40.316 Game Theory

Understanding Drivers in Car-lite Singapore and the impact of Cycling

Final Report

Group 5

Authors	*Student ID*
Lee Min Shuen	1004244
Samuel Sim Wei Xuan	1004576
Hazwan Hafiz	1004122
Goh Ray Fong	1003322
Matthew Phua	1004128
Yew Seow Shuen	1004313

August 8, 2022

Executive Summary

Urban planning in Singapore aims to reduce carbon emissions and traffic congestion on her roads. As such, cycling has increased uptake as a transportation option in tandem with developing urban infrastructure that includes and accommodates bicycles. However, having more cyclists and prioritising bike lanes might make car lanes further congested, causing an increasing uptake of bicycles to be counterproductive to easing traffic congestion. Hence, this game theory study aims to verify if the Law of Induced Demand would lower car usage in bike lanes and reduce road congestion. This will be done by developing a Simple Traffic Network model, showing the existence of Braess' Paradox through the addition of capacity in an Augmented Traffic Network model and whether incorporating bicycles and bike lanes helps ease congestion on roads in a Modified Traffic Network model.

The networks created have a latency function to help understand and model latency. According to the most common road impedance function, the Bureau of Public Roads (BPR) function relates travel time to traffic flow. However, we want to model our traffic networks with the following three parameters for a better estimation of latency - the proportion of users travelling a certain path, the average number of lanes along the path, and the average traffic congestion on the path. This is modelled alongside a separate latency function specifically for bicycles, accounting only for travel time on a bicycle. These models are formulated in both the peak hour (0830hrs) and the non-peak hour scenario (1330hrs).

Analysing traffic congestion in both the peak hour and non-peak hour scenarios, with the establishment of these network models and unique holistic latency functions, the Nash Equilibrium and Social Optimum can be both derived and contrasted with one another, discovering optimal solutions for both individual travellers and the society as a whole respectively. With these, we might be able to make insightful conclusions through the usefulness analysis with the Price of Anarchy and spar with Braess' Paradox, to evaluate optimal solutions that our models and latency functions obtain, asserting whether the incorporation of bicycles and bike lanes help ease traffic conditions.

Contents

1 Introduction

1.1 Motivation

Urban road planning in Singapore has changed over recent years, with an increased emphasis on bike lanes. This can be attributed both to the growing desire to reduce carbon emissions as well as an increased uptake in cycling.

A concern about prioritising bike lanes is that the car lanes would get more congested. Still, we hypothesise that the Law of Induced Demand would lower car usage and thus reduce the travel time of road users. Therefore, we seek to formulate a simple road network with game theory and analyse if our hypothesis is true. Specifically, this study will be using the scenario of our own group member who drives and cycles to school, as a way to formulate our traffic network values.

1.2 Objectives of Study

Using game theory reasoning, we know that choosing a specific route to travel from point A to B as a rational individual has very different results from decisions made by considering society as a whole. We need to evaluate these routes taken, in the presence of the number of road users and even the number of lanes, for instance. Furthermore, as mentioned above under motivation, we also need to consider the effect of cycling instead of driving.

Therefore the objectives of the study are to:

1. Develop a traffic network model that accounts for the proportion of users travelling a certain path, the average number of lanes, and the current traffic congestion (during peak and non-peak hour timings);
2. Show how Braess' Paradox exists, where adding capacity to the traffic network might slow down traffic instead;
3. Examine our hypothesis that the introduction of new bike-lanes can help to ease traffic conditions.

2 Problem Formulation

Firstly, we need to come up with a model for an arbitrary transportation network to understand how it reacts to changes in the latency and hence total delay in the network. As such, similar to the lecture, we model the transportation network using a directed graph.

2.1 Directed Graph components

- Edges: Roads
- Nodes: Points along the road or checkpoints

For instance, a simple network is shown in the diagram below, whereby nodes A and B represent the starting and ending locations, respectively. Nodes C and D would be intermediary nodes, which represent two intermediate locations along the path taken to travel from A to B. Following this, each i^{th} edge has a designated latency function, $L_i(x_i)$ that depends on the current congestion level of the path.

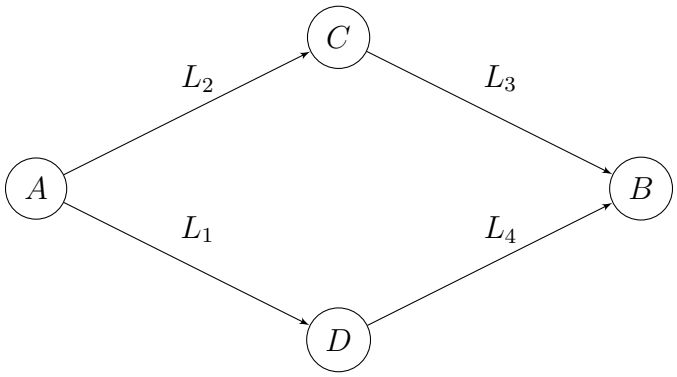

Figure 1: Example of Directed Graph: Traffic network, with each edge labelled by their respective latency function

2.2 Traffic Networks

2.2.1 Simple Traffic Network

For this study, we will narrow down our traffic network to the simplest case and select two arbitrary start and end points in Singapore to test our objectives. The starting location will be Marine Terrace (one of our group member's housing),

and the destination will be the Singapore University of Technology and Design.

Referencing Figure 1 from the previous section, node A and B represent our starting and ending locations, respectively. The top path with latency functions, L_2 and L_3, represents the direction along Upper Changi Road. Following this, the bottom path with latency functions, L_1 and L_4, represents the direction along East Coast Parkway (ECP). More information of the choice of locations for nodes C and D will be covered in the next section.

2.2.2 Augmented Traffic Network

For the augmented traffic network, the augmented route with a congestion-free intermediary link with latency functions, L_5 and L_6, is set to be Xilin Avenue. As such nodes, C and D will be the intersection location between ECP, Upper Changi Road with Xilin Avenue, respectively.

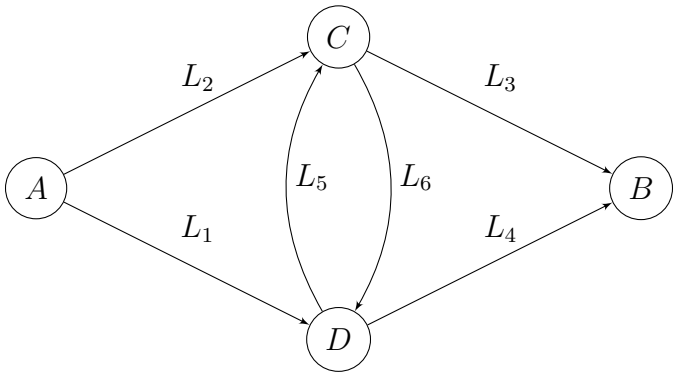

Figure 2: Directed Graph: Augmented Traffic Network, from Marine Terrace to Singapore University of Technology and Design, with inclusion of Xilin Avenue road

The figure below is the augmented traffic network in Google Maps:

Figure 3: Google Maps Traffic Network with Augmented Path

2.2.3 Modified Traffic Network (with Bicycles)

As mentioned in our objectives, our study hypothesizes that riding bicycles as an alternative mode of transport helps to ease traffic conditions. Therefore, we have included a separate augmented route (with a different latency function, Λ_1, explained in a later section) directly from node A to B, which is the cycling route.

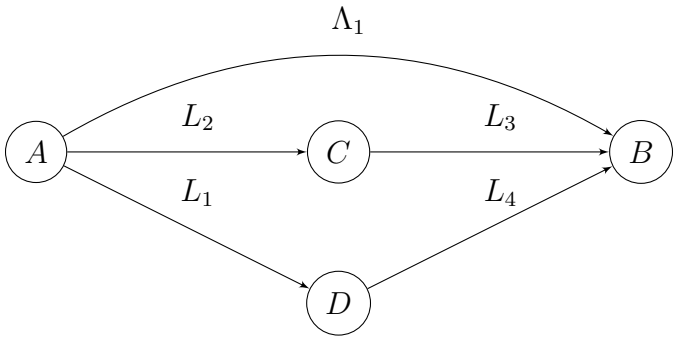

Figure 4: Directed Graph: Modified Traffic Network, from Marine Terrace to Singapore University of Technology and Design, with inclusion of cycling route

The figure below, is the cycling route in Google Maps:

Figure 5: Google Maps cycling route

2.3 Latency Function

The next step is to model the latency function for each edge. According to the 1^{st} objective, our latency function L_i, which measures the delay in the i^{th} edge needs to account for:

1. Proportion of users travelling a certain path;
2. Average number of lanes;
3. Current traffic congestion.

First, let us define the following parameters:

2.3.1 Proportion of users travelling a certain path

To account for the proportion of the users travelling a certain path, our latency function would have to be a function of x_i, which is the proportion of users travelling down i^{th} edge. As our assumption, the total proportion of users in the traffic network will then sum to 1. For example in Figure 1, $x_2 = x_3$ and $x_1 = x_4$ where $x_2 + x_3 = x_1 + x_4 = 1$. In this study, we will use the term proportion of users travelling a certain path and traffic volume on an edge interchangeably.

