UNIVERSITY COLLEGE CORK

Traffic Flow Optimization in an Urban Environment

by

Kealan Gifford

Student Number: 115140157

A project report submitted in partial fulfillment for the degree of BSc Computer Science

in the

Department of Computer Science

Under Supervision of

Dr. Aisling O'Driscoll

April 2018

Declaration of Authorship

Declaration of Originality In signing this declaration, you are confirming, in writing, that the submitted work is entirely your own original work, except where clearly attributed otherwise, and that it has not been submitted partly or wholly for any other educational award. I hereby declare that:

- this is all my own work, unless clearly indicated otherwise, with full and proper accreditation;
- with respect to my own work: none of it has been submitted at any educational institution contributing in any way to an educational award;
- with respect to another's work: all text, diagrams, code, or ideas, whether verbatim, paraphrased or otherwise modified or adapted, have been duly attributed to the source in a scholarly manner, whether from books, papers, lecture notes or any other student's work, whether published or unpublished, electronically or in print.

Signed: ____

Date: ____

UNIVERSITY COLLEGE CORK

Abstract

by Kealan Gifford

As cities grow in size and population it is becoming increasingly important to employ intelligent traffic management practices. According to the 2017 INRIX Global Traffic report (based on 2016 data) Cork is third most congested city in Ireland with drivers spending on average 25 hours stuck in traffic every year, wasting fuel, increasing emissions and impacting on drivers transport experience.

This project report investigates the impact that different traffic light control algorithms (static, actuated and adaptive) have on the flow of traffic in an urban environment. A busy junction within Cork City has been examined via simulation utilizing SUMO (Simulation of Urban Mobility), a microscopic traffic simulator to model the environment, implement the algorithms and carry out the analysis. Several performance measures such as average vehicle journey time and average vehicle waiting time are considered using the same flows of traffic over a given time period.

Acknowledgements

I would like to thank Dr. Aisling O'Driscoll for her continued guidance and support for every step along the way. I would also like to thank my family and friends, without your continued support and motivation throughout the year. I am forever grateful to you all.

Contents

List of Figures

Abbreviations

ATLC	**A**adaptive **T**raffic **L**ight **C**ontrol
TLC	**T**raffic **L**ight **C**ontrol
NS	**N**orth and **S**outh
EW	**E**ast and **W**est
CFP	**C**yclic **F**low **P**rofile

Dedicated to my parents. Without your sacrifices I would never have had the opportunities I have today.

Chapter 1

Introduction

1.1 Background

As urban areas grow in size and population there is a growing demand for intelligent transportation management practices to be implemented in these areas. As cities are hubs for employment, as they grow so does the number of commuters flowing into and out of them which causes an increase in the amount of road users. As seen in figure 1.1, according to the 2017 INRIX Global Traffic report (based on 2016 data) Cork is third most congested city in Ireland with drivers spending on average 25 hours stuck in traffic every year, wasting fuel, increasing emissions and impacting on drivers transport experience.

1.2 This Project

The purpose of this report is to investigate the impact traffic light control algorithms have on the flow of traffic in an urban environment. Several traffic light control algorithms were examined via simulation of Dennehy's Cross, a busy junction located in Cork City. The traffic light control algorithms investigated include: static, actuated and adaptive.

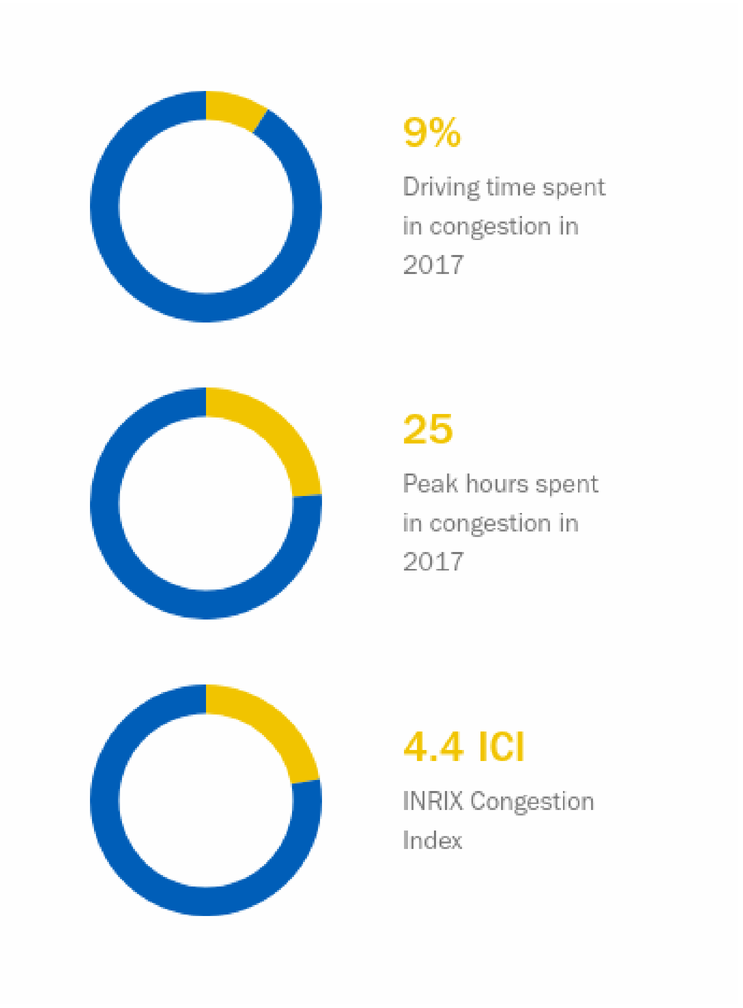


FIGURE 1.1: Cork Traffic Data, [1]

Dennehys Cross was modelled using the micro-simulation tool Simulation of Urban Mobility (SUMO). Using this tool to model the environment, the traffic light control algorithms were implemented and tested and the required traffic data generated for use in performance analysis and comparison. Several performance measures are investigated such as average vehicle journey times and average vehicle waiting times.

Chapter 2

Analysis

2.1 Objectives

The main objective of this project was to determine how the three approaches to traffic light control perform under the same traffic flow conditions, with the aim being to reduce average vehicle waiting times at this junction and the average vehicle journey times through this junction. The project aims to compare the different algorithms and be able to answer which has the greatest positive effect on traffic flow in relation to these objectives.

2.2 Literature Review

As the location of the urban area this project focuses on is located in Ireland, a literature review of the existing traffic control approaches currently in use in Irish cities was carried out. In Dublin the Sydney Coordinated Adaptive Traffic System (SCATS) originating from Australia, is in place as the traffic control system[2]. In Cork, the Split Cycle Offset Technique (SCOOT) originating from the United Kingdom, is used to control the traffic signals in urban areas [3]. These proprietary systems are two of the most widely deployed Adaptive Traffic Control System in urban environments in the world[4].

They use different approaches to evaluate traffic flows and calculate the cycle times yet they are both alike in the sense that they use hardware to detect traffic flows and both are adaptive making use of splits, cycles and offsets that update themselves in real-time. Furthermore an adaptive traffic light control algorithm (Adaptive TLC) proposed in a research paper originating in Dublin Institute of Technology is reviewed and is the main adaptable traffic control algorithm implemented during the process of this report.

2.3 SCOOT

SCOOT is a centralized adaptive traffic light control system that was originally developed in the UK during the 1980s by the Transport Research Laboratory for the Department of Transportation[5]. SCOOT is considered to be online due to its nature of predictions of delay and stops being recalculated every few seconds using the current measurements of the traffic flow. The main operation of SCOOT is to predict what changing the current signal timings would have on the flow of traffic [5].

2.3.1 Vehicle Detection

The current implementation of SCOOT makes use of induction loop sensors buried in the roads to sense vehicles driving over it. The detector doesnt need to be an induction loop so long as it can provide the required information about vehicles present on the road. The detectors should be placed as far up the road away from the junction they are in use for with the ideal location for them being just downstream of an adjacent junction so long as it can provide data on the traffic flow for that road[5].

2.3.2 Traffic Model

SCOOT continuously measures the traffic demand on all roads in its coordinated network and optimizes the signal times in accordance with the traffic demand detected[4]. SCOOT uses data that it collects such as green light phase durations and vehicle flows, and preset data for the area under control such as sensor locations to predict traffic queues, delay and stops[5]. An important feature of SCOOT is the use of cyclic flow profiles (CFP) which are built using the data from the vehicle detectors on vehicle flow and road occupancy. A profile is maintained for each approach to the signalled junction. Each profile is a histogram that is a record of how the traffic flow rate varied throughout a cycle. They are referred to as being cyclic because the patterns of the histogram tend to be repeated in the subsequent CFPs and new data is entered into each element once per cycle in a cyclic sequence[5].

Where CFPs are measured, SCOOT makes a prediction of the queue length of the vehicles downstream taking into account the time it takes a vehicle to reach the downstream stopline travelling at cruise speed. The cruise time is used to predict when the flow of vehicles are likely to arrive at the stop line, or in the case of a red light the back of the queue already there[5].

The key ideas behind SCOOT is the real-time measurement of these CFPs, using them to update the descriptions of queues in real time and to optimize the setting of the junction lights incrementally[6]. Timings for the splits, offsets and cycle lengths are each optimized separately[4].

2.3.3 Advantages

Recent developments in SCOOT allow the concept of gating which is a procedure that allows higher priority given to buses approaching an intersection by allowing the junction to skip a stage if it would benefit the flow of the junction[2]. SCOOT is an online model and it makes its adjustments in real-time instead of basing its adjustments on old traffic flow data from previous cycles[5]. There are no large or sudden changes in the signal timings, instead they are updated incrementally to prevent these large fluctuations in timings[5].

2.4 SCATS

SCATS is a two-level hierarchical adaptive traffic light control system, it was developed in the 1980s by the Roads Traffic Authority, a former Australian government agency[4]. Where SCOOT attempts to optimize, SCATS acts as a heuristic feedback system that adjusts signal timings based on changes in traffic flow in previous cycles. SCATS makes the assumption that a higher cycle time indicates that the junction is at higher capacity[4]. It aims to maximize the amount of effectively used time by vehicles crossing the junction compared to the total available green light time[2].

2.4.1 Vehicle Detection

Like SCOOT, SCATS also makes use of vehicle detectors however, instead of them being located as far upstream from the junction they are located in each lane immediately in advance of the stop line[4]. This is to allow it to adjust the signal timings of the junction in response to variations in traffic demand and the systems capacity[4].

2.4.2 Traffic Model

SCATS strategy makes use of splits proportional to approach demand and longer offsets where increased traffic volumes (that are flowing slowly) are detected. SCATS makes adjustments of the various signal timings separately, similarly to SCOOT[4]. SCATS collects the traffic flow data from the detectors and calculates the Degrees of Saturation (DS) and Link Flows (LF). These are then used to calculate cycle lengths, phase splits and offsets. This is carried out once per cycle as soon as new values for the DS and LF is obtained[4].

2.4.3 Advantages

Some of the advantages of SCATS in urban area is decreases in vehicle travel times, the level of accidents occurring and overall fuel consumption[2].

Chapter 3

Design

3.1 Simulation of Urban Environment (SUMO)

This project made use of the Simulation of Urban Mobility (SUMO) tool to simulate the road traffic environment necessary to carry out the investigation of the traffic light control algorithms. SUMO is an open source simulation package that is mainly developed by the Institute of Transportation Systems at the the German Aerospace Center.

After initial setup of the SUMO tool, an Open Source Map (OSM) was used by SUMO to build the network. One of the tools in SUMO is a Python script that is capable of importing the necessary data from an OSM map called OSMWebWizard.py. The script opens up a browser window that connects to the OSM map using an API and allows you to select a region of the map to import into SUMO.

The junction at Dennehys Cross was selected and the tool imported the data and began generating the road traffic network using the NETCONVERT tool. NETCONVERT takes as input a definition of a road network (in this case, an OSM map) and outputs a generated SUMO road network description in a SUMO-specific XML file. A list of the files that can be generated using the web wizard are shown in the figure below.

