# Dynamically Routing Reserve Buses to Reduce Bus Passenger Wait Time 

by

Mark Jacob, B.A. (Mod.)

Dissertation
Presented to the
University of Dublin, Trinity College
in fulfillment
of the requirements
for the Degree of

## Masters in Computer Science

# University of Dublin, Trinity College 

Supervisor: Siobhán Clarke

May 2017

## Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

## Mark Jacob

May 17, 2017

## Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Mark Jacob

May 17, 2017

## Acknowledgements

I would like to express my sincere gratitude to my supervisor, Siobhán Clarke, for her guidance and support throughout the course of this dissertation.

I would like to thank my family members for the support and encouragement they have shown me throughout my years of study.

Finally, I would like to thank my fellow classmates for their support throughout the duration of my time in college.

## Abstract

Even with the technological and analytical advances available today, passengers waiting at stops for public buses still suffer from long waiting times and witness the bus bunching problem. The bus bunching problem is the phenomenon by which a group of two or more buses which were scheduled with even headways along the same route arrive at the same location at the same time. External influences to the bus route such as pedestrian and vehicle traffic, local event crowds and construction delays can often drastically effect the punctuality of a bus. This is because urban bus transportation networks often still operate on a fixed schedule basis. This can leave passengers waiting at their stop for an elongated period of time and result in further delays in their day, especially in situations where a scheduled bus unexpectedly reaches capacity.

Buses in urban public transportation networks are not typically re-routed off their designated route until the route has been completed. There are several approaches that have been proposed in order to reduce bus bunching, optimize bus routing and therefore reduce passenger wait time. Bus holding methods researched in previous works propose
holding the bus at control points on different conditions. Approaches adopting selfequalizing headways between buses and utilising a speed change \& holding control model have been proposed in previous studies. However, although these approaches can reduce bus bunching and slightly reduce passenger wait time, these systems would not adapt to cater for a sudden large unexpected influx of passengers.

This paper outlines a new approach to allow for a small surplus of reserve buses to have the ability to be dynamically routed to areas in need of extra passenger capacity. The aim is to reduce passenger wait time at bus stops, particularly investigating the behaviour when a large unexpected influx of passengers arrives at a bus stop or area of bus service and fills a bus to capacity. A web application was developed using Leaflet JS, D3.js and the Graphhopper API which allows for the input and adjustment of parameters such as traffic density and the volume of passengers waiting at bus stops. The application can simulate a bus's journey along its typical route and, if the bus reaches capacity, the journey of dynamically routed reserve buses will be simulated and the journey statistics will be displayed at the end for comparison.

The findings of this work can be summarised as follows: using dynamically routed reserve buses, particularly during events involving a scheduled bus unexpectedly reaching capacity, is feasible to potentially reduce passenger wait time compared to the average scheduled bus. The reserve bus must be waiting within close proximity to the problem area to produce significant reductions but with further work this distance could be increased and thus the feasibility of a reserve bus implementation increased.

