Network traffic flow simulator for AutoDiff-based lightning-fast optimization

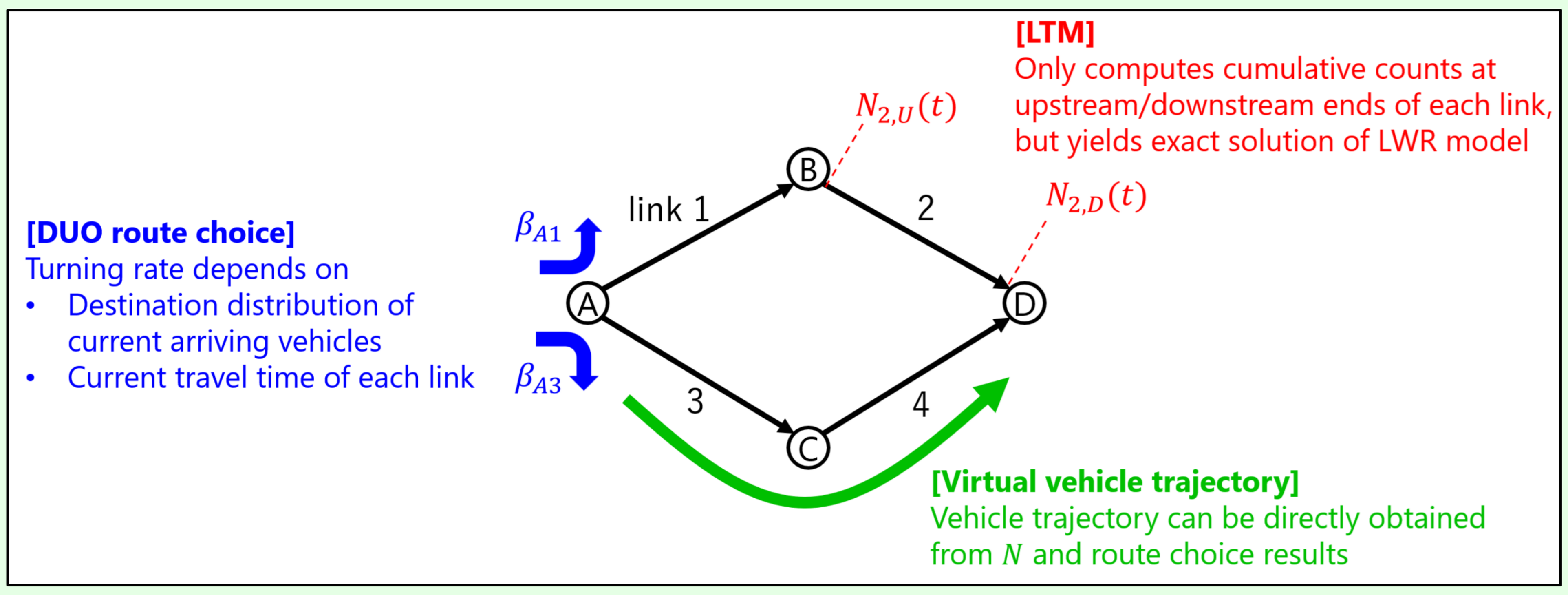
Toru Seo (Institute of Science Tokyo; seo.t.aa@m.titech.ac.jp)

Background

- Network traffic flow simulation is often used for various optimization tasks
  - E.g., Optimal control, OD demand calibration by minimizing residuals
- In general, optimization requires *gradient of objective functions with respect to input parameters*
  - E.g., sensitivity of toll to each link to total travel time (TTT): $\partial TTT/\partial \tau_l$, ∇TTT
- However, derivation of such gradients for existing network traffic simulation is usually difficult or impossible
- This work develops end-to-end differentiable network traffic simulator based on *automatic differentiation (AD)*, in which exact gradients from an arbitrary output to all inputs can be obtained simultaneously
  - Equivalent to backpropagation of deep learning. No need of costly approximations like finite difference
- The proposed simulator, termed *UNsim*, is implemented by Python and JAX and published as OSS
  - https://github.com/toruseo/UNsim or `pip install unsim`

