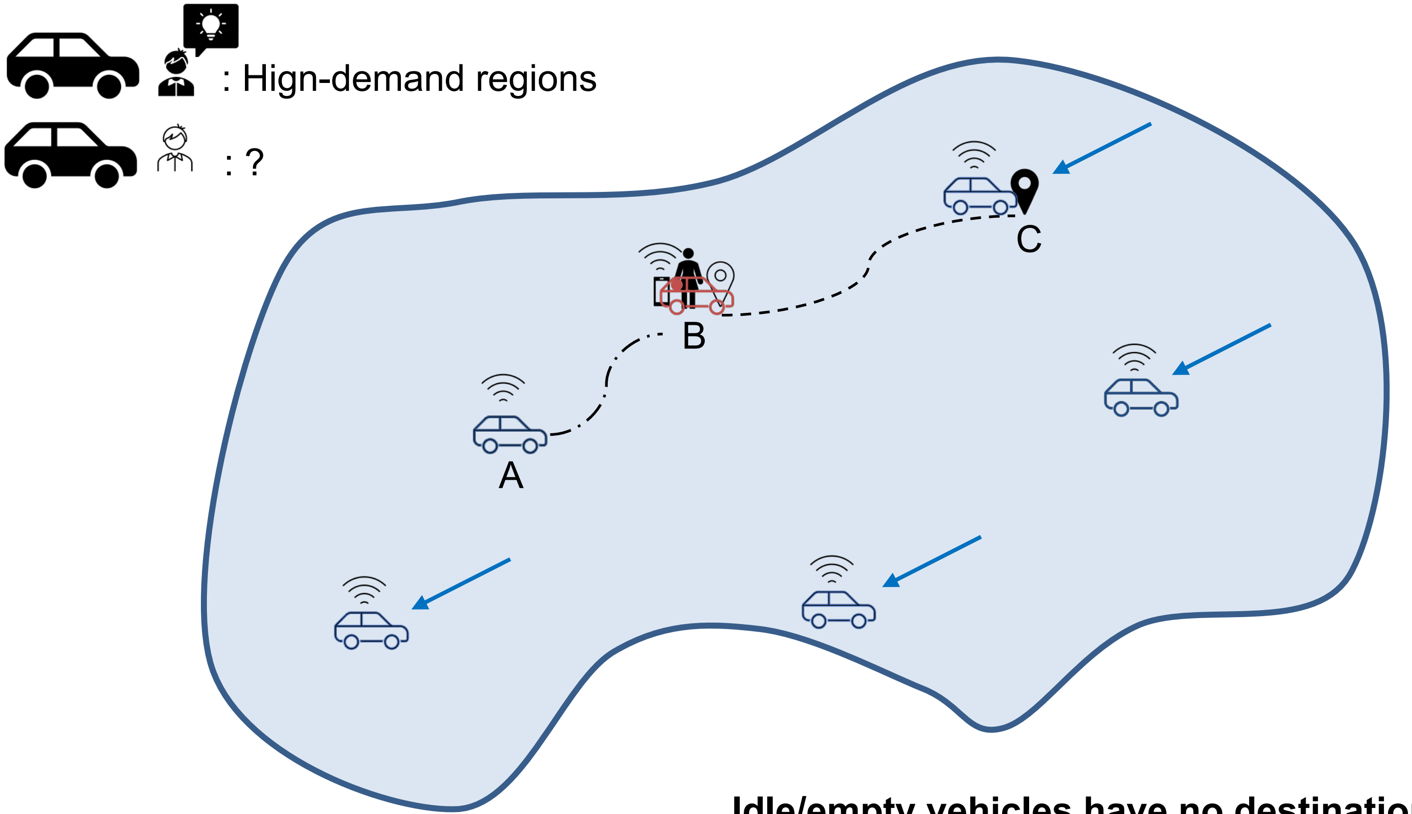
Rebalancing Idle Vehicles via Distributed Coverage Control in Mobility-on-Demand systems

Pengbo Zhu, Isik Ilber Sirmatel, Giancarlo Ferrari Trecate, Nikolas Geroliminis
Urban Transport Systems Laboratory (LUTS)
EPFL

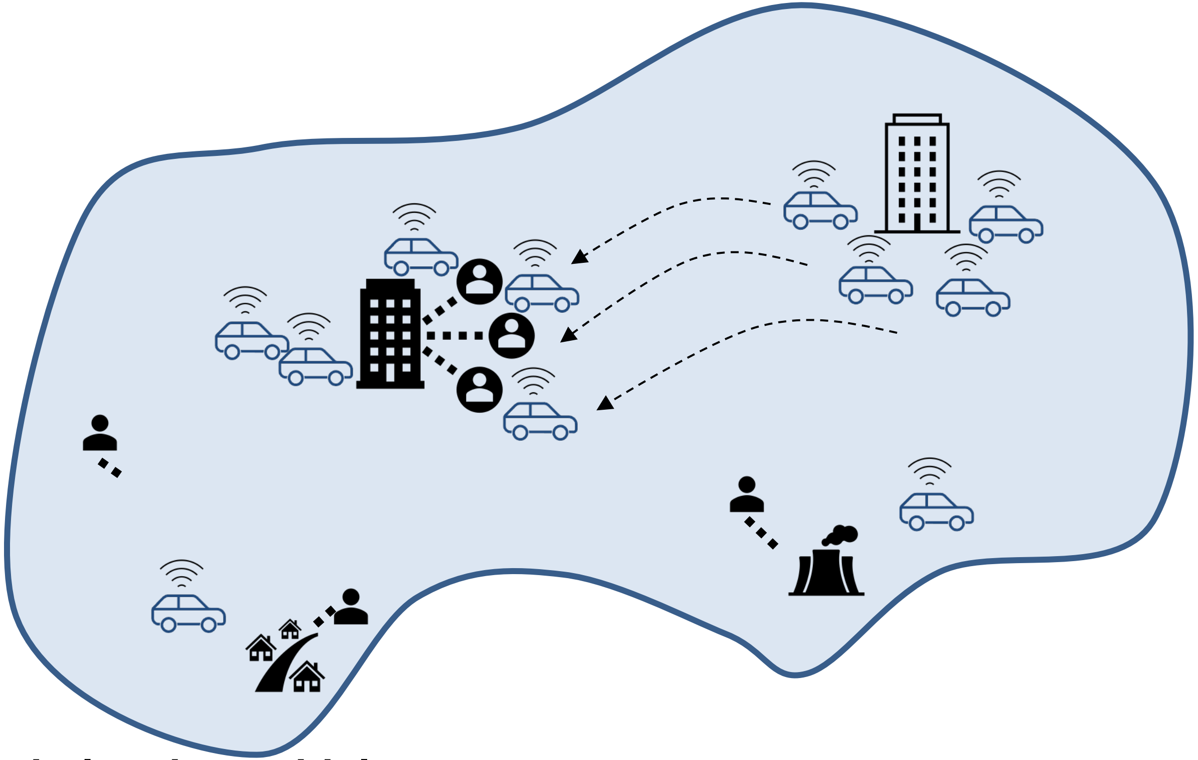
National Centre of Competence in Research Automation



