ネットワーク交通流理論の基礎・応用・実践

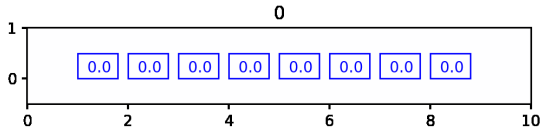
瀬尾亨

東京科学大学
seo.t.701d@m.isct.ac.jp
第8回交通・都市理論ドクター・若手勉強会 2026-10-26 金沢大学

- 基礎
  - 交通流理論
  - 動的交通量配分
  - メソ交通流モデル・シミュレーション
  - 疑似DUE・DSO
- 応用・実践 1
  - メソ交通流シミュレータUXsim
- 応用・実践 2
  - 微分可能交通流シミュレータUNsim
- OSS活動の実践で得られた知見

ネットワーク交通流理論



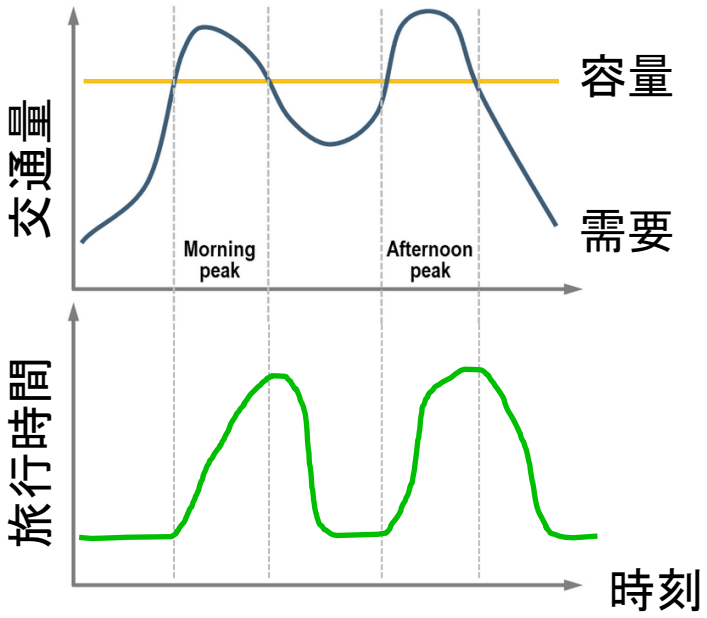


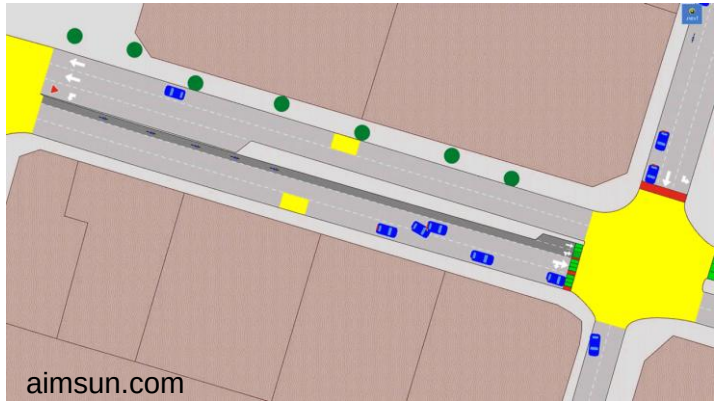
$$\text{find } h^* \in \Lambda \text{ s.t.}$$

$$\langle \Psi_p(t, h^*), h - h^* \rangle \geq 0 \quad \forall h \in \Lambda$$

- ネットワーク交通流理論
  - ネットワークでの交通流の挙動を記述
  - その部品
    - リンクでの交通流・渋滞の伝播
    - ノードでの交通流・渋滞の伝播
    - ネットワーク上での旅行者の経路選択
- 交通流理論
  - 道路上の交通流の挙動を記述
  - 基本的にはリンク重視だが，ノードも含む
- 動的交通量配分（DTA: Dynamic Traffic Assignment）
  - ネットワーク交通流理論と重複もする
  - 特に経路選択・配分原則に関する理論的取扱いを重視

ネットワーク交通流理論



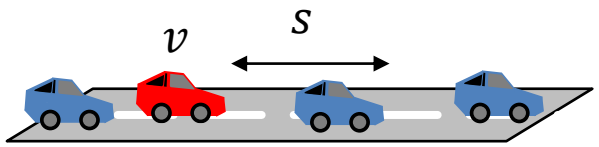


- 静的交通量配分との違い
  - 時間変動を直接記述
  - 空間的な渋滞の伝播を記述
    - 静的交通量配分では，あるリンクの混雑は他に影響しない．現実および動的では，上流リンクに溢れていく
- ミクロ交通流モデルとの違い
  - 細かい車両挙動（加減速など）の記述は粗い
  - 面での2次元的な挙動は考えない
    - 車線変更
    - 交差点での交錯
  - パラメータが少なく因果関係が明確

ネットワーク交通流理論



- 都市のような大規模な交通現象の分析に適している
  - そもそも，交通混雑とは需要が時空間的に偏って生じる現象
  - 例：通勤時間帯，混雑課金，自動運転・シェアリング・EVが都市に及ぼす影響など

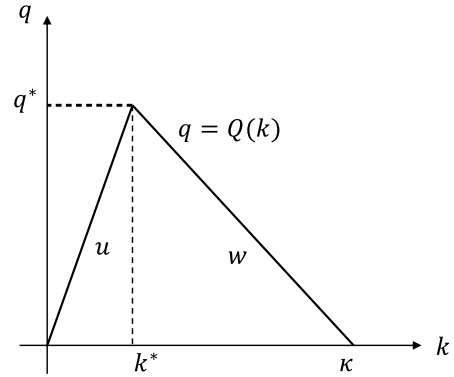


- 原理原則に基づく一貫的な体系
  極論すれば以下の原則だけで全てが組みあがっている
  - 「車は安全の範囲内で可能な限り速く前に進む」
  - 「合流部では方向別に優先権がある」
  - 「旅行者は経路を*なんらか*の基準で選択する」
    - ここだけは人間行動の柔軟さがあるので，オープンになっている．単純にしたければ単純なのを選択すればよい

Kinematic Wave理論

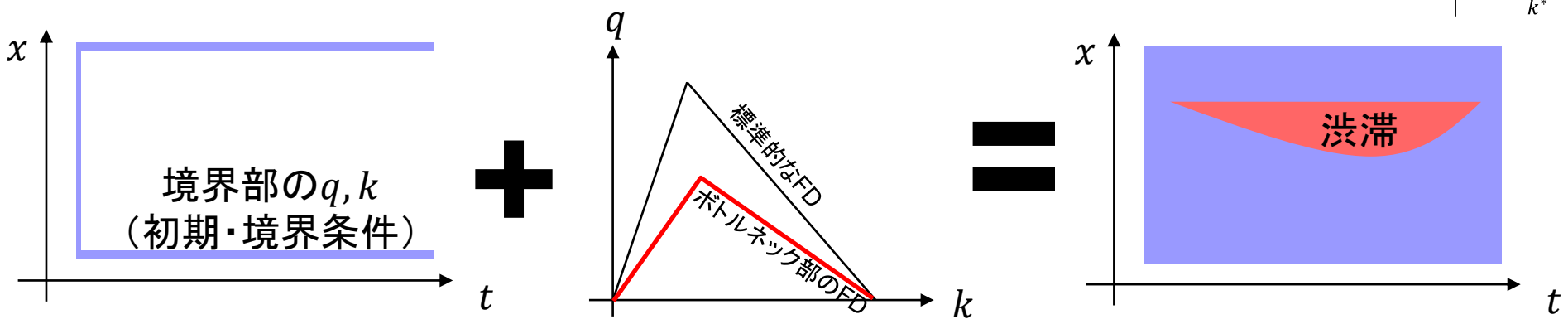
- Kinematic Wave (KW)理論：交通流を流体として近似し交通流の時空間伝播を記述するモデル
  - 交通状態：流率q，密度k，速度v
  - 交通流の流体近似：交通状態が時空間的に連続的に分布している：$k(t,x)$など

- モデルの原則
  - 交通流保存則：車両は理由なく発生，消滅，瞬間移動しない
    - 流体の質量保存則に相当
  - 流率密度関係（Fundamental Diagram: FD）：任意の時空間点の交通状態は必ずFDに従う
  - 交通流は可能な限り前に進む：複数解から妥当なものを選ぶルール



- 初期・境界条件が与えられたときに，他の領域の交通状態を求める

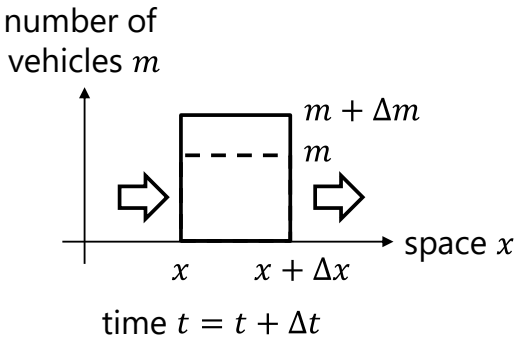
- KW理論には等価な様々な表現がある．順番に紹介する



LWRモデル

- KW理論の基本定式化（導出略）
  - 交通流保存則：車両は理由なく発生，消滅，瞬間移動しない

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0$$



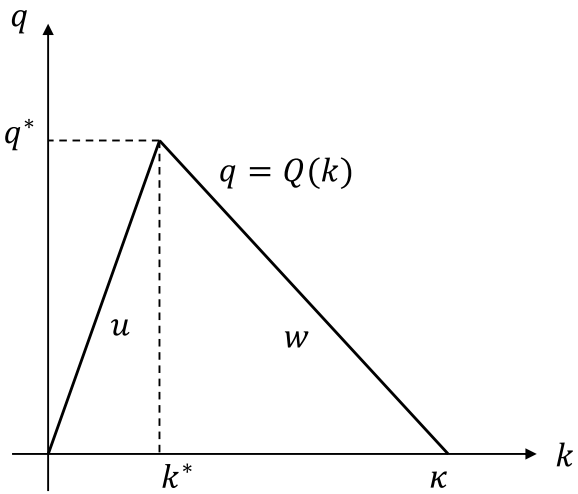
  - FD仮定：任意の時空間点の交通状態は必ずFD関数Qを満たす

$$q = Q(k)$$

  - まとめて

$$\begin{cases} \dfrac{\partial k}{\partial t} + \dfrac{\partial q}{\partial x} = 0 \\ q = Q(k) \end{cases} \quad \text{or} \quad \frac{\partial k}{\partial t} + \frac{\partial Q(k)}{\partial x} = 0$$

  - 未知変数：$k(t,x)$
    - FDと$q = kv$は常に満たされるので，kが決まればq, vも定まる
    - 交通状態が決まれば，車両軌跡も計算可能
  - これを特にLighthill-Whitham-Richards (LWR)モデルと呼ぶ[1,2]



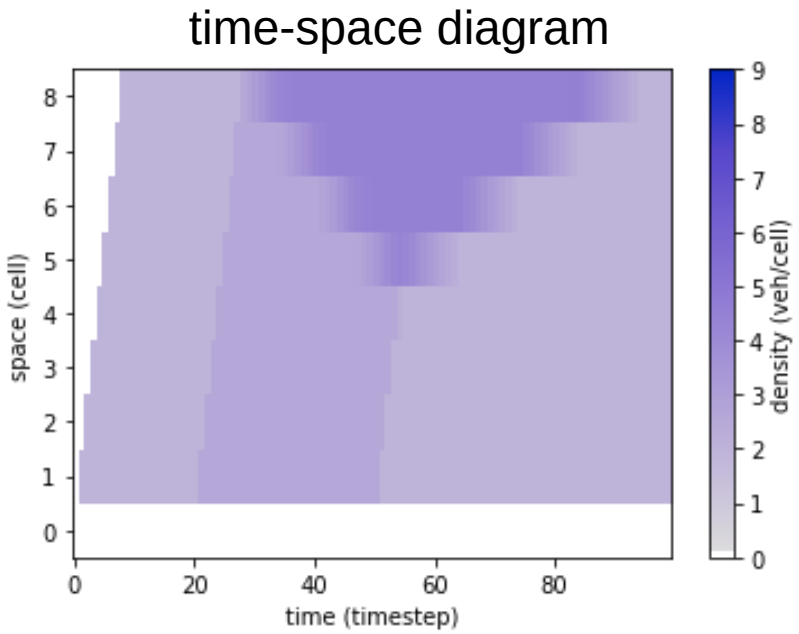
[1] M. J. Lighthill and G. B. Whitham. 1955. On kinematic waves. II. A theory of traffic flow on long crowded roads. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 229, 1178: 317–345.
[2] P. I. Richards. 1956. Shock waves on the highway. Operations Research 4, 1: 42–51.

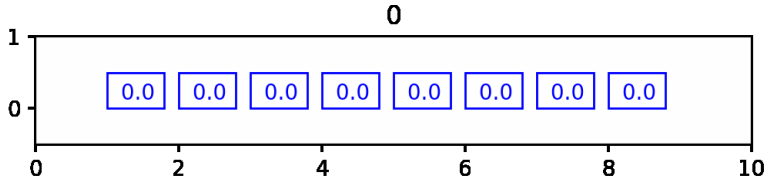
LWRモデル

- LWRモデル

$$\frac{\partial k}{\partial t} + \frac{\partial Q(k)}{\partial x} = 0$$

- 解析的解法：特性曲線法，Variational Theory
- 数値解法・シミュレーション法：Cell Transmission Model





LWRモデルの変形（導出簡略化）

- KW理論には等価な様々な定式化がある
- LWRモデルはEuler座標系(t, x)上で密度kを状態変数とした定式化
- Nモデル：Euler座標系(t, x)上で累積台数Nを状態変数とした定式化

$$\frac{\partial k}{\partial t} + \frac{\partial q}{\partial x} = 0$$
$$q = Q(k)$$

$$k = -\frac{\partial N}{\partial x}$$
$$q = \frac{\partial N}{\partial t}$$

$$-\frac{\partial^2 N}{\partial x \partial t} + \frac{\partial^2 N}{\partial t \partial x} = 0$$
$$\frac{\partial N}{\partial t} = Q\left(-\frac{\partial N}{\partial x}\right)$$

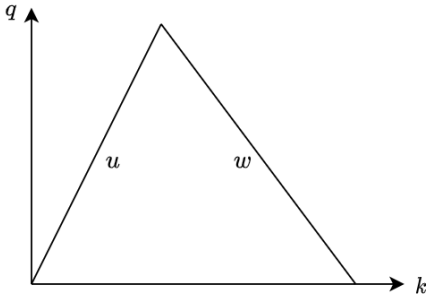
Nがほぼいたるところで連続で微分可能（定義より）

$$\frac{\partial N}{\partial t} = Q\left(-\frac{\partial N}{\partial x}\right)$$

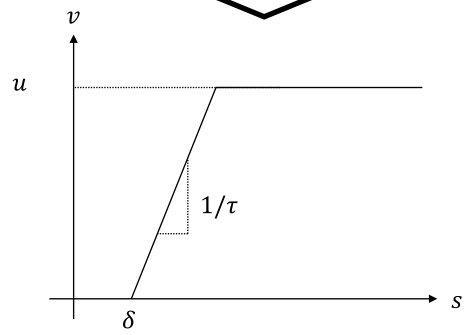
LWRモデルの変形（導出簡略化）

- さらに，Nモデルを Lagrange座標系(t,n)上の車両位置Xを状態変数としたモデルに変換

- Xモデル
  - 状態変数$X(t,n)$：時刻tの車両nの位置
  - $V(s)$：車頭距離sのときの速度．追従曲線に相当
  - これは，個別車両の位置を直接記述するという意味ではミクロ的



FD: $q = Q(k)$



FD: $v = V(s)$

$$v(t,n) = \frac{\partial X(t,n)}{\partial t}$$

$$s(t,n) = -\frac{\partial X(t,n)}{\partial n}$$

$$k = 1/s$$

$$q = kv \Leftrightarrow q = v/s$$

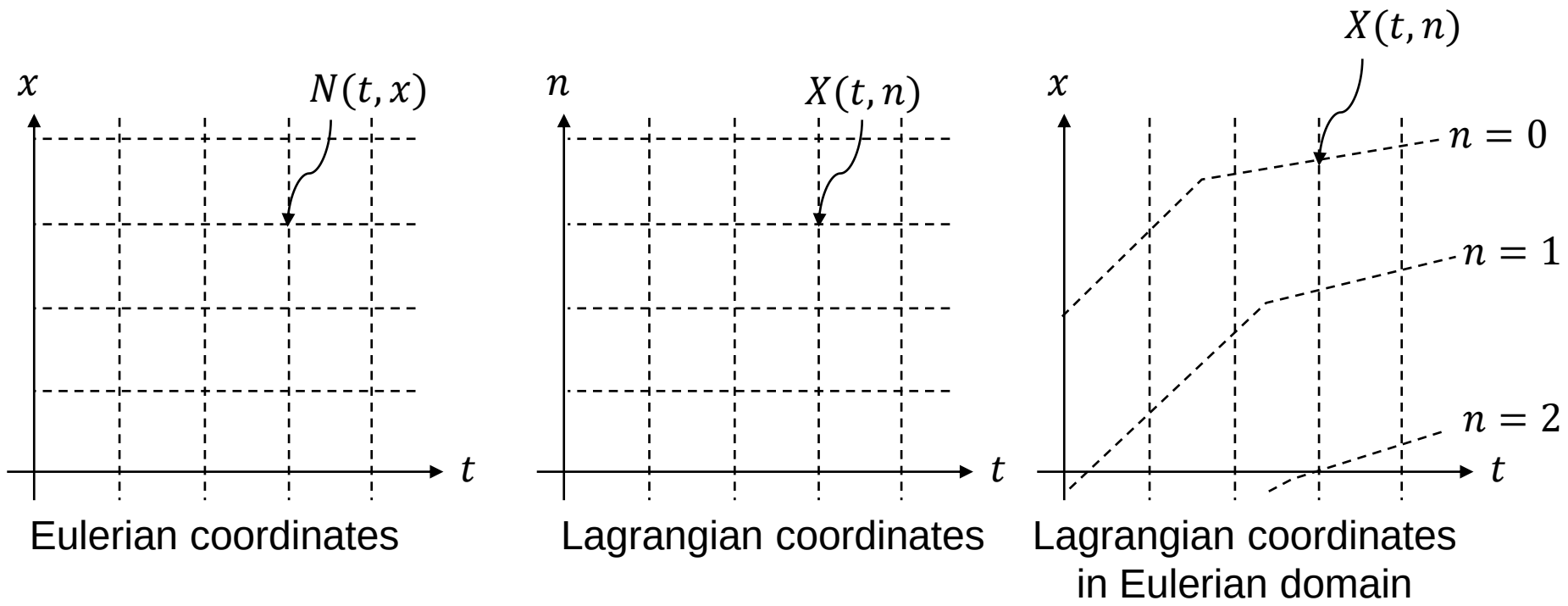
$$\frac{\partial N}{\partial t} = Q\left(-\frac{\partial N}{\partial x}\right) \Rightarrow \frac{\partial X}{\partial t} = V\left(-\frac{\partial X}{\partial n}\right)$$

LWRモデルの変形（導出簡略化）

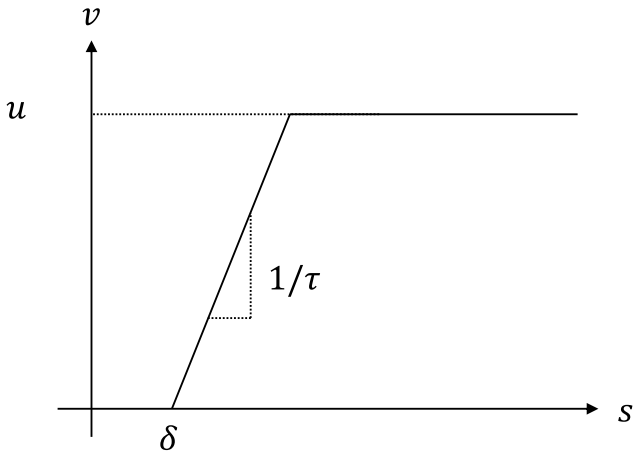
- Xモデル
  - 状態変数$X(t,n)$：時刻tの車両nの位置
  - $V(s)$：車頭距離sのときの速度
  - これは，個別車両の位置を直接記述するという意味ではミクロ的

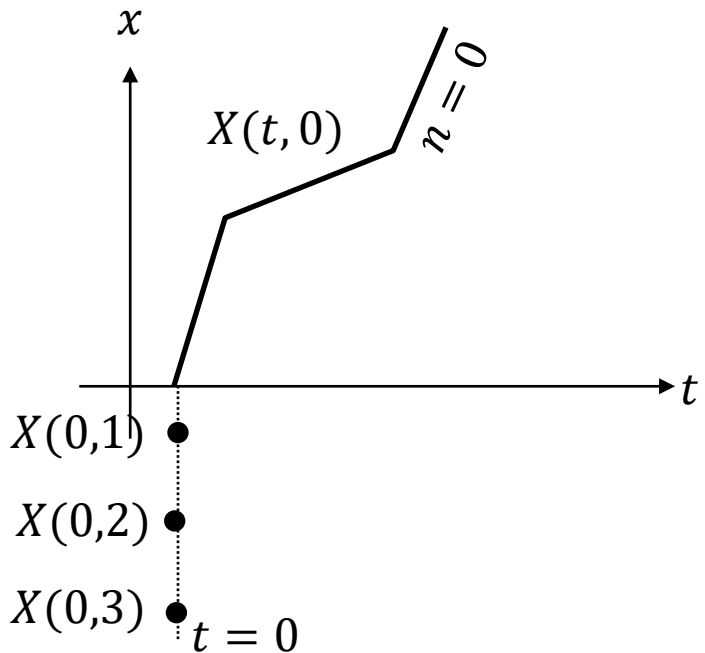
$$\frac{\partial X}{\partial t} = V\left(-\frac{\partial X}{\partial n}\right)$$