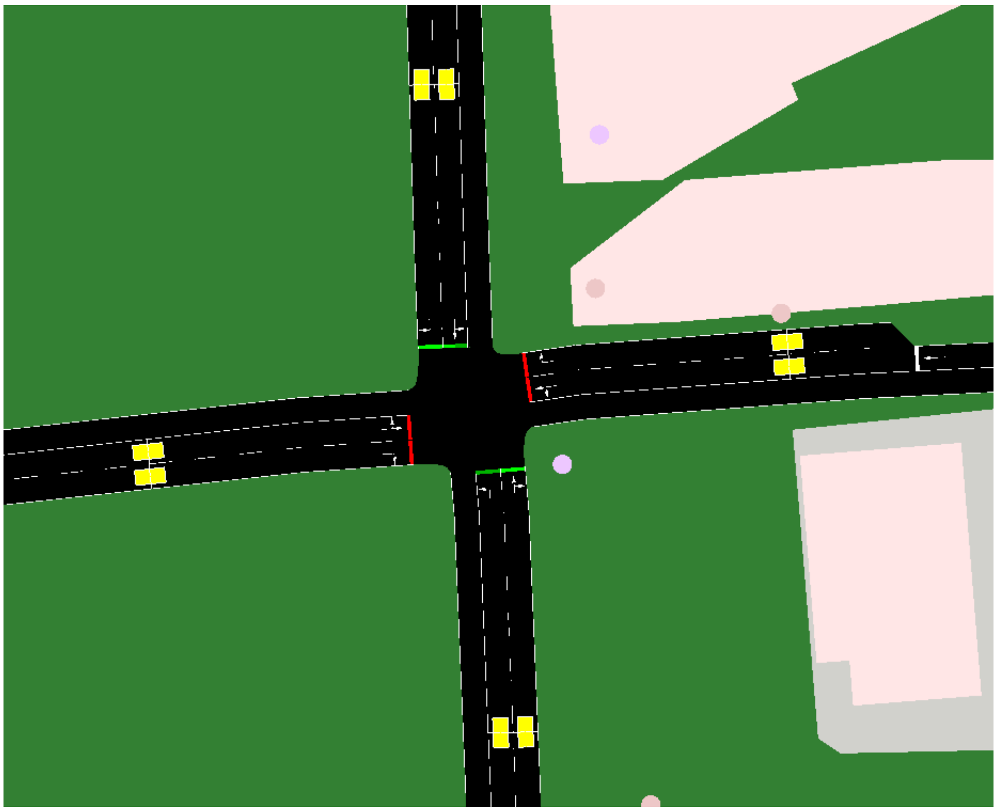


FIGURE 3.1: SUMO GUI Showing Simulated Dennehy's Cross

The osm.net.xml file is SUMO specific and contains the road network file. It describes all the road traffic elements of the map, the roads and intersections the simulated vehicles drive along/through. SUMO is a microsimulation tool but looking at it macroscopically, the network it produces is a description of a directed graph. The nodes, which are referred to as junctions in this report, represent the intersections that vehicles travel through, the edges describe the roads/streets. Although the graph is directed, the edges are unidirectional.

The osm.net.xml file contains information about:

- Every edge (street) a number of lanes including position, shape and speed limit of each lane,
- Traffic light logics which have the ids associated with the junction they are controlling,
- Nodes (junctions) and the right of way regulation between the edges involved with this junction,

- and connections between the lanes at the junctions.

3.2 NETEDIT

NETEDIT is a graphical user interface allowing the editing of networks. As the adaptive traffic light control algorithm requires vehicle detection sensors in order to function, it is possible to use NETEDIT to generate these sensors as induction loops. One sensor was added to the map for every lane that approaches the junction.

When running a simulation, it executes as a series of time steps. This can be set to any number when generating the network and what that number represents is an amount of time. For example, setting the time step to 0.1 would mean that every time step in the simulation represents the passing of 10ms in real time. An important setting when generating this network was to set the time step equal to 1 second. The reason for this was to make the calculation of results simpler where it is possible to produce XML files holding the data from the simulation every time step which meant that every entry in the file would represent the equivalent of a real world second. Another major reason for this was the fact that the adaptive traffic light control requires the calculation of time in seconds in its execution and compares it with the simulations time steps.

Although the time step represents 1 second, the simulation can execute these time steps incredibly fast, meaning that a simulation that is to run for an entire hour will completely execute in a matter of seconds. For the remainder of this report, a simulation second will be referred to as a time step.

3.3 Traffic Control Interface (TraCI)

The Traffic Control Interface (TraCI) is a TCP client/server architecture allowing access to a running SUMO simulation. SUMO acts as a server that can allow an optionally variable number of clients to connect to it, the default being one client. It was unnecessary in the context of this report to have multiple clients so only one client was used. TraCI allows the access to values of the objects in the simulation and allows the manipulation of these in an online/live manner.

TraCI can be used for value retrieval, state changing, subscriptions and access generic parameters of simulated objects. TraCI allows interfaces for different programming languages, and the one used in this instance was Python.

3.3.1 Value Retrieval

The values necessary for retrieval via TraCI was those of the induction loop sensors on the edges. These were made use of by the adaptive traffic light control algorithm when operating.

3.3.2 State Changing

As this report investigates the effect of the traffic light control algorithm, it was only necessary to change the state of the traffic lights.

3.4 Traffic Light Control

Python scripts were written to interact with SUMO via TraCI. A Python script *traci_script.py* was responsible for launching SUMO via TraCI. It consisted of three methods:

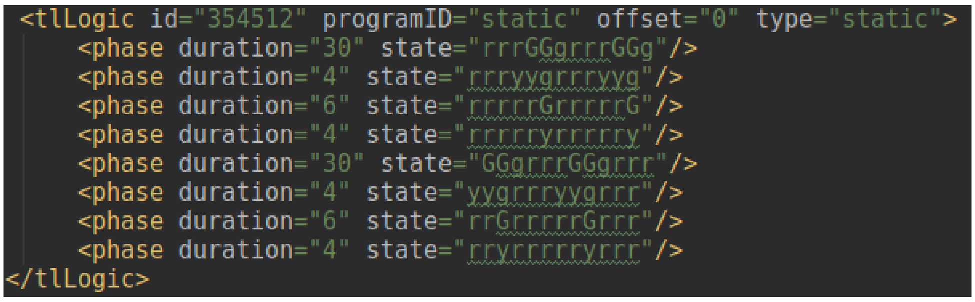
1. *run_static*
2. *run_actuated*
3. *run_dynamic*

which run the simulation using the the static, actuated and adaptive traffic light control methods respectively.

3.5 Traffic Light Controllers

3.5.1 Static

The static traffic light control is included in the SUMO and is the default traffic light control in generated networks. The traffic lights in the network are generated with a fixed cycle with a default cycle time of 90 seconds. All green phases are followed by a yellow phase, the length of the yellow phase is computed using the maximum speed a vehicle is allowed to travel at on the incoming roads. If a green phase allows for partially conflicting flows of traffic, e.g. vehicles travelling straight and vehicles turning left from the opposing direction, it is succeeded by another green phase th full priority to the conflicted streams (i.e. the left turning phase gets its own main green phase). The default duration of this conflicted priority phase is 6 seconds.

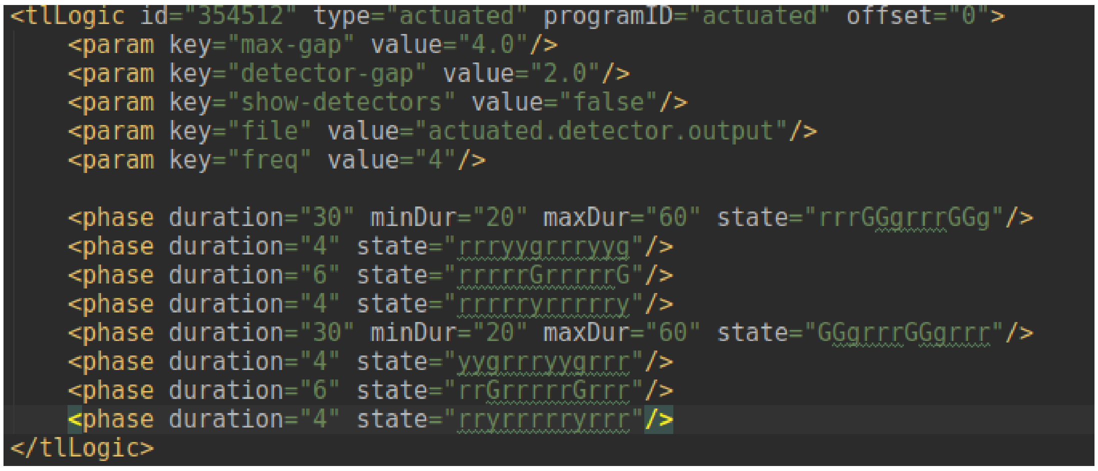


```
 <tlLogic id="354512" programID="static" offset="0" type="static">
     <phase duration="30" state="rrrGGgrrrGGg"/>
     <phase duration="4" state="rrryygrrryyg"/>
     <phase duration="6" state="rrrrrGrrrrrG"/>
     <phase duration="4" state="rrrrryrrrrry"/>
     <phase duration="30" state="GGgrrrGGgrrr"/>
     <phase duration="4" state="yygrrryygrrr"/>
     <phase duration="6" state="rrGrrrrrGrrr"/>
     <phase duration="4" state="rryrrrrryrrr"/>
</tlLogic>
```

FIGURE 3.2: Description for Actuated Traffic Light Logic Control

3.5.2 Actuated

The actuated traffic light control is also supported in SUMO however, as it is not the default, they need to be generated. When generated they have the same cycle plan as the static (initially) method does however the main green phases are allowed a variable length from a minimum to a maximum duration. The default for these is 5 second minimum and a 50 second maximum. These minimum and maximum durations were altered to be (initially) 20 seconds and 60 seconds respectively. The actuated control method is common in Germany and it works by extending the main green phase when a flow of traffic is detected with little distance between one vehicle and the next.



```
<tlLogic id="354512" type="actuated" programID="actuated" offset="0">
    <param key="max-gap" value="4.0"/>
    <param key="detector-gap" value="2.0"/>
    <param key="show-detectors" value="false"/>
    <param key="file" value="actuated.detector.output"/>
    <param key="freq" value="4"/>

    <phase duration="30" minDur="20" maxDur="60" state="rrrGGgrrrGGg"/>
    <phase duration="4" state="rrryygrrryyg"/>
    <phase duration="6" state="rrrrrGrrrrrG"/>
    <phase duration="4" state="rrrrryrrrrry"/>
    <phase duration="30" minDur="20" maxDur="60" state="GGgrrrGGgrrr"/>
    <phase duration="4" state="yygrrryygrrr"/>
    <phase duration="6" state="rrGrrrrrGrrr"/>
    <phase duration="4" state="rryrrrrryrrr"/>
</tlLogic>
```

Figure 3.3: Description for the Actuated Traffic Logic Control

SUMO supports actuation based on time gaps between vehicles or time loss for vehicles. The gap-based actuated traffic light control was the one chosen for this report. It holds the notion of a max-gap which is the maximum time allowed between a vehicle travelling over a (automatically generated) detector and the vehicle that is following it. So if the max-gap value is 3 seconds, then if the time that passes from the first detected vehicle and the next is greater than 3 seconds, the phase will change to the next phase. The green phase will run for the default minimum amount of time and then begins detecting the time gaps between vehicles as described above.

3.5.3 Adaptive

The adaptive traffic light control is not included in SUMO and it was necessary to write a script with an algorithm that was capable of interacting with the simulation according to the method described in the academic paper by Rademacher (2017)[2]. The adaptive traffic light control (ATLC) main philosophy is to be able to create an intelligent junction that can alter its phase durations based on the number of approaching vehicles. It was inspired by a combination of the existing static traffic light method in SUMO and other existing traffic light control methods in literature[2].

The fewer number of possible phases for a junction of traffic lights causes the cycle time to be shorter which in turn makes the traffic light control more efficient [7]. Based on the argument that the shorter the queues at a junction are, the shorter the cycle time is while still letting enough cars cross the junction [7]. Using this theory, the ATLC consists only of two green light phases. One for all flows travelling North/South (NS) and one for all flows travelling East/West (EW).

After a green light phase, an orange phase must follow for safety purposes with a duration of 5 seconds. The green light phases can range between a minimum of 20 seconds and a maximum of 80 seconds. For the remainder of this report, the North/South and East/West directions will be referred to as NS and EW respectively.

