IIB Project Proposal
Information Engineering
Proposed Topic: Reinforcement Learning for N-D network efficiency

Anirudh Bhalekar

April 6, 2024

1 Introduction

I wish to explore network algorithms and metrics, with a particular focus on 2-D networks. I developed an interest in how road networks have evolved throughout history over the past term - and pursued some small projects exploring their efficiency. I explored how 'efficient' routes are, on average, in various cities and towns London, Cambridge, my home town of Gurgaon, India, Los Angeles etc.

The 'efficiency' metric I initially explored (this will be adjusted later) was a simple one - you take two random points, use Dijkstra's algorithm to calculate the most efficient route between them. We then take the euclidean distance between the two points - divide the euclidean distance by the 'path' distance to get a pseudo-efficiency metric (A value of 1 means an ideal path, a value close to 0 means a very inefficient path).

Python's osmnx library was used to get road network, osmnx calls upon the data from open street maps and can create graph data frames which can then be plotted and tested for random routes.

Histograms can be constructed for these road efficiencies:

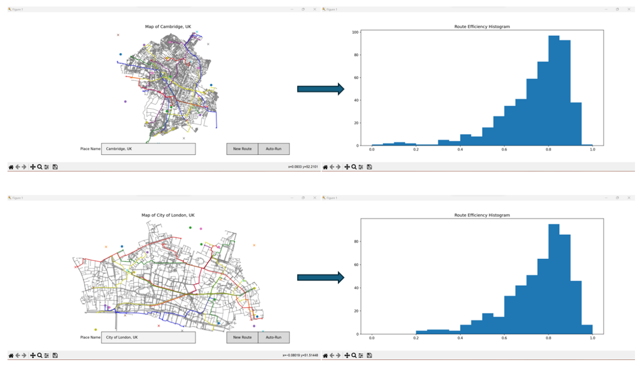


Figure 1: Network Comparison for Cambridge and City of London - with route efficiency histogram

However, if we abstracted this idea to construct a network tessellation, an ideal city - with this metric - would have roads everywhere and no actual area to build anything substantial. We have cost as a limiting factor, and intersections make routes slower too (we want fewer 'turns' and fewer convergence points for traffic). If we want to test an efficiency metric closer to the real world - we must also penalize this in a network.

I have read some literature regarding similar concepts, but haven't seen the analysis I am interested in. Some people have explored road efficiency for truck routes in the US and Europe - purely for academic interest.

Others have used reinforcement learning to analyse *existing* road networks and how to adapt artificially driven vehicles to reduce lane congestion. I will enumerate these studies below with a short description of them (see references for links to studies):

1. [Wolfram, Christopher] - analysis of cross-country truck routes and their efficiency in the US and Europe. Purely a study of factors affecting efficiency - natural formations, economic development, scale etc.

2. [Fan, Xudong, et al.] - uses multiple agents to develop models to expedite network recovery in existing towns

3. [Pung] - How to use graph theory and topological techniques to simplify existing road networks based on traffic.

4. [Rao] - Assessment of current urban networks in a specific city - they evaluate many metrics in the data such as local 'closeness', local 'straightness', and junction weighting

5. [Faster Capital] - Overview study on how to improve network efficiency.

Most papers on the subject are concerned with optimizing route navigation or optimizing existing networks. I'd like to focus on generating networks or looking at transformations to existing networks that are most efficient. While this may seem as if it won't be applicable to any road network design in the real world (as we have historic constraints - we cannot scrap a city and start from scratch) - my intention is to build from 2-D networks to N-D networks. Having a system that generates the best network to optimize paths based on a set criteria and set of constraints can be used in warehouse design (when considering 2-D), design of novel molecules (treating 'traffic' and free energy as interchangeable), semantic image segmentation (compressing a dataset of images down to N length vectors and generating a network that is optimum in path, image segmentation can be done based on a 'radius' of distance). Initially, I would like to start with the 2D case and lay the groundwork.