2.3.2 Average number of lanes

To calculate the average number of lanes per metre along i_{th} edge, Γ_i:

Let Y_i represent the total edge/path length, y_{ij} represent the road length of j^{th} road, γ_{ij} represent the number of lanes of j^{th} road and there are J roads along the i_{th} edge. The path will be discretized into J roads according to the number of lanes. For example, if along the path there are only 3 connected roads with 3 lanes, 2 lanes and 3 lanes respectively, then $J = 3$.

$$\Gamma_i = \frac{1}{Y_i} \sum_{j=1}^{J} (\gamma_{ij} \cdot y_{ij}), \text{ where } Y_i = \sum_{j=1}^{J} (y_{ij}) \tag{1}$$

2.3.3 Current traffic congestion

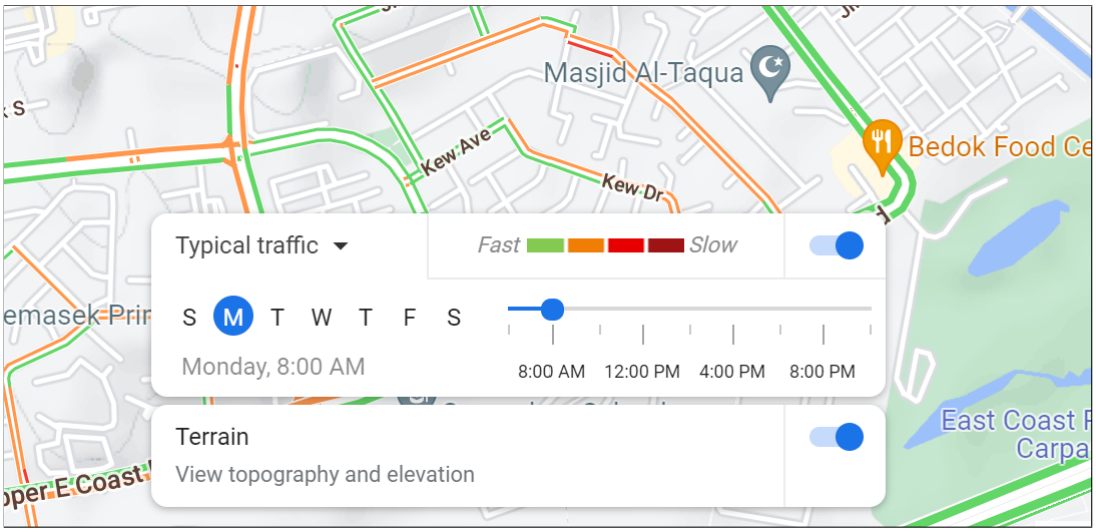

Figure 6: Google Maps typical traffic legend

From the above figure, we can see the typical traffic (state of the traffic) along all the roads which is categorised into 4 shades of varying speed. For our study, we shall assume that a discrete value between 0 and 1 quantifies the traffic congestion along a road. According to the state of the traffic from Google Map:

Traffic congestion along k^{th} road of i^{th} path, $0 \leq \theta_{ik} \leq 1$: {Green: 0.125}, {Orange: 0.375}, {Red: 0.625}, {Brown: 0.875}.

To calculate the average traffic congestion per metre along i_{th} edge, Θ_i:

Let Y_i represent the total path length, y_{ik} represent the road length of k^{th} section, θ_{ik} represent the traffic congestion along k^{th} section and there are K sections along the i_{th} edge. The path will be discretized into K sections according to the state of the traffic. For example, if along the path there are 3 sections with traffic state colours of green, red and green respectively, then $K = 3$.

$$\Theta_i = \frac{1}{Y_i} \sum_{k=1}^{K} (\theta_{ik} \cdot y_{ij}), \text{ where } Y_i = \sum_{k=1}^{K} (y_{ik}) \tag{2}$$

2.3.4 Forming the Latency Function

According to a 2013 research paper [1], the most common road impedance function is the Bureau of Public Roads (BPR) function that relates the travel time to the traffic flow. BPR function for t, time taken to travel between two intersections:

$$\begin{aligned} t &= t_0 \left(1 + \beta \left(\frac{v_a}{c_a}\right)^{\alpha}\right) \text{ where} \\ t_0 &= \text{travel time given 0 traffic volume, minimum time to travel} \\ v_a &= \text{vehicle flow rate of the road (passenger car unit (pcu)/hour)} \\ c_a &= \text{capacity of the road (pcu/h)} \\ \alpha, \beta &= \text{coefficients} \end{aligned} \tag{3}$$

However, in our objective, we want to incorporate the three parameters for every i^{th} edge of our directed graph traffic network.

Firstly, the definition of induced travel is the additional travel that occurs when the cost of travel is lower. It includes the additional travel that is induced by the lower costs resulting from capacity expansion [2]. The research paper published by Ronald Milam, and his research team was aimed at understanding the induced vehicle travel effects. The paper used Vehicular Miles Travelled (VMT) with respect to lane-miles[1] metric, which is the percentage change in total VMT resulting from a percentage change in lane miles (number of lanes times center-line miles)[2] . In a nutshell, the

[1]Note a center-line mile is one mile of a single roadway.

paper summarised that for every 1 percent increase in lane-miles, VMT increases by up to 0.68 percent in a short-term span of 1 year.

Even though most of the research on induced travel is done in the United States, the study types are done in urban and metropolitan areas. Therefore, an assumption is that there is a positive correlation between the number of lanes and the amount of traffic, at least in an urban context, which matches Singapore.

As such, to measure the delay along an edge, we will be using the following latency function for each i^{th} edge:

$$
\begin{aligned}
L_i(x_i, \Gamma_i, \Theta_i) = t_i\left(1 + x_i \cdot \sigma\left(\frac{1}{\Gamma_i \cdot \Theta_i}\right)\right) \text{ where} \\
t_i &= \text{travel time given 0 traffic volume, minimum time to travel} \\
x_i &= \text{proportion of users travelling down the edge, for } 0 \leq x_i \leq 1 \\
\Gamma_i &= \text{average number of lanes per metre along edge, for } \Gamma_i \geq 1 \\
\Theta_i &= \text{average traffic congestion per metre along edge, for } 0 \leq \Theta_i \leq 1 \\
\sigma &= \text{sigmoid function}
\end{aligned}
\tag{4}
$$

The above function is largely similar to the BPR function, function (3). $\Gamma_i \cdot \Theta_i$ can be interpreted as the average number of congested lanes per square metre. To account for the effect of the induced demand relationship, we took the reciprocal of $\Gamma_i \cdot \Theta_i$. Then in order to normalize the unbounded values to between 0 and 1, we applied a sigmoid function.

As such $t_i(x_i \cdot \sigma(1/(\Gamma_i \cdot \Theta_i)))$ is the extra delay in minutes as a fraction of the t_i based on the proportion of users with reference to the induced demand effect of congestion and capacity. Therefore, the latency function would be the summation of t_i and the extra delay. The latency function has a limitation which would be discussed in a later section.

2.3.5 Another Latency function (Bicycles)

Since one of our objectives is to test our hypothesis on how the incorporation

of using bicycles as an alternative mode of transport might help to ease traffic conditions. We need a separate latency function for cycling.

The latency function for riding bicycles will be assumed to be just the time taken to travel between the two nodes on bike, which will be adapted from Google Maps. To differentiate from the previous latency function, we shall denote it using Λ_i.

$$\Lambda_i(x_i) = t_i \text{ , travel time on bicycle} \tag{5}$$

2.4 Peak vs Non-peak Hours

As mentioned in our objectives, we want to do analysis on both peak and non-peak hour periods. Referencing the average speed on roads (peak hours) statistics provided by the Land Transport Authority, AM Peak is from 0800hrs to 0900hrs and PM Peak is from 1800hrs to 1900hrs [3]. The ranges between both peak-hours are assumed to be non-peak hours. Using the middle points of the ranges, we have 0830hrs and 1830hrs for AM and PM peak hours respectively. Likewise for non-peak hours, we have 1330hrs and 0100hrs.

To keep our analysis shorter and more concise, we will be focusing only on 0830hrs (AM peak hour) and 1330hrs (PM non-peak hour). These are more common timings that university students (and our group member) are heading to school for morning and afternoon classes respectively.

3 Dataset

As mentioned in the earlier sections, this study will be narrowing down to a simple traffic network, an augmented traffic network and a modified traffic network. The choice of our arbitrary locations will be Marine Terrace and Singapore University of Technology and Design. As such our traffic networks will be evaluating the scenarios of travelling from Marine Terrace to Singapore University of Technology and Design.