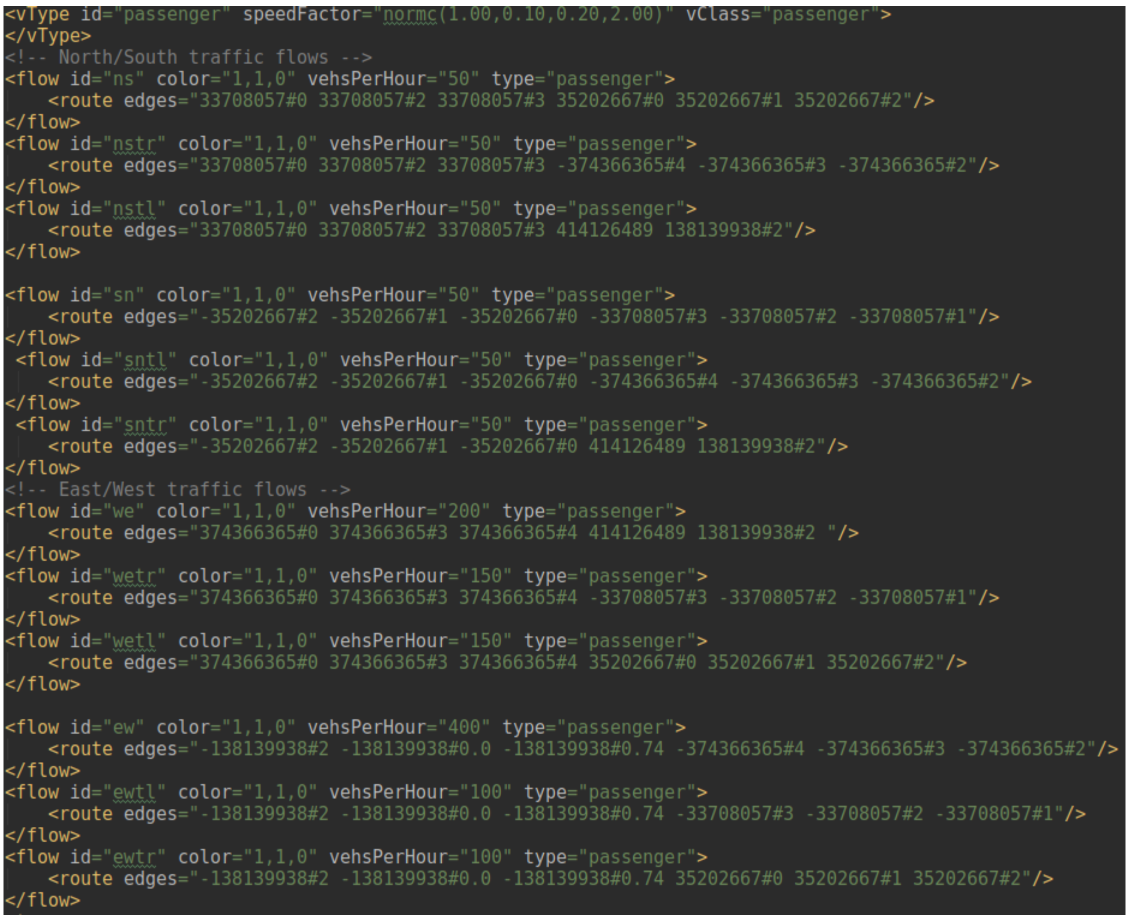
3.6 Vehicle Simulation

3.6.1 Random Trips

SUMO comes with a ”randomTrips.py” script that takes in the ”osm.net.xml” network file, and produces a vehicle trips file for that network where the routes and trips, start times and end times of vehicles being inserted into the network are completely random. This is not typical of real traffic flows as there are patterns to be observed in real traffic flows. Especially in urban areas with signalized intersections. For this reason this method of generating vehicles for the simulation was discarded.

3.6.2 Vehicle Flows

Cork City council was approached for access to vehicle flow counts however, there was no response. An inquiry with Atkins Ireland if they would be able to provide the information necessary was also conducted and although there was a timely response, they did not have the exact information needed in regards to the vehicle counts. In lieu of this information being readily available, vehicle flows were simulated in an attempt to mimic real traffic flows. SUMO allows for descriptions of vehicle flows.



```
<vType id="passenger" speedFactor="normc(1.00,0.10,0.20,2.00)" vClass="passenger">
</vType>
<!-- North/South traffic flows -->
<flow id="ns" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="33708057#0 33708057#2 33708057#3 35202667#0 35202667#1 35202667#2"/>
</flow>
<flow id="nstr" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="33708057#0 33708057#2 33708057#3 -374366365#4 -374366365#3 -374366365#2"/>
</flow>
<flow id="nstl" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="33708057#0 33708057#2 33708057#3 414126489 138139938#2"/>
</flow>

<flow id="sn" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="-35202667#2 -35202667#1 -35202667#0 -33708057#3 -33708057#2 -33708057#1"/>
</flow>
 <flow id="sntl" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="-35202667#2 -35202667#1 -35202667#0 -374366365#4 -374366365#3 -374366365#2"/>
</flow>
 <flow id="sntr" color="1,1,0" vehsPerHour="50" type="passenger">
    <route edges="-35202667#2 -35202667#1 -35202667#0 414126489 138139938#2"/>
</flow>
<!-- East/West traffic flows -->
<flow id="we" color="1,1,0" vehsPerHour="200" type="passenger">
    <route edges="374366365#0 374366365#3 374366365#4 414126489 138139938#2 "/>
</flow>
<flow id="wetr" color="1,1,0" vehsPerHour="150" type="passenger">
    <route edges="374366365#0 374366365#3 374366365#4 -33708057#3 -33708057#2 -33708057#1"/>
</flow>
<flow id="wetl" color="1,1,0" vehsPerHour="150" type="passenger">
    <route edges="374366365#0 374366365#3 374366365#4 35202667#0 35202667#1 35202667#2"/>
</flow>

<flow id="ew" color="1,1,0" vehsPerHour="400" type="passenger">
    <route edges="-138139938#2 -138139938#0.0 -138139938#0.74 -374366365#4 -374366365#3 -374366365#2"/>
</flow>
<flow id="ewtl" color="1,1,0" vehsPerHour="100" type="passenger">
    <route edges="-138139938#2 -138139938#0.0 -138139938#0.74 -33708057#3 -33708057#2 -33708057#1"/>
</flow>
<flow id="ewtr" color="1,1,0" vehsPerHour="100" type="passenger">
    <route edges="-138139938#2 -138139938#0.0 -138139938#0.74 35202667#0 35202667#1 35202667#2"/>
</flow>
```

Figure 3.4: Vehicle Flow Descriptions

The image above describes a flow of traffic from North to South with a list of the edges vehicles from this flow will travel along. The parameter "vehsPerHour" indicates that this flow will produce 300 vehicles of the course of a simulated hour and the type will be a passenger vehicle (car). It will insert these vehicles into the simulation evenly over the course of the hour. With 300 vehicles per hour, it will insert a new vehicle every 12 seconds so long as there is room for the vehicle on at the starting edge, otherwise it will wait until it is safe to insert it.

The idea was to produce several flows of traffic for each approaching edge to mimic a green light occurring at a junction outside the simulation which would produce a wave of vehicles entering this simulation via the outermost edges. Another option instead of defining flows with vehicles per hour, it is possible to replace the ”vehsPerHour” parameter with probability. This takes floating point number value between 0 and 1. This represents the probability that a vehicle will be inserted into the simulation at each time step. However this was not used for the same reason as generating random vehicle trips using the ”randomTrips.py” script included in SUMO.

Chapter 4

Implementation

4.1 Adaptive Traffic Light Control (ATLC)

As the adaptive traffic light control (ATLC) algorithm is not included in SUMO, it was necessary to implement it in order to be able to carry out comparisons between it and the static and actuated methods. It was implemented using Python 3.6.3 and the PyCharm Professional Edition. The main objective of the ATLC is to give green light priority to the roads with higher vehicle flows. This section describes the method of operation of the adaptive traffic light control algorithm described by Rademacher (2017) and which was implemented in the Python programming language as part of this project report.

A Python class was written to hold the logic for this algorithm and to make applying it to any junction simpler in the future instead of having it hard coded to a single junction and is located in the atlc.py file. Initially an Atlc object must be created. It requires the SUMO network ID of the junction it is to control, and the file location holding the description of the induction loop detectors for that junction.

Once the instance is created, it is necessary to inform the ATLC controller the number of induction loops associated with each approaching edge by assigning them to it. To do so one could call the ATLC assign_junction_detectors method with, for example, the following parameters (2, 2, 2, 2) which tells the ATLC that two induction loops are associated per approaching edge for the North, East, South, West edges respectively.

After associating the induction loops the simulation may begin by running TraCI. As described above, TraCI runs SUMO as a server allowing the ATLC program to interact with the simulation as a client.

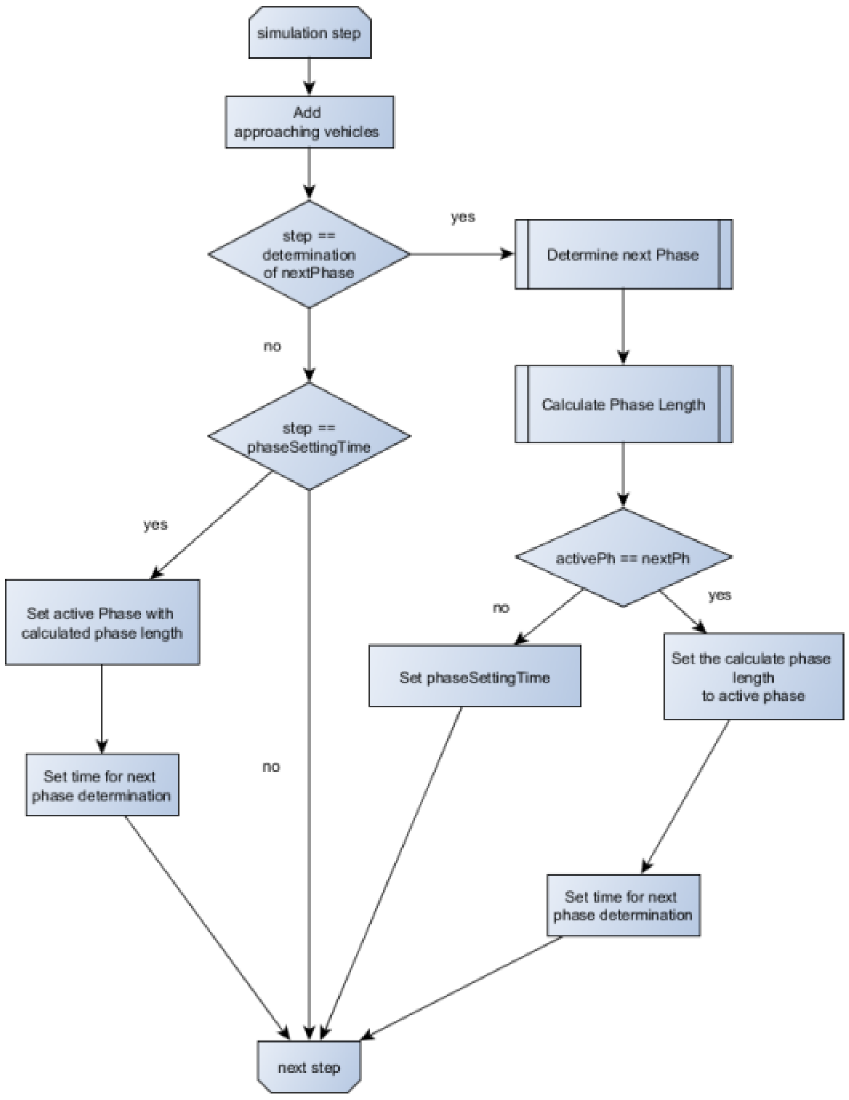

simulation step
Add approaching vehicles
step == determination of nextPhase
yes
Determine next Phase
no
Calculate Phase Length
step == phaseSettingTime
yes
activePh == nextPh
Set active Phase with calculated phase length
no
yes
Set phaseSettingTime
Set the calculate phase length to active phase
Set time for next phase determination
no
Set time for next phase determination
next step

FIGURE 4.1: Main Path of Adaptive Algorithm [2]

After initialization, the ATLC instance will keep information on:

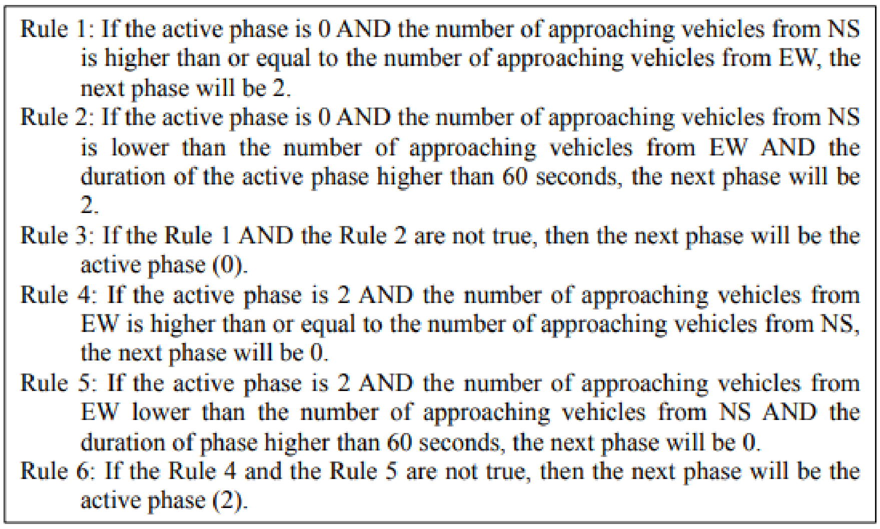
1. the current active phase,

2. the most recent green phase duration,
3. the time step when the next green phase is to be determined
4. and the time step in which the duration of the next phase has to be determined when the green phase is changing.