2 Preliminary Experiments and Derivations

Let's first consider a city-block (or 'Manhattan Style') grid:

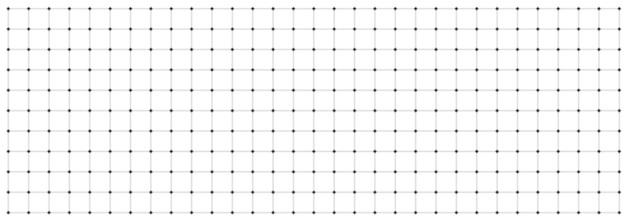


Figure 2: City Block Grid

Sticking with the naive efficiency metric, we can try and derive the expected efficiency of a random route on this grid. If we consider two points to be chosen on the grid from a uniform distribution (for simplicity) - this gives the following path and euclidean distances: $d_{path} = h + v$ and $d_{euclidean} = \sqrt{h^2 + v^2}$ where h and v are the horizontal and vertical distances, and:

$$\eta = \frac{d_{euclidean}}{d_{path}} = \frac{\sqrt{h^2 + v^2}}{h + v}$$

We want:

$$\mathcal{E}[\eta] = \mathcal{E}[\frac{\sqrt{h^2+v^2}}{h+v}] \approx \frac{1}{N}\sum_{i=0}^{N} \frac{\sqrt{h_i^2+v_i^2}}{h_i+v_i}$$

Where h_i and v_i are generated from distributions $f_h(x)$ and $f_v(x)$ respectively. We will consider a few simplifications by considering:

$$\beta = \frac{1}{\eta^2} = 1 + \frac{2hv}{h^2+v^2}$$

$$E[\beta] = 1 + 2E\left[\frac{hv}{h^2+v^2}\right] = 1 + 2E\left[E\left[\frac{hv}{h^2+v^2}|v\right]\right]$$

After integrating this (the key being assuming the lengths h and v are distributed uniformly, in reality if *points* are chosen uniformly, the *lengths* h and v will have a binomial distribution - we will see this is an alright approximation), this gives an expected value of β:

$$E[\beta] = \frac{1}{4}\frac{1}{\alpha}\ln\left(\alpha^2+1\right) + \frac{1}{4}\alpha\ln\left(\frac{1}{\alpha^2}+1\right)$$

Where we define α to be a slenderness ratio ($\alpha = \frac{h}{v}$). We see as α tends to 0 (the ratio of slenderness will tend to infinity) only straight paths will be possible and the 'efficiency metric' will tend to 1.

We also see, this curve will be symmetric in the logarithmic sense (an α value of 2 is functionally the same as $\alpha = 0.5$). We see this agreement through some simulations in figure 3 as the theoretical curve lower bounds the value from simulations.