3.1 Data Collection

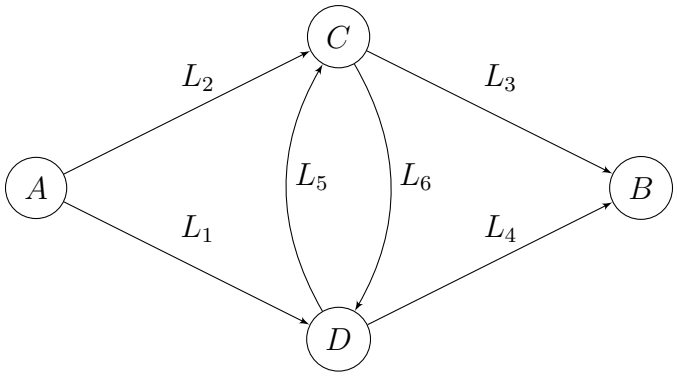

Figure 7: Directed Graph: Augmented Traffic Network, from Marine Terrace to Singapore University of Technology and Design, with inclusion of Xilin Avenue road

Therefore for our data collection, we would need to collect data to model the latency functions for $L_1, L_2, ..., L_6$ where $L_i(x_i, \Gamma_i, \Theta_i) = t_i(x_i \cdot \sigma(1/(\Gamma_i \cdot \Theta_i)))$. Specifically we need to measure t_i, Γ_i, Θ_i for all the edges. Recall, t_i is the travel time given 0 traffic volume, minimum time to travel, Γ_i is the average number of lanes per metre along the edge and Θ_i is the average traffic congestion per metre along the edge.

Nodes: Locations (Latitude, Longitude):

1. Node A: Marine Terrace
   (1.30556, 103.91534)

2. Node B: Singapore University of Technology and Design
   (1.34163, 103.9634)

3. Node C: Intersection between Xilin Avenue and Upper Changi Road
   (1.33241, 103.95494)

4. Node D: ECP exit 2B towards Xilin Avenue
   When driving towards C
   (1.32266, 103.97028)

5. Node D: Towards ECP from Xilin Avenue
   When driving from C
   (1.32680, 103.97291)

3.1.1 Collecting travel time

Firstly, for the travel time, t_i, given 0 traffic volume: we choose a timing of 0300hrs in the early morning where we assumed that there would be little to no traffic and selected the lowest estimated travel timings. The routes match those shown above in Figure 3. The timing according to Google maps are:

	t_i/min
From (A) to (D)	7
From (A) to (C)	10
From (C) to (B)	5
From (D) to (B)	7
From (C) to (D)	3
From (D) to (C)	3

Table 1: Google Maps driving travel times, t_i

3.1.2 Collecting cycling timing

Additionally, for our modified traffic network, with an additional augmented cycling route (with a different latency function, Λ_1) directly from node A to B, we need to measure Λ_1. Λ_1 is the time taken to cycle from Marine Terrace to Singapore University of Technology and Design. Since we assumed cycling to have 0 congestion, we standardized the same timing of 0300hrs on Google Maps. The cycling route matches the one shown above in Figure 5. The timing according to Google maps is:

	Λ_1/min
Cycling from (A) to (B)	34

Table 2: Time taken to cycle from start to end location

3.1.3 Collecting average number of lanes

Next, for Γ_i, the average number of lanes per metre along the i^{th} edge. Recall, let Y_i represent the total edge length, y_{ij} represent the road length of j^{th} road, γ_{ij} represent the number of lanes of j^{th} road and there are J roads along the i_{th} edge.

$$\Gamma_i = \frac{1}{Y_i}\sum_{j=1}^{J}(\gamma_{ij} \cdot y_{ij}), \text{ where } Y_i = \sum_{j=1}^{J}(y_{ij}) \tag{6}$$

(A) to (D)	**(A) to (C)**	**(C) to (B)**	**(D) to (B)**	**(C) to (D)**
{110: 1}	{110: 1}	{1830: 3}	{1600: 3}	{2200: 3}
{240: 2}	{240: 2}	{732: 1}	{450: 1}	
{540: 3}	{540: 3}		{1300: 3}	**(D) to (C)**
{400: 1}	{400: 2}		{700: 2}	{2200: 3}
{6900: 3}	{1000: 3}		{750: 3}	
	{1400: 2}		{150: 5}	
	{1700: 3}		{680: 3}	
	{700: 2}		{732: 1}	
	{1300: 3}			
	{150: 4}			

Table 3: Number of lanes for each edge from Google Maps, $\{y_{ij} : \gamma_{ij}\}$

The table above will be used for the calculations of Γ_i using equation (6). For each column, i^{th} edge, we have discretized it into sequential roads with similar number of lanes. Hence, each row of {Distance/m: Number of lanes}, $\{y_{ij} : \gamma_{ij}\}$ is calculated accordingly using Google Maps.

Note for the calculation of nodes D to B and A to D, we shall standardize D location to be the ECP exit 2B towards Xilin Avenue even though node D has two different geo-coordinates based on the direction of travel. This applies to the later calculations also.

Then the table below shows the resulting average number of lanes per metre for all the edges:

	Γ_i/Avg number of lanes per metre
Γ_1 from (A) to (D)	2.85
Γ_2 from (A) to (C)	2.63
Γ_3 from (C) to (B)	2.43
Γ_4 from (D) to (B)	2.57
$\Gamma_5 = \Gamma_6$ to and from (C), (D)	3.00

Table 4: Γ_i, average number of lanes per metre along edge rounded to 3 sig. figures

3.1.4 Collecting average congestion levels

Lastly, for Θ_i, the average traffic congestion per metre along the i^{th} edge. Recall, let Y_i represent the total path length, y_{ik} represent the road length of k^{th} section, θ_{ik} represent the traffic congestion along k^{th} section and there are K sections

within the i_{th} edge: θ_{ik}: {Green: 0.125}, {Orange: 0.375}, {Red: 0.625}, {Brown: 0.875}. Note for the average congestion levels, we will be calculating for both 0830hrs and 1330hrs as mentioned earlier.

$$\Theta_i = \frac{1}{Y_i} \sum_{k=1}^{K} (\theta_{ik} \cdot y_{ij}), \text{ where } Y_i = \sum_{k=1}^{K} (y_{ik}) \tag{7}$$

(A) to (D)	**(A) to (C)**	**(C) to (B)**	**(D) to (B)**	**(C) to (D)**
{450: 0.375}	{520: 0.375}	{400: 0.125}	{4100: 0.125}	{850: 0.375}
{250: 0.625}	{230: 0.125}	{820: 0.375}	{1300: 0.375}	{1350: 0.125}
{190: 0.375}	{1100: 0.375}	{30: 0.125}	{1012: 0.125}	
{7300: 0.125}	{100: 0.125}	{420: 0.375}		
	{200: 0.375}	{1012: 0.125}		**(D) to (C)**
	{300: 0.125}			{1300: 0.125}
	{300: 0.375}			{600: 0.375}
	{80: 0.125}			{100: 0.125}
	{200: 0.375}			{200: 0.375}
	{220: 0.125}			
	{850: 0.375}			
	{400: 0.125}			
	{350: 0.375}			
	{160: 0.125}			
	{440: 0.375}			
	{110: 0.125}			
	{590: 0.375}			
	{150: 0.125}			
	{500: 0.375}			
	{100: 0.125}			

Table 5: Traffic congestion for each edge from Google Maps, $\{y_{ik} : \theta_{ik}\}$ for 0830hrs

(A) to (D)	**(A) to (C)**	**(C) to (B)**	**(D) to (B)**	**(C) to (D)**
{240: 0.375}	{240: 0.375}	{300: 0.125}	{5000: 0.125}	{1100: 0.125}
{110: 0.625}	{110: 0.625}	{800: 0.375}	{300: 0.375}	{100: 0.375}
{450: 0.375}	{350: 0.375}	{1462: 0.125}	{1012: 0.125}	{1000: 0.125}
{7390: 0.125}	{1050: 0.125}			
	{45: 0.375}			**(D) to (C)**
	{250: 0.125}			{950: 0.125}
	{30: 0.375}			{150: 0.375}
	{220: 0.125}			{1100: 0.125}
	{735: 0.375}			
	{520: 0.125}			
	{550: 0.375}			
	{550: 0.125}			
	{200: 0.375}			
	{190: 0.125}			
	{160: 0.375}			
	{150: 0.125}			
	{800: 0.375}			
	{650: 0.125}			
	{100: 0.375}			

Table 6: Traffic congestion for each edge from Google Maps, $\{y_{ik} : \theta_{ik}\}$ for 1330hrs

The tables above will be used for the calculations of Θ_i using equation (7). For each column, i^{th} edge, we have discretized it into sequential road sections with similar state of traffic. Hence, each row of {Distance/m: Traffic condition value}, $\{y_{ik} : \theta_{ik}\}$ is calculated accordingly using Google Maps. Then the table below shows the resulting average traffic congestion per metre for all the edges, do note the higher average traffic congestion during peak hours which is expected:

	Θ_i/Avg traffic congestion per metre	
	0830hrs (AM peak)	1330hrs (PM non-peak)
Θ_1 from (A) to (D)	0.160	0.153
Θ_2 from (A) to (C)	0.308	0.249
Θ_3 from (C) to (B)	0.241	0.203
Θ_4 from (D) to (B)	0.176	0.137
Θ_5 from (D) to (C)	0.216	0.142
Θ_6 from (C) to (D)	0.222	0.136

Table 7: Θ_i, average traffic congestion per metre along edge rounded to 3 sig. figures

For detailed examples of how we utilised Google Maps to obtain the various table values, refer to the appendix's data collection section.