The ATLC algorithm does not calculate the next cycle and cycle length or even the current cycle and cycle length. It only calculates the duration that the next green phase will run for, or for how long the current green phase will be extended by. This information is used when the ATLC is at the beginning of the next determined phase to allow it to set the calculated duration.

The initial green phase duration is 30 seconds. This means the initial time step for the next phase determination is at time step 29.

At every time step in the simulation, the numbers of vehicles that pass over the inductive loops are summed and added to the total number of vehicles for each traffic flow direction (EW and NS) over the course of the current active phase. The the induction loops on the North and South approaching vehicle counts are summed together and the East and West vehicle counts are summed together.



Rule 1: If the active phase is 0 AND the number of approaching vehicles from NS is higher than or equal to the number of approaching vehicles from EW, the next phase will be 2.
Rule 2: If the active phase is 0 AND the number of approaching vehicles from NS is lower than the number of approaching vehicles from EW AND the duration of the active phase higher than 60 seconds, the next phase will be 2.
Rule 3: If the Rule 1 AND the Rule 2 are not true, then the next phase will be the active phase (0).
Rule 4: If the active phase is 2 AND the number of approaching vehicles from EW is higher than or equal to the number of approaching vehicles from NS, the next phase will be 0.
Rule 5: If the active phase is 2 AND the number of approaching vehicles from EW lower than the number of approaching vehicles from NS AND the duration of phase higher than 60 seconds, the next phase will be 0.
Rule 6: If the Rule 4 and the Rule 5 are not true, then the next phase will be the active phase (2).

FIGURE 4.2: Rules for Determining Which Phase Should be Next [2]

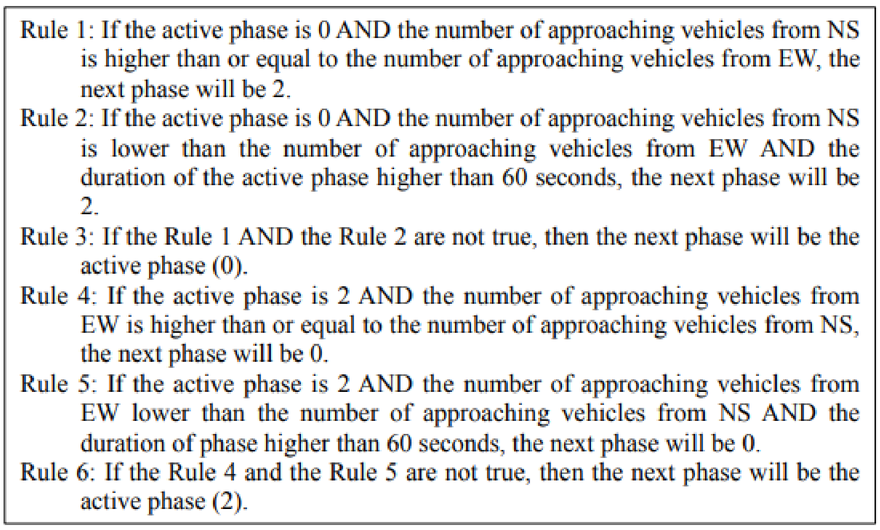
At every time step there is also a check comparing the time step value with the ATLC variable holding the time step in which the next phase has to be determined.

Initially this check will happen when the time step is equal to 29. If this check returns true then the next phase is decided upon. The ATLC uses information on:

1. the current active phase,
2. the total numbers of approaching vehicles for the NS and EW directions.

the total numbers of approaching vehicles for the NS and EW directions.

After this decision, the duration that the next phase will last for is calculated. If the phase is not going to change due to a detection of higher traffic demand along that phases traffic flow direction (EW or NS), its duration is extended by a certain amount. After this calculation it is necessary to update the value of the time step in which the next green phase must be determined. For example, the initial phase ran for 30 seconds and the next green phase was determined 1 second before it ended at time step 29. If it was decided that the phase was to be prolonged by 10 seconds then the time step where the next green phase is to be determined will be time step 39, 1 second before the end of the extended phase. If the decision is that a phase change must occur, the phase of the traffic light is switched to its respective intermediate orange phase which will run for a total of 5 seconds (time steps).



Rule 1: If the active phase is 0 AND the number of approaching vehicles from NS is higher than or equal to the number of approaching vehicles from EW, the next phase will be 2.
Rule 2: If the active phase is 0 AND the number of approaching vehicles from NS is lower than the number of approaching vehicles from EW AND the duration of the active phase higher than 60 seconds, the next phase will be 2.
Rule 3: If the Rule 1 AND the Rule 2 are not true, then the next phase will be the active phase (0).
Rule 4: If the active phase is 2 AND the number of approaching vehicles from EW is higher than or equal to the number of approaching vehicles from NS, the next phase will be 0.
Rule 5: If the active phase is 2 AND the number of approaching vehicles from EW lower than the number of approaching vehicles from NS AND the duration of phase higher than 60 seconds, the next phase will be 0.
Rule 6: If the Rule 4 and the Rule 5 are not true, then the next phase will be the active phase (2).

FIGURE 4.3: Rules for Determining the Next Phase's Duration [2]

Whether the phase is being prolonged or if the phase is going to change, the value of the total summed vehicles detected for the decided upon phases direction (EW or NS) is reset to a count of zero after the phase change. So for the example in the previous paragraph regarding the extension of the initial phase, that initial phase directions vehicle count will be reset to zero after the decision to prolong it. This is to ensure that the other phase direction (the phase currently in a red light state) will have a fair chance to be allowed to be chosen to be the next green phase at the next green phase determination. This reduces the number of unnecessary extension of green light phases.

Finally, at each time step when no phase is being determined, the time step is also compared to the variable holding the time step value for setting the green phase length. If the phase is going to change, then the phase which is being changed to will have a default duration of 20 seconds. If the phase is being extended, it will be extended by a fraction of the default duration.

4.2 Experiments

For all the test scenarios mentioned in this section, the simulation was run for 1800 seconds (time steps) which is 30 minutes.

4.2.1 Inputs

The inputs for each of the test cases include the following files:

- passenger.trips.xml: describes the routes of the journeys that the passenger vehicles (cars) will take in the simulation execution. This held the description of the flows of passenger vehicles.
- bus.trips.xml: describes the routes of the journeys that the buses will take in the simulation execution. This held the description of the flows of buses.

- e1.add.xml: describes the induction loops to be included in the network when running the simulation. It holds an entry for each induction loop for the network including their location.
- tl.add.xml: holds the initial traffic light logic for the adaptive traffic light control. It describes the amount of phases, their initial durations, how many traffic lights are controlled and which signals to display based on the phase that is active.
- tl.other.xml: contains similar information to the tl.ad.xml file except that it describes the static and actuated traffic light logic control.
- tl.states.xml: an additional file that is needed to ensure SUMO outputs the information about the state of the traffic lights at each timestep. Causes the tl_info.xml file to be outputted.
- edge.info.xml: an additional file required to ensure SUMO outputs the information about each edge during the running of a simulation.

The outputs produced by running a simulation include the following files:

- edge.output.xml: contains all the important data that SUMO outputs such as that about vehicles waiting on an edge at each time step.
- tl_states.xml: contains the data about what the state of the traffic lights were at each time step in the simulation.
- trip.infos.xml: holds the data describing each vehicles trip, e.g. the journey time the vehicle took from start to end.

4.3 Test Cases

High traffic flows for a certain direction = 1100 vehicles per hour.

Lower traffic flows for a direction direction = 300 vehicles per hour.

4.3.1 High traffic flows from one direction

Initially, an examination of the performance of the three traffic light control algorithms against high volumes of traffic flows approaching from the EW directions and a lesser flow from the NS direction. The reason for this is to create an imbalance in the number of vehicles flowing to force the ATLC and the actuated traffic control methods to perform their function.

4.3.2 Actuated traffic light control with varying max-gap values

Variations in the value of the max-gap time allowed between vehicles flowing through the junction were examined against the high traffic flows from the EW direction and a lesser flow from the NS direction. Values between 2 seconds up to 6 seconds were tested.

4.3.3 ATLC performance with varying induction loop locations on the approaching edges

Variations in the value of the induction loop placement on the lanes approaching the junction were examined against the high traffic flows from the EW direction and a lesser flow from the NS direction. Locations at 20 meters, 25 meters, 30 meters, and 35 meters were examined.

4.3.4 Vary the minimum green light duration for all three of the traffic light control algorithms

Vary the minimum green light duration for all three of the traffic light control algorithms Variations in the minimum value of the green phase durations for all three traffic light control algorithms at the junction were examined against the high traffic flows from the EW direction and a lesser flow from the NS direction. The green phase minimum durations were examined at 20, 25 and 30 seconds. This was conducted in parallel to the induction loop variance in order to maximize time efficiency when carrying out these tests.

4.4 Alteration of the Adaptive Traffic Light Control Algorithm

The adaptive TLC uses this idea of a default duration for green phase durations. When the phase is changed it will always run for at least this minimum default duration of, for example, 20 seconds. If it decides to extend the current green phase duration, it extends it by a fraction of the default duration. This default duration is also the maximum amount of time a phase can be extended by in one prolongation.

In this section, the adaptive TLC algorithm was altered by separating this idea of a default duration being used for the minimum running time and the maximum extension time. A phase will always run for a minimum default duration, however the idea of a maximum extension time is introduced. The effects of this on the adaptive TLCs performance is examined by varying the extension time from 20, 16, 12 and 8 seconds. The default green duration was chosen to be 20 seconds as the adaptive TLC performed the best in the experiments above and it was tested with varying the distance of the induction loops from 20, 25, 30 and 35 meters from the junction.

This test was repeated for with the maximum green phase duration being 60 seconds.

4.5 Final Test

After carrying out and analysing the results from the all of the test cases above, the best performing settings for each of the static, actuated and adaptive were used in the following scenario.

4.5.1 High traffic flows from all directions

An examination of the performance of the three traffic light control algorithms against high traffic flows from both the EW and and the NS approaches. Each algorithm was tested on the same scenario, even if one would expect that what was being altered would not have an effect on that algorithms performance. For example, changing the induction loop locations should not affect the performance of the static or actuated traffic control algorithms as they do not rely upon them to carry out their function.

4.6 Errors Encountered

4.6.1 Incorrect Edge Definition

Although the OSMWebWizard mentioned in the design chapter is a fantastic tool, it can sometimes make a mistake regarding the number of lanes held in the description of an edge, among other errors when building the network by reading in the information from an OSM map. This problem occurred in the generation of the Dennehys Cross network where it omitted the turning only lanes from the East and West approaching roads. Furthermore it was incorrect about how the different lanes connected to each other across the junction - e.g. which lanes held the traffic travelling straight and turning left were incorrect. This caused significant hindrance in the performance by all three traffic light control approaches, with the static TLC being the most detrimentally affected.

Although all the information regarding the contents of the network is held in XML files and they are human readable, it is not advised to alter these by hand - especially as the networks grow larger and more complex. Instead, it is a better idea to use the SUMO tool NETEDIT to alter road network files and so, NETEDIT was employed to debug these errors and was used to add the excluded lanes to and fix the connections across the Dennehys Cross junction.

4.6.2 Resetting Vehicle Counts

Another error encountered was during the writing of the code for the adaptive traffic control algorithm. When the next green phase is decided upon and its duration is set, it is necessary to reset the vehicle count for the direction (NS or EW) that green phase will be servicing. This is in order to allow a more reasonable decision at the next green phase determination[2] when comparing the two vehicle counts. However, a mistake was made in the code relating to this resetting of vehicle counts and instead of only resetting the one directions vehicle counts, both vehicle counts were reset to zero. This completely removed the fairness intended by the reset and negatively impacted the performance of the adaptive TLC. Once this error was corrected, the adaptive TLC was allowed to perform its intended function without impediment.