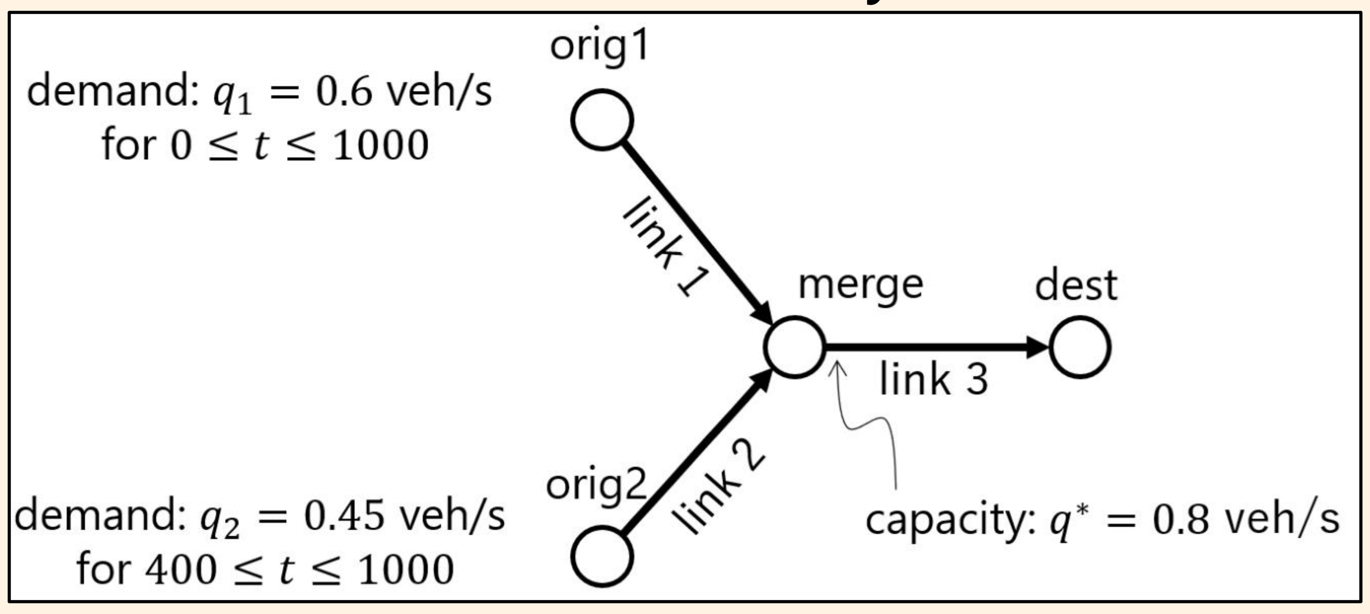
Simulation model

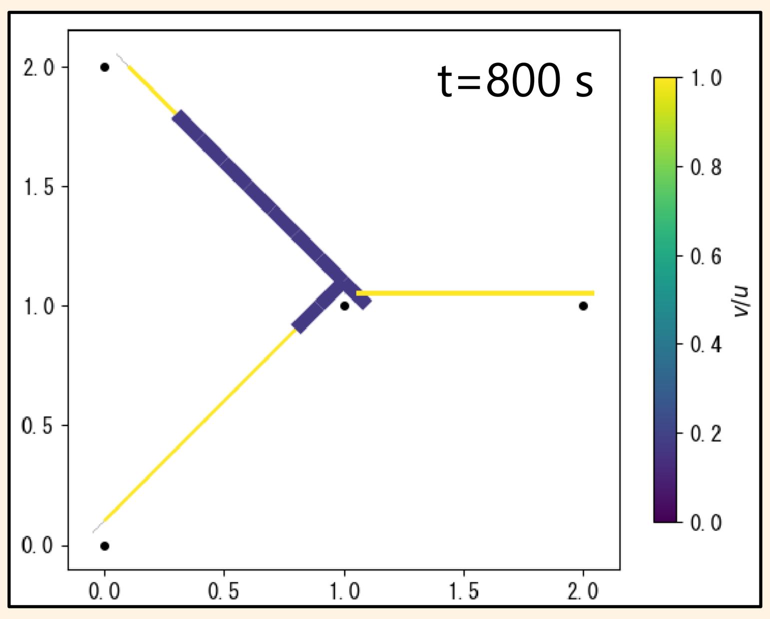
- Inputs: network structure, dynamic OD demand link and node attributes, etc.
- Outputs: TTT, time-varying link traffic states, individual vehicle trajectories, etc.
- Base models suitable for AD are employed
  - Link Transmission Model for link traffic dynamics
  - Incremental Node Model for intersections
  - Dynamic User Optimum model for shortest-path seeking route choice
- They are reformulated to make them applicable for AD
- By using AD, we can directly and simultaneously obtain gradients from an arbitrary output to all inputs
  - The computation time of AD is 2-5x larger than the forward simulation regardless of the dimension of inputs