## Contents

Acknowledgements ..... iv
Abstract ..... v
List of Tables ..... x
List of Figures ..... xi
Chapter 1 Introduction ..... 1
1.1 Subject of Research ..... 1
1.2 Motivation for Research Topic ..... 2
1.3 Dissertation Overview ..... 3
Chapter 2 State of the Art ..... 4
2.1 Introduction ..... 4
2.2 Reducing Passenger Wait Times ..... 5
2.2.1 Gap in State of the Art ..... 9
2.3 Vehicle Routing ..... 10
Chapter 3 Design ..... 15
3.1 Introduction ..... 15
3.1.1 What is a 'reserve bus'? ..... 15
3.1.2 What is unique about this proposed use of the reserve bus? ..... 15
3.1.3 How does the reserve bus know when to begin operating? ..... 16
3.2 Operational Stages ..... 16
3.2.1 Waiting Stage ..... 17
3.2.2 Travelling to Start Stage ..... 18
3.2.3 Operational Stage ..... 18
3.3 Reserve Bus Trigger ..... 19
3.3.1 Trigger Format ..... 20
3.4 Decision Making Process ..... 22
3.4.1 Percentage of Route Complete ..... 23
3.4.2 Estimated Time Until Next Scheduled Bus Arrival ..... 23
3.4.3 Estimated Time Until Reserve Bus Arrival ..... 24
3.4.4 Minimum Reduction in Passenger Wait Time by Reserve Bus ..... 24
3.4.5 Example Decision Situations ..... 25
Chapter 4 Simulation Design ..... 29
4.1 Introduction ..... 29
4.2 Data Retrieval ..... 30
4.2.1 Real Time Passenger Information API ..... 30
4.2.2 Directions via Graphhopper API ..... 33
4.3 Front End Libraries ..... 34
4.3.1 Leaflet ..... 35
4.3.2 OpenStreetMap ..... 35
4.3.3 D3.js ..... 36
4.4 Running The Simulation ..... 37
4.4.1 Random Passenger Generation ..... 38
4.4.2 Traffic Weight ..... 39
4.4.3 Bus Travel Speed ..... 39
4.4.4 Second Scheduled Bus ..... 40
4.4.5 Reserve Bus Trigger ..... 40
Chapter 5 Evaluation ..... 42
5.0.1 Approach ..... 42
5.0.2 Results ..... 45
5.0.3 Evaluation ..... 51
Chapter 6 Conclusion ..... 55
Chapter 7 Further Work ..... 57
7.1 Capacity Prediction ..... 57
7.2 Passenger Crowd Information Retrieval/Estimation ..... 58
Bibliography ..... 59

## List of Tables

3.1 Reserve Bus Trigger Parameters ..... 19
3.2 Example Reserve Bus Trigger Situations. ..... 26
3.3 Example 1: Evaluation for operation of reserve bus. ..... 27
3.4 Example 2: Evaluation for operation of reserve bus. ..... 27
3.5 Example 3: Evaluation for operation of reserve bus. ..... 28
3.6 Example 4: Evaluation for operation of reserve bus. ..... 28
5.1 Calculating the average $W_{k m}$. ..... 53

## List of Figures

2.1 State Of the Art: Bus Control Method Evaluations. ..... 10
2.2 State Of the Art: Routing Problem Algorithm Evaluatiosn ..... 14
3.1 Reserve Bus Trigger Activity Diagram ..... 20
4.1 Dublin Bus Real Time Information GET Request Example ..... 31
4.2 Simulation Objects Created With D3 ..... 36
4.3 Example of User Input: Bus Route, Passenger Volume and Traffic Volume ..... 37
4.4 Example of User Input: Selecting Reserve Bus Starting Point ..... 38
5.1 The Five Simulated Routes ..... 43
5.2 Dublin Bus Depots for Reserve Bus Starting Points. ..... 44
5.3 Results: Route 1 ..... 48
5.4 Results: Route 44 ..... 49
5.5 Results: Route 45a ..... 49
5.6 Results: Route 66 ..... 50
5.7 Results: Route 77a ..... 50

## Chapter 1

## Introduction

### 1.1 Subject of Research

This dissertation explores the design and simulation of a dynamic routing system for reserve buses which provides relief to waiting passengers at bus stops, paying particular attention to situations where a normal scheduled bus unexpectedly reaches capacity and is no longer able to accept passengers at future stops. Not only can the reduction in passenger wait time increase passenger, i.e. customer, satisfaction but it could also potentially reduce revenue lost for the bus organisation to other mediums of transportation.