Chapter 5

Evaluation

5.1 Data Preparation

It was necessary to write scripts to parse the XML files that SUMO outputs after running a simulation. The parsers were developed using Python 3.6.3 and the ElementTree XML API. The files that were parsed include:

- edge.output.xml: contains all the important data that SUMO outputs such as that about vehicles waiting on an edge at each time step.
- tl_states.xml: contains the data about what the state of the traffic lights were at each time step in the simulation.
- trip.infos.xml: holds the data describing each vehicles trip, e.g. the journey time the vehicle took from start to end.

The parsers output CSV files by the same name (e.g. edge.output.csv). The data from these CSV files were then read and plotted on graphs (seen in the results section further in this report) using PlotLy[8], an open source data visualization service.

5.2 Performance Factors

Two factors were compared after analyzing the results of each traffic control algorithm.

1. Average vehicle waiting time at the junction: the amount of time a vehicle is considered to be waiting due to a red light signal or in congestion due to a red light signal.
2. Average vehicle journey time: the time taken by each vehicle to travel from its source to destination via this junction

The average vehicle waiting time was chosen as a performance measurement to be used in comparison due to junction traffic lights being one of the primary causes of causing vehicle to become stationary (red lights). The reasoning behind choosing vehicle journey times as a performance measurement was due to the assumption that as traffic congestion builds, so does the amount of time a vehicle takes to reach its destination. This is not due to the vehicle sitting stationary in traffic (waiting) as SUMO counts a vehicle to be ”stationary” if it is moving less than 0.1 meters per second. Instead it can be attributed to the vehicles having to drive much slower than the speed limit due to congestion. In contrast, a vehicle travelling along an edge with very little congestion can move at or just below the speed limit, and in some instances cross the junction at speed, allowing it to travel the overall distance in less time due to this decreased congestion.

5.3 Results

5.3.1 Minimum Green Time Variation and Induction Loop Sensor Location Variation

The minimum green phase duration and the induction loop location tests were carried out in parallel to maximize the efficient use of time and so will be evaluated together as it gives a clearer picture of how each variable performed in accordance with the other.

5.3.1.1 Static Performance

It was found that the static traffic light control (TLC) performed the worst with its best performance being a minimum green light duration of 30 seconds. This produced average vehicle waiting and journey times of 41.47 seconds and 130.89 seconds respectively.

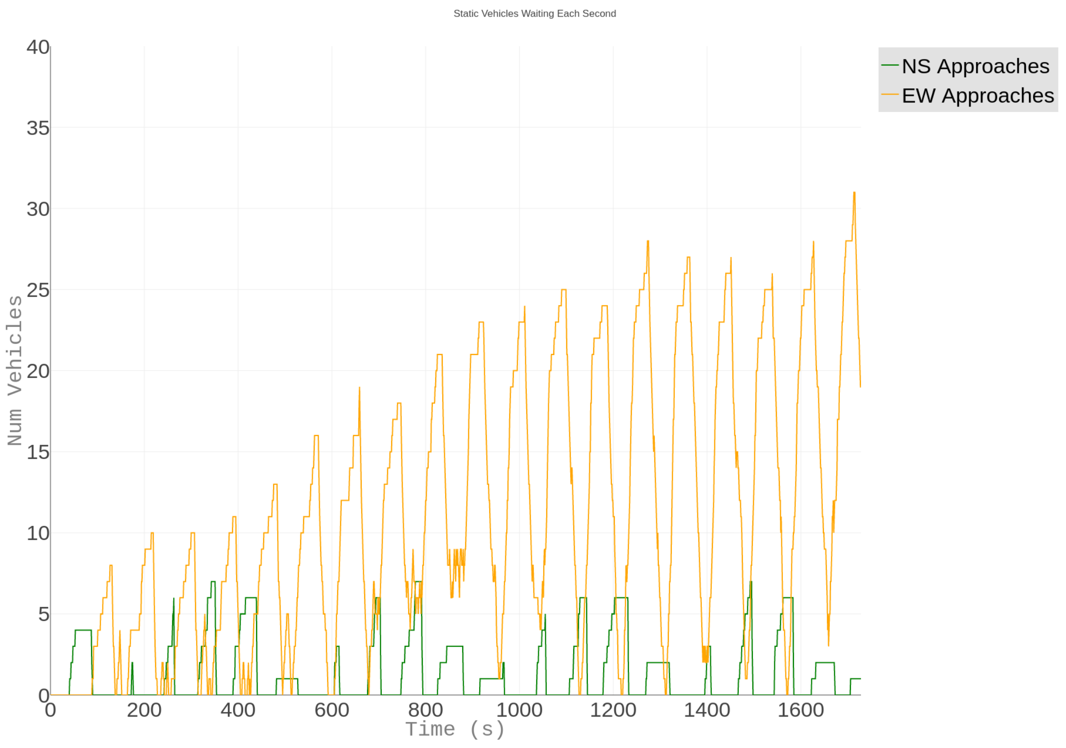


FIGURE 5.1: Number of Vehicles Waiting Over 30 Minute Simulation During Static TLC Test

As you can see in the graph, the performance of the static traffic light control deteriorates drastically over time. The peaks in the graph represent red light signals being displayed for that traffic flow direction. As the peaks grow over the duration of the simulation, it indicates that the queue of traffic build-up at one red light is unable to fully discharge across the junction when it is given a green phase. This causes the increase in traffic queues at each subsequent red light which experience the same problem. This shows that for constant high volumes of traffic that the static TLC's performance is very limited and congestion can occur in a matter of minutes. The graphed signal durations for the static TLC may be viewed in the appendix in Figure A.3.

5.3.1.2 Actuated

The actuated TLC performed moderately better than the static TLC. In contrast to the static TLCs best performance using a longer 30 second minimum green phase duration, the actuated TLC performed best with a smaller minimum green phase duration of 20 seconds. It produced average vehicle waiting and journey times of 16.41 seconds and 99.43 seconds respectively.

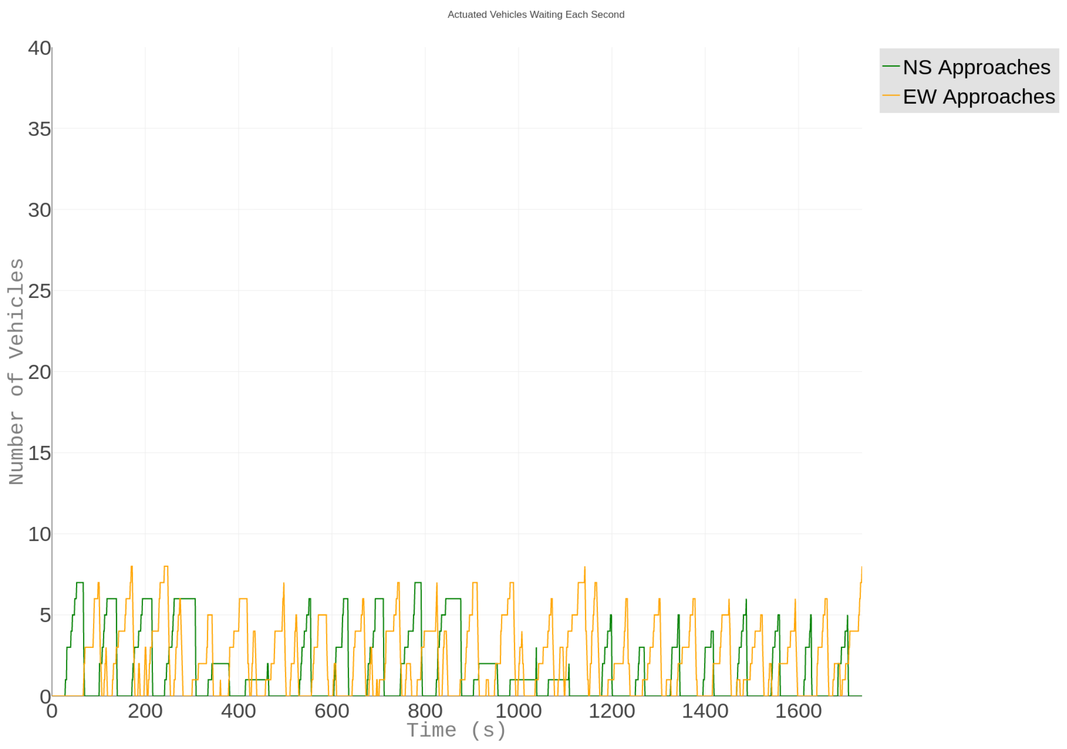


Figure 5.2: Number of Vehicles Waiting Over 30 Minute Simulation During Actuated TLC Test

As is visible in the graph, the performance of the actuated TLC is a huge improvement over the static TLC. It clearly is able to handle high volumes of traffic from the EW direction and low volumes of traffic from the NS very well in comparison. This is attributed to its ability to dynamically extend its green phase duration up to a maximum of 60 seconds, allowing higher priority given to the higher traffic demand of the EW direction. The graphed signal durations for the actuated TLC may be viewed in the appendix in Figure A.4.

5.3.1.3 Adaptive Performance

The adaptive TLC outperformed both of the other two TLCs. As the green phase duration changes was the only variable that would affect the static and actuated TLCs, it is only with the adaptive TLC that the variation of the induction loop held sway. The adaptive TLCs best performance was with a minimum green phase duration of 20 seconds and the induction loops located 30 meters from the junction stop lines. This differs from the findings of Rademacher (2017) only slightly in the finding that the induction loop location providing the best performance was located at 30 meters and not 25 meters as in theirs. The difference between the induction loops being located at 25 and 30 meters is extremely small and can be attributed to a quirk in the traffic flow being provided. Rademacher (2017) does not compare the effects of varying the minimum green phase duration. This report investigated the effects of these variations.

The adaptive control performed best with a configuration of 20 second minimum green phase duration and induction loops located 30 meters from the junction. The average vehicle waiting and journey times for this configuration were 8.47 seconds and 88.40 seconds respectively. As you can see in the figure regarding the amount of adaptive vehicles at each point in time, it was greatly successful in handling the high flow of traffic from the East/West (EW) directions. The peaks are much lower in comparison to the static and actuated peaks for the EW directions and do not vary by any great amount. Interestingly, notice the peaks for the EW direction and the NS direction are similar in height. The adaptive TLC is attempting to equalize these peaks and it attempts to achieve this goal by the vehicle being count reset to zero when changing or prolonging the green phase. This is where the idea mentioned in section 4.1 of giving a "fair chance" in the phase decision based on vehicle counts is at work. The graphed signal durations for the adaptive TLC may be viewed in the appendix in Figure A.5, notice the

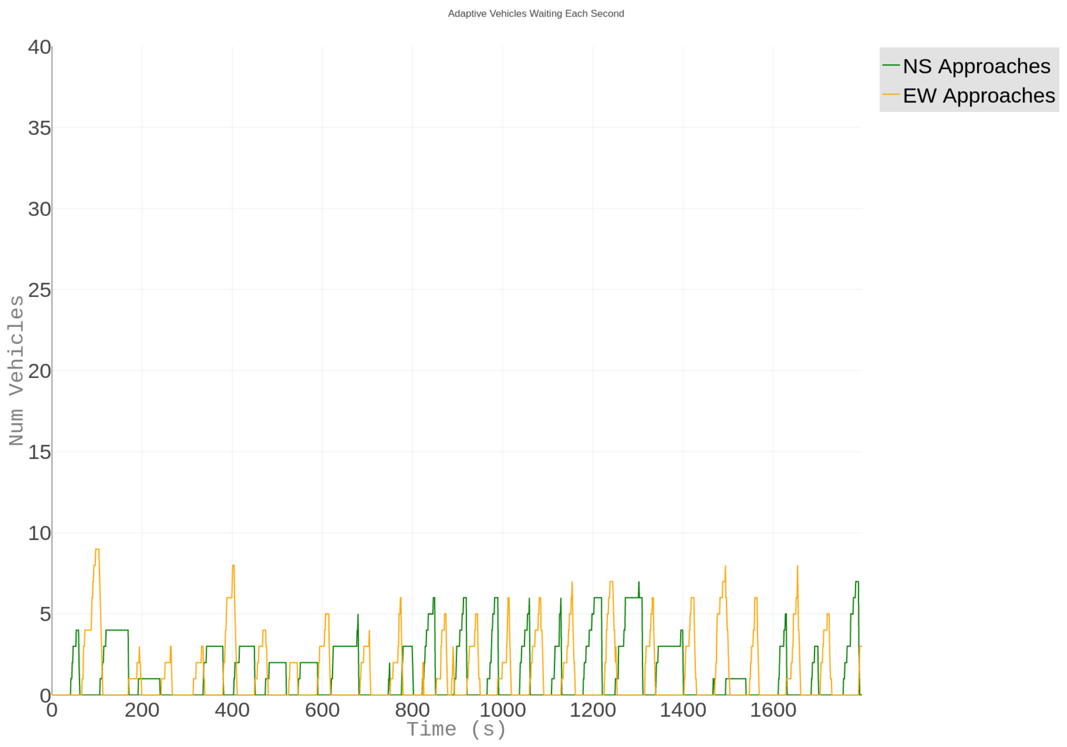


FIGURE 5.3: Number of Vehicles Waiting Over 30 Minute Simulation During Dynamic TLC Test

variations in green phase lengths. This is due to the adaptive TLC extending the green phase duration.

