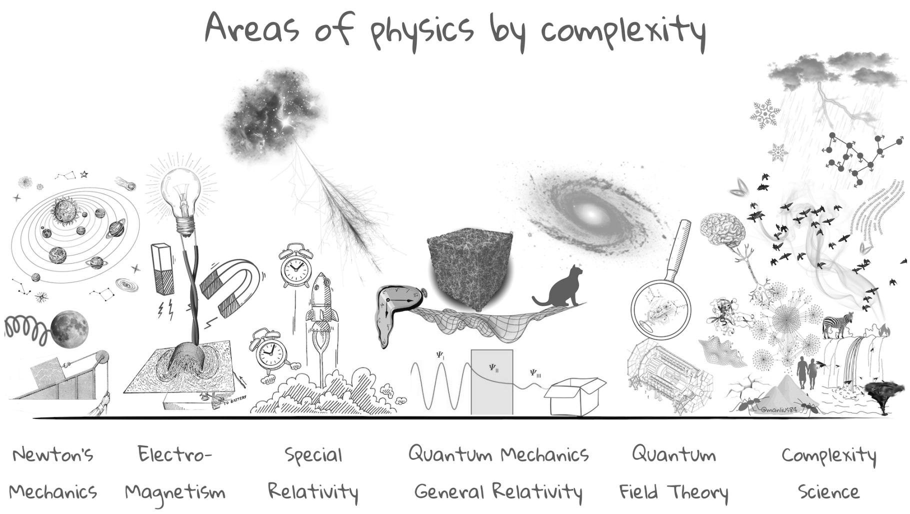
Final Report

Physics of Complex Networks: Structure and Dynamics

Università degli Studi di Padova



Complex Networks Projects

Foster, Marco Tavis

Last update: July 13, 2025

Contents

1 | Axelrod model for global polarization

Task leader(s): *Marco Tavis Foster*

1.1 | Introduction

In this project we develop a model that explores cultural dissemination - with a specific focus on how nodes that have more similarities become more similar as they interact [2]. Over the simulation we see distinct cultural clusters develop and global polarisation emerge where all global clusters are distinct enough where further interactions cannot occur - we reach dynamic stability. While clusters with completely different traits can no longer interact, local nodes that are similar will converge.

1.2 | Exploring a 10x10 grid

We first experiment with a 10x10 grid of 'villages' (represented by nodes) where each village has a number representing their culture. Each different digit in the number represents the number of cultural features each village has, and the different number a digit can take represents different cultural traits. The more features two neighbouring villages have in common the more similar their cultures are and the more likely they will interact to become more similar. We witness this setup in Fig. 1.1.

Once the setup is created we can add dynamic cultural evolution to the system. For each timestep:

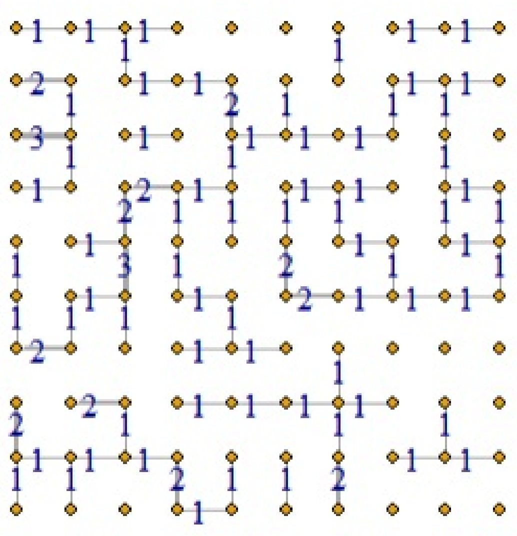
1) A random village is selected and one of its neighbours is randomly selected too.

2) The similarity of the culture of these two villages is determined (for how many features they have the same trait).

3) A random number between 0 and 1 is determined and we check if it is smaller than the ratio of the number of shared features/number of total features.

4) If it is smaller than this ratio then we select a random feature of the node's culture to gain the same trait as that of its neighbour - these two villages' culture becomes more similar.