4 Analysis

4.1 Traffic at Equilibrium

Recall our traffic network:

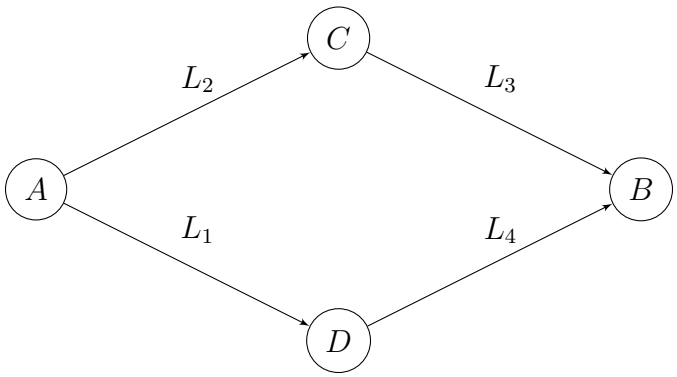

Figure 8: Directed Graph: Simple Traffic Network, from Marine Terrace to Singapore University of Technology and Design

4.1.1 Nash Equilibrium

For AM peak, 0830hrs:

$$\begin{aligned}
\text{Total Delay top path} &= x \cdot (L_2 + L_3) \\
&= 10\left(1 + \sigma\left(\frac{1}{0.81004}\right)\right) + 5\left(1 + \sigma\left(\frac{1}{0.58563}\right)\right) \\
&= 26.98 \text{ (4 sf.)} \\
\text{Total Delay bottom path} &= x \cdot (L_1 + L_4) \\
&= 7\left(1 + \sigma\left(\frac{1}{0.456}\right)\right) + 7\left(1 + \sigma\left(\frac{1}{0.45232}\right)\right) \\
&= 26.61 \text{ (4 sf.)} \\
&\text{where } x = 1, \text{ is all users travelling the path}
\end{aligned} \tag{8}$$

Therefore at the Nash equilibrium, all road users would choose the bottom path, where $x = 1$ and the total delay at equilibrium is 26.61 minutes. This is because, selfish users would want to reach the end location with the shortest delay, which is the bottom path in this case.

For PM non-peak, 1330hrs:

$$\begin{aligned}
\text{Total Delay top path} &= x \cdot (L_2 + L_3) \\
&= 10\left(1 + \sigma\left(\frac{1}{0.65487}\right)\right) + 5\left(1 + \sigma\left(\frac{1}{0.49329}\right)\right) \\
&= 27.63 \text{ (4 sf.)} \\
\text{Total Delay bottom path} &= x \cdot (L_1 + L_4) \\
&= 7\left(1 + \sigma\left(\frac{1}{0.43605}\right)\right) + 7\left(1 + \sigma\left(\frac{1}{0.35209}\right)\right) \\
&= 26.97 \text{ (4 sf.)} \\
&\text{where } x = 1, \text{ is all users travelling the path}
\end{aligned} \tag{9}$$

Therefore at the Nash equilibrium, all road users would choose the bottom path, where $x = 1$ and the total delay at equilibrium is 26.97 minutes. Similar reasoning to the AM peak hour.

4.1.2 Social Welfare

At the social optimum to minimize total delay, for both 0830hrs (AM peak) and 1330hrs (PM non-peak), the optimization problem is:

$$\begin{aligned}
&\min \left(\sum_{i=1}^{4} L_i x_i\right) \text{ where} \\
&L_i = t_i\left(1 + x_i \cdot \sigma\left(\frac{1}{\Gamma_i \cdot \Theta_i}\right)\right) \forall i = \{1, 2, 3, 4\} \\
&x_1 = x_4 \\
&x_2 = x_3 \\
&x_1 + x_2 = 1 \\
&0 \leq x_i \leq 1
\end{aligned} \tag{10}$$

Note the values for Γ_i and Θ_i are taken from table 4 and table 7 respectively. Using cvxpy - Python library for solving convex optimization problems:

```
import cvxpy as cp
import numpy as np

# Variables
t_1, theta_1, gamma_1 = 7, 0.160, 2.85
t_2, theta_2, gamma_2 = 10, 0.308, 2.63
t_3, theta_3, gamma_3 = 5, 0.241, 2.43
t_4, theta_4, gamma_4 = 7, 0.176, 2.57
x = cp.Variable()

# Objective function
objective = cp.sum([(t_2*(x +
↪  x**2*cp.inv_pos(1+cp.exp(-(1/(theta_2*gamma_2))))),
                    (t_3*(x + x**2*cp.inv_pos(1+cp.exp(-(1/(theta_3*gamma_3))))),
                    (t_4*((1-x) +
                    ↪  (1-2*x+x**2)*cp.inv_pos(1+cp.exp(-(1/(theta_4*gamma_4))))),
                    (t_1*((1-x) +
                    ↪  (1-2*x+x**2)*cp.inv_pos(1+cp.exp(-(1/(theta_1*gamma_1))))))
                     ])

# Construct the problem
objective = cp.Minimize(objective)
constraints = [0 <= x, x <= 1]
prob = cp.Problem(objective, constraints)

# Optimal objective value is returned by `prob.solve()`
result = prob.solve()
print("Optimal proportion of users travel to top path is {x_1} and to bottom path
↪  is {x_2}".format(
    x_1=np.round(x.value, 3), x_2=np.round(1-x.value, 3)))
print("Total Delay is {result}".format(result=np.round(result, 3)))
```

Listing 1: Optimization solver for Simple Traffic Network at 0830hrs

```
# Variables
t_1, theta_1, gamma_1 = 7, 0.153, 2.85
t_2, theta_2, gamma_2 = 10, 0.249, 2.63
t_3, theta_3, gamma_3 = 5, 0.203, 2.43
t_4, theta_4, gamma_4 = 7, 0.137, 2.57
x = cp.Variable()

# Objective function
objective = cp.sum([(t_2*(x +
↪  x**2*cp.inv_pos(1+cp.exp(-(1/(theta_2*gamma_2))))),
                    (t_3*(x + x**2*cp.inv_pos(1+cp.exp(-(1/(theta_3*gamma_3))))),
                    (t_4*((1-x) +
                    ↪  (1-2*x+x**2)*cp.inv_pos(1+cp.exp(-(1/(theta_4*gamma_4))))),
                    (t_1*((1-x) +
                    ↪  (1-2*x+x**2)*cp.inv_pos(1+cp.exp(-(1/(theta_1*gamma_1))))))
                     ])

# Construct the problem
objective = cp.Minimize(objective)
constraints = [0 <= x, x <= 1]
prob = cp.Problem(objective, constraints)

# Optimal objective value is returned by `prob.solve()`
result = prob.solve()
print("Optimal proportion of users travel to top path is {x_1} and to bottom path
↪  is {x_2}".format(
    x_1=np.round(x.value, 3), x_2=np.round(1-x.value, 3)))
print("Total Delay is {result}".format(result=np.round(result, 3)))
```

Listing 2: Optimization solver for Simple Traffic Network at 1330hrs

Therefore, when users are non-selfish and aims to minimize the entire traffic network delay, the results are very different from the nash equilibrium case of selfish road users.

From our code, the social optimum proportion of users travelling along the top path is 0.492 and along bottom path is 0.508 for 1330hrs, AM peak hour. Likewise the social optimum proportion of users travelling along the top path is 0.487 and along bottom path is 0.513 for 0830hrs, PM non-peak hour.

The total delay is 20.645 min (0830hrs, AM peak hour) and 20.897 min (1330hrs, PM non-peak hour) at social optimum. These delay values at social welfare optimum levels are much lower than the nash equilibrium scenario.

4.1.3 Price of Anarchy (PoA)

Recall, that PoA is defined as the ratio of social welfare between social optimum and the worst nash equilibrium, and measures how the efficiency of a system degrades due to selfish player behavior.

$$\begin{aligned} \text{PoA}_{\text{peak}} &= (1/20.645)/(1/26.61) = 1.29 \\ \text{PoA}_{\text{non-peak}} &= (1/20.897)/(1/26.97) = 1.29 \end{aligned} \tag{11}$$

Therefore from our analysis, we see that due to selfish behaviours of drivers, the system traffic network performs 29% worse in terms of total delay which is actually quite reasonable.