- LWRモデル，Nモデル，Xモデルは究極的には等価だが，表現・長所・短所が異なる



Xモデルの解法



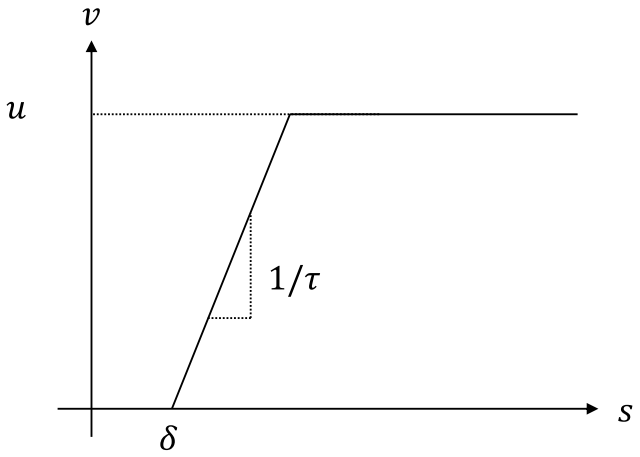


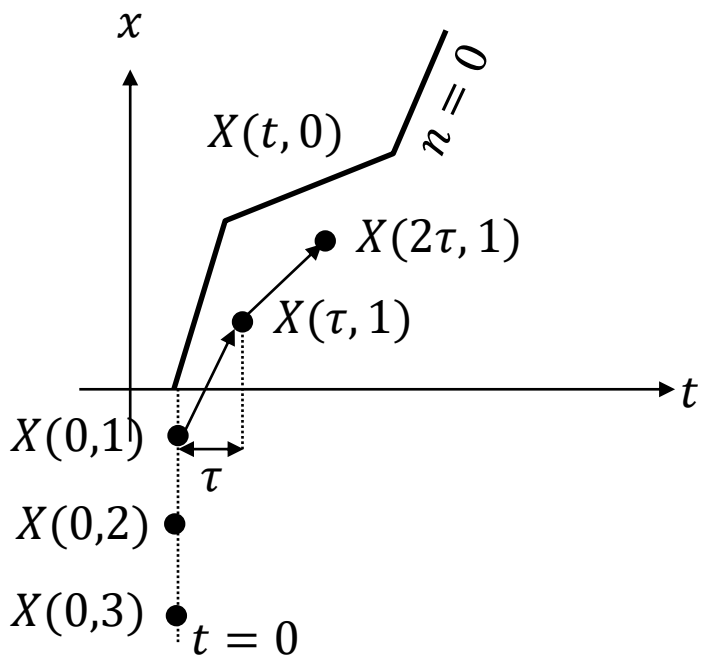
- Xモデル

$$\frac{\partial X}{\partial t} = V\left(-\frac{\partial X}{\partial n}\right)$$

- 初期条件$X(0,n)$: 時刻0の全車両の位置
- 境界条件$X(t,0)$: 車両0の軌跡
- 所与の関係$V(s)$: 車頭距離sのときの速度
- 目的: 全車両の軌跡$X(t,n)$の計算

Newell単純追従モデル

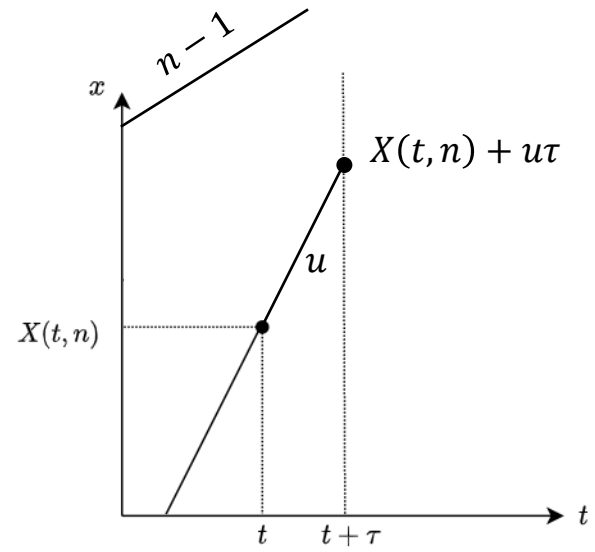


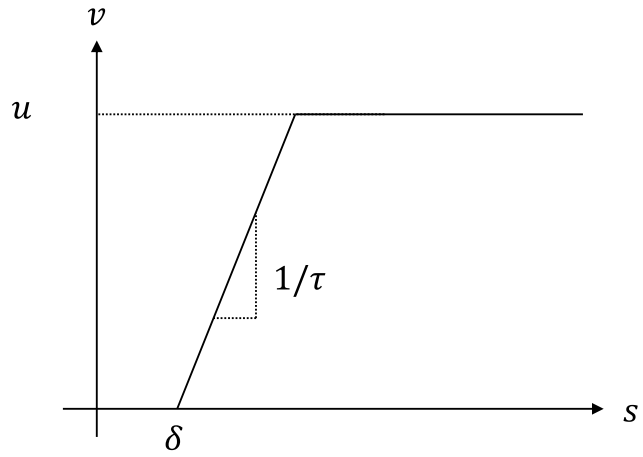


- Newell単純追従モデル[1]：Xモデルの解法
- 速度車頭距離関係は左図とする
- 時刻tの状態に基づき，時刻$t+\tau$の状態を求める
  - $X(t+\tau,n)=f\big(X(t,n),X(t,n-1)\big)$
  - τ: ドライバー反応時間
  - $X(t,n-1)$: 先行車両の位置
  - f: ドライバ―挙動を表す何らかの関数
- 直観的方法で解法を導出する
  - 厳密には変分法を使わないといけないので，この講義からは省略

[1] Gordon Frank Newell. 2002. A simplified car-following theory: a lower order model. Transportation Research Part B: Methodological 36, 3: 195–205.

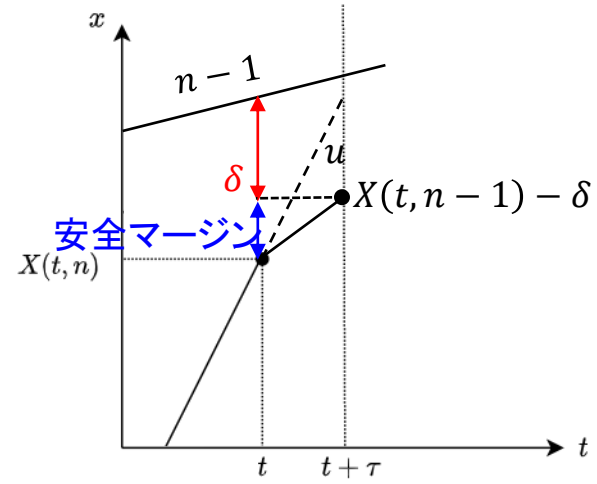
Newell単純追従モデル：直観的導出

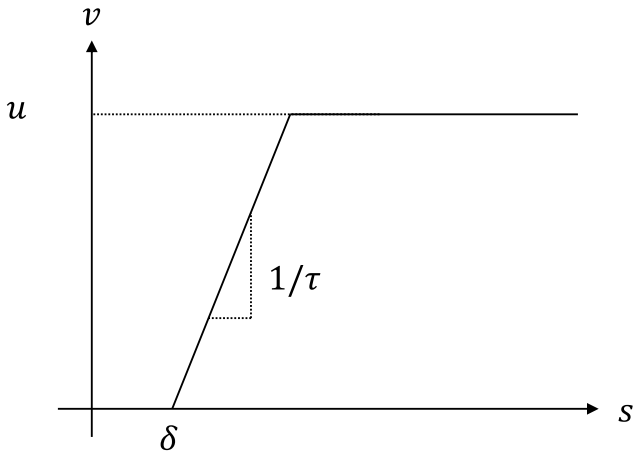




- 時刻tの情報に基づき$X(t+\tau,n)$を求めたい
- この交通流は自由流状態か渋滞状態のどちらかである
- 仮に自由流状態であるとする
  - つまり，先行車$n-1$が十分遠方である
- すると，車両nの速度は最大値uである
- よって，$X(t+\tau,n)=X(t,n)+u\tau$

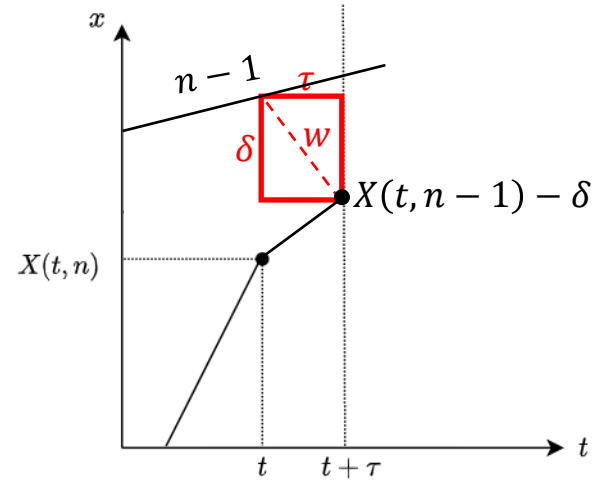
Newell単純追従モデル：直観的導出

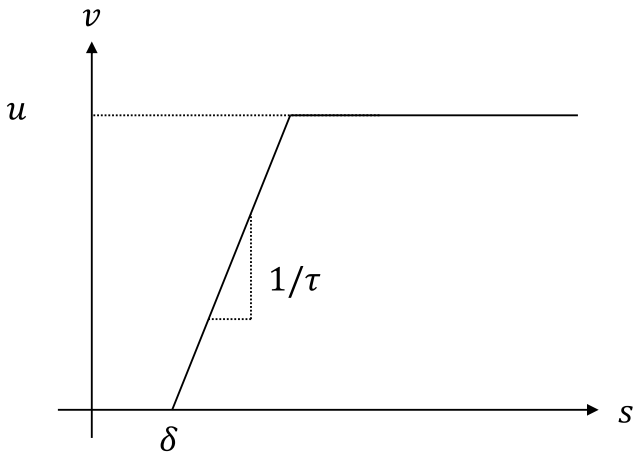




- 次に，仮に渋滞状態であるとする
  - 先行車$n-1$が近い
- このとき，nの速度はu未満になる
- 先行車に対し，最小車頭距離δは絶対に確保しなければならない
- 時刻$t+\tau$まで確実にδを確保するには，$t+\tau$にδの位置まで進めば良い
  - $X(t+\tau,n)=X(t,n-1)-\delta$

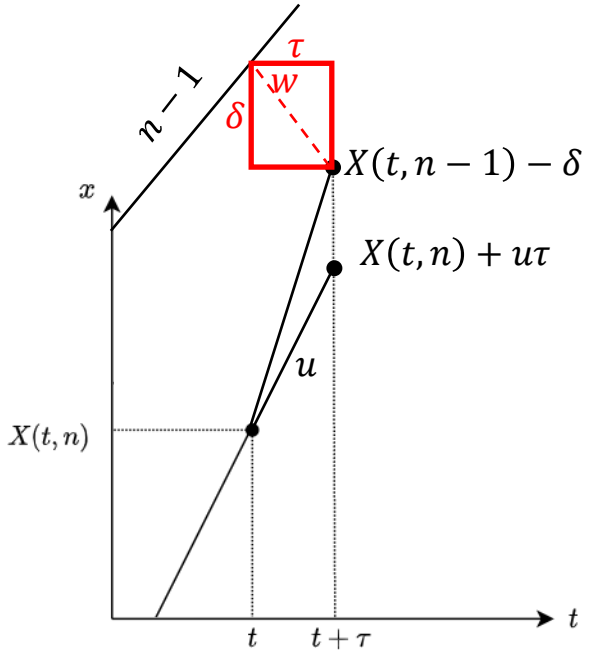
Newell単純追従モデル：直観的導出

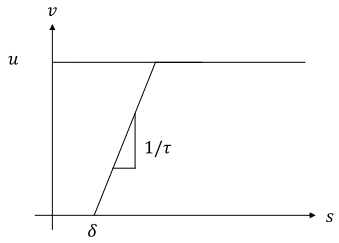


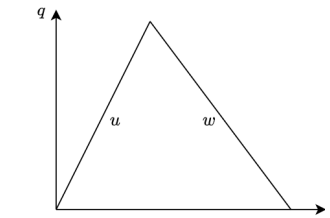


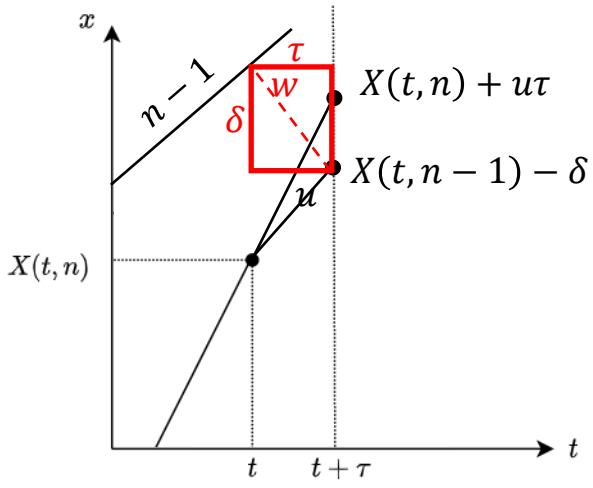
- 言い換えると，交通流が渋滞状態のとき，車両nは先行車$n-1$に対し最小車頭距離δと最小車頭時間τを確保するように動く
  - Newell box

Newell単純追従モデル：直観的導出



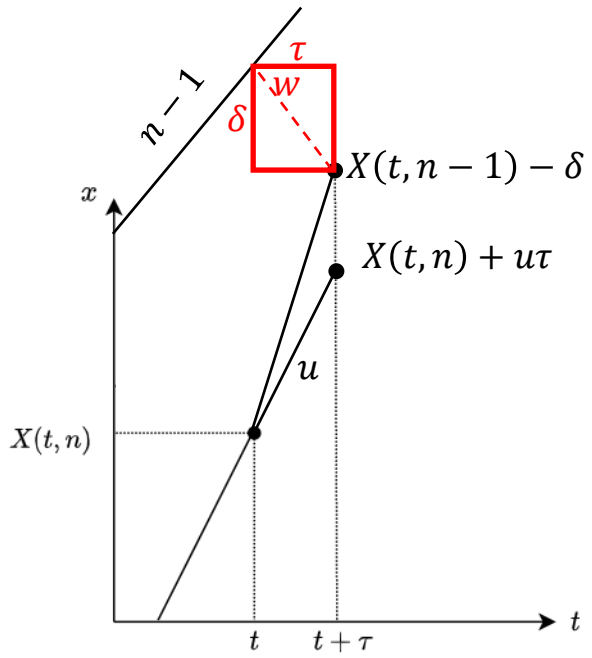


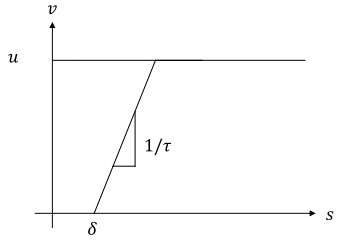


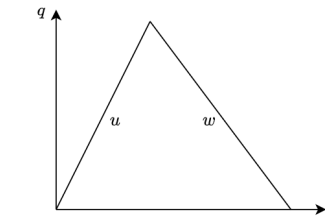


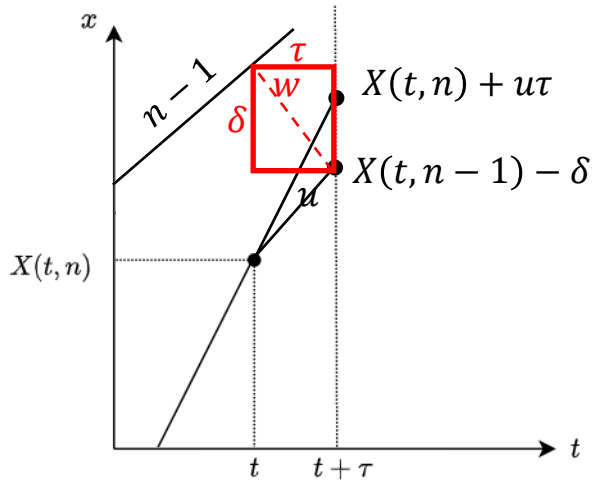
- 交通流は自由流状態か渋滞状態のどちらかである
- 仮に自由流状態であれば
  - $X(t+\tau, n) = X(t, n) + u\tau$
- 仮に渋滞状態であれば
  - $X(t+\tau, n) = X(t, n-1) - \delta$
- どっちが正しい？

Newell単純追従モデル：直観的導出





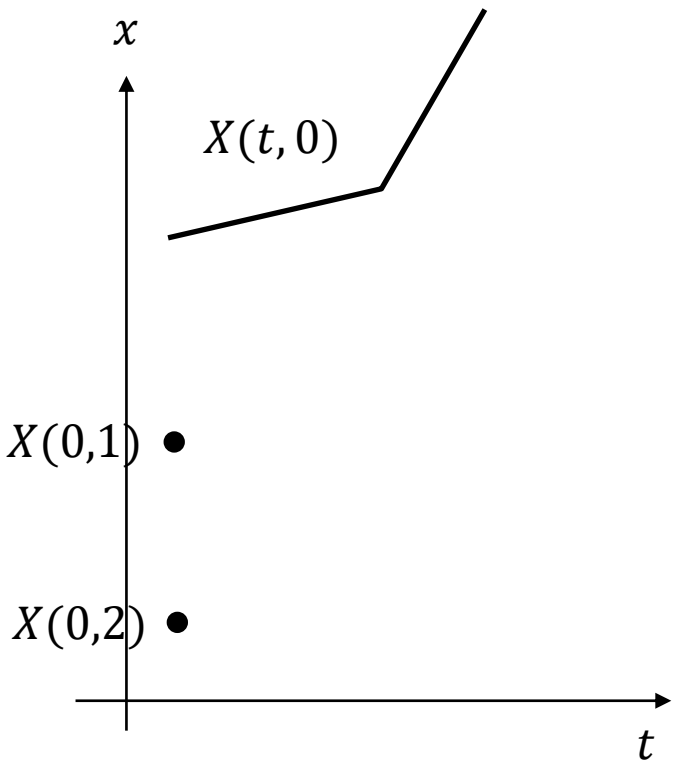


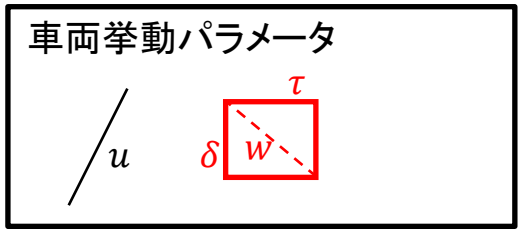


- 二つの候補のうち，小さいほうが正解
- Newell単純追従モデル

$$X(t+\tau, n) = \min \left\{ \begin{array}{l} X(t,n) + u\tau, \\ X(t, n-1) - 1/\kappa \end{array} \right\}$$

- ドライバーの行動原理の解釈
  - 車両は最小車頭距離と最小車頭時間をもつ
  - 車両は最大速度を持つ
  - その制約下で，車両は可能な限り前進する
- KW理論の基本原則再掲
  - 「車は安全の範囲内で可能な限り速く前に進む」

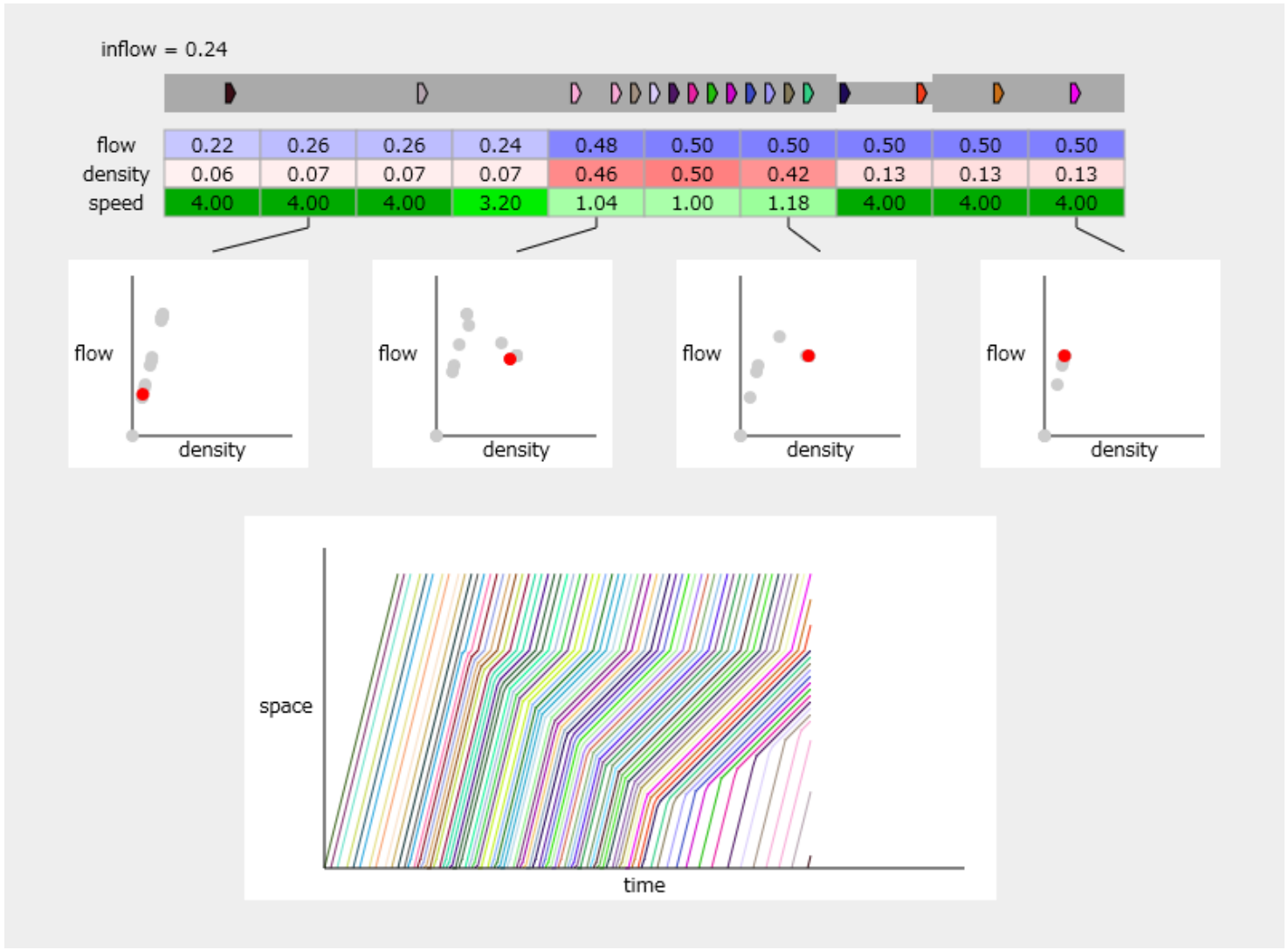




Q. $n=1$と2の車両軌跡の概形をNewell単純追従モデルに従って描け

- ドライバーの行動原理の解釈
  - 車両は最小車頭距離δと最小車頭時間τをもつ（Newell box）
  - 車両は最大速度uを持つ
  - その制約下で，車両は可能な限り前進する

インタラクティブデモ

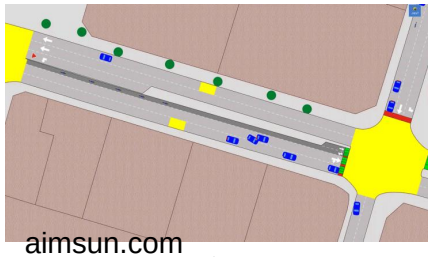


https://toruseo.jp/trafficflow/bottleneck_jp.html

メソ交通流モデル

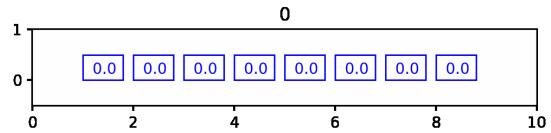
- メソ交通流モデル
  - メソ（mesoscopic）：ミクロとマクロの中間
  - マクロのように大ネットワークを扱い，ミクロのように個別車両を扱いたい
  - 特に，個別車両を扱えると経路選択を直接表現できて都合よい．マクロは若干面倒

- 「メソ交通流モデル」を名乗るモデルには様々なものがある[1,2]
  - gas-kineticモデル：統計物理的アプローチによる交通流モデル
    - Ilya Prigogine [3]（ノーベル化学賞）など．今は下火
  - 離散的な車両を記述するが，その挙動モデルはミクロより粗いモデル
    - リンク平均速度や簡易的な速度密度関係に基づき車両を動かすモデル．DynaMIT [4], SUMO mesosim [5]など
    - リンクの出入りのみを扱うモデル，簡易的な待ち行列を使うモデルなど．MATSim [6], SUMO meso-ltm [7]など



ミクロ

メソ



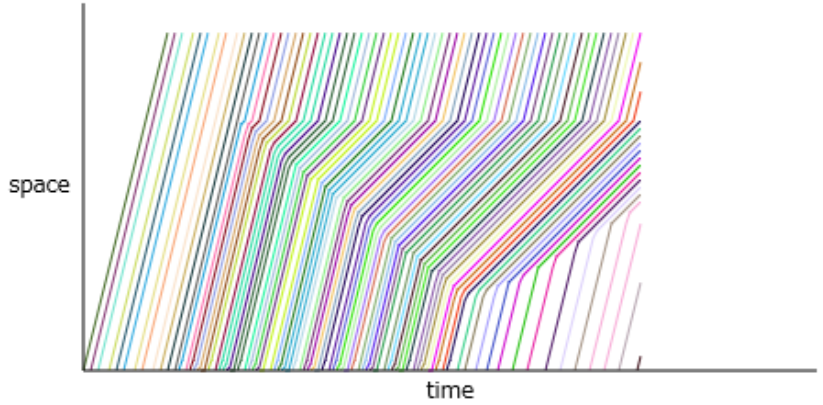
マクロ

[1] S P Hoogendoorn and P H L Bovy. 2001. State-of-the-art of vehicular traffic flow modelling. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 215, 4: 283–303.
[2] Sven Maerivoet and Bart De Moor. 2005. Transportation planning and traffic flow models. arXiv preprint physics/0507127.
[3] Ilya Prigogine and Frank C Andrews. 1960. A Boltzmann-like approach for traffic flow. Operations Research 8, 6: 789–797.
[4] Moshe Ben-Akiva, Michel Bierlaire, Haris Koutsopoulos, and Rabi Mishalani. 1998. DynaMIT: a simulation-based system for traffic prediction. In DACCORD short term forecasting workshop.
[5] Nils Gustaf Eissfeldt. 2004. Vehicle-based modelling of traffic. Theory and application to environmental impact modelling. Universität zu Köln.
[6] Andreas Horni, Kai Nagel, and Kay W Axhausen (eds.). 2016. The multi-agent transport simulation MATSim. London: Ubiquity Press.
[7] Ying-Chuan Ni, Anastasios Kouvelas, and Michail A Makridis. 2026. Simulating link-level interrupted flow traffic dynamics and the comparison between different models for urban road networks. Simulation Modelling Practice and Theory: 103252.

