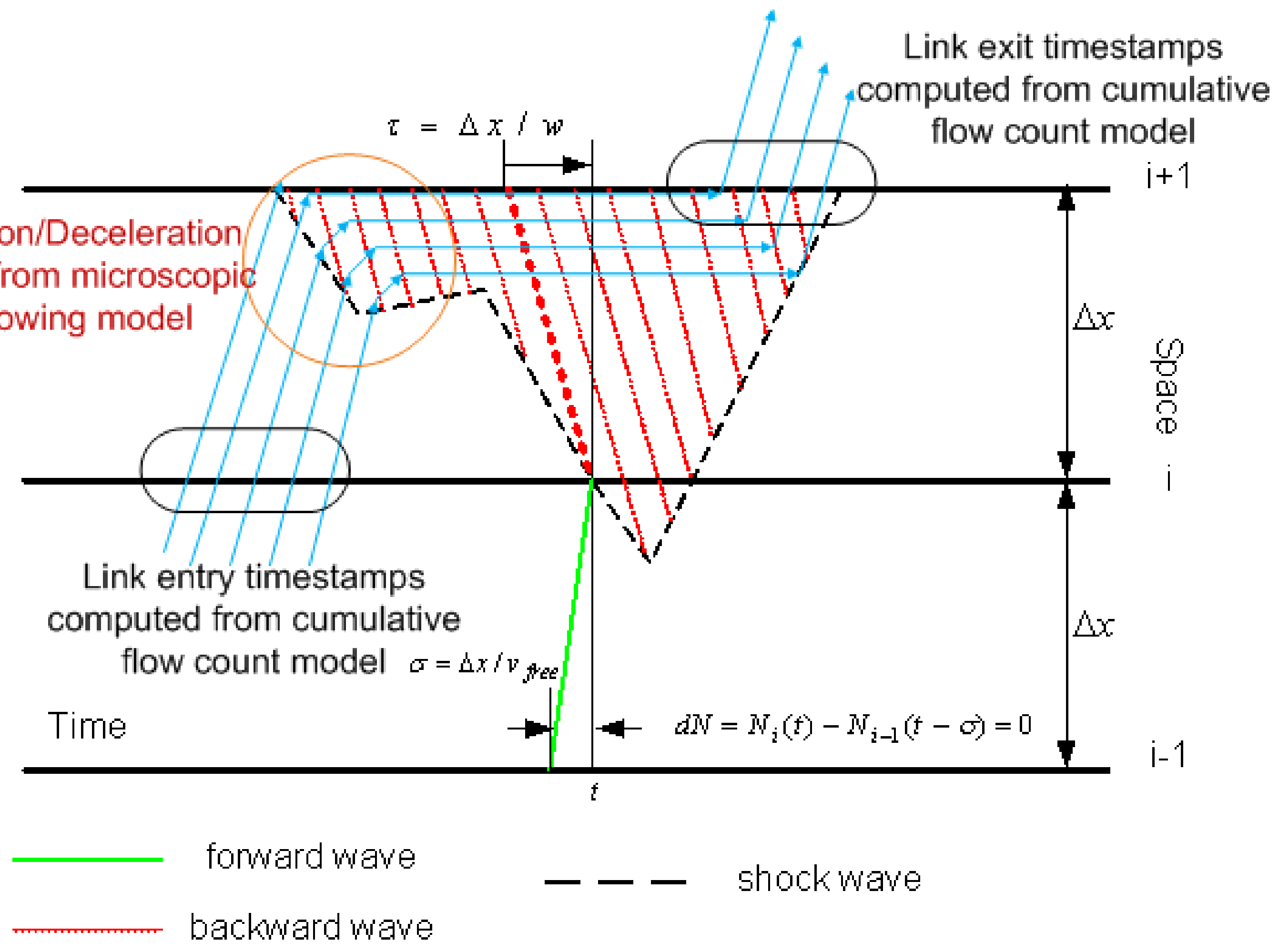


Illustration of N-Curve Computation For Tracking Queue Spillback





Construct Microscopic Vehicle Trajectory from Mesoscopic Simulation Results using Consistent Simplified Kinematic Wave Model and Simplified Car following Model



White Paper: DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration:
Cogent Engineering (2014): http://www.tandfonline.com/doi/abs/10.1080/23311916.2014.961345

Mesoscopic Dynamic Traffic Assignment for Emission Evaluation

Emission Result Aggregation

Microscopic Vehicle Trajectory Reconstruction

Emission Estimates

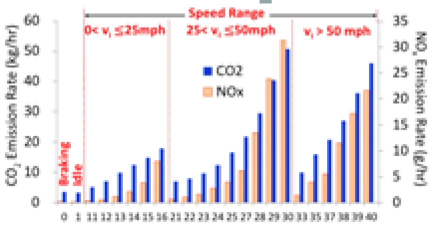
Project level

MOVES Lite

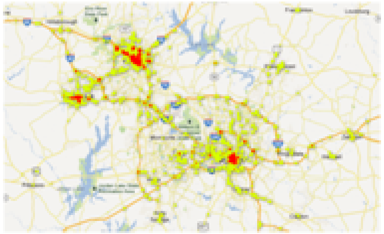
Simplified Emission Estimation Method

DTALite

Large-scale Dynamic Traffic Assignment & Simulator



Network level



Zhou, X., S. Tanvir, H. Lei, J. Taylor, B. Liu, N. M. Rouphail, H. C. Frey. (2015) Integrating a Simplified Emission Estimation Model and Mesoscopic Dynamic Traffic Simulator to Efficiently Evaluate Emission Impacts of Traffic Management Strategies. Transportation Research Part D: Transport and Environment. 37, 123-136

Topic 2:
Modeling next-generation of transportation systems: from simulation to optimization

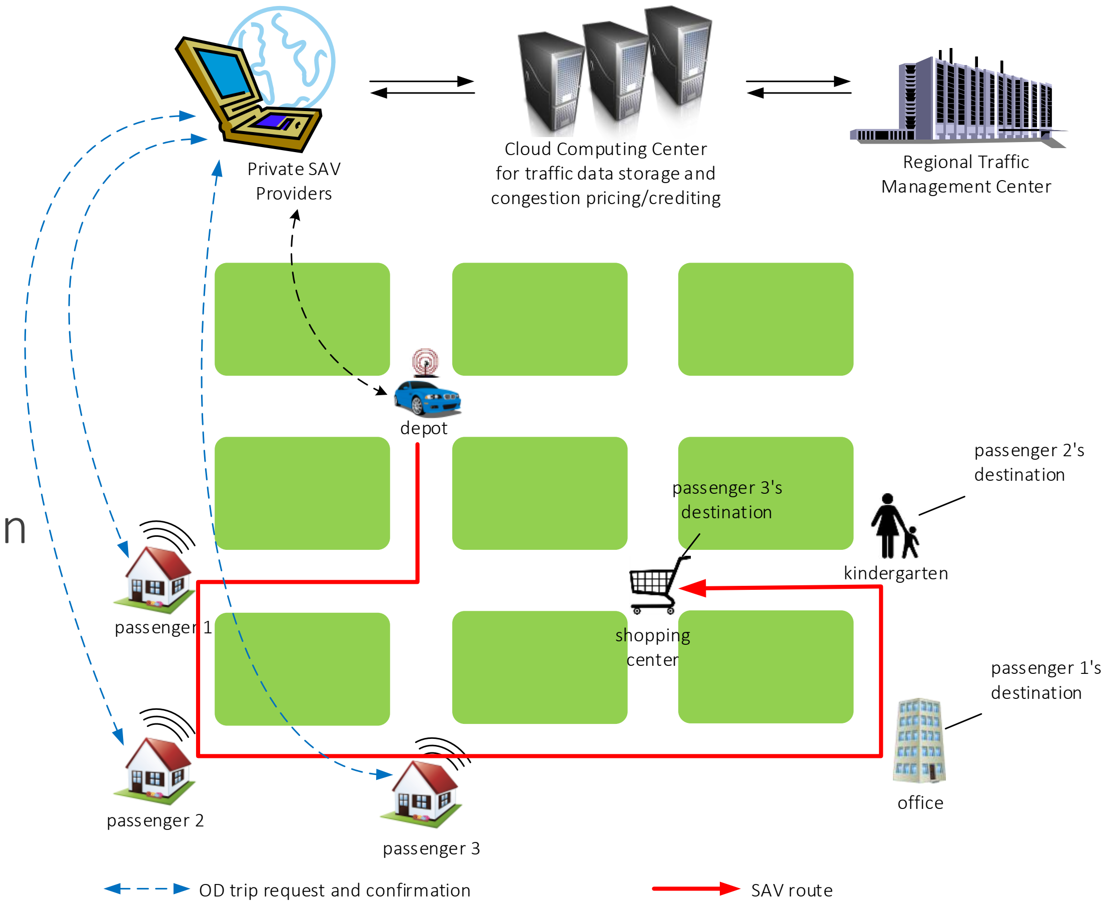
Based on Paper titled “Finding Optimal Solutions for Vehicle Routing Problem with Pickup and Delivery Services with Time Windows: A Dynamic Programming Approach Based on State-space-time Network Representations”
Monirehalsadat Mahmoudi, Xuesong Zhou
Submitted Transportation Research Part B; http://arxiv.org/abs/1507.02731

Motivation

Concept of Ride Sharing:

Advantages of Ride Sharing:

- Reducing driver stress and driving cost
- Increasing safety
- Increasing road capacity and reducing costs
- Increasing fuel efficiency and reducing pollution



Ride Sharing Companies

Side·car
a whole new way to get around

Flywheel

UBER

lyft

summon

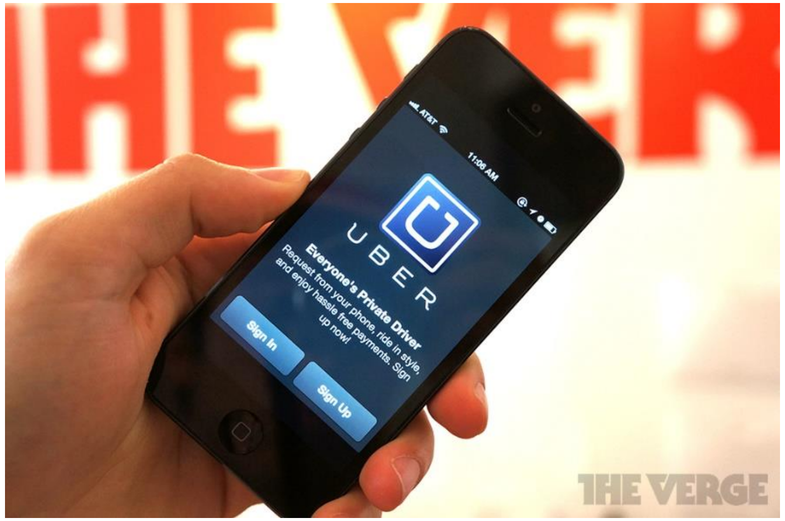
curb

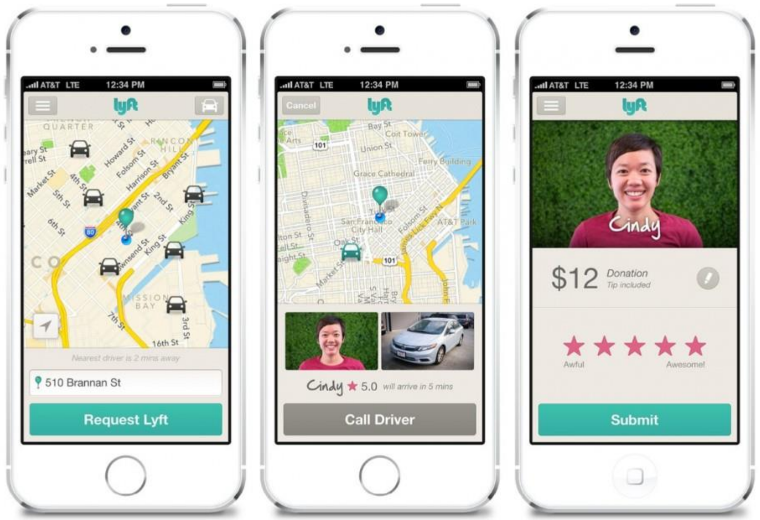
RUBY Ride

HAIL O ™

wingz

Ridesharing Apps





One-to-One Matching

Innovative Pricing Mechanism

Accessibility for Low-income Families

Key Questions

– *How many cars a city should use to support the overall transportation activity demand, at different levels of coordination and pre-trip scheduling?*

– *How much energy* is used at the optimal state (optimal state is a condition in which 100% of travelling is supported by ride sharing)?

– *How much emissions* can be minimized?

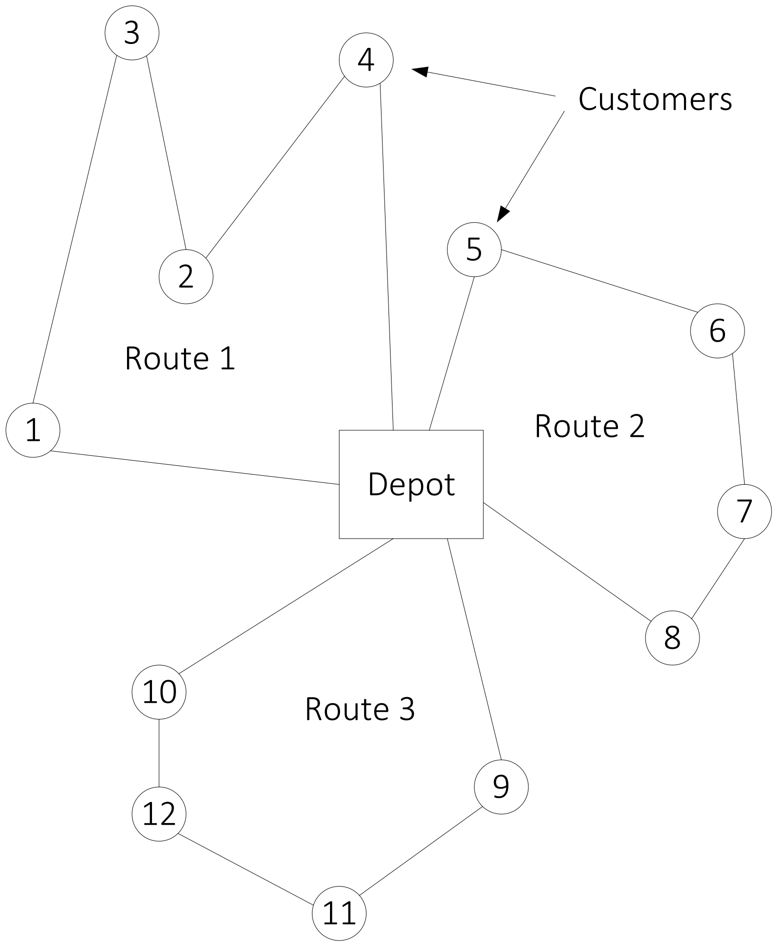
To address the first question:

we propose a new mathematical model for ***pickup and delivery problem with time windows (PDPTW)*** to present a holistic optimization approach for synchronizing travel activity schedules, transportation services, and infrastructure on urban networks.