4.2 Braess' Paradox

Recall our augmented traffic network, with a congestion-free intermediary link:

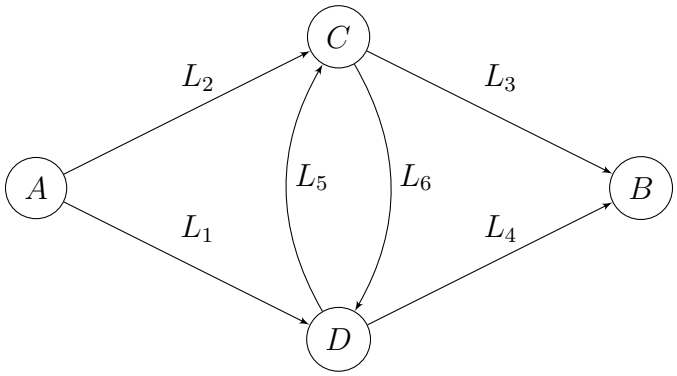

Figure 9: Directed Graph: Augmented Traffic Network, from Marine Terrace to Singapore University of Technology and Design, with inclusion of Xilin Avenue road

Even though we collected data for the augmented path with regards to its traffic state and average number of lanes. For our analysis here, we shall assume a zero-congestion path. Hence the latency function L_5 and L_6 would just be t_5 and t_6 accordingly, without the extra time delay component.

For AM peak, 0830hrs, assuming all selfish users goes along the edge, $x_i = 1$:

$$\begin{aligned}
L_1 &= 7\left(1 + 1 \cdot \sigma\left(\frac{1}{0.456}\right)\right) = 13.30 \text{ (4 sf.)} \\
L_2 &= 10\left(1 + 1 \cdot \sigma\left(\frac{1}{0.81004}\right)\right) = 17.75 \text{ (4 sf.)} \\
L_3 &= 5\left(1 + 1 \cdot \sigma\left(\frac{1}{0.58563}\right)\right) = 9.233 \text{ (4 sf.)} \\
L_4 &= 7\left(1 + 1 \cdot \sigma\left(\frac{1}{0.45232}\right)\right) = 13.31 \text{ (4 sf.)} \\
L_5 &= L_6 = 3(1 + 0) = 3
\end{aligned} \tag{12}$$

For PM non-peak, 1330hrs, assuming all selfish users goes along the edge, $x_i = 1$:

$$\begin{aligned}
L_1 &= 7\left(1 + 1 \cdot \sigma\left(\frac{1}{0.43605}\right)\right) = 13.36 \text{ (4 sf.)} \\
L_2 &= 10\left(1 + 1 \cdot \sigma\left(\frac{1}{0.65487}\right)\right) = 18.22 \text{ (4 sf.)} \\
L_3 &= 5\left(1 + 1 \cdot \sigma\left(\frac{1}{0.49329}\right)\right) = 9.418 \text{ (4 sf.)} \\
L_4 &= 7\left(1 + 1 \cdot \sigma\left(\frac{1}{0.35209}\right)\right) = 13.61 \text{ (4 sf.)} \\
L_5 &= L_6 = 3(1 + 0) = 3
\end{aligned} \tag{13}$$

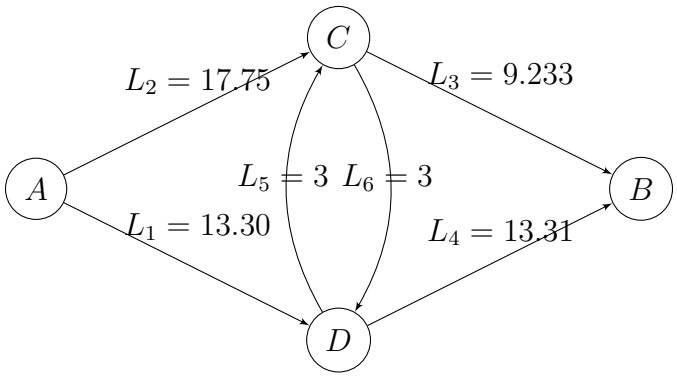

Figure 10: Directed Graph: Augmented Traffic Network with Latency Values (assuming all users follow along that edge (AM peak, 0830hrs)

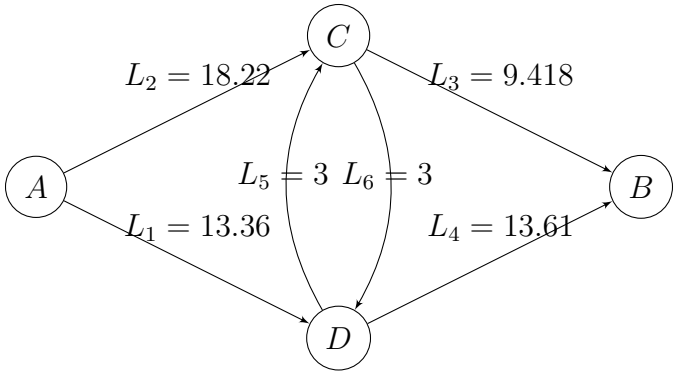

Figure 11: Directed Graph: Augmented Traffic Network with Latency Values (assuming all users follow along that edge (PM non-peak, 1330hrs)

Looking at both figures, we can see that the selfish road users would choose the bottom path, then along the augmented path upwards, followed by the top path (Travelling along node A to D to C to B). This is because for both timings, the delay from $L_2 > L_1$ and the delay from $L_4 > L_5 + L_3$.

With the congestion-free augmented path:

1. AM peak, 0830hrs, Total delay = 25.533 min
2. PM non-peak, 1330hrs, Total delay = 25.778 min

However, recall without the congestion-free augmented path: At Nash equilibrium (rational users):

1. AM peak, 0830hrs, Total delay = 26.61 min
2. PM non-peak, 1330hrs, Total delay = 26.97 min

At social optimum:

1. AM peak, 0830hrs, Total delay = 20.645 min
2. PM non-peak, 1330hrs, Total delay = 20.897 min

The Braess' Paradox scenario in the class lecture assumed both the top and bottom paths to have similar delays, hence the Nash equilibrium results in an equal balance of users. The addition of the congestion free intermediary link resulted in all the rational (selfish) users to deviate towards it, which results in a higher total delay than the Nash equilibrium case.

However, in our current example, both the top paths and bottom paths have different delays. We do not observe the paradox happening here whereby the delay with the congestion-free augmented path is better than our Nash equilibrium (traffic network without the congestion-free augmented path). Yet, the network with the congestion-free augmented path still performs worse than the social optimum (traffic network without the augmented path).

Therefore, even though our study does not show the Braess' paradox happening (for this specific traffic network), it does prove the point that the paradox requires the right combinations of factors to happen in real-life examples and does not happen all the time [4].

4.3 Cycling Route

Recall our modified traffic network with a cycling route incorporated with latency Λ_1:

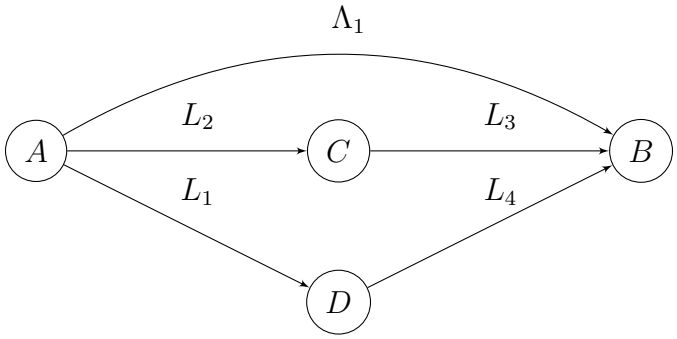

Figure 12: Directed Graph: Modified Traffic Network, from Marine Terrace to Singapore University of Technology and Design

Suppose the proportion of users changing to cycling option is $c = [0.1, 0.2, ..., 0.8, 0.9]$, in intervals of 0.1. As such, $c + \sum_i x_i = 1$. This ignores the rationality of the users changing to cycling (in terms of travel time). Also, we assumed that cycling is congestion free regardless of value of c.

According to Land Transport Authority the width of an inner lane is 3.2m, while the width of a cycling lane according to Land Transport Authority and the Urban Redevelopment Authority is 2m. Hence, even though the cycling timing to derive

Λ_i is not along the road paths of our network, we still assumed the (capacity) average number of lanes per metre is decreased by $2/3.2 = 0.625$ for every edge to accommodate for cycling lanes in our traffic network.