メソ交通流モデル

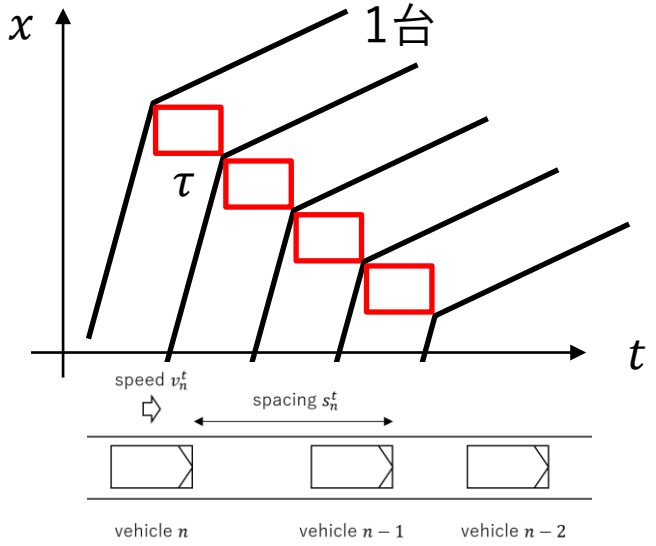
- 原理原則に基づくモデルが欲しい
  - マクロ交通流理論の原理原則：保存則，流率密度関係，最大前進

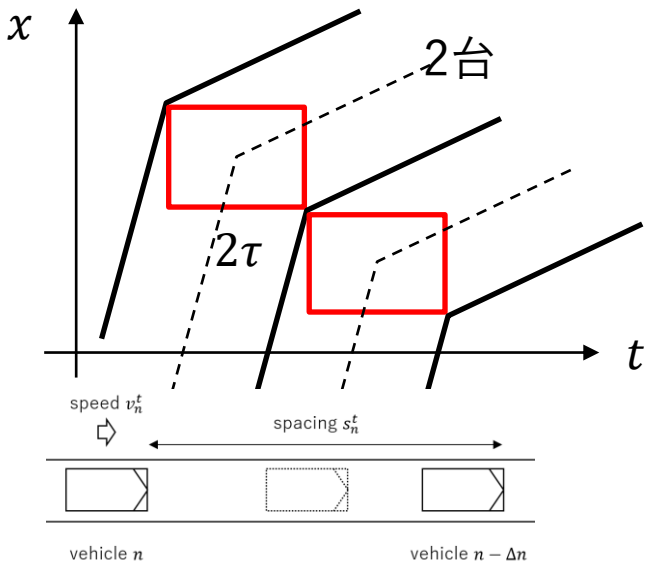
- Newell単純追従モデルはメソといえる
  - マクロ交通流モデルと等価
  - 離散的な車両を考慮
  - 本物のミクロモデルほど精緻な加減速ではない
  - ただし，通常のNewellモデルでは計算時間間隔は反応時間（≒1秒）であり，ミクロモデルと計算コストが同等



メソ交通流モデル

- Newell単純追従モデルの車群表現を用いた集約
  - Newell単純追従モデルは，車両を集約（or間引く）しても適切に計算すれば等価なリンク交通状態の解を得られる[1,2]
  - Δn台の車両を一つの車群とし，Newell単純追従モデルで動かすイメージ
  - 計算時間間隔も$\tau\Delta n$倍になるので，計算コストを$1/\Delta n^2$に削減できる
  - これをメソNewell単純追従モデルと呼ぶ
    - 最初に言いだしたのが誰なのかはわからないが，[3]がこれをメソ追従モデルと呼んでいる





[1] L. Leclercq, J. A. Laval, and E. Chevallier. 2007. The Lagrangian coordinates and what it means for first order traffic flow models. In Transportation and Traffic Theory 2007, 735–753.
[2] Jorge A. Laval and Ludovic Leclercq. 2013. The Hamilton–Jacobi partial differential equation and the three representations of traffic flow. Transportation Research Part B: Methodological 52: 17–30.
[3] Jorge A. Laval and Felipe Castrillon. 2015. Stochastic approximations for the macroscopic fundamental diagram of urban networks. Transportation Research Part B: Methodological 81: 904–916.

動的交通量配分：基本

- 動的交通量配分：ある経路選択ルールのもとで，各時刻・各リンクの交通状態を決定する
  - 交通流理論と経路選択ルールの組合せ

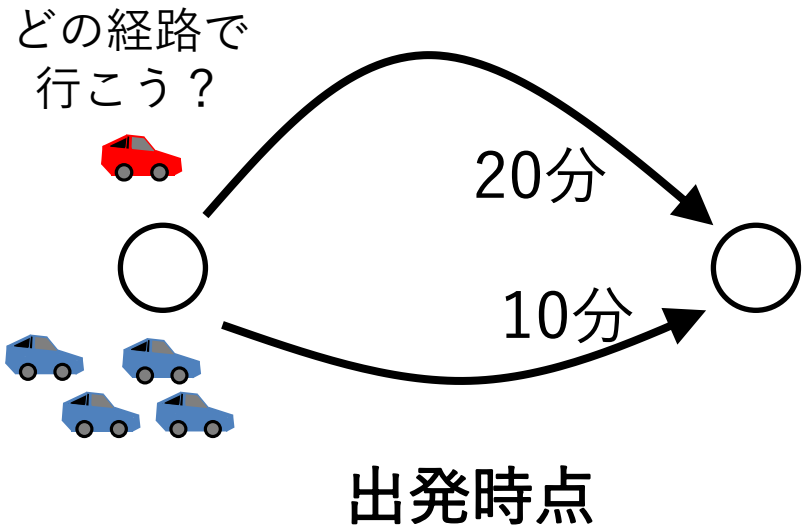
- Wardrop原則：代表的な経路選択ルール
  - Dynamic User Equilibrium (DUE): 各車両が自車にとって最小コスト経路を選択する．自己利益最大化
  - Dynamic System Optimum (DSO): 各車両が，システム全体のコストを最小化するよう経路を選択する．社会利益最大化

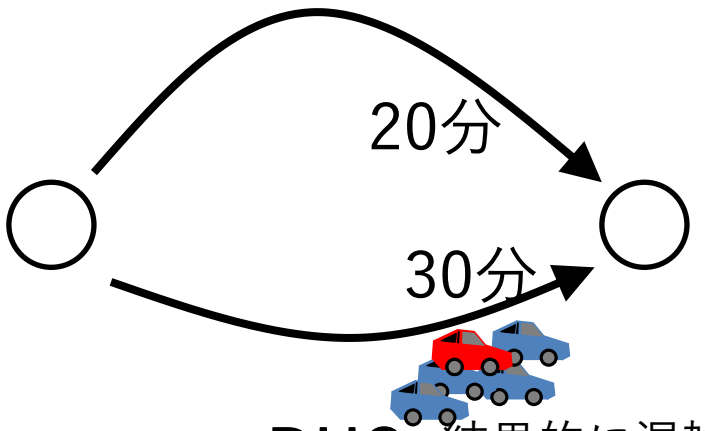
- DUE・DSOは理論的に明らかになっていない点も多く，取り扱いは非常に難しい
  - 経路を選択する時点では，経路のコストは将来の話なのでわからない
  - 車両同士の相互作用が複雑で，取るべき経路を求めるのが困難
  - 一般的な場合には，解の唯一性は否定済みで，確実な解法も知られていない

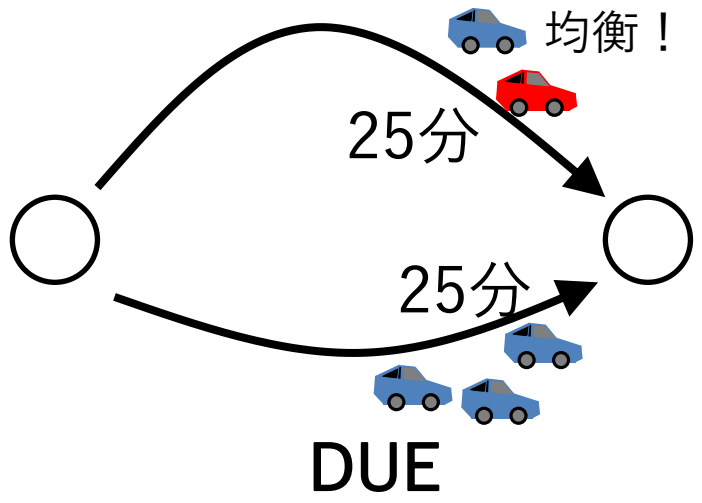
動的交通量配分：DUO

- 理論的厳密性を妥協し，取り扱いを容易にした経路選択ルールもある

- Dynamic User Optimum (DUO): 各車両が自車にとって今の瞬間コストが最小の経路を選択する
  - c.f., DUE: 各車両が自車にとって結果的なコストが最小の経路を選択する
  - 今の瞬間コスト：現時点の各リンクの状態を足し合わせた経路コスト
  - 近視眼的な選択．例：今の交通情報だけをみて経路を決める．非現実的ではない
  - 計算は比較的容易



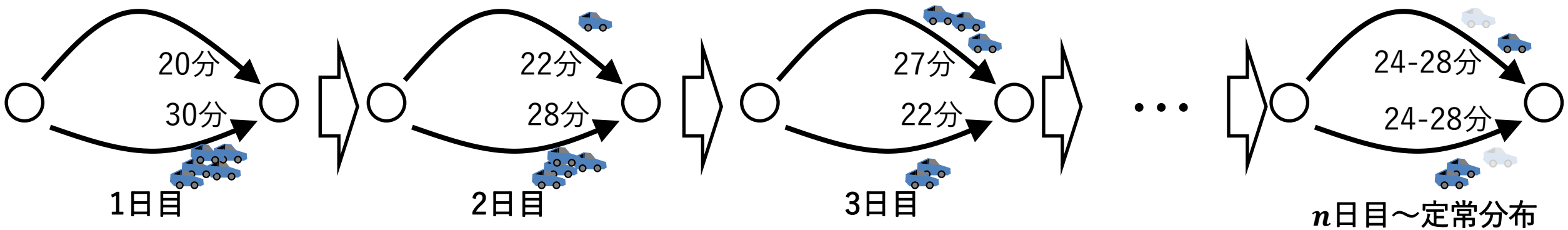




動的交通量配分：疑似DUE・DSO

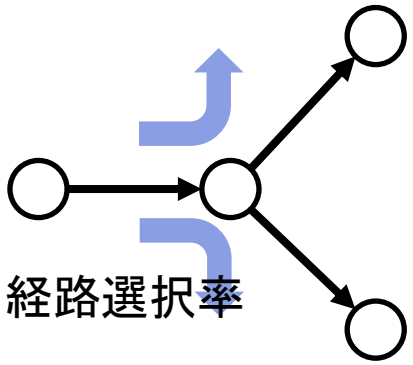
- DUOはある程度合理的だが近視眼的．DUE・DSO的な配分も欲しい

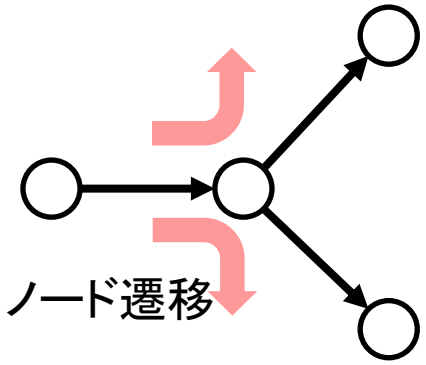
- 井料のマルコフ連鎖day-to-day配分[1,2]
  - 毎日p%の人が経路をより低コストのものに変えるという過程を繰り返す
  - これを，各車両の選択経路を並べたものを状態としたマルコフ連鎖とみなす
  - マルコフ連鎖は緩い条件のもとで唯一の定常分布に収束することが保証されている
  - 収束先をDUEの代替物とみなす
  - 外部性をコストに含めればDSO的な解も求められる[3,4]
  - 仮に疑似DUE・疑似DSOと呼ぶことにする（しかるべき人に正式に命名してほしい）

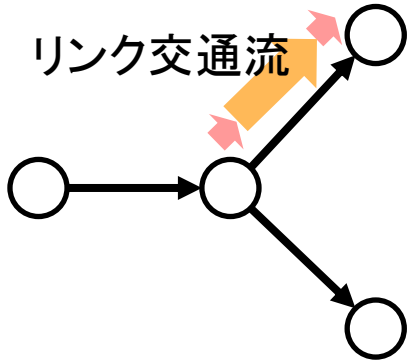


[1] 石原 雅晃, 井料 隆雅. 2015. マルコフ連鎖による動的ネットワーク交通量配分. 土木学会論文集D3（土木計画学） 71, 5: I503–I509.
[2] Takamasa Iryo, David Watling, and Martin Hazelton. 2024. Estimating Markov Chain Mixing Times: Convergence Rate Towards Equilibrium of a Stochastic Process Traffic Assignment Model. Transportation Science 58, 6: 1168–1192.
[3] Behzad Bamdad Mehrabani, Jakob Erdmann, Luca Sgambi, Seyedehsan Seyedabrishami, and Maaike Snelder. 2023. A multiclass simulation-based dynamic traffic assignment model for mixed traffic flow of connected and autonomous vehicles and human-driven vehicles. Transportmetrica A: Transport Science 21, 2.
[4] Koki Satsukawa, Kentaro Wada, and David Watling. 2022. Dynamic system optimal traffic assignment with atomic users: Convergence and stability. Transportation Research Part B: Methodological 155: 188–209.

ネットワーク交通流理論の一般的計算枠組み





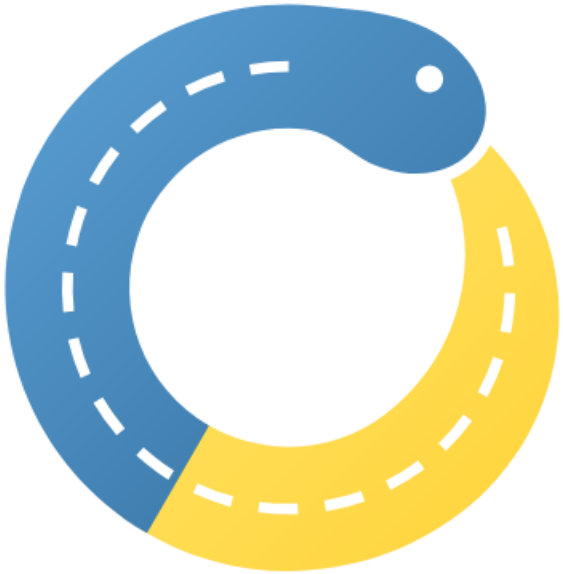


1. 初期化，シナリオ定義
   - 時間帯別OD需要，ネットワーク構造，交通流モデルパラメータ等
2. 各ノードで経路選択モデルを計算
   - DUO，所与の経路選択率など
3. 各ノードでリンク間交通流伝播を計算
   - ノードモデル（今回の講義では略）
4. 各リンクでリンク内交通流モデルを計算
   - メソNewell単純追従モデルなど
5. 時刻をインクリメント
6. 最終時刻到達で終了
   - 疑似DUE・DSOの場合，シミュレーション自体を繰り返し計算

- 理論はわかったとしても，これを自分で組み合わせて計算するのはかなり大変
- 既存の交通シミュレータはミクロモデルやアクティビティモデルが多く，純粋なマクロ交通シミュ規模は少ない
  - 都市交通流だけを見たいときに，SUMO・MATSimは大げさ

- 研究で使うにはどうすればいいのか？使いやすい効率的なものが欲しい！

- 基礎
  - 交通流理論
  - 動的交通量配分
  - メソ交通流モデル・シミュレーション
  - 疑似DUE・DSO
- 応用・実践 1
  - メソ交通流シミュレータUXsim
- 応用・実践 2
  - 微分可能交通流シミュレータUNsim
- OSS活動の実践で得られた知見



UXsim
Python traffic simulator

- UXsim: 瀬尾によるPython向けのネットワーク交通流シミュレータ
- オープンソースソフトウェア（OSS）としてGitHubで公開
  - https://github.com/toruseo/UXsim
  - 他人による利用論文数（arXivなど含む）：25本/3年
- Pythonコーディングで使用することを前提とし，可能な限りシンプル・汎用的・効率的に実装したもの

Seo, T. UXsim: lightweight mesoscopic traffic flow simulator in pure Python. Journal of Open Source Software, Vol. 10, No. 106, p. 7617, 2025.

