The 2017 Summer Course “Behavior Modelling in Transportation Networks”
Urban science and behavioral informatics with AI/machine learning

Optimization in behavioral modelling with HPC

Sep. 15, 2018

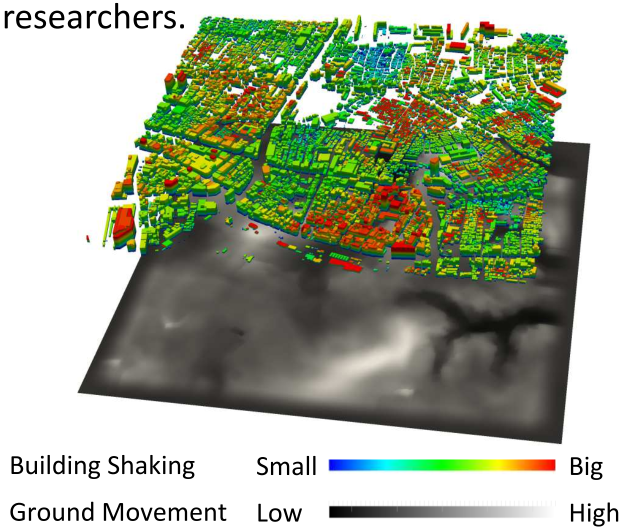
Junji Urata (Univ. Tokyo)

Collaboration work with Takamasa Iryo, Peque, G. Jr., Riki Kawase

(Kobe University)

(Post K Project) Development of Traffic Demand and Flow simulator in a disaster restoration period

Simulations for Tsunami, Earthquake, Building Collapse by physics and structual researchers.



Road damage

Collapse of buildings
Damage of infrastructure for lifeline

Traffic Demand Simulator:
Obtain a set of OD demand pattern

	Dest. 1	Dest. 2	. .
Orig. 1	1003	512	391
Orig. 2	454	896	402
:	344	385	786

:	315	354	812

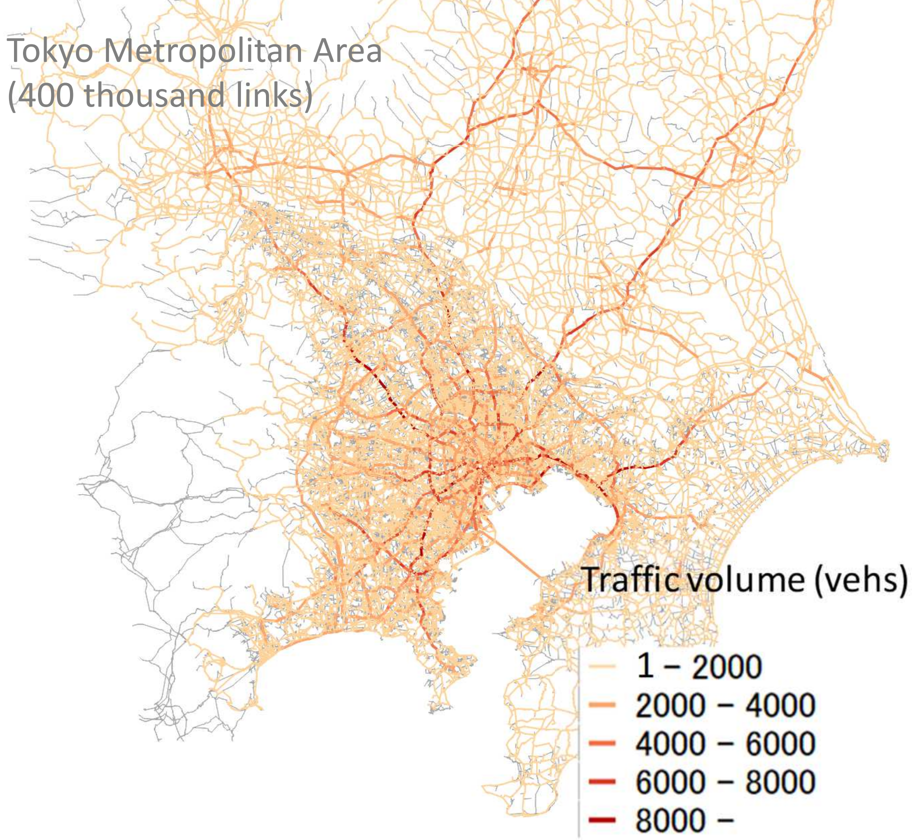
:	365	371	803

Travel Time

Traffic Demand

Dynamic Traffic Flow Simulation:

- Iterative calculations (Shortest Path calculator->Traffic Flow calculator->SP->TF…) are made to find near-equilibrium solutions, in which most vehicles select their shortest paths.
- By sampling results of the iterations, we will obtain a stationary distribution of traffic patterns that are nearly equilibrated.



What is a Disaster Restoration Period?

Distinguish three periods on disaster situation

- Evacuation period:
  - People have to go to a safety place.
  - The period is within a few hours or a day in Japanese-type disaster.
    ex. Tsunami, Mudslide.
- Rescue period:
  - The period is during a rescue operation.
  - The periods is within 3 days after disaster.
- Disaster Restoration period:
  - The period is until temporal restoration of lifeline infrastructure.
  - The periods is within a month or more.
  - Some evacuees still live in shelters but many people restart to work.

Traffic Congestion on Disaster Restoration Period
Kure city case (July 2018 heavy rain disaster)

Demand:

Commuting Volume before the disaster

Damages for transportation network (more than 2 months)

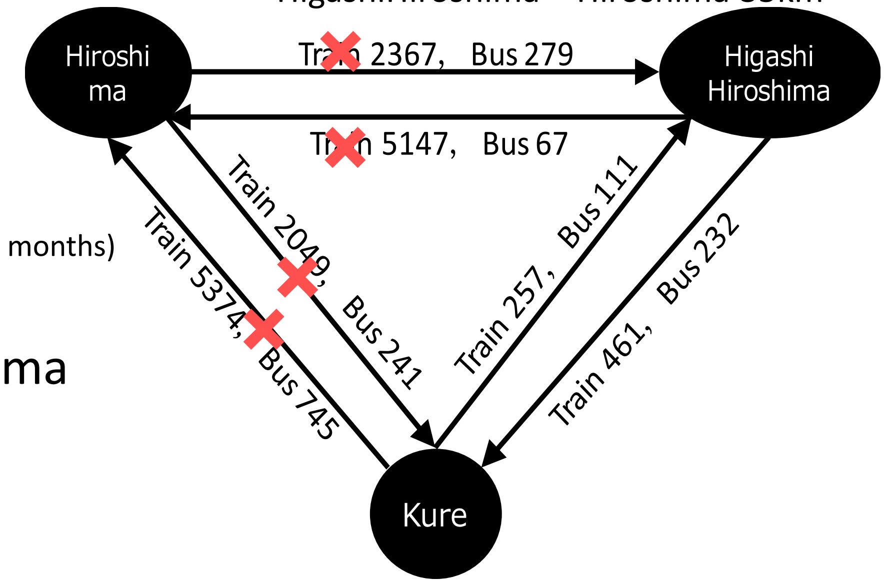
Railway between Hiroshima & Kure

Railway between Hiroshima & Higashi-Hiroshima

Highway between Hiroshima & Kure

Kure – Hiroshima 25km

HigashiHiroshima – Hiroshima 35km



Traffic Congestion and Emergency Supply Management:

Many people rode own car through national roads.
Many trucks which conveyed debris drove.

Urgent temporal Bus and Shikansen were managed however the supply volume is half of usual demand.

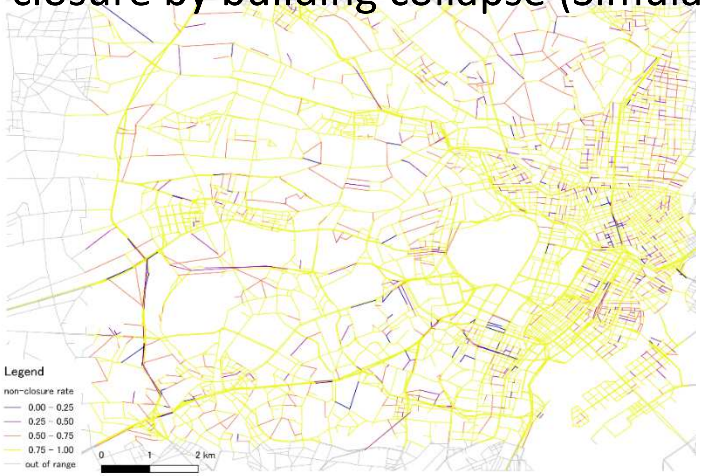
Shinkansen Station

Temporal Bus Station

Collaboration work with Makoto Chikaraishi and Daisuke Yoshino

Optimization on restoration period

Road closure by building collapse (Simulation)



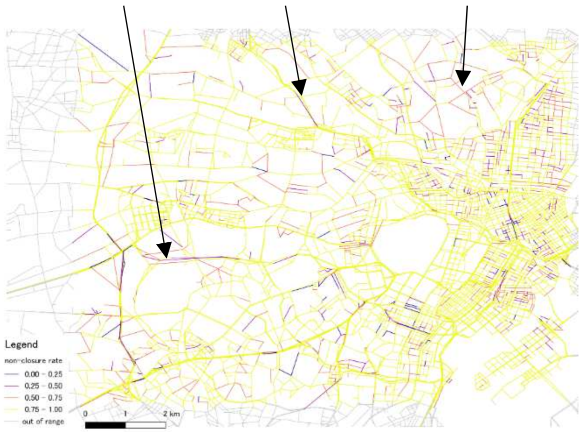
How to recover and How to keep necessary trips?

- Debris clearance
- Network design for urgent vehicles
- Temporal bus network and timetabling

and so on.

→ We apply bi-level optimization to know valid solution.

Upper Problem:
Choose clearance links



Scenario →

← Score

Lower Problem:

Dynamic Traffic Flow Simulation
evaluate travel time by time unit.

Link travel time ↓ ↑ OD pattern

Dynamic Demand Simulation
evaluate origin-demand pattern.

- Upper problem is combinatorial optimization, so the number of candidate solution is too much.
- Evaluation time of lower problem should be as small as possible to test numerous scenarios.

How to decrease calculation time on lower problem?