Repeating the same optimization of the simple network with a reduced proportion of users driving (for 0830hrs, AM peak):

```
import pandas as pd
# empty dataframe with 6 columns: top_path, bottom_path, cycling, delay_driving,
↪  delay_cycling, delay_total
df = pd.DataFrame(columns=['top_path', 'bottom_path', 'cycling', 'delay_driving',
↪  'delay_cycling', 'delay_total'])

for i in np.arange(0.1, 1, 0.1):
    # Variables
    t_1, theta_1, gamma_1 = 7, 0.160, (-0.625+2.85)
    t_2, theta_2, gamma_2 = 10, 0.308, (-0.625+2.63)
    t_3, theta_3, gamma_3=5, 0.241, (-0.625+2.43)
    t_4, theta_4, gamma_4=7, 0.176, (-0.625+2.57)
    x_1 = cp.Variable()
    x_2 = cp.Variable()

    # Objective function
    objective = cp.sum([(t_2*(x_1 +
    ↪  x_1**2*cp.inv_pos(1+cp.exp(-(1/(theta_2*gamma_2))))),
                    (t_3*(x_1 +
                    ↪  x_1**2*cp.inv_pos(1+cp.exp(-(1/(theta_3*gamma_3))))),
                    (t_4*(x_2 +
                    ↪  x_2**2*cp.inv_pos(1+cp.exp(-(1/(theta_4*gamma_4))))),
                    (t_1*(x_2 +
                    ↪  x_2**2)*cp.inv_pos(1+cp.exp(-(1/(theta_1*gamma_1)))))
                    ])

    # Construct the problem
    objective = cp.Minimize(objective)
    constraints = [0<=x_1, x_1+x_2 == 1-i, 0<=x_2]
    prob = cp.Problem(objective, constraints)

    # Optimal objective value is returned by `prob.solve()`
    result = prob.solve()
    # add row to dataframe
    df.loc[len(df)] = [np.round(x_2.value,3), np.round(x_1.value,3), i,
    ↪  np.round(result,3), 34*i, np.round(result+34*i,3)]
df
```

Listing 3: Optimization solver for Modified Traffic Network at 0830hrs

Next for 1330hrs, PM non-peak:

```
# empty dataframe with 6 columns: top_path, bottom_path, cycling, delay_driving,
↪  delay_cycling, delay_total
df = pd.DataFrame(columns=['top_path', 'bottom_path', 'cycling',
                  'delay_driving', 'delay_cycling', 'delay_total'])

for i in np.arange(0.1, 1, 0.1):
    # Variables
    t_1, theta_1, gamma_1 = 7, 0.153, (-0.625+2.85)
    t_2, theta_2, gamma_2 = 10, 0.249, (-0.625+2.63)
    t_3, theta_3, gamma_3 = 5, 0.203, (-0.625+2.43)
    t_4, theta_4, gamma_4 = 7, 0.137, (-0.625+2.57)
    x_1 = cp.Variable()
    x_2 = cp.Variable()

    # Objective function
    objective = cp.sum([(t_2*(x_1 +
    ↪  x_1**2*cp.inv_pos(1+cp.exp(-(1/(theta_2*gamma_2))))))),
                        (t_3*(x_1 +
                        ↪  x_1**2*cp.inv_pos(1+cp.exp(-(1/(theta_3*gamma_3))))))),
                        (t_4*(x_2 +
                        ↪  x_2**2*cp.inv_pos(1+cp.exp(-(1/(theta_4*gamma_4))))))),
                        (t_1*(x_2 + x_2**2)*cp.inv_pos(1 +
                         cp.exp(-(1/(theta_1*gamma_1)))))
                        ])

    # Construct the problem
    objective = cp.Minimize(objective)
    constraints = [0 <= x_1, x_1+x_2 == 1-i, 0 <= x_2]
    prob = cp.Problem(objective, constraints)

    # Optimal objective value is returned by `prob.solve()`
    result = prob.solve()
    # add row to dataframe
    df.loc[len(df)] = [np.round(x_2.value, 3), np.round(x_1.value, 3),
                       i, np.round(result, 3), 34*i, np.round(result+34*i, 3)]
df
```

Listing 4: Optimization solver for Modified Traffic Network at 01330hrs

The tables below are the results from our optimization:

Proportion of Users			Delay/min		
Top path	Bottom path	Cycling	Driving	Cycling	Total
0.471	0.429	0.1	18.152	3.4	21.552
0.421	0.379	0.2	15.611	6.8	22.411
0.372	0.328	0.3	13.2	10.2	23.4
0.323	0.277	0.4	10.919	13.6	24.519
0.273	0.227	0.5	8.77	17.0	25.77
0.224	0.176	0.6	6.751	20.4	27.151
0.175	0.125	0.7	4.862	23.8	28.662
0.125	0.075	0.8	3.105	27.2	30.305
0.076	0.024	0.9	1.478	30.6	32.078

Table 8: Results of our optimization for the modified traffic network with cycling users (0830hrs, AM peak)

Proportion of Users			Delay/min		
Top path	Bottom path	Cycling	Driving	Cycling	Total
0.475	0.425	0.1	18.34	3.4	21.74
0.425	0.375	0.2	15.761	6.8	22.561
0.375	0.325	0.3	13.316	10.2	23.516
0.325	0.275	0.4	11.006	13.6	24.606
0.275	0.225	0.5	8.832	17	25.832
0.225	0.175	0.6	6.792	20.4	27.192
0.175	0.125	0.7	4.887	23.8	28.687
0.125	0.075	0.8	3.118	27.2	30.318
0.075	0.025	0.9	1.483	30.6	32.083

Table 9: Results of our optimization for the modified traffic network with cycling users (1330hrs, PM non-peak)

Indeed, as expected, cycling reduces the delay from driving. Even with our smallest proportion of 0.1 users cycling, the delay from driving is already reduced to 18.152 min and 18.34 min for AM peak and PM non-peak respectively.

Recall for our simple traffic network: (only driving)

At nash equilibrium (rational users):

1. AM peak, 0830hrs, Total delay = 26.61 min
2. PM non-peak, 1330hrs, Total delay = 26.97 min

At social optimum:

1. AM peak, 0830hrs, Total delay = 20.645 min
2. PM non-peak, 1330hrs, Total delay = 20.897 min

In fact at $c = 0.1$, the total delay of the traffic network is 21.552 min and 21.74 min, which is much lower than the nash equilibrium and only slightly higher than the social optimum. However as we increase the proportion of users cycling, the total delay of the traffic network gets much higher to values of 30. This is expected as the time taken to cycle to the end location is 34 minutes.

5 Conclusion

Looking back at the objectives of this study:

1. Develop a traffic network model that accounts for the proportion of users travelling a certain path, the average number of lanes, and the current traffic congestion (peak and non-peak hour timings);
2. Show how Braess' Paradox exists, where adding capacity to the traffic network might slow down traffic instead;
3. Examine our hypothesis on how the incorporation of using bicycles as an alternative mode of transport and building bike-lanes can help to ease traffic conditions.

Firstly, our traffic network model and latency function does account for those factors. However a limitation is that our latency function might be overestimating the delays as our values are higher than the travel timings shown in Google Maps at the respective time of the day. Furthermore the delays during peak-hour is shorter than the delays during the non-peak hours which is strictly speaking incorrect, hence there is a need to adjust our latency function.

A possible improvement would be to add in coefficient values similar to the Bureau of Public Road function (3). However that requires further studies and analysis and support from Land Transport of Authority to provide more specific data.

Secondly, we concluded that our scenario does not directly showcase Braess' Paradox and which supports how the paradox only appear in specific scenarios and not all cases as mentioned by this textbook [4].

Lastly, we proved our hypothesis that incorporation of using bicycles as an alternative mode of transport and building bike-lanes can help to ease traffic conditions.

6 References

[1] R. Wang and Z. T. Wang, "Research on improvement road impedance function based on vehicle navigation system," in *Advanced Materials and Computer Science II*, ser. Advanced Materials Research, vol. 659. Trans Tech Publications Ltd, 4 2013, pp. 45–48.

[2] R. T. Milam, M. Birnbaum, C. Ganson, S. Handy, and J. Walters, "Closing the induced vehicle travel gap between research and practice," *Transportation research record*, vol. 2653, no. 1, pp. 10–16, 2017.

[3] "Average speed on roads (peak hours)." [Online]. Available: https://www.lta.gov.sg/content/ltagov/en/who_we_are/statistics_and_publications/statistics.html

[4] D. Easley and J. Kleinberg, *Modeling Network Traffic using Game Theory*. Cambridge University Press, 2019, p. 231–233.