Under these rules we allow for cultures that are similar to become more similar, while cultures that share less features in common are less likely to become more similar.



01726	18732	28199	52495	08004	57253	22642	95439	64799	56869
22190	52280	61594	23503	20426	45087	11670	47965	07306	60986
67377	65277	33701	31472	20643	85493	93243	43087	09530	73422
81755	91337	85558	05155	15513	32925	69320	92395	76634	36180
22047	79883	69058	03530	91916	12067	09974	69552	78423	76091
42780	30036	69013	35564	64572	92657	62207	02351	89392	25294
62107	69000	55612	27790	60758	50423	14092	65834	12680	49415
01676	38258	37153	69981	97201	29807	94888	18282	45733	04208
03474	47184	10135	88169	05557	41626	64105	49834	68635	13642
87173	75694	92527	70163	86583	02096	69465	34010	83273	27588

Figure 1.1: a) The 10x10 grid of 'villages'. The strength of the connection between villages represent how similar their cultures are. b) The cultural value of each village. In this example every village has 5 cultural features, each of which can have a trait of 0-9.

Observing Figures 1.2-5 we see how the system evolves over time. As the villages interact we see clusters of similar cultures emerge - by the end of the simulation we see that there is one large cluster of villages with the same/very similar cultures, along with a cluster of 5 villages with the same culture and 4 villages that don't share a culture with any of their neighbours.

We repeat the simulation with different variables to test how the final system is different depending on changes to the initial parameters. I investigate varying the number of traits/features, the number of neighbours each village has, and the grid size. To compare differences in the final systems we will compare the number of stable regions - regions where all villages within it share cultural ties. In the last example (Fig 1.5) we observe 6 cultural regions for instance (4 of which are singular villages).

Performing simulations when varying the number of features and number of traits per feature we get the data seen in Table 1.1. We note that increasing the number of traits increases the number of stable region, whilst doing the same with the number of features decreases it. Adding more features provides more opportunities for villages to interact, whereas increasing the number of traits makes it more difficult for a given feature to be the same between neighbours.

Num Features (below), Num Traits (right)	5	10	15
5	1	2	5
10	1	1	1
15	1	1	1

Table 1.1: Simulations with varying number of traits and features per cultural value

We also run experiments where we alter the number of neighbours each village has. In order to visualise this better I changed the grid shape to be a 1D regular lattice where each village is connected to its k-nearest neighbours, where k is variable. We see the results in Table 1.2. The data shows that as k increases the number of stable regions present at the end of the simulation decreases, which makes intuitive sense as the villages have more neighbours to interact with and so are more likely to adopt similar cultures

k	Number of Stable Regions
1	10
2	6
3	3
4	2
5	2
6	2
7	3
8	1
9	1
10	2

Table 1.2: Number of stable regions present as the number of neighbours k for each village changes

Finally we experiment with increasing the number of villages in the system. We choose to utilise the same 1D lattice structure as it is simpler to add nodes to this structure. We fix $k = 2$, num-features = 5 and num-traits = 10. Results are shown in Table 1.3. We see that the number of stable regions increase with the number of villages as we expect.

Number of Villages	Number of Stable Regions
10	3
20	3
30	9
40	11
50	10
60	15
70	19
80	18
90	21
100	26
110	22
120	17
130	32
140	28
150	34