UXsimのシナリオ定義


```python
from uxsim import *

# シミュレーション本体の定義
# 単位は全て秒とメートル
W = World(
    name="",    # シナリオ名
    deltan=5,   # 車両集計単位
    tmax=1200,  # シミュレーション時間
    print_mode=1, save_mode=0, show_mode=0,    # オプション
    random_seed=0    # ランダムシード
)

# Define the scenario
W.addNode("orig1", 0, 0) # ノードの作成
W.addNode("orig2", 0, 2)
W.addNode("merge", 1, 1)
W.addNode("dest", 2, 1)
W.addLink("link1", "orig1", "merge", length=1000, free_flow_speed=20, jam_density=0.2, merge_priority=0.5) # リンクの作成
W.addLink("link2", "orig2", "merge", length=1000, free_flow_speed=20, jam_density=0.2, merge_priority=2)
W.addLink("link3", "merge", "dest", length=1000, free_flow_speed=20, jam_density=0.2)
W.adddemand("orig1", "dest", 0, 1000, 0.4) # 交通需要の作成
W.adddemand("orig2", "dest", 500, 1000, 0.6)

# シミュレーション実行
W.exec_simulation()
```

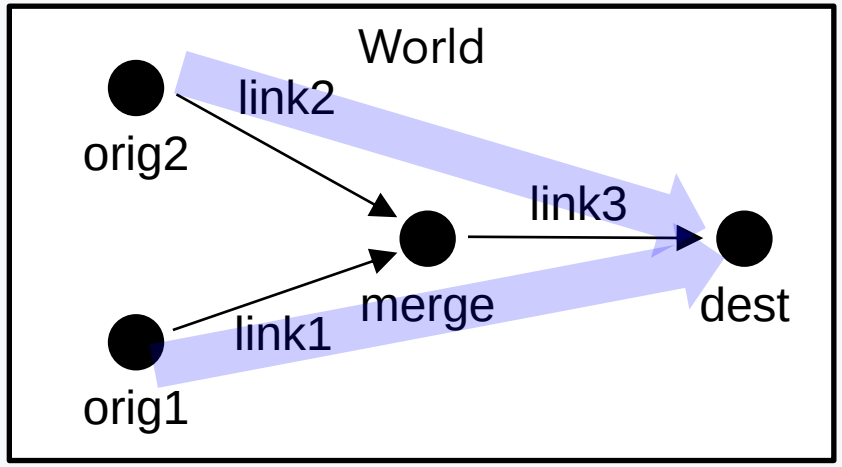
World: シミュレーション基本設定

addNode: 位置(0,0)に"orig1"というノードを作る

addLink: ノード"orig1"と"merge"の間に"link1"というリンクを作る

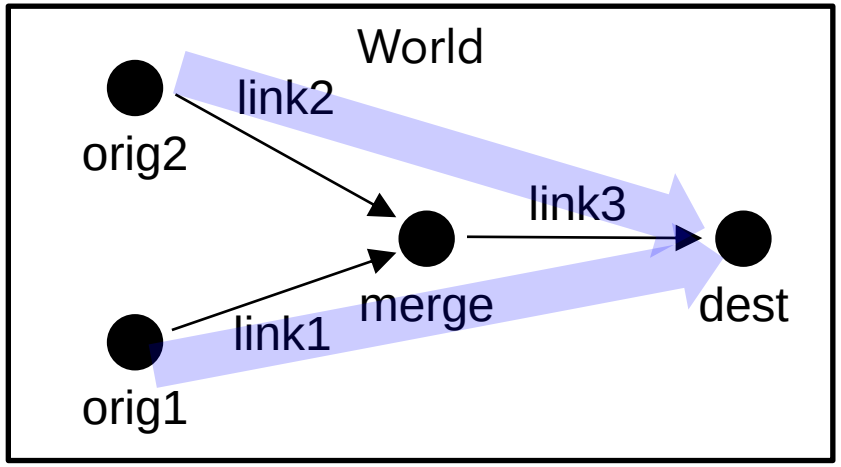
adddemand: ノード"orig1"と"dest"の間に交通需要を設定

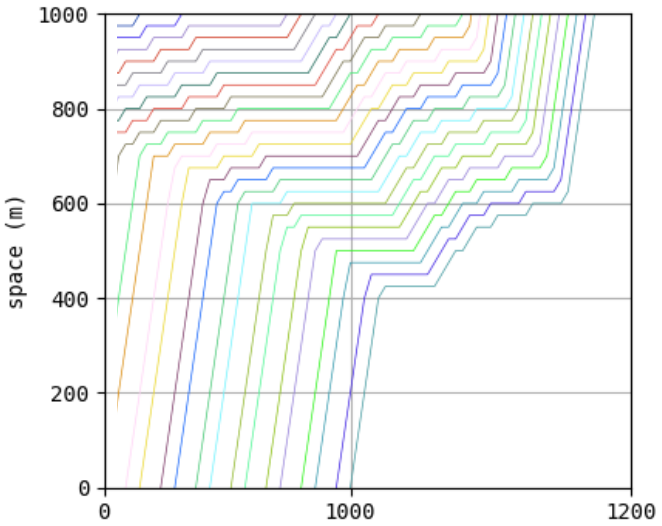
exec_simulation: シミュレーション実行



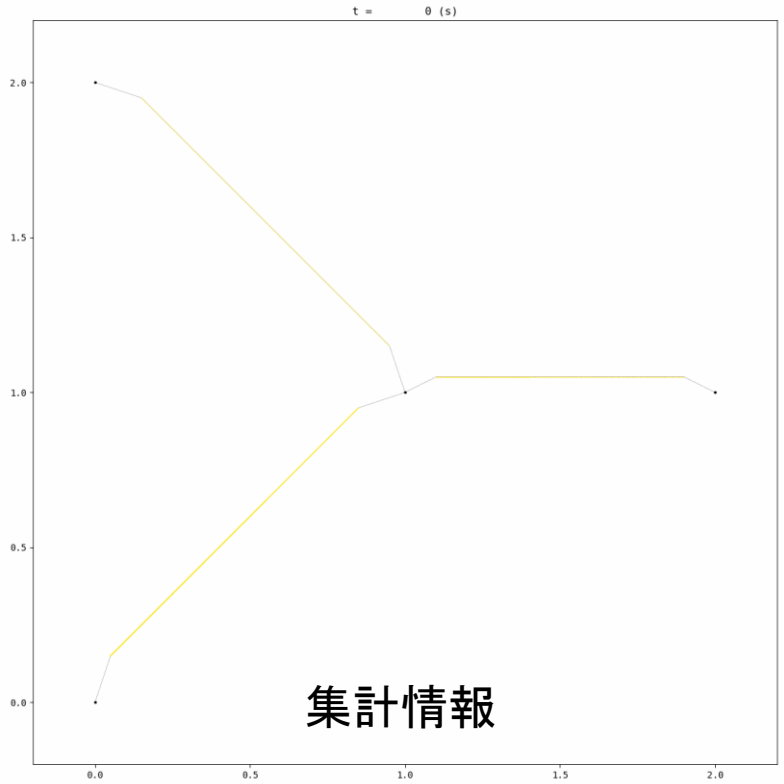
UXsimの結果分析

- 内部データはPythonで自由にアクセス可能
- pandas.DataframeやCSVでのエクスポート
- デフォルトの可視化機能
  - リンク交通状態
  - エリア交通状態
  - ネットワーク交通状況アニメーション





リンク内車両軌跡

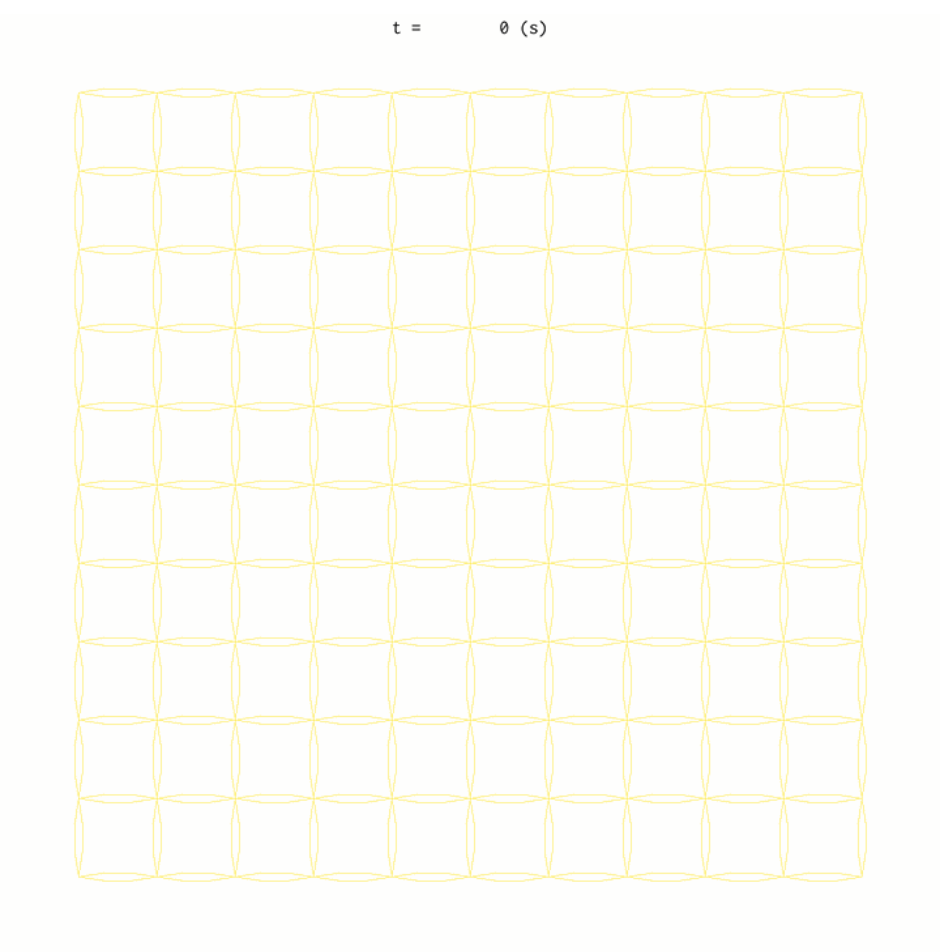


集計情報



サンプル軌跡

UXsimの計算例

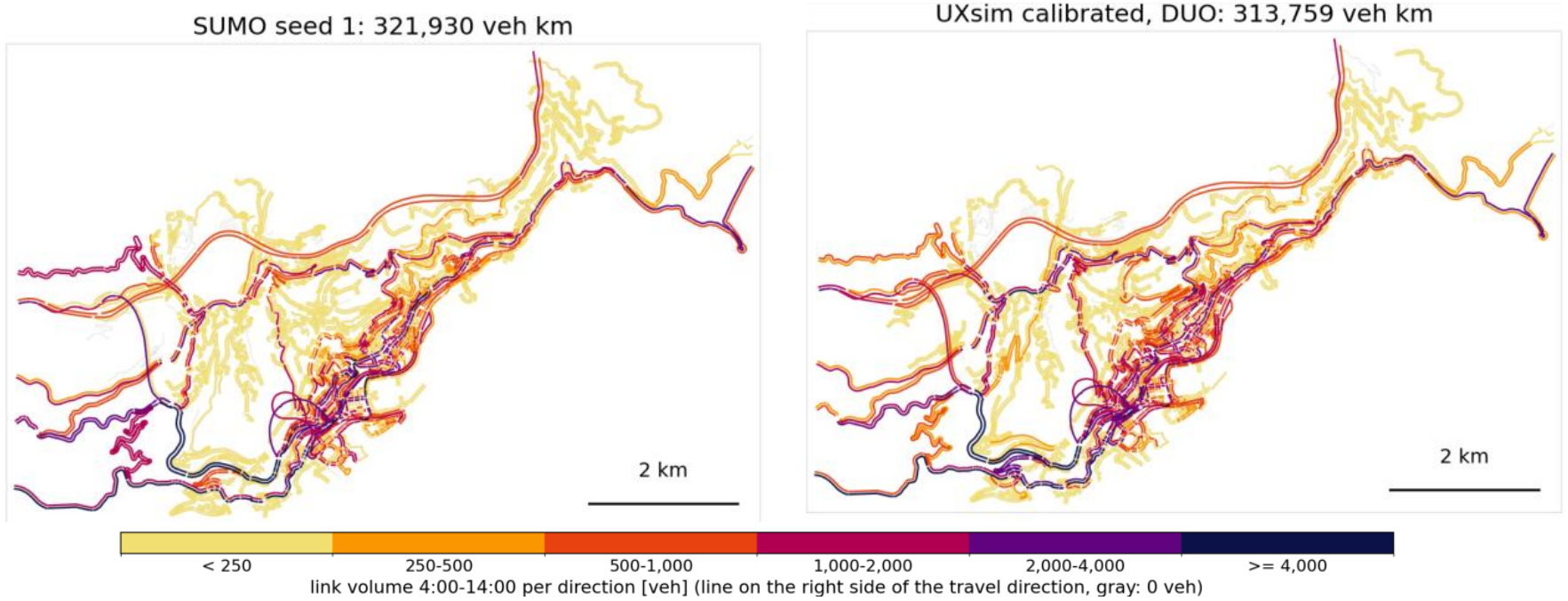




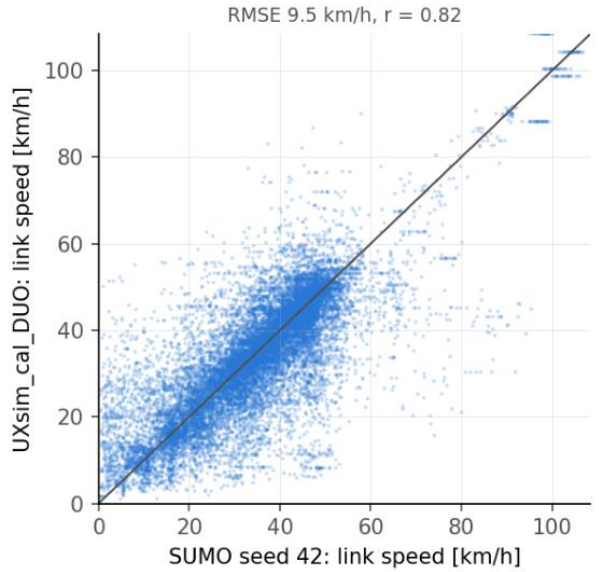
- シナリオ：2時間，6万台，10km x 10kmのグリッドネットワーク
- コード：約50行
- 計算時間：約1秒（内部C++エンジン使用の場合）

UXsimの計算例

- ミクロ交通シミュSUMOの定番シナリオであるMoSTシナリオ[1]の再現実験
  - 対象は自動車のみ
  - シナリオ規模概数：10時間，5万台，10km x 6kmエリア，1000ノード，2300リンク
  - 調整：ネットワーク形状はメソに適切になるよう自動整形，リンク自由流速度はSUMO実効最大速度から設定，交差点交通容量は総旅行時間が一致するよう調整
  - 計算時間：SUMO=30分，UXsim=5秒



交通量分布



リンク速度比較

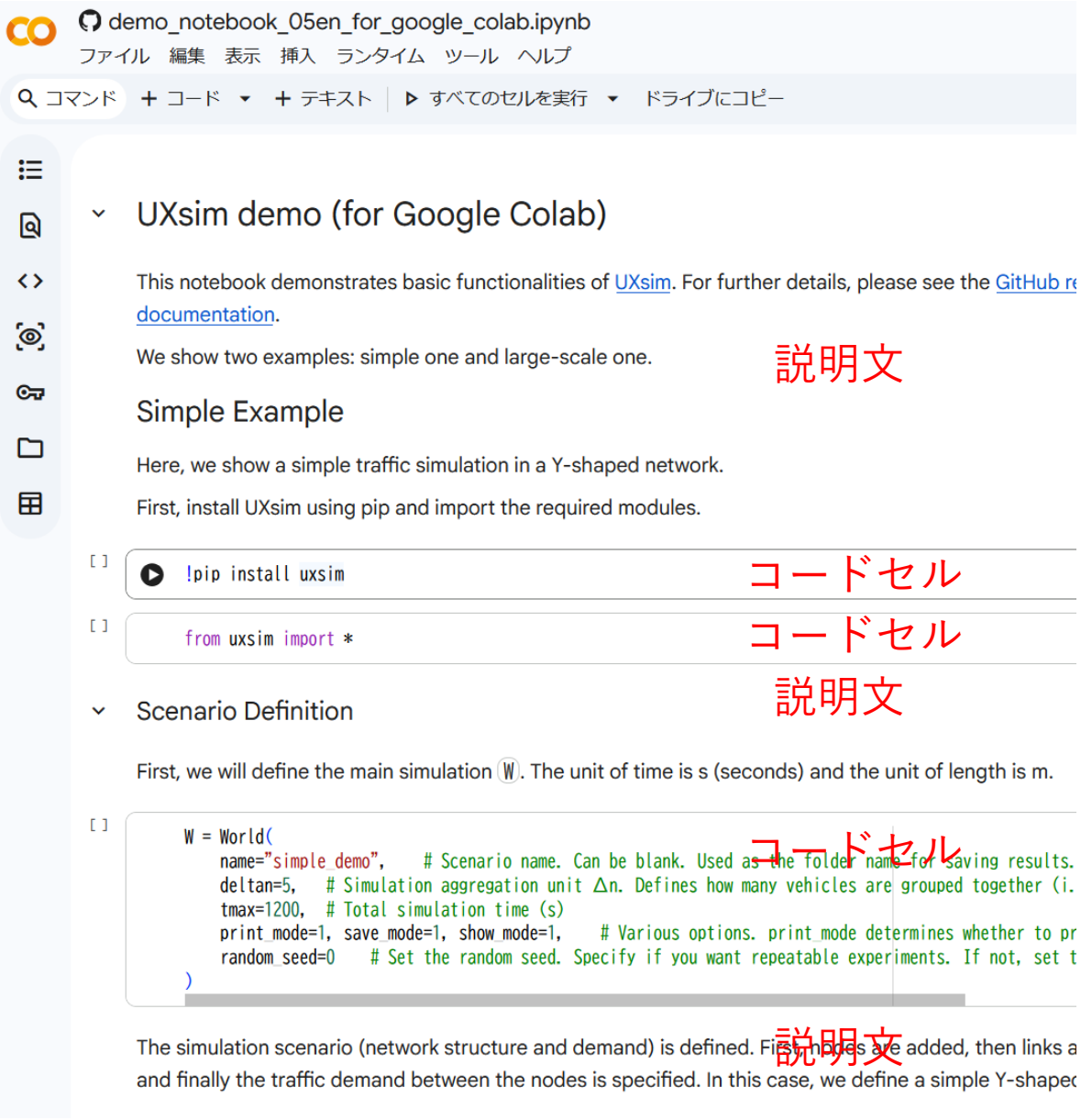
[1] Lara Codeca and Jérôme Härri. 2018. Monaco SUMO traffic (MoST) scenario: A 3D mobility scenario for cooperative ITS. EPiC series in engineering 2: 43–55.

演習：ネットワーク交通流シミュレーション入門



1. ノートPC・Googleアカウントのある人向け
2. https://github.com/toruseo/UXsimにアクセス
   - “uxsim github”でネット検索すれば上の方に出てくるはず
3. “Open in Colab”ボタンをクリック

演習：ネットワーク交通流シミュレーション入門



- Google Colabの使い方
  - Jupyter Notebookと同じ
  - 説明文とコードセルからなる
  - コードセルで再生ボタン（or Shift+Enter）を押すとそのセルのコードが実行される
  - 結果はその直下に表示される
  - 上から順番に実行し，一連のプログラムを実行できる

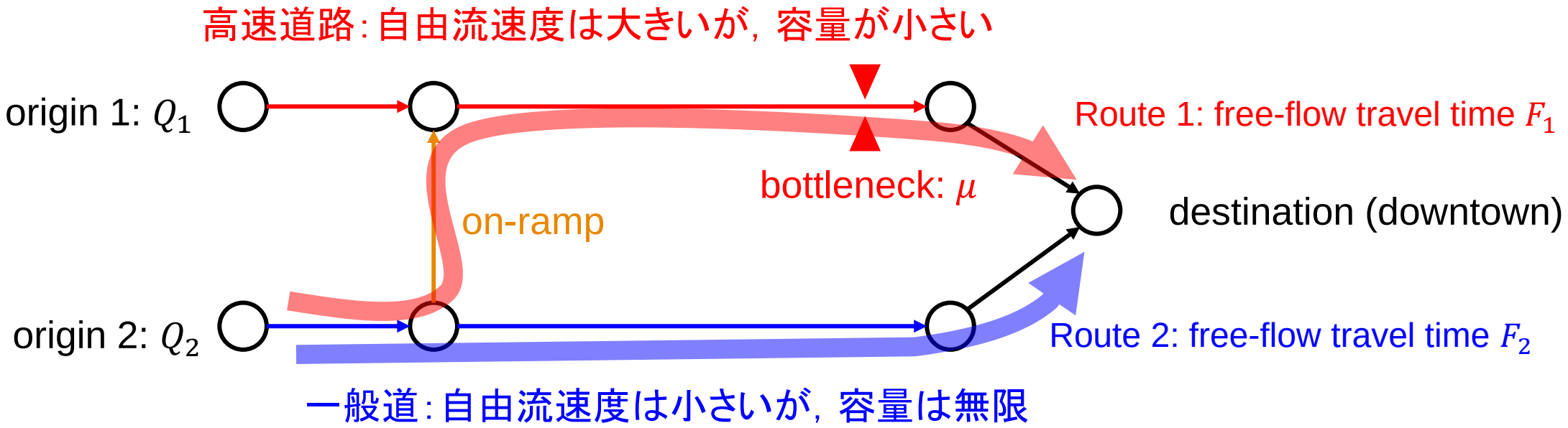
UXsimによるDTA

- 経路選択モデルはデフォルトはDUO．繰り返し計算不要で一発で計算できるため

- それに加えて，疑似DUE・DSOも計算可能

- 繰り返し計算に若干時間がかかるので，本講義では計算結果だけ示す
  - 詳細：https://toruseo.jp/UXsim/docs/notebooks/demo_notebook_09en_dynamic_traffic_assignment.html

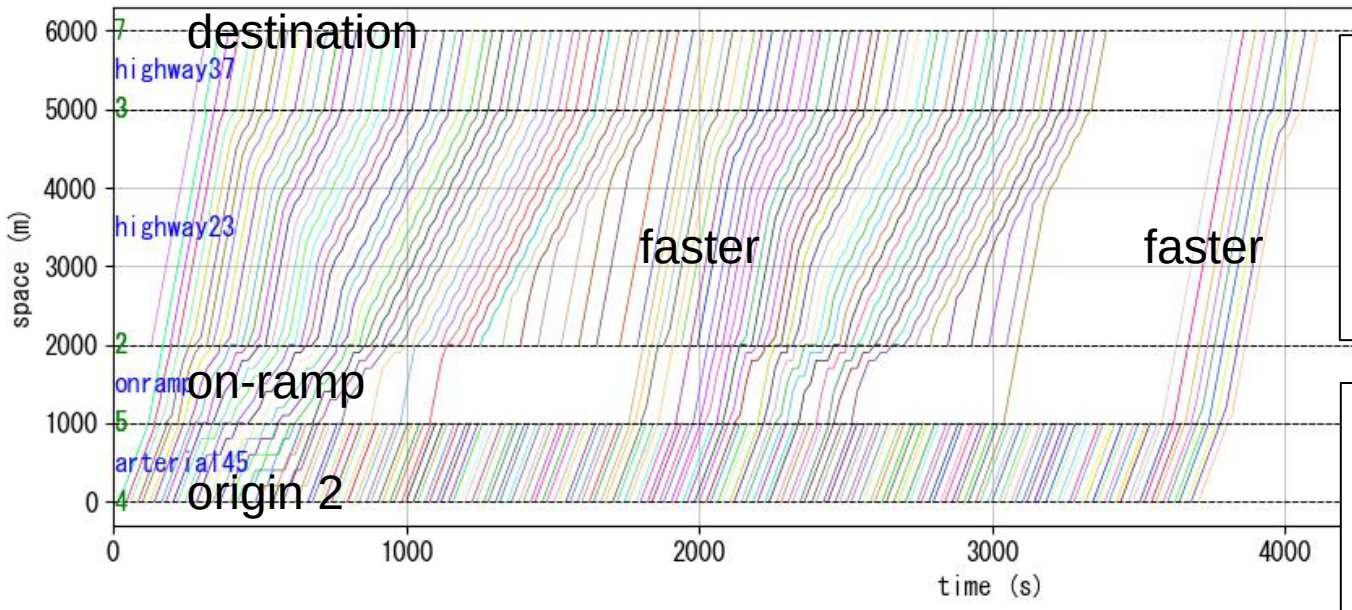
DTAの簡単な例



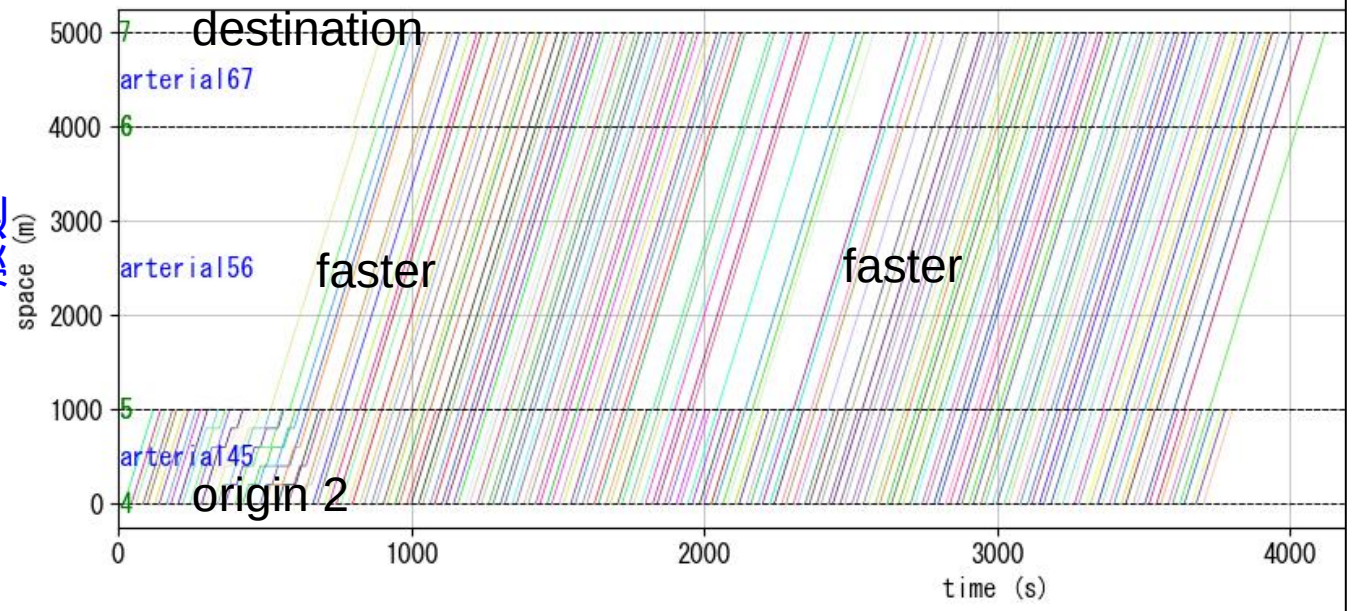
- 高速道路と一般道が平行になったネットワークでの通勤交通シナリオ
- origin 1からの需要: Q_1 (veh/h)
  - 高速道路しかつかえない
- origin 2からの需要: Q_2 (veh/h)
  - Route 1: on-rampで高速道路に乗り換える
  - Route 2: 一般道を使い続ける
- 高速道路ボトルネック容量: μ (veh/h)
- 自由流旅行時間は高速道路が短い: $F_1 < F_2$
- 経路選択モデルごとに何が起きるか？

