Arise of congestion states on Bologna urban network

Complex Systems

Mario Massimo
1102698

Academic year 2024/2025

Contents

Contents **2**

1 Graph theory **3**

1.1 Adjacency matrix 3

1.2 Degree 4

1.3 Laplacian 4

2 Dynamic in a transport network **4**

2.1 Stationary state 6

3 Methods **7**

3.1 Synchronous and asynchronous dynamics 8

4 Results **8**

5 Conclusions **14**

Abstract

This report investigates the emergence of congestion states in the Bologna urban network through a graph-based approach. The city's road system is modeled as an undirected, unweighted graph, where nodes are intersections and edges are streets. A Markovian random walk framework is used to simulate traffic flow, incorporating two constraints, a maximum flux rate and a maximum node capacity. Two types of simulation dynamics, synchronous and asynchronous, are compared to study their effects on traffic behavior. The results show that traffic congestion begins to emerge when the average load per node reaches half the maximum capacity, with transport efficiency peaking around this value. As traffic load increases further, flow decreases due to the saturation of nodes and the states with higher traffic loads become more and more probably. A relationship between local and global network quantities is suggested from the analysis of deviations of node loads. While this model provides insights into macroscopic traffic dynamics, it simplifies real-world complexity by excluding traffic lights and vehicle interactions. The simulation script is available at this Github page.

1 Graph theory

Since the network approach has been widely used in urban studies, here I briefly describe the mathematical model of graphs.

A graph G is is defined as a pair (V, E), where $V = \{v_1, v_2, \cdots, v_M\}$ is the set of vertices, or *nodes*, and $E \subseteq V \times V$ is the set of edges, or *links*, connecting nodes. A graph is called *undirected* if edges have no direction, i.e. if there is an edge between nodes A and B you can travel both ways: $A \to B$ and $A \leftarrow B$. Otherwise it's called directed graph, or Digraph.

If at each edge is assigned a numerical value, the graph is said *weighted.*

A *multigraph* is a graph having more than one edge between at least one couple of nodes. A graph without multiple edges and self-loops (links connecting a node to itself) is called *simple.*

A typical representation of an urban network is the so called primal approach [1], a graph where nodes identify intersections and the edges are roads. In the present report, this primal graph structure serves as the basis for simulating a random walk of particles, where particles represent cars navigating the network.

1.1 Adjacency matrix

Given a graph with M nodes, the adjacency matrix $\mathbf{A}_{ij}$ is a $n \times n$ matrix defined as:

$$\mathbf{A}_{ij} = \begin{cases} w_{ij} & \text{if } \exists \text{ a link between nodes } i \text{ e } j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For an unweighted graph $w_{ij} = 1 \ \forall i, j$.

In an undirected graph the adjacency matrix is symmetric, since if exist an edge from node v_i to v_j is true also viceversa, and if the graph is also simple, the diagonal elements $\mathbf{A}_{ii}$ are null due to the absence of self-loops.

1.2 Degree

The degree of a node is defined ad the number of edges that are connected to it. The degree k_i of the node v_i can be computed, for an undirected graph, using the adjacency matrix:

$$k_i = \sum_{j=1}^{M} A_{ij} \tag{2}$$

Where M is the number of nodes.

The degree histogram of an urban network can be useful to see how connectivity is spread among nodes and to understand the road structure.

1.3 Laplacian

The Laplacian matrix is a fundamental matrix associated with the adjacency matrix, encoding structural and dynamic properties of a graph.

Given an adjacency matrix $\mathbf{A}$ and a degree matrix $\mathbf{D}$, a diagonal matrix where the elements $\mathbf{D}_{ii}$ is the degree of node v_i, the Laplacian matrix is defined as:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{3}$$

And its elements are:

$$\mathbf{L}_{ij} = \begin{cases} k_i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and a } \exists \text{ a link between } v_i \text{ and } v_j \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Or, using the Kronecker delta matrix δ_{ij}:

$$\mathbf{L}_{ij} = \delta_{ij} k_i - \mathbf{A}_{ij} \tag{5}$$

Where $\delta_{ij} = 1$ only if $i = j$.