Vehicle Routing Problem (VRP) & Vehicle Routing Problem with Time Windows (VRPTW)

Inputs:
- ❑ Passengers' Location
- ❑ Passengers' Preferred Service Time Window

Output: Routing



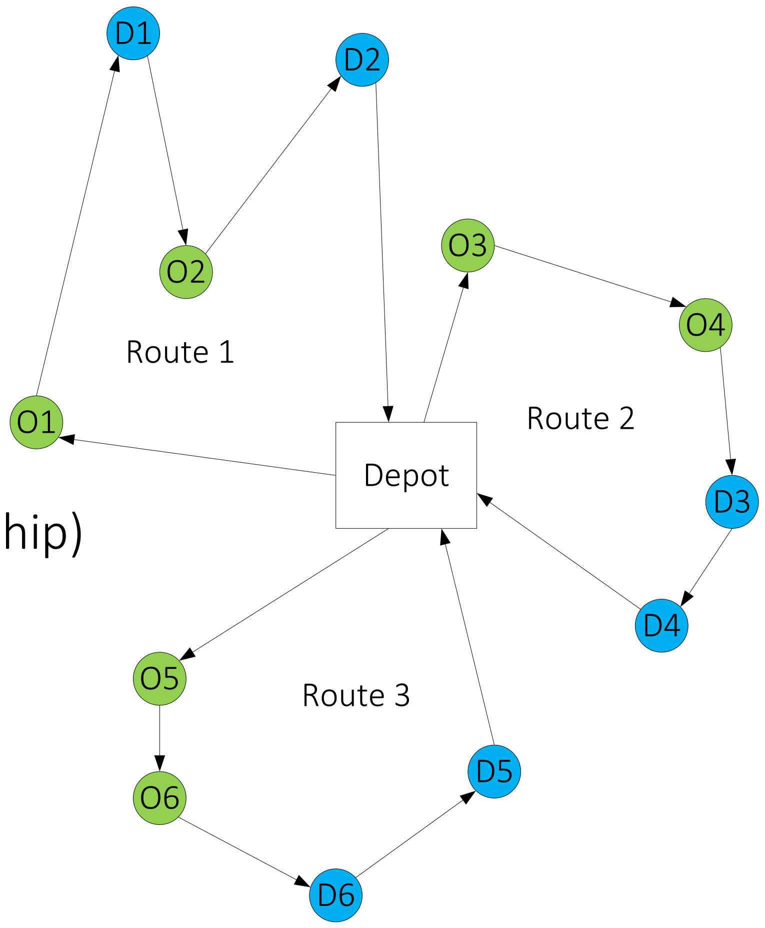
Vehicle Routing Problem with Pick up and Delivery with Time Windows (VRPPDTW)

Inputs:

- ❑ Passengers’ Origin and Destination Location
- ❑ Passengers’ Preferred Pick up and Delivery Time Windows

Outputs:

- ❑ Vehicle Routing
- ❑ Vehicle Scheduling
- ❑ Assigning Passengers to Vehicles (Many-to- One Relationship)
- ❑ Pricing



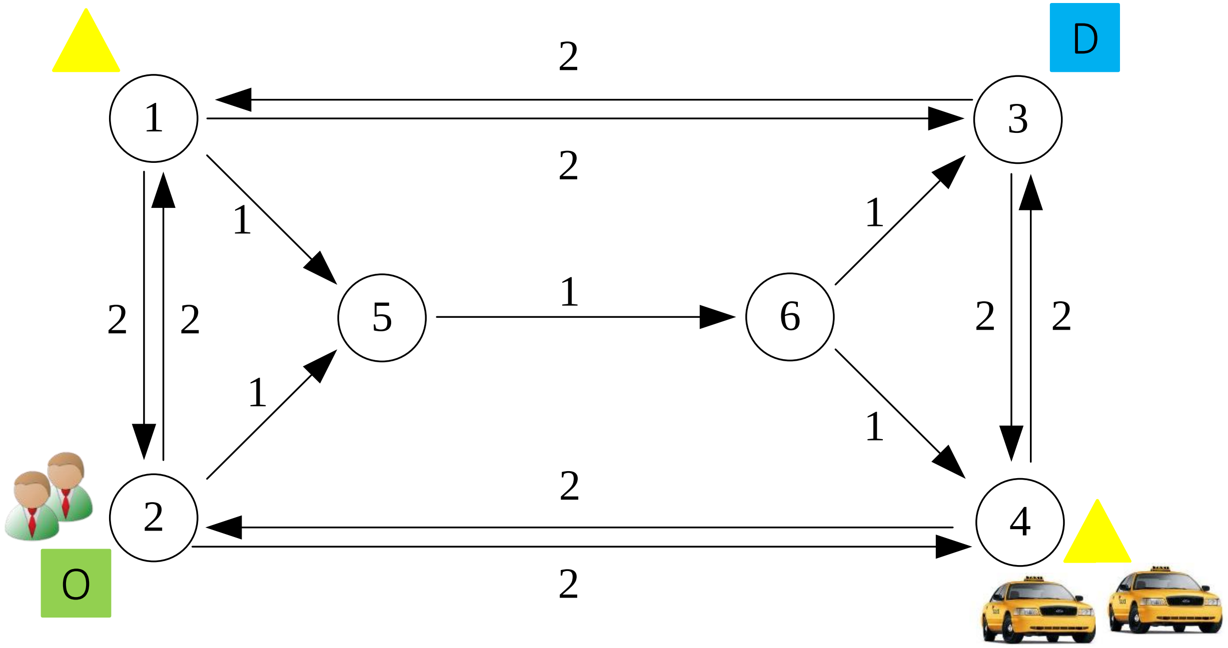
Optimization-based Approaches for Solving VRPPDTW

Reference	Method (Algorithm)	Type of problem	Objective Function
Psaraftis (1980)	Exact backward dynamic programming	Single vehicle VRPPD	Weighted combination of the total service time and the total customer inconvenience
Psaraftis (1983)	Forward dynamic programming	Single vehicle VRPPDTW	Sum of waiting and riding times
Sexton and Bodin (1985)	Benders' decomposition	Single vehicle VRPPD with one sided windows	Total customer inconvenience
Desrosiers, Dumas, and Soumis (1986)	Exact forward dynamic programming	single vehicle VRPPDTW	Total distance traveled
Dumas, Desrosiers, and Soumis (1991)	Column Generation	Multiple vehicle VRPPDTW	Total travel cost

Optimization-based Approaches for Solving VRPPDTW (contd.)

Ruland (1995, 1997)	Polyhedral approach	Single Vehicle VRPPD without capacity constraints	Total travel cost
Savelsbergh and Sol (1998)	Branch-and-price	Multiple vehicle VRPPDTW	Primary objective function: Total number of vehicles, secondary: Total distance traveled
Lu and Dessouky (2004)	Branch-and-cut	Multiple vehicle VRPPDTW	Total travel cost and the fixed vehicle cost
Ropke, Cordeau, Laporte (2007)	Branch-and-cut-and-price	Multiple vehicle VRPPDTW	Total traveled distance
Ropke and Cordeau (2009)	Branch-and-cut-and-price	Multiple vehicle VRPPDTW	Total traveled distance
Baldacci, Bartolini, Mingozzi (2011)	Set-partitioning formulation improved by additional cuts	Multiple vehicle VRPPDTW	Primary objective function: Route costs, secondary: Sum of vehicle fixed costs and then sum of route costs

Problem Statement by a Simple 6-node Transportation Network



Two passengers

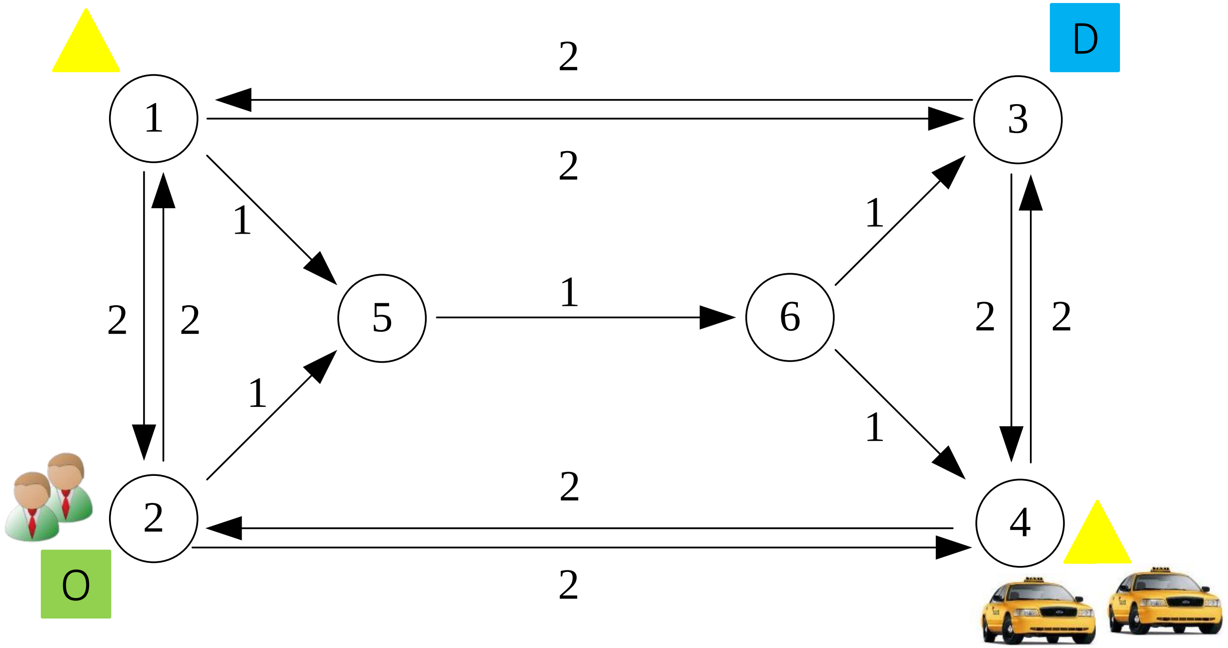
Two vehicles

Passenger 1 and 2's origin: node 2
Passenger 1 and 2's destination: node 3

Vehicle 1 and 2's origin depot: node 4
Vehicle 1 and 2's destination depot: node 1

O Origin
D Destination
Depot

Problem Statement by a Simple 6-node Transportation Network



Passenger 1's preferred time window for departure from origin: [4,7]
Passenger 1's preferred time for arrival at destination: [9,12]

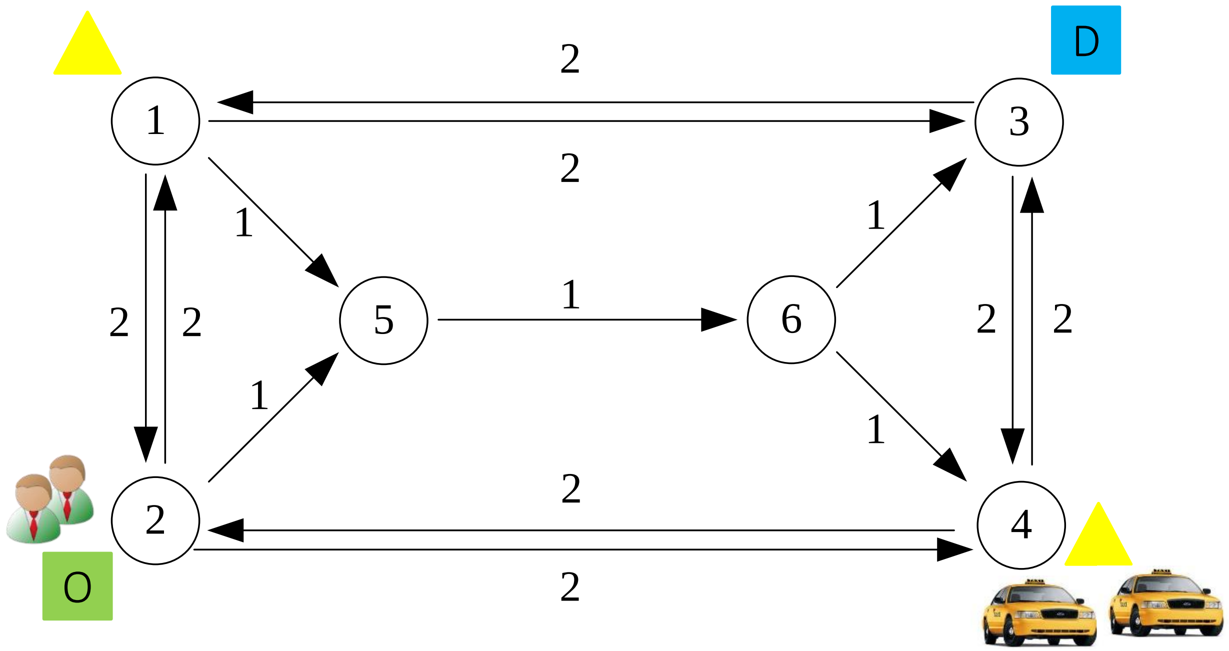
Passenger 2's preferred time window for departure from origin: [10,12]
Passenger 2's preferred time for arrival at destination: [15,17]