As discussed in the State of the Art in Chapter 2, a large amount of literature in the bus transportation topic focuses on methods to reduce the effects of 'bus bunching'. 'Bus bunching' refers to a group of two or more buses that were scheduled to run at different times or intervals on the same route but end up running in the same location at the same time. This is not an ideal situation for passengers as it typically results in longer wait times since two buses are essentially operating on the same stops at
the same time rather than being separated by a certain time interval. Although the methods discussed in the State of the Art are effective at reducing the effects of bus bunching, they still do not specifically cater for cases when there is a large influx of passengers that is capable of filling a bus to capacity in a short time period where additional bus capacity will be required.

This dissertation will detail how the use of dynamically routed reserve buses can be used to address this gap in methods used to reduce passenger wait time.

### 1.2 Motivation for Research Topic

Public bus transportation remains to be one of the major forms of transport utilised in major urban areas across the world. Taking Dublin, Ireland as an example, $61 \%$ of all public transport trips into Dublin city centre are on public Dublin Bus services, with Dublin Bus remaining the largest public transport provider in the country [10].

One of the disadvantages of public bus networks is that they typically rely on a fixed departure timetable and fixed routes regardless of the presence of external influencers such as weather, accidents, construction works, local events, etc. that can significantly effect the number of passengers waiting at bus stops.

With recent technological advances, a large number of cities are implementing public buses equipped with GPS tracking technology and accompanying real time information infrastructure ([12], [34], [26]). However, despite this rise in technological presence in the public bus transportation networks, many bus networks still rely on a fixed departure timetable and fixed routes. Although these timetable designs may have been guided by statistics of past passenger behaviour to provide the most optimal timetable, they do not cater for unexpected large influxes of passengers.

The lack of innovation in dealing with external influencers that can cause a large influx of passengers unexpectedly can increase waiting time for passengers if a bus reaches capacity unexpectedly. This forces waiting passengers at future stops to wait for the next scheduled bus, which may not be for another 20-40 minutes, before they can board to continue their journey. This also leads to a larger number of passengers than usual boarding this next scheduled bus as it includes the typical passenger load plus the passengers who were denied entry to the first bus due to it reaching capacity.

### 1.3 Dissertation Overview

Chapter 2 contains the State of the Art which discusses background information on previous methods studied in attempt to reduce passenger wait time such as bus holding, self-equalizing headways and a model which implements speed alteration and holding as well as approaches to vehicle routing in a commercial setting.

Chapter 3 provides an in depth description of the proposed reserve bus system design, explaining the concept of the reserve bus, the operational stages involved and how the decision on whether to route the reserve bus or not is made.

Chapter 4 contains a description of the design and implementation of the web application tool built for running simulations,

Chapter 5 provides some simulation results and discussions thereof.

Chapter 6 provides the conclusions gathered from the analyses of results.

Chapter 7 discusses some potential further work on this topic.

## Chapter 2

## State of the Art

### 2.1 Introduction

Currently in public transport, particularly in the bus industry of public transport, it is typical to have a route timetable with static departure times. These timetables do not cater for external influences in public transport such as heavy traffic congestion, road works or construction delays and traffic \& pedestrian influx due to local events. These external influences may change on a day-by-day, or even hour-by-hour, basis and a static timetable cannot provide optimal route departure times under these circumstances. A change from the norm, with regards to the external influences, can lead to a larger than usual crowd of travellers waiting at any given bus stop.

This section of the dissertation will provide a state-of-the-art report for routing and reaction approaches that relate to the public transport sector. Section 2.2 will detail approaches to the bus-bunching problem and section 2.3 which discusses routing (and re-routing) algorithm approaches.

### 2.2 Reducing Passenger Wait Times

When a bus network redesign is not an option, which it typically is not, bus transportation services must make other actions in order to reduce passenger waiting times. Without changing the route(s) of the affected buses, the remaining option is to implement some form of control method on the bus behaviour in an attempt to reduce the amount of time spent by passengers waiting at any given stop.

Typically, buses in public bus networks are not re-routed off of their scheduled route until they have completed the route, beginning at the starting stop and completing at the final scheduled stop. Due to this, the control methods used to reduce bus bunching are often the only intentional action performed by the bus driver that results in a variation in the buses movement.