It's called Laplacian since it replaces the Laplacian operator ∇^2 in the analogous diffusion equation for the particles moving within a graph [2]:

$$\frac{d\psi}{dt} + C\mathbf{L}\psi = 0 \tag{6}$$

Where ψ is the vector whose components ψ_i are the number of particles in the node v_i moving from node v_j to v_i, and C is a diffusion constant.

2 Dynamic in a transport network

To model traffic flow dynamics it's possible to simulate a Markovian random walk on a graph when the only knowledge available is the transition probability matrix π_{ij} at road intersections, measuring the probabilities of observing a path moving between the roads $j \to i$. This process maximizes the information entropy of the particle trajectories distribution given π_{ij}.

The transition probabilities do not require the whole knowledge of the path distribution and can be measured locally at the intersections. Using these transition probabilities and continuity arguments, one can estimate average traffic flows and identify critical congestion points.

Traffic congestion on urban roads network arise due to two local factors: finite traffic flow rates at intersections, which can cause queues when incoming flow exceeds capacity, and the existence of maximum road capacities, which can lead to gridlocks.

In a transport network each node (intersection) is characterized by an internal state n_i, which quantify the traffic load, and the flow ϕ_{ij} from node j to node i depends on the node states (Markov field model [3]):

$$\phi_{ij} = \phi_{ij}(n_i, n_j) \tag{7}$$

By definition $\phi_{ij}(n_i, n_j) \geq 0$ and $\phi_{ij}(n_i, 0) = 0 \; \forall \; n_i$.

A maximum flow rate is commonly set as a higher limit for the outgoing flow from node j:

$$\sum_i \phi_{ij} = \phi_j \leq \phi^{max} \tag{8}$$

A maximum node traffic load n^{max} is defined such as the maximum number of cars a node can host without canceling the incoming flux:

$$\exists \quad n^{max}: \quad \phi_{ij}(n_i, n_j) = 0 \quad if \quad n_i > n^{max} \tag{9}$$

The total number of cars flowing in the network is given by:

$$N = \sum_i^M n_i \tag{10}$$

And the average traffic load, which will be useful later, is computed as:

$$\bar{n} = \frac{N}{M} \tag{11}$$

Where M is the total number of nodes.

The average dynamics of the transport network follow the equation:

$$\dot{n} = \sum_j [\phi_{ij}(n_i, n_j) - \phi_{ji}(n_j, n_i)] + s_i(t) \tag{12}$$

Where $s_i(t)$ describes the particle sources or the sinks in the system. To study the stationary solution s_i is set to $0 \; \forall \; i$ so the total traffic load N is constant.

The equilibrium equation reads:

$$\sum_j [\phi_{ij}(n_i^*, n_j^*) - \phi_{ji}(n_j^*, n_i^*)] = 0 \tag{13}$$

The congestion transition occurs when a solution exists with $n_i = n^{max}$ for a subset of nodes that became congested.

2.1 Stationary state

The flow $\phi_{ij}(n_i, n_j)$ can be defined as:

$$\phi_{ij}(n_i, n_j) = \pi_{ij}\phi_j^{max} c(n_i/n_i^{max})\phi(n_j) \tag{14}$$

Where the flow $\phi(n_j) \in [0, 1]$ is a monotonic increasing function with the asymptotic limit $lim_{n\to\infty}\phi(n) = 1$, and $c(n_i/n_i^{max}) \in [0, 1]$ is a capacity function that drop to 0 when $n_i \geq n_i^{max}$.

Using the relation 14 the equilibrium solution 13 becomes:

$$\sum_j \pi_{ij}\phi_j^{max} c(n_i/n_i^{max})\phi(n_j) = \sum_j \pi_{ji}\phi_i^{max} c(n_j/n_j^{max})\phi(n_i) \tag{15}$$

In case of low traffic load $c(n_i/n_i^{max}) = 1$, so the eq. 15 becomes:

$$\sum_j [\pi_{ij} - \delta_{ij}d_i]\phi(n_j) = 0 \quad with\ \ d_i = \sum_j \pi_{ji} \tag{16}$$