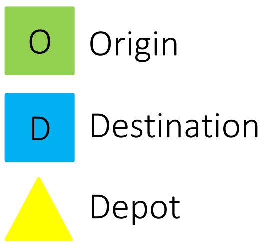
O Origin

D Destination

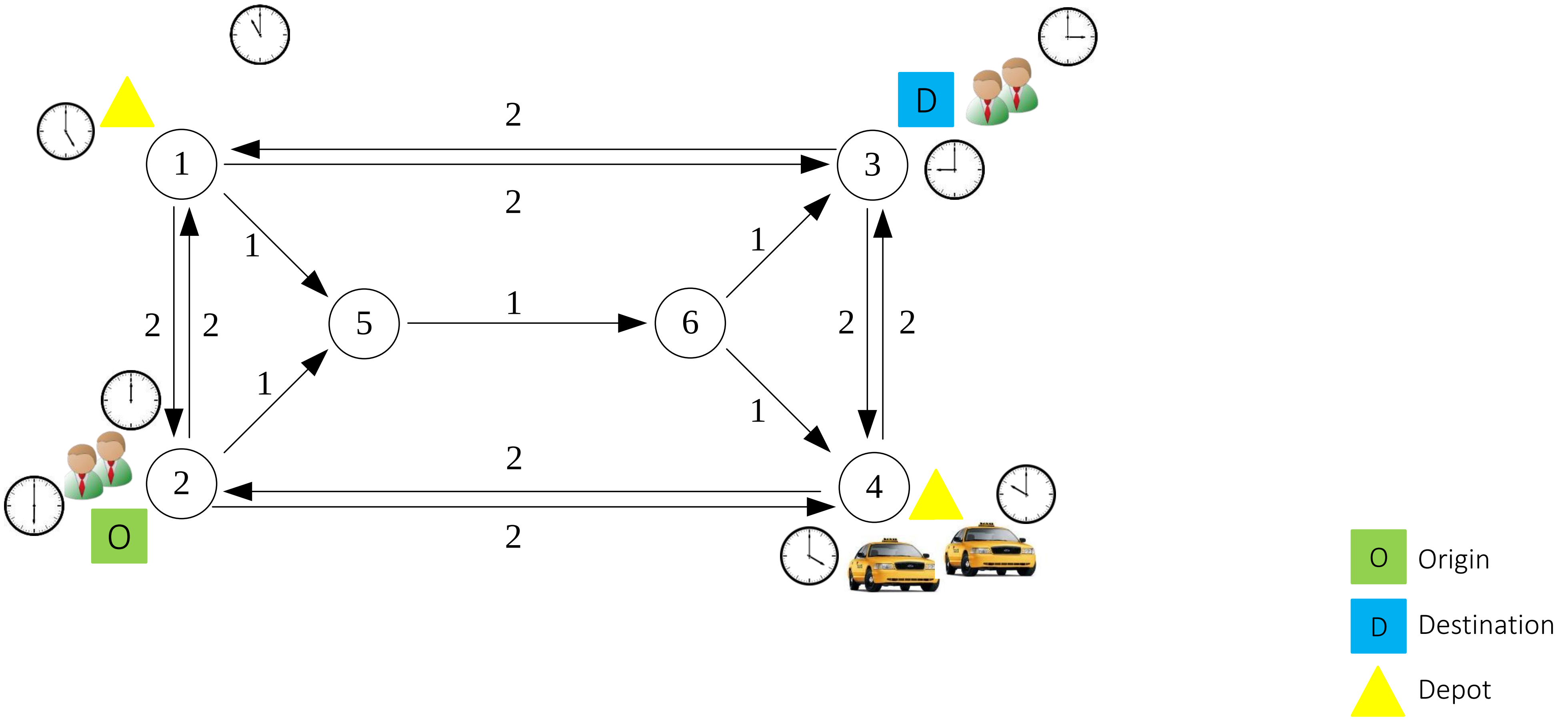
Depot

Problem Statement by a Simple 6-node Transportation Network



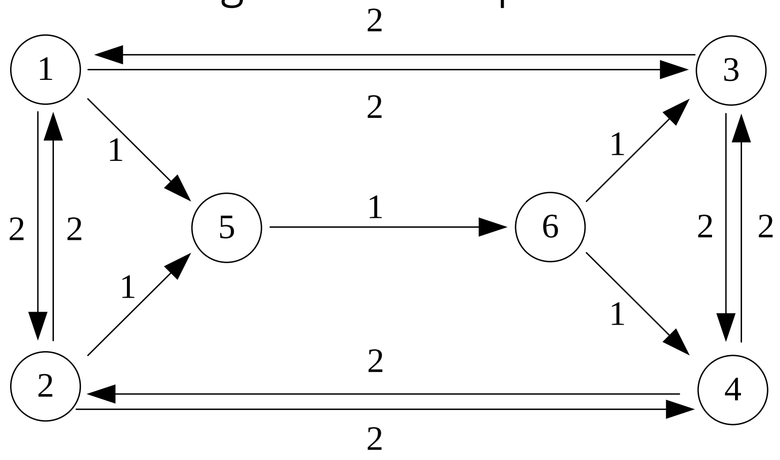


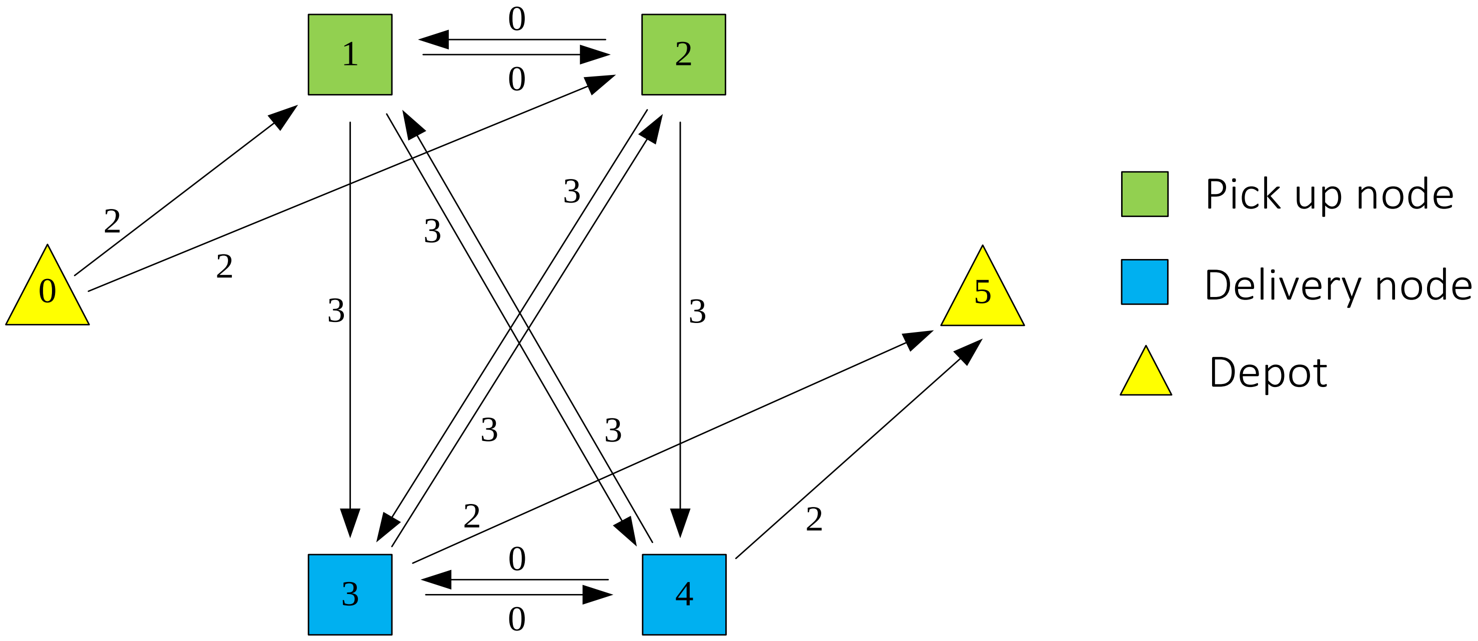
Problem Statement by a Simple 6-node Transportation Network



Opening Statement about Our Method

Existing Network representation [Cordeau (2006)]





Current Mathematical Model for VRPPDTW[Cordeau (2006)]

$$Min \sum_{v \in V} \sum_{i \in N} \sum_{j \in N} c_{ij}^{v} x_{ij}^{v}$$

objective function: minimizing the total routing cost

$$\sum_{v \in V} \sum_{j \in N} x_{ij}^{v} = 1 \qquad \forall i \in P$$

guarantees that each passenger is definitely picked up

$$\sum_{j \in N} x_{ij}^{v} - \sum_{j \in N} x_{n+i,j}^{v} = 0 \qquad \forall i \in P, v \in V$$

ensure that each passenger's origin and destination are visited exactly once by the same vehicle

$$\sum_{j \in N} x_{0j}^{v} = 1 \qquad \forall v \in V$$

each vehicle v starts its route from the origin depot

$$\sum_{j \in N} x_{ji}^{v} - \sum_{j \in N} x_{ij}^{v} = 0 \qquad \forall i \in P \cup D, v \in V$$

flow balance on each node

$$\sum_{i \in N} x_{i,2n+1}^{v} = 1 \qquad \forall v \in V$$

each vehicle v ends its route to the destination depot

Current Mathematical Model for VRPPDTW[Cordeau (2006)] (contd.)

$x_{ij}^{v}(B_i^v + d_i + t_{ij}) \leq B_j^v$	Non-linear Constraint	$\forall i \in N, j \in N, v \in V$	validity of the time variables
$x_{ij}^{v}(Q_i^v + q_j) \leq Q_j^v$	Non-linear Constraint	$\forall i \in N, j \in N, v \in V$	validity of the load variables
$L_i^v = B_{n+i}^v - (B_i^v + d_i)$		$\forall i \in P, v \in V$	defines each passenger's ride time
$B_{2n+1}^v - B_0^v \leq T_v$		$\forall v \in V$	impose maximal duration of each route
$e_i \leq B_i^v \leq l_i$		$\forall i \in N, v \in V$	impose time windows constraints
$t_{i,n+i} \leq L_i^v \leq L$		$\forall i \in P, v \in V$	impose the ride time of each passenger constraints
$max\{0, q_i\} \leq Q_i^v \leq min\{Q_v, Q_v + q_i\}$		$\forall i \in N, v \in V$	impose capacity constraints
$x_{ij}^v \in \{0,1\}$		$\forall i \in N, j \in N, v \in V$	

Current Theoretical and Computational Challenges

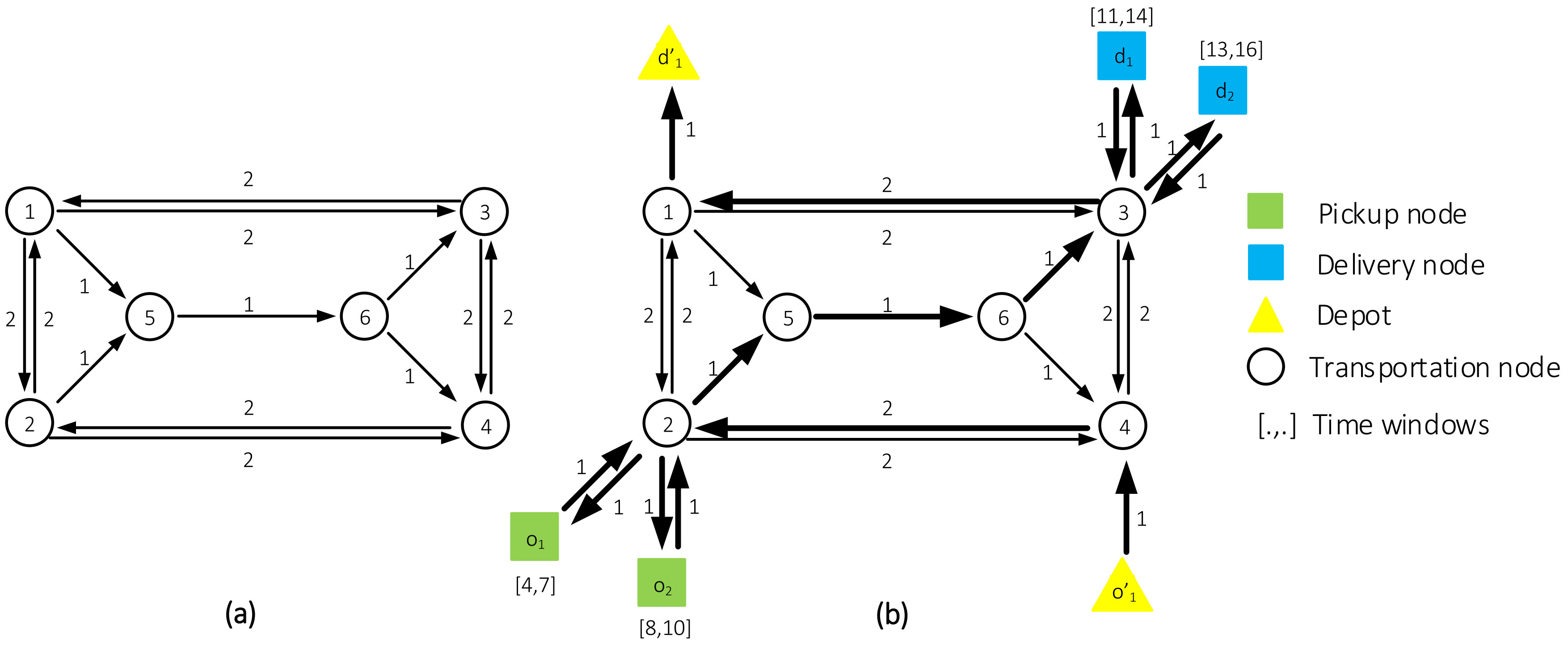
- Single vs multiple vehicles
  - The focus of most research was on solving the PDPTW for a single vehicle (simpler case)
- Single vs multiple depots
- Limited number of transportation requests (passengers)
  - The most Current Algorithm: Baldacci et al. (2011) based on a set-partitioning formulation solved instances of approximately 500 requests with tight time windows.
- Time windows
  - Some preprocessing steps to find feasible transportation requests are needed
  - Research only focus on tight time windows to prevent some fluctuation in results

Current Theoretical and Computational Challenges (contd.)

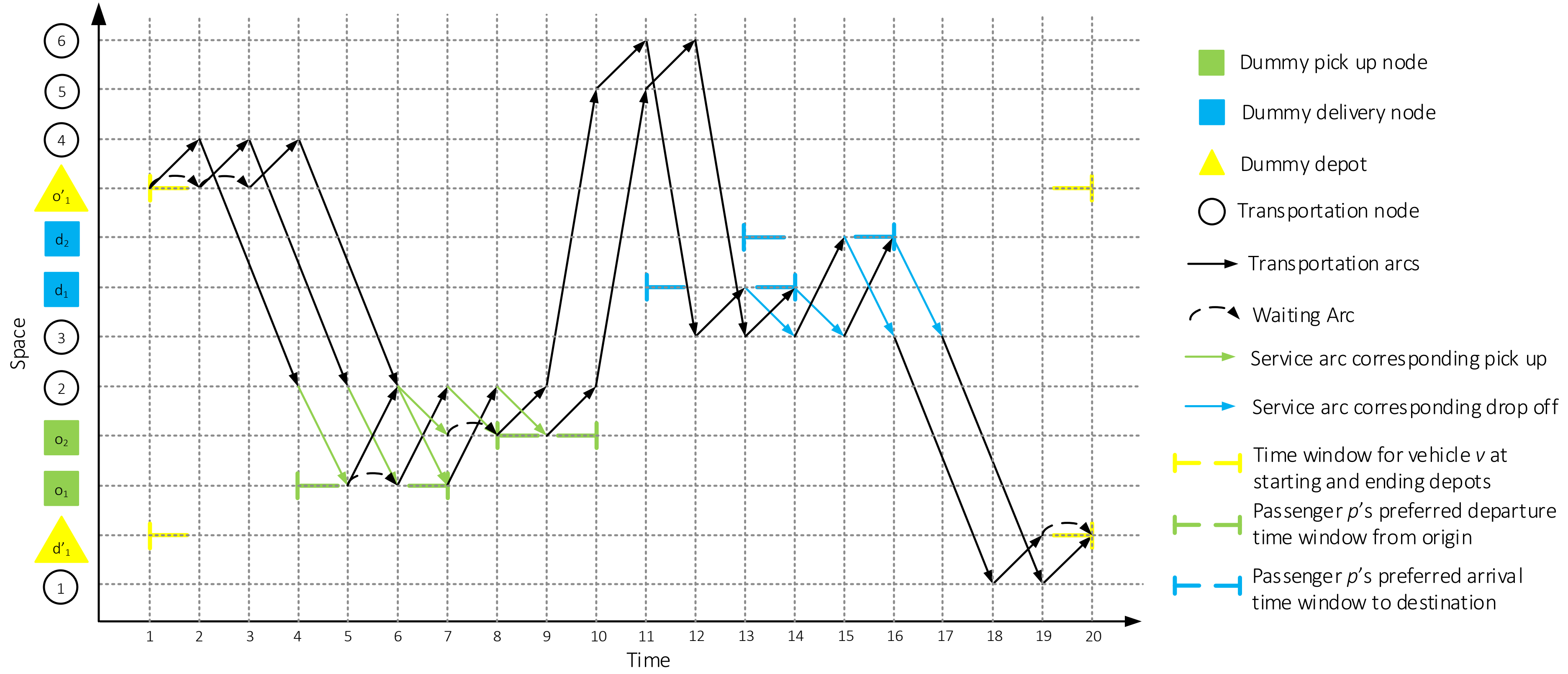
- Fixed routing cost (travel time) over time
  - Existing network for PDPTW: an offline network in which each link has a fixed routing cost (travel time)
- Existence of sub-tour in the optimal solution
  - Some additional constraints are needed to avoid the existence of any sub-tour in the optimal solution

Opening Statement about Our Method (contd.)