## Bus Bunching

In 1964, Newell and Potts [28] discovered that if buses do not make use of a control policy the buses are unable to remain on schedule. If a bus is slightly delayed it will end up waiting for a longer period of time at its next stop as the number of waiting passengers has increased making the time required for passenger boarding at the stop longer. With the increased delay, the headway (the time between consecutive vehicles) is reduced and will eventually result in both buses arriving at the same location at the same time. This is known as the bus bunching phenomenon.

When a transportation company is creating a scheduled timetable for its fleet, it typically inserts a small amount of slack into the schedules to attempt to reduce the vehicle bunching effect. This involves designing the schedule with longer vehicle trip times between stops than is required on average, and then ensuring that vehicles do not depart before the scheduled time. Daganzo [6] outlined this approach and found that it is more beneficial to have control points spaced out more widely across stops.

There are various different real time control strategies for routing buses through a network of bus stops, such as holding, stop-skipping, dead-heading and the use of reserve buses.

## Bus Holding

Bus holding involves holding a bus at a bus stop or depot to intentionally delay the bus' movement. This is done to reduce the variation in headway between buses on the same route, generally when a bus is ahead of schedule, therefore also reducing the occurrences of bus bunching. A "schedule based" bus holding method is often implemented which instructs bus drivers to not depart a stop before the dictated scheduled departure time for that stop.

Prior to the work done by Eberlein et al. in 2001 [13], research on the holding problem had always assumed that no real-time information was available. Eberlein et al. formulated the holding problem as a deterministic quadratic program in a rolling horizon scheme and designed an algorithm to solve the problem. It was tested with headway data collected by an automated system and the impact of the holding policies was investigated.

Eberlein et al. found that the passenger demand pattern did not heavily influence the holding solution, rather the solution heavily depended on the vehicle headway pattern at the control point and the headway of the previous vehicles along the route line. Thus they concluded real-time information on vehicle headways may be sufficient when examining holding policies and real-time information on passenger demand may not be required. They found that holding can reduce passenger waiting times and riding times in high frequency transit lines where interstation stopping frequently occurs. However, it's important to note that the testing done was based on collected data from a light rail system and not a bus network, which has some different limitations: (i) buses may easily pass each other, (ii) no minimal headway exists between buses and (iii) trains are not effected by other traffic to the extent that buses are.

Similarly, the study performed by Mark D. Hickman (2001) [17] describes a model that may be used for real-time control purposes. It describes an analytic stochastic model for the transit vehicle holding problem to determine the optimal vehicle holding time at a control stop along a transit route. Hickman models a hypothetical bus route and uses this route to illustrate the holding model and to perform the simulation. This work differs from that of Eberlein et al. [13] because Hickman's analysis explicitly includes stochastic elements. The bus running times and the passenger arrival and alighting processes are taken to be stochastic rather than deterministic. From his simulations he found that his full model gave an average reduction in the wait time of passengers of approximately $3.5 \%$ of the total waiting time and a $20 \%$ reduction in waiting time due to headway variability.

Holding buses at discrete control points for brief periods of time was proposed by Daganzo (2009) [7], where the holding times for a bus are determined dynamically based on real-time headway information. Results of a simulation of the approach show that by holding a bus at control points with this approach allows buses to travel at a reasonably fast speed compared to the schedule based approach, thus reducing in-vehicle passenger delay. Building on [7], Daganzo (2011) [8] proposes an adaptive control scheme that adjusts the speed of the bus in real-time based on the space between it and both the (i) bus ahead and (ii) the bus behind. This approach was shown to result in regular headways with faster bus travel than previously founded control methods such as those mentioned above.