In which d_i represent the degree of node i. In eq. 16 ϕ is the eigenvector with null eigenvalue of the Laplacian matrix:

$$L_{ij} = \delta_{ij}d_i - \pi_{ij} \tag{17}$$

ϕ_j is the outgoing flux from node j at the equilibrium and it's possible to compute the values n_j^s at each node, solving the equation:

$$\phi(n_j^s) = \Phi^{tot}p_j \tag{18}$$

Where Φ^{tot} is the total flux in the network, and p_j is probability to observe a particle in the node j, which satisfy:

$$\sum_i^M p_i = 1 \tag{19}$$

A family of solutions $n_j^s(\Phi^{tot})$ can be found imposing:

$$\sum_j n_j^s(\Phi^{tot}) = N \tag{20}$$

In a homogeneous transport network all the roads are equivalent ($p_j = const$), ϕ^{max} is the same for each road and π_{ij} satisfies:

$$\sum_i \pi_{ij} = \sum_j \pi_{ij} \iff \sum_j L_{ij} = 0 \tag{21}$$

This condition is satisfied if exists a matrix $\hat{\pi}_{ij}$ such that the sum over its columns gives 1 $\sum_j \hat{\pi}_{ij} = 1$, and its associated eigenvector p_j can define π_{ij}:

$$\pi_{ij} = \hat{\pi}_{ij}p_j \tag{22}$$

In such way the balance condition is satisfied:

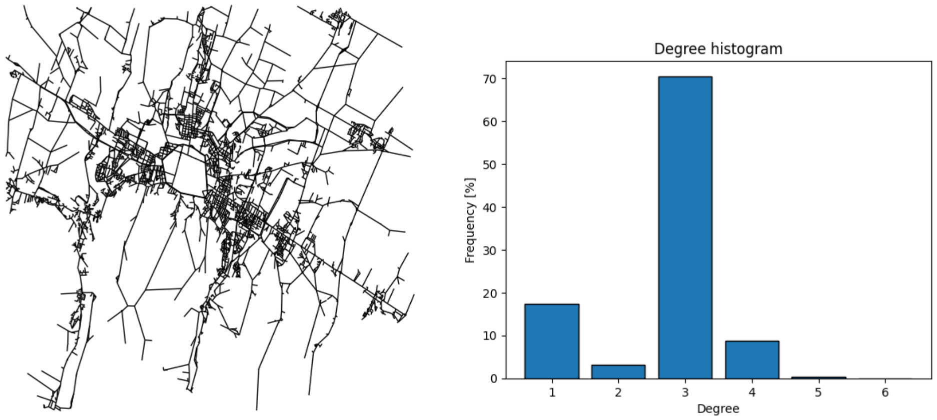
$$\sum_j \hat{\pi}_{ij}p_j = p_i = \sum_j \hat{\pi}_{ji}p_i \tag{23}$$

This condition guarantees the existence of an equilibrium solution of the eq. 12.

3 Methods

I simulated a random walk of cars over the Bologna urban network using a Python script available at this Github page. To build and manage the network I employed the Python library *Networkx*.

I employed Bologna graph data building an undirected and unweighted network having 7229 nodes, representing intersections, and 9825 links. The average degree of this network is $d \simeq 3$. Figure 1 shows the Bologna urban network and its degree histogram. The vast majority of nodes has a degree of 3, reflecting an high number of intersection with 3 neighboring streets. An high percentage of those nodes with a degree of 1 is due to the most external nodes of the network, connected most of the time by just one street.



(a) (b)

Figure 1: **a)** Visualization of the Bologna urban network, **b)** Degree histogram of the network

At the beginning of each simulation the average traffic load $\bar{n}$ is set, i.e. the number of cars per node, and N cars are randomly distributed through the nodes, where $N = \bar{n}M$, M being the number of nodes. For N times a random node is selected following an uniform distribution and its state, the number of cars stored, increases by one. N remains constant during the simulation, so no other sources of cars are present.

The maximum flow rate is set as $\phi^{max} = 1$ and the maximum node traffic load is $n^{max} = 10$. If at a given time step a node has a state $n_i \geq 10$ it cannot receive cars but only an outgoing flux is allowed. We expect this condition to allow congestion formation when the average traffic load increases. If $\bar{n}$ is an half-integer or a fraction, the integer part of $\bar{n}M$ is taken as N.