Idle/empty vehicles have no destinations.
Where should they go?

Imbalance in the spatial distribution of vehicles:

- Non-uniform passenger's demand for rides in different districts
- Asymmetry between origin and destination distributions of trips



Goal: rebalancing vehicles

- Relocating idle/empty vehicles to the high-demand regions

Problem Formulation

Vehicle rebalancing problem

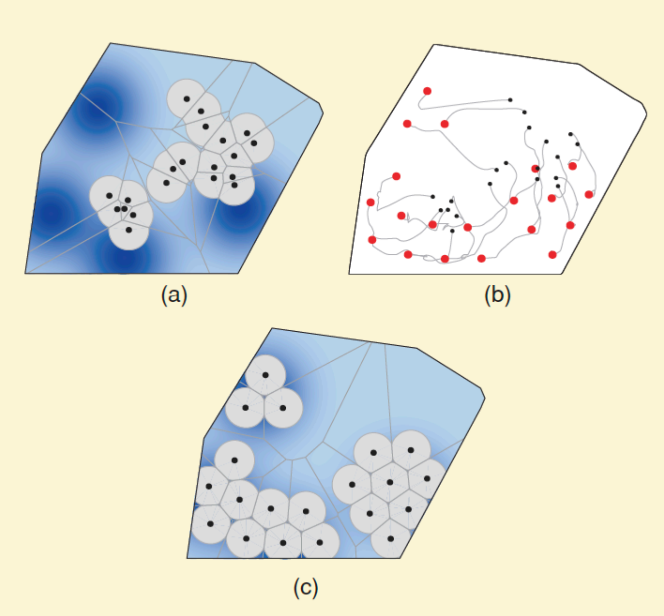
Coverage control problem:

Every agent/vehicle is responsible for covering a certain area

$$H(X,W) = \sum_{i=1}^{n} H(x_i, W_i{}') = \sum_{i=1}^{n} \int_{q \in W_i'} f(\|x_i - q\|^2)\varphi(q)dq$$

$\varphi(q)$: demand density function,

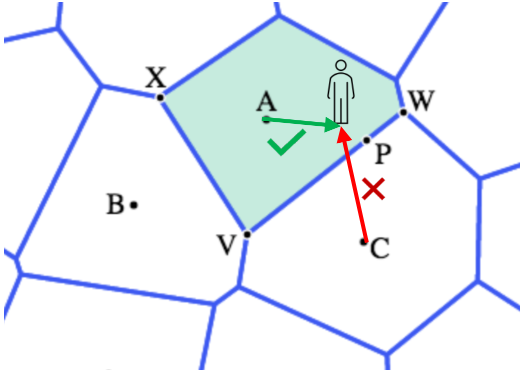
$f: [0, \infty) \rightarrow \mathcal{R}$,a performance function which degrades with distance.



Ref: S. Martinez(2007), J. Cortes and F. Bullo, "Motion Coordination with Distributed Information"

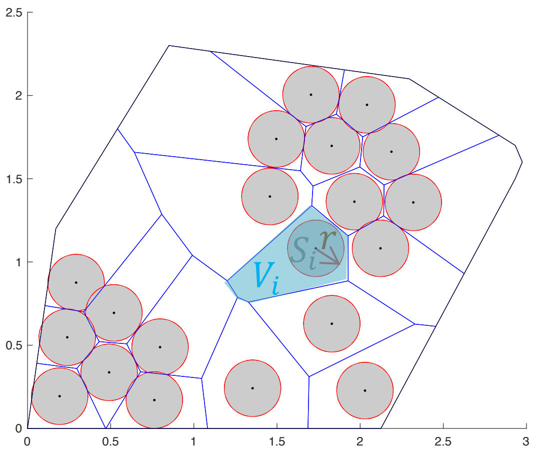
Voronoi partition:

- The partitioning of a plane with n points into covex polygons
- Each cell contains one generator/seed
- Every point in a given cell is closer to its seed than to any other



  - With Voronoi diagram, we can disperse the vehicles in the region

Coverage Control Algorithm



Important variables :

- N = Number of agents
- r = Agent coverage radius

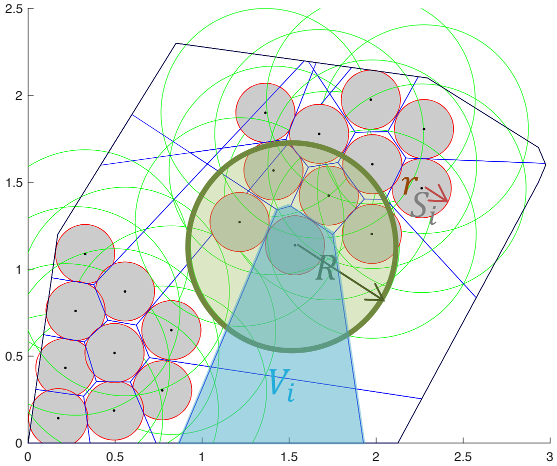
Objective :

