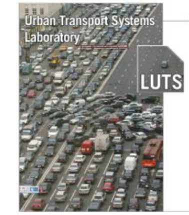
Optimization-based clustering of urban networks through snake segmentation

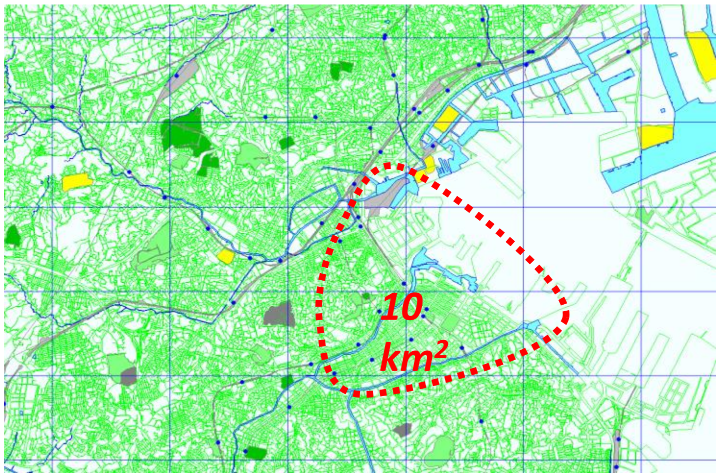
Mohammadreza Saeedmanesh
Nikolas Geroliminis

Urban Transport Systems Laboratory (LUTS)

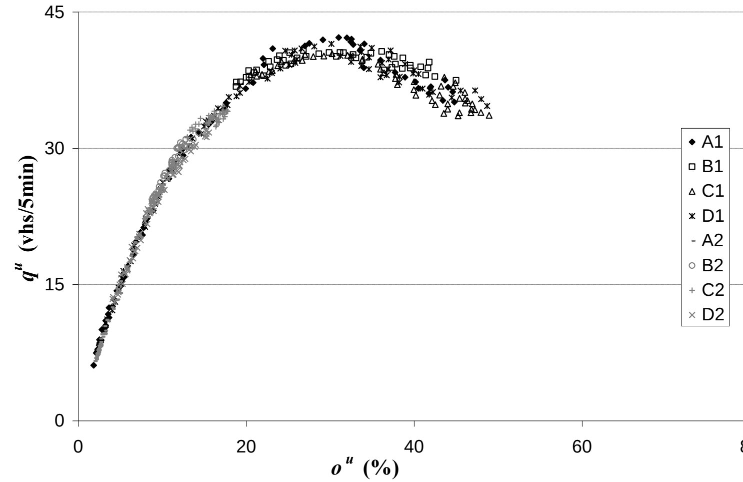


Macroscopic Fundamental Diagram (MFD)

Yokohama (Japan)

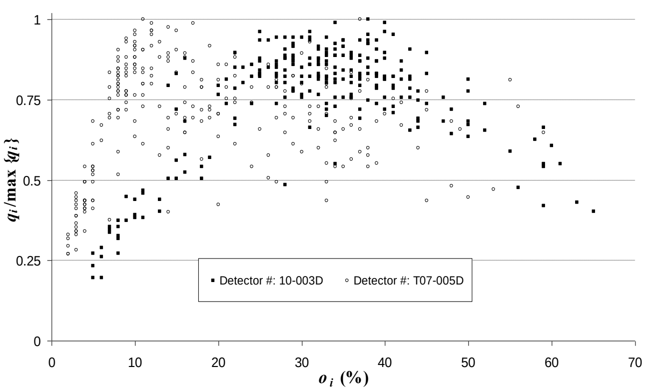


Average flow



Average occupancy

Individual flow



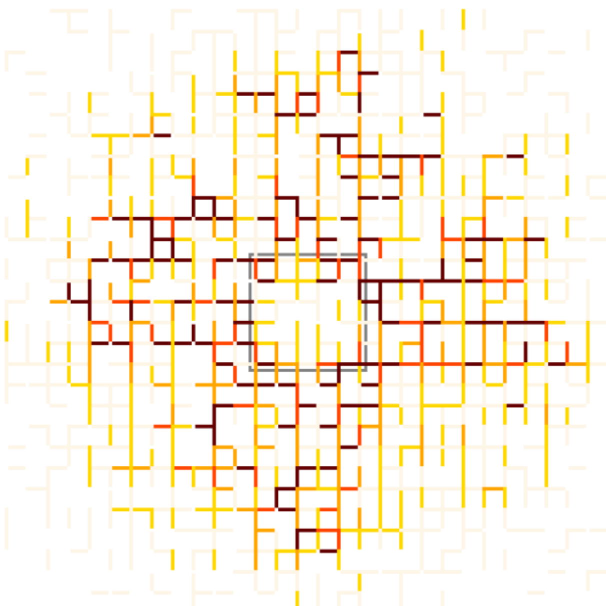
occupancy

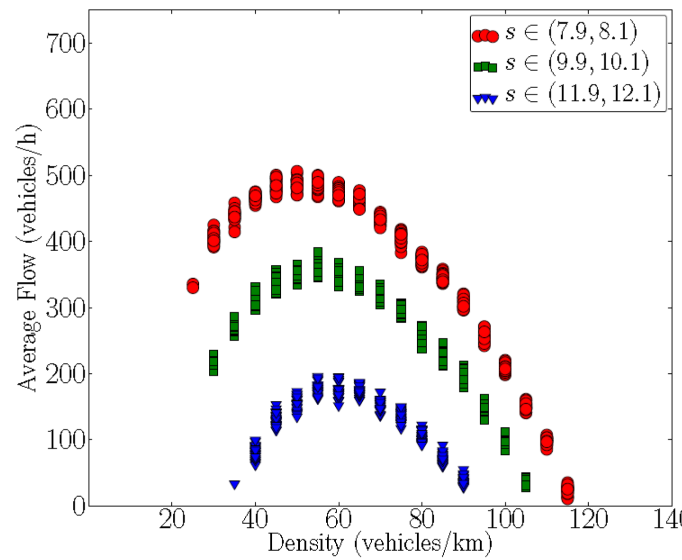
Geroliminis and Daganzo (2008) – Tr. Res. Part B