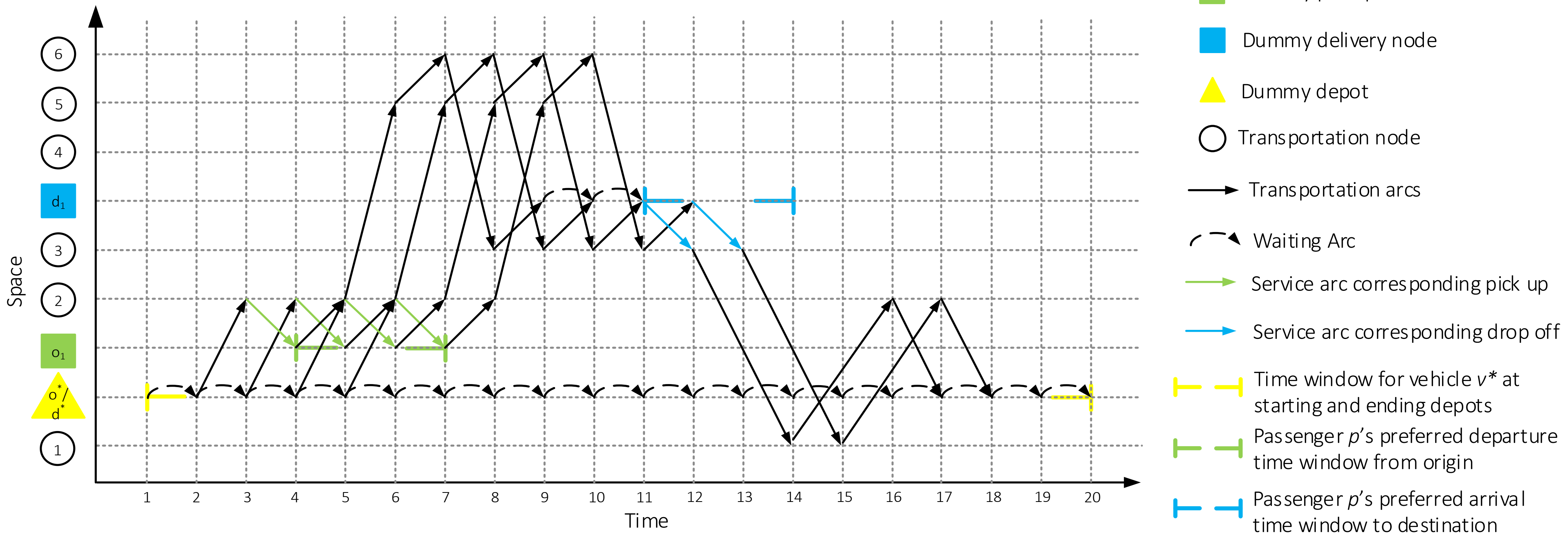
Description of the PDPTW in Space-Time Transportation Network [Mahmoudi, M. & Zhou, X. (2014)]



Physical Vehicle's Space-time Transportation Network

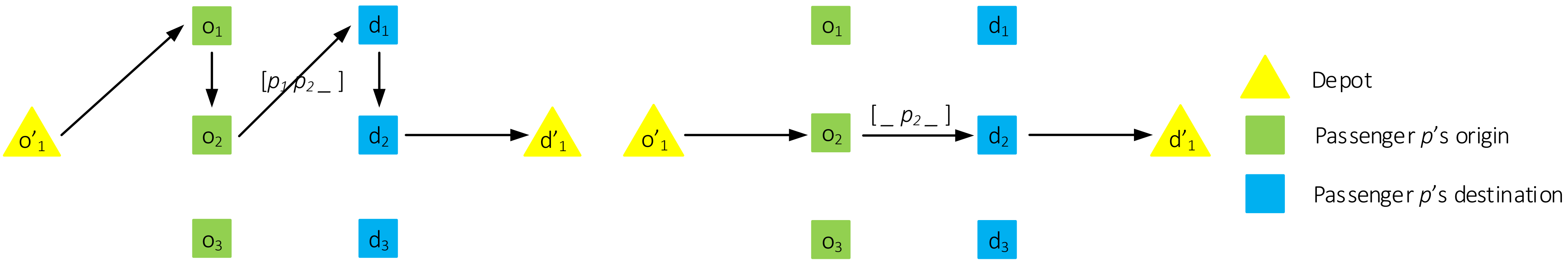


Virtual Vehicle's Space-time Transportation Network



Binary representation and equivalent character-based representation for passenger carrying states

Binary representation	Equivalent character-based representation
[0,0,0]	[_ _ _]
[1,0,0]	[p_1 _ _]
[0,1,1]	[_ p_2 p_3]



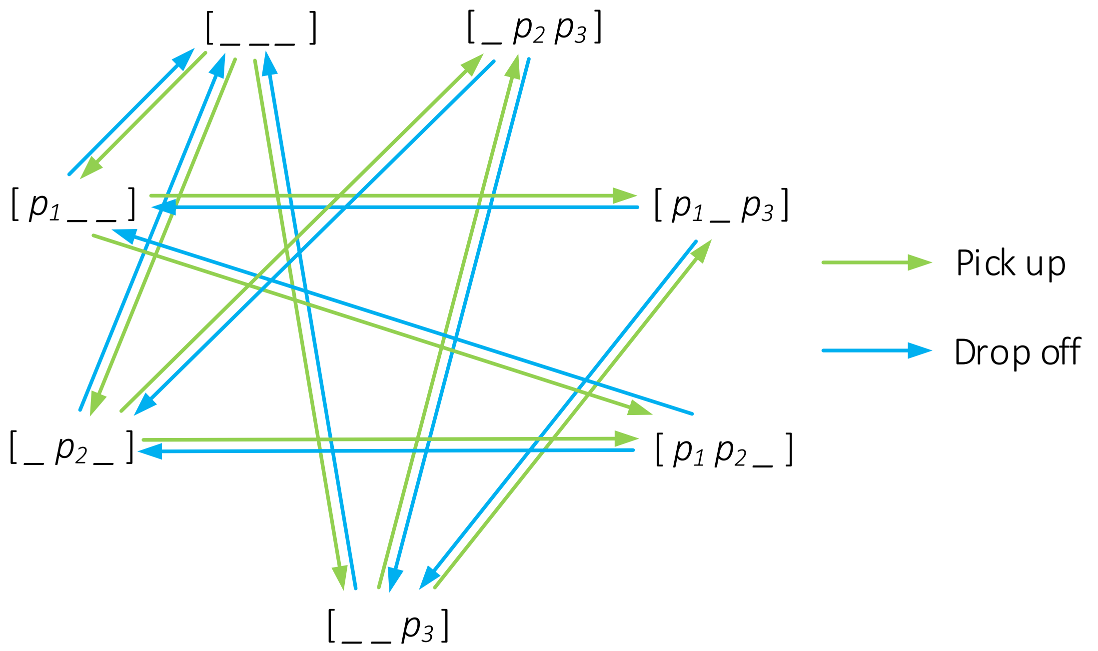
(a) Shared ride

(b) Serving single passenger once a time

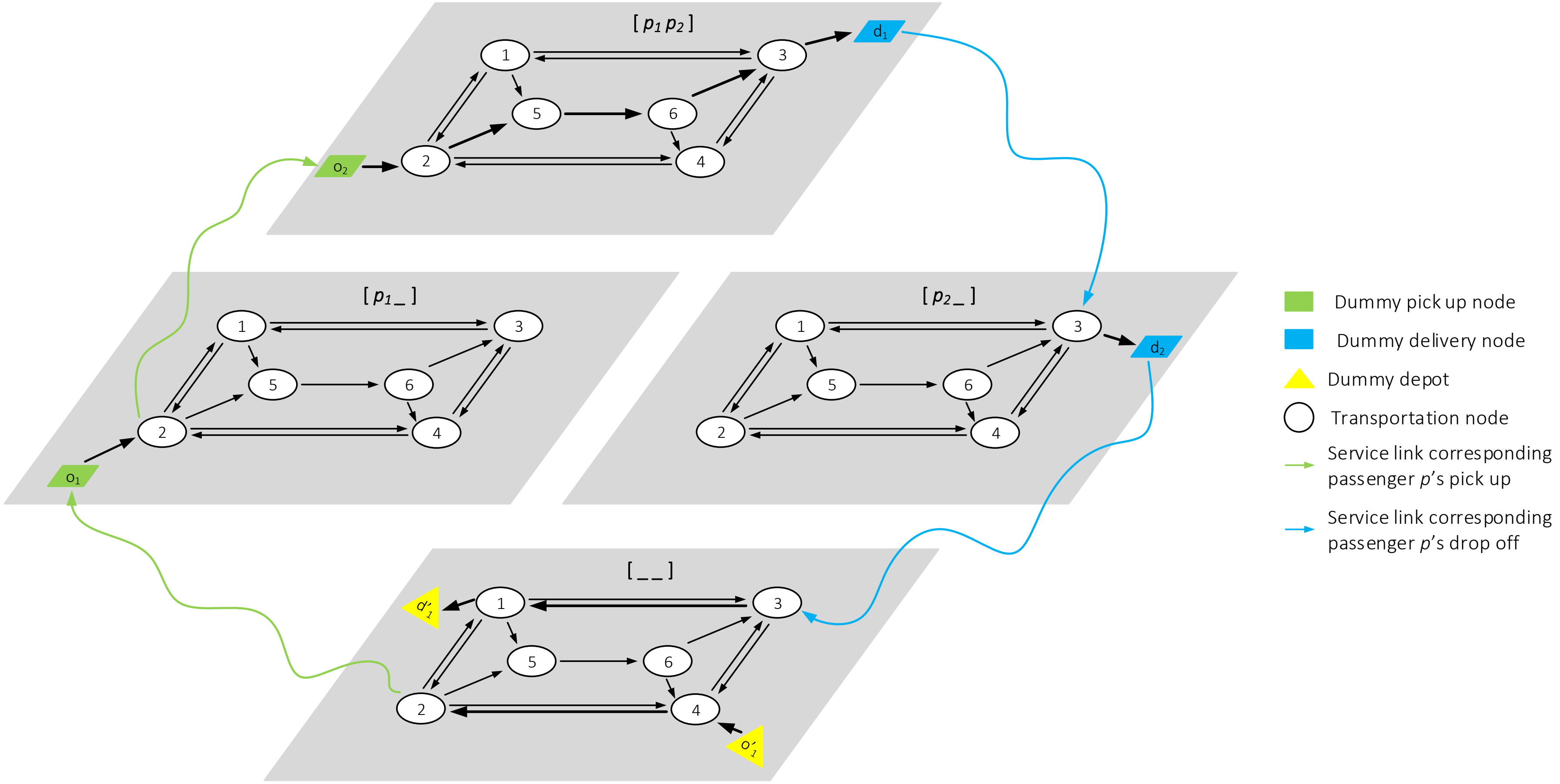
All possible combinations of passenger carrying states

w \ w'	$[\_\ \_\ \_]$	$[p_1\ \_\ \_]$	$[\_\ p_2\ \_]$	$[\_\ \_\ p_3]$	$[p_1\ p_2\ \_]$	$[p_1\ \_\ p_3]$	$[\_\ p_2\ p_3]$
$[\_\ \_\ \_]$	no change	pickup	pickup	pickup			
$[p_1\ \_\ \_]$	drop-off	no change			pickup	pickup	
$[\_\ p_2\ \_]$	drop-off		no change		pickup		pickup
$[\_\ \_\ p_3]$	drop-off			no change		pickup	pickup
$[p_1\ p_2\ \_]$		drop-off	drop-off		no change		
$[p_1\ \_\ p_3]$		drop-off		drop-off		no change	
$[\_\ p_2\ p_3]$			drop-off	drop-off			no change

Finite states graph showing all possible passenger carrying state transition (pickup or drop-off)



Projection on state-space network representation for ride-sharing path (pick up passenger p_1 and then p_2).



Problem Definition

$$Min\ Z = \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in B_v} c(v,i,j,t,s,w,w') y(v,i,j,t,s,w,w')$$

s.t.

Flow balance constraints at vehicle v's origin vertex

$$\sum_{(i,j,t,s,w,w') \in B_v} y(v,i,j,t,s,w,w') = 1 \qquad i = o'_v, t = e_v, w = w' = w_0, \forall v \in (V \cup V^*)$$

Flow balance constraint at vehicle v's destination vertex

$$\sum_{(i,j,t,s,w,w') \in B_v} y(v,i,j,t,s,w,w') = 1 \qquad j = d'_v, s = l_v, w = w' = w_0, \forall v \in (V \cup V^*)$$

Flow balance constraint at intermediate vertex

$$\sum_{(j,s,w'')} y(v,i,j,t,s,w,w'') - \sum_{(j',s',w')} y(v,j',i,s',t,w',w) = 0 \quad (i,t,w) \notin \{(o'_v, e_v, w_0), (d'_v, l_v, w_0)\}, \forall v \in (V \cup V^*)$$

Problem Definition (contd.)

Passenger p's pick-up request constraint

$$\sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Psi_{p,v}} y(v,i,j,t,s,w,w') = 1 \qquad \forall p \in P$$

Passenger p's drop-off request constraint

$$\sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Phi_{p,v}} y(v,i,j,t,s,w,w') = 1 \qquad \forall p \in P$$

Binary definitional constraint

$$y(v,i,j,t,s,w,w') \in \{0,1\} \qquad \forall (i,j,t,s,w,w') \in B_v, \forall v \in (V \cup V^*)$$

Lagrangian Relaxation-based Solution Approach

- Pickup and drop-off constraints : each passenger is picked up and dropped off exactly once by a vehicle (either physical or virtual).
- Flow balance constraints on intermediate nodes force the vehicle to end its route at the destination depot with the empty passenger carrying state.
- If vehicle v picks up passenger p from his origin, to maintain the flow balance constraints on intermediate nodes , the vehicle must drop-off the passenger at his destination node so that the vehicle comes back to its ending depot with the empty passenger carrying state.
- As a result, constraint (6) is redundant

$$L = \sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in B_v} c(v,i,j,t,s,w,w') y(v,i,j,t,s,w,w') + \sum_{p \in P} \lambda(p) \left[\sum_{v \in (V \cup V^*)} \sum_{(i,j,t,s,w,w') \in \Psi_{p,v}} y(v,i,j,t,s,w,w') - 1 \right]$$

New Relaxed Problem

$Min\ L$

s.t.