Maximize the value of H

$$H(X,W) = \sum_{i=1}^{n} H(x_i, W_i) = \sum_{i=1}^{n} \int_{q \in W_i} f(\|x_i - q\|^2)\varphi(q)dq$$

$$W_i = S_i \cap V_i$$

Distributed Coverage Control Algorithm



Important variables :

- N = Number of agents
- r = Agent coverage radius
- **R = Agent communication limitation radius**

Objective :

Maximize the value of H

$$H(X, W') = \sum_{i=1}^{n} H(x_i, W_i') = \sum_{i=1}^{n} \int_{q \in W'_i} f(\|x_i - q\|^2)\varphi(q)dq$$

$$W_i' = S_i \cap V_i'$$

Distributed Coverage Control Algorithm

Proposition: The local maximum of H can be obtained when all x_i are located at centroids (centers of mass, $C_{W_i'}$) of their respective Voronoi cells (W'_i), i.e., ***Centroidal Voronoi Configuration(CVC)***.

Distributed Control Law Formulation:

$$\frac{dx_i}{dt} = u_i, \ \frac{\partial H}{\partial x_i} = -2M_{W_i'} \left\| x_i - C_{W_i'} \right\|,$$

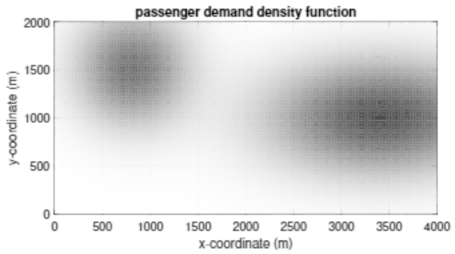
$$\boldsymbol{u_i = -k_p \left(x_i - C_{W_i'} \right)},$$

$$\frac{\partial H}{\partial t} = \sum_{i=1}^{n} \frac{\partial H}{\partial x_i} \frac{dx_i}{dt} = 2k_p \sum_{i=1}^{n} M_{W_i'} \left\| x_i - C_{W_i'} \right\|^2 > 0.$$

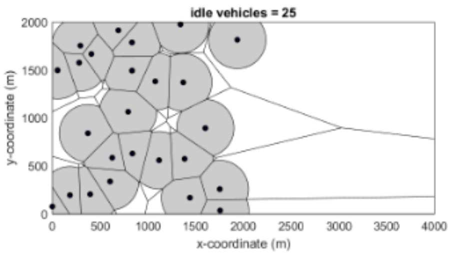
which steer the agent team to converge to CVC.

Ref: Du, Q., V. Faber and M. Gunzburger (1999) *Centroidal voronoi tessellations: Applicationsand algorithms*

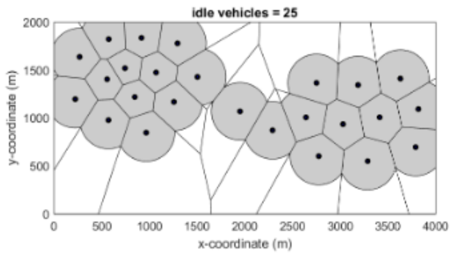
Case Study I: Continuous Case



(a) Demand density function

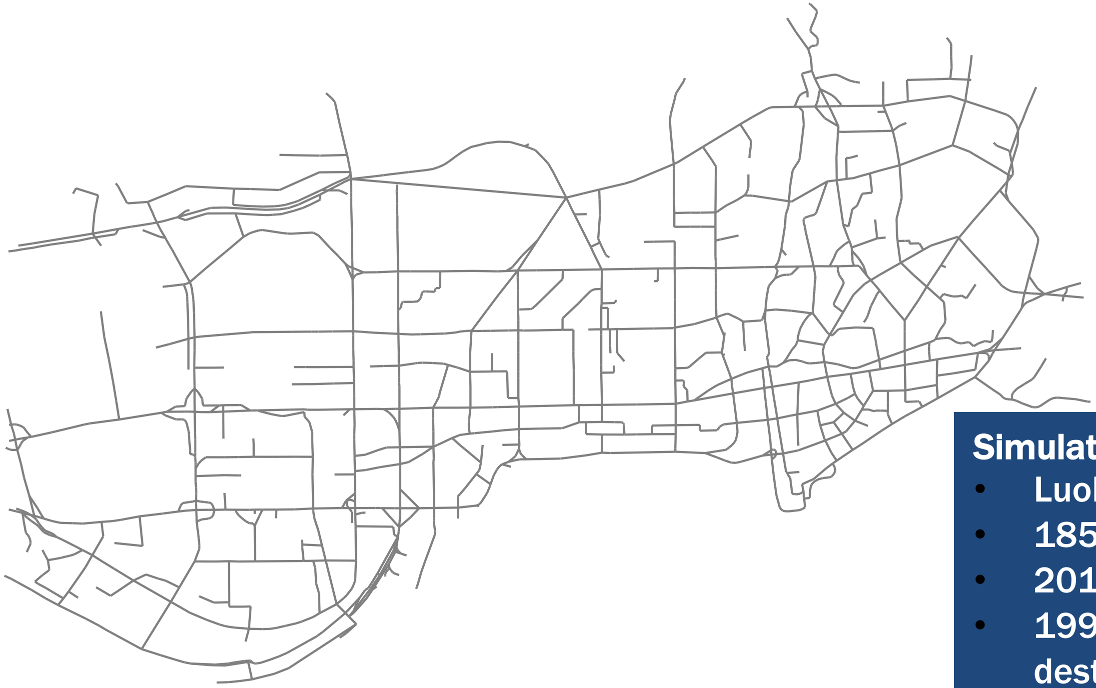


(b) Initial configuration



(c) Final configuration

Case Study II: Shenzhen, China



Simulated network:

- Luohu District of Shenzhen, China
- 1858 nodes
- 2013 links
- 199,819 trips consisting of origins, destinations, and time