Table 1.3: Number of stable regions against the number of villages

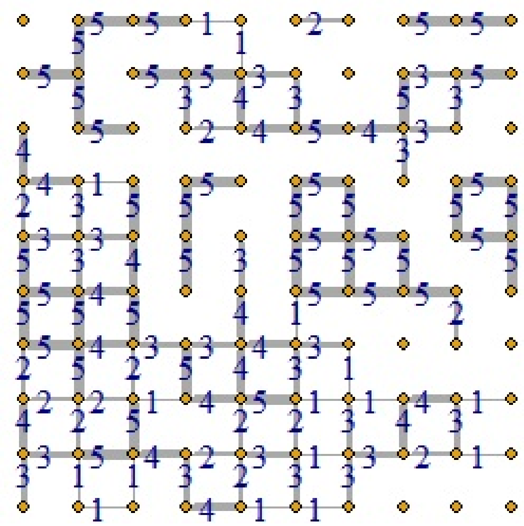


Figure 1.2

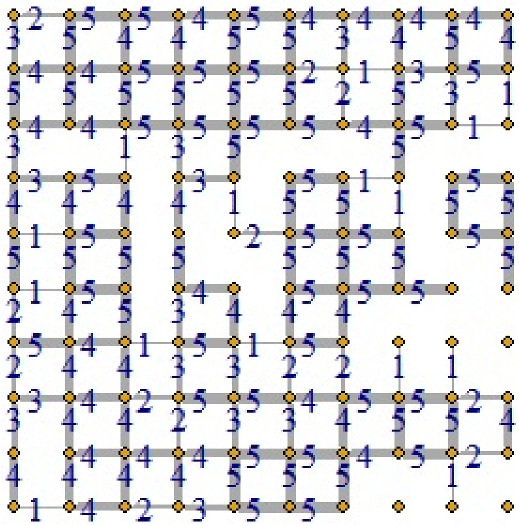


Figure 1.3

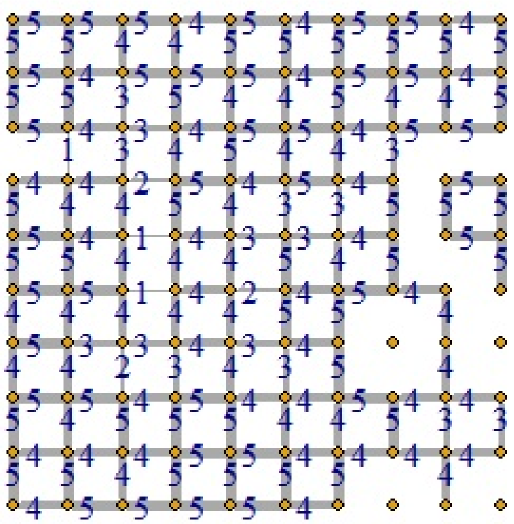


Figure 1.4

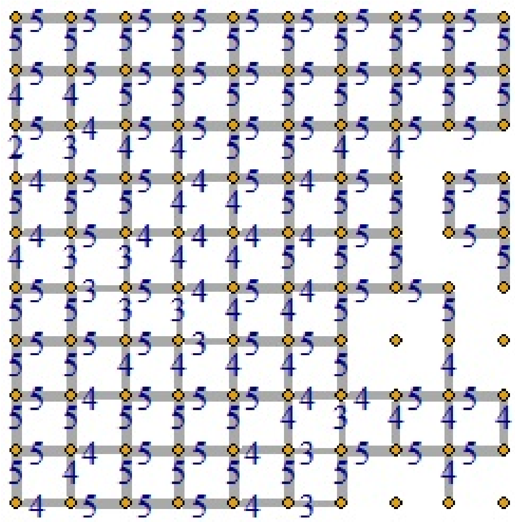


Figure 1.5

Figure 1.6: The state of the system at timsteps: 1.2) t = 20,000. 1.3) t = 40,000. 1.4) t = 60,000. 1.5) t = 80,000

2 | Traffic Congestion

Task leader(s): *Marco Tavis Foster*

2.1 | Introduction

In this project we explore the effects of traffic congestion in a complex system [1] [4]. We will work with a hierarchical network structure and explore how the dynamical system evolves as 'packets' move through it in discrete time steps. We impose that the probability of a packet moving from its current node to a target node is determined by a weight, which is affected by the number of packets already in the current node. The effect is that, as the number of packets in a node grows, the transmissibility decreases exponentially. In this way we witness the effect of traffic congestion on the dynamical processes in a complex network.

2.2 | Method

We begin by selecting the network structure itself, expressly one best suited to witness the movement of the packets and the effects of packet overload. To this end a hierarchical tree structure is chosen; this forces the packets to move through one path as there is only a single path between any pair of nodes. Furthermore, we expect that the root will likely be the first to be overloaded as any packet that needs to travel to the other half of the network must go through the root. Intuitively we can imagine this as the lines of communication between employees at a company, with packets representing information. There are two factors in a tree network structure we can vary:

- z = branching factor (number of children each node has)
- m = number of levels

For our results we will be working with z = 3 and m = 3 - G(z, m) = G(3, 3). A visual representation of this graph can be seen in Fig. 2.1.