7 Appendix

7.1 Data Collection

Recall:

Nodes: Locations (Latitude, Longitude):

1. Node A: Marine Terrace
   (1.30556, 103.91534)
2. Node B: Singapore University of Technology and Design
   (1.34163, 103.9634)
3. Node C: Intersection between Xilin Avenue and Upper Changi Road
   (1.33241, 103.95494)
4. Node D: ECP exit 2B towards Xilin Avenue
   (1.32266, 103.97028)

a) Measuring minimum time to travel, t_i:

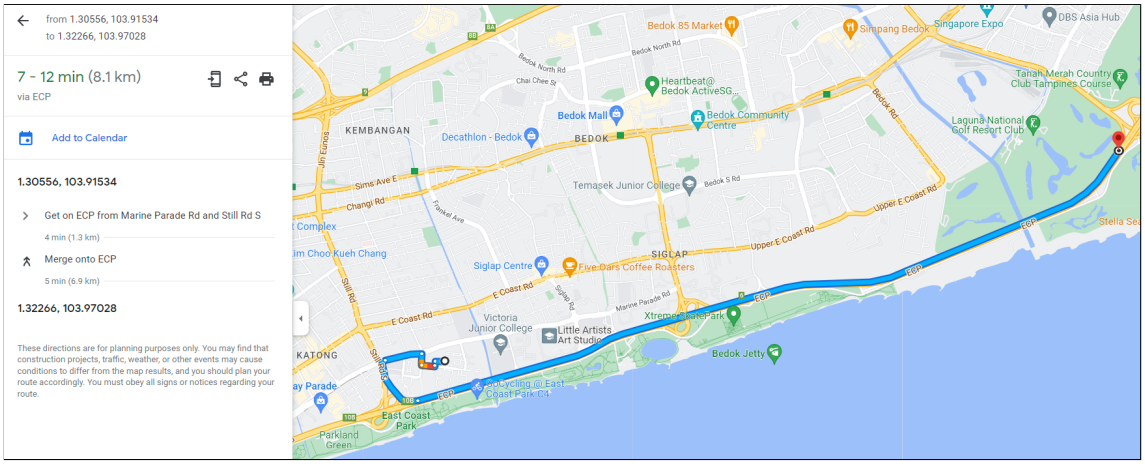

Figure 13: Driving from (A) to (D)

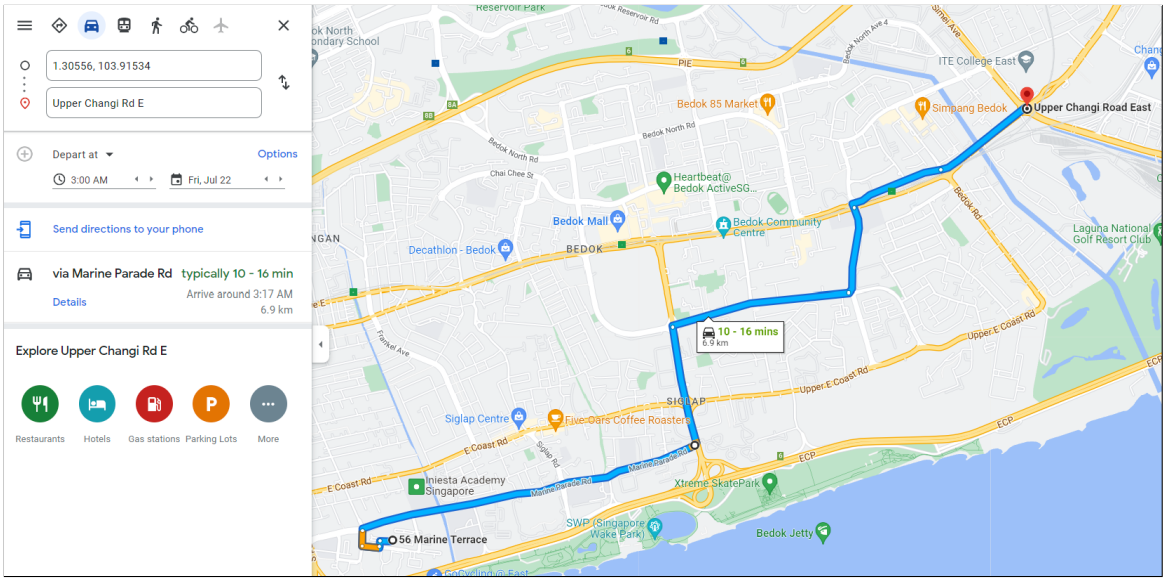

Figure 14: Driving from (A) to (C)

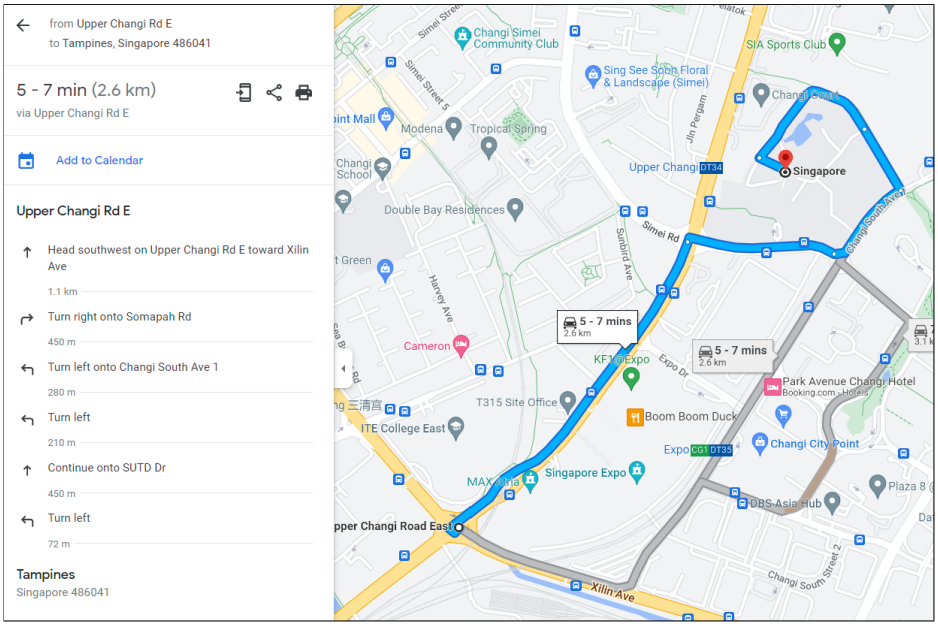

Figure 15: Driving from (C) to (B)

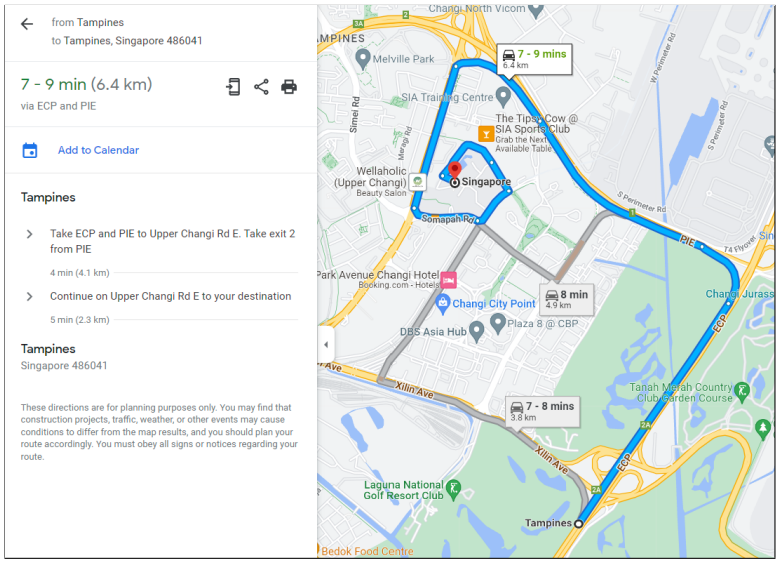

Figure 16: Driving from (D) to (B)

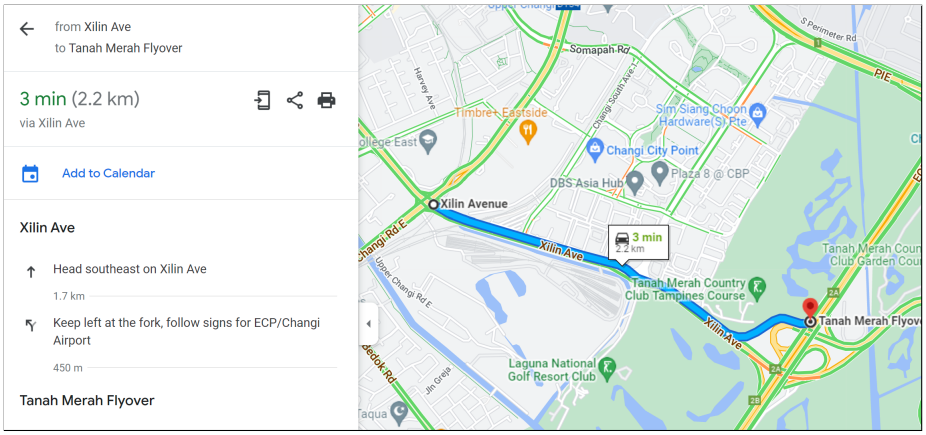

Figure 17: Driving from (C) to (D)

