Université Gustave Eiffel

Management in Intelligent Systems

Mme BHOURI Neila & M. FRANÇOIS Olivier

Master 2 SIA : Exam 2024-2025

Resilience of Public Transportation Networks

This project aims to investigate the resilience of public transportation networks using the principles of percolation theory. Specifically, the study computes the percolation characteristics of bus networks under static conditions (i.e. no variable depends on time).

Attached material

- Research paper :
  - *Hamedmoghadam et al., **Percolation of heterogeneous flows uncovers the bottlenecks of infrastructure networks***, Nature Communications 12(1):1254, February 2021. DOI : 10.1038/s41467-021-21483-y.
- Code used in the tutorials
  - `ComputeAdj.m` (and ComputeSizeAdj, SearchIndex subfunctions);
  - `Dijkstra2.m` (and RefreshWeight, SearchLightNode subfunctions);
  - `BetweennessCentrality.m`.
- Code to describe the new application networks
  - `Mandl18routes.m`, that contains the definition of `TravelLinkTime`, `Routes` and an Origin/Destination matrix (`OD`) needed to build a network consisting in 15 stops crossed by 18 bus routes. Each route is defined by the succession of bus stops and is circulated in both directions (see Figure 1).

PART 1 : Understanding the percolation Theory

Question 1 : Definition

Define the percolation phenomenon based on the attached paper, lecture content, and independent research.

Question 2 : Relevance

Explain the relevance of percolation theory to the study of network resilience.

Part 2 : Computing multiple quality matrices

Question 1 : Adjacency matrix

Modify `ComputeAdj.m` so that it could use the following StopList as an input to build the adjacenty matrix of the network (then transform it as an **unoriented** network).

```
>> Stops={'1','2','3','4','5','6','7','8','9','10','11','12','13','14','15'};
>> [DAdj,DwAdj] = ComputeAdj2(ROUTES,TravelLinkTime,Stops);
```

Question 2 : Quality matrix based on Travel Time

Build a quality matrix where each edge has $q_{ij} = \dfrac{1}{TravelTime_{i,j}}$.

(Note that the $TravelTime_{i,j} = 0$ if there is no link from i to j, that lead to an Infinite quality).

This case illustrates that the longer the journey, the more likely a problem could occur along the way.

Question 3 : Quality matrix based on inverse Travel Time

Build a quality matrix where each edge has the following quality value.

$$q_{ij} = \frac{1}{\max\limits_{i,j}(TravelTime_{i,j}) * \dfrac{1}{TravelTime_{i,j}}}.$$

This quality definition will run the algorithm using the opposite order than the previous one. Take care of getting *Infinite values* at the right place in the Quality matrix *(where there are no edge, may be easier using this given writing)*!

Question 4 : Quality matrix based on Betweenness centrality

Build a quality matrix where each edge has the following quality value.

$$q_{ij} = \frac{1}{1 + EdgeBetweennessCentrality_{i,j}}$$

using the following definition for the *Betweenness Centrality of edges* :

```
>> BC = BetweennessCentrality(Adj);
>> EdgesBC = ( BC*ones(1,size(Adj,2))+ones(size(Adj,1),1)*BC' ) .* Adj
```

Explain this definition and what could represent this study case.

Université Gustave Eiffel • Campus de Marne-la-Vallée
5, Boulevard Descartes • Champs-sur-Marne • 77454 Marne-la-Vallée CEDEX 2
univ-gustave-eiffel.fr

Part 3 : Main algorithm

The objective of this algorithm is to study the percolation behavior of the network by iteratively removing links as in the attached paper. It will then be applied to the different scenarios listed above.

Question 1 : Subfunction updating reachable passengers demand

Propose a function that updates the feasible Origin/Destination demand matrix given an updated adjacency matrix *(where edges has been removed)*.

```
>> [OD_updated disconnected] = updateOD(OD,Adj_updated)
```

One might use the `[Ws,P]=Dijkstra2(Adj,Origin)` seen in course.
After the loop, check if every path is reachable to output the `'disconnected'` logical value for graphs with more than one connected subgraph.