5.3.1.4 A Look At All Three

The results of the tests above show that the static TLC performs worst with a lower value for the minimum green duration (which it is important to note is also the maximum green phase duration for the static TLC), and the placement of induction loops has no affect on it. The static algorithm performance increases with a high green phase duration of 30 seconds however making this too high will begin to have a negative impact. The opposite was true for the actuated and dynamic TLCs.

Both perform increasingly worse the longer the duration of the minimum allowable green phase duration. The changes in performance were not as drastic as they are for the static TLC, which can be attributed to their adaptable nature. The reasoning behind the reduced performance as the minimum duration increases for these two TLCs is due to the ability for them to react is constricted by a shorter period of time in which they are allowed to adapt. For example with the adaptive TLC, if the minimum duration is 20 seconds and the maximum is 80 seconds, then it has 60 seconds in which it is able to extend by. Whereas if it had a minimum duration of 30 seconds, that space is only 50 seconds. This means that the TLC must wait for a longer period of time before it is allowed to adapt to the traffic flow even if it would have been better to adapt at an earlier point in time.

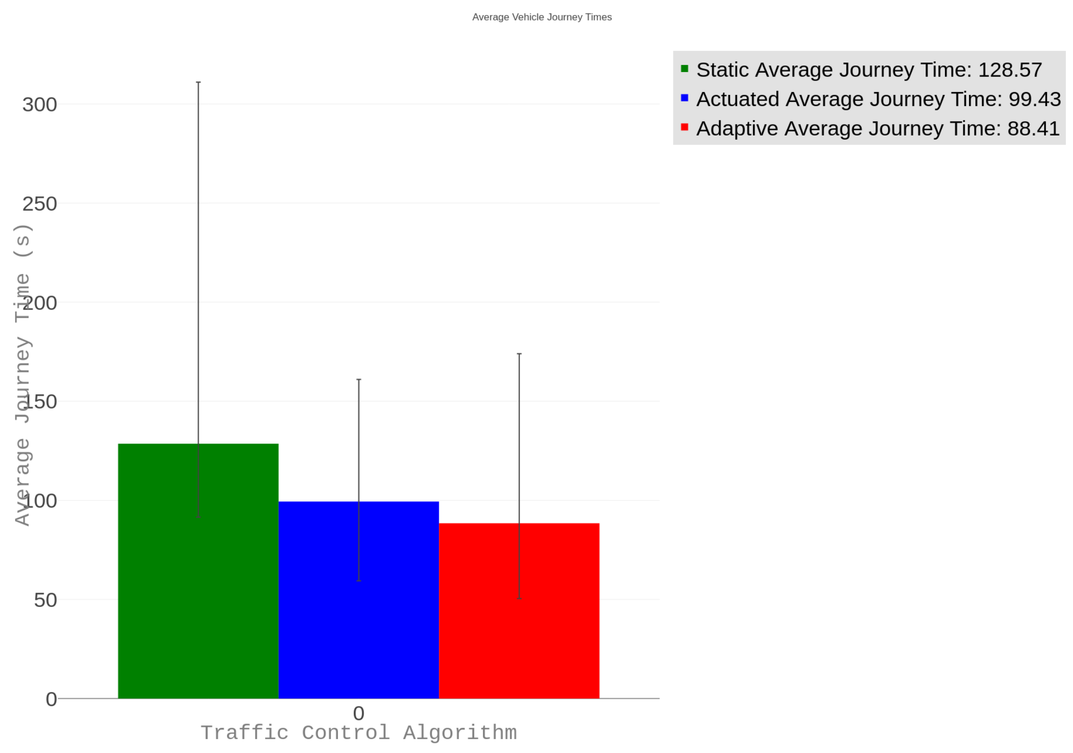


Figure 5.4: The Best Performing Configuration of Each TLC Regarding Average Trip Duration

Looking at Figure 5.4, the best performing configurations for each of the TLCs are graphed next to each other and it is clear to see the performance of each. The static had the highest average journey duration for vehicles and its margin of error is extremely high. The actuated TLC performed quite well in comparison and with a much tighter margin of error. The adaptive TLC performed the best, however this is where averages might hide the overall picture: the margin of error was considerable compared to the actuated, indicating that the actuated TLC

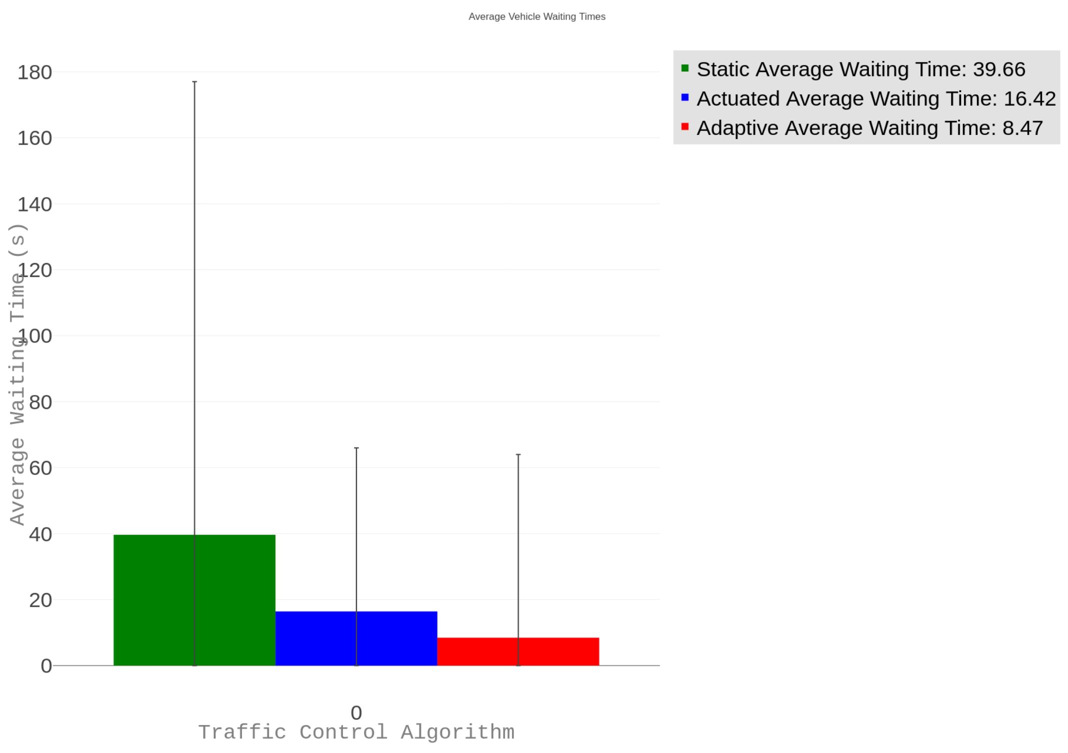


FIGURE 5.5: The Best Performing Configuration of Each TLC Regarding Average Waiting Times

serviced the traffic demand better for all the vehicles involved and not just the majority. A similar result can be seen in Figure 5.5 with the average vehicle waiting times in regards to the difference in margin of error for the actuated and adaptive TLCs.

5.3.2 Actuated Max-gap Time Variation

After the testing and analyzing of the green phase duration and induction loop distance variations, the best performing configuration for the actuated TLC from those tests was taken and optimized by altering the max-gap value. The best performing actuated configuration had a minimum green phase duration of 20 seconds. The values tested for the max-gap parameter ranged from 2 seconds to 6 seconds. It was discovered the actuated TLC performed best with a max-gap value of 4 seconds, with average vehicle waiting and journey times being 15.57 seconds and 97.37 seconds respectively. This can be seen in Figure 5.6 and 5.7.

Although the difference in performance with the max-gap value set to 3 seconds is extremely small for the average waiting and journey times with a difference of 0.23 seconds and 0.55 seconds respectively. It is noticed that increasing the max-gap time by too much reduces the performance for the actuated TLC in a similar fashion as increasing the minimum green duration. This is due to the green phase being prolonged unnecessarily when it would be a better decision to switch the phase to service the vehicles waiting.

5.3.3 Optimized Adaptive TLC

Separating the idea of the default duration and the maximum extension time proved to be of benefit to the performance of the adaptive TLC. This optimized adaptive TLC performs best with the maximum extension time set to 16 seconds and with the induction loops located 35 meters from the junction. No difference was noticed when using a maximum green phase duration of 80 seconds. This is due to the adaptive TLC having too much space to extend the green phase duration up to. This modified version of the adaptive TLC performs best when the maximum possible green phase duration is set to 60 seconds.

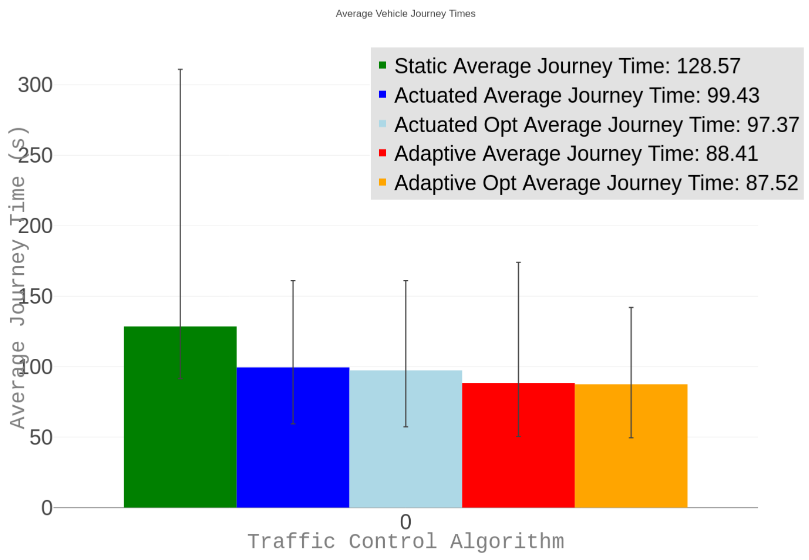


FIGURE 5.6: The Best Performing Configuration of Each TLC Regarding Average Journey Durations - Including Optimized Versions of Actuated and Adaptive TLCs

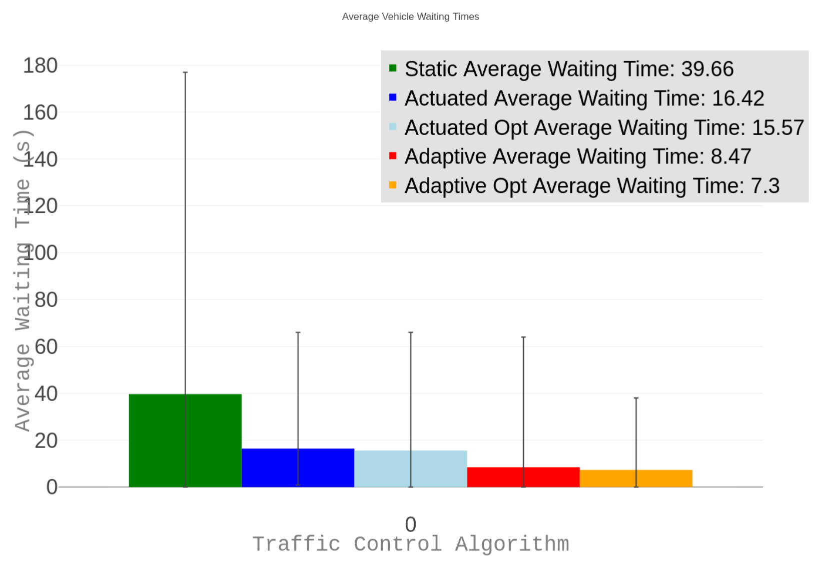


FIGURE 5.7: The Best Performing Configuration of Each TLC Regarding Average Waiting Times - Including Optimized Versions of Actuated and Adaptive TLCs

Following this, reducing the maximum extension time by any more than 16 seconds has no more perceived positive effects on performance. The values for the average waiting and journey times at this configuration were 7.29 seconds and 87.51 seconds respectively. This was an improvement of 0.89 seconds for the average journey time and an improvement of 1.18 seconds on average waiting times for vehicles.

Although the averages didn't improve drastically, the benefits can be seen in the error bars for the maximum values for both the waiting and duration times which indicates the overall performance of this version has marked benefits to the flow of traffic compared to its predecessor.