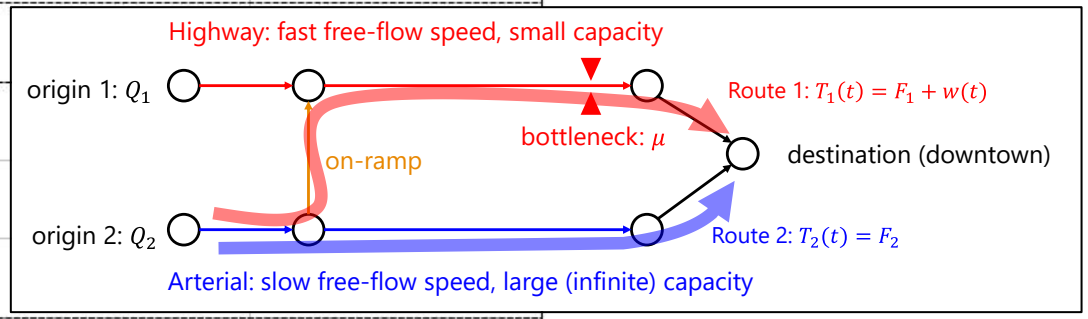
DTAの簡単な例：DUO

高速道路



一般道

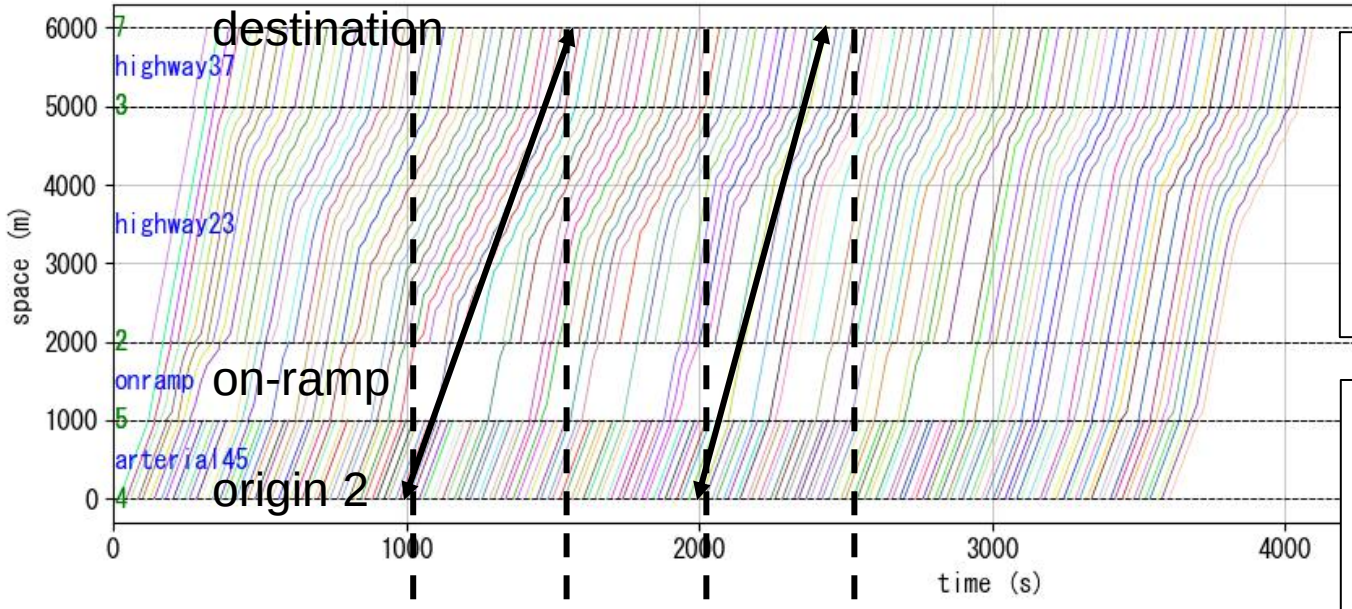




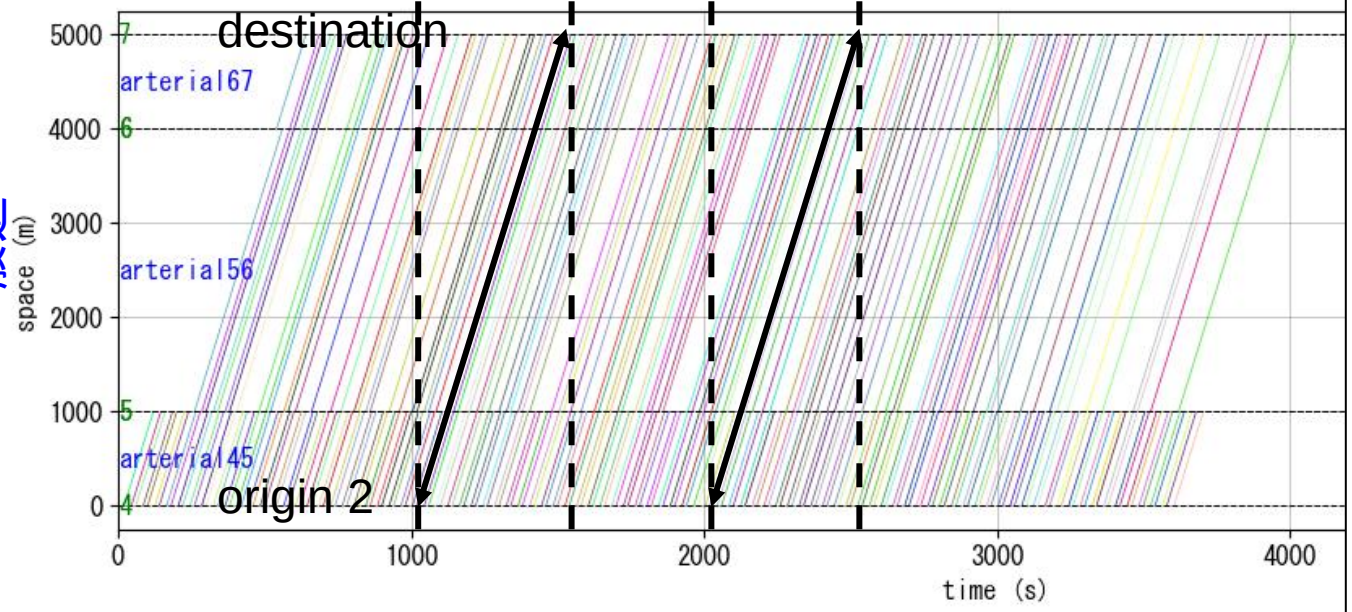
- 2経路の車両軌跡図
- DUOの近視眼的経路選択
  - 最初はほぼすべての車両が高速道路を選択
  - その結果渋滞が起き，一般道に逃げ出す
  - その結果渋滞が無くなり，高速道路に集中
  - （以下ループ）

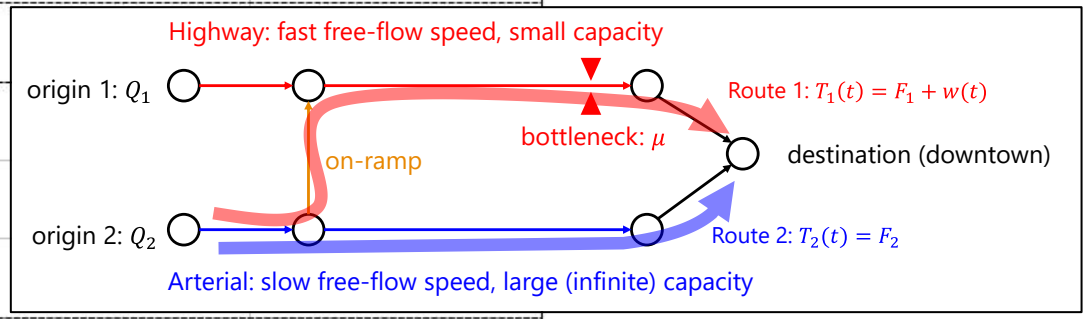
DTAの簡単な例：疑似DUE

高速道路



一般道

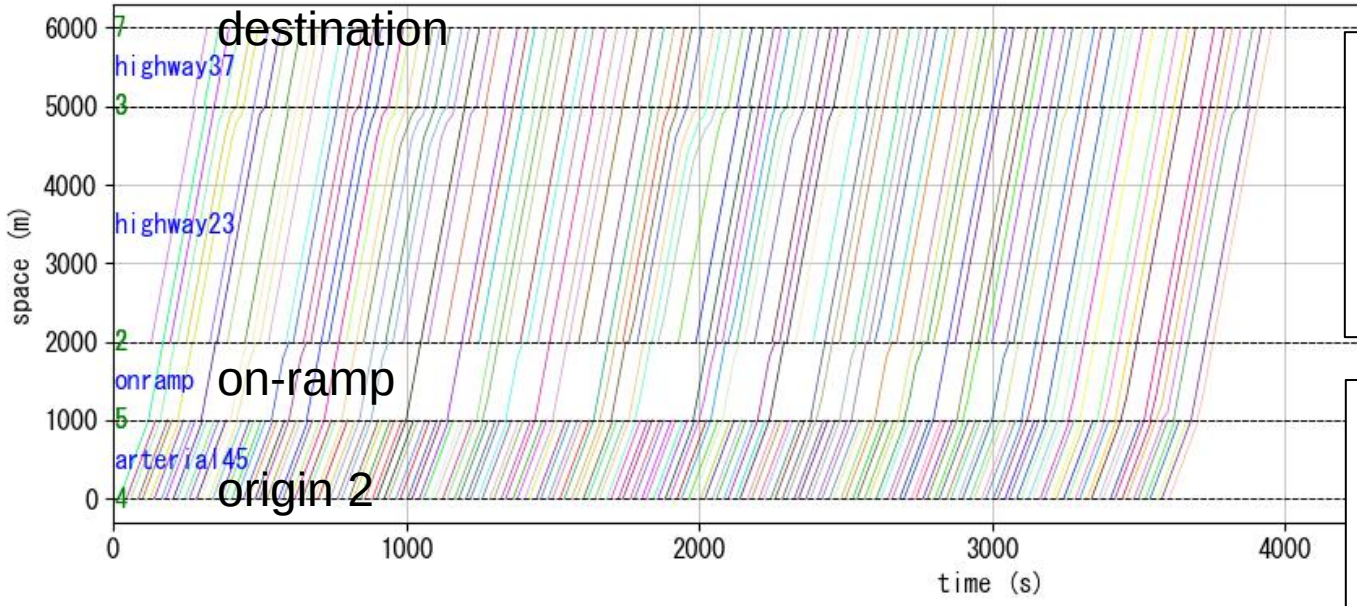




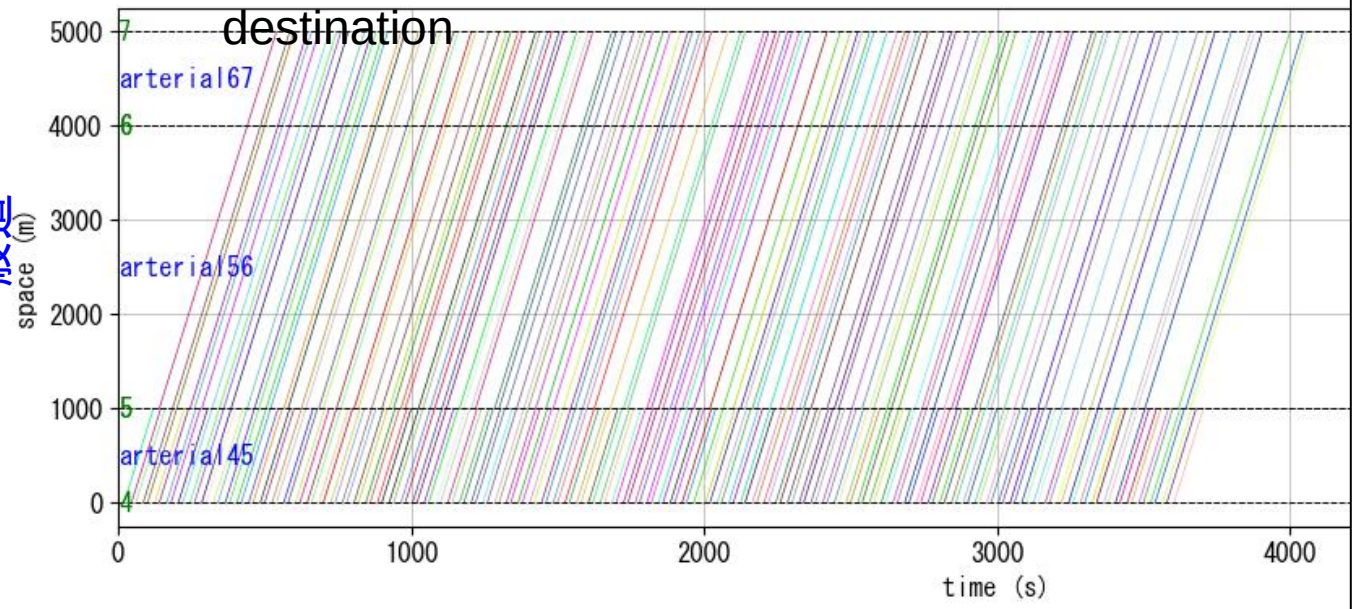
- 定常分布に落ち着いたと思われる50日目の結果
- 時刻1000sに出発する場合は，どちらの経路も旅行時間が同じ
  - 次の日に経路は変えない
- 時刻2000sに出発する場合は，高速道路の方がわずかに旅行時間が短い
  - 次の日に経路が変わる可能性
- 全体としてはバランスが取れていて，日々の変動はあるものの定常にみえる

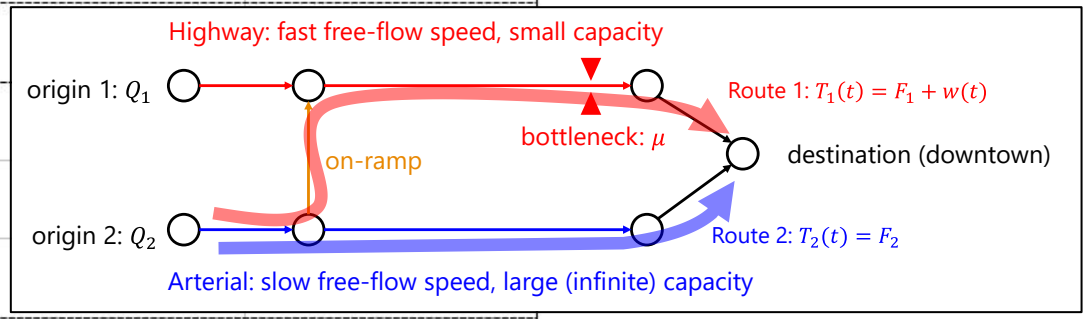
DTAの簡単な例：疑似DSO

高速道路



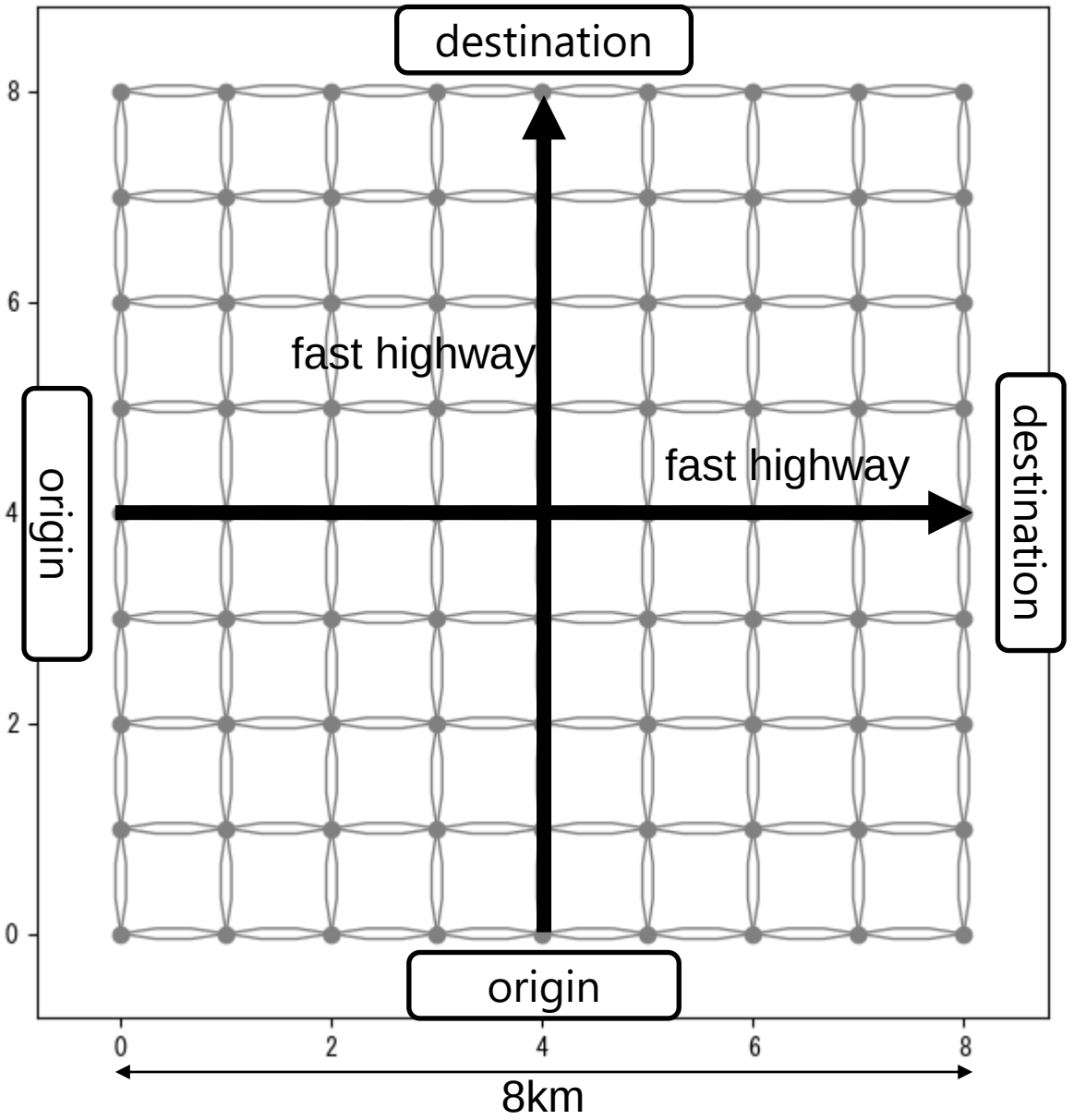
一般道





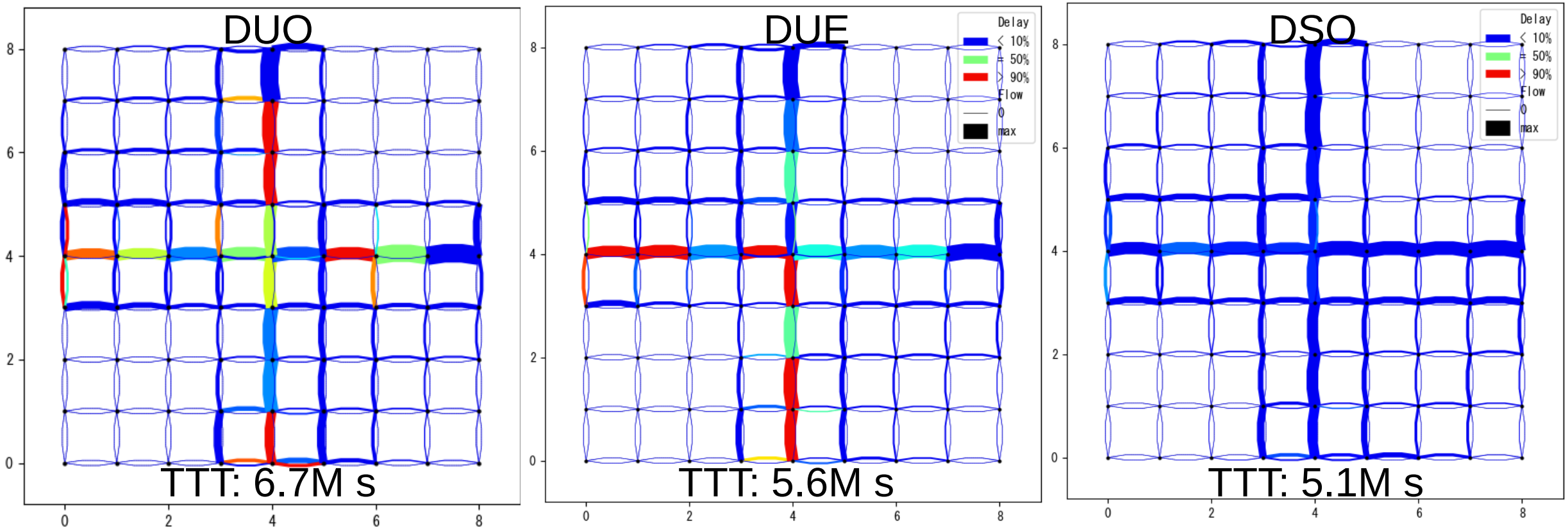
- 疑似DSO解では渋滞がほとんどなく，高速道路の容量をほぼちょうど使い切る配分になる
- 総旅行時間比較
  - DUO: 3.6M s
  - 疑似DUE: 3.2M s
  - 疑似DSO: 2.9M s