An important part of our process is the initialisation of packets. At each time step a packet is created at each node with probability p, and when a packet is generated we assign it a 'destination node' that is not the same as the origin node. At each time step it is also important to move the packets. This is determined by the 'quality of communication, given by:

$$q_{ij} = \sqrt{k_{ij}k_{ji}} \tag{2.1}$$

where k_{ij}, the capability for i to communicate with j, is given by:

$$k_{ij} = \xi_{ij} f(n_a) \tag{2.2}$$

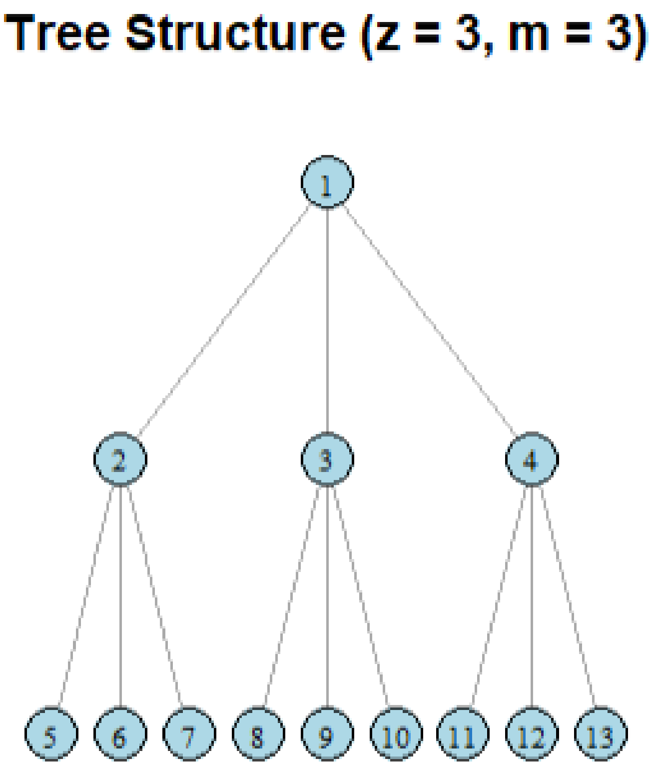


Figure 2.1: This is the caption of the figure.

where ξ_{ij} is a uniformly distributed number between 0 and 1, and $f(n_a)$ is a function that we can select. In this project we elected to set it to:

$$f(n_a) = \begin{cases} 1 & \text{for } n = 0 \\ n^{-\gamma} & \text{otherwise} \end{cases}$$

with $\gamma \geq 0$.

Moving the packet involves the following procedure. First we compute the shortest path (in this case the only path) between its current node and its destination, and we select the next node in the shortest path as the 'target node'. The packet - with probability q_{ij} - moves to this target node. If the target node is the same as the destination mode the packet is 'delivered' and removed from the network. This procedure, along with the generation of new packets, is repeated for n time steps.