$\sum_{(i,j,t,s,w,w')\in B_v} y(v,i,j,t,s,w,w') = 1 \qquad i = o'_v, t = e_v, w = w' = w_0, \forall v \in (V \cup V^*)$

$\sum_{(i,j,t,s,w,w')\in B_v} y(v,i,j,t,s,w,w') = 1 \qquad j = d'_v, s = l_v, w = w' = w_0, \forall v \in (V \cup V^*)$

$\sum_{(j,s,w'')} y(v,i,j,t,s,w,w'') -$

$\sum_{(j',s',w')} y(v,j',i,s',t,w',w) = 0 \quad (i,t,w) \notin \{(o'_v,e_v,w_0),(d'_v,l_v,w_0)\}, \forall v \in (V \cup V^*)$

$y(v,i,j,t,s,w,w') \in \{0,1\} \qquad \forall (i,j,t,s,w,w') \in B_v, \forall v \in (V \cup V^*)$

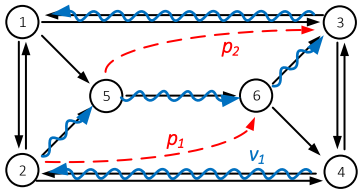
The simplified Lagrangian function *L:*

$$L = \sum_{v\in(V\cup V^*)} \sum_{(i,j,t,s,w,w')\in B_v} \xi(v,i,j,t,s,w,w')y(v,i,j,t,s,w,w') - \sum_{p\in P} \lambda(p)$$

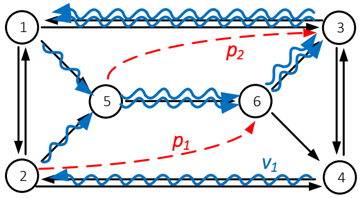
Time-dependent Forward Dynamic Programming

```
for each vehicle v ∈ (V ∪ V*) do
    L(.,.,.) := +∞;
    node pred of vertex (.,.,.) := −1; time pred of vertex (.,.,.) := −1; state pred of vertex (.,.,.) := −1;
    L(o'_v, e_v, w_0) := 0;
    for each time t ∈ [e_v, l_v] do
        for each link (i, j) do
            for each state w do
            derive downstream state w' based on the possible state transition on link (i, j);
            derive arrival time s = t + TT(i, j, t);
            if (L(i, t, w) + ξ(v, i, j, t, s, w, w') < L(j, s, w')) then
                L(j, s, w') := L(i, t, w) + ξ(v, i, j, t, s, w, w') ; //label update
                node pred of vertex (j, s, w') := i; time pred of vertex (j, s, w') := t; state pred of vertex (j, s, w') := w;
```

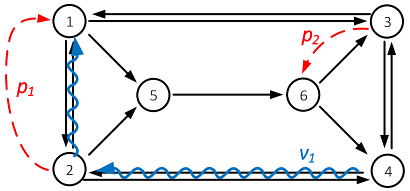
Computational Results for Testing Different Cases



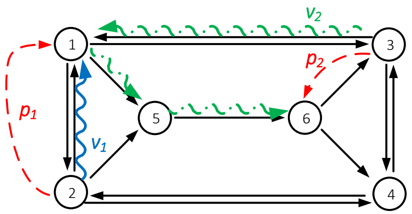
Scenario I. Two passengers are served by one vehicle through ride-sharing mode.



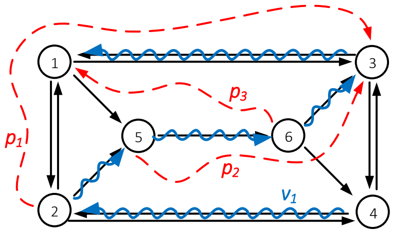
Scenario II. Two passengers are served by one vehicle (not through ride-sharing mode).



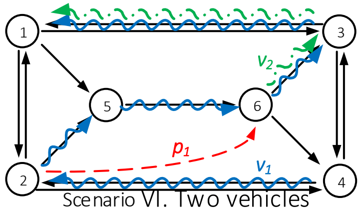
Scenario III. Two passengers and one vehicle; one passenger remains unserved.



Scenario IV. Two passengers and two vehicles; each vehicle is assigned to a passenger



Scenario V. Three passengers are served by one vehicle through ride-sharing mode



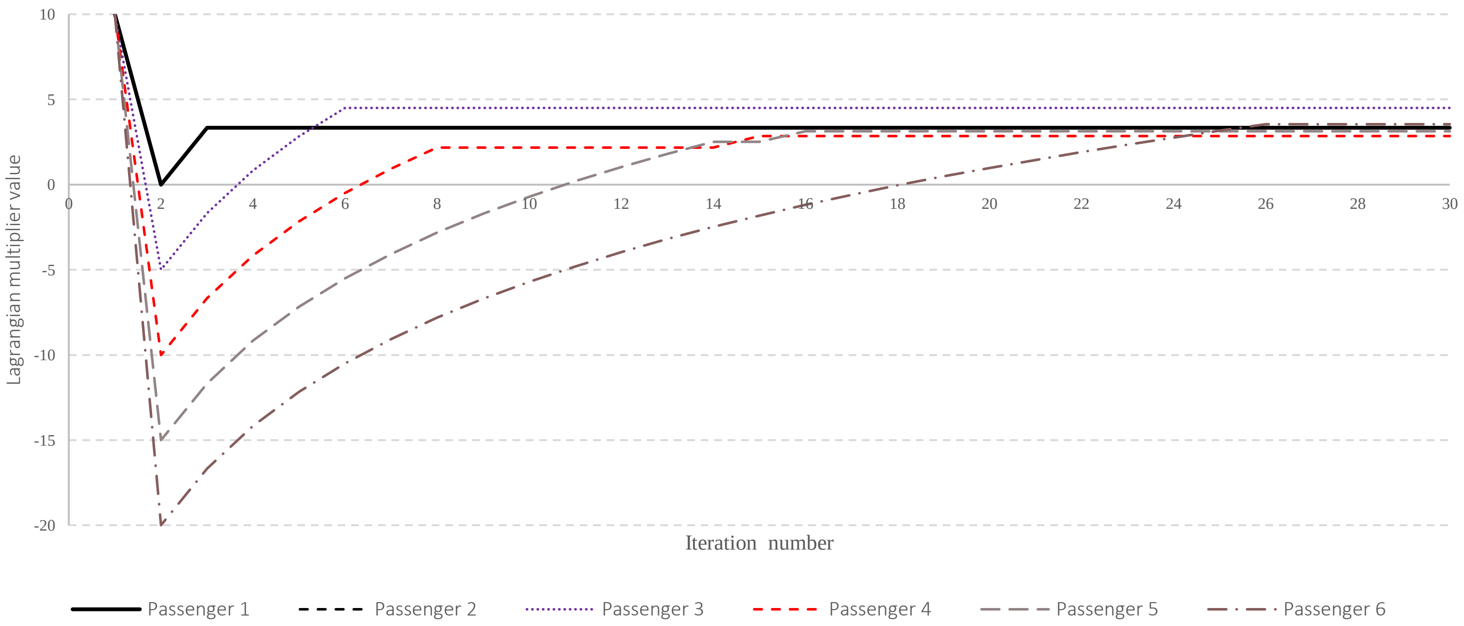
Scenario VI. Two vehicles compete for serving a passenger

Transportation link

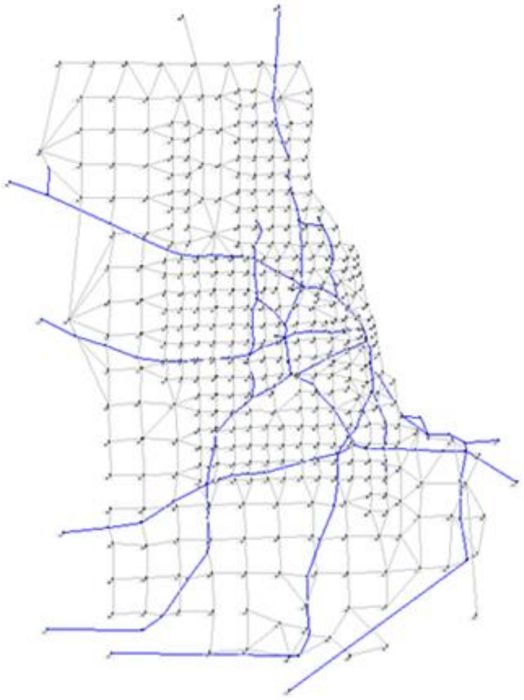
Representative of passenger *p*'s origin-destination pair

Representative of vehicle routing

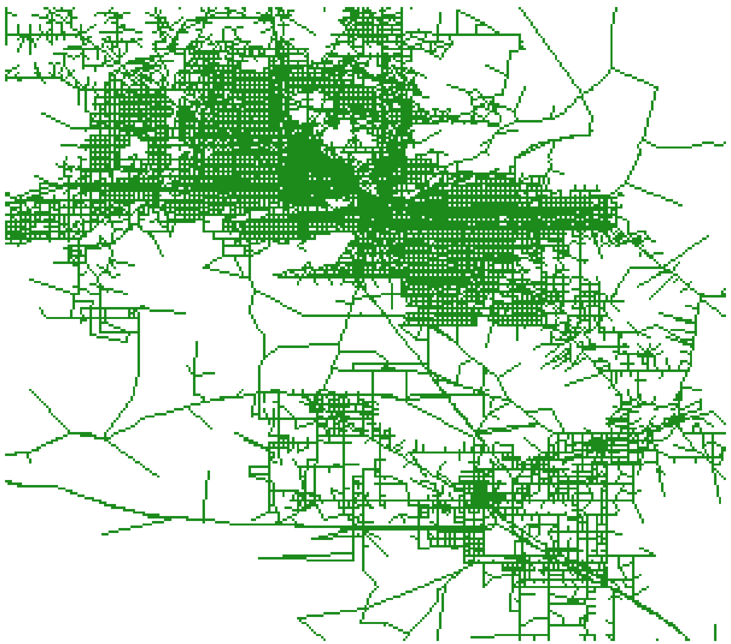
Lagrangian multipliers along 30 iterations in test case 1 for the six-node transportation network: System Marginal Cost!



Medium and large-scale transportation networks for computational performance testing



(a) Chicago sketch network



(b) Phoenix metropolitan regional network

Results for the Chicago network with 933 transportation nodes and 2,967 links

Test case number	Number of iterations	Number of passengers	Number of vehicles	LB^*	UB^*	Gap (%)	Number of passengers not served	CPU running time (sec)
1	20	2	2	108.43	108.43	0.00%	0	17.43
2	20	11	3	352.97	352.97	0.00%	0	91.87
3	20	20	5	616.66	626.18	1.52%	1	327.51
4	20	46	15	1586.81	1664.07	4.64%	2	4681.52
5	20	60	15	1849.98	1878.55	1.52%	3	7096.50

Results for the Phoenix network with 13,777 transportation nodes and 33,879 links

Test case number	Number of iterations	Number of passengers	Number of vehicles	LB^*	UB^*	Gap (%)	Number of passengers not served	CPU running time (sec)
1	6	4	2	70.95	70.95	0.00%	0	110.39
2	6	10	5	191.55	207.05	7.49%	1	398.37
3	6	20	6	310.37	310.37	0.00%	0	1323.18
4	6	40	12	622.23	622.23	0.00%	0	3756.505
5	6	50	15	784.07	784.07	0.00%	0	6983.189

Short Summary

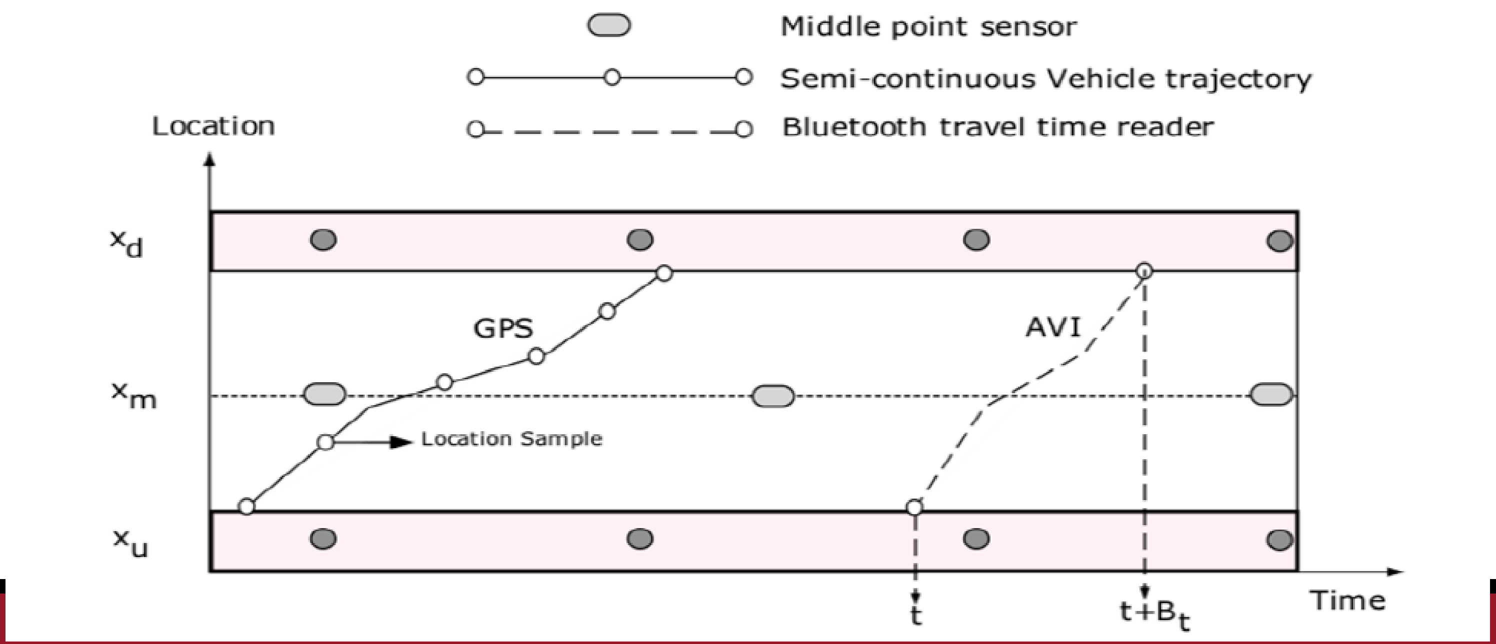
- We propose a new mathematical formulation in state-space-time network for PDPTW
- Based on time-dependent forward dynamic programming approach in the Lagrangian reformulation framework, the main problem is transformed to easy sub-problems (Time-dependent least cost path sub-problems) which is solved independently without much effort
- Unlike former proposed models for PDPTW, this model is now able to solve PDPTW in large scale transportation networks.

Topic 3:
Extensions of state-space-time modeling framework:
Traffic flow state estimation, and traffic signal control and train timetabling…

Application 1: Traffic State Estimation

How much information is sufficient?