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Macroscopic Fundamental Diagram (MFD)

- MFD is not a universal law
- Spatial distribution of congestion is crucial
  - Average flow is a function of both average and variance of link densities
  - Well-defined MFD exists in homogeneous regions



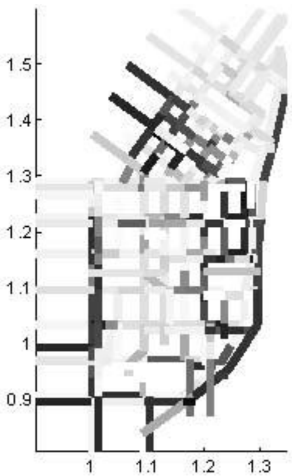


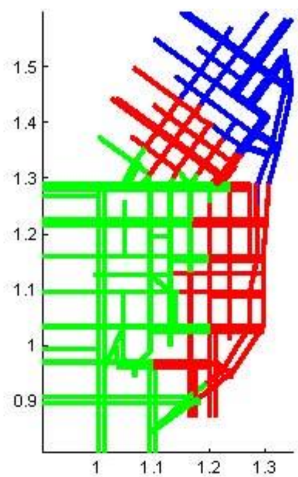
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

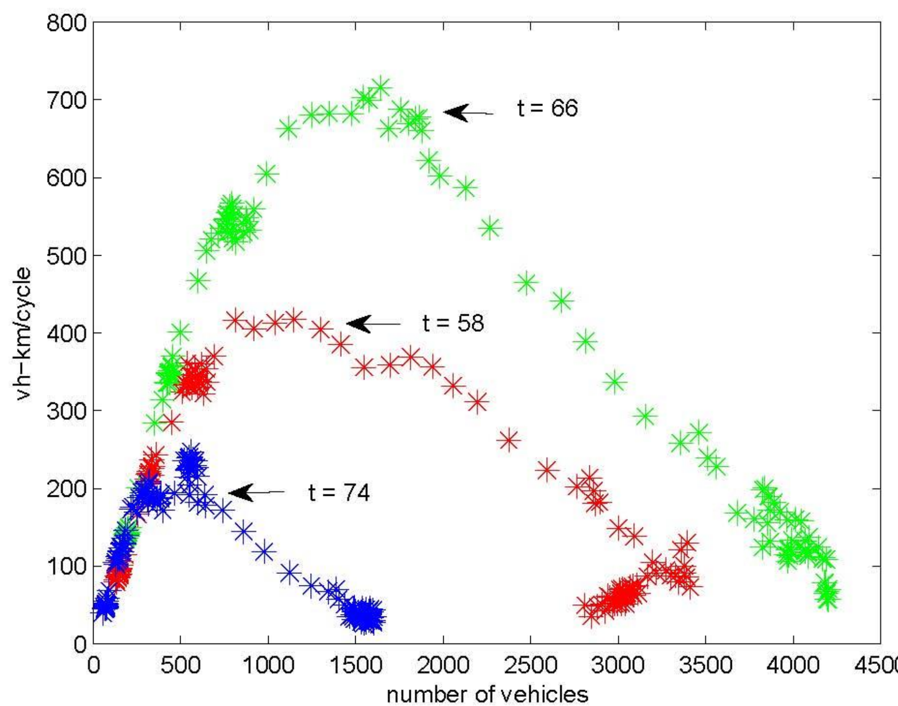
Mazloumian, Geroliminis and Helbing (2010) – Ph.Trans.Roy. Soc. A

Intro. to Clustering - Congestion Spreading

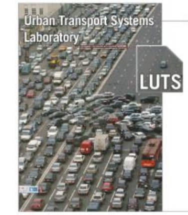
- Well-defined MFD
- Information about congestion time and location







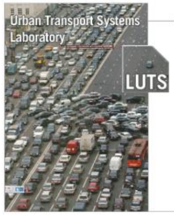
Ji and Geroliminis (2012) – Tr. Res. Part B:



Existing clustering approaches

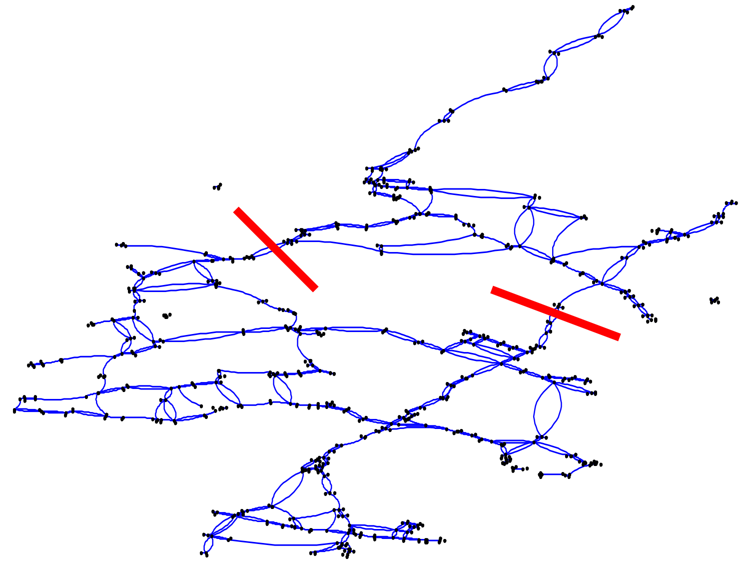
Goal: Find homogeneous compact clusters ⟶ Apply perimeter control