HPC (High-performance computer)

- Using HPC (High Performance Computer) is the most straightforward approach to fast calculation.
- Recent computers gain a power by combining a number of CPUs.
  - 663,552 CPUs in K computer
  - 8 threads in CPU (Fujitsu SPARC64 VIIIfx)

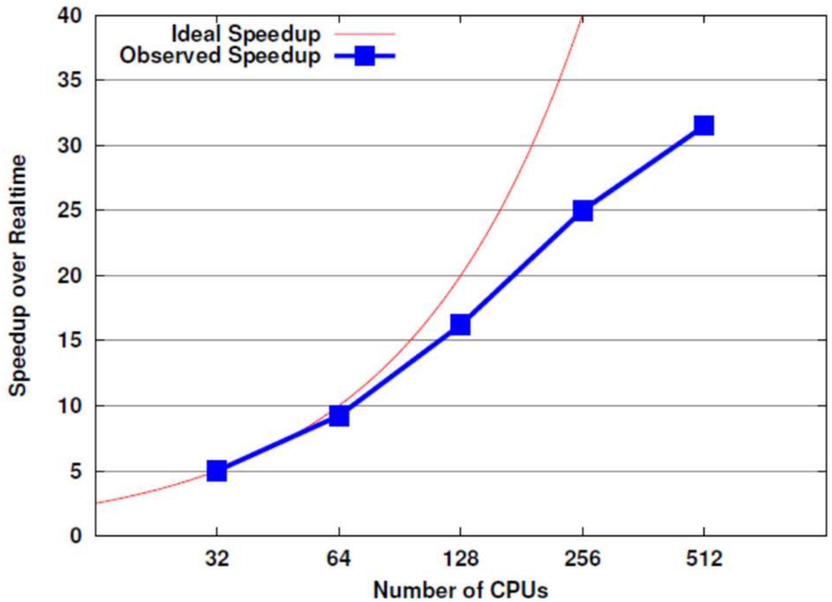
- Actually, the performance of a single CPU is equivalent to a typical desktop server, hence we have to develop a suitable parallel algorithm (efficient algorithm) which evenly split the task into sub-tasks.
- Moreover, we need to carefully consider how to divide the calculation task so as to minimize the need of data exchange between CPUs.

Dynamic Traffic Flow Simulation

Existing parallelised implementations

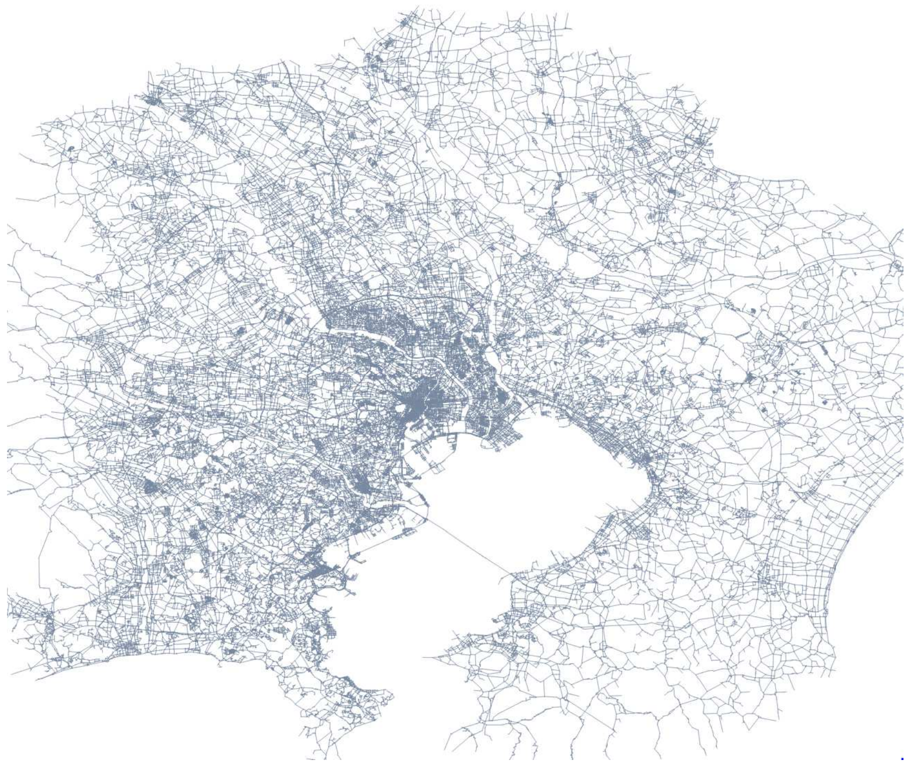
- Not many parallel implementations for traffic simulators in past studies.
- Scalable implementation up to 15,000 cores has been shown in a recent study (Turek, 2018), while it only considers ideal grid networks.
    - Actual road networks are more hierarchized and unevenly distributed than a simple grid.
- Other studies use < 1000 cores (e.g. Nagel and Rickert, 2001; Thulasidasan and Eidenbenz, 2009) and their scaliability is low with 100 cores.

Thulasidasan and Eidenbenz, 2009



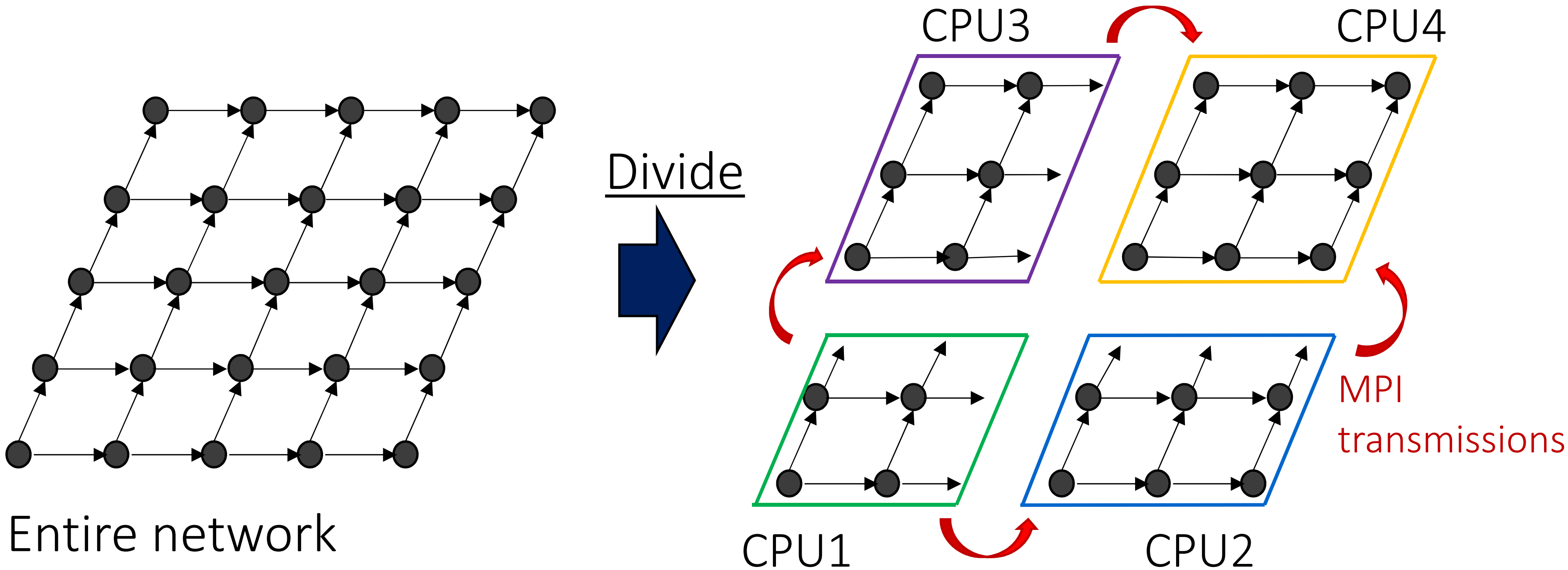
Aim of high-parallelised DTA algorithm

- This study aims to propose an parallelised network loading algorithm suitable for many-core CPUs and HPC.
- An asynchronous algorithm for calculating flow propagations is proposed to reduce communication cost between CPUs.
- The proposed algorithm is implemented on K computer.
- Test case by Kanto Ken-o-do network (approx. 0.35 million links) is shown.



Domain decomposition of DTA problem

- For the parallel implementation, the entire network is divided into sub-networks.
- Each CPU handles one of them.



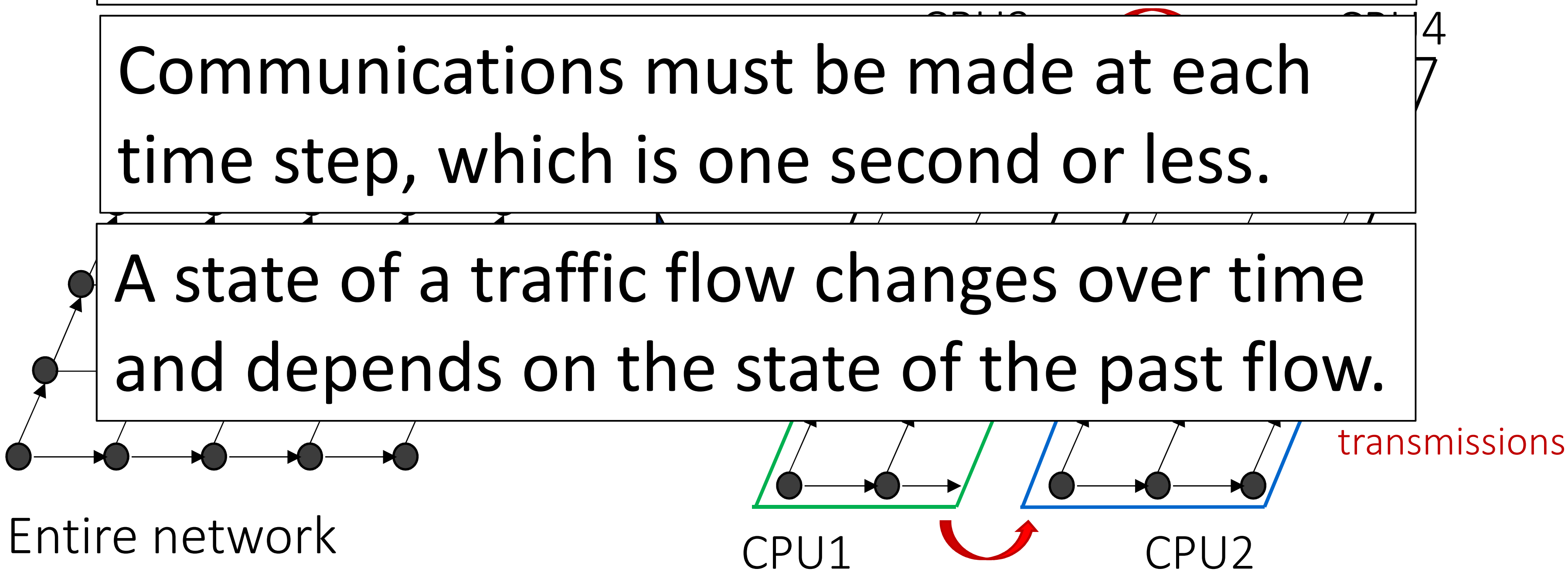
Domain decomposition of DTA problem

- For ... s
  div...
- Eac...