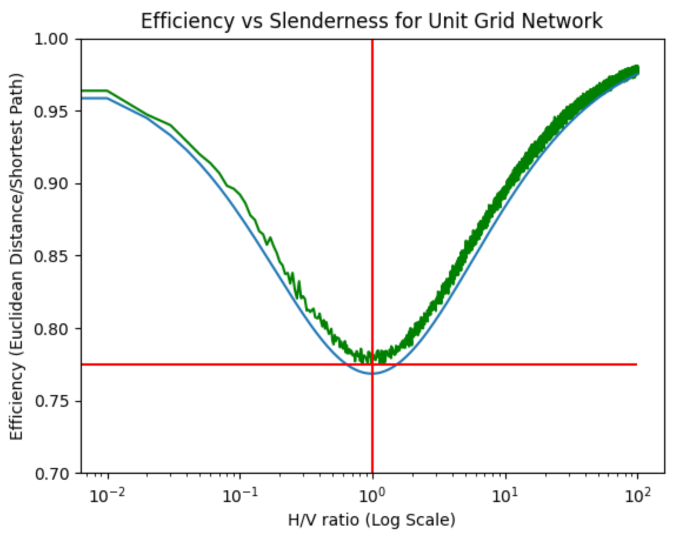


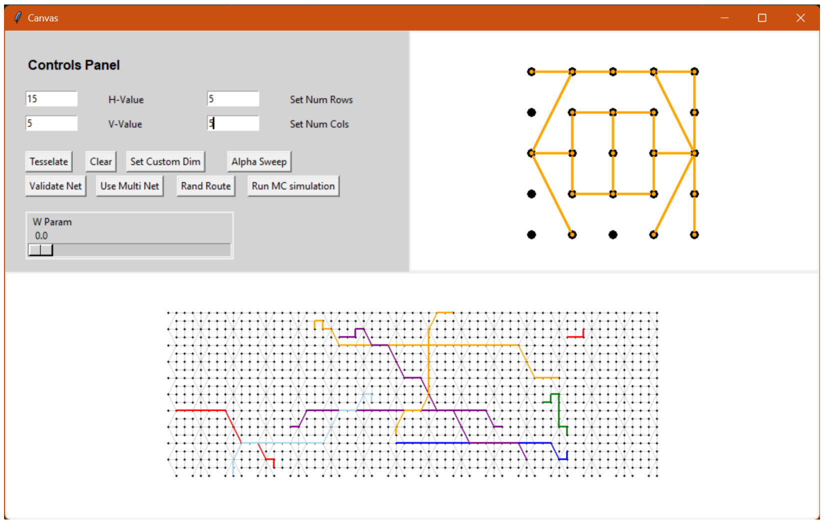
Figure 3: Grid Efficiency Measured (Monte Carlo Simulation) vs Theoretical Curve

Here the simulation takes two *points* at random and computes the 'city block' distance over the euclidean distance (so h and v are chosen from a binomial distribution). We see the theoretical curve (blue) approximates the average efficiency quite well.

We extend this to a simulation of any arbitrary network tessellation with a lightweight python application that tests network efficiency (with added metrics).

3 Application Tests

Figure 4 is a small network testing application I built for testing network efficiency. Networks can be built - for them to be able to tessellate a rectangular grid certain rules have to be fulfilled that I won't enumerate here - but they allow one to travel between any point on the grid to any other point via the network.



15	H-Value	5	Set Num Rows
5	V-Value	5	Set Num Cols

W Param
0.0

Figure 4: Testing application

The network is built by the user, and converted into a dictionary in the form: {(source_node): ((dest_node_1, w1), (dest_node_2, w2), (dest_node_3, w3)), ...}

This is then converted into a full network by tessellating the network horizontally and vertically.

The network is then re-weighted - we inflate weights of nodes that are connected to many nodes (intersection weighting) which is tuned by a parameter W. For a given parameter W - we will multiply each node n's outgoing weight by a term β_n where:

$$\beta_n = W * max\left(\frac{len(n)}{2}, 1\right)$$

So a node with only 1 or 2 connections is not weighted more than it already was. A node with 3 connections is weighted $W * 1.5$ more, 4 connections is weighted $W * 2$ and so on. This works based on a fixed path capacity model.

Random routes can be plotted using Dijkstra's algorithm. Their efficiency is then calculated as the euclidean distance between source and destination nodes over the total weighting.

With intersection weighting a route efficiency metric won't make intuitive sense as we are artificially re-weighting path weights. We will now refer to the efficiency metric as a *relative efficiency metric* - as it can no longer truly equal 1. Networks with higher average *relative efficiencies* will be better with our fixed path capacity model - they offer an average route that doesn't involve many intersections (think less traffic lights to stop at, fewer convergence points for traffic - or in a more abstract sense less 'stress' on the node), and minimizes average travel distance between points. Eventually more parameters can be added to model how real time traffic in any scenario (packet transmission, road traffic etc) is affected in these networks.

Some examples are given below in figure 5, for *relative path efficiencies* for networks with a fixed aspect ratio of the grid being 3:1 - note this is for a W value of 1 - so intersections of 3 paths increase the node weight by 50%, intersections of 4 increase the node weight by 100% etc. We also note: source and destination points are chosen from a uniform distribution