DTAの大規模シナリオ



- グリッドネットワーク
  - 8km x 8km
  - 速いが少ない高速道路と，遅いが多い一般道
  - 1.5時間で6840台の需要

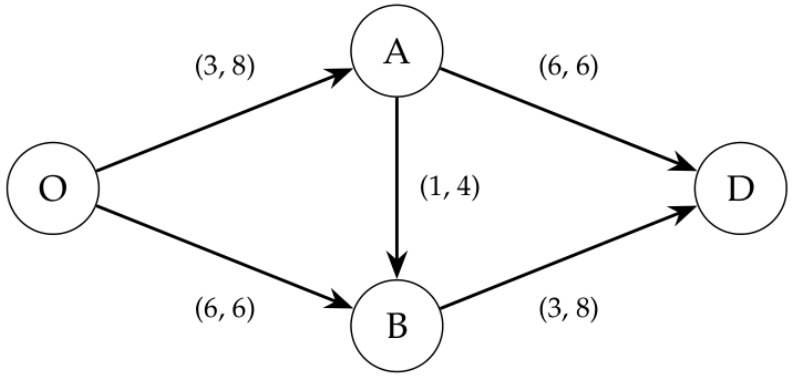
DTAの大規模シナリオ

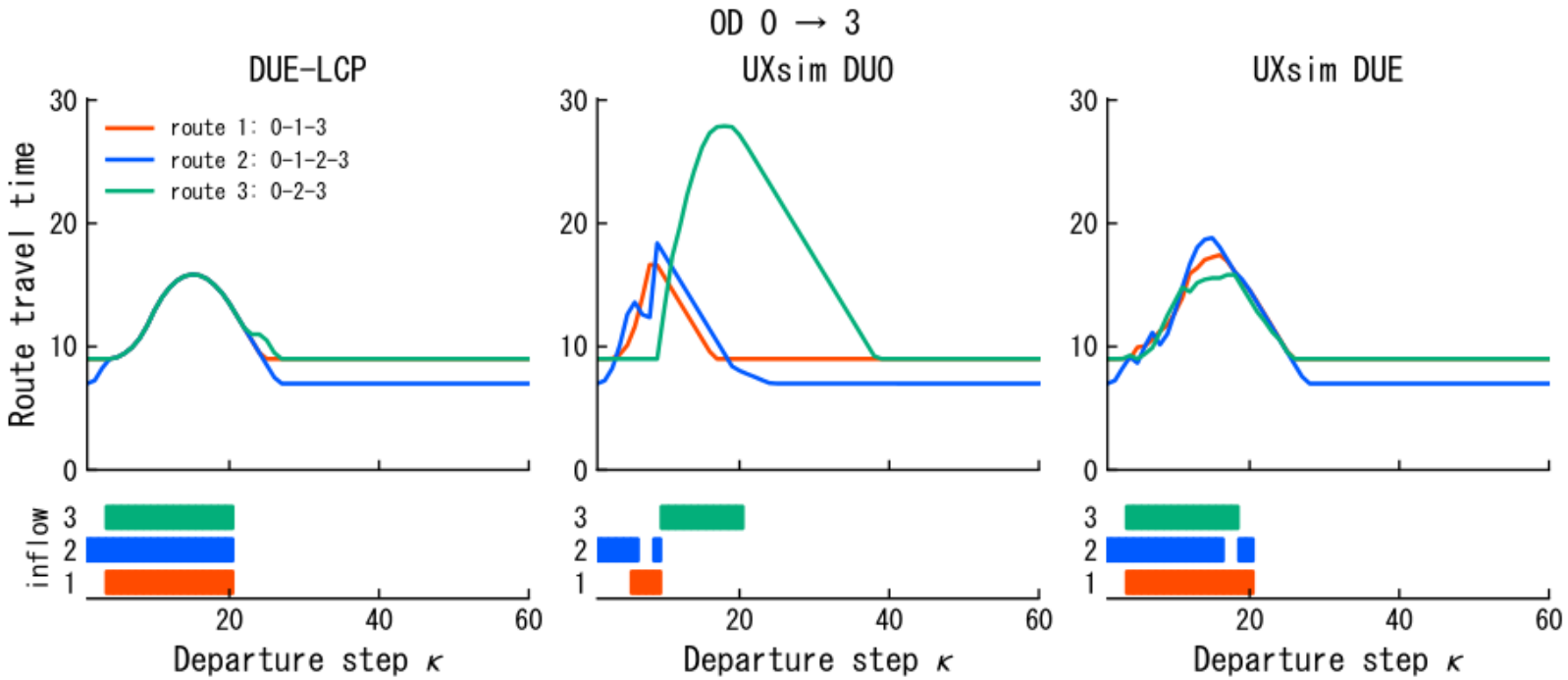


- リンク毎の平均遅れのプロット
  - DUO：高速道路に車両が集中する
  - 疑似DUE：一部高速道路に車両が集中するが，周りの一般道もある程度使われる
  - 疑似DSO：ほぼ渋滞がない．DUOより25%，疑似DUEより10%効率的

疑似DUEと真正DUEの比較

- スタートアップ研究開発型企業「くるみ堂」により，疑似DUEがどれくらい真のDUEに近いのか検証[1]
- 小規模ネットワークでは，相当程度一致している

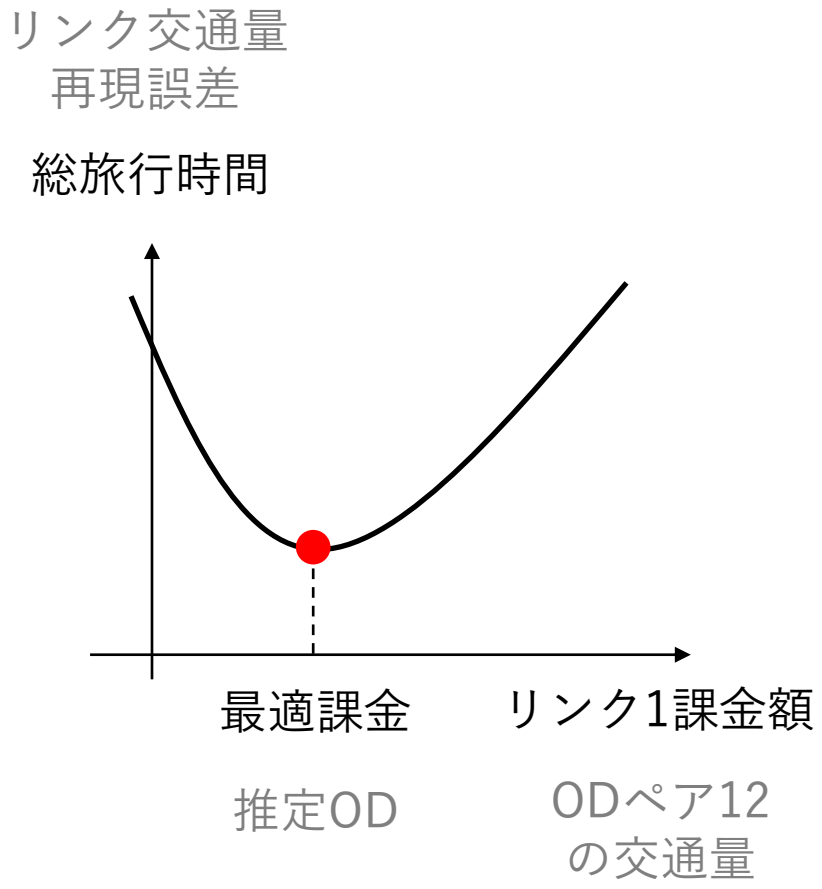




[1] https://github.com/takala4/route-choice-due-lcp

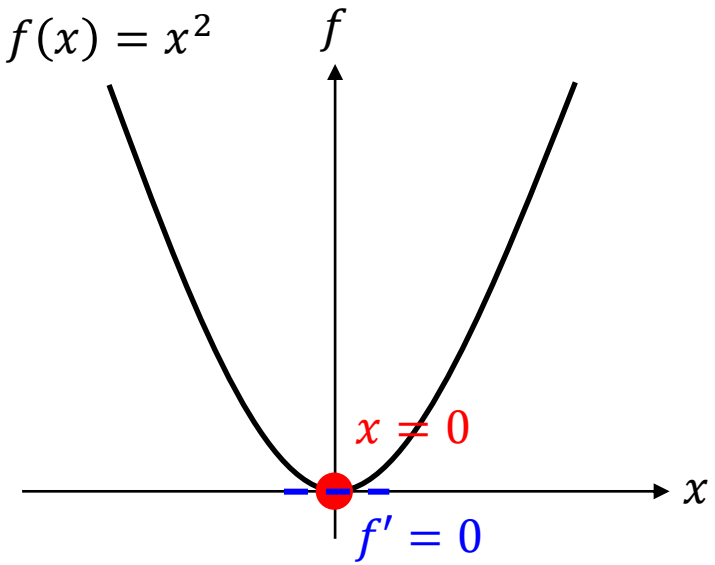
- 基礎
  - 交通流理論
  - 動的交通量配分
  - メソ交通流モデル・シミュレーション
  - 疑似DUE・DSO
- 応用・実践 1
  - メソ交通流シミュレータUXsim
- 応用・実践 2
  - 微分可能交通流シミュレータUNsim
- OSS活動の実践で得られた知見

交通シミュレーションによる最適化



- 交通シミュ：動的な交通流を，ネットワークでの経路選択・渋滞を考慮して計算するものを考える
- 最適化のツールとしての応用
  - 交通システム最適化：総旅行時間を最小化するような，リンク混雑課金を求める
  - OD推定：現況再現誤差を最小化するような，OD交通量を求める
  - パラメータ較正：現況再現誤差を最小化するような，交通容量を求める
- 課題：最適化対象の変数が多くなるほど難しくなる場合がある＝次元の呪い

最適化問題の基礎（c.f. 土木計画学）



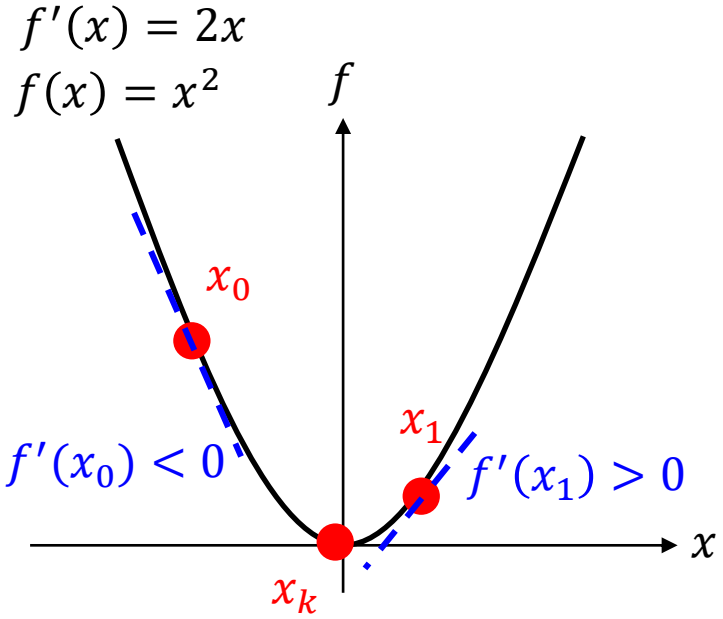
- 最適化問題：目的関数を最小化する決定変数の値を求めよ

$$\min_{\boldsymbol{x}} f(\boldsymbol{x})$$

  - f: 目的関数
  - $\boldsymbol{x}$: 決定変数ベクトル
  - $|\boldsymbol{x}|$: 決定変数の数，問題の次元

- 例：$\min_{x} x^2$
  - 人間が見れば解はすぐわかる
    - $x = 0$
  - 厳密に求める・証明するには？
    - **傾き＝0** ⇔ 微分$= \frac{df}{dx}(x) = f'(x) = 0$ （かつfが凸）
  - コンピュータで自動的に計算するには？
    「$f' = 0$を解け」という命令は自動的にはやってくれない！
    - 繰り返しアルゴリズム

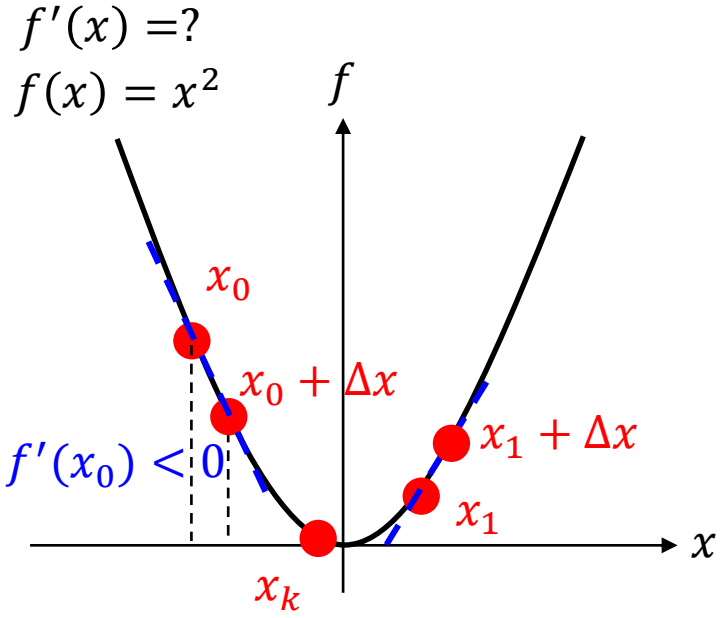
最適化問題の求解アルゴリズム

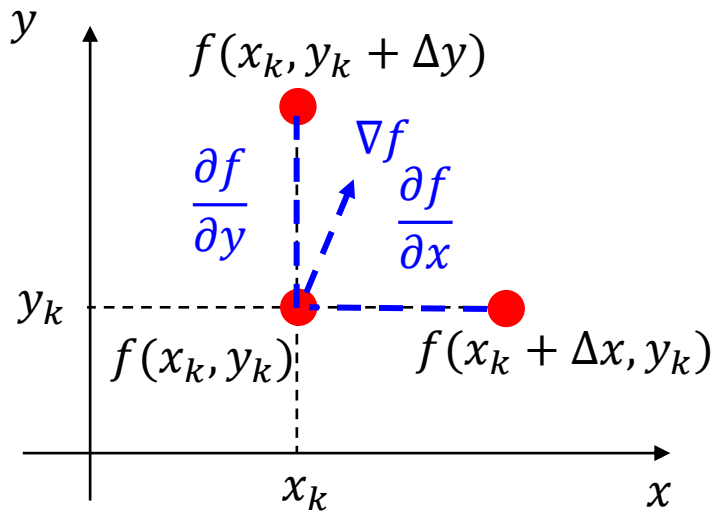


- 微分f'の式が**解析的にわかっている**場合
  1. 適当な初期値x_0を定める
  2. 点x_0の微分値$f'(x_0)$を求める
  3. $f'(x_0)$が負ならx_0より大きい点x_1に進む.
     $f'(x_0)$が正ならx_0より小さい点x_1に進む.
     $f'(x_0)$が十分小さいならx_0が解
  4. 点x_1の微分値$f'(x_1)$を求める
  5. （繰り返し）
  6. $f'(x_k)$が十分小さくなったらx_kが解

- 代表的なアルゴリズム：最急降下法，ニュートン法，Frank-Wolfe法

最適化問題の求解アルゴリズム





- 微分f'の式が**解析的にわからない**場合
- 決定変数xが一次元の場合
  1. $f(x_k)$の値を求める
  2. 微小幅Δxに対し，$f(x_k + \Delta x)$の値を求める
  3. 数値差分$\frac{f(x_k+\Delta x)-f(x_k)}{\Delta x}$で$f'(x_k)$を近似する
  4. あとは先ほどのアルゴリズムの手順に従う
  - 点x_k毎に2回関数fを計算する．解析的に求められる場合はf'を1回計算するだけなので，比べると約2倍の計算回数
  - しかも，求まるf'は近似なので精度が悪い
- 決定変数$\boldsymbol{x}$が二次元$\boldsymbol{x} = (x, y)$の場合
  - $f(x_k, y_k), f(x_k + \Delta x, y_k), f(x_k, y_k + \Delta y)$を計算し，差分から$\nabla f$を近似
  - 約3倍の計算回数
- 決定変数が10000次元の場合
  - 約10000倍の計算回数
  - ∇fの近似精度も悪い

最適化問題の求解

- 決定変数がN個の最適化問題では，**目的関数の微分が解析的にわからない場合は，わかる場合のN倍オーダーの計算時間**がかかる
  - つまり，繰返し計算の1回ごとにシミュレーションを少しずつ条件を変えてN回回す必要がある
  - さらに，数値的に求める勾配の精度はかなり悪いので，繰り返し計算の回数も増える or どうやっても良い解に到達できない
  - 大規模な問題では実質的に不可能といってよい

- 微分を解析的に求める代表的方法：数式をひたすら手計算する
  - $f(x,y,z) = x^4 + 5xy^2 + 2z + 4 \quad \Rightarrow \quad \frac{\partial f}{\partial x} = 4x^3 + 5y^2,\ \frac{\partial f}{\partial y} = \cdots$

- 交通シミュで解析微分するには，シミュロジックを数式で書き下してひたすら手計算すればできるかもしれないが．．．？
  - メソのUXsimは離散なので原則として不可
  - マクロであれば理論上はできる可能性があるが，巨大ロジックを数式表記して，さらに微分するのは恐ろしくしんどい
  - シナリオ毎に違う数式になる
  - 複雑なif文，for文は微分可能な数式で書けない
  - **常識的に考えると無理なので，既存の交通シミュ最適化は解析微分・数値微分はせずに，精度に限界のある近似最適化がメイン．遺伝的アルゴリズム，SPSA，メタモデル・サロゲートモデル**

微分可能シミュレーション

- 微分可能シミュレーション：シミュレーションを自動的に微分する技術
  - 昔は職人技だったが，近年のニューラルネットを微分して学習する技術が一般化され，汎用化した

- シミュロジックを単純な要素に分解し計算グラフに変換し，そのつながりに連鎖律 $\left(\frac{df}{dx} = \frac{df}{dg}\frac{dg}{dx}\right)$ を適用
  - 本質的に微分不可能な要素が含まれている場合は適用不可
  - 折れ線など，単純な微分不能点であれば劣勾配で近似可能

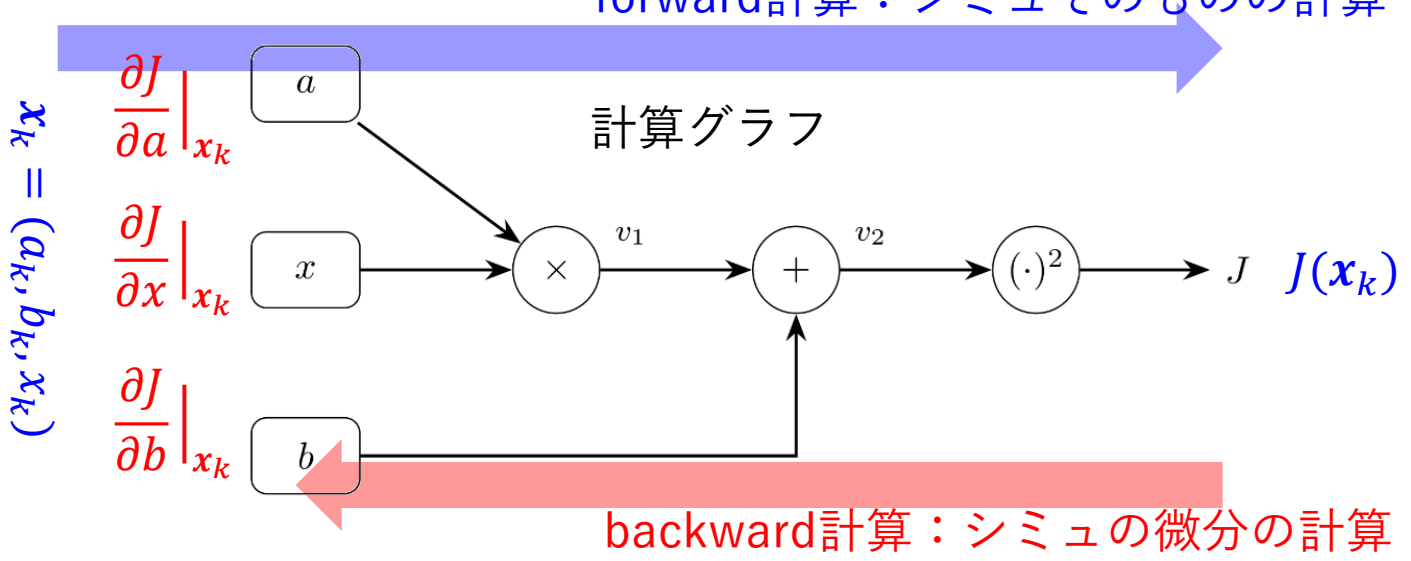
- シミュレーション1回の計算時間の2~5倍の時間で，全ての変数に対する微分値が計算できる
  - つまり，シミュ関数を$J(x)$と表記すると，$\nabla J|_{x=x_0}$ を一発で計算できる
  - シミュレーション変数数に依存しない！近似ではなく厳密に求められる！

シミュレーションの数式・コード表記
入力$\boldsymbol{x} = (a, b, x)$から出力$J$を求める

$$J = (ax + b)^2$$

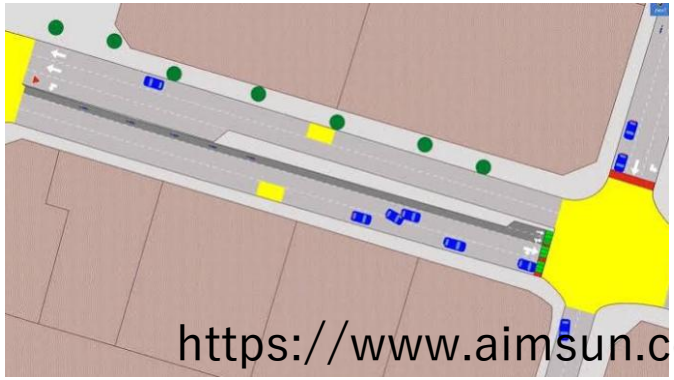
```
def compute_J(xx):
    a = xx[0]
    b = xx[1]
    x = xx[2]
    return (a*x + b)**2
```

変換



微分可能交通流シミュレーション

- 交通流シミュレーションを微分可能にするには？
  - ミクロ・メソだと，車両と経路選択が離散的
  - 大量の複雑なif文，for文
  - 離散変数や，複雑なif文，for文は微分できない！
  - 情報分野の研究で，ミクロシミュを（かなり大胆に）近似して微分した例[1]が提案されてはいるが，経路選択が無いなど本格的とはいえない



https://www.aimsun.com

- 交通流と経路選択を連続的に記述すれば，そのまま微分できるのでは？
  - マクロ交通流モデル！
  - 桑原・赤松[2]のKWモデル×DUO経路選択理論！！
    - 累積台数：連続
    - ノード分岐率：連続
    - タイムステップ幅が大きく，変数数が少ない
    - 複雑なif文・for文を使わずに表現できる
    - min/max演算は微分不能だが，劣勾配で概ね合理的に近似可能

[2]

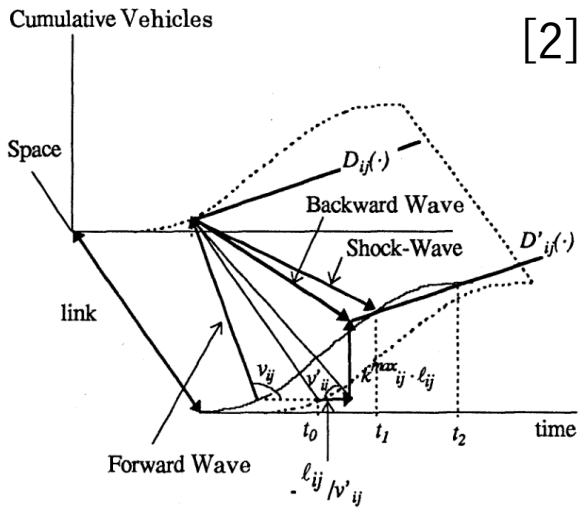


Fig.4 Backward Wave Propagation on link *(i,j)*

[1] Fumiyasu Makinoshima, Yuya Yamaguchi, Eigo Segawa, Koichiro Niinuma, and Sean Qian. 2026. Ultra-fast Traffic Nowcasting and Control via Differentiable Agent-based Simulation. arXiv preprint arXiv: 2603.25068.
[2] 桑原 雅夫, 赤松 隆. 1997. 多起点多終点ODにおける渋滞延伸を考慮したリアクティブ動的利用者最適交通量配分. 土木学会論文集 1997, 555: 91–102.