Numerical experiments

1. Verification: The accuracy of the model is confirmed



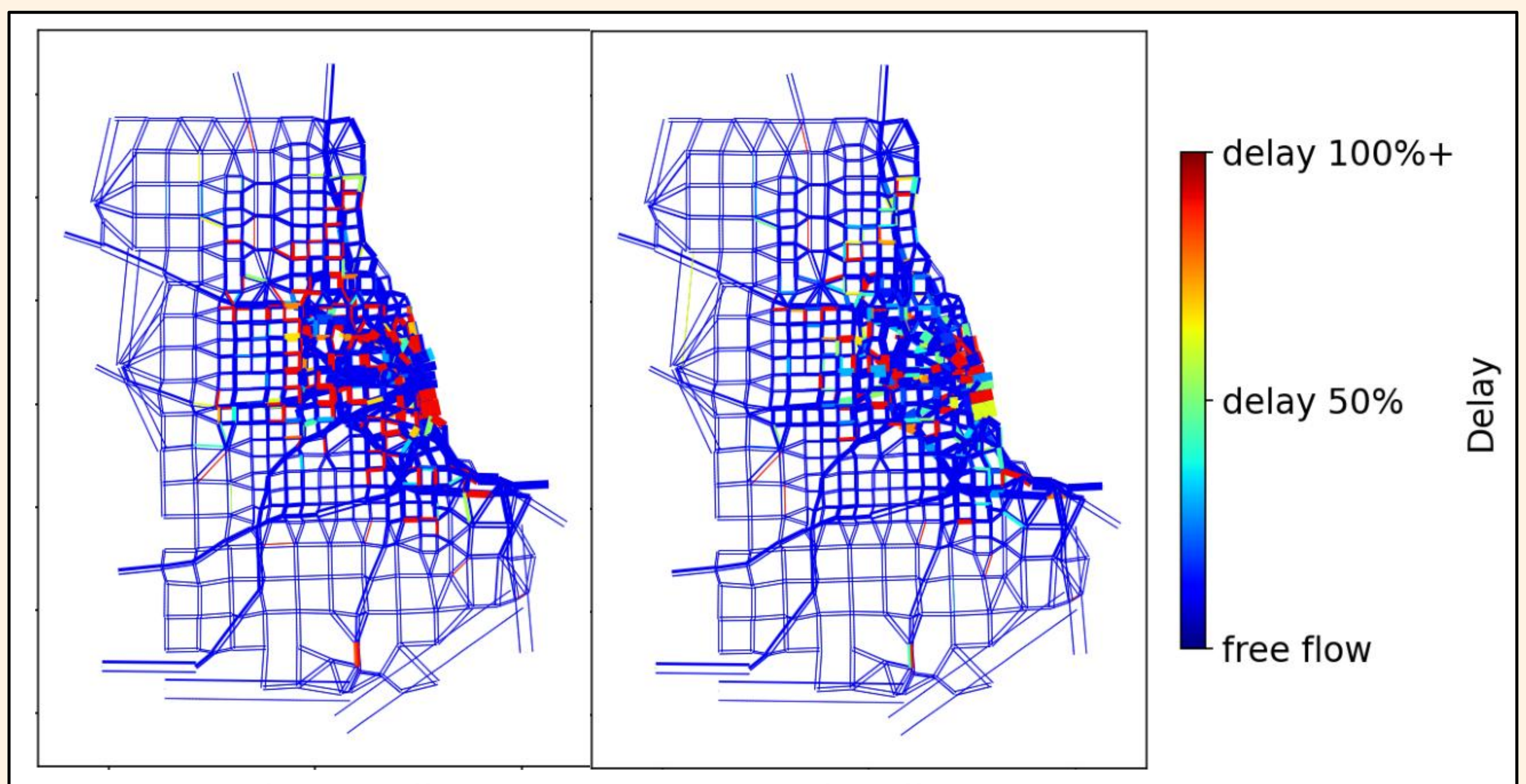


Partial derivative	AD	Finite Difference $\epsilon = 10^{-3}$
∂TTT / ∂q_1	437455.688	448187.500
∂TTT / ∂q_2	421503.938	434820.313
∂TTT / ∂u_1	−1278.282	−1320.313
∂TTT / ∂u_2	−616.873	−578.125
∂TTT / ∂u_3	−2024.685	−2023.438
∂TTTlink1 / $\partial \alpha_1$	−45900.031	−44746.094
∂TTTlink2 / $\partial \alpha_1$	40725.027	35679.688