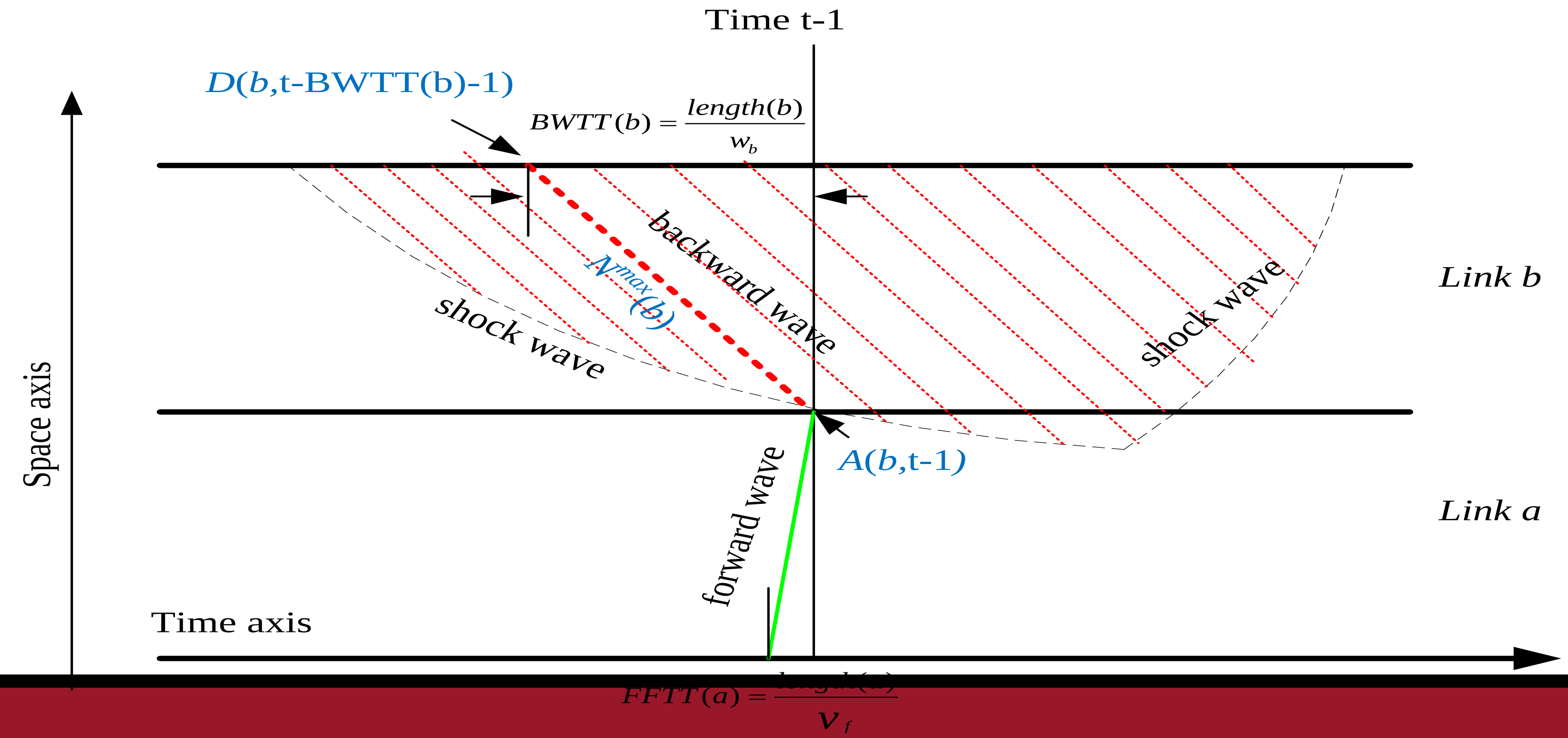
- How to locate point sensors on a traffic segment?
- How to locate Bluetooth reader locations?
- How much AVI/GPS market penetration rate is sufficient?



Space-state-time network: N(x,t): state N as cumulative flow counts

Dr. Newell's three-detector model provides a unified framework

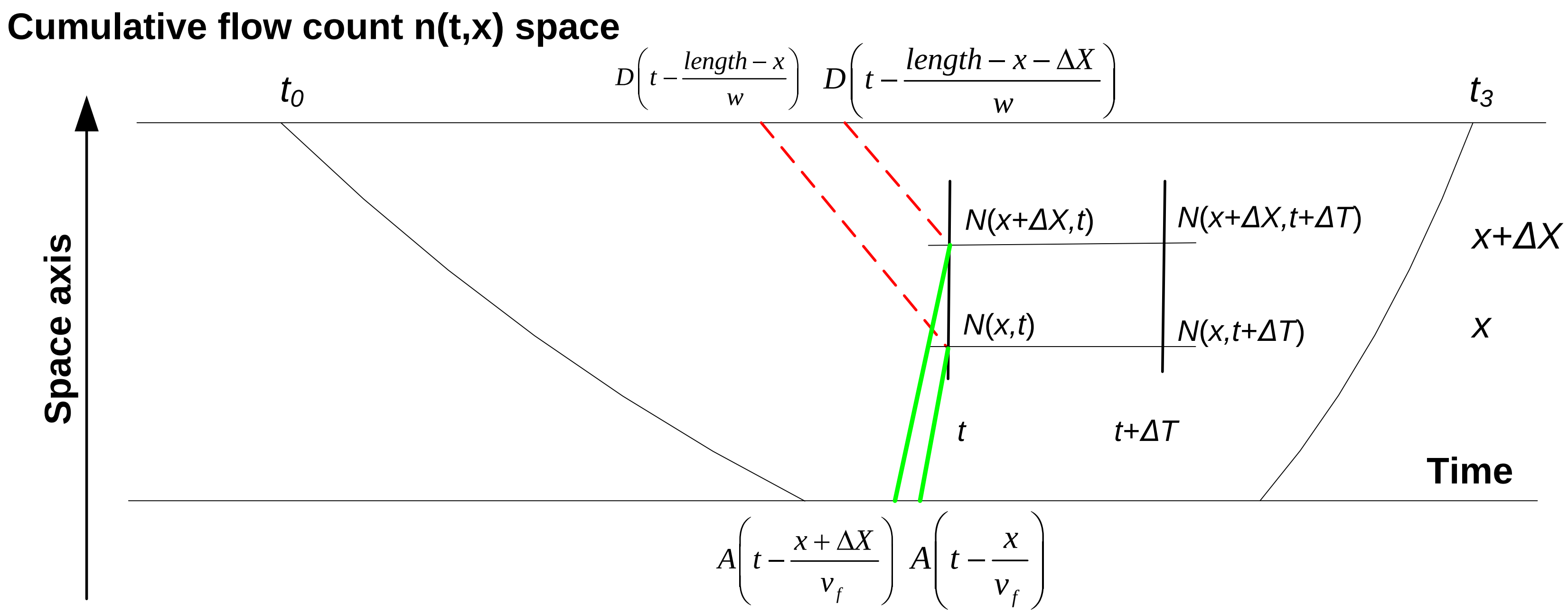
- $N(t,x) = \text{Min}\{N_{upstream}(t-BWTT) + Kjam * distance,\ N_{downstream}(t-FFTT)\}$



1: From Point Sensor Data to Boundary N-curves

Cell density and flow are all functions of cumulative flow counts

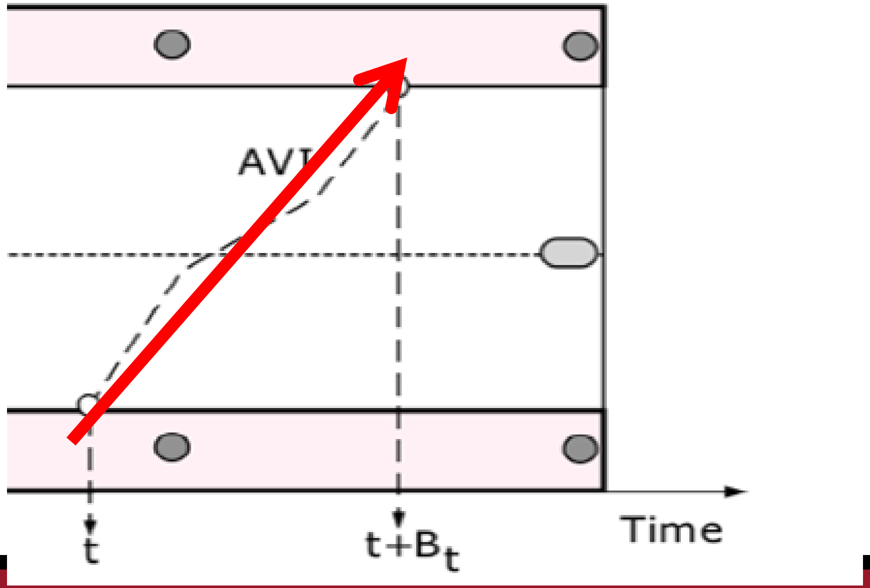
$$k(t,x) = \frac{n(t,x) - n(t,x+\Delta x)}{\Delta x}$$



2: From Bluetooth Travel Time to Boundary N-curves

Downstream and upstream N-Curves between two time stamps are connected

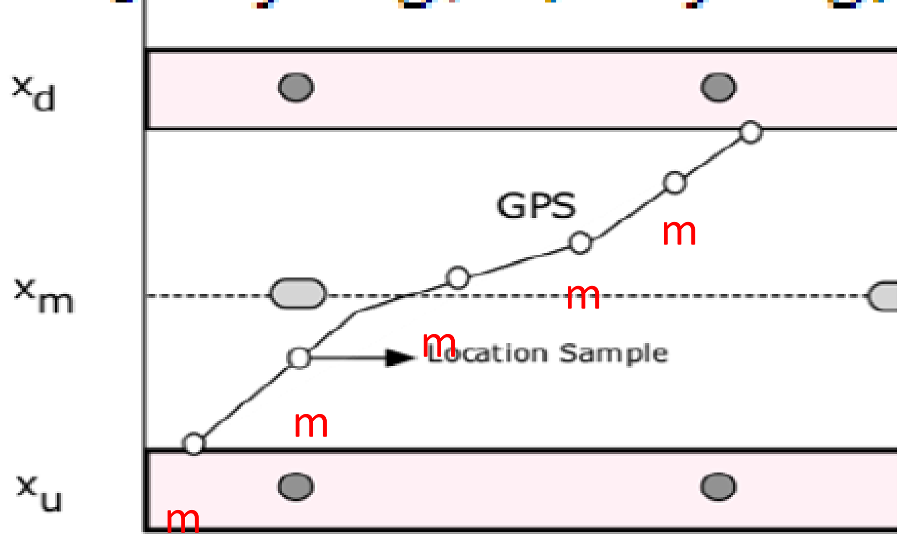
$$n_u(t) = n_d(t + B_t)$$



3: From to GPS Trajectory Data to Boundary N-curves

Under FIFO conditions, GPS probe vehicle keeps the same N-Curve number (say m)

$$m = n[t + j''\Delta g, x(t + j''\Delta g)]$$



Stochastic 3-detector Model

All sensors have errors→ error propagation

$$\text{Min}\ \{N_{upstream}(*)+e_u,\ N_{downstream}(*)+e_d\}$$

Deng, W. Lei H. ,Zhou, X. (2013) Freeway Traffic State Estimation and Uncertainty Quantification based on Heterogeneous Data Sources: A Three Detector Approach. Transportation Research Part B. 57, 132-157

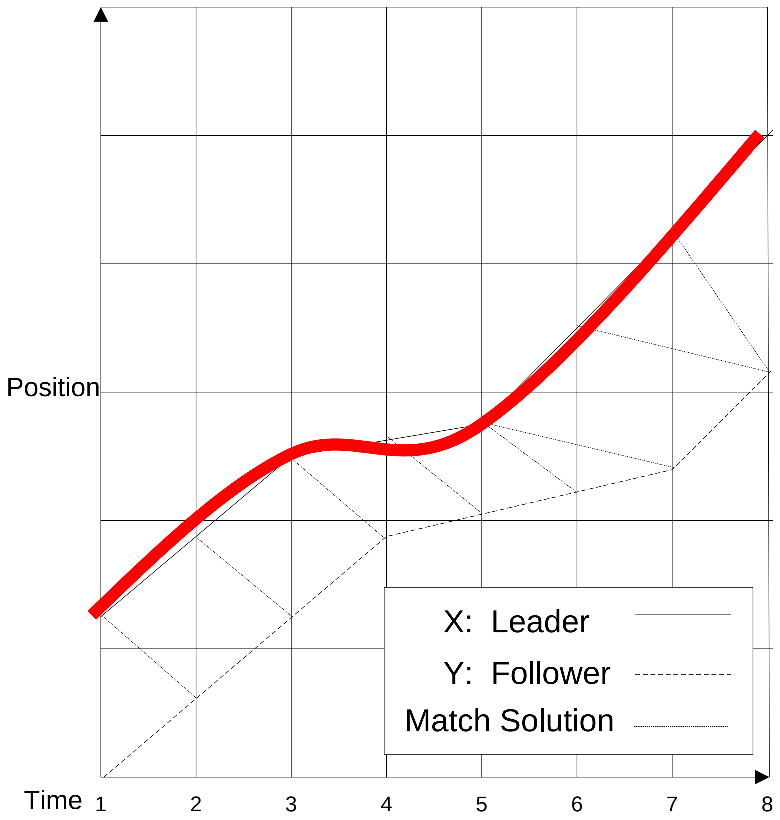
From single segment to corridor

Lei, H., & Zhou, X. (2014). Linear Programming Model for Estimating High-Resolution Freeway Traffic States from Vehicle Identification and Location Data. Transportation Research Record: Journal of the Transportation Research Board, 2421, 151-160.

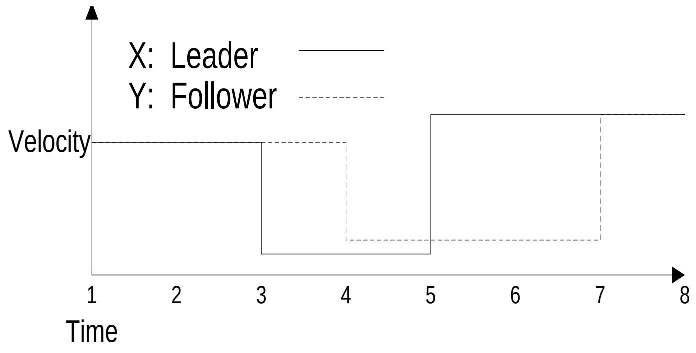
Application 2: State-space-time path → State as speed for trajectory