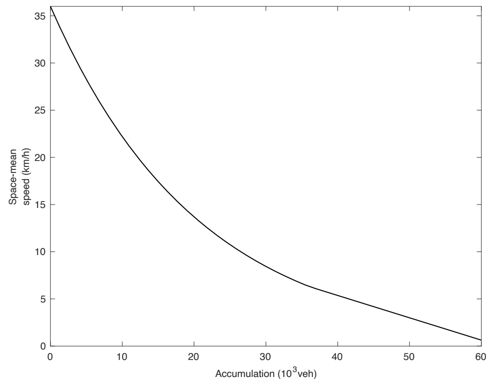
Ref: Beojone, Caio & Geroliminis, Nikolas. (2020). *On the inefficiency of ride-sourcing services towards urban congestion.*

Experimental Setup

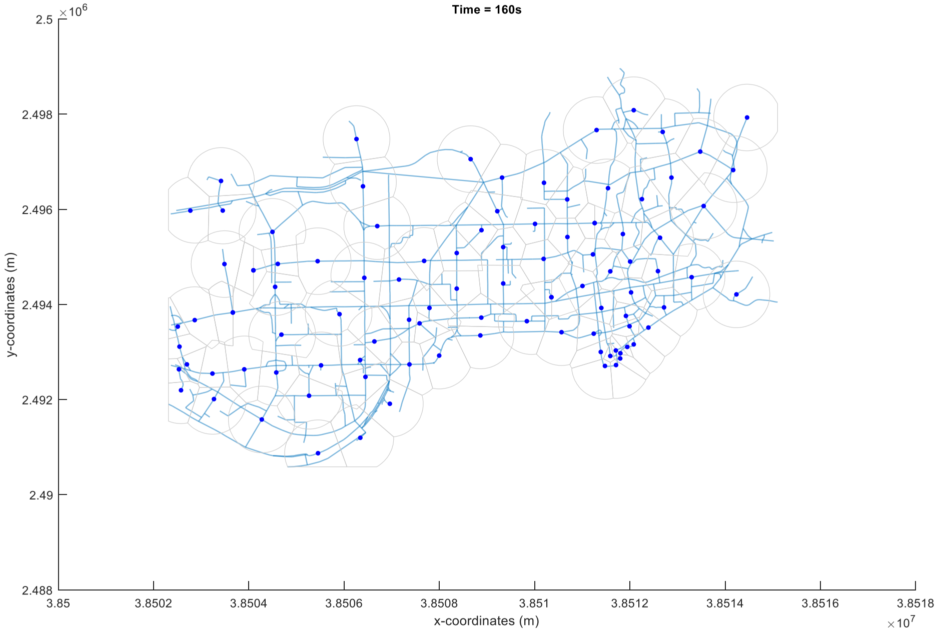
- 3-hour simulation
- Time Pattern of demand: low-high-low, each period lasts for 1 hour
- 2400 orders
- Fleet size = 100

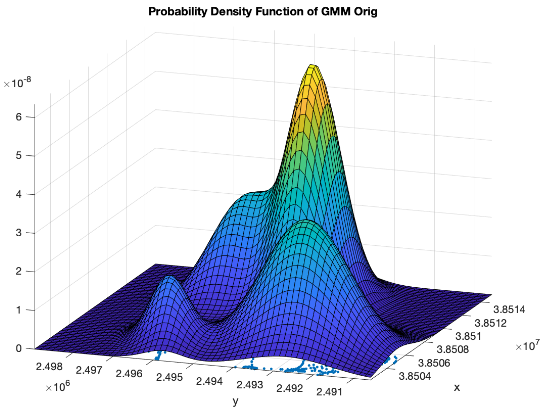
- Private vehicles; Ride-hailing vehicles
- macroscopic fundamental diagram (MFD) for Shenzhen

$$v(n) = \begin{cases} 36e^{(\frac{29m}{600})}, & if \quad m \leq 36, \\ 6.31 - 0.28(m-36), & if \quad 36 < m \leq 60, \\ 0, & if \quad m > 60 \end{cases} \quad where \quad m \equiv \frac{n}{1000}.$$