Problem:
How frequent the communications between different CPUs have to be made?

Communications must be made at each time step, which is one second or less.

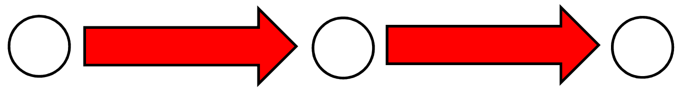
A state of a traffic flow changes over time and depends on the state of the past flow.



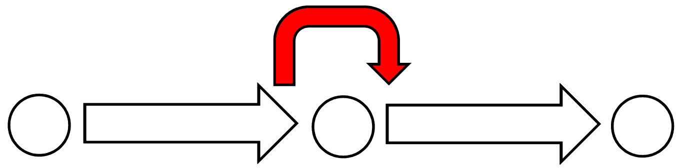
Asynchronised update

Proposed Algorithm

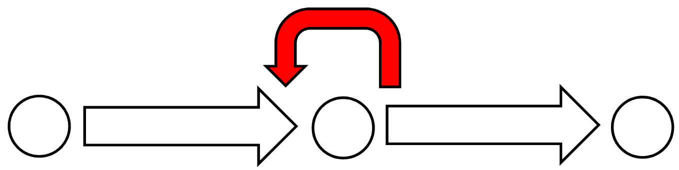
1a. Internal update:
Calculate traffic flow pattern within each link independently on full time step.



1b. Forward relay:
Transmit information on outgoing traffic flow to downstream link(s).



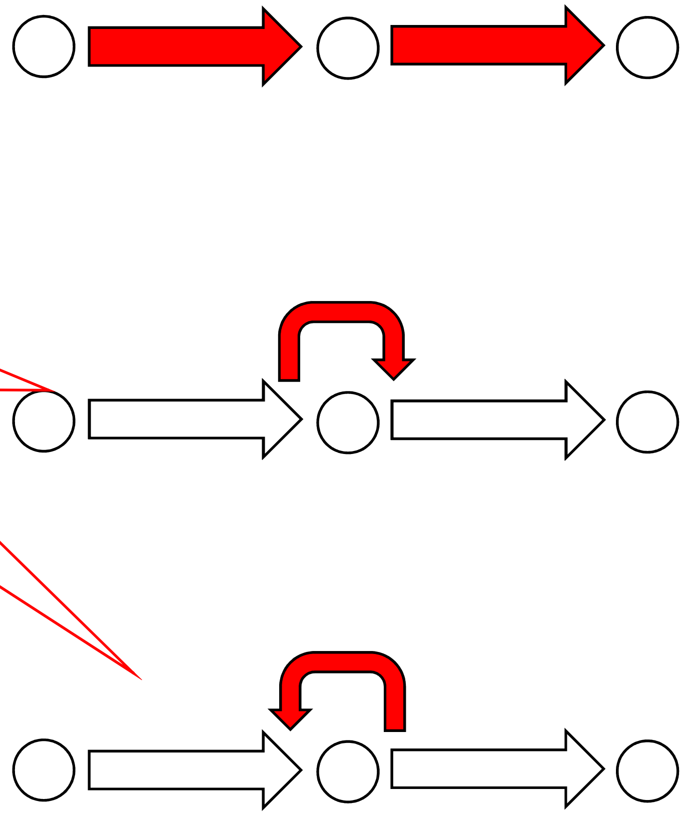
2. Backward relay:
Transmit information on queue spillbacks to upstream link(s).



Asynchronised update

Proposed Algorithm

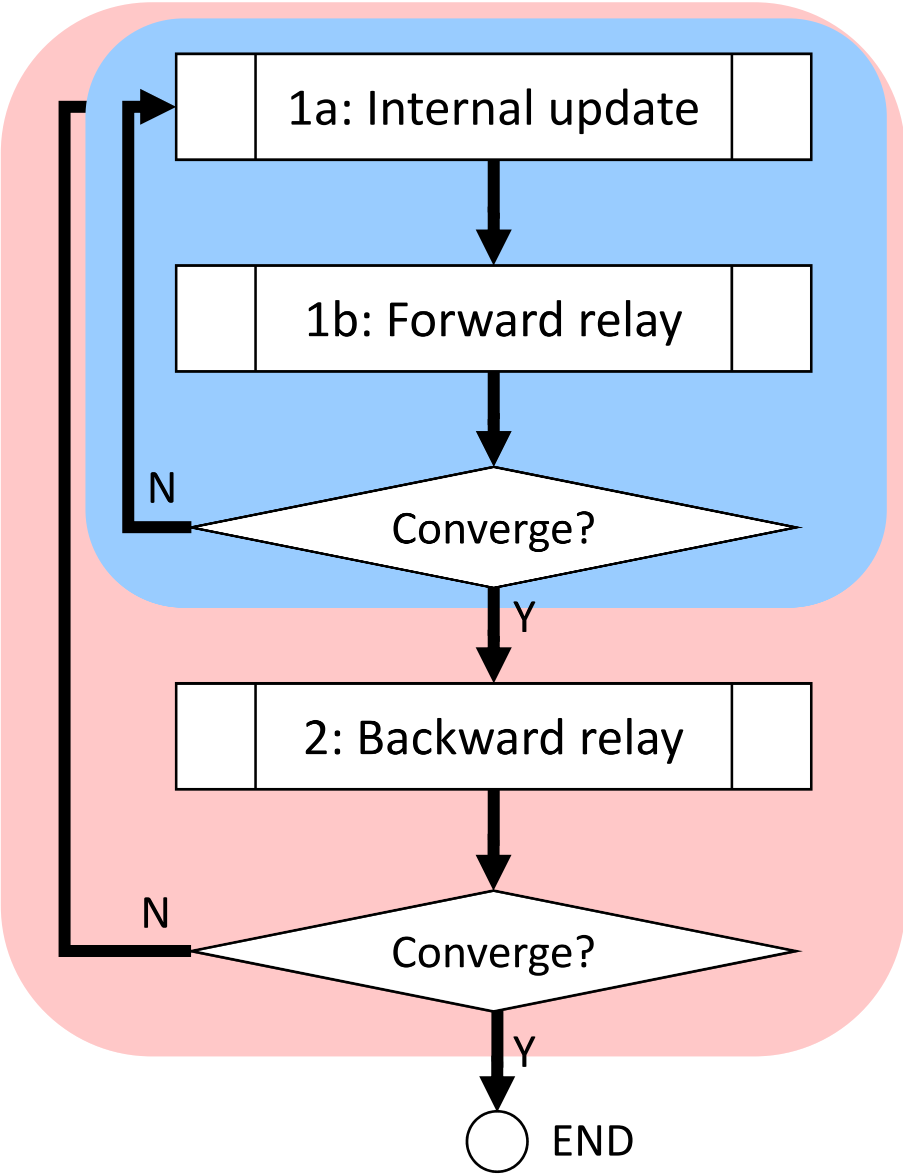
1a. Internal update:
Calculate traffic flow pattern within each link independently on full time step.



MPI communications are made at once for the entire time step (typically several hours), reducing substantial amount of communication time between different CPUs.

Transmit information on queue spillbacks to upstream link(s).

Iteration flowchart



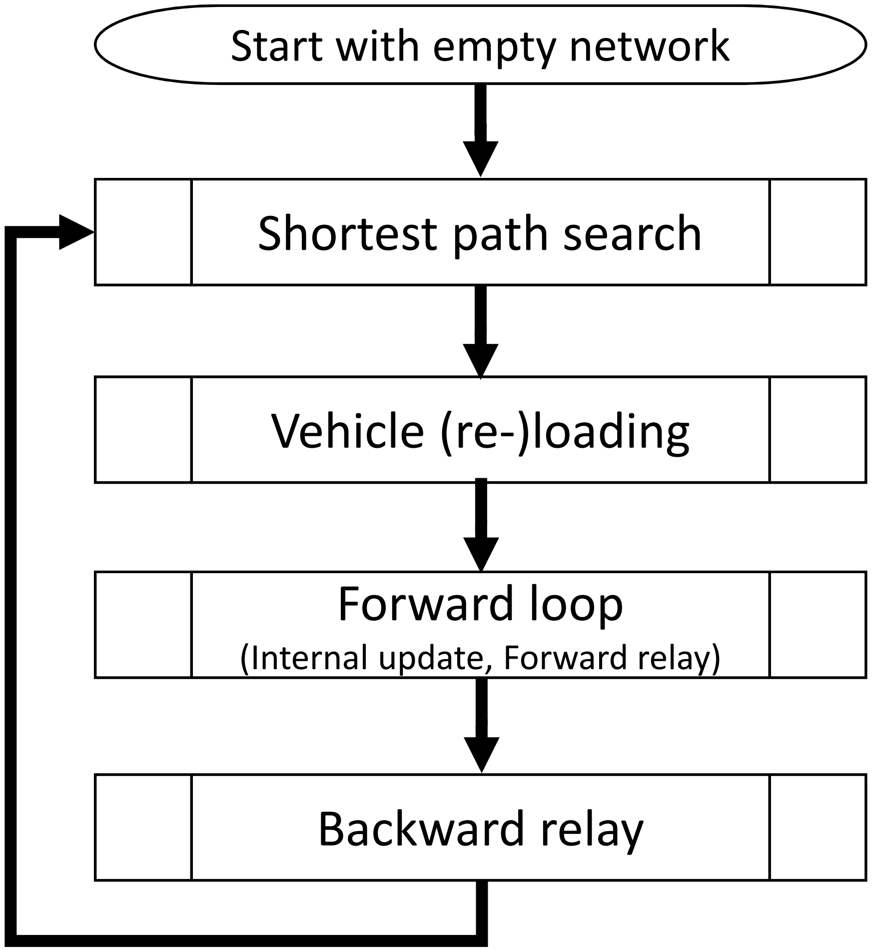
Forward loop:

Internal update and forward relay are first repeated till it converges, in which no change of incoming traffic flow is found in the forward relay.