Regarding the random walk of cars, at each time step the traffic flux from node j to node i is described by an uniform transition probability. Considering a node with $k \in \mathbb{N}$ neighbors, the probability of each neighbors to be the destination of the cars flux is $\frac{1}{k}$. A random integer is generated from 1 to k and the corresponding node in the list of neighbors is chosen as arrival point. As already said, if the arrival node has an internal state $n_i \geq 10$, no cars are moved from node j, and the simulation continues focusing on another node.

If a transition is possible between two nodes, another parameter needs to be considered, the transport capacity. It's an higher limit of how many cars a node can move at each time step. I set it to 3, so that if a node has less than 3 cars it can possibly move all of them to the next node, but if it stores more than 3 cars it's allowed to move at maximum 3 of them. The number of cars moving from one node to another is again generated randomly using an uniform distribution and following the previous constraint. The number of cars moved from a node to another is stored in the edge linking them as an additional variable, and at the end of the simulation that number is divided by the number of time steps in order to compute the average flux of that road.

The output of the simulation are the initial graph with node and edge variables updated, and an array containing the population of each node per each time step, that will be used for further analysis (see section 4).

3.1 Synchronous and asynchronous dynamics

I repeated every measure two simulations dynamics: synchronous and asynchronous.

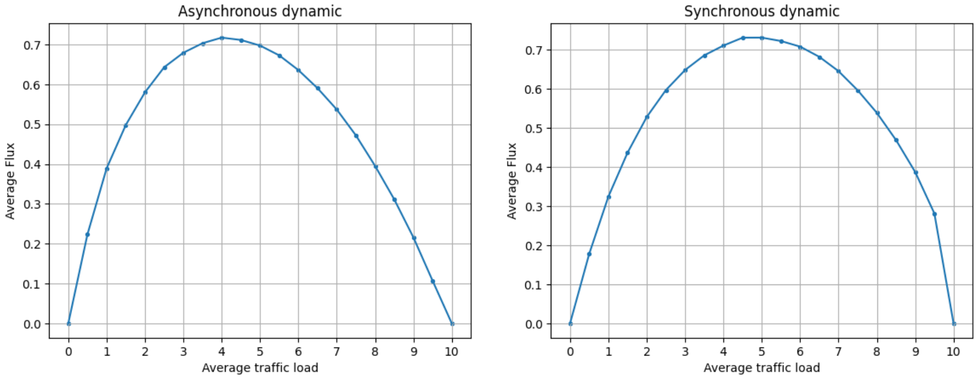
In the synchronous dynamics at each time step each node can move the cars it stores at the same time. This dynamic allows a node to exceed the maximum traffic load at a given time step, since if its degree is $k \geq 2$, it can receive, in a single time step, cars from two different nodes bypassing the limit of 10 maximum cars, that refers to the previous iteration.

On the contrary, the asynchronous dynamics, also called *one step dynamics*, allows only one node to modify its state at each time step, leading to the impossibility of exceeding 10 cars in the internal state of each node. It follows that the number of iterations required to reach the same number of cars moved in the synchronous case is much higher.

I chose 100 as the number of time steps when simulating with the synchronous dynamics, and consequentially simulating the system with an asynchronous dynamics took 722900 iterations, 100 times the number of nodes.

4 Results

To investigate the congestion formation, the first result presented are the average flows per average traffic load per node $\bar{n}$ for asynchronous and synchronous dynamics, shown in Figure 2. As expected, at low average load values the flux increases with the load since there are no congested nodes and cars can move freely through the network. The maximum flux is reached in both case around $\bar{n} \simeq n^{max}/2$, pointing the load at which the transport efficiency reaches its peak. Further increasing of the load leads to an increase of the number of congested nodes, producing a decrease of the flux even if the total number of cars is increasing. Observing the two plots it's relevant to note that in the asynchronous dynamics the peak is reached at $\bar{n} = 4$, slightly before the half of the maximum traffic load. The system becomes less efficient at lower loads due to the nature of its dynamics. Updating node states immediately let congestion propagate locally during an iteration, i.e. if a node becomes full it cannot accept more incoming flux and has to wait several iterations to generate outgoing flux.