5.3.4 Heavy Traffic Flows

After conducting the previous experiments and analyzing their performances, the best performing configurations were used to test how each TLC handled high flows of traffic from both the NS and EW approaches. The best performing static TLC configuration had a minimum (and therefore maximum) green phase duration of 30 seconds. Figure 5.8 shows that the static TLC has an average journey duration of 117.92 seconds and average vehicle wait times of 33.99 seconds can be seen in Figure 5.9.

The best performing actuated TLC configuration consisted of a minimum green phase duration of 20 seconds and a maximum green phase duration of 60 seconds, with the max-gap value set to 4 seconds. The performance began to break down with high traffic flows and the margin of error is much higher than expected of the actuated after having an adequate performance under high traffic flows from one direction in previous examples.

The best performing adaptive TLC is the optimized version with the minimum green phase duration of 20 seconds and a maximum of 80 seconds with the induction loops at 35 meters and the maximum extension time set to 16 seconds. Interestingly the this adaptive TLC tends to be stable under both the high flows from one direction and the high flows tested in this section as shown in figures 5.8 and 5.9. The average vehicle journey duration and average vehicle waiting times manage to keep their values low while maintaining that lower upper bound for the margin of errors. This shows that this optimized adaptive traffic light control algorithm is the most stable approach for handling both high flows of traffic from one direction (EW or NS) and high flows of traffic from all directions.

The breakdown of the vehicle numbers waiting over the course of the simulation can be seen in the appendix in Figure A.1 and Figure A.2.

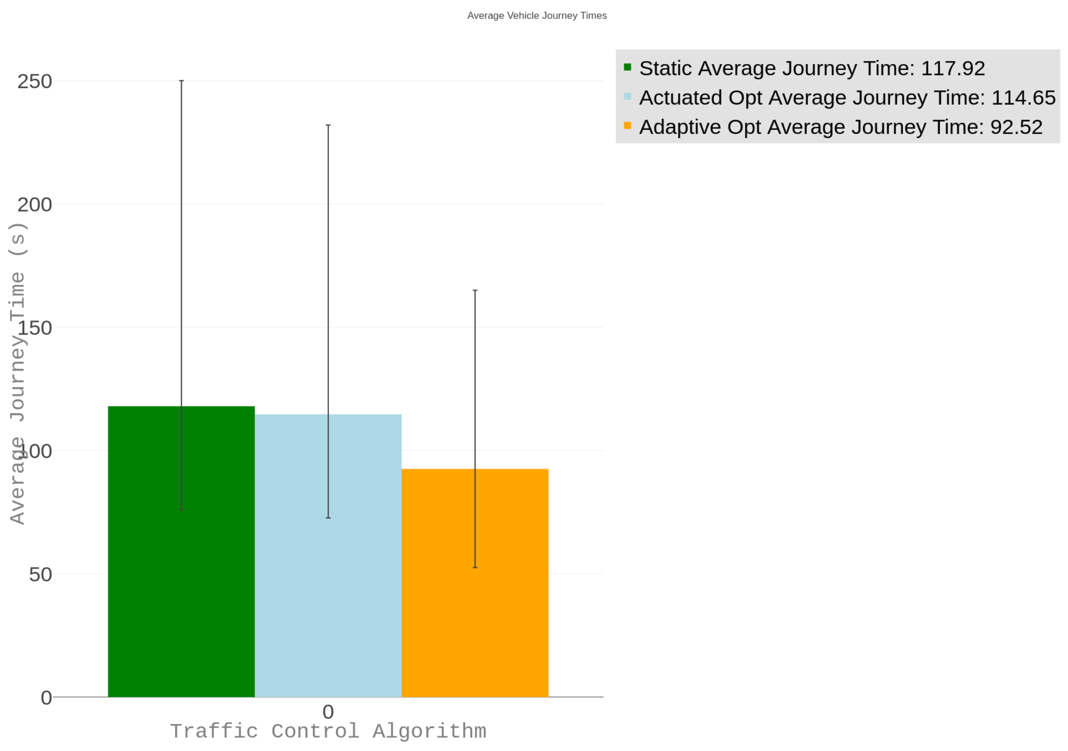


FIGURE 5.8: The Best Performing Configuration of Each TLC Regarding Average Journey Durations - Using Optimized Versions of the Actuated and Adaptive TLCs

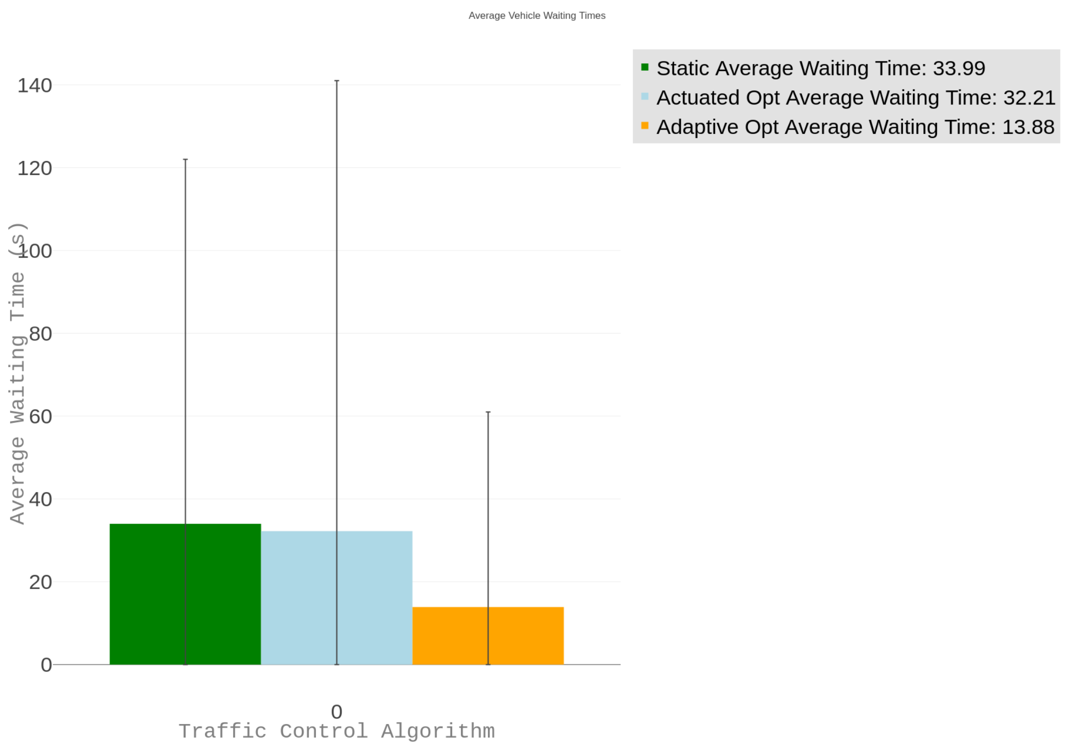


FIGURE 5.9: The Best Performing Configuration of Each TLC Regarding Average Waiting Times - Using Optimized Versions of the Actuated and Adaptive TLCs

Chapter 6

Conclusion

6.1 Summary

This project aimed to provide a clear comparison between the static, actuated and adaptive methods of traffic light control (TLC) and their effects on a busy urban junction. The use of the Simulation of Urban Mobility (SUMO) microsimulation tool to model Dennehy's Cross (an urban junction located in Cork City, Ireland), implement the static, actuated and adaptive TLCs on this model and to output data from the simulation in order to be able to carry out an analysis of the performance of each under varying tests. Each TLCs performance was compared to the performance of the other two in order to determine which performed best under varying circumstances.
The adaptable traffic light control algorithm described by Rademacher (2017) was successfully implemented using Python scripts. The results in this report were successful in generally replicating some of the findings by Rademacher (2017). One of the points Rademacher (2017) makes in conclusion is that the developed algorithm needed more testing in relation to the induction loop sensor location, this report carried out these tests as well as testing what affects different green phase durations had with the sensors at these varying locations.
Furthermore, this report managed to add to the development of the adaptable

traffic control algorithm developed by Rademacher (2017) with the idea of the maximum extension time being separated from the default phase duration. The adaptable TLC was tested with varying default minimum green phase durations as mentioned earlier with the induction loop sensors at varying distances from the junction; it was found that performance improved with these located at 35 meters from the junction when using this modified version.

6.2 Experiment Results Analysis

The static TLC performed the worst in every test and the performance was shown to deteriorate over time as described by Hunt et al. (1981) when mentioning that fixed-time plans tend to be years out of date due to the amount of human-hours required to optimize a single junction.

The actuated TLC performance has a marked improvement in its ability to decrease average vehicle waiting and journey times and is a noteworthy approach to the control of signalized junctions in so far as its minimal costs and complexity to implement compared other more complicated and expensive intelligent traffic control approaches.

The dynamic TLC had the best performance out of all three TLCs and outperformed the other two approaches in every conducted test with the lowest average vehicle waiting and average journey times. Furthermore, this approach performed better at its worst performance out of all the tests than even the best performance of the other two TLCs out of all the tests. The only TLC that outperformed this adaptive approach was the optimized version of itself. Although the optimized approach does not improve drastically upon the averages, it does achieve lower maximum values for these averages, improving the overall servicing of better traffic flow management.

6.2.1 Application in Urban Environments

The results found in this report contribute to the analysis of the influence that different traffic light control strategies have on the flow of traffic and how they impact traffic conditions. Of the three traffic control strategies investigated during the course of this report, when optimized the adaptive approach presents itself as the better solution to managing traffic flow in an urban setting with the lowest average vehicle waiting and journey times.

6.2.2 Future Development Opportunity

6.2.2.1 Vehicle Data

It would be beneficial to the future development of this work if vehicle count data could be acquired from Cork City council. This would improve the accuracy of the analysis of the different traffic control strategies investigated during the process of this report. The only input that would change is the definitions of the vehicle demand in SUMO, the tests could simply be run again and their results parsed and analyzed.

6.2.2.2 Macroscopic Investigation

This report focused on one busy junction in Cork City. A macroscopic investigation on how the different traffic control strategies perform when applied larger road networks. It is suspected that the static and actuated strategies would perform similarly as they were found to in this report and by Rademacher (2017). The adaptive strategy, although performing well in a microscopic environment might overwhelm adjacent junctions due to over optimization. The development of some sort of intelligent communication between junctions may be required to be incorporated into the adaptive algorithm, such as estimated vehicle platoons leaving a junction and approaching an adjacent junction for example.

Appendix A

Graphs

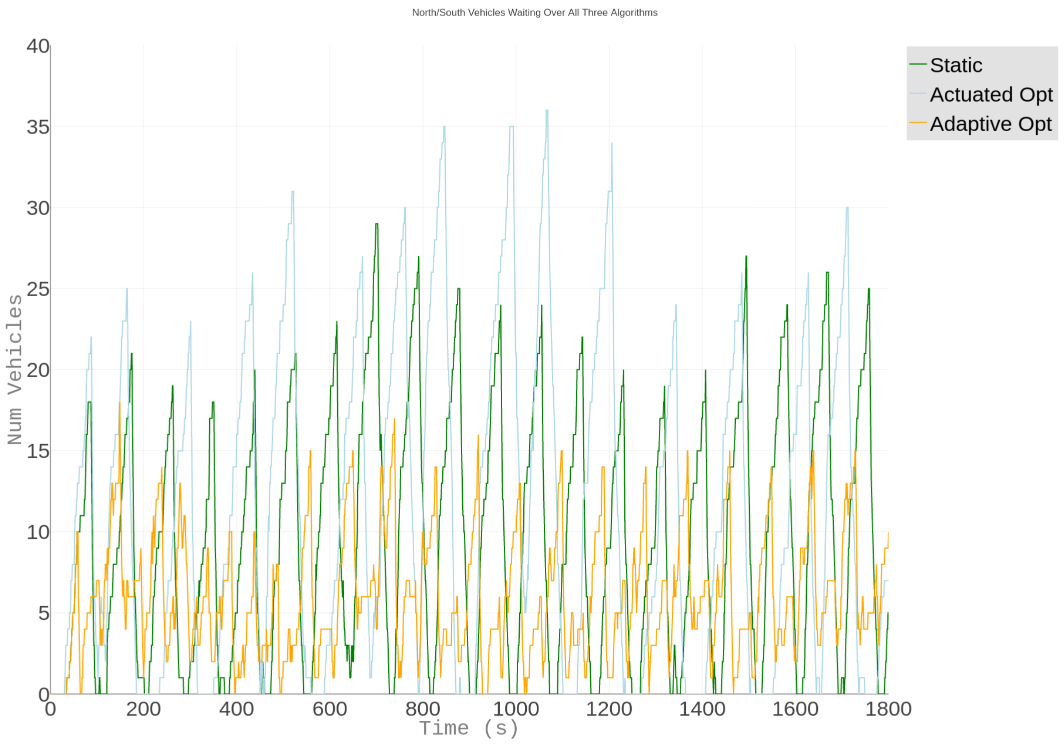


FIGURE A.1: The Best Performing Configuration of Each TLC Showing the Total Numbers of Vehicles Waiting at any Particular Point During the Simulation - Using Optimized Versions of the Actuated and Adaptive TLCs, this shows the North/South Waiting Vehicles