Outer loop:

Backward relay is made once the forward update/relay is converged. Other processes such as route choices may be also included in outer loop.

Overall process (with route choices incorporated)



- In the earlier iterations, vehicles are incrementally loaded onto the network.
- A few percent of vehicles already loaded rechoose their route to reduce their own travel times.
- After the iteration, we could obtain a state that is close to DUE solution (more specifically, a stationary distribution of the Markov chain representing a day-to-day dynamics).

+ ALT algorithm which is advanced A-star algorithm will be implemented on shortest path choice part and pre-processing process for a calculation of A-star score is parallelized.

Peque, G. Jr., Urata, J., Iryo, T.: Preprocessing Parallelization for the ALT-Algorithm, Computational Science - ICCS 2018, pp. 89-101.
Peque, G. Jr., Urata, J., Iryo, T.: Implementing an ALT Algorithm for Large-Scale Time-Dependent Networks, The 22nd HKSTS International Conference (HKSTS 2018), Hong Kong, December 9-11, 2018.

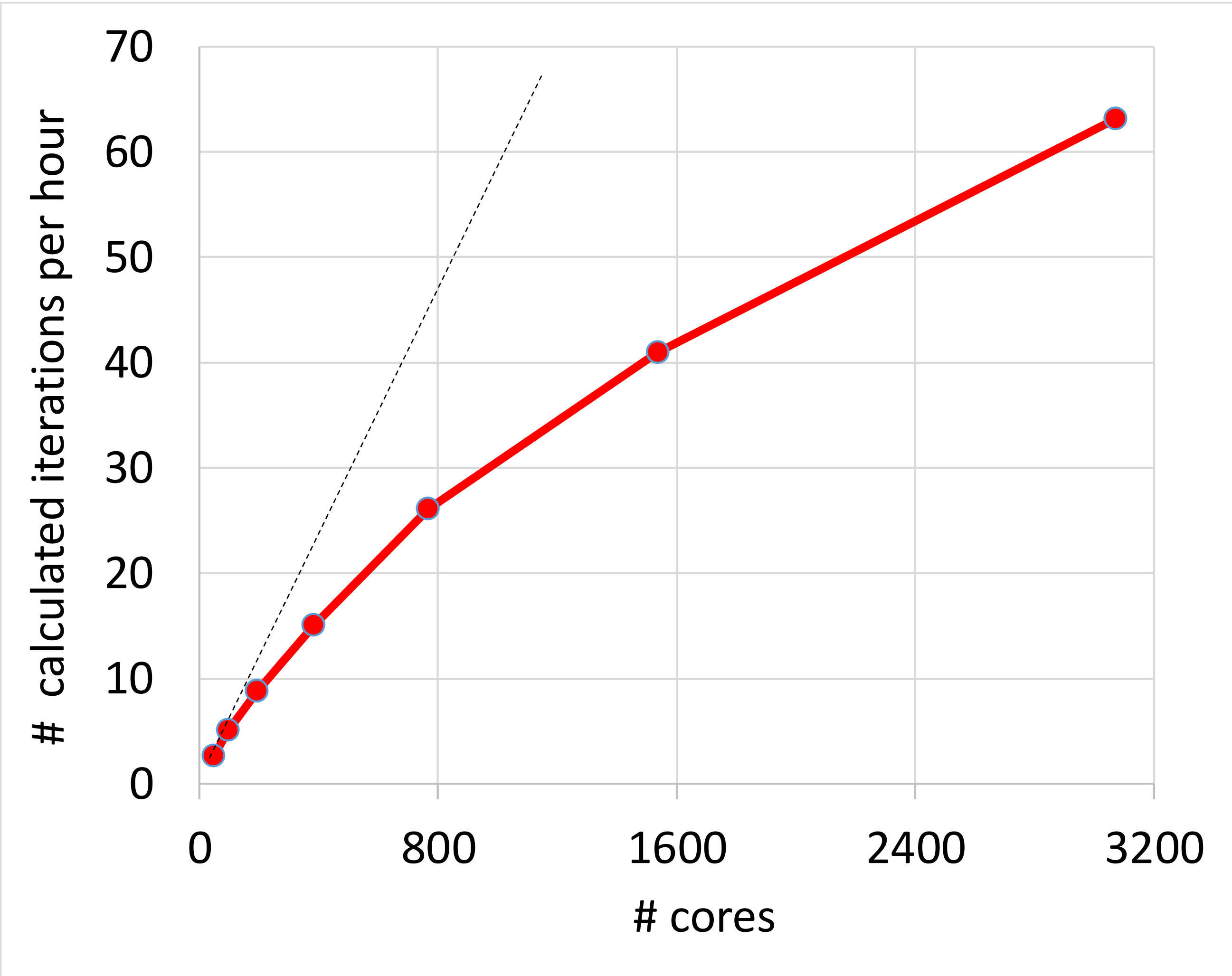
Case study: Settings

- Kanto region (incl. Tokyo) , within Ken-o-do road. # links = 347,691.
- In this case study, trips from 4 a.m. till 10 a.m. is loaded to the network.
  - Packet size = 10 vehicles / packet (5 for trucks).
  - Total # of packets = 558,572 packets.
  - For the test purpose, we used a special setting (mechanisms and parameters) for the calculation process of merging sections / intersections.
  - A constant demand pattern is given.

Result: Traffic volume

Scalability of the proposed simulation

- Highly scaled with 400 cores
- Need to implement dynamic load balancing to be more efficient parallelization



Traffic Demand Simulation

Key points of Demand simulator

- In order to assess congestion of roads in a disaster restoration period, it is essential to construct a stochastic demand model and sample various demand patterns that are likely to happen by a Monte Carlo simulation.
- Obtaining a sufficient number of samples of demand patterns and simulating a traffic flow with them will incur a huge amount of calculations.
- Only a high-performance computer can deal with this task when a size of network is large with sampling a lot of OD demand patterns.

Background of Demand model

OD (origin-destination) matrix has a strong effect on road congestion.

Just after Kumamoto earthquake (1 week later)

After a disaster occurred,

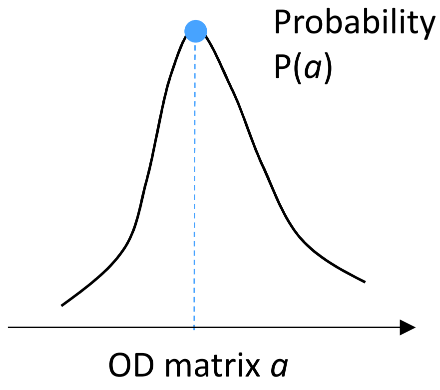
- Cars cannot pass through some links which are damaged by a disaster.
- OD matrix will be changed after a disaster,
  - People come in a damaged area for restoration and humanitarian logistic.
  - Habitants go and get foods & necessaries at somewhere.
- Road network is set for ordinary OD matrix and doesn't prepare for extra-ordinary OD matrix.
- That's why this changed OD matrix will occur a heavy congestion.

How can we predict a OD matrix for unobserved network?

→ Focus on the probability of OD matrix with less data situation.

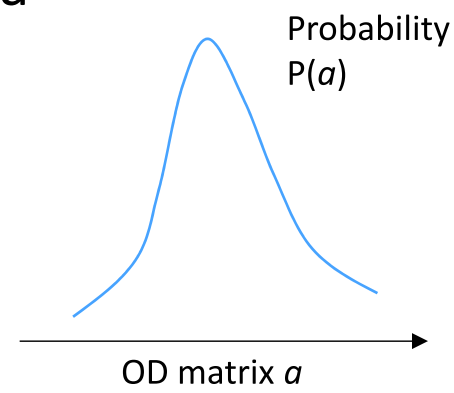
Main Idea: Interval Estimation

Previous studies obtain *one* OD matrix which has the highest likelihood and utilize the OD matrix for planning.

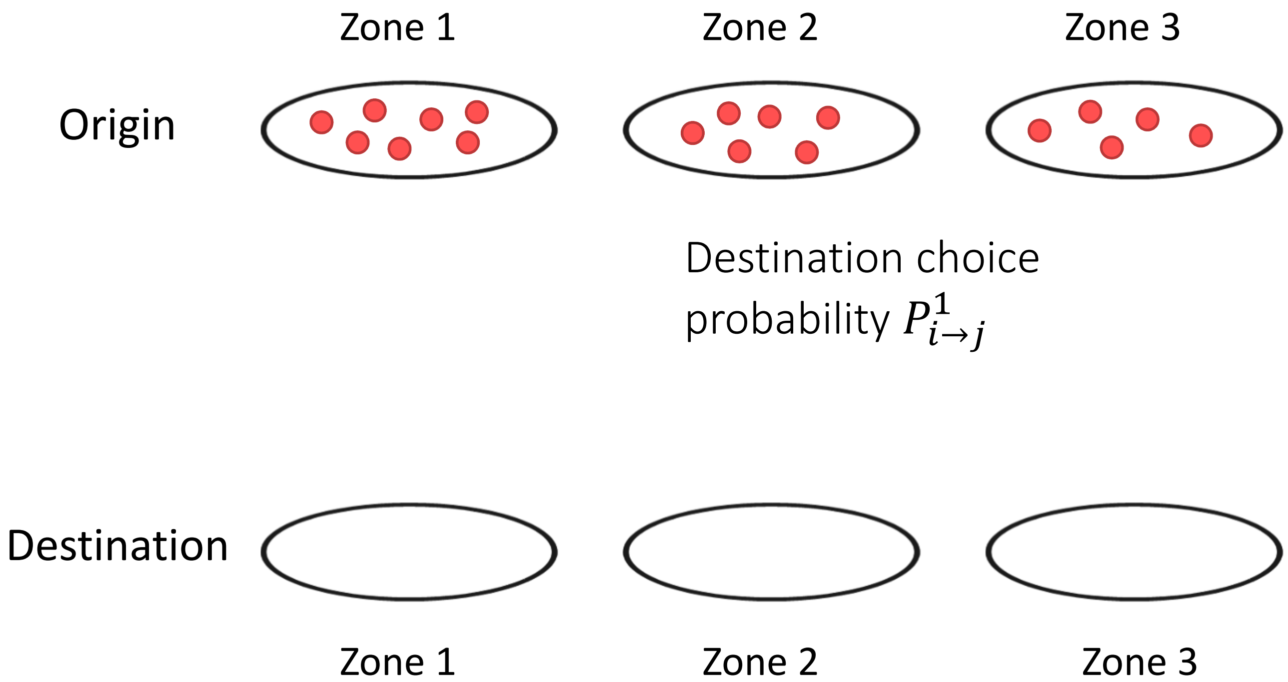


We try to obtain a set of OD matrices which show a distribution of OD matrix.