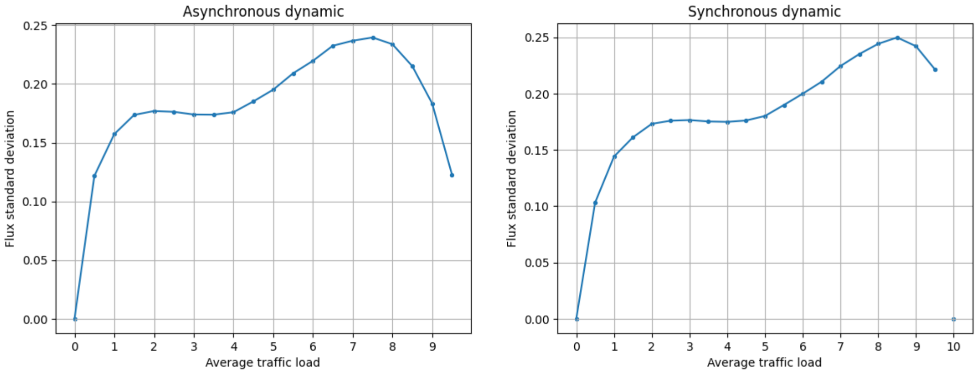


(a)

(b)

Figure 2: Average traffic flux vs. average traffic load per node in **a)** Asynchronous and **b)** Synchronous dynamics

To further investigate the arise of congestion I plotted the standard deviations of flux values, i.e. a measure of the fluctuations respect to the mean flux, as a function of average load, as can be seen in Figure 3 . We can observe the same trend for both dynamics: there is an initial increase up to $\bar{n} \simeq 2$ due to a rapid dispersion between node flows at low traffic load, then a plateau between $2 \leq \bar{n} \lesssim 5$ indicates a stabilization of the standard deviation due to a more evenly distribution of traffic load among nodes, this region is consistent with the one of maximum average flux. A second increase between $5 \lesssim \bar{n} \lesssim 7$ indicates that an increasing amount of nodes starts to saturate while some others others remain more empty, increasing the inequality between flows again. The decline of flux standard deviation after $\bar{n} \gtrsim 7$ is the evidence of the critical traffic load after which the system enters advanced congestion, the saturated nodes cannot handle any more traffic, so the flow on them drops dramatically. If almost all nodes saturate or drop traffic, the variability between them decreases.



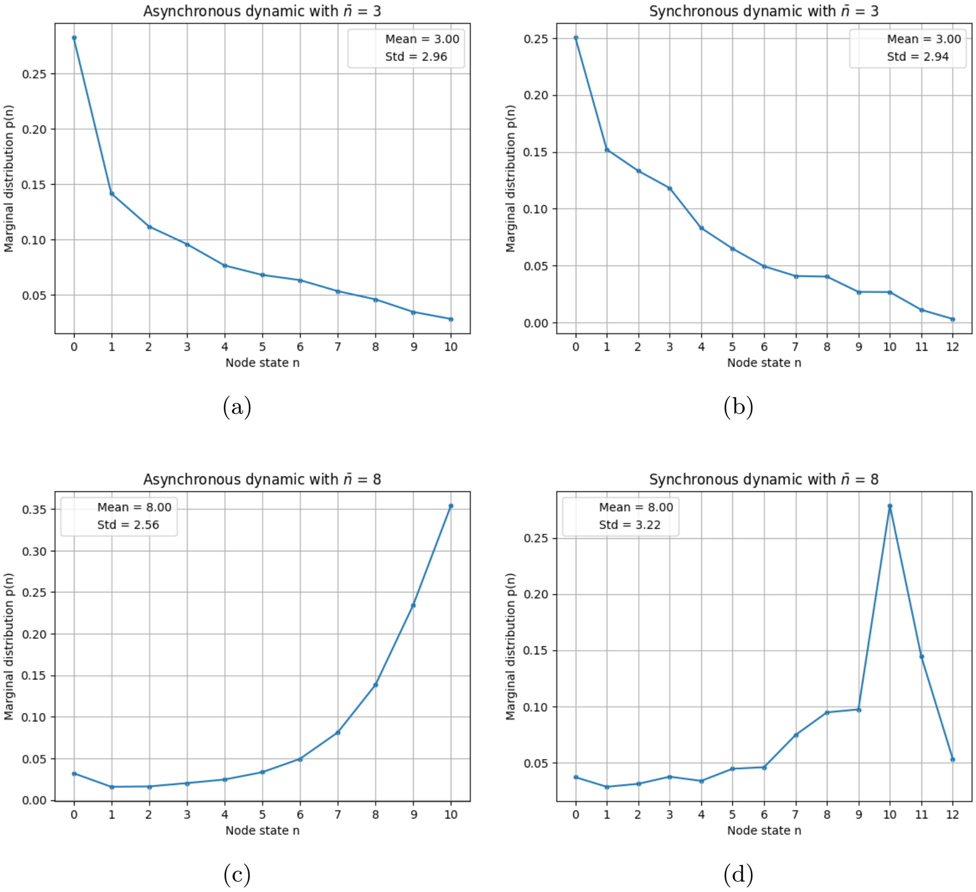
(a)