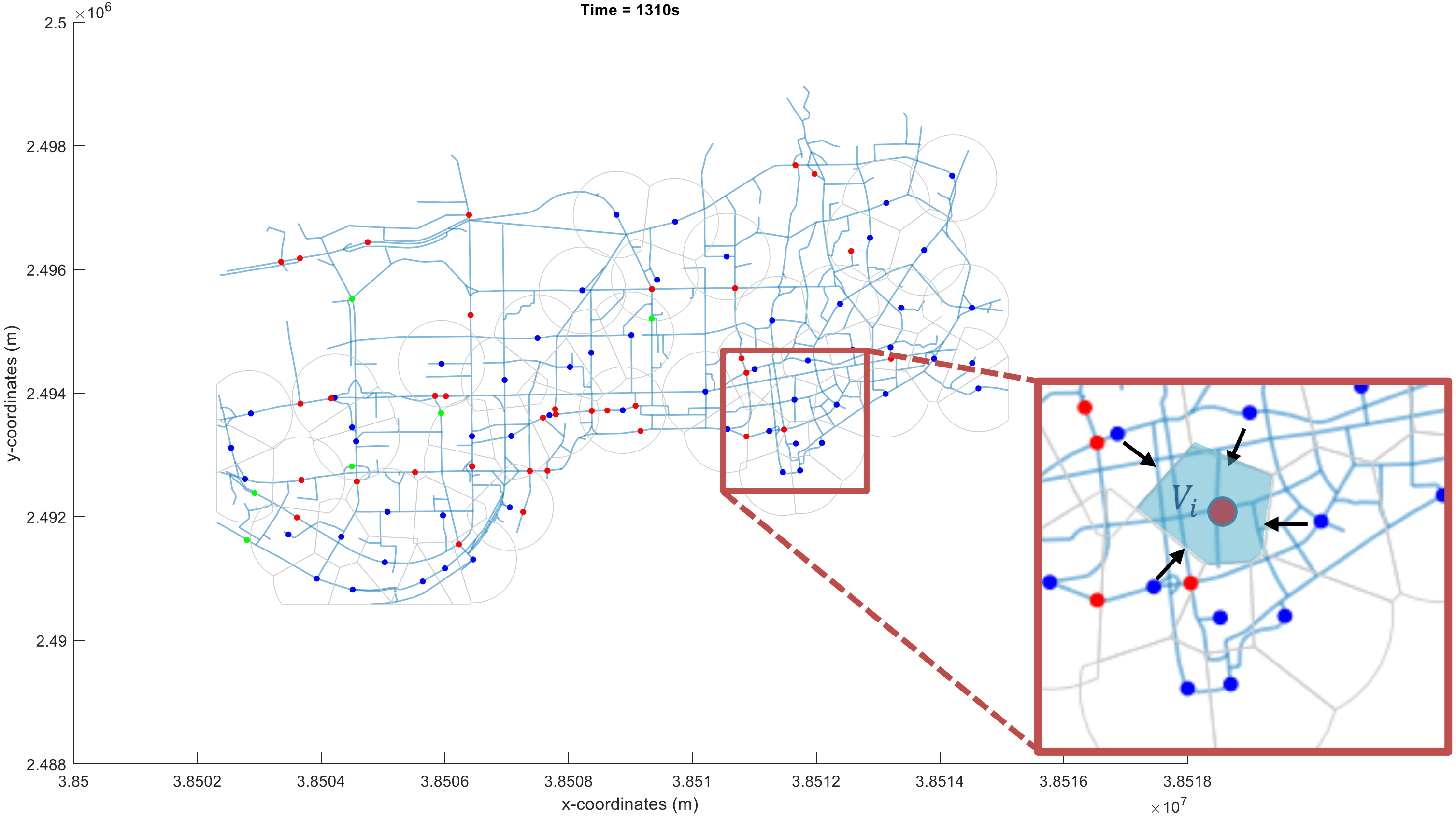


Ref: Ji Y, Luo J, Geroliminis N(2014). *Empirical Observations of Congestion Propagation and Dynamic Partitioning with Probe Data for Large-Scale Systems.*


Time = 160s
y-coordinates (m)
x-coordinates (m)

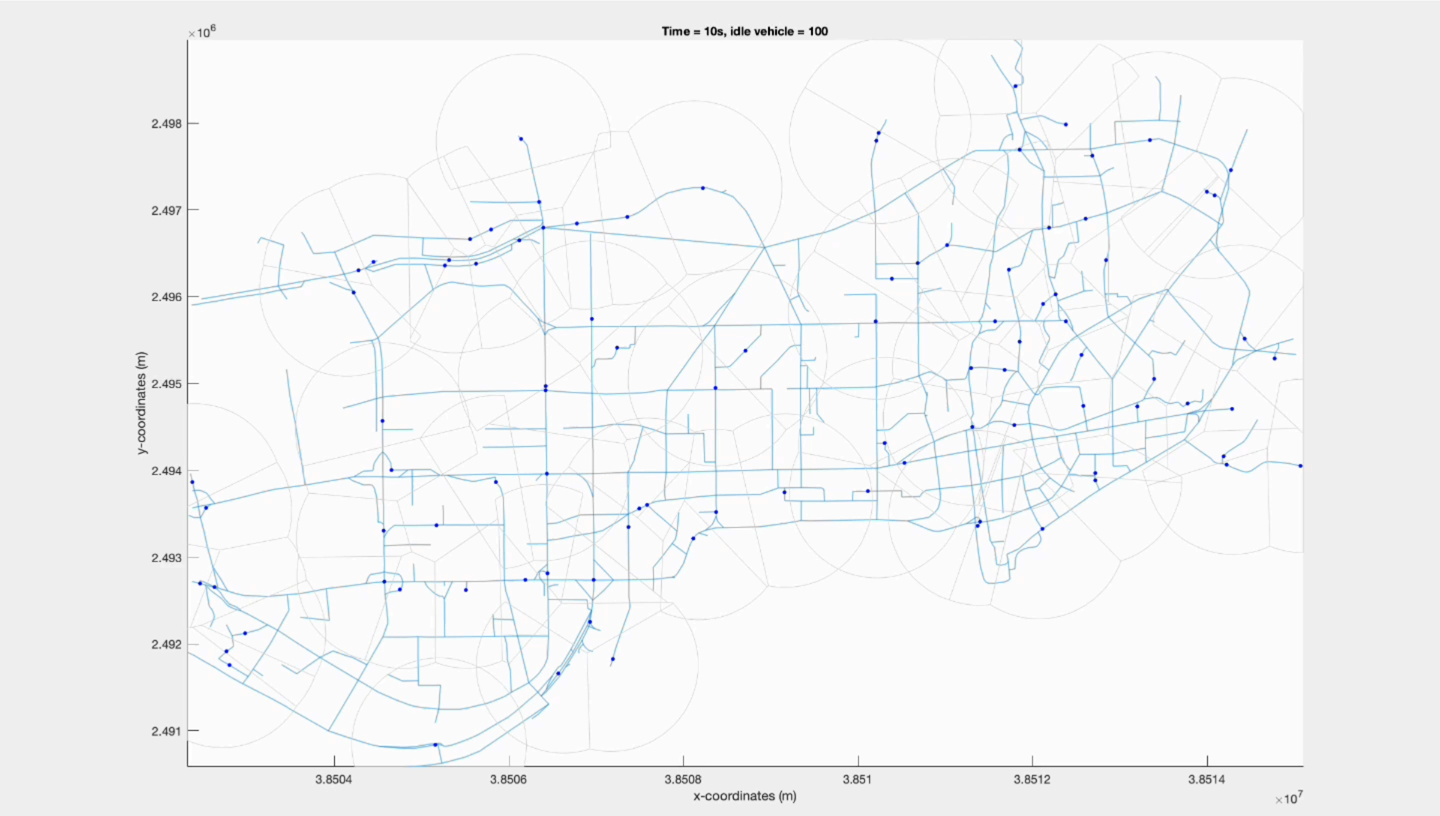

Probability Density Function of GMM Orig
y
x

Real map: Shenzhen city



blue: idle vehicle (i.e., empty, looking for a passenger),
red: passenger-assigned vehicle,
green: passenger-carrying vehicle.