OSSシミュレータUNsim





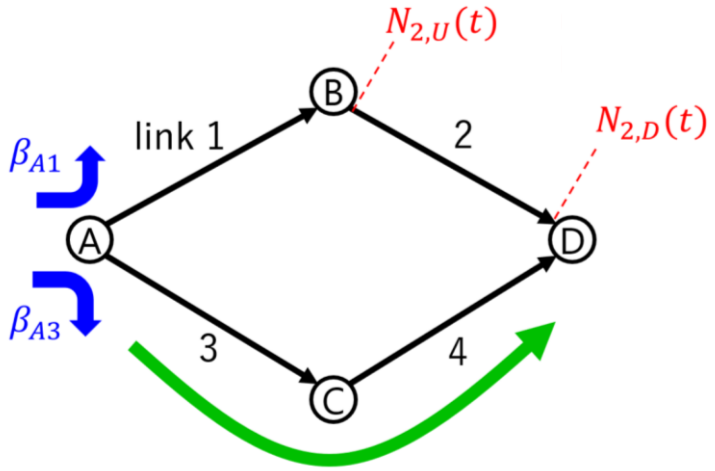
- GitHubで公開：https://github.com/toruseo/UNsim
- UXsimとほぼ同じ感覚で利用可能
- UXsimの資産を利用し2026年に開発・公開

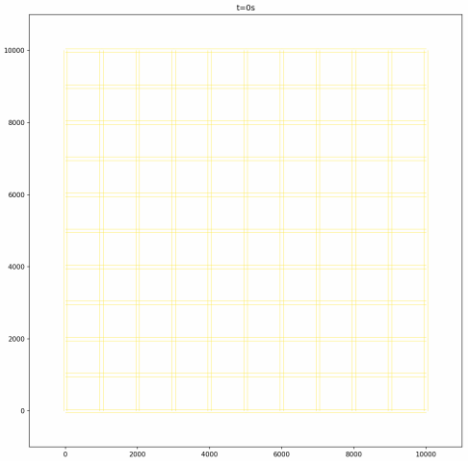
Seo, T. End-to-end differentiable network traffic simulation with dynamic route choice. arXiv preprint arXiv: 2604.11380, 2026.

提案シミュレータUNsim

- UNsim：桑原・赤松のKWモデル×DUO経路選択理論を微調整して実装
  - リンクモデル：KWモデル（Link Transmission Model）
  - ノードモデル：Incremental Node Model（Daganzoモデルの一般化）
  - 経路選択モデル：ロジットDUO．各瞬間での最小瞬間旅行コスト経路に確率的に配分されるモデル
  - PythonとJAXを使用

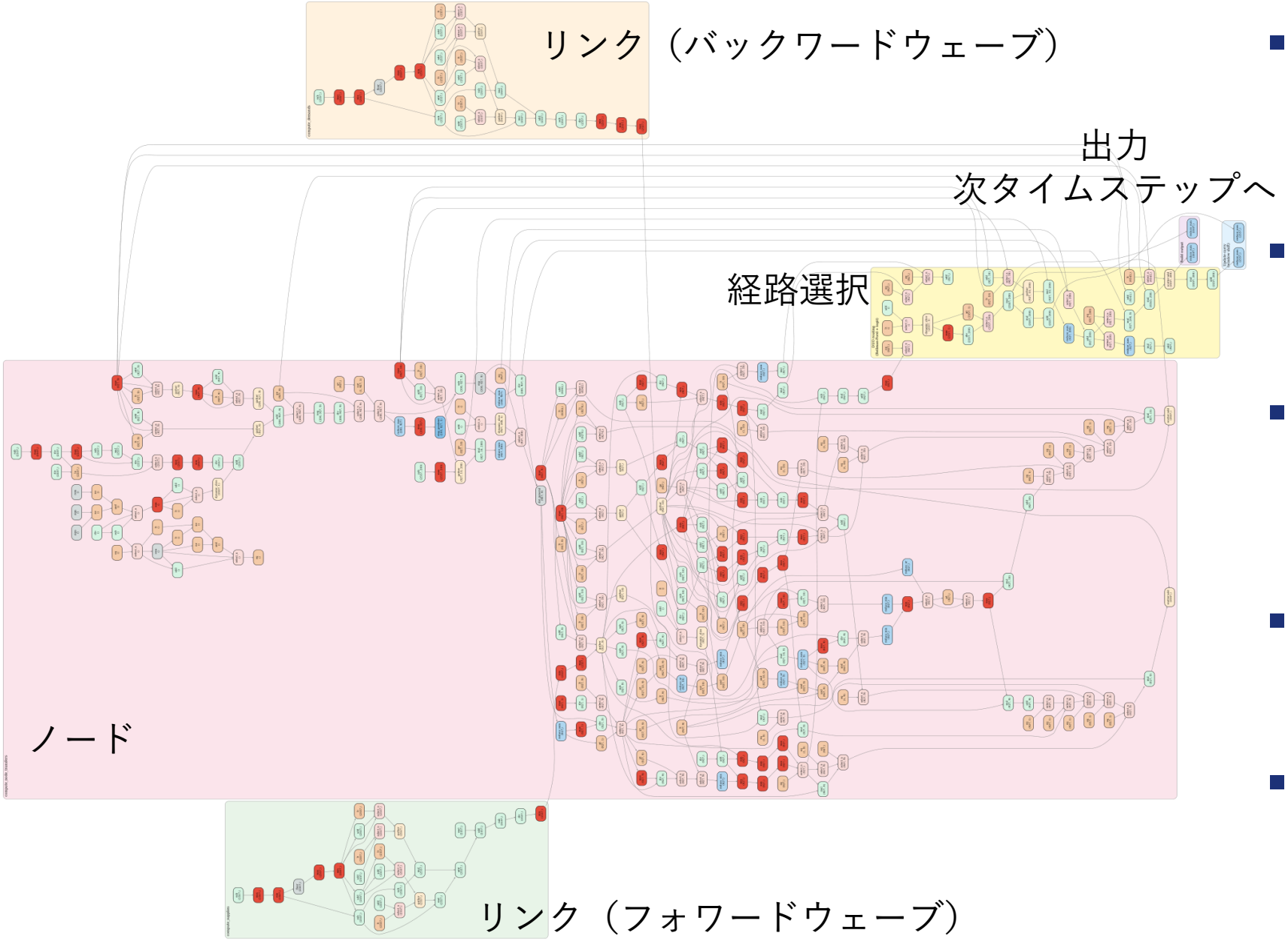
- 任意の出力変数に対する，
  全ての入力変数による微分が計算可能





主な入力変数	主な出力
リンクFD変数値 時間帯別OD需要 リンク動的混雑課金	総旅行時間 時刻ごとのリンク流率・密度・速度 時刻ごとのOD・経路旅行時間

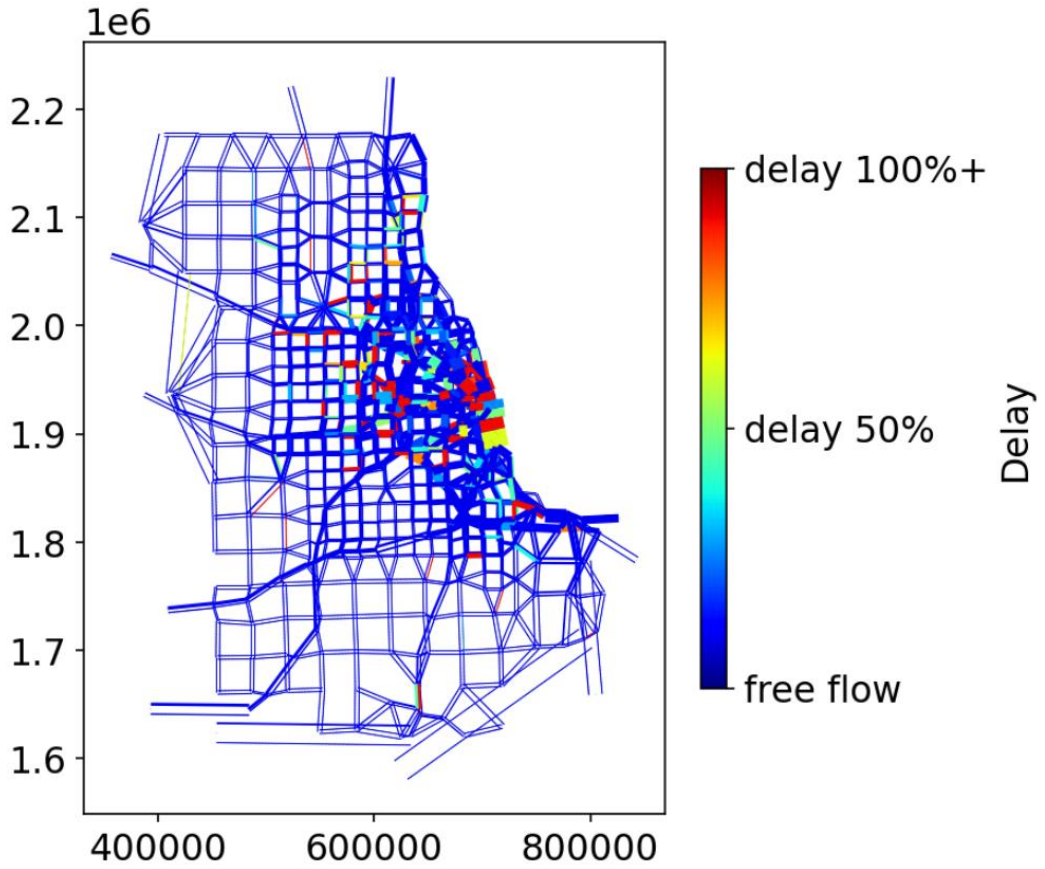
1タイムステップの計算グラフ



- Pythonで書いたシミュロジックを，JAXの計算グラフに変換
- これがタイムステップ数分つながる
- 後述するシカゴシナリオの計算グラフは合計頂点数45万，辺数140万
- GPUによる並列計算可能なので計算時間はとても短い
- ただし，メモリ消費量はとても大きい

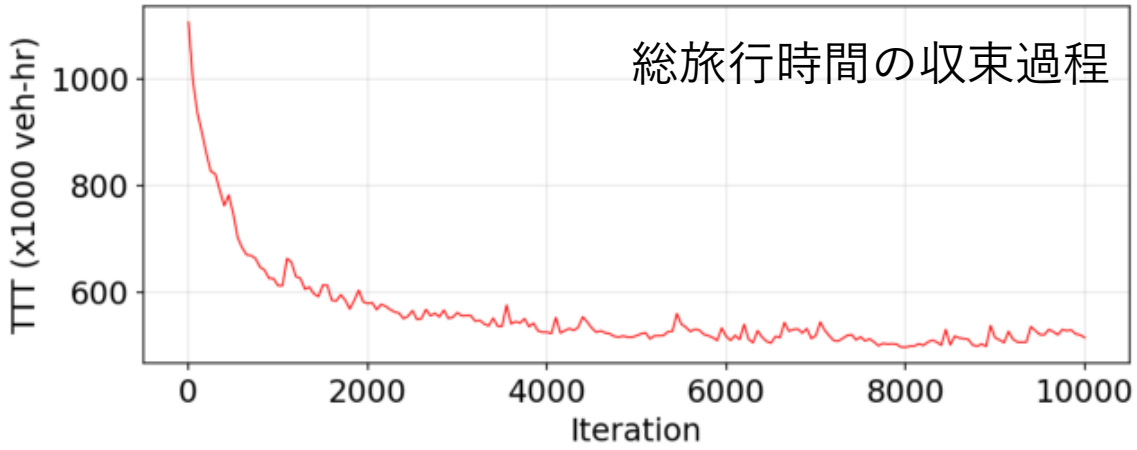
計算例1：シカゴ都市圏の動的混雑課金

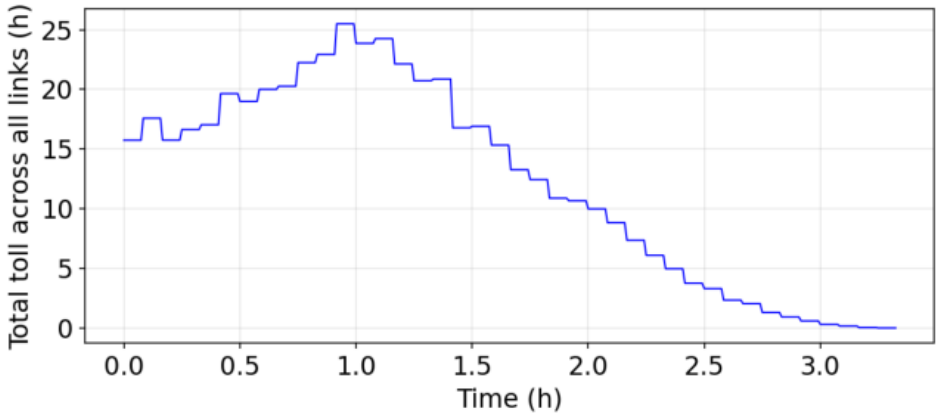
- Chicago-Sketchシナリオ
  - 3時間，927ノード，2557リンク，117万台
  - 計算コスト
    - forwardシミュ計算時間：0.2秒
    - backward微分計算時間：0.6秒
    - VRAM消費量：約70GB
    - 産業用サーバー（GH200）での実測
  - 計算コスト：省メモリ重視
    - forwardシミュ計算時間：0.3秒
    - backward微分計算時間：1.5秒
    - VRAM消費量：約2GB
    - 家庭用PC（RTX5060Ti）での実測



- 最適混雑課金問題
  - 代表的な383リンクに，約5分毎に変化する混雑課金を課し，総旅行時間を最小化する
  - 目的関数：総旅行時間＋正則化項
  - 決定変数：課金額変数15320個

シカゴ都市圏の動的混雑課金

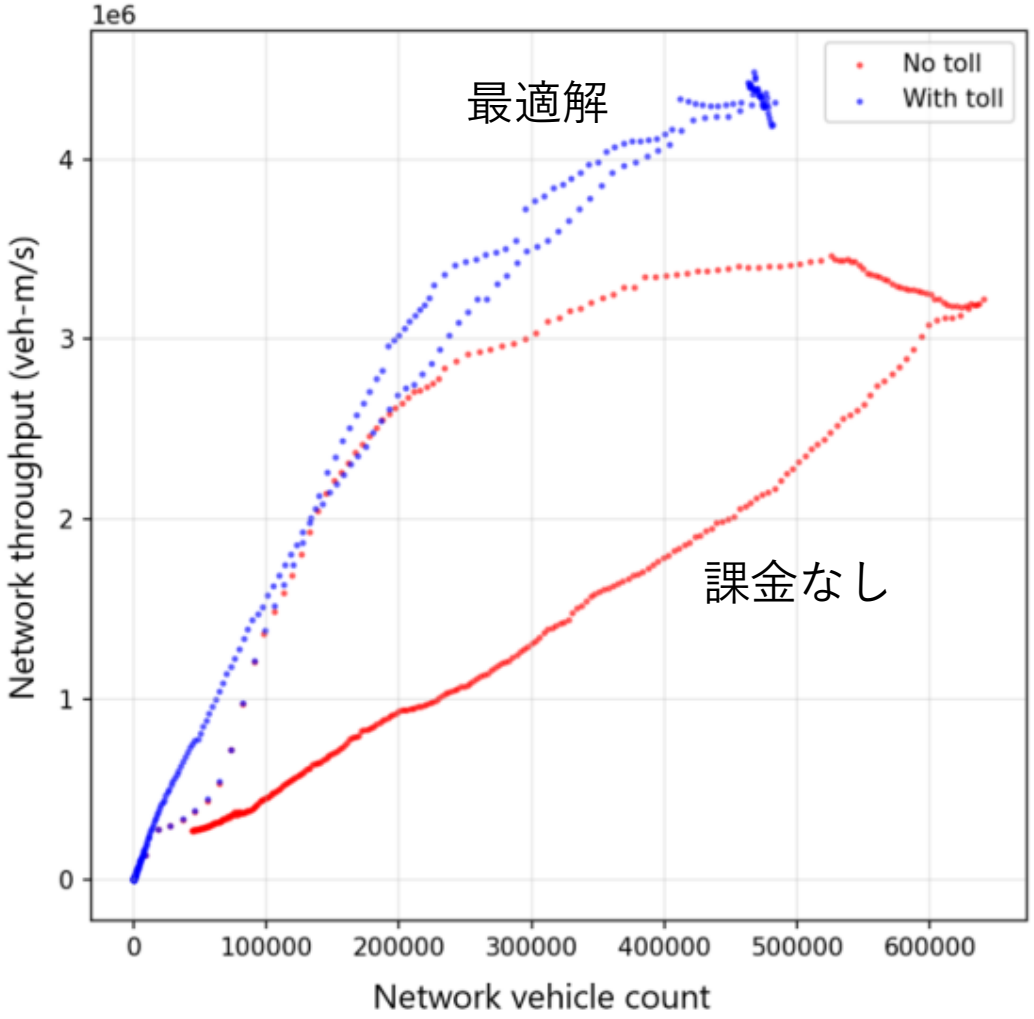




時間帯ごとの平均課金額



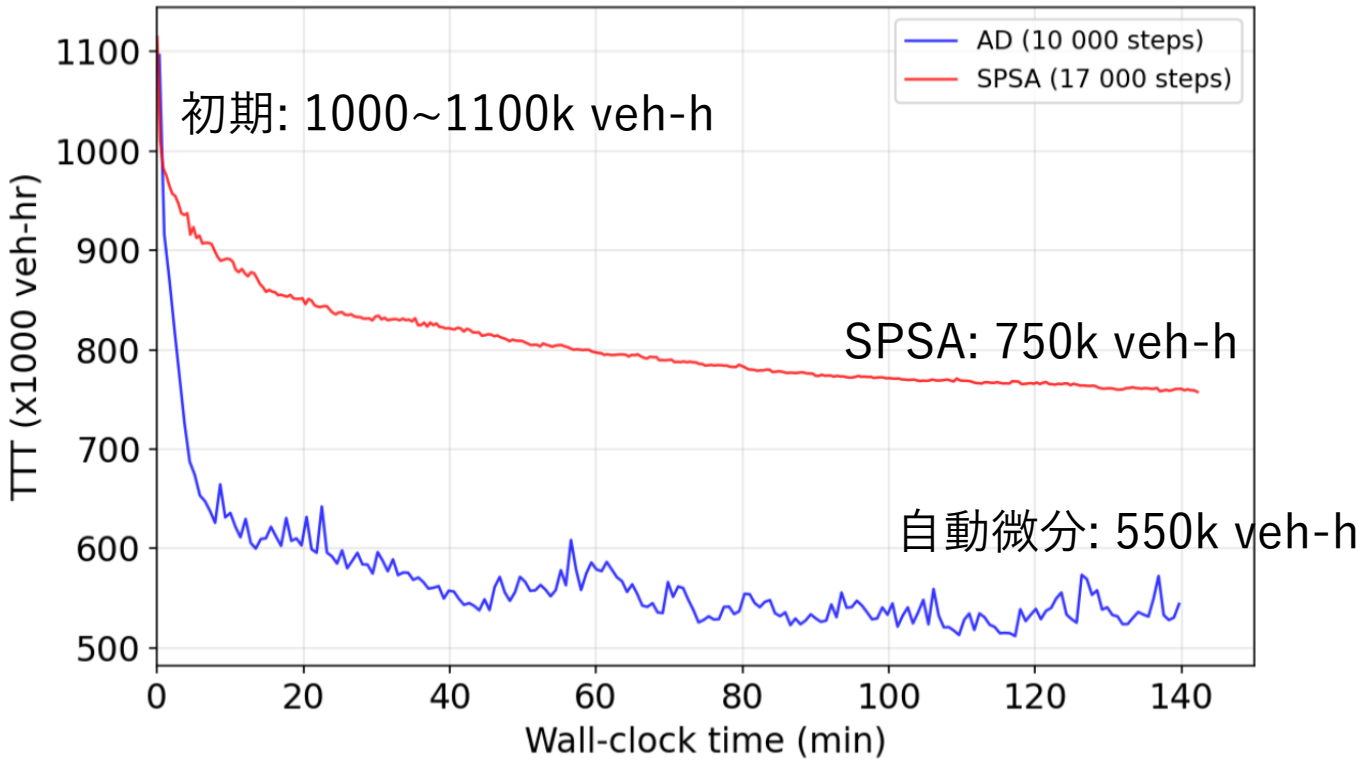
リンクごとの平均課金額

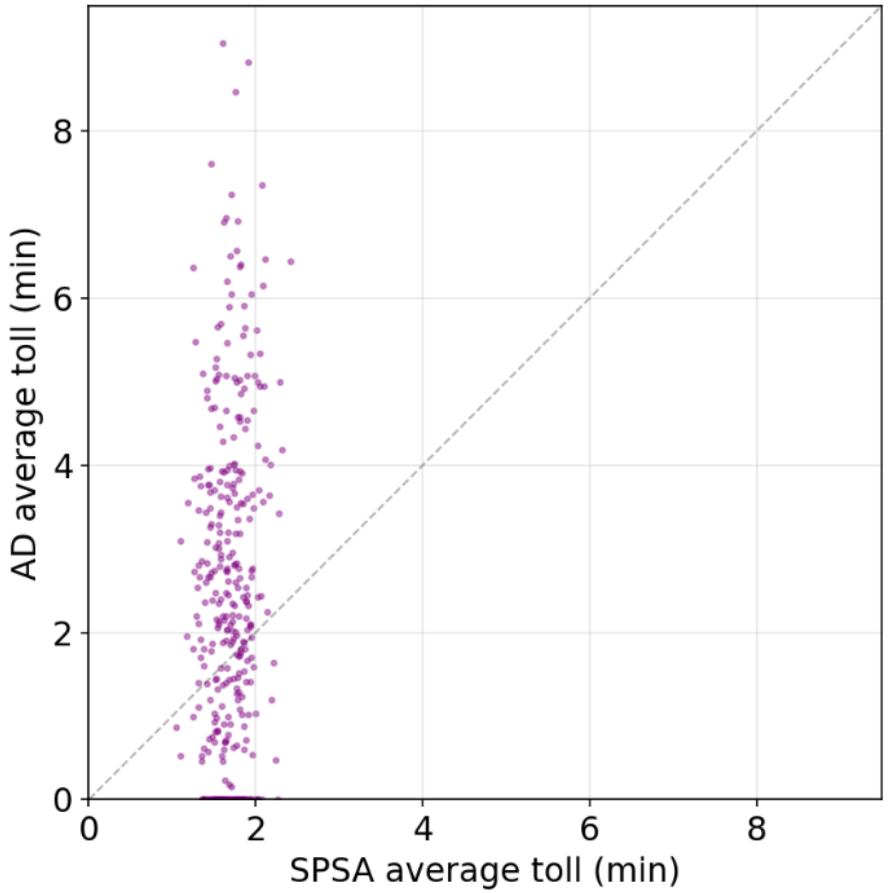


MFDによる状態遷移

シカゴ都市圏の動的混雑課金

- 代表的な交通シミュ最適化の近似解法SPSAとの比較
  - 総旅行時間改善量は，自動微分はSPSAの1.5倍程度
  - SPSAは全リンクにほぼ同じ料金を課している
  - 自動微分はリンク毎に細かく料金を設定