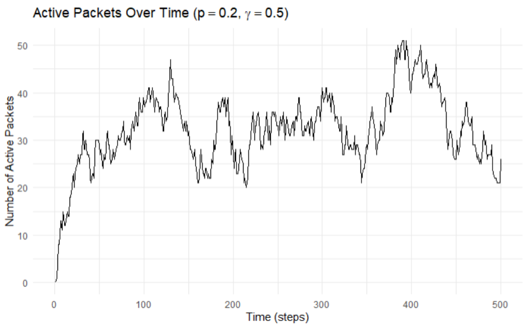


Figure 2.2: Sytem evolution for p = 0.2 and $\gamma = 0.5$

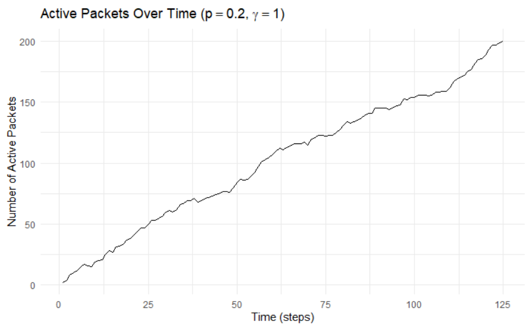


Figure 2.3: Sytem evolution for p = 0.2 and $\gamma = 1$

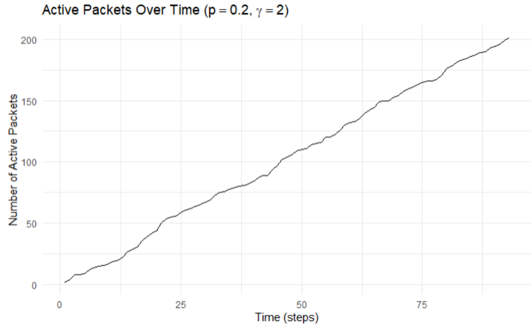


Figure 2.4: System evolution for p = 0.2 and $\gamma = 2$

Figure 2.5: State evolution when the packet generation probability is kept constant and γ is varied. Here we explore three regimes; Fig 2.2) $\gamma \leq 1$, Fig 2.3) $\gamma = 1$, Fig 2.4) $\gamma \geq 1$

For our simulations we explore three regimes:

1. $\gamma < 1$: The number of transmitted packets grow as n_a does. Thus the number of delivered packets increases as N grows until an equilibrium between delivered and generated packets is reached.
2. $\gamma = 1$: The number of delivered packets is constant irrespective of the number of packets already in the system.
3. $\gamma > 1$: The number of delivered packets decreases as n_a does. We therefore expect a dependence on the generation probability in regards to the evolution of the system. If p is small enough the system can handle the number of packets travelling through it and so a steady state is reached. However, if p is large enough then the system will become overloaded - the number of packets in the system will become so large that everything is blocked and eventually no packets can be delivered.

We define p_c, the criticality threshold for the initialising probability. Above this value we expect the bottleneck root node to be overloaded and travelling through it to become impossible. This is given by:

$$p_c = \frac{\sqrt{z}}{\frac{z(z^{m-1}-1)^2}{z^{m-1}}+1}$$

For our network (z = m = 3) $p_c = 0.2065749$.

Finally, I was curious if the breakdown of communication through the root node would cause a cascade failure through the network - where the system is able to bare the 'load' right up until one node no longer functions properly, at which point more nodes fail in a cascade. In our case we do not see this as one node failing does not put additional strain on its neighbouring nodes (for example, by sharing out packets to neighbouring nodes once the node fails). We could explore the effects of an overloaded node on the ability of its neighbouring nodes to communicate with it by modifying the ratio between k_{ij} and k_{ji}, which would change whether communication depends more on the number of packets at the current node or the number of packets at the target node.

3 | Public Transport in the UK

Task leader(s): *Author name(s)...*

3.1 | Introduction

In this project we will create networks representing different cities in the United Kingdom that are made up of all public transport options that stay within that city. We utilise a dataset created by Riccardo Gallotti and Marc Barthelemy [5] that integrates data from the timetables of all different types of public transport options in the UK to create a complete picture of all public transport options for a time period of a week in October 2010. The aim of this project is to recreate layered networks of public transport for UK cities. We will restrict our work to cities with at least 50k inhabitants [6] and, for each city, we will create two files; one for nodes and one for edges. With this data we will then construct our networks for all these cities.

3.2 | Method

3.2.1 Creating the Nodes and Edges Datasets

With the nodes, edges and the inhabitants of each UK city read from the initial data files we proceed with filtering the population data to just cities with at least 50k inhabitants. We then assign our nodes to one of those cities - we compare the longitude/latitude of each node with the longitude/latitude of each city centre and select the one closest to the node. If the node is further than 10km from any city centre then we leave it unassigned (left as NA). This adds a 'city' column in the nodes dataset - the city that the node is in. Once this is done for all nodes we loop through each city and create two datasets for the city's nodes and edges. The procedure for doing this is the following:

1. First we run an if statement to only consider nodes where the city value is not NA.
2. We then filter the nodes to include only those that are in the city we are looking at
3. We also must get the edges that are within this city - these would be the edges that link between two nodes that are both within the relevant city. You can see we filter only for the edges where both the origin and destination nodes are both in the 'city nodes' dataset.
4. With all the nodes and edges in the city accounted for we write two new .csv files for them both. Repeat for each city.