Use Dynamic Time Warping (DTW) to Estimate dynamic car following model

- Matches points by measure of similarity



(A) Vehicle Trajectories with DTW Match Solution

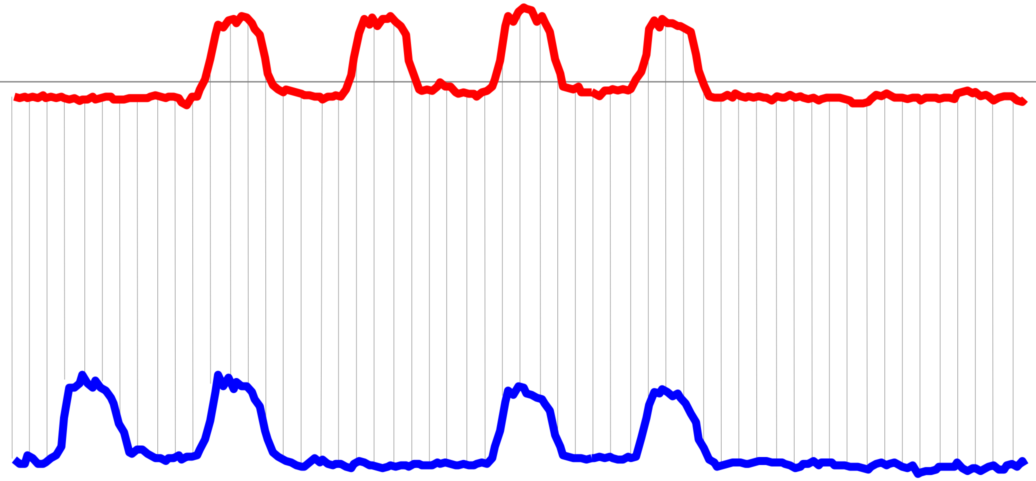


(B) Vehicle Velocity Time Series

Reference: **Eamonn Keogh**

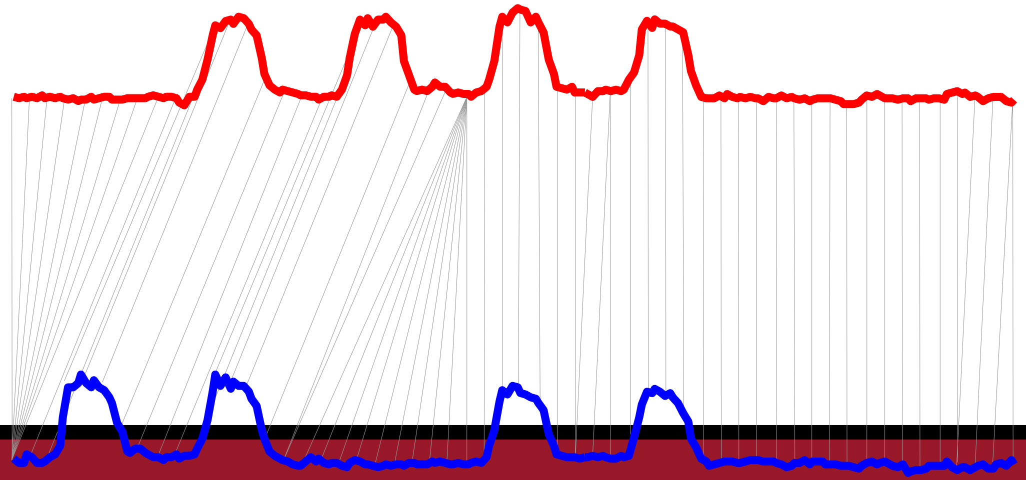
Computer Science & Engineering Department
University of California - Riverside

Euclidean Vs Dynamic Time Warping



Euclidean Distance

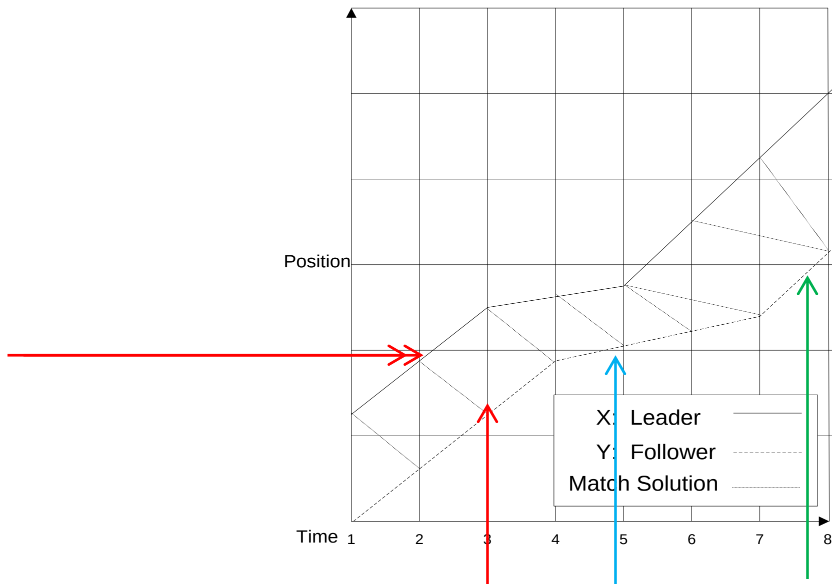
Sequences are aligned "one to one".



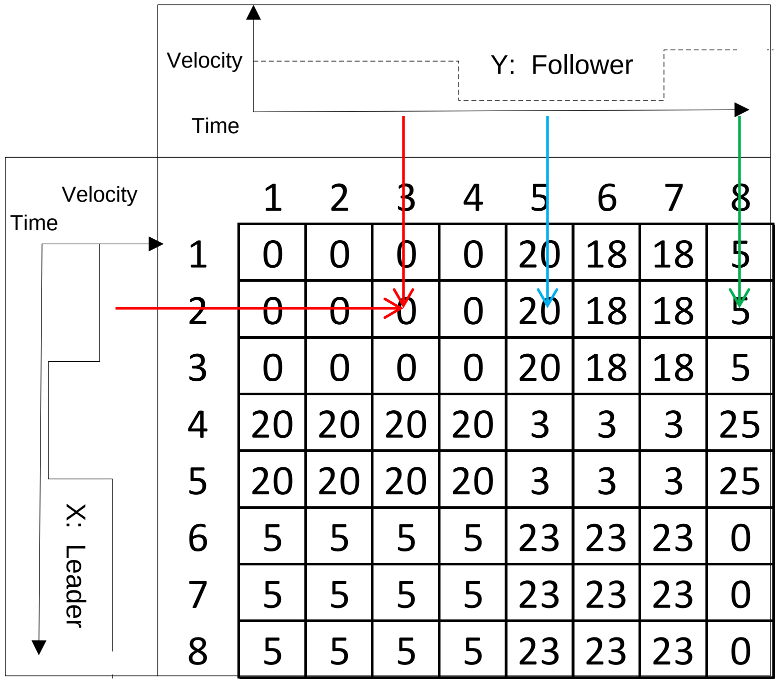
"Warped" Time Axis

Nonlinear alignments are possible.

Construct Cost Matrix for Traffic Trajectory Matching



(A) Vehicle Trajectories with DTW Match Solution



	1	2	3	4	5	6	7	8
1	0	0	0	0	20	18	18	5
2	0	0	0	0	20	18	18	5
3	0	0	0	0	20	18	18	5
4	20	20	20	20	3	3	3	25
5	20	20	20	20	3	3	3	25
6	5	5	5	5	23	23	23	0
7	5	5	5	5	23	23	23	0
8	5	5	5	5	23	23	23	0

(C) Cost Matrix (for Velocity)

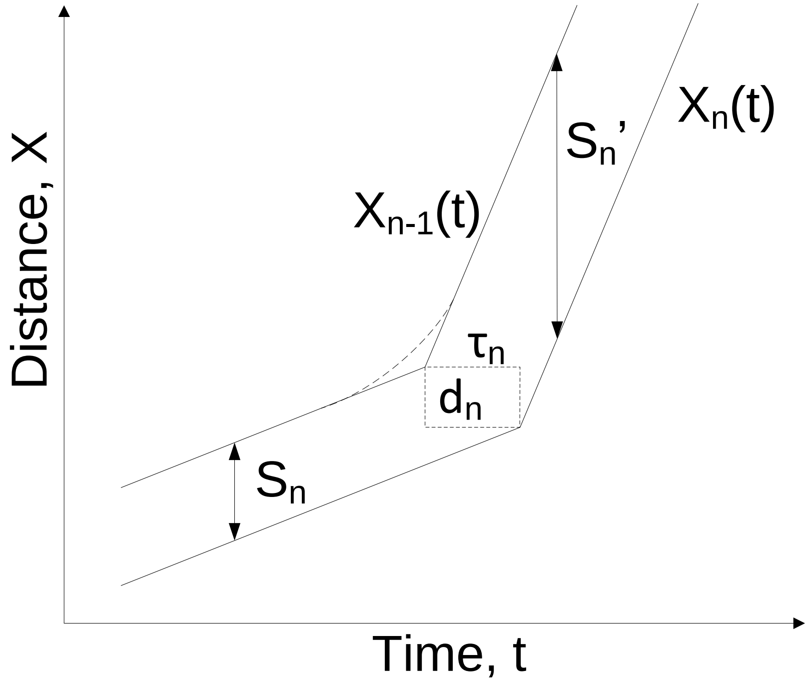
$$C(X_i, Y_j) = \left|X_i - Y_j\right| + \alpha\left|j - i - \tau^-\right| + \beta\left|x_L(i) - x_F(j) - d^-\right| + r\left|\frac{x_L(i) - x_F(j)}{j - i} - w^-\right|$$

Application to Newell's Model

Follower separated by leader by reaction time and critical jam spacing

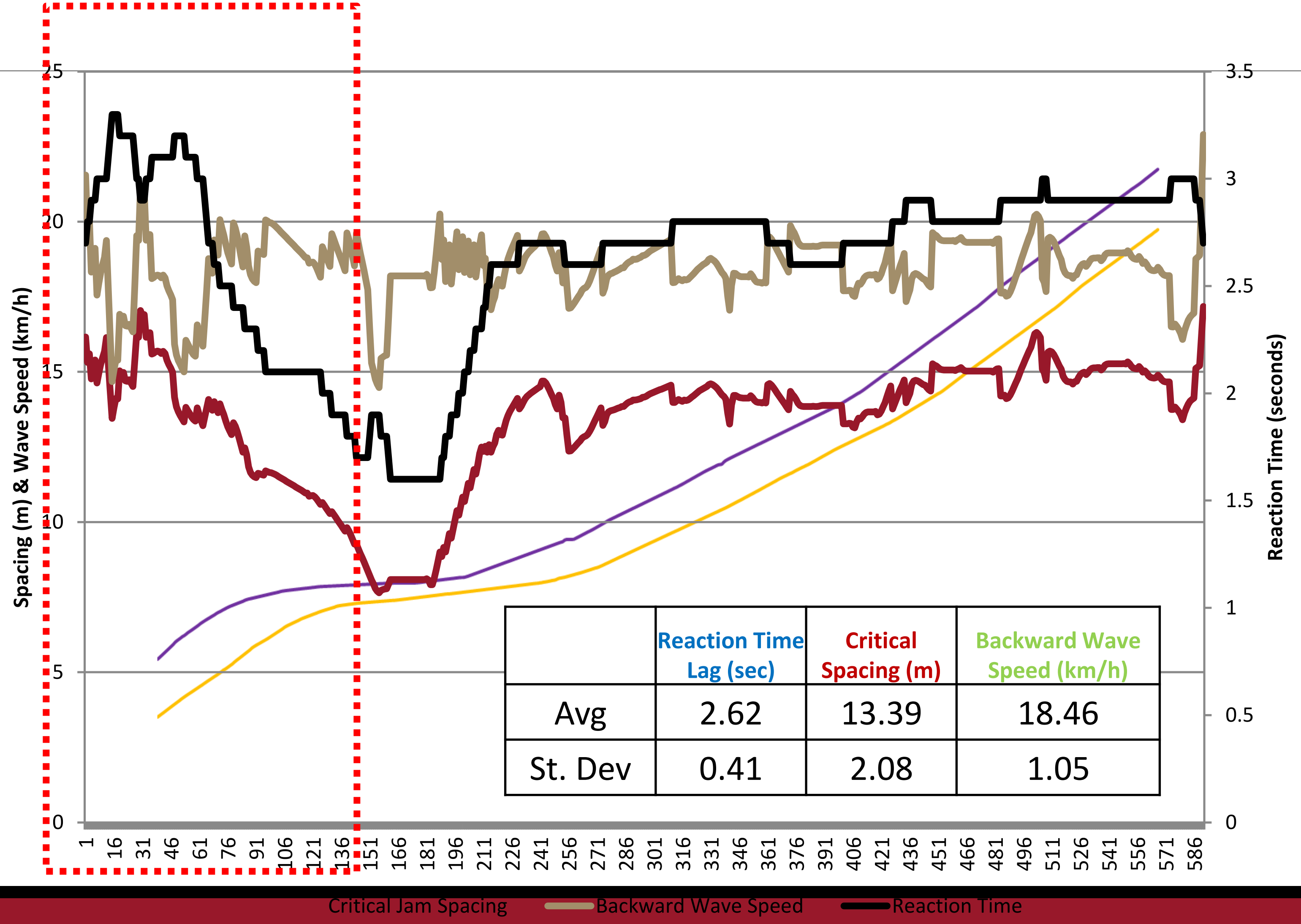
Algorithm finds optimal τ_n (time lag) for best velocity match

- Calculate d_n for all time steps along the trajectory



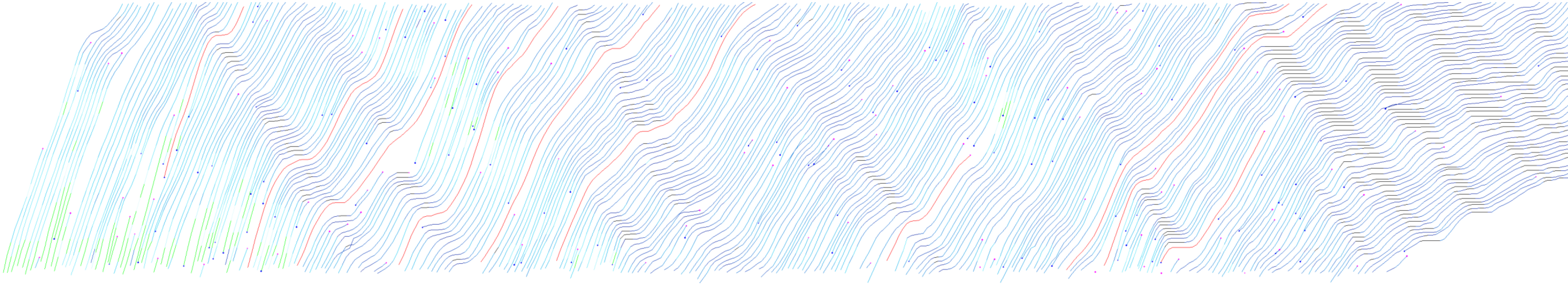
$$x_n(t+\tau_n) = x_{n-1}(t) - d_n$$

Calibrated Parameters: Car 1737

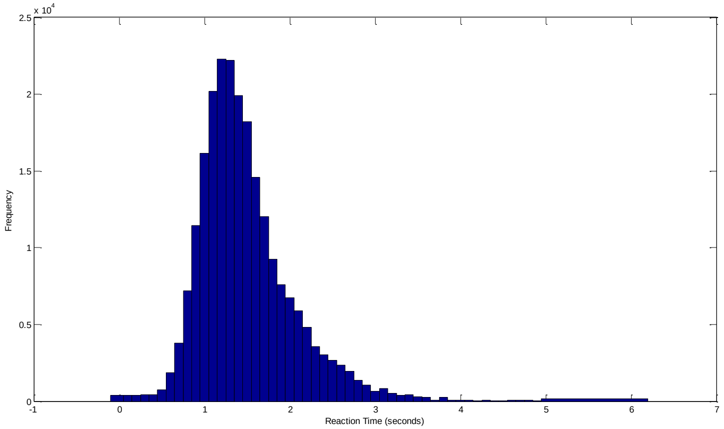


	Reaction Time Lag (sec)	Critical Spacing (m)	Backward Wave Speed (km/h)
Avg	2.62	13.39	18.46
St. Dev	0.41	2.08	1.05

NGSIM Data: I-80 Lane 4



Taylor, J., Zhou, X. Rouphail, N., Porter, R.J. (2015) Method for investigating intradriver heterogeneity using vehicle trajectory data: A Dynamic Time Warping approach. Transportation Research Part B, 73, 59-80