Question 2 : percolation function

To avoid considering edges twice in the following part, replace all values in the lower triangular part of the matrix Q by infinity values *(as one edge appear to be an directed arc)*.

```
>> Qtriu=Q.*(tril(Inf(size(Q)))+1);
```

Build a function that ouput the unaffected demand given the percolation threshold, reporting the `'Critical_percolation_threshold'` from where all the stops are not connected from each others anymore *(disconnected graph)*.

```
>> [Proba_OD_decrease, Proba_ARC_removal, Critical_percolation_threshold]=...
     Percolation(Qtriu, Adj, OD)
```

One might consider a while loop over the non infinity values remaining in `Qtriu` : removing the edge with the lower quality; then updating the matrices until there are no more edges.

Question 3 : Plotting the results on different Quality matrices

Using plot function to print `Proba_ARC_removal` given `Proba_OD_decrease`. Then add a mark for the `Critical_percolation_threshold` in the figure. Perform the work for the three different Quality matrices built in the Part 2 :

- Scenario (a) : Ascending Order of Quality from question 2.
- Scenario (b) : Ascending Order of Quality from question 3 (reverse than a).
- Scenario (c) : Ascending Order based on the `Edge_Betweeness centralities` values of each edge.

Université Gustave Eiffel • Campus de Marne-la-Vallée
5, Boulevard Descartes • Champs-sur-Marne • 77454 Marne-la-Vallée CEDEX 2
univ-gustave-eiffel.fr

Question 4 : Evaluating the Alphas

Build a function to estimate the area under the curve using rectangular (over and under) approximations.

```
>> Alpha = manual_integration(Proba_OD_decrease, Proba_ARC_removal)
```

(Remark that this `Alpha` might be evaluted during the algorithm in Question 2. If you've done so, just go through this question.)

Part 4 : Conclusions

Question 1 : Node failures

Considering the nodes are now blocked/attaked in the order of their beetweenness Centrality values *(from lower to higher values, then in reverse, from the highest to the lowest)*, edit a new percolation function that removes all edges linked to a node at each step following a given order.

```
>> [tmp Node_order]=sort(BetweennessCentrality(Adj));

>> [Proba_OD_decrease, Proba_NODE_removal, Critical_node]=...
     Percolation(Qtriu, Adj, OD, Node_order)
```

Question 2 : New plots

Draw a new percolation curves from node removals and their impact on feasible passengers demand.

Question 3 : About resilience

Conclude about the resilience of the propose study case.

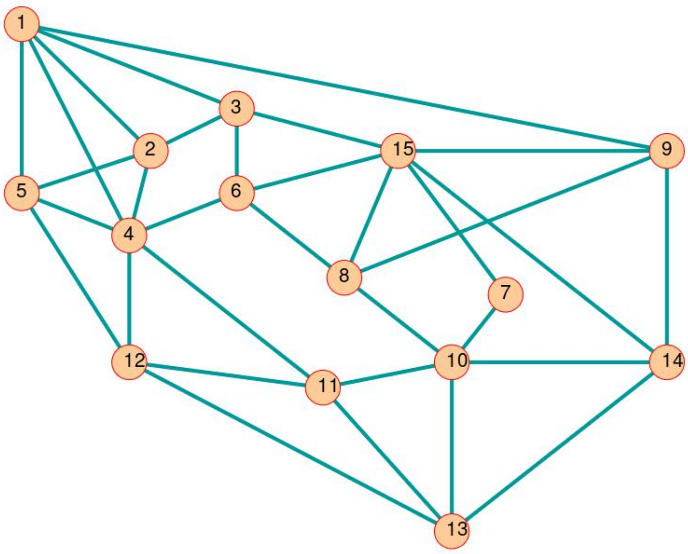

FIGURE 1 – Capture of the case study network.

Université Gustave Eiffel • Campus de Marne-la-Vallée
5, Boulevard Descartes • Champs-sur-Marne • 77454 Marne-la-Vallée CEDEX 2
univ-gustave-eiffel.fr