## Self-Equalizing Headways

The papers that examine control methods to reduce bus bunching mentioned thus far have all been based off of systems incorporating the traditional fixed bus schedule. Bartholdi and Eisenstein [1] proposed a new method of coordinating buses in order to resist bus bunching. Their method completely abandons the idea of a fixed schedule for bus departure and promotes dynamically self-equalizing headways. It continuously attempts to keep the headway between buses at a suitable value to prevent bus bunching,
requiring control points, which are certain bus stops where a bus can be purposefully delayed to prevent it from catching up to the bus ahead and resulting in bus bunching. By eliminating a static schedule, their approach can manage the departure of buses to work efficiently with the buses already in operation or en route.

The headway of each bus is changed when it arrives to the control point to a weighted average of its former headway and the former headway of the trailing bus. If its former headway was larger, its new headway becomes smaller and vice versa. This results in the headways being continuously adjusted in an attempt for them to become nearly equal. In the case that buses do become severely bunched, there is a stipulation that successive buses departing the control point must be separated by a minimum amount of time. The approach was tested on a real bus route and a simulation of a real route was performed. They found that their self-equalising approach had lower standard deviations and lower average headways compared to a typical target schedules approach and compared to a target headway approach (like that proposed by Daganzo (2009)) [7].

## Speed Change \& Holding Control Model

Nesheli et.al [27] investigates using real-time control actions to minimise the occurrence of vehicle bunching, with a particular focus on reducing vehicle bunching while increasing the chance of a direct passenger transfer (a passenger departing from one bus at a certain stop and boarding another bus at the same stop). The authors outline two control models, a Speed-Change Control Model and a Holding Control Model. The Speed-Change Control Model involves continuously adjusting the speed of the vehicles in such a way that an acceptable headway can be restored between the two vehicles with the aim of preventing bunching. The required alteration in speed is calculated based on the relative location of the buses. They assume that the GPS location of both buses is known. The Holding Control Model involves holding a vehicle at a 'control point' to improve its on-time performance. A vehicle can be held in relation to headway or in relation to its scheduled departure time. When using a headway holding control
strategy, the vehicle is held at the 'control point', which is typically a regular stop, until the headway between it and the other vehicle is at an acceptable value. When using a holding control strategy that is based on the scheduled departure time of a vehicle, this simply involves holding the vehicle at the control point until the scheduled departure time. The study found that the proposed control policy, a combination of the "Holding and Speed-Changes" policies, performed "significantly better", providing a "much more stable performance with standard deviations meaningfully lower than the no-control policy". There was also a reduction in total waiting time and a reduction in headway variations.

### 2.2.1 Gap in State of the Art

Although the methods described such as bus holding, self-equalizing headways and the speed change $\&$ holding control model can prove to normalise headways between buses and thus reduce passenger wait time, none of the methods cater for the situation where extra capacity is required on a route, in the form of another bus. If an unexpected large influx of passengers was to congregate across a small number of bus stops, causing the scheduled bus to reach capacity then adjusting the headway between this bus and the following bus to self-equalise will not reduce the wait time of those passengers who can now not board the bus due to its capacity.

Figure 2.1 displays a visual summary of the discussed methods and the practices adopted within those methods.

|  | Bus <br> Network Specific | Uses dynamic scheduling | Vehicle control method considers position of other vehicles | Uses real-time informatio n | Deploys additional bus when passenger demand increases |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Bus Holding |  |  |  |  |  |
| Eberlein (2001) | (light-rail system) | X | X | $\checkmark$ | X |
| Hickman (2001) | $\checkmark$ | X | X | $\checkmark$ | X |
| $\begin{aligned} & \text { Daganzo } \\ & \text { (2009) } \end{aligned}$ | $\checkmark$ | X | $\checkmark$ | $\checkmark$ | X |
| $\begin{aligned} & \text { Daganzo } \\ & \text { (2011) } \end{aligned}$ | $\checkmark$ | X | (considers position of both bus behind and on front of current bus) | $\checkmark$ | X |
| Self-Equal izing Headways |  |  |  |  |  |
| Bartholdi \& Eisenstein (2012) | $\checkmark$ | (Schedule abandoned continuously attempts to keep headway suitable) | $\checkmark$ | $\checkmark$ | X |
| Speed <br>  <br> Holding <br> Control <br> Model |  |  |  |  |  |
| Nesheli et. al (2016) | $\checkmark$ | (assumes GPS location of buses known \& uses this to decide on speed/hold control) | $\checkmark$ | $\checkmark$ | X |

Figure 2.1: State Of the Art: Bus Control Method Evaluations.

