Proposal for a type (b) MEng Project for 2024-25

Name of Student: Anirudh Bhalekar

College: Peterhouse   Email: ab2810@cam.ac.uk

Director of Studies: Dr. James Talbot

Title of Proposed Project: Reinforcement Learning for N-D network efficiency

Does this project involve a collaboration with any organisation outside the Engineering Department? ~~YES~~ / NO

Name of collaborating organisation: ……………………………………

Name of contact person at this organisation: ……………………………………

Address: ……………………………………

……………………………………

Telephone number: ……………………… Email: ………………………

Signature of Student: ………………………

This form should be accompanied by a typed single-sided A4 page description of the proposed project. Form and description should be submitted to the Project Coordinator for the Group most closely related to the subject matter of the project. A list of Project Coordinators is given in the document 'First Notice about Fourth Year Projects'.

Proposals should be submitted as early as possible, preferably before the end of Michaelmas term. **NB: Proposals submitted after Tuesday 23 April 2024 will not be considered.**

NOTE: On receipt of your form, the Coordinator will try to locate a suitable supervisor for your proposed project. All projects must have a departmental supervisor. The Coordinator will contact you as soon as possible.

If a suitable supervisor is **not** available within the Department, you may not do the type (b) project proposed. You should now choose a type (a) project instead (see document 'First Notice about Fourth Year Projects').

If a suitable supervisor **is** available within the Department, the Project Coordinator will tell you their name. You should then contact the supervisor directly to complete the **Project Planning Form** together. You should also enter type (b) project details on COMET as described in the document 'First Notice about Fourth Year Projects'.

FOR COMPLETION BY PROJECT COORDINATOR ONLY:

Date form received: ………………

Suitable Supervisor available: No / Yes (name of staff member): ……………………………

If project involves external organisation approval obtained from Deputy Head (Teaching): No/Yes

Student notified on: ……………… Signed: ………………………

IIB Project Proposal

I wish to explore network generation and associated metrics, with a particular focus on 2-D networks and then extend analysis to N-D networks. I have conducted some basic background studies and analysis on the same, existing studies usually explore topological representations of urban networks or 3-D lattices and explore how to build reinforcement learning systems that allow vehicles to navigate networks that minimize global congestion. I wish to see how to generate networks given a distribution of source and destination nodes – maximizing the path efficiency while minimizing global congestion.

A small network testing application I built using python helped me explore this. Networks can be built - for them to be able to tessellate a rectangular grid certain rules must be fulfilled that I won't enumerate here - but they allow one to travel between any point on the grid to any other point through the network. 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)), ….}

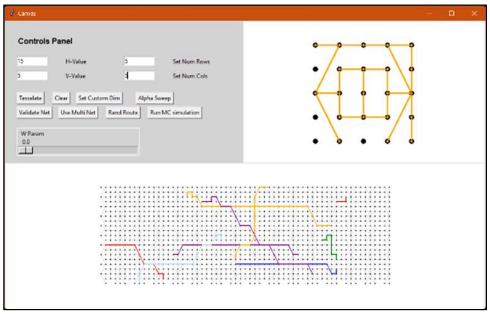

Figure 1: Network Testing Application

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's outgoing weight by a term β_n where – $\beta_n = W * \max\left(\frac{len(node)}{2}, 1\right)$. This will ensure that networks with too many intersections and junctions are penalized.

Random routes can be plotted using Dijkstra's algorithm. Their efficiency is then calculated as the ratio of 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.

Some examples are given below in figure 2, 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.

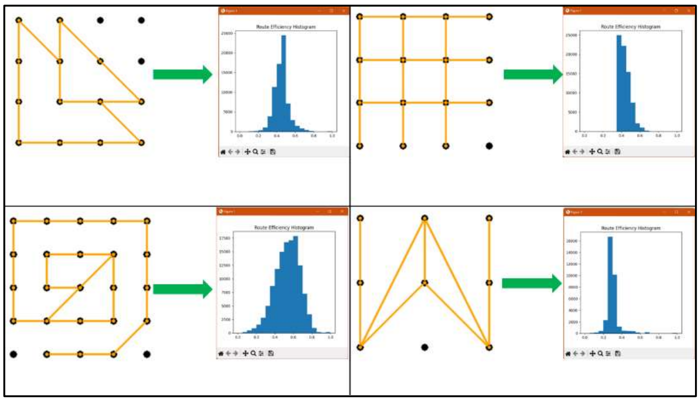

Figure 2: Comparison of relative efficiency histogram (Monte Carlo Simulations) for different networks with aspect ratio 3:1.

Several things interest me regarding this project, while the analysis before is 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.

2. We can now expand this to an arbitrary probability distribution over 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 maximize distance efficiency whilst minimizing intersectional weighting (ensuring the network is planar with no discontinuities)

4. 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 maximize 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?