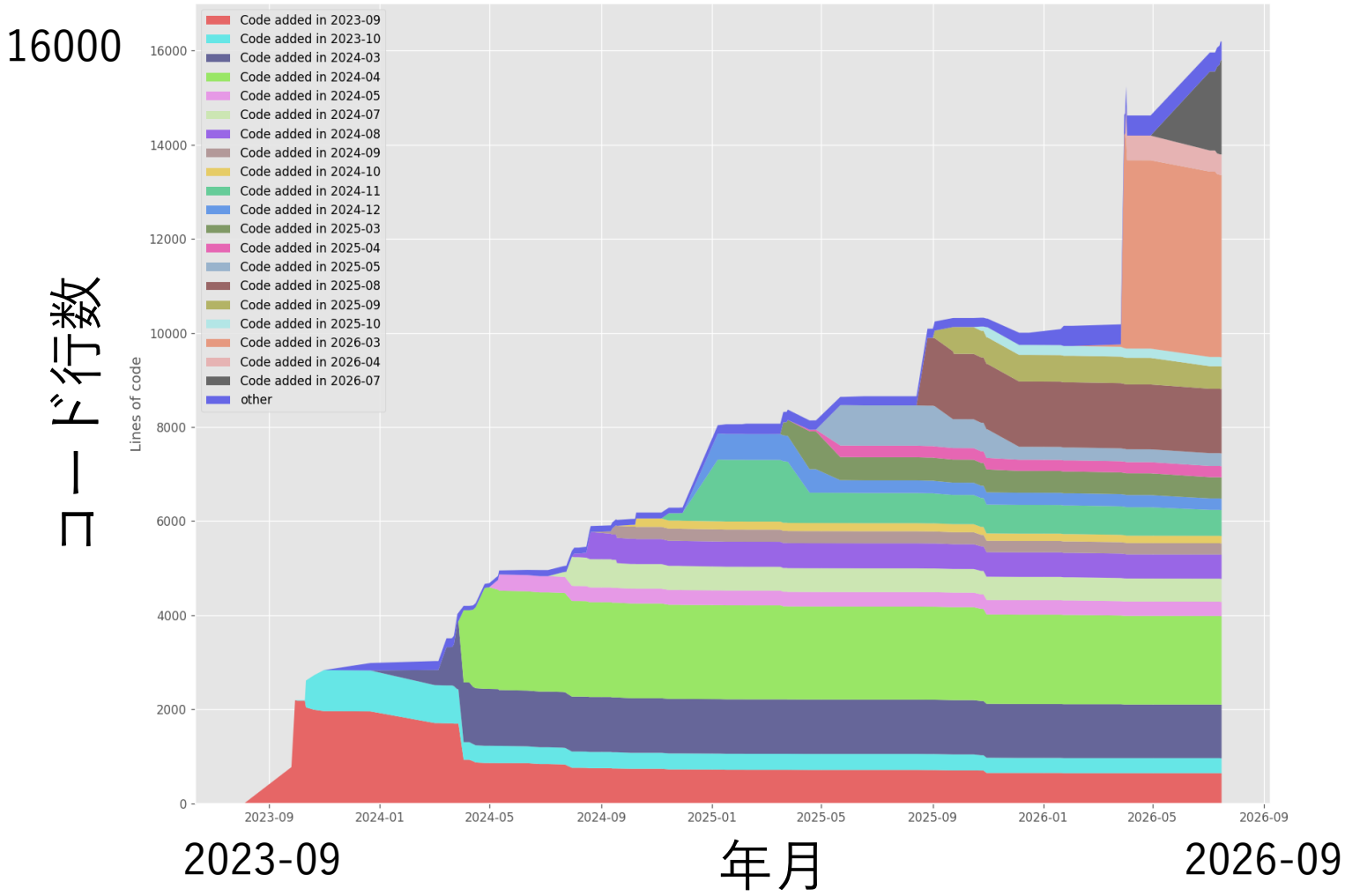




- 基礎
  - 交通流理論
  - 動的交通量配分
  - メソ交通流モデル・シミュレーション
  - 疑似DUE・DSO
- 応用・実践 1
  - メソ交通流シミュレータUXsim
- 応用・実践 2
  - 微分可能交通流シミュレータUNsim
- OSS活動の実践で得られた知見

UXsim開発・公開の実践で得られた知見

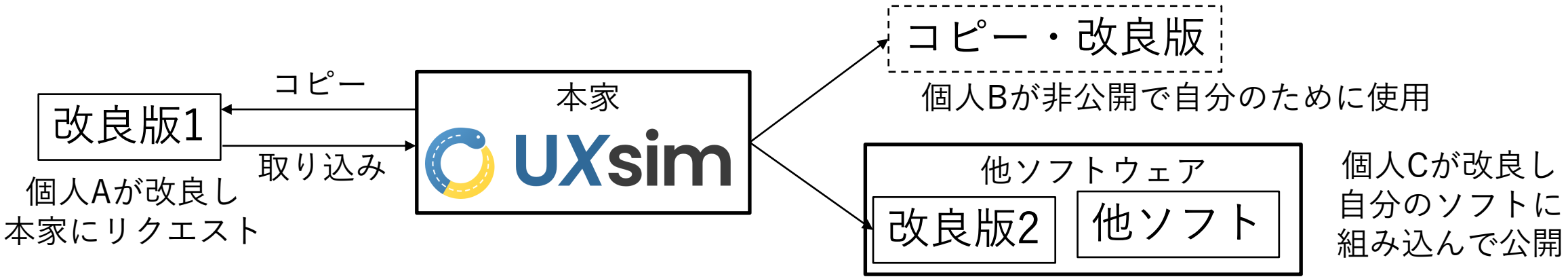
- UXsimは2023年から3年間開発・公開を継続中
- 日本の土木計画学・交通工学分野ではOSS事例はあまり無いので，実践的研究活動として何らかの意味があると思いその知見をいくつか紹介



そもそもOSSとは

- OSS（オープンソースソフトウェア）とは何か
  - ソースコードを公開し，誰でも自由に閲覧・使用・改変・再配布できるようになっているソフトウェア
  - 義務は適切にクレジット・出典を付けることくらい
    - 例外もあるが，細かい話は今は置いておく
  - 改変と再配布の自由が本質的に重要
  - 他の人が勝手に改良してくれる可能性がある．その改良を本家に取り込める
    - ただの慈善事業として無料で公開するわけではない．実利もある

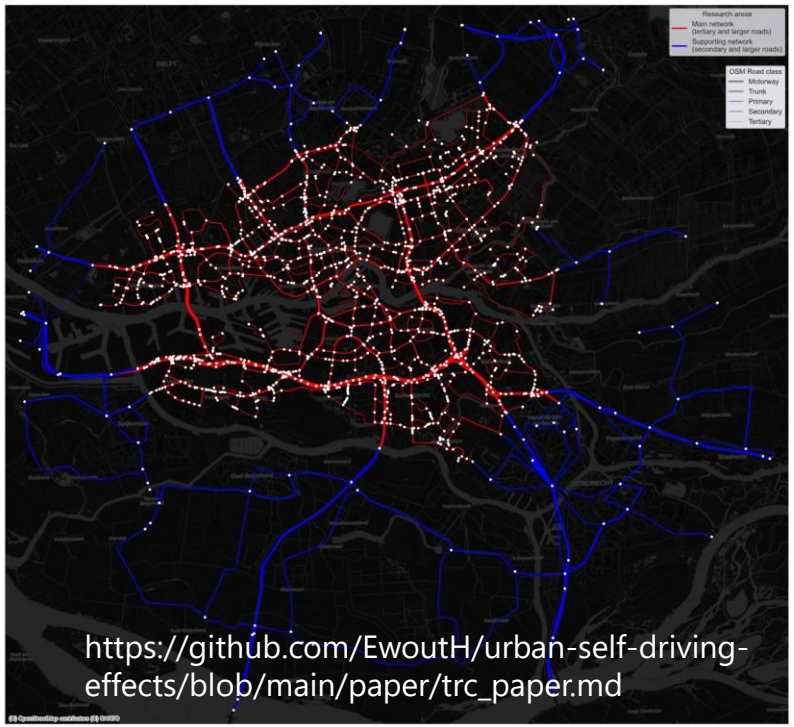
- コモンズの悲劇が起こらないコモンズ：土木計画学的に理想の場
  - デジタル資産で使っても減らないので，フリーライダーが大勢いても問題ない
    - 使ったときに出典さえつけてくれればプラス効果になる
    - （ただし，最近のAIによる計算資源使用量の爆発的増大のため，これは色々変わるかも）
  - 開かれた場なので，協力的行動（改良など）をする人に来てもらいやすい

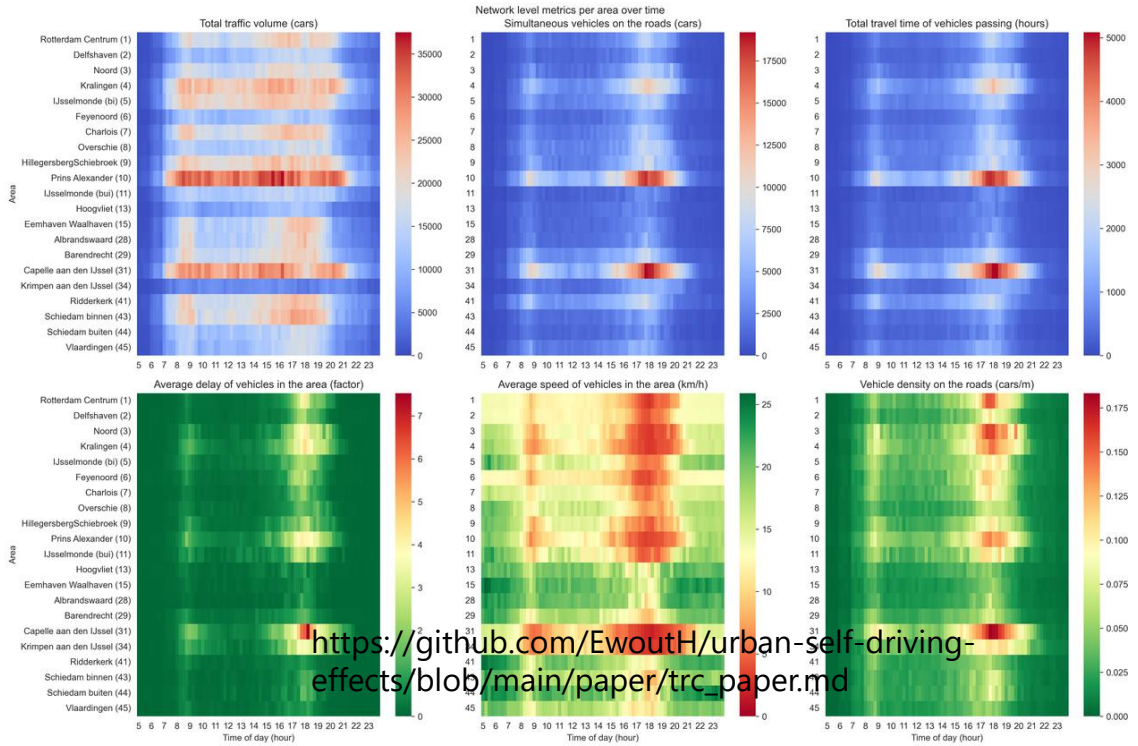


実践で得られた知見：開発

- 他の人が開発に参加してくる
  - コメントフィードバックから，実際のコード更新作業まで

- 「自分の修士研究で使いたいからもっと効率化したい」「ここの計算が遅いから，自分の知ってる技法で高速化してみた」
  - この人は都市全体の1日間のシミュレーション実験をやろうとしていて，当初は想定外だった
  - 結果的に，経路選択モジュールの計算速度が100倍になった





実践で得られた知見：開発

- 「ちゃんとしたプログラム開発の方法を取り入れたほうが良い」「使えるようにしたから，そっちでも使ってみて」
  - プログラム開発中に，ある個所の更新が他の思いもよらない箇所に波及することはよくある
  - その結果，他の機能を壊したまま気づかずに更新が進んでしまい，回復不能になる恐れがある
  - 大規模プログラムでは，これは人間の注意では防げない

- 機械的に検査する方法論：CI/CD，テスト駆動型開発
  - コードが更新されるたびに，既存機能が正しく動くかを自動でテストする
  - そのテストに合格しないと更新が反映されない
  - 大規模プロジェクトで複数の人間が協働するための方法
  - 余談だが，これは人間×AIの協働でも極めて有用

実践で得られた知見：利用

- 意外と利用される．全く想定してなかった使われ方もある

- この3年間でUXsimを利用した論文数：29本
  - Google Scholar等で測定．arXiv，会議論文を含む．うち瀬尾が関係しているものは4本
  - 有名どころ：NeurIPS, Journal of Transport Geography, Transport Policyなど
  - CS分野での使用が目立つ
  - 一覧：https://github.com/toruseo/UXsim/wiki

- 主なテーマのカテゴリ
  - 通常の交通シミュとしての使用．交通政策評価，信号制御，交通データ分析など
  - カスタム可能な交通シミュとしての使用． PyTorchの強化学習コードに組み込むなど
  - EV充電と課金の疑似DUE分析[1]
  - AI自体が交通シミュコードの内部に入り込む交通制御手法[2]→これはUNsimのヒントになった
  - AIによるプログラム開発の実験対象[3]

[1] Zheng, B. L., Wei, F. F., & Chen, W. N. (2026). Surrogate-Assisted Fictitious Play for EV Charging Station Pricing Game. In 2026 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE.
[2] Cheng, C. A., Nie, A., & Swaminathan, A. (2024) Trace is the Next AutoDiff: Generative Optimization with Rich Feedback, Execution Traces, and LLMs. NeurIPS.
[3] Orlanski, G., Zhang, A. L., Trost, A., Chen, V. S., Sala, F., Albarghouthi, A., & Schmidt, L. (2026). Can Agents Design Libraries for Agents?. arXiv preprint arXiv:2609.36730.

実践で得られた知見：利用

- Can Agents Design Libraries for Agents? [1]
  - 最近はAIエージェントによる自動コーディングが盛んになっている
  - 研究用のプログラムもAIで全自動で作れることもある．むしろAIの方が優秀な場合すらある
  - いくつかの既存ライブラリの仕様をAIに与えて開発させ，成果物のクオリティを定量的に測定した研究

Production library	Domain
pydantic	Schema validation
pandas	Dataframes
uxsim	Traffic simulation
fastapi	HTTP API framework
simpy	Discrete-event simulation
beautifulsoup4	HTML parsing

AI版UXsim ⇨

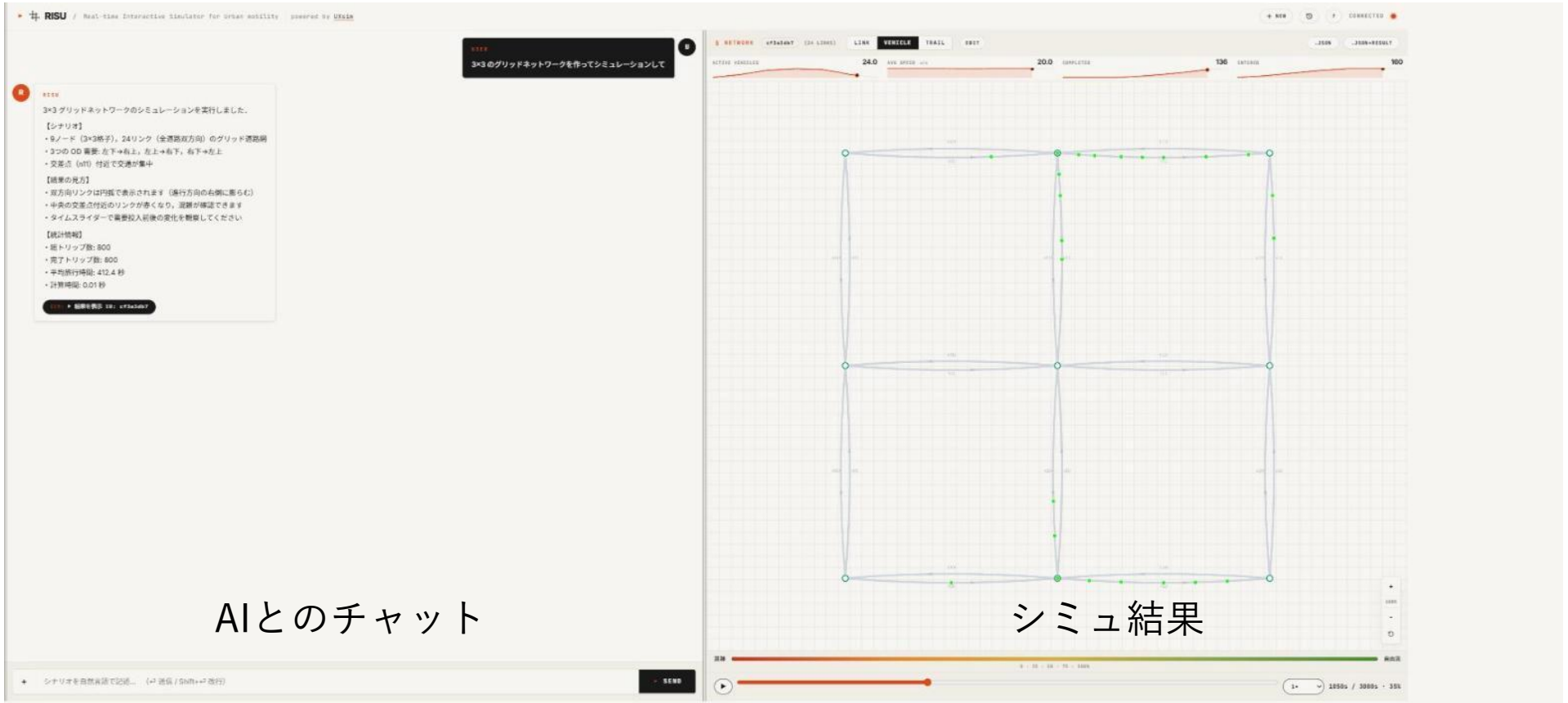
	Opus 5.5	Fable	Astra	Astra (Codex)	Kimi K3	GLM 5.3	Fable CC	Grok 4.6	GPT-5.6 Sol	GPT-6 Sol	DS V4 Pro	No library
	Python											
canon	+6.4	+0.9	−3.2	−1.6	−5.2	−10.4	−14.2	−11.5	−4.2	−9.6	−23.7	−25.0
pyda	+1.2	−2.9	−9.1	−7.0	−7.2	−12.5	−7.7	−14.4	−14.8	−13.7	−22.7	−7.0
roadkill	−10.8	−12.7	−17.5	−19.0	−13.5	−15.4	−16.2	−17.1	−18.5	−19.3	−20.9	−20.4
sapi	−6.5	−4.2	−8.7	−9.0	−12.9	−8.1	−15.1	−12.3	−12.1	−11.0	−26.9	−28.5
simu	−4.5	−1.5	−0.1	−5.1	−3.2	−0.4	−7.0	−6.9	−0.7	−3.7	−7.9	−4.1
uglypie	+1.9	+0.5	+4.1	+0.3	−0.3	−1.2	−29.1	−5.7	−4.0	−6.9	−15.2	−24.8

赤：AI製の方が劣る，緑：AI製の方が優れる

[1] Orlanski, G., Zhang, A. L., Trost, A., Chen, V. S., Sala, F., Albarghouthi, A., & Schmidt, L. (2026). Can Agents Design Libraries for Agents?. arXiv preprint arXiv:2609.36730.

実践で得られた知見：利用

- 自分のソフトウェアに組み込んでリリースしている事例もある
  - EVを考慮した電力グリッドシミュレータ
  - AIに交通シミュ分析をさせるソフト



https://github.com/kurumido-org/risu-local

まとめ

- ネットワーク交通流理論は一見難しいかもしれないが（実際難しいが），ある程度まではシンプルに理解でき，初学者でも使えるようになっている
- マクロ・メソシミュレーションはミクロと比べて圧倒的に効率的に都市規模シミュが回せる
  - シナリオ準備（パラメータ設定）が楽
  - 計算が速い
  - このスケールには土木計画学・交通工学的に興味のある対象が多い
- 動的利用者均衡・システム最適に準ずる計算もできるようになってきている
- OSSのメソ交通流シミュUXsim，微分可能マクロ交通シミュUNsimを開発・公開
  - 利用・フィードバック・開発参加歓迎します！
- コロナ社から交通流理論・シミュレーションの専門書も出版
  - 微分可能シミュ以外の話はこちらで詳述



UXsim　UNsim