- Normal Clustering approaches
  - K-means
  - Fuzzy clustering
  - Artificial Neural networks
  - Spectral clustering
  - …
- Contiguity constrained clustering approaches
  - Penalized k-means
  - Hierarchical Agglomerative
  - NMB (Normalized cut + Merging + Boundary Adjustment)
  - …

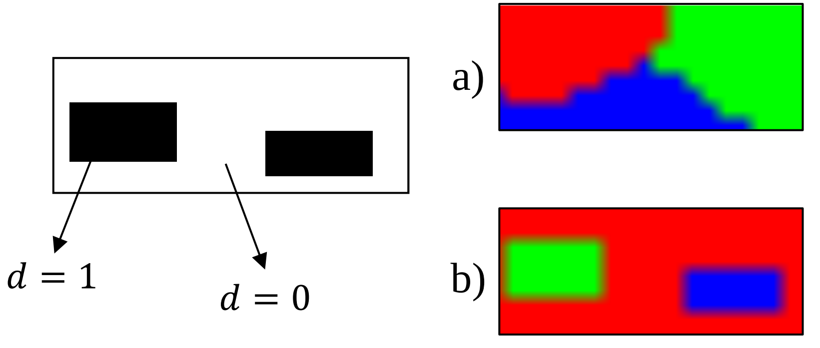


Issues of Existing Methods

- Sensitive to network structure
- Network is partially observable (Missing data)
- Sensitive to parameter calibration
- Incapable of finding directional congestion



Sydney (Australia)



Methodology

- Develop static clustering algorithm to find connected homogeneous regions in heterogeneous urban network
  - Detection of directional
  - Not sensitive to network structure (not perfect grid, well-connected)
  - Deal with missing data (partially observable networks)
  - Not sensitive to parameter calibration
- Three-step algorithm (Snake algorithm):
  - Running snakes
  - Defining similarities
  - Symmetric Nonnegative Matrix Factorization (SNMF)

Running Snakes

- Algorithm
  1) Start from point ‘i’ (link ‘i’) → ($S_i = [i]$)
  2) While snake covers all the network

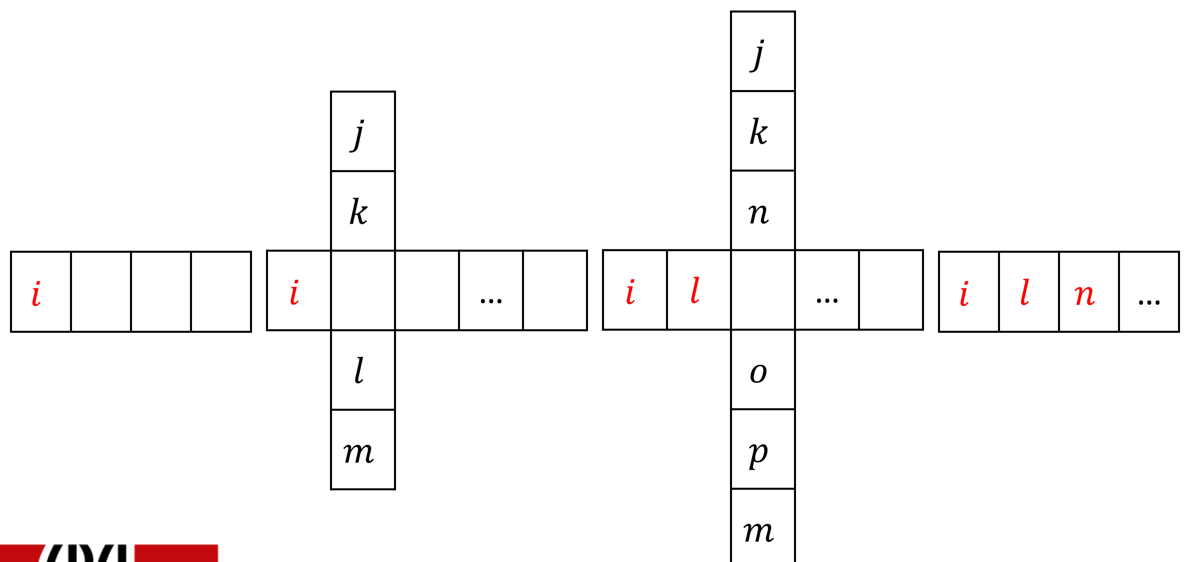
     {

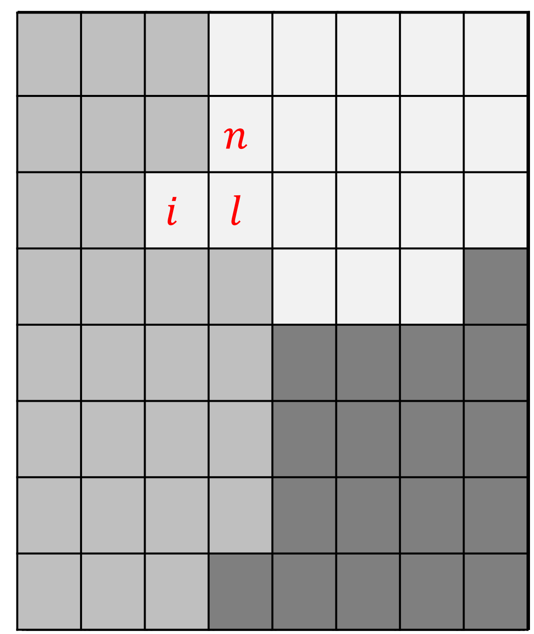
     i. Find all adjacent points → (S')