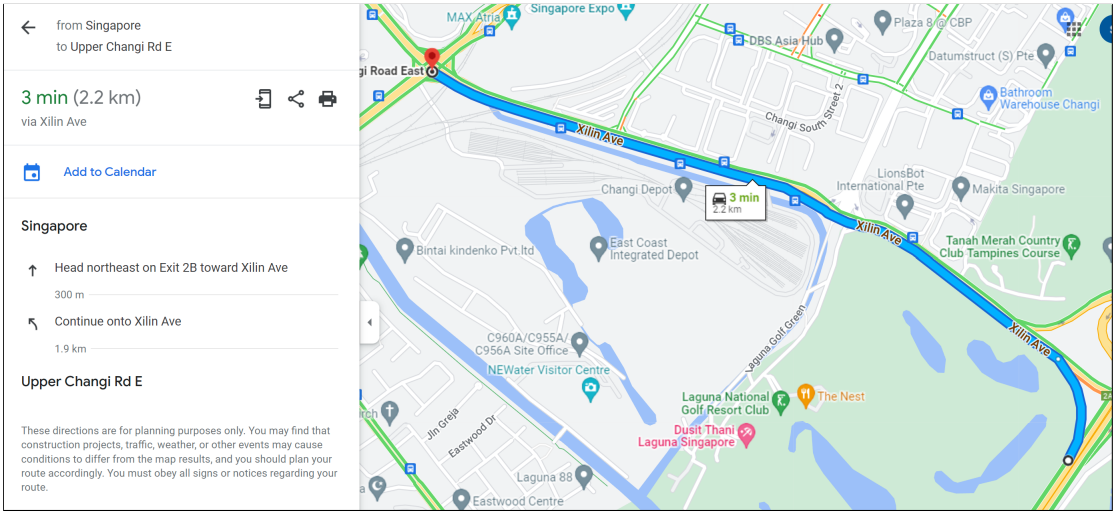

Figure 18: Driving from (D) to (C)

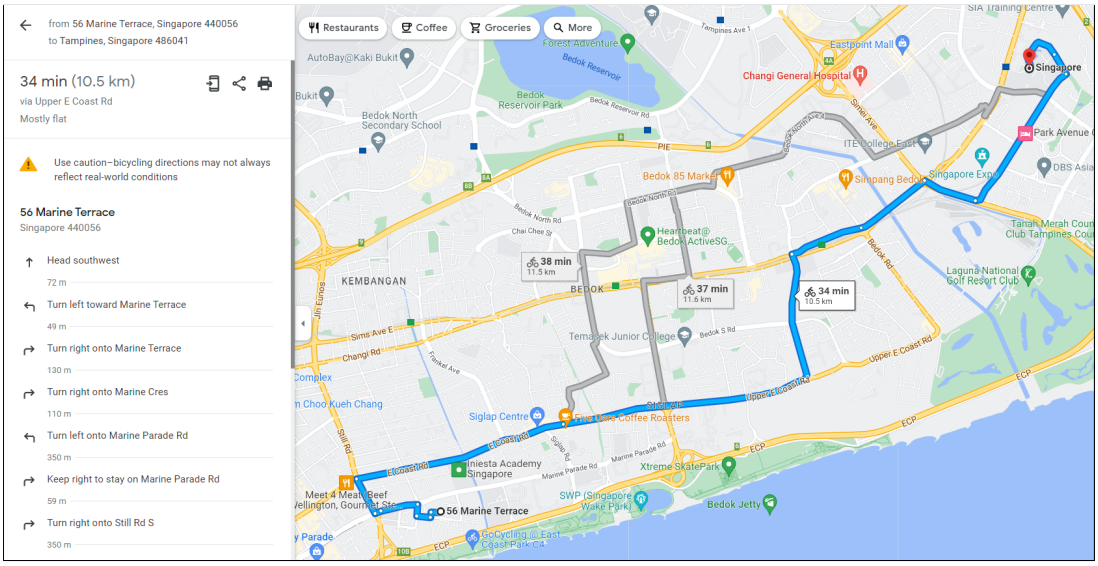

Figure 19: Cycling from (A) to (B)

b) Measuring number of lanes, γ_{ij}:

Just showing an example, following the directions, distance measure (left side) and live viewer (right side) to count the number of lanes:

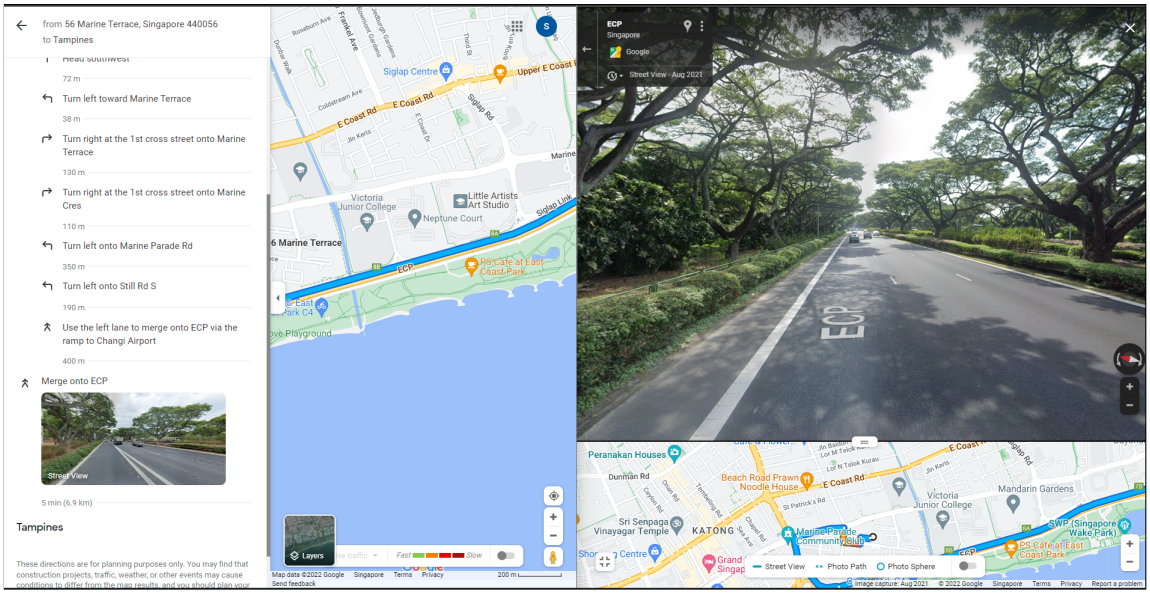

Figure 20: Example of counting the number of lanes

c) Measuring traffic states, θ_{ik}:

Just showing an example, following the distance measure and typical traffic states at the chosen timing from Google Maps:

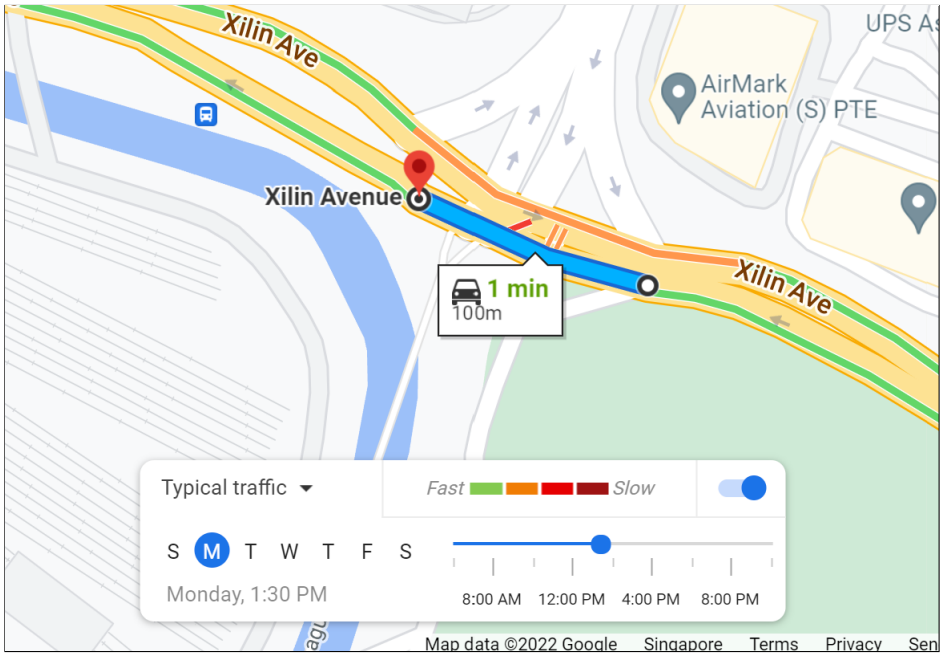

Figure 21: Example of counting the number of lanes