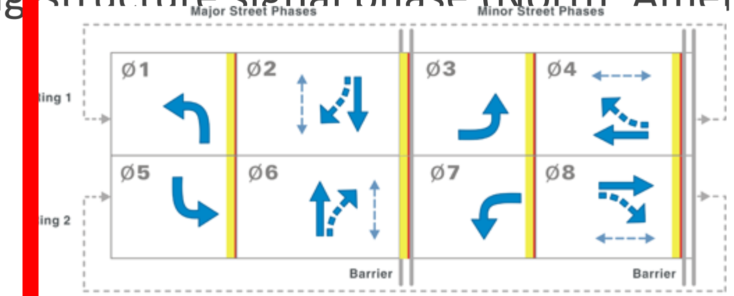


Reaction Time Distribution

Application 3: Phase-time network for Signal Optimization

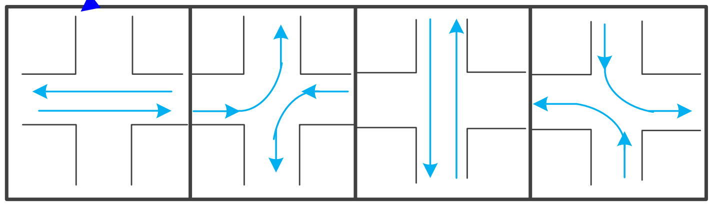
Traffic signal phasing sequence representation

- NEMA ring structure signal phase (North America)

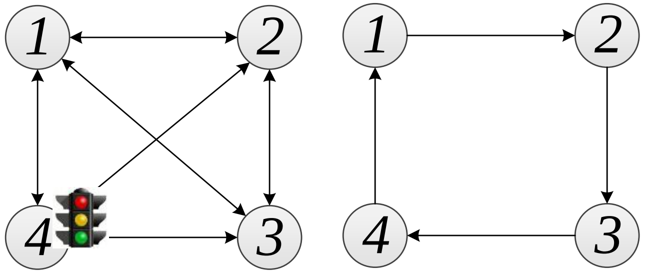


Source: Traffic Signal Timing Manual: FHWA

- Movement-based signal phase (same control flexibility, fewer variables)



Signal Phases



Flexible Phasing Sequence

Cyclic Phasing Sequence

P. Li., P. Mirchandani, X. Zhou, Solving Simultaneous Route Guidance and Traffic Signal Optimization Problem Using Coupled Space-time and Phase-time Networks. Transportation Research Part B. 81, 103-130.

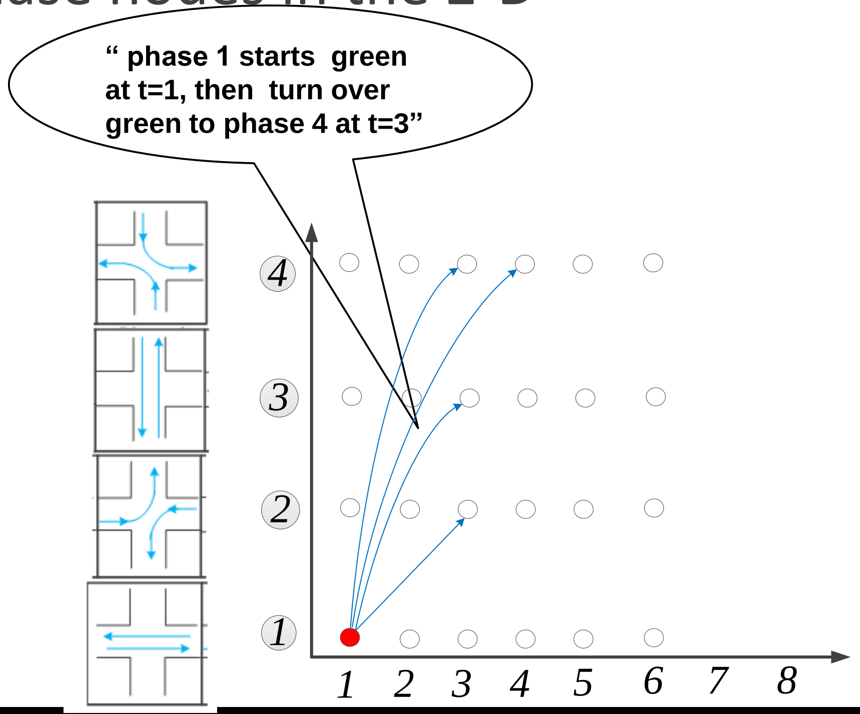
Phase-time network for Signal Optimization

- A signal timing plan is composed of:
  - Phasing sequence
  - Phase duration
- Signal timings can also be represented with a series of phase nodes in the 2-D phase-time network

- Similar structure with a space-time network!

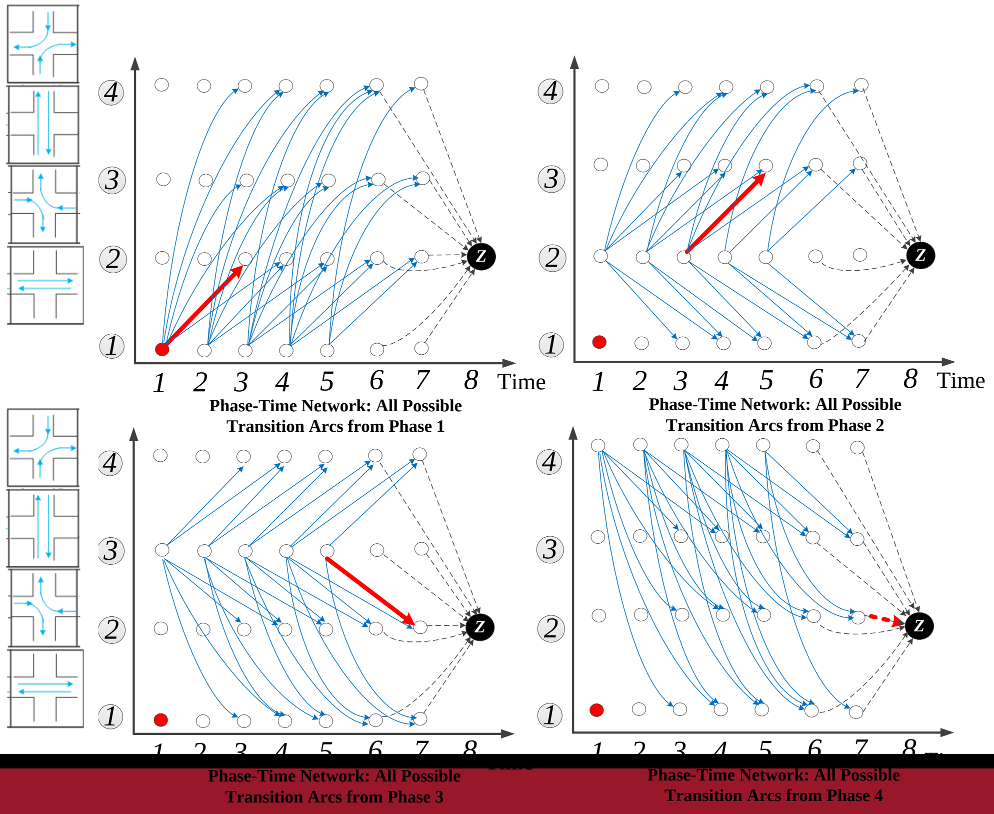
- Solution is provided like:

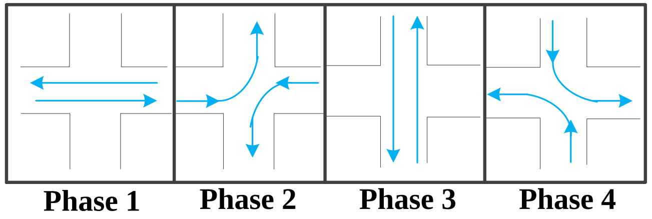
$$w_{(m,k,\tau,h)} = \begin{cases} 1, \text{ if signal phase } m \text{ is green at } \tau \\ \quad \text{and then turns green to signal phase } k \text{ at } h \\ 0, \ otherwise \end{cases}$$



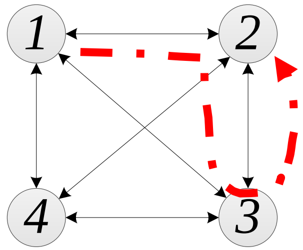
Optimize traffic signal in phase-time network

- Each green phase will generate cost on other phases (e.g., delays)
- Find a least-cost path from origin (starting phase) to destination (end of horizon)





Signal Phases

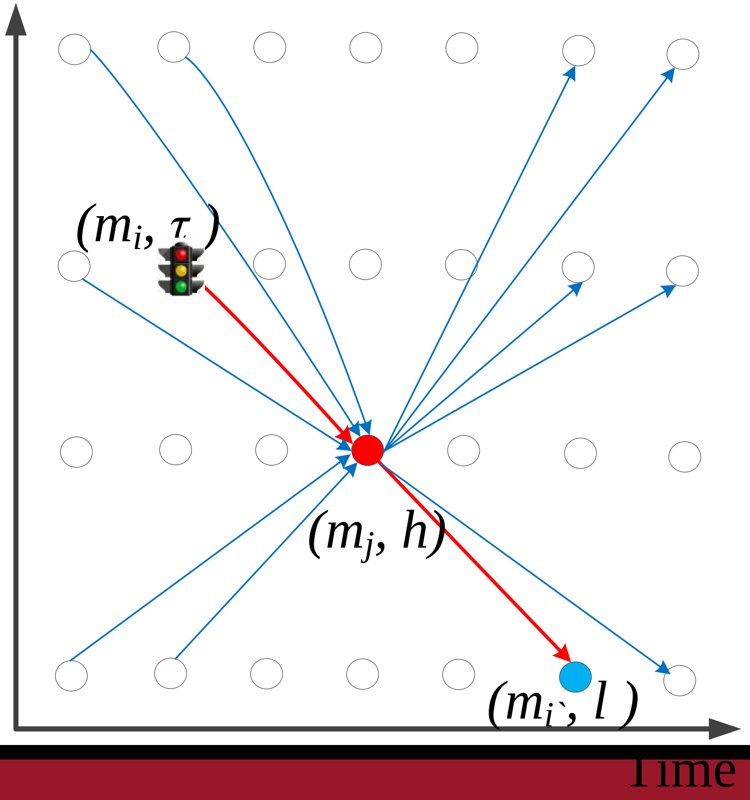


Flexible Phasing Sequence

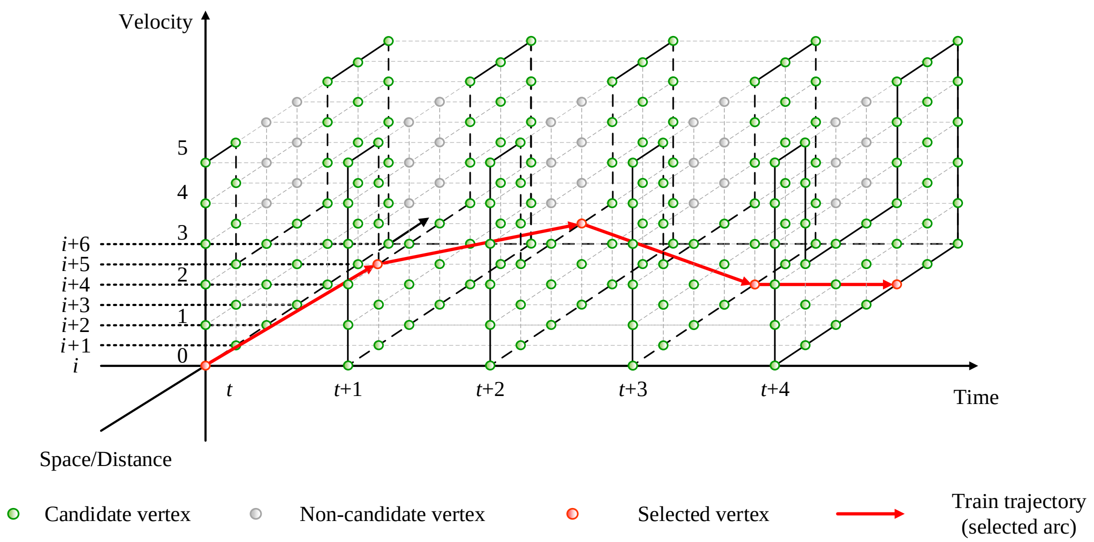
Intersection control constraints

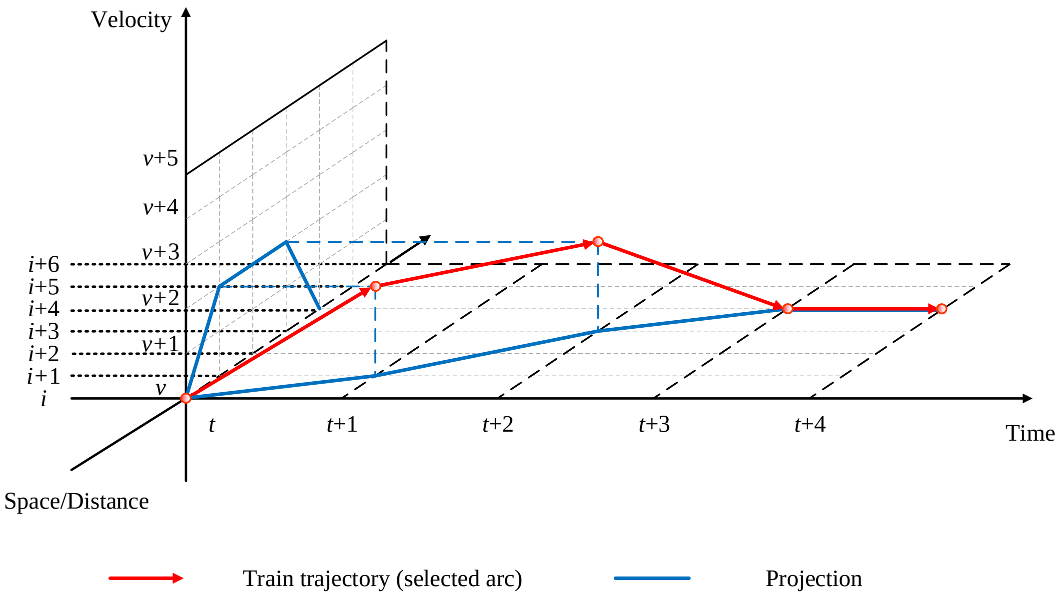
Mutual exclusiveness of signal phases

- Any signal phase has one and only one predecessor phase and successor phase at one time.



Application 4: Speed-space-time network for high-speed train timetabling and speed control





Conclusions

Present a state-space-time (SST) based modeling framework

By adding additional state dimensions, we prebuild many complex constraints into a multi-dimensional network

- Computationally efficient solution algorithm: forward dynamic programming + Lagrangian relaxation
- Wide range of applications
- (i) how to estimate macroscopic and microscopic freeway traffic states from heterogeneous measurements,
- (ii) how to optimize transportation systems and ride-sharing services involving vehicular routing decisions with pickup and delivery time windows (VRPPDTW).

Challenges

- Selecting state is an art…
- How to overcome curse of dimensionality
- Dynamics and uncertainty (demand/supply)
- Smart search space reduction and metaheuristics algorithms for real-time applications