     ii. Compute the distance to the average

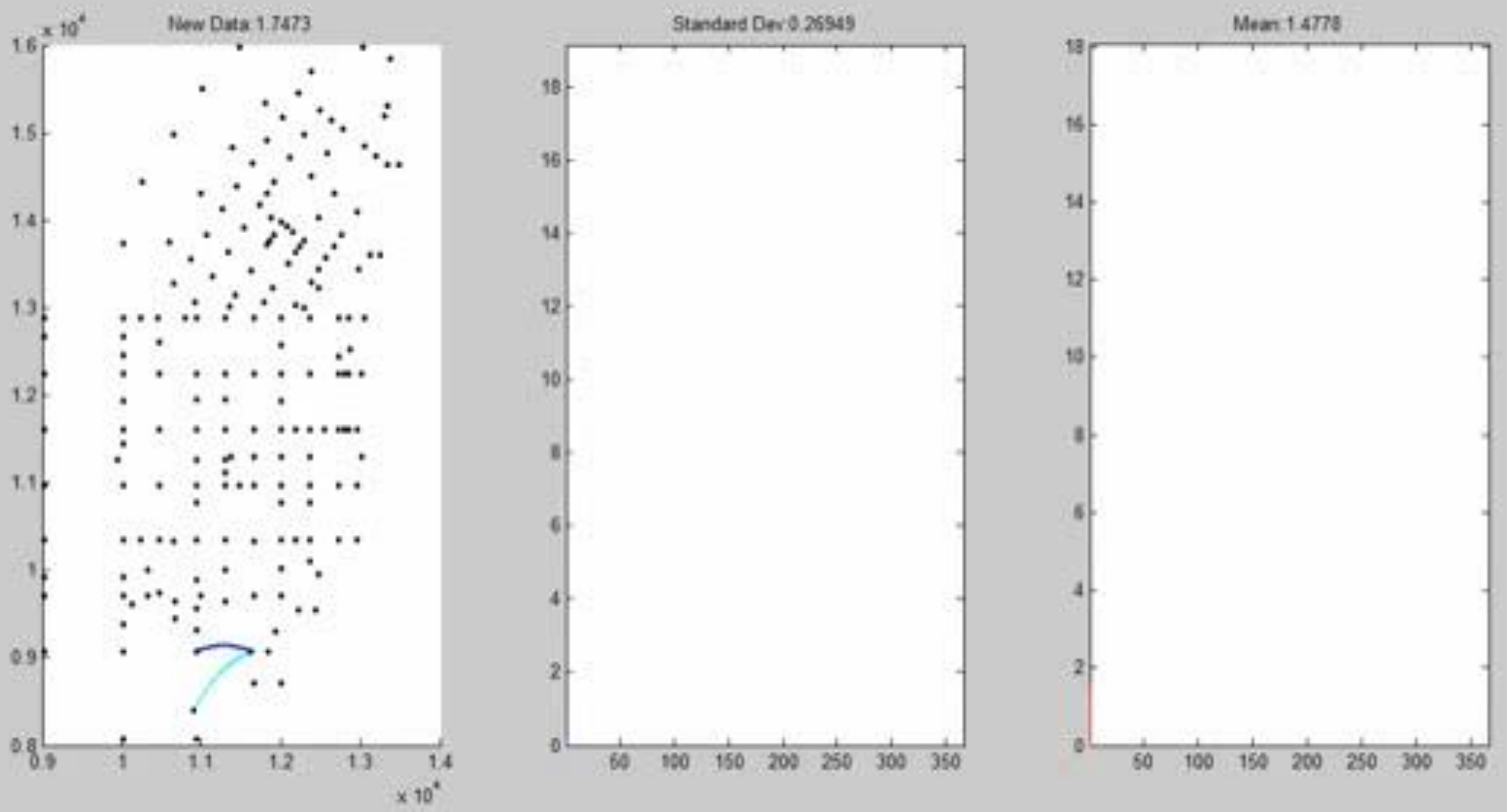
     iii. Pick the one with the closest value to the average and add it

     }





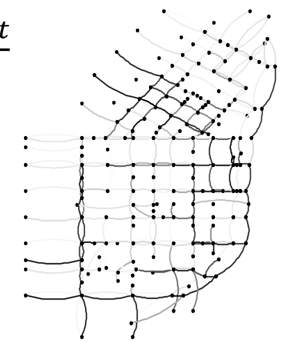
Running Snakes (step 1)