Demo video: Shenzhen city



blue: idle vehicle (i.e., empty, looking for a passenger),
red: passenger-assigned vehicle,
green: passenger-carrying vehicle.

Performance metrics

- Order completion rate: $\frac{N_1}{N} = \frac{N_1}{N_1 + N_2} \times 100\%$

- Average Waiting time: $t_w = \frac{\sum_{i=1}^{N_1}(t_p^r(i) - t_o(i))}{N_1}$

- Average System time(with penalty for cancelled orders):

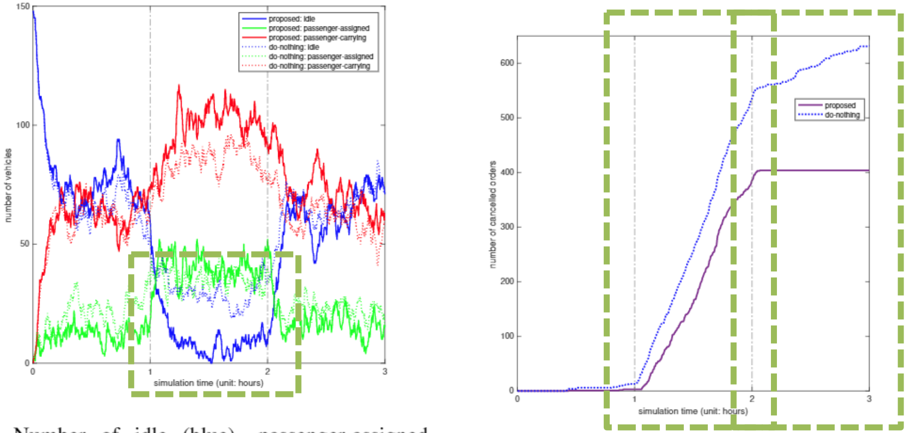
$$t_{sys} = \frac{\sum_{i=1}^{N_1}(t_p^r(i) - t_o(i)) + N_2 \cdot \alpha \cdot w_{tol}}{N}$$

$$\alpha = 1.5, w_{tol} = 5\ min$$

- 3-hour simulation
- Time Pattern of demand: low-high-low, each period lasts for 1 hour
- 2400 orders
- Fleet size = 150

	Proposed method	**Do-nothing policy**	**Improvement**
Completion rate(%)	82.9	73.2	13.3%↑
Average waiting time(s)	132.5	173.9	23.8%↓
Average system time(s)	186.8	247.9	24.6%↓

Fig : Comparison of different states of vehicles and number of canceled orders

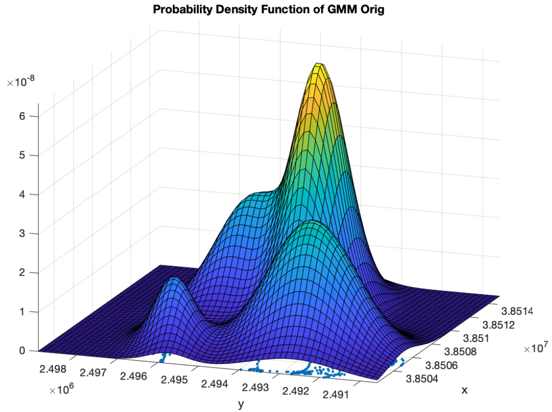


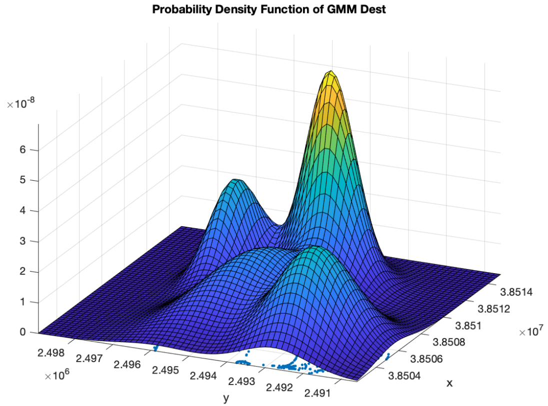
(a) Number of idle (blue), passenger-assigned (green), and passenger-carrying (red) vehicles

(b) Number of accumulated canceled orders

- Operate the fleet more efficiently as a larger amount of vehicles are actively serving passengers

Depict imbalance between the origin and destination distributions





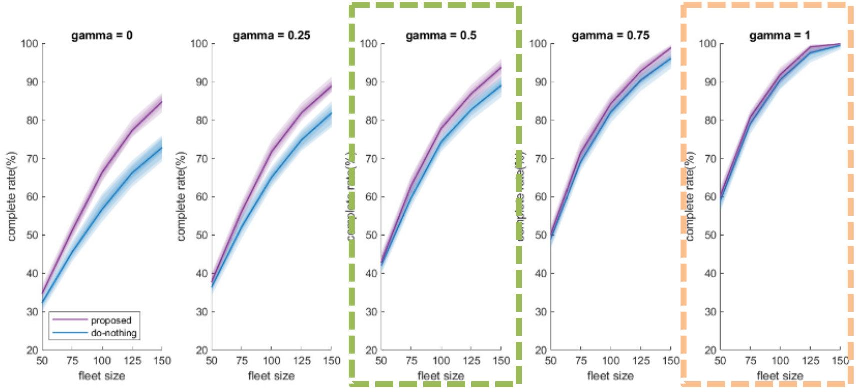
$$p'_d(q) = \boldsymbol{\gamma} \cdot p_d(q) + (\boldsymbol{1} - \boldsymbol{\gamma}) \cdot p'_o(q)$$

where p'_o is an artificial distribution which has the maximum difference from the origin distribution

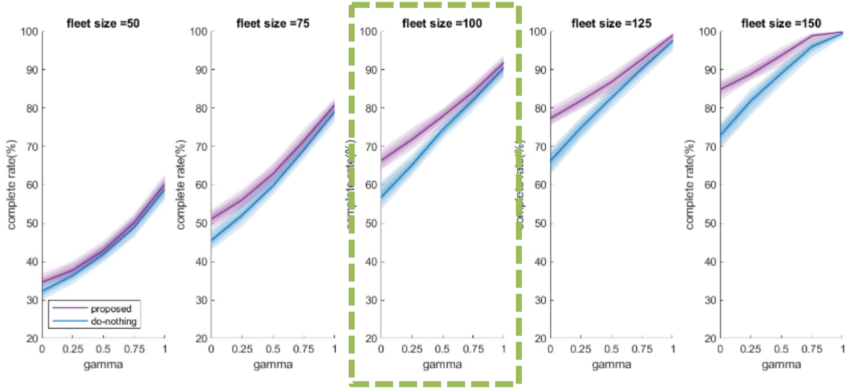
- When $\gamma = 1$, the generated destination distribution is equal to the original destination distribution.
- The smaller the γ is, the more discrepancy is introduced between the origin and generated destination distributions.
- When $\gamma = 0$, the generated destination distribution has a shape that is maximally different than the origin one.