- Only one estimated OD matrix cannot correspond to a real one.
- A set of calculated OD matrices can contain a real one.
- It is similar to a interval estimation.

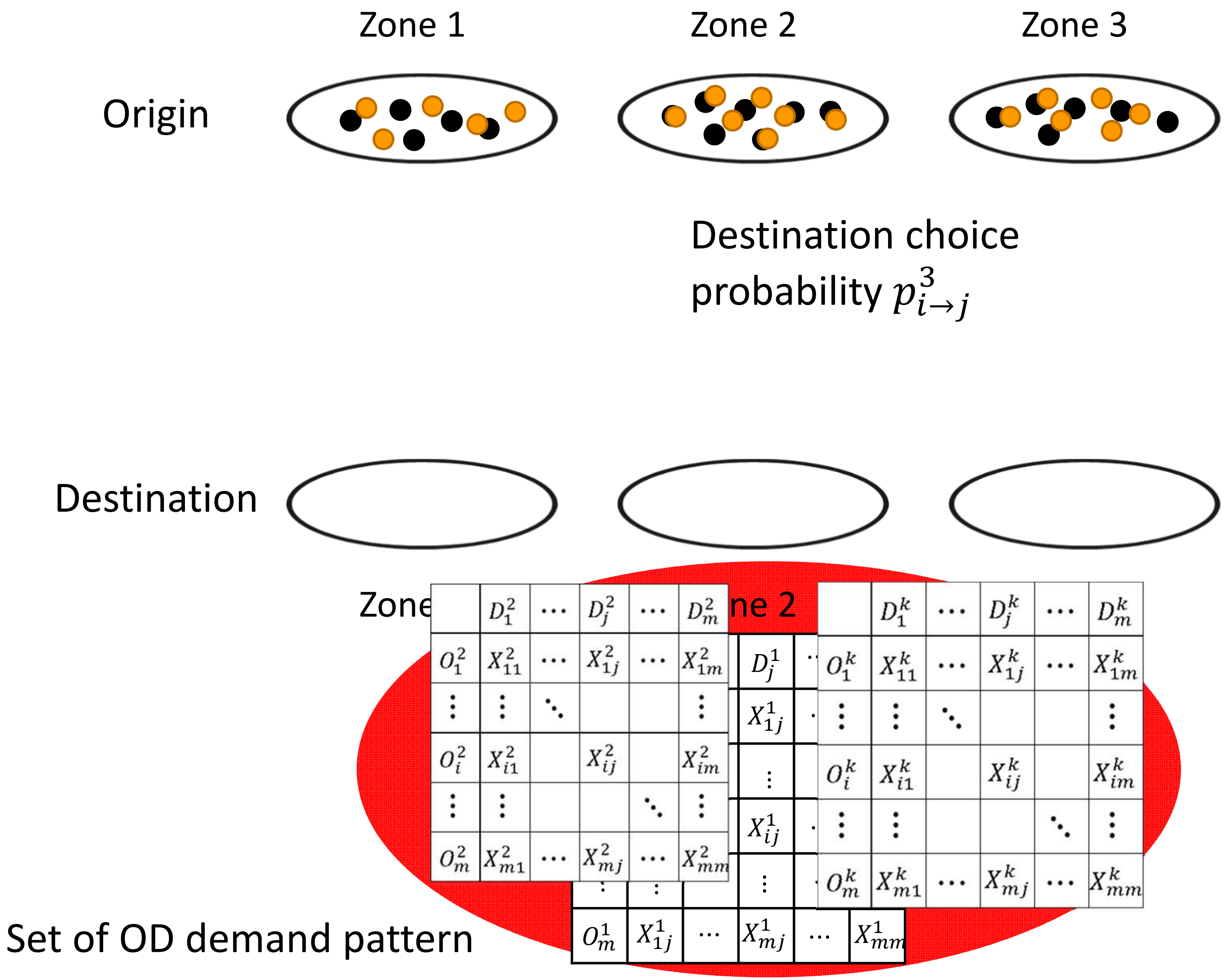


Model & Algorithm:
Image of Generating a set of OD demand pattern



	D^1_1	$\cdots$	D^1_j	$\cdots$	D^1_m
O^1_1	X^1_{11}	$\cdots$	X^1_{1j}	$\cdots$	X^1_{1m}
$\vdots$	$\vdots$	$\ddots$	$\vdots$		$\vdots$
O^1_i	X^1_{i1}	$\cdots$	X^1_{ij}	$\cdots$	X^1_{im}
$\vdots$	$\vdots$		$\vdots$	$\ddots$	$\vdots$
O^1_m	X^1_{1j}	$\cdots$	X^1_{mj}	$\cdots$	X^1_{mm}

Generating a set of OD demand pattern



Model Concept

Wilson's Entropy model

X_{ij}: number of Trip from origin i to destination j
O_i: Total number of trips from origin i
D_j: Total number of trips to destination j
C: Total travel cost

Entropy
$$P(\{X_{ij}\}) = \frac{w(\{X_{ij}\})}{\sum w(\{X_{ij}\})},$$
$$w(\{X_{ij}\}) = \frac{X!}{\prod_{ij} X_{ij}!} \quad (X = \sum_i \sum_j X_{ij}),$$

Constraint
$$\sum_j X_{ij} = \bar{O}_i, \quad \sum_i X_{ij} = \bar{D}_j, \quad \sum_i \sum_j X_{ij} c_{ij} = \bar{C},$$

This model uses minimum external inputs and can connects to discrete choice model

	Entropy model	**Proposed model**
Destination choice	Independent from one another	Affected by spatial correlation
Constraint	Strict on generated and attracted trips	Strict on generated trips and relaxed on attracted trips
Obtained OD pattern	Most likely one	Likely set

Individual a's destination utility
$$U^a_{ij} = u(C_{ij}) + G_j + \eta_{ij}$$
spatial correlation

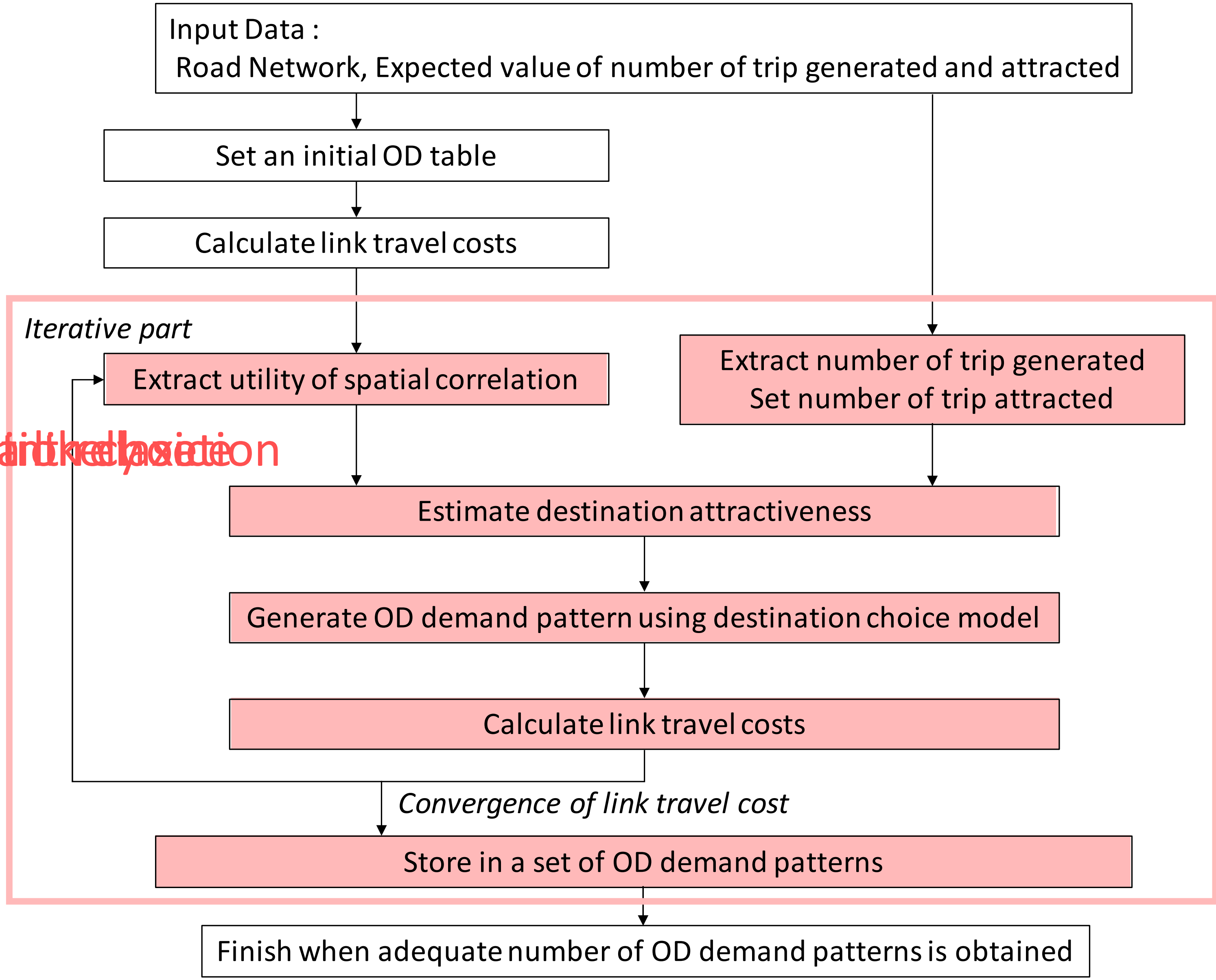
Choice Probability by GEV model
$$P^a_{i \to j} = \frac{\exp U^a_{ij}}{\sum_j \exp U^a_{ij}}$$

Spatial correlation in OD pair ij
$$\eta_{ij} = \nu_{ij} u(C_{ij})$$

Gj: Destination Attractiveness

ν_{ij} : stochastic distribution which average is 0

Sampling Algorithm



Destination Choice: GEV model

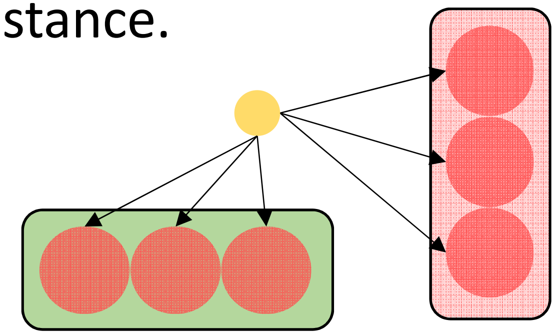
GEV based Destination Choice Model (from zone i to zone j)

$$V_{ij}^{a} = U_{ij}^{a} + \epsilon_{ij}^{a}, \quad P_{i\to j}^{a} = \frac{\exp U_{ij}^{a}}{\sum_j \exp U_{ij}^{a}}$$

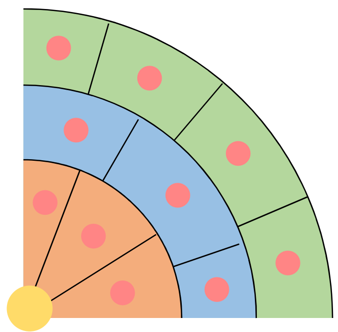
Error correlation between destination by network GEV model

(Daly & Bierlaire(2006))

- Destinations are divided as mesh. A number of destination is a lot.
- Network-GEV model can formulate error correlation flexibly.
  - This numerical example introduce the error correlation between adjacent destinations and between destinations of similar distance.