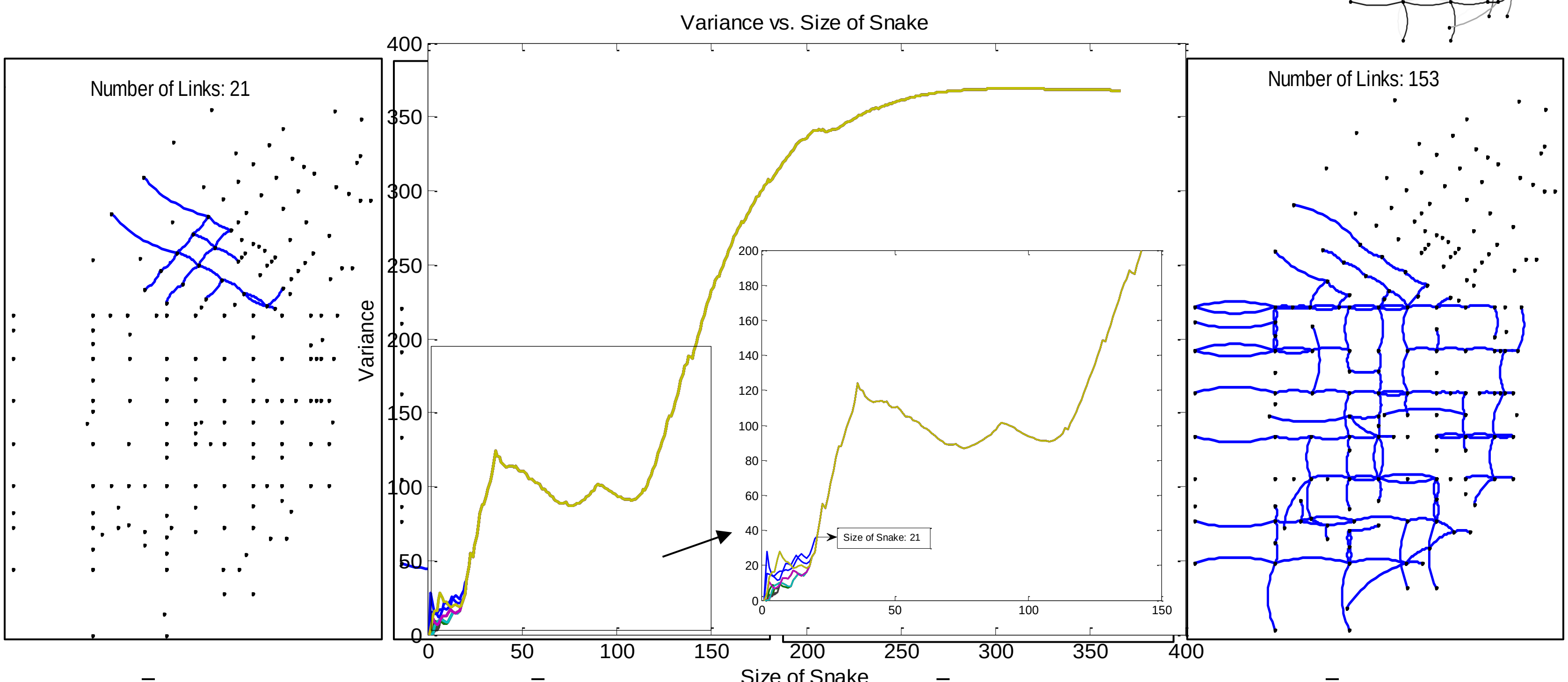


Running snakes

Robustness of the component: $\frac{Number\ of\ the\ snakes\ converge\ to\ the\ same\ component}{number\ of\ the\ links\ in\ the\ component}$

– Full robust components: Robustness = 1





$\bar{d}$: 48.51 $\bar{d}$: 51.63 $\bar{d}$: 5.36 $\bar{d}$: 4.86

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Defining similarities (step 2)

- Algorithm
  1) Set window size equal to one ($k = 1$)
  2) While (size of window $\leq$ size of network)

     {

     i. Calculate common points in first k elements of S_i and S_j arrays $\rightarrow$ (m)

     ii. $w_{ij} = w_{ij} + p^k \times m$

     iii. Increase the size of window $\rightarrow$ ($k = k + 1$)