∂TT(100, orig1, dest) / $\partial \alpha_1$	0.000	0.000
∂TT(500, orig1, dest) / $\partial \alpha_1$	−56.250	−56.244
∂TT(100, orig2, dest) / $\partial \alpha_1$	0.000	0.000
∂TT(500, orig2, dest) / $\partial \alpha_1$	75.000	75.043

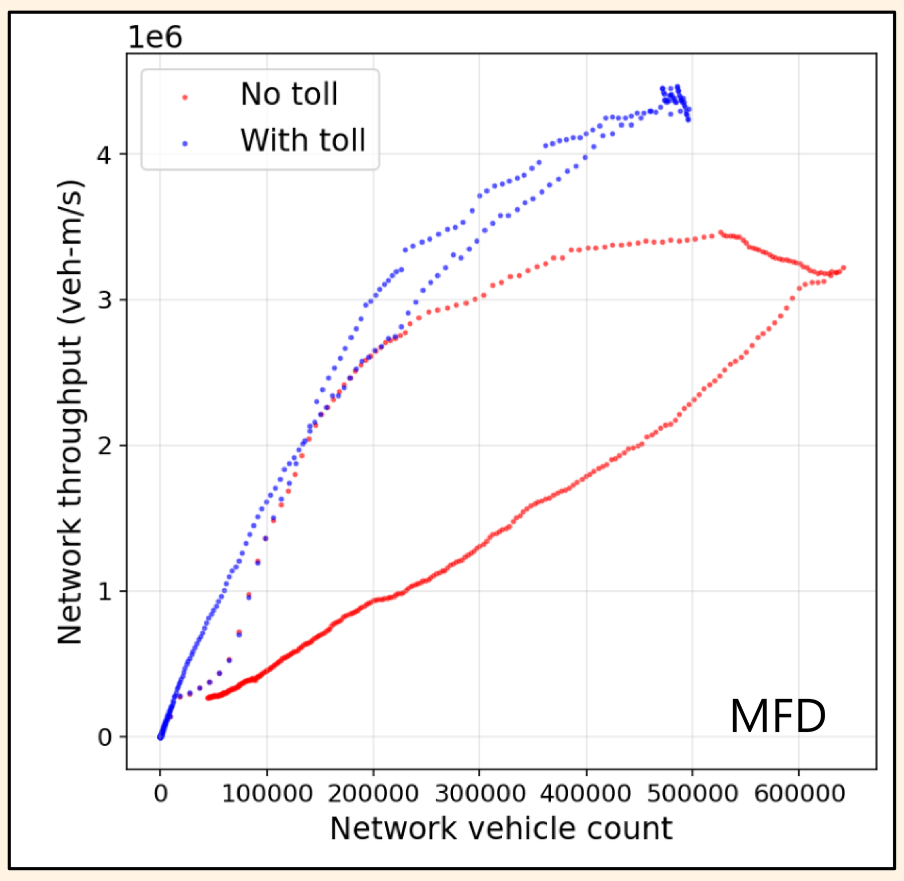
q_r: OD demand from origin r
u_l: free-flow speed of link l
α_l: merging priority
TTTlink$l$: total travel time on link
TT(t, r, s): vehicle trip travel time