Attribution(State)-based correlation



Distance-based correlation

Destination Choice: Spatial Correlation

Main idea for obtaining a likely set

Introduce stochastic utility η_{ij} which shows spatial correlation and unobserved variations

$$U_{ij}^{a} = u(C_{ij}) + G_j + \eta_{ij}$$

Travel cost function   Destination attractiveness   Stochastic utility

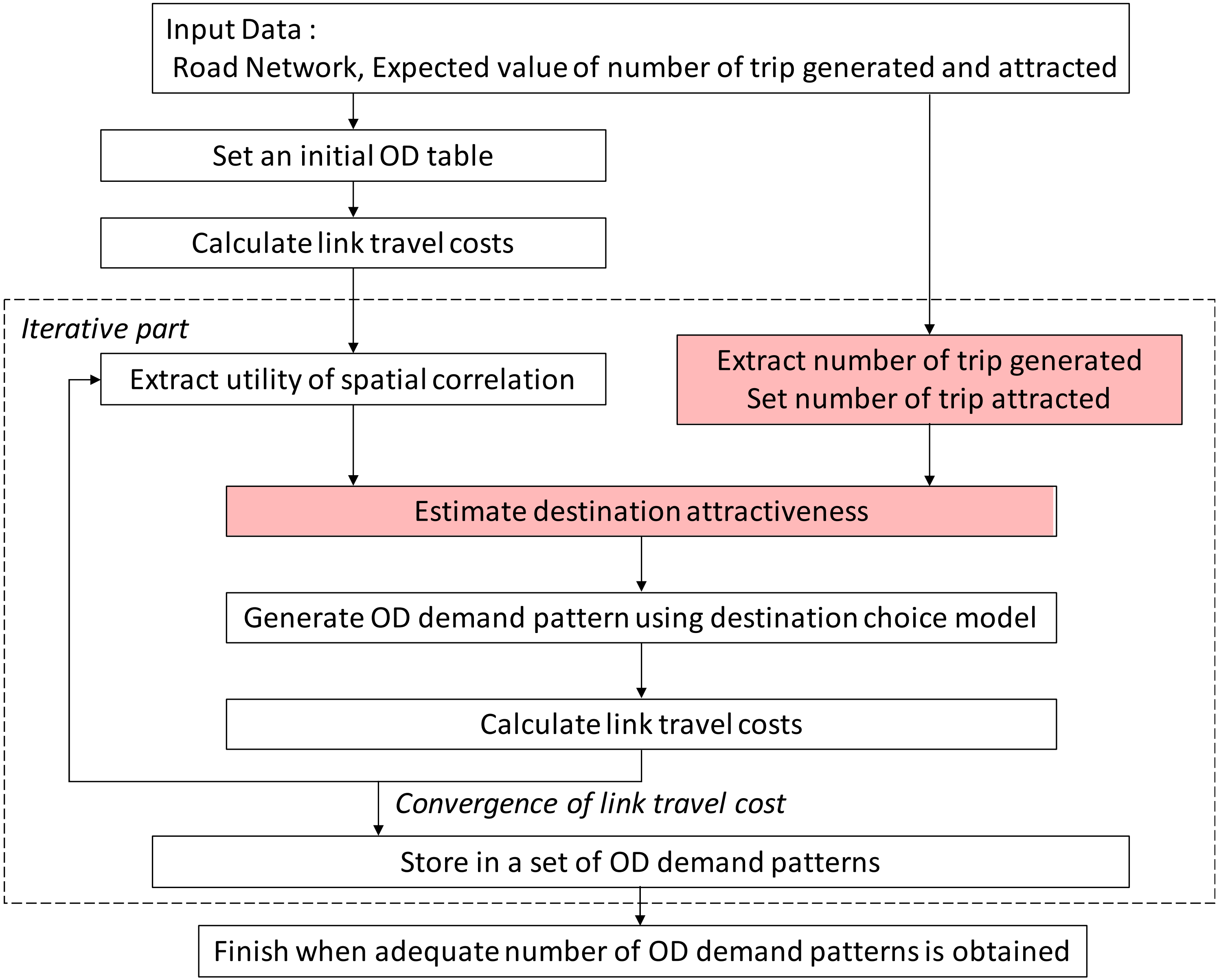
Spatial correlation is proportional to travel utility

$$\eta_{ij} = \nu_{ij} u(C_{ij})$$

ν_{ij}: stochastic distribution with an average equal to 0

- The spatial correlation from stochastic term η_{ij} influences OD pair ij only.
- This stochastic utility leads to our model's advantage: variation in OD demand patterns.
- A final set of OD demand patterns has intervals of OD traffic volumes and these intervals show variations.

Sampling Algorithm



Relaxed Constraints

- Extract number of trip generated O_i from a distribution which average equal to expected value E(O_i) which are exogenously given.
- O_i is sampled by a Poisson process because people depart independently

$$\mathrm{P(X{=}O}_i) = \frac{\lambda^{O_i} e^{-\lambda}}{O_i!}, \ (\lambda = E(O_i))$$

- The sum of the number of trips attracted D_j' can be expanded to the sum of O_i simply.

$$\mathrm{D'}_j = \frac{\sum_i O_i}{\sum_j E(D_j)} E(D_j)$$

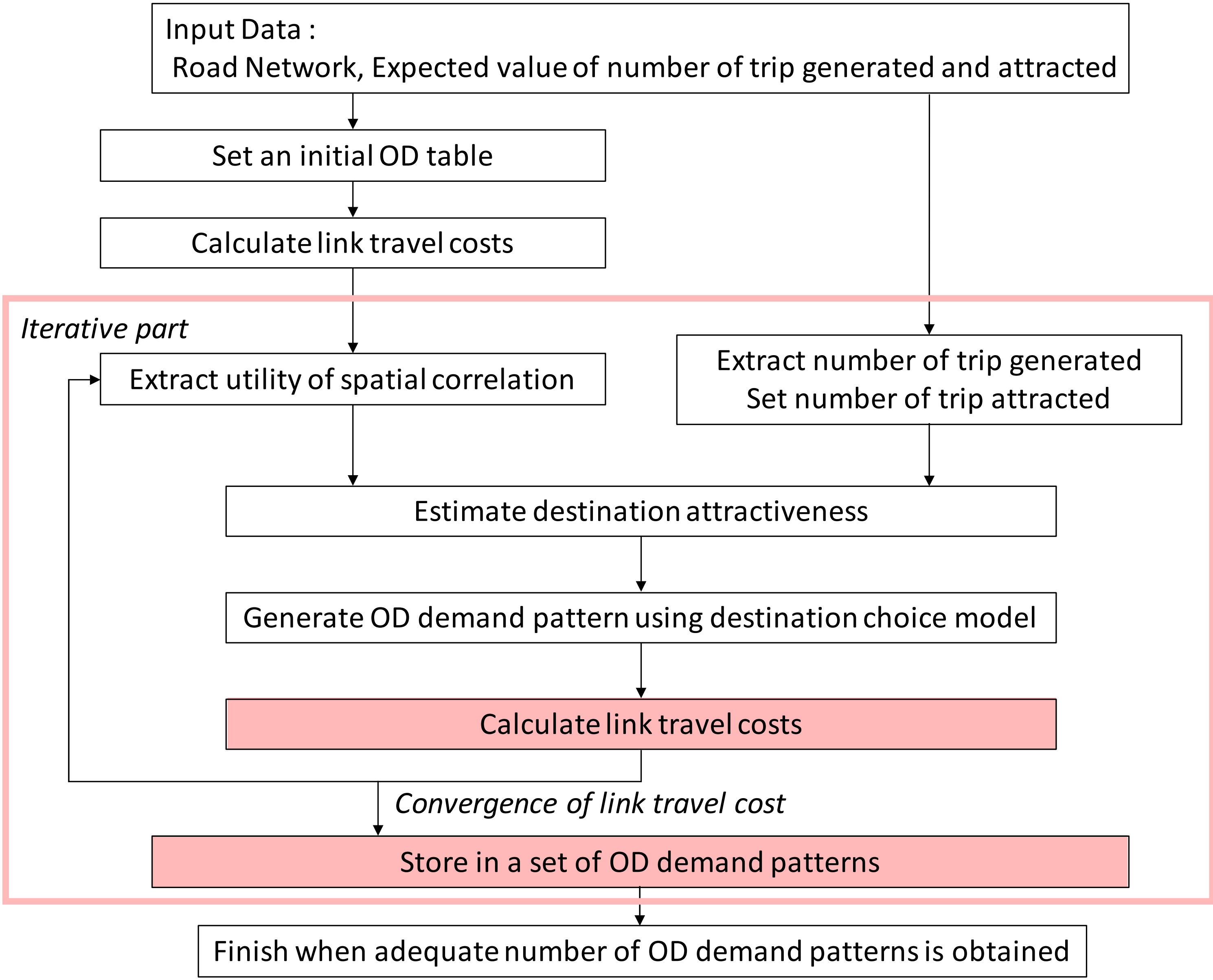
Estimation of Destination attractiveness