     }

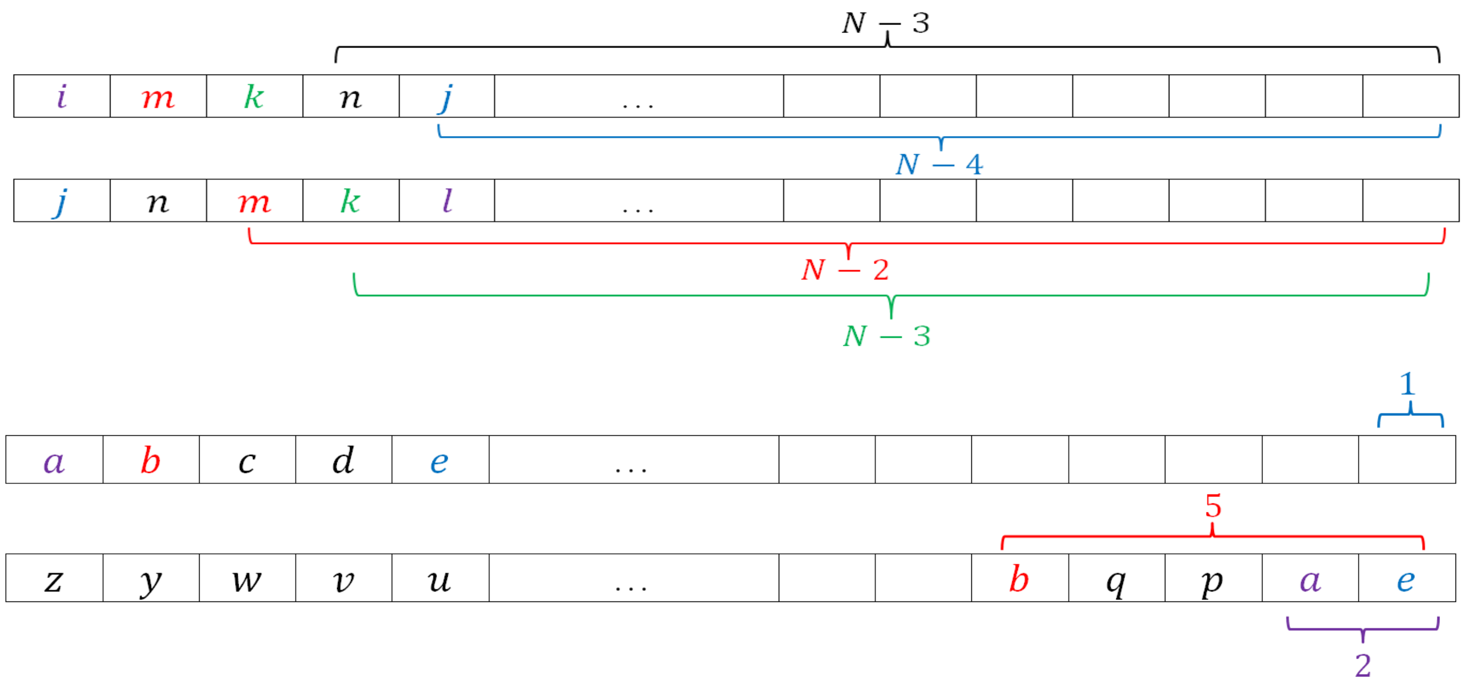
$k = 1$ $k = 2$ $k = 3$ $k = 4$

S_i	$\rightarrow$	i	b	c	d	…
S_j	$\rightarrow$	j	i	n	c	…

$m = 0$
$w_{ij} = 0 + m = 0$

Defining similarities

$$w(i,j) = \sum_{t \in links} \left\{ N - \max\left(L_{s_i}(t), L_{s_j}(t) \right) + 1 \right\}$$

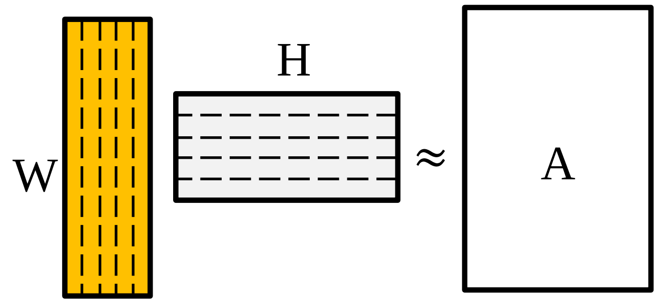


✓ Put more weight on adjacent links → (Connectivity, compact shape)

✓ Keep variance as lowest as possible → (Homogeneity)

Nonnegative Matrix Factorization (Step 3)

$$\left(\min_{W\geq 0, H\geq 0} \|A - WH\|_F^2\right)$$

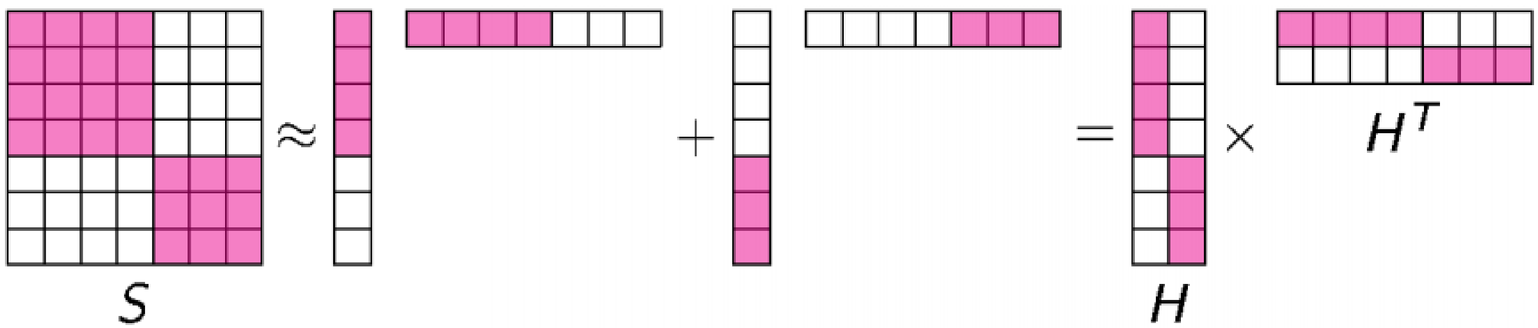


Given $A \in \mathcal{R}_+^{(m\times n)}$ and desired rank $k \ll \min(m,n)$, Find $W \in \mathcal{R}_+^{(n\times k)}$ and $H \in \mathcal{R}_+^{(k\times m)}$ such that $A \approx WH$

- Non-convex
- Lower rank approximation

Symmetric NMF

$$\min_{H \geq 0} \|S - HH^T\|^2$$

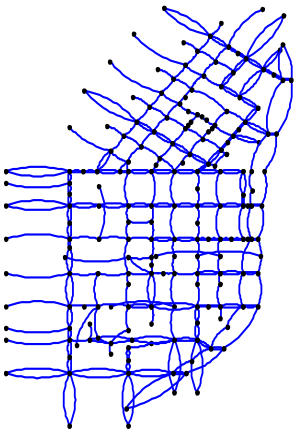


S: Similarity matrix

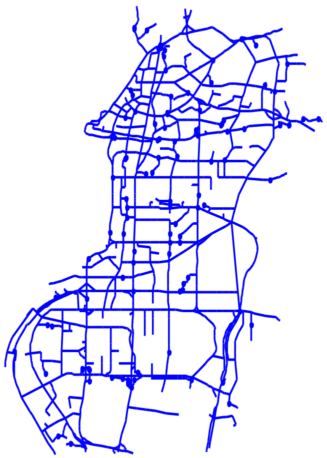
H: Clustering membership

Case Study

- San Francisco
  - About 100 intersections and 366 links
  - Average density for each 2 min time interval