(b)

Figure 3: Standard deviation of flux values vs. average traffic load per node in **a)** Asynchronous and **b)** Synchronous dynamics

We notice again a difference in the curve shape between one step and synchronous dynamics, that is consistent with the differences in the average flux plot. The end of the plateau, indicating the start of the congestion, happens at $\bar{n} \simeq 4$ for the one step dynamics and at $\bar{n} \simeq 5$ for the synchronous dynamics. The critical point at which the congestion is present all over the network happens at $\bar{n} \simeq 7.5$ and $\bar{n} \simeq 8.5$ respectively for one step and synchronous dynamics.

In Figure 4 are shown the probability distribution $p(n)$ of finding a node with traffic load n in both dynamics with $\bar{n} = 3$ and $\bar{n} = 8$. The curves have a decreasing exponential behavior at low traffic load and an exponential behavior at high traffic load, with the distribution peak at the congested nodes for the latter. The mean of those distribution are equal, as expected, to the relative average traffic load. Since both $\bar{n} = 3$ and $\bar{n} = 8$ are examples of traffic load resulting in a low average flow, we notice how the reduction of traffic flow is linked to a high probability of observing both empty or congested nodes. The presence of states with $n > n^{max} = 10$ in the synchronous dynamics is due to the fact that at each iteration if a node has a degree $k \geq 2$ it can receive cars from more than one node at the same time. The frequency of the overloaded node is fast decaying due to the constraint of the maximum load $\phi_{ij}(n_i, n_j) = 0$ if $n_i > n^{max}$. Figure 5 instead shows the curves of the $p(n)$ standard deviations as a function of the average traffic load for both dynamics. In both case, as expected, the maximum is reached at $\bar{n} = n^{max}/2 = 5$ for the one step dynamics and at $\bar{n} = 6$ in the synchronous case, indicating an high variability of states in which a node can be found right before congestion start.



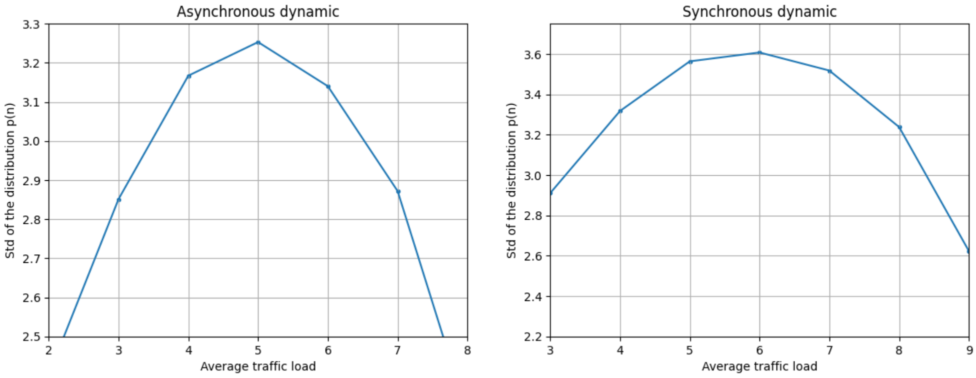
(a)

(b)

(c)

(d)

Figure 4: **Top**: Probability distribution $p(n)$ of finding a node with traffic load n with average traffic load $\bar{n} = 3$ for Asyncrhonous (**a**) and Synchronous (**b**) dynamics. **Bottom**: Probability distribution $p(n)$ with $\bar{n} = 8$ for Asyncrhonous (**c)**) and Synchronous (**d)** dynamics.