Destination Attractiveness $\{G_j\}$ are estimated to fit to the constraint of the number of trips attracted:

$$\min_{\{G_j\}} \mathrm{MSE}(\{G_j\}) = \sum_{\forall j} \left(D'_j - \sum_{\forall i} O_i P_{i\to j}(\{G_j\}) \right)^2$$

- Attractiveness $\{G_j\}$ is estimated so that D'_j matches the number of trips attracted in a generated OD demand pattern X_{ij} using the destination choice probability $P_{i\to j}$.
- The estimation process refers to the trip-end conditions of the entropy model
- The estimation employs the least square method

Sampling Algorithm



Obtain a set of OD demand pattern

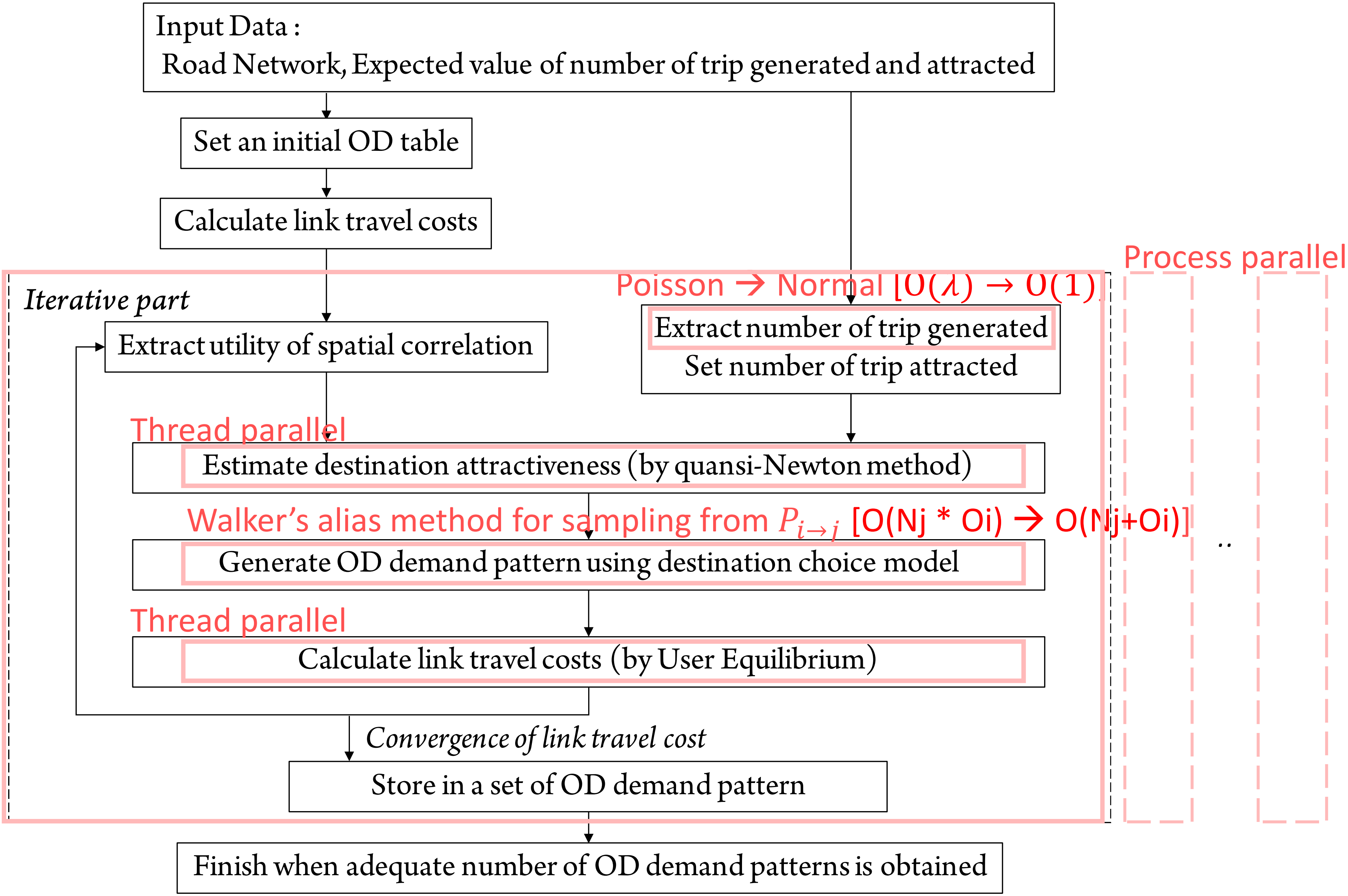
A suitable OD demand patterns is generated using a Monte Carlo approach with the destination choice probability.

- The destination selection process is applied to all people who are leaving.
- At every origin *i* on iteration *k*,
  - Sample a stochastic utility η_{ij}^{k} and calculate $P_{i \to j}^{k}$
  - Destinations of O_i^k people is determined according to the destination choice probabilities $P_{i \to j}^{k}$ from her/his origin *i*.
- Generate a OD demand pattern $\{X_{ij}^{k}\}$

A generated OD demand pattern is added to a set of OD demand patterns if the link travel costs c_{rs}^{k} converge in fully converged algorithm.

- The link travel costs on a generated OD demand pattern are calculated by static user equilibrium assignment.
- Convergence test: $\dfrac{\sum_{\forall rs} \frac{|c_{rs}^{k} - c_{rs}^{k-1}|}{c_{rs}^{k}}}{N_{rs}} \leq \delta$

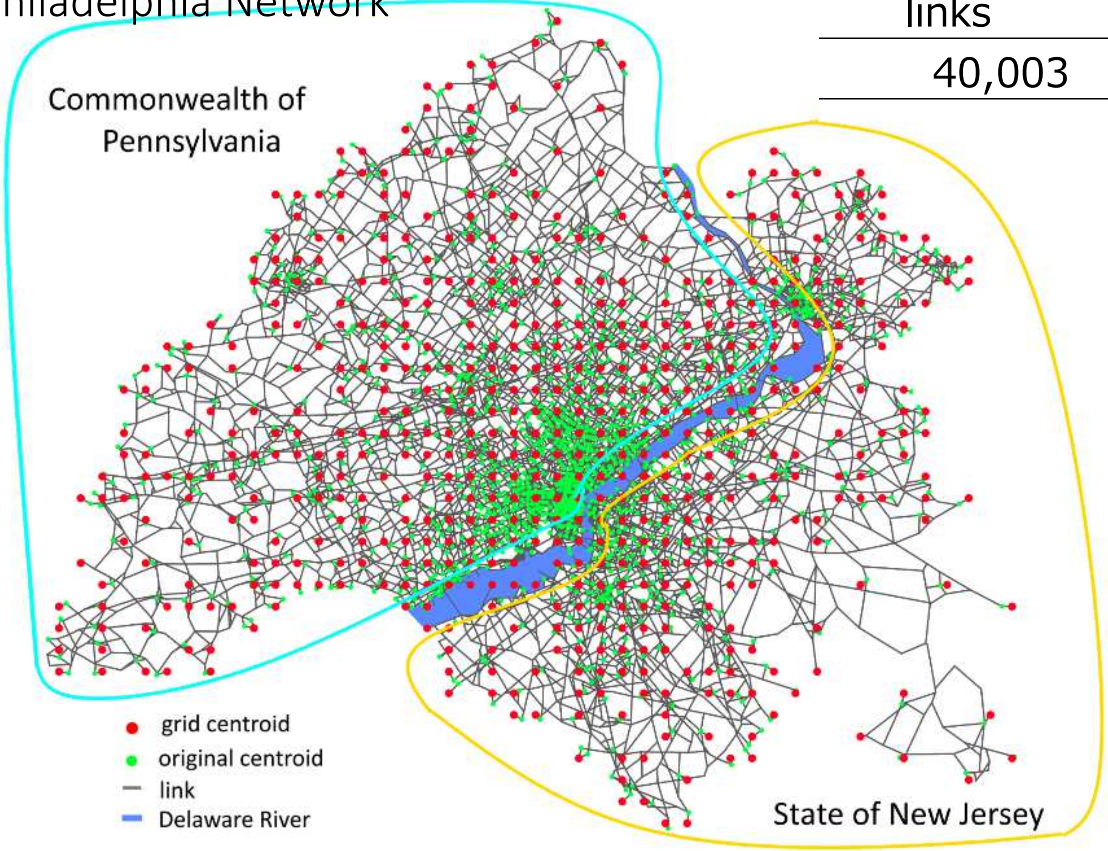
Decreasing computation cost



Numerical example: Test Network

- One of the large-scale networks by Bar-Gera
- OD volume is integer.
- Boyce et al(2004) used this network & OD for practical analysis.