3.2.2 Constructing the Networks

With the data separated into different .csv files dependent on which city the nodes reside in we can move on to using these datasets to construct networks showing these journeys. We create a vector of colours to apply to the edges to represent the different transport options. These are:

YELLOW – Bus

CYAN – Coach

GREEN – Metro

BLUE – Rail

PURPLE – Ferry

RED – Air

We also create a new function called 'calculate relative coordinates'. This function takes values of longitude and latitude for a node and determines the displacement of the node from the city centre. We then convert this into cartesian coordinates so that it can be used effectively in the plotting.

Finally we use everything we have done to plot the networks for each city. We cycle through every relevant city and save the values for the longitude and latitude of the city centre. We then also load the city nodes and edges whilst also filtering out the edges that loop back to the same destination node as its origin node.

We run into an issue where the same node is used for each layer that it exists in (each layer represents the network of a singular transport option) so contracting all layers into one network would result in overlapping nodes where a stop/station services multiple transport options. To prevent this we aggregate all nodes with the same coordinates into one and and have all relevant edges connect to this node.

We now determine the relative coordinates of the nodes to the city centre and use these to plot our network. The networks for the cities can be visualised below. [3]

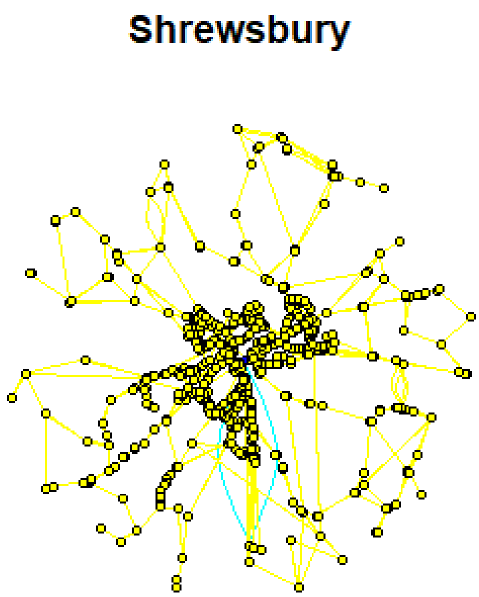


Figure 3.1: Transport map of Shrewsbury, which is an example of a standard map we'd seee in our dataset. The town centre has a dense bus network, the periphary has a sparse bus network, and a few long-range transport options are used to connect Shrewsbury to other towns (coach/train).

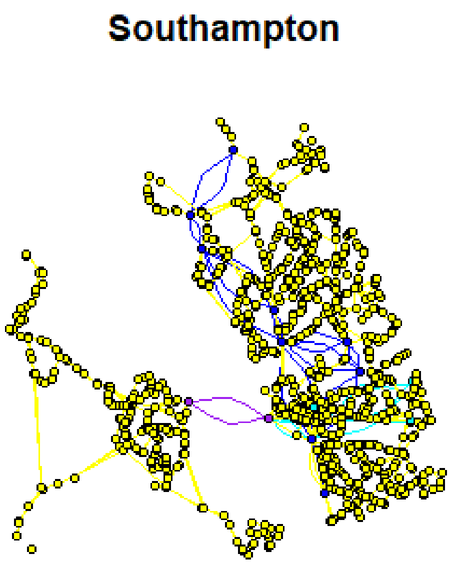


Figure 3.2: Transport map for Southampton, a larger city on the south coast of England. I chose this partly because its familiar to me as my father lives here, but mainly for its unique ferry connection to the Isle of Wight, an island off the coast. The city itself is well connected to other cities through train and coach connections (I personally take the coach from London and then the ferry to the island), but the Isle of Wight has nothing except for buses. This map is an example of how natural obstacles can impact network topology and the types of permissable edges we can have between nodes