Comparison of performance metrics for **various γ and fleet size value**

- Fleet size
- Gamma



(a) Results with varying values of origin destination demand imbalance parameter γ.

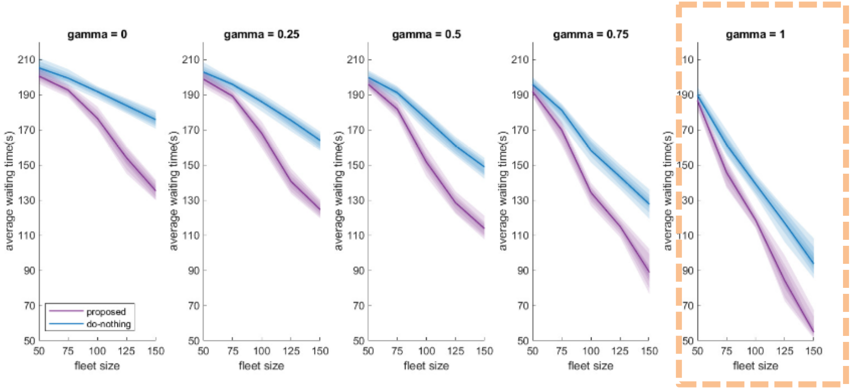


(b) Results with varying values of fleet size.

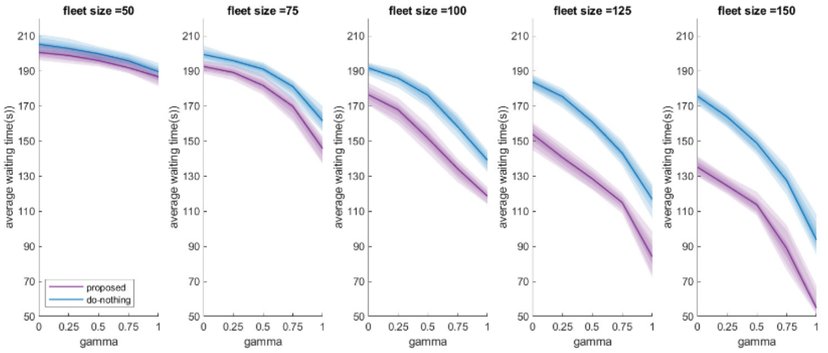
Completion rate

Comparison of performance metrics for **various γ and fleet size value**

- Fleet size
- Gamma



(a) Results with varying values of origin destination demand imbalance parameter γ.



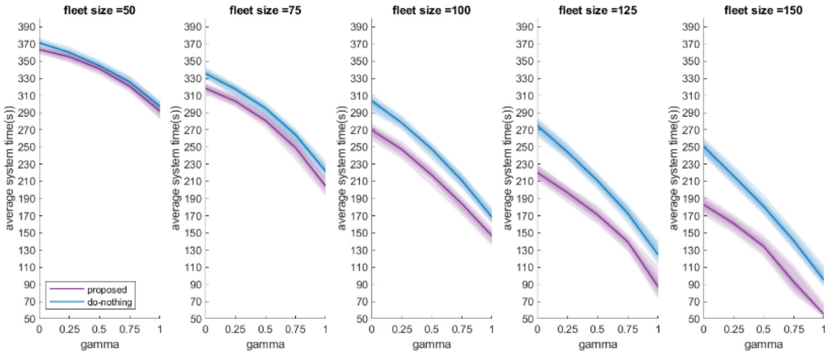
(b) Results with varying values of fleet size.

Average Waiting time

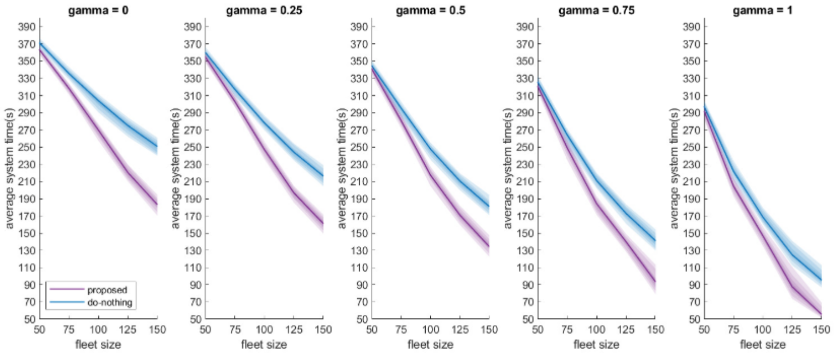
Comparison of performance metrics for **various γ and fleet size value**

- Fleet size

- Gamma



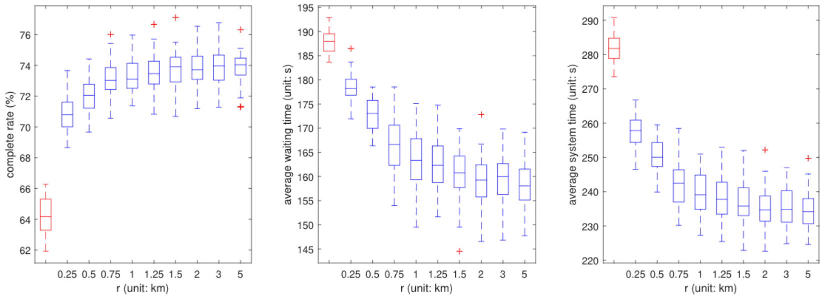
(a) Results with varying values of origin destination demand imbalance parameter γ.