- Shenzhen
  - About 600 intersections and 2000 links
  - Daily GPS data of taxis
  - Average speed of vehicles for each 5 min interval

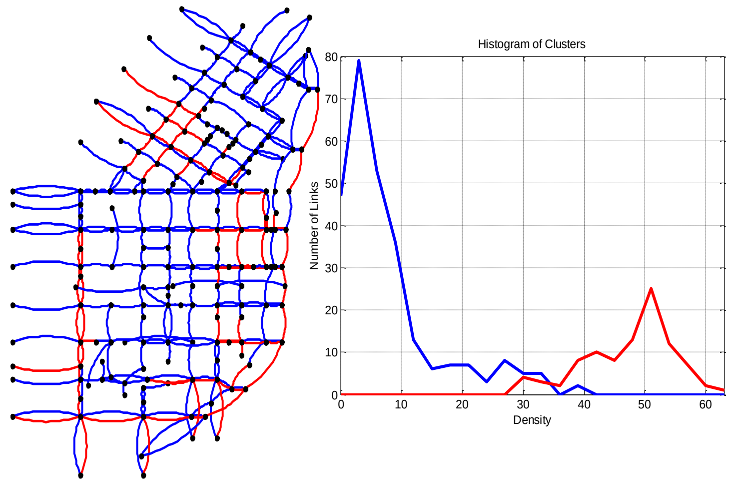


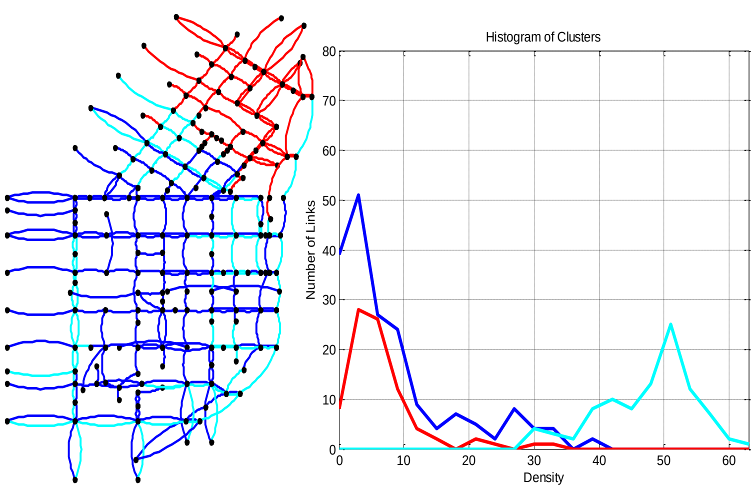
Case Study (San-Francisco)

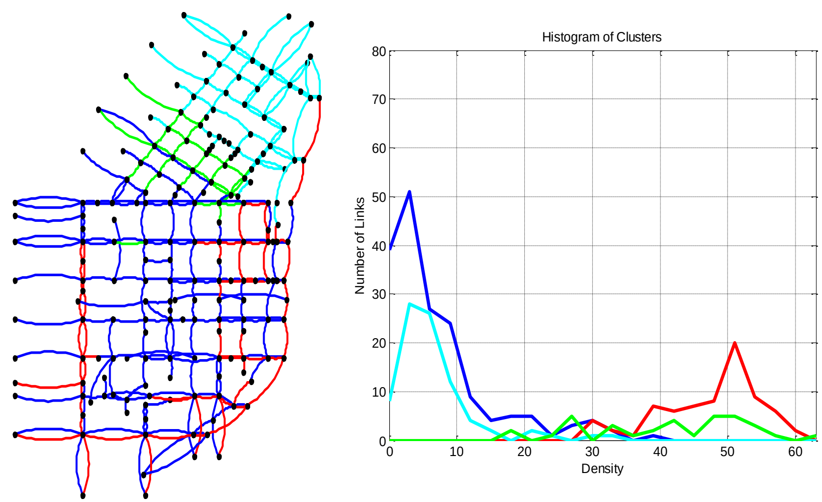
$$TV = \sum_{A \in C} N_A \times var(A)$$

Table 1: Average values and standard deviation of link densities for different clustering results [veh/(km.lane)]

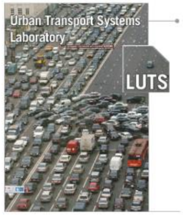
(Mean/Standard deviation)	Blue	Red	Cyan	Green	Yellow	TV
2	7.73/8.28	47.43/7.22	-	-	-	2.36
3	8.30/9.16	6.48/5.71	47.43/7.22	-	-	2.34
4	7.11/7.94	47.32/7.38	6.48/5.71	41.15/11.5	-	2.22
5	9.57/9.64	6.48/5.71	47.80/6.68	47.32/7.38	3.36/4.33	2.23