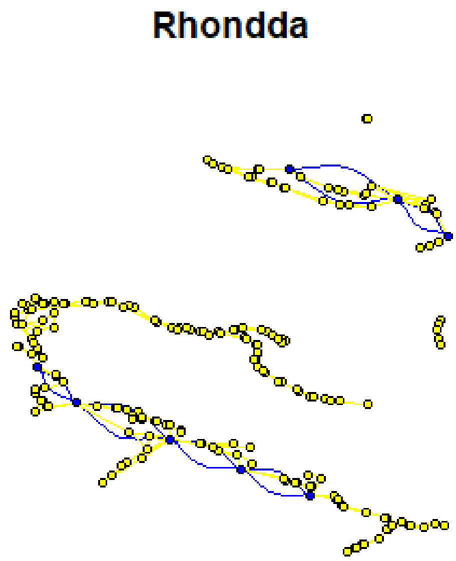


Figure 3.3: Transport map of Rhondda, a small Welsh town located within the Rhondda Valley in the Rhondda Cynon Taf mountain range. This town is another example of how natural, real world features can affect our network, this time due to mountain ranges that restricts the types of transport that can be used. We see as the network goes further into the real-world valley only buses remain as a realistic public transport option.

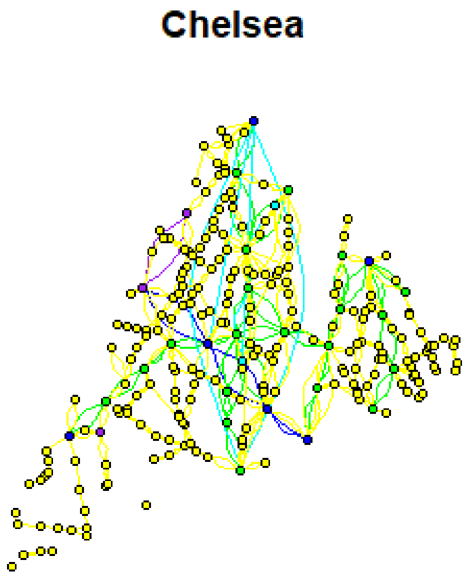


Figure 3.4: Transport map for Chelsea, a borough in London and an example of how areas of an incredibly large city are visualised in our dataset of maps. We see that, compared to other maps, there is an abundance of different transport options; the London Underground (metro), the Overground (trains), the ferry connections on the river Thames and the coaches that service the many coach stations in London. We also especially note the unique shape and smaller number of nodes - this effect is due to London boroughs are much closer together than towns and cities are, meaning they 'steal' nodes that otherwise would have gone to a 'city' like Chelsea. Chelsea was chosen in particular because it is where I went to school (and the location of the football team I support).

4 | Bibliography

[1] A. Arenas, A. Díaz-Guilera, and R. Guimerà. Communication in networks with hierarchical branching. *Phys. Rev. Lett.*, 86:3196–3199, Apr 2001. doi: 10.1103/PhysRevLett.86.3196. URL https://link.aps.org/doi/10.1103/PhysRevLett.86.3196.

[2] R. Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of Conflict Resolution*, 41(2):203–226, 1997. doi: 10.1177/0022002797041002001.

[3] ChatGPT (OpenAI). Assistance with latex formatting and code structure. Personal communication via ChatGPT interface, 2025. [Accessed during project writing].

[4] Pablo Echenique, Jesús Gómez-Gardeñes, and Yamir Moreno. Improved routing strategies for internet traffic delivery. *Phys. Rev. E*, 70:056105, Nov 2004. doi: 10.1103/PhysRevE.70.056105. URL https://link.aps.org/doi/10.1103/PhysRevE.70.056105.

[5] Barthelemy M. Gallotti, R. The multilayer temporal network of public transport in great britain. *Sci Data*, 2, 2015. doi: https://doi.org/10.1038/sdata.2014.56.

[6] World Population Review. United kingdom population by city. https://worldpopulationreview.com/cities/united-kingdom, 2025. [Accessed Apr-2025]: Used to filter cities above 50k inhabitants.