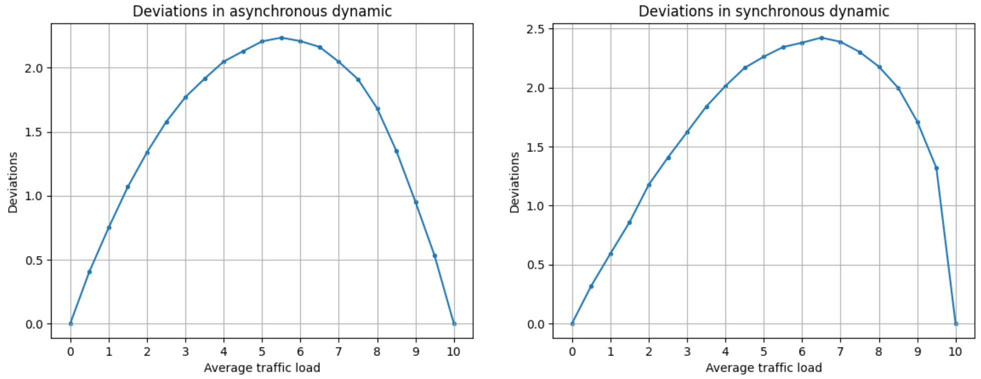
(a)

(b)

Figure 5: Standard deviation of probability distribution $p(n)$ of finding a node with traffic load n vs. average traffic load per node $\bar{n}$ in **a)** Asynchronous and **b)** Synchronous dynamics

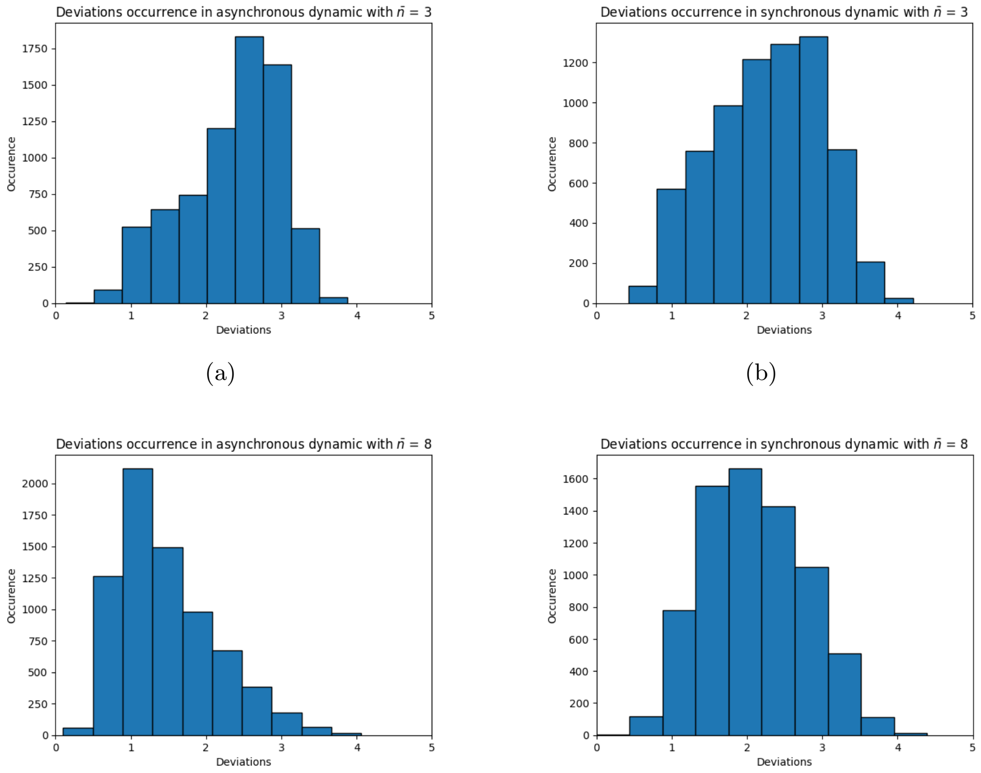
Finally the last measures I show are the deviations, computed calculating the mean square deviation on the population of the individual nodes. They are a measure of the temporal variability of a node. Deviations as a function of traffic load are plotted for one step and synchronous dynamics in Figure 6 . They of course never approach 3, the value of the transport capacity, the higher limit of how many cars a node can move at each time step. The one step dynamics curve is more symmetric respect to the synchronous dynamics case, and again the latter is shifted to the right. Both curves reaches their maximum slightly after $\bar{n} = n^{max}/2$, when the congestion starts to arise. The trend of those curves mimic the ones in Figure 5, referring to the standard deviations of the probability distributions, establishing a link between local quantities (deviations) and global quantities computed on the entire network state (probability distributions), suggesting that the analysis of a limited number of streets can give information about global network properties.