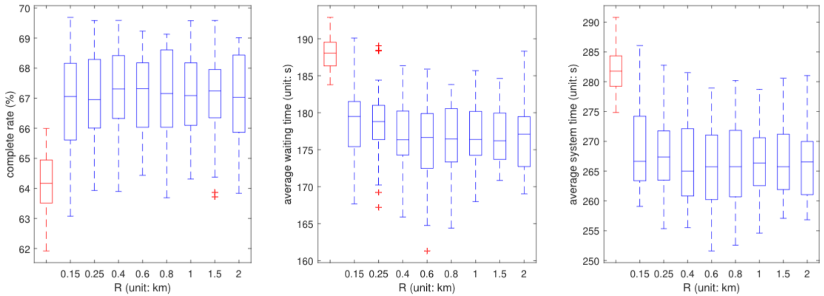
(b) Results with varying values of fleet size.

Average system time

Comparison of performance metrics for **various r and R**

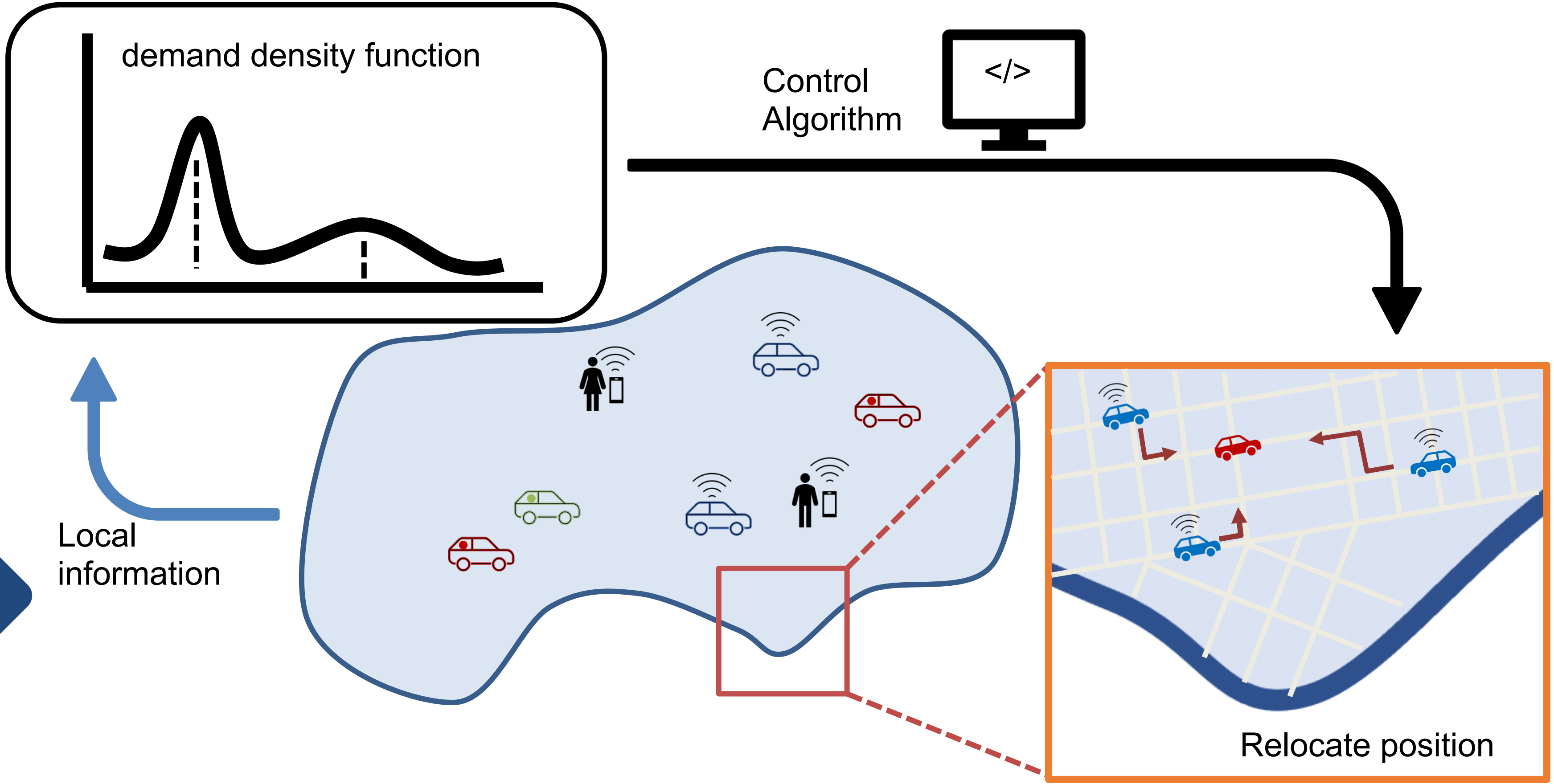


(a) Results with varying values of covering radius r.



(b) Results with varying values of the communication limitation radius R.

Conclusion


demand density function
Control Algorithm
</>
Local information
Relocate position

Conclusion

Distributed Coverage Control Algorithm:

- ➢ Application to rebalancing of vehicle fleets for urban Mobility-on-Demand systems

- ✓ Countering spatiotemporal imbalances in the origins and destinations of trip demands
- ✓ Dynamically rebalance spatial distribution, serve **more trips** with **less waiting time.**
- ✓ Tested on both continuous and discrete space compared with a do-nothing policy.
- ✓ The effects of coverage and communication radius are demonstrated respectively.

Future steps

1. Estimate time-varying demands from real-time data
2. Ride-sharing/pooling
3. Detour problem
4. Price Policy

Thanks for your attention!

Avez-vous des questions ?

Pengbo ZHU
LUTS, EPFL
pengbo.zhu@epfl.ch

National Centre of Competence in Research Automation

EPFL

LUTS