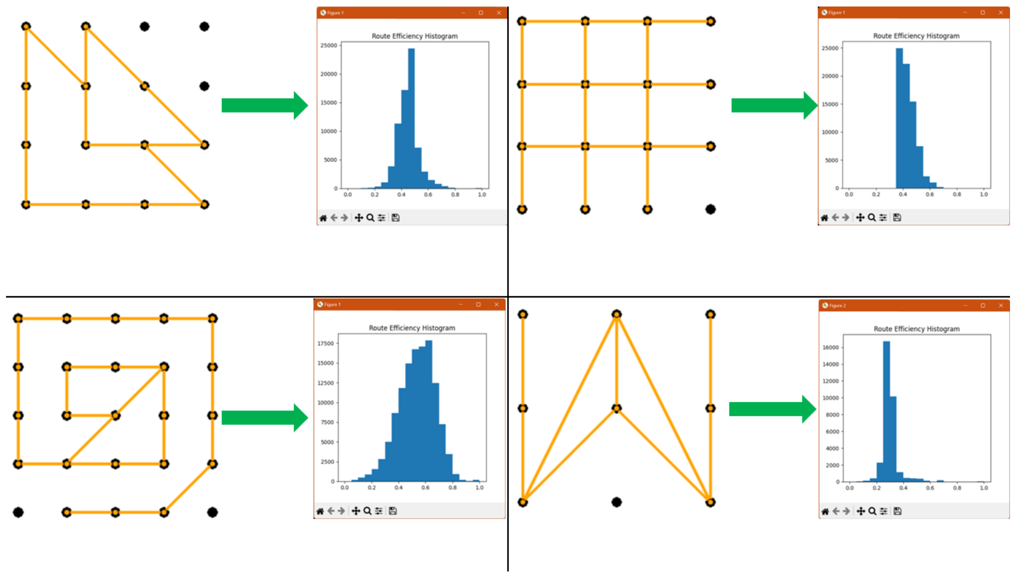


Figure 5: Comparison of relative efficiency histogram (Monte Carlo Simulations) for different networks

4 Extension for IIB Project

Several things interest me regarding this project, while the analysis before is fairly basic, I want to use it as a starting point to explore some ideas about graph networks:

1. With the uniform distribution of points, and a fixed aspect ratio - what would be the theoretical highest average relative efficiency possible. How would one approach this theoretical maximum.

2. We can now expand this to an arbitrary probability distribution over the space - we can use a mixture of Gaussians model to indicate 'popular' source and destination nodes. How would a theoretical best network look if one was varying the parameters of the MoG model - could one use reinforcement learning techniques to draw relationships between parameters of the population distribution and the network tessellation required

3. Abstracting this further, removing the need for a 'tessellation'; given a population distribution - what would a deep learning network draw to maximise distance efficiency whilst minimizing intersectional weighting (ensuring the network is planar with no discontinuities)

4. We can venture into two separate routes for investigation from here - one would be exploring *existing* networks - finding the lowest cost optimization we can perform on said network (i.e. adding/removing a node/path that will have the greatest positive impact on efficiency given its cost) or one can explore how we can *generate* networks that are most efficient from a given space and probability distribution.

5. We can then extend this into N-D space. For 3-D space we can generate a probability distribution; for N » 3 (e.g. for vectorized images - with some dimensionality reduction) we can use an image dataset to generate our p-distribution. How will a neural network with the same motivations build a graph in N-D space that will maximise N-D distance efficiency and minimize number of intersections. And could the properties of this network be used to build semantic relationships between image classes?

5 References

1. [Wolfram, Christopher]. 'Efficiency of Road Networks.' Christopher Wolfram. Accessed 28 Mar. 2024.

2. [Fan, Xudong, et al.] 'A Deep Reinforcement Learning Model for Resilient Road Network Recovery under Earthquake or Flooding Hazards.' Journal of Infrastructure Preservation and Resilience, vol. 4, no. 1, 23 Feb. 2023, doi:10.1186/s43065-023-00072-x.

3. [Faster Capital] 'Road Network Optimization: Maximizing Efficiency with RTTC Strategies.' FasterCapital, fastercapital.com/content/Road-Network-Optimization–Maximizing-Efficiency-with-RTTC-Strategies.html. Accessed 28 Mar. 2024.

4. [Pung] Pung, J., DâSouza, R. M., Ghosal, D., and Zhang, M. (2022). A road network simplification algorithm that preserves topological properties. Applied Network Science, 7(1). https://doi.org/10.1007/s41109-022-00521-8

5. [Rao] Ahmadzai, F., Rao, K. M. L., and Ulfat, S. (2019). Assessment and modelling of urban road networks using integrated graph of natural road network (a GIS-based approach). Journal of Urban Management, 8(1), 109â125. https://doi.org/10.1016/j.jum.2018.11.001