One last interesting plot is the histogram of the mean of deviations over all nodes with two traffic load configuration $\bar{n} = 3$ and $\bar{n} = 8$, shown in Figure 7. Simulations with synchronous dynamics result again in a more right-shifted distribution compared to the asynchronous one, because of the intrinsic nature of the dynamics previously discussed. It's interesting to notice how the distribution peak value drastically decreases when going from a low to an high traffic load, especially in the asynchronous case, clear symptom of the congestion.



(a) (b)

Figure 6: Deviations as a function of average traffic load per node $\bar{n}$ in **a)** Asynchronous and **b)** Synchronous dynamics



(a) (b)

(c) (d)

Figure 7: **Top**: Mean of deviations over all nodes with average traffic load $\bar{n} = 3$ for Asyncrhonous (**a**) and Synchronous (**b**) dynamics. **Bottom**: Mean of deviations over all nodes with average traffic load $\bar{n} = 8$ for Asyncrhonous (**c**)) and Synchronous (**d**) dynamics.

5 Conclusions

This report explores the use of a Markov random process on a graph to highlight the macroscopic features of an urban network dynamics. In particular the development of a simulation of a random walk of cars through nodes with a constraint on the maximum node load and on the maximum flow rate turned out to be effective in studying the emerging of congestion over the Bologna urban network.

Such simulation was carried out considering two dynamics: asynchronous dynamics, when a randomly chosen node moves at each time step, and a synchronous dynamics, when all nodes move simultaneously. The various measurements we can extract from simulations show how the congestion mechanism starts to arise after the average traffic load reaches the half of the maximum load permitted, and becomes predominant after a traffic load of 7 cars per node. The most efficient transport is reached when the traffic load is around 5, half of the maximum load, when the probability distribution of finding a node in the state n has the maximum standard deviation.

The analysis of deviations suggest also a link between local and global quantities, since the behavior of the deviations of a node state load per average traffic load mimic the standard deviation curve of probability distribution.

On the other hand the limitations of such approach are the assumption of a balanced network, where the average incoming and outgoing flows at each node are equal, the absence of interaction between cars and the absence of the complex dynamics is the one related to traffic light. The lack of reality lies in the equal velocity with which cars come out of a node and are injected into the next one, whether the node is congested or not.

References

[1] Sergio Porta, Paolo Crucitti, and Vito Latora. “The network analysis of urban streets: A dual approach”. In: *Physica A: Statistical Mechanics and its Applications* 369.2 (2006), pp. 853–866. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2005.12.063. URL: https://www.sciencedirect.com/science/article/pii/S0378437106001282.

[2] Enrico Lenzi. “Caratterizzazione della nascita di stati congestionati in una rete stradale”. Bachelor thesis. University of Bologna, 2023. URL: https://amslaurea.unibo.it/id/eprint/31426/.

[3] Lorenzo Di Meco et al. *Congestion transition on random walks on graphs*. 2024. arXiv: 2405.16100 [physics.soc-ph]. URL: https://arxiv.org/abs/2405.16100.