Philadelphia Network

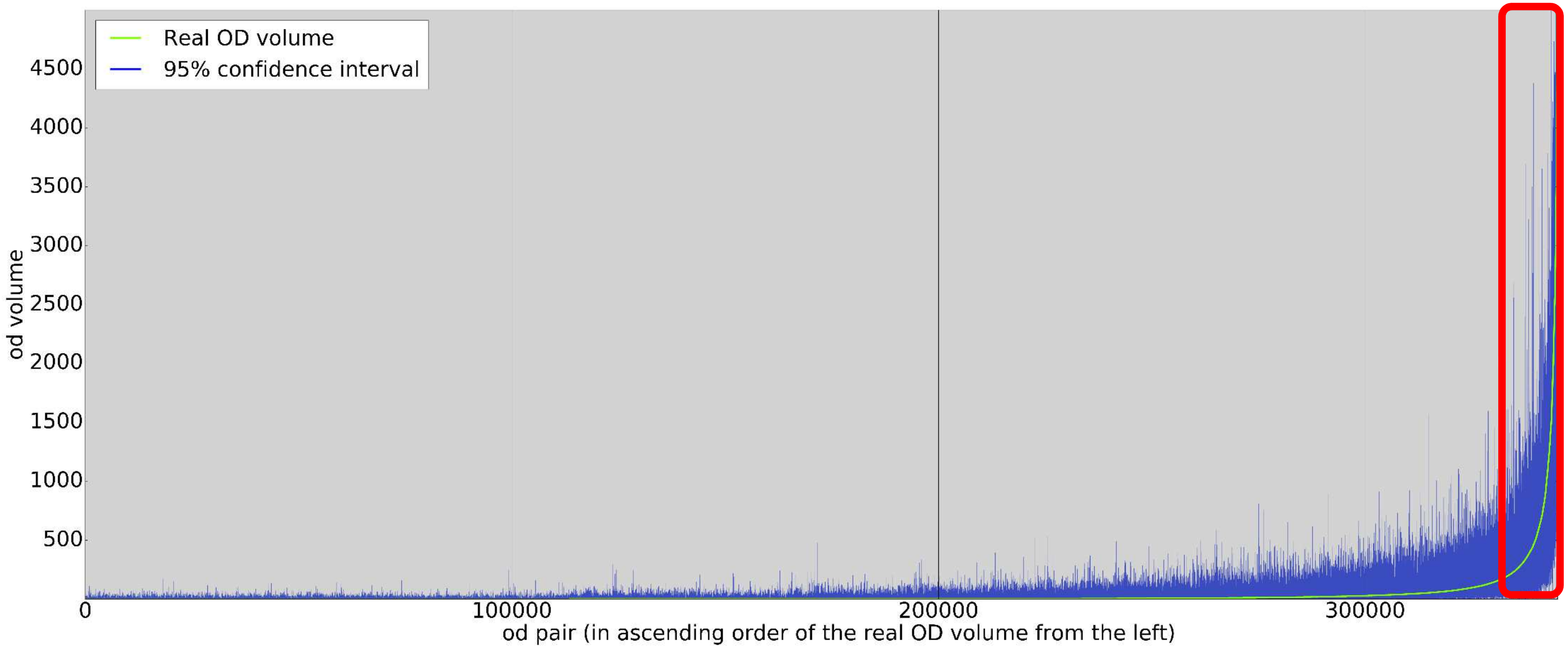


links	nodes	OD trips	zones
40,003	13,389	18,503,872	588*

* Original data - 1,489 centroids

Numerical example: Verification Results

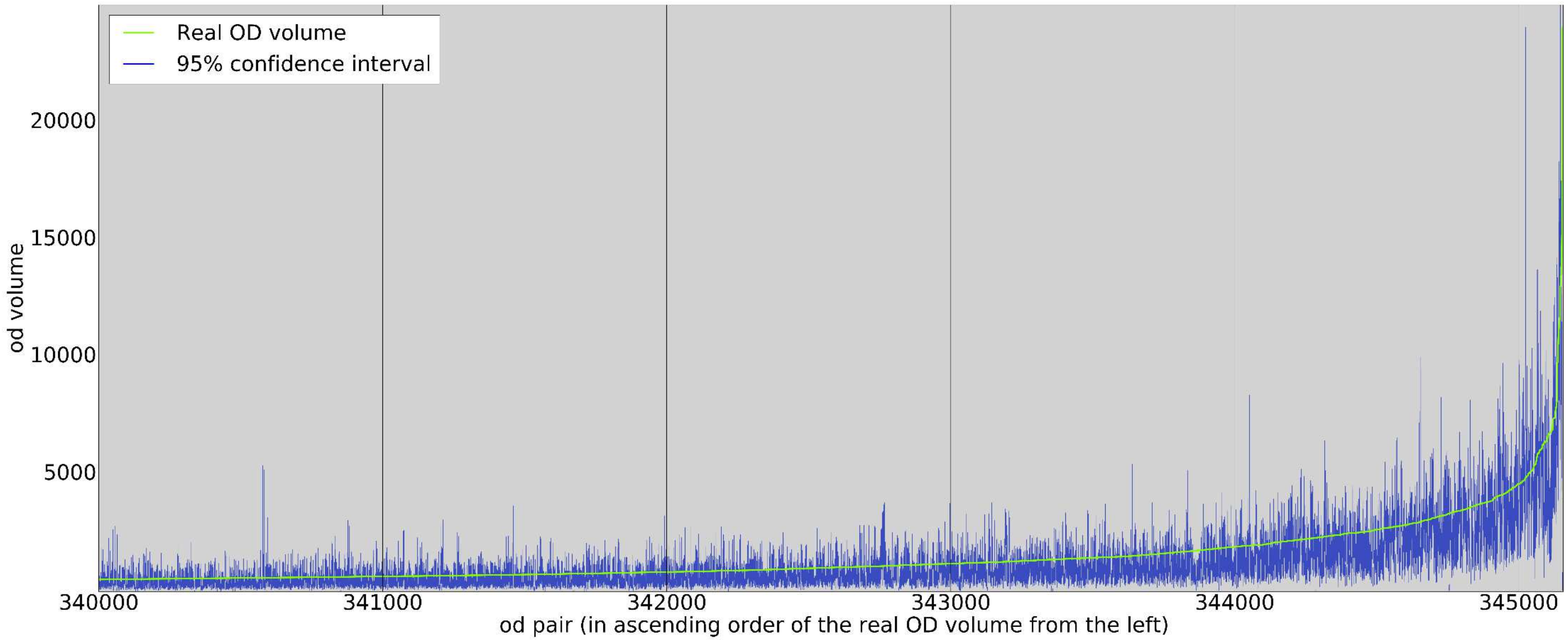
Enlarged
in the next slide



Confidence interval of the OD volume (up to 5,000 vehicles and 10,000 samples)

- The sampled OD volumes increase as the real OD volume increases.
- Almost all the real OD volume is included in the 95% confidence interval.

Numerical example: Verification Results



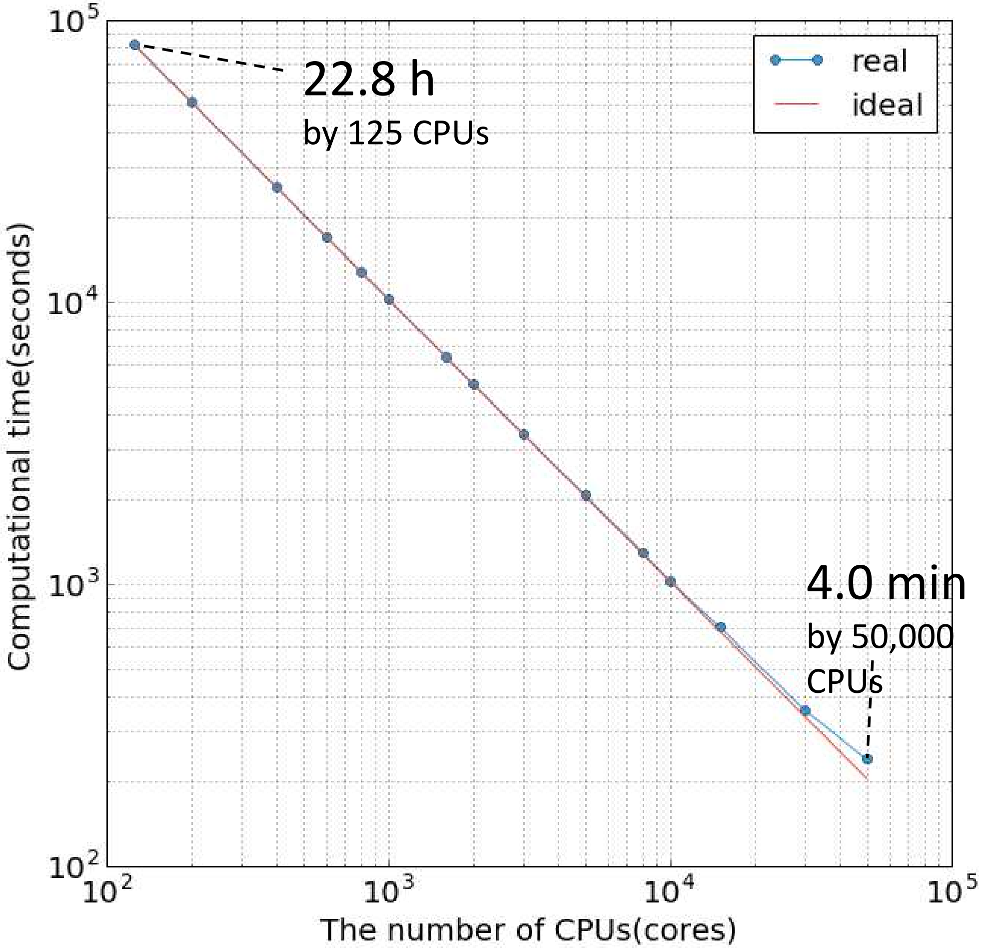
Confidence interval of the OD volume (enlarged view after the 340,000 th OD volume)

	All	Low cost[1]	Moderate cost[2]	High cost[3]
95% Interval(%)	90.3	80.1	83.8	90.5
RMSE	126.3	988.3	365.1	58.8

[1]OD travel cost $C_{ij} < 10$; [2]$10 \leq C_{ij} < 15$; [3]$15 \leq C_{ij}$

Computational time by HPC

Speed-up ratio by the sampling algorithm



number of sample:10,000,000

Calculated by Super Computer K

Because each sampling can be performed individually, the scalability is sufficiently high.

Ideal computational time

$$T(p) = \frac{T}{p} \cdots (13)$$

T: computational time by 125 CPU cores

$T(p)$: ideal computational time by p CPU cores

Take-home message

- Optimization process in large-size network has to evaluate many cases.
- Calculation time of lower problem which includes dynamic traffic assignment and activity simulator should be less.
- In lower problem, obtaining dynamic OD demand pattern should be fit dynamic travel time. The collaboration with dynamic traffic flow simulator and dynamic demand simulator is needed.
- One possible approach to reduce calculation time is using High performance computation technique.
- Suitable algorithm for HPC technique is needed for getting highly scalable results.
- The OD demand study proposes an approach for obtaining a set of sampled possible OD demand patterns that includes variation.

References

Turek, Erlang-based desynchronized urban traffic simulation for HPC systems, Future Generation Computer Systems, Vol. 79, pp. 645-652, 2018.

Nagel, K., Rickert, M.: Parallel implementation of the TRANSIMS micro-simulation, Parallel Computing, Vol. 27(12), pp. 1611-1639, 2001.

Thulasidasan, S., Eidenbenz, S.: Accelerating Traffic Microsimulations: A Parallel Discrete-event queue-based approach for speed and scale, Proceedings of the 2009 Winter Simulation Conference, IEEE, pp. 2457-2466, 2009.

Iryo, T., Urata, J., Fukuda, K., Peque, G. Jr., Lalith, W., Petprakob, W.: Fast calculation of dynamic traffic assignment by parallelised network loading algorithm, ISTS & IWTDCS 2018, Ehime University, Matsuyama, Japan, August 4-6, 2018.

Kawase, R., Urata, J., Iryo, T., Sampling approach on spatial variation for travel demand forecasting, 6th Symposium of the European Association for Research in Transportation (hEART 2017), No. 143, Haifa, Israel, September 12-14, 2017.