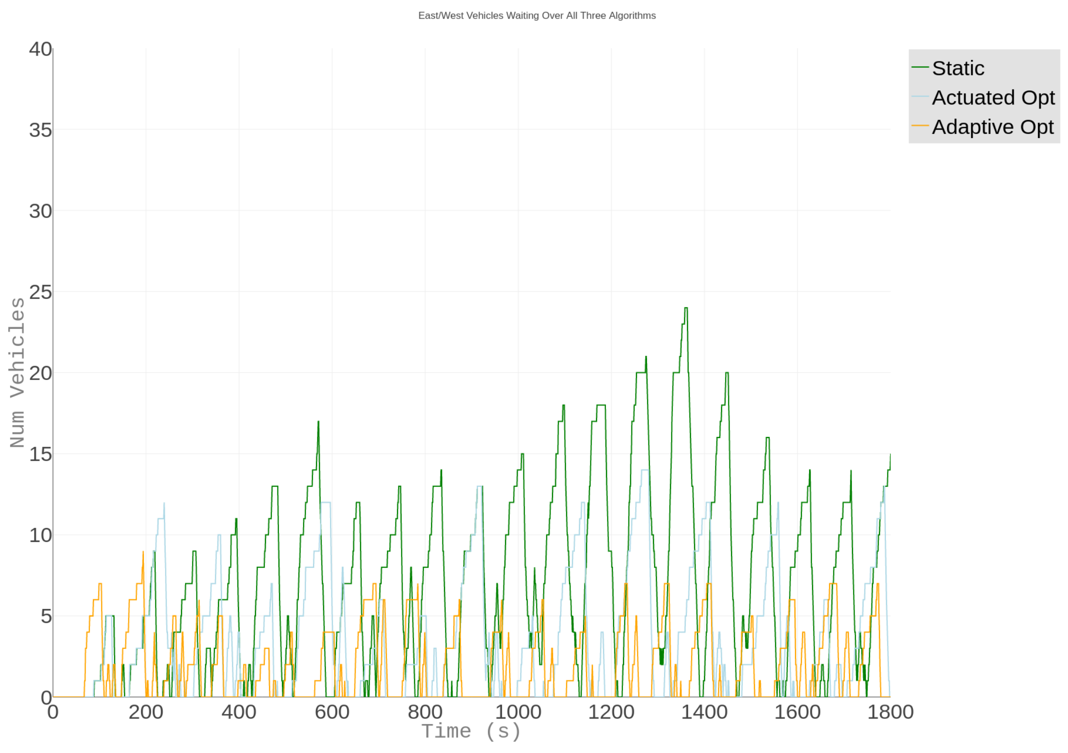


FIGURE A.2: The Best Performing Configuration of Each TLC Showing the Total Numbers of Vehicles Waiting at any Particular Point During the Simulation - Using Optimized Versions of the Actuated and Adaptive TLCs, this shows the East/West Waiting Vehicles

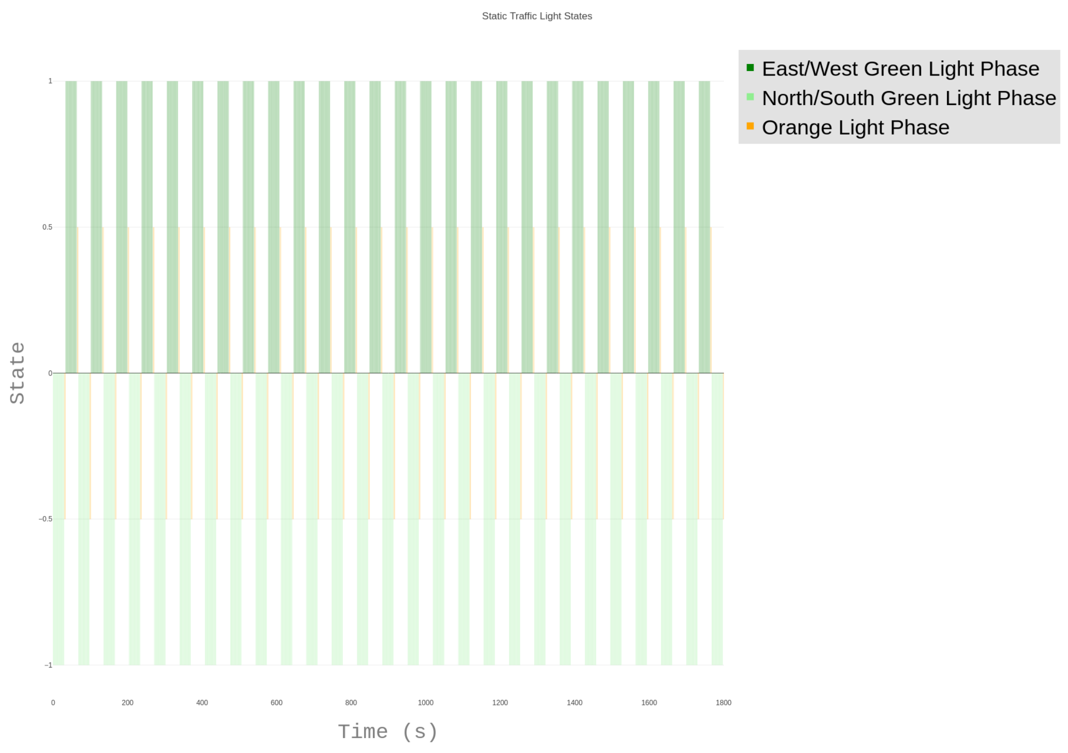


FIGURE A.3: Shows the Static Phase Switches Between Green Phase Directions over time with Intermediate Orange Phases

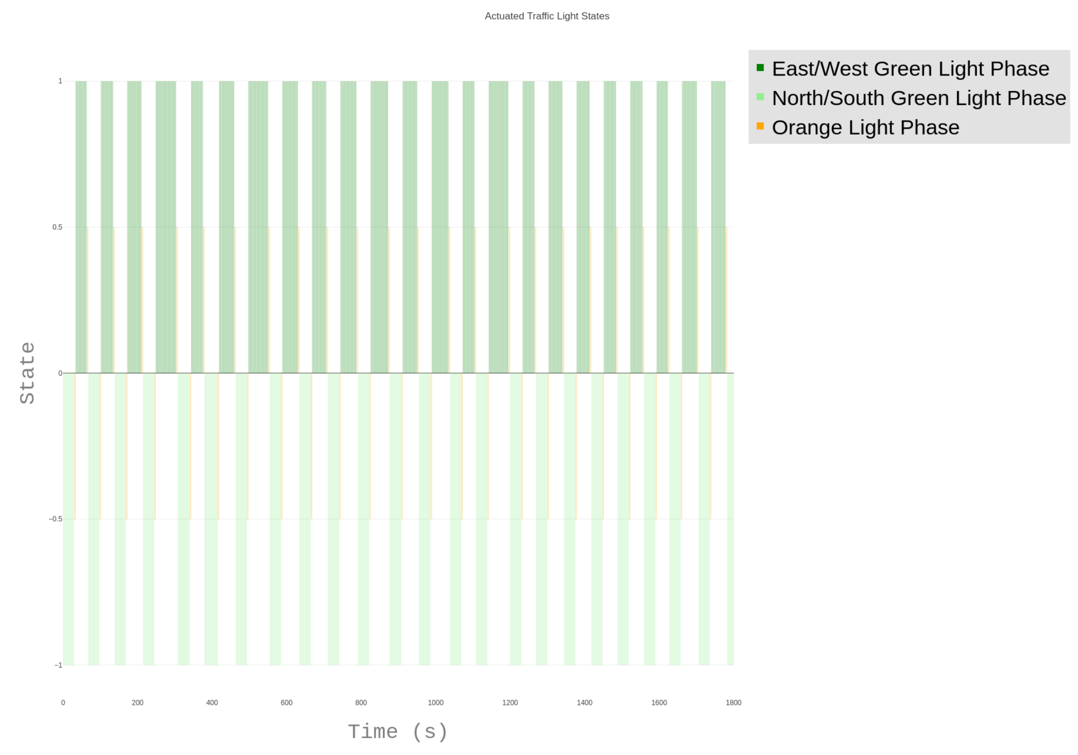


FIGURE A.4: Shows the Actuated Phase Switches Between Green Phase Directions over time with Intermediate Orange Phases

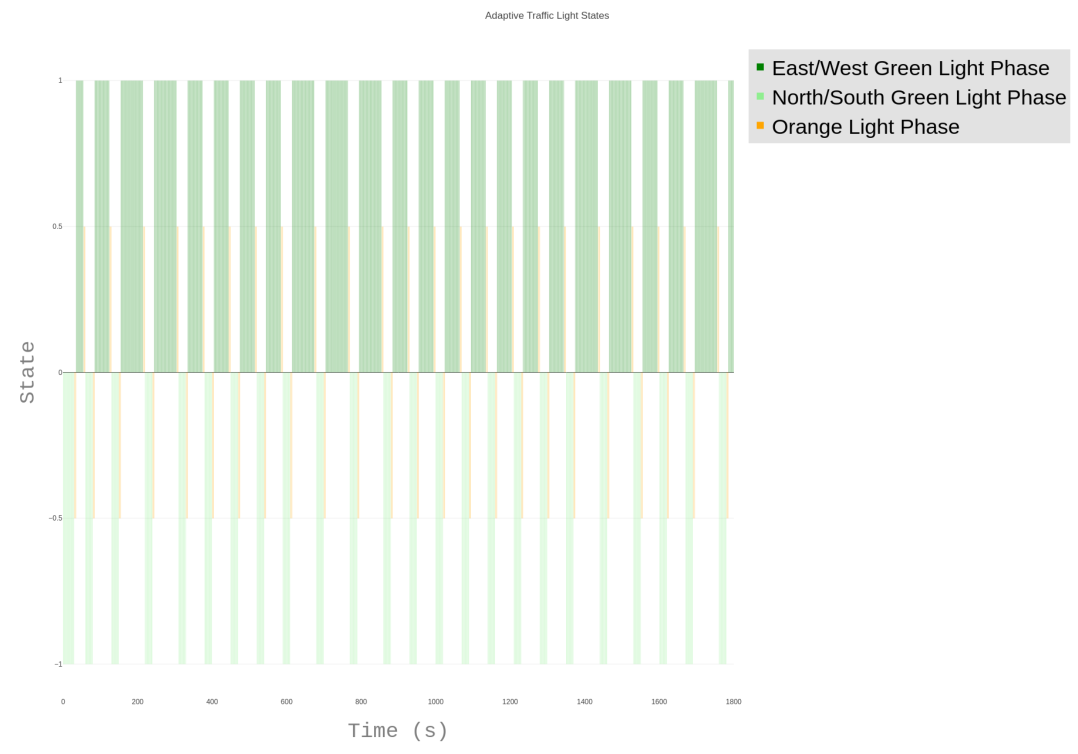


FIGURE A.5: Shows the Adaptive Phase Switches Between Green Phase Directions over time with Intermediate Orange Phases

Bibliography

[1] INRIX, 2017. URL http://inrix.com/scorecard-city/?city=Cork&index=396.

[2] K. Rademacher. The influence of sensor-based intelligent traffic light control on traffic flow in dublin, 2017. URL https://arrow.dit.ie/cgi/viewcontent.cgi?article=1104&context=scschcomdis.

[3] Cork City Council. URL http://www.corkcity.ie/services/roadstransportation/transportationdivision/trafficcontrol/.

[4] Kergaye Stevanovic and Martin.

[5] R D Bretherton P B Hunt, D I Robertson and R I Winton, 1981. URL https://trl.co.uk/sites/default/files/LR1014.pdf.

[6] Robertson & Bretherton. Optimizing networks of traffic signals in real time-the scoot method, 1991. URL https://doi.org/10.1109/25.69966.

[7] & Boukerche A. Younes, M. B. Intelligent traffic light controlling algorithms using vehicular networks, 2016. URL https://doi.org/10.1109/TVT.2015.2472367.

[8] PlotLy. URL https://plot.ly/.