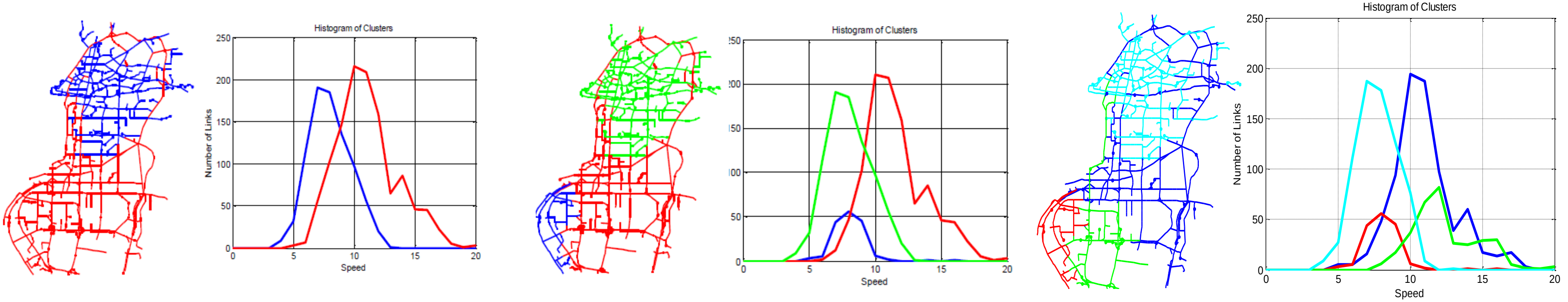


ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE



Case Study (Shenzhen)

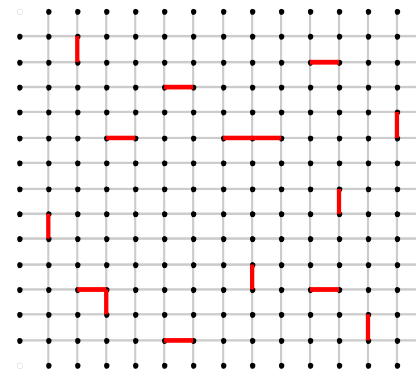
Table 2: Average values and standard deviations of link speeds for different clustering results (2 -5 clusters) [m/s]						
(Mean/Standard deviation)	Blue	Red	Green	Cyan	Black	TV ($\times 10^4$)
2	8.05/1.69	11.03/2.49	-	-	-	0.97
3	8.07/1.30	11.51/2.30	8.05/1.69	-	-	0.80
4	10.95/2.17	8.07/1.30	12.40/2.26	7.71/1.40	-	0.71
5	10.87/2.86	7.32/1.24	12.40/2.26	8.07/1.30	10.28/1.40	0.74



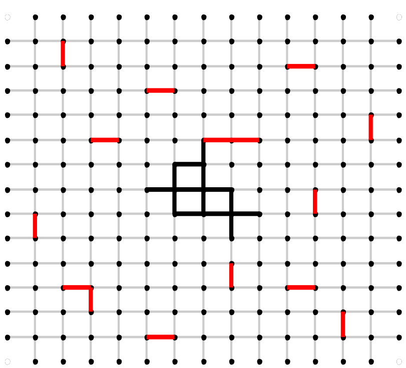
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

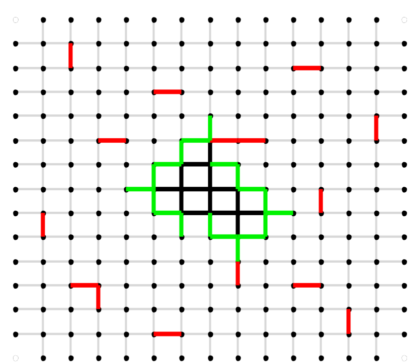
Algorithm Extension

- Investigate virtual connections through links with missing data
- Put a priority to the links that are closer
- Assign threshold to avoid bypassing heterogeneous parts

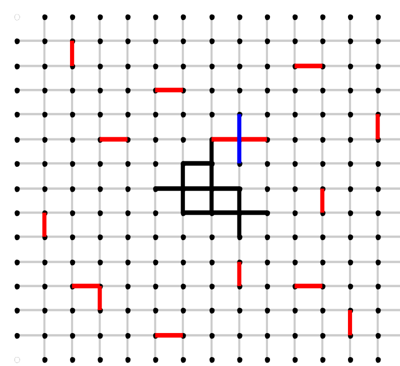


$$d_i = a^{r_i - 1} \times |D_i - \mu| \,, a \geq 1$$

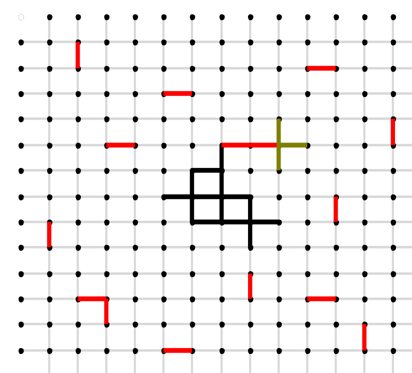




$d = |D_i - \mu|$



$d = a \times |D_i - \mu|$



$d = a^2 \times |D_i - \mu|$

Algorithm Extension

- Keep index of links with data in the arrays
- Define similarity matrix for links with data
- Finding symmetric lower rank approximation (Symmetric NMF)

Conclusion & Future work

- Conclusions
  - Congestion spreading
    - Congestion is spatially correlated in adjacent road
    - This correlation allows us define pocket of congestion
  - Partitioning algorithm
    - Independent of network structure
    - Capable of finding directional congestion
    - No need to parameter calibration
- Future work
  - Model the problem mathematically (size constraint – Contiguity constraints)
  - Dynamic clustering (Congestion evolution)

Thank you for your attention