2. Large-scale optimization

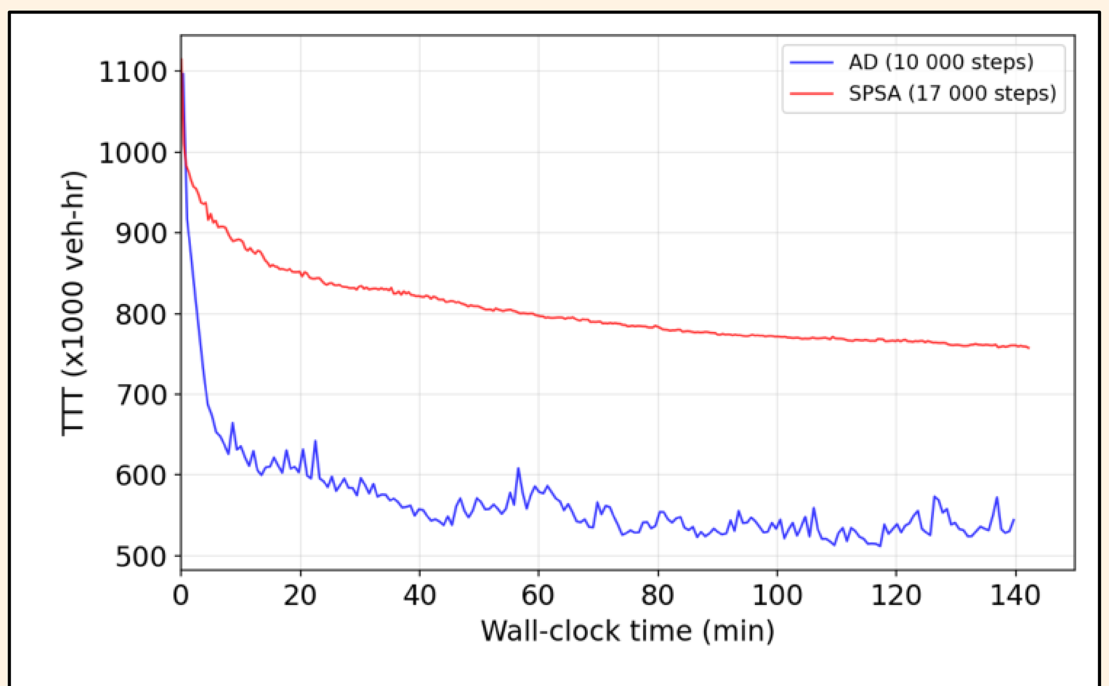
- Dynamic congestion pricing optimization problem on Chicago network is solved
  - 927 nodes, 2557 links, 3 hour durations, 1 million vehicles
  - Decision variables: 15320 toll-price values (383 congested links x 40 time periods)
- 1 forward simulation: 0.2 sec, 1 reverse AD: 0.8 sec using NVIDIA GH200
- TTT was reduced from 1.12 million veh-hr to 0.51 million veh-hr by 10000 iterations
- The performance was significantly higher than SPSA, a conventional gradient-free optimization method



without toll with toll



MFD



Convergence plot, with comparison with SPSA

Further reading

Seo, T. End-to-end differentiable network traffic simulation with dynamic route choice. *arXiv preprint arXiv: 2604.11380*, 2026