### 2.3 Vehicle Routing

The decision on where and when to route a reserve bus to an area in need of extra resources can have many influencing factors. The distance and time between the current
point of the bus and the starting point of the route the bus will be joining are two important factors. When this reserve bus has completed its required route, it will then be available to be assigned to another route. However, the bus is now at a different starting point than before. With several reserve buses in operation, it would be beneficial to attempt to route the buses to routes in such a way that the time spent deadheading, the term used to describe when a bus operates without carrying or accepting passengers, is minimised. This problem is a form of the Vehicle Routing Problem.

## Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) is an integer NP-complete programming problem which is a form of the Travelling Salesman Problem. The VRP is the problem of designing the most effective and optimal route for vehicles used in the distribution of goods or services to geographically distributed customers or nodes. The problem is an old, well-known problem proposed by Dantzig and Ramser (1959) [9] as the 'truck dispatching problem' with a vast array of research examining it, see Bellman (1958) [2], Magnanti (1981) [25], Raff (1983) [30], Laporte \& Nobert (1987) [20] and Laporte (1992) [19]. More recently, the book by Golden et. al [14] provides an overview of more recent work involving the VRP. The approaches to the VRP are often measured in terms of computation performance with regards to time and with regards to the quality of the solution.

Tabu search is a metaheuristic search method proposed by Glover [4]. Rego [31] presents a Tabu search algorithm for the VRP with capacity and distance restrictions which utilises a master-slave model and parallel processing. The goal of the algorithm is to converge to a good solution in a short timeframe and also to build a knowledge base that can be used in later phases of the algorithm. Analysis between the algorithm running in parallel and sequentially show that the parallel algorithm never returns a solution that is worse than the sequential algorithm and even returns a result that is an improvement for seven solutions by the sequential algorithm. Comparisons with computation times
for previous tabu search algorithms show an improvement in computation time.

Gror et. al (2011) [16] proposed a parallel algorithm for the VRP. The algorithm combines a heuristic local search improvement procedure with integer programming aiming for optimal performance, testing them against well known sets of benchmark problems. Their algorithm adopts a master-slave architecture, utilising a single master processor that coordinates the search and keeps track of the best solutions found by the slaves. The slaves can be either a heuristic solver or a set covering solver, where a:

- Heuristic Solver is a processor that generates and improves the solutions to the problem instance by running a metaheuristic algorithm
- Set Covering Solver is a processor that solves instances of a set covering problem with rotues taken from solutions discovered by other processors

The paper outlines the experimentation of the algorithm against different benchmark problems and varying the algorithm using different numbers of processors, different numbers of set covering solvers, varying the time and varying the master-slave distribution of tasks. They discover that, for up to 64 processors, by doubling the number of processors the solutions of "roughly equivalent quality" could be discovered in about half the time.

## Open Vehicle Routing Problem (OVRP)

The Open Vehicle Routing Problem (OVRP) is a variation of the VRP whereby vehicles do not, or are not required to, return to their depot once they have completed their deliveries. So rather than a vehicle beginning its route at the depot and ending it at the depot, the route ends at their last delivery location. The problem was introduced by Sariklis and Powell (2000) [32] and other research has been conducted on the problem since. A tabu search algorithm for the OVRP was presented in Brando
(2004) [3], Letchford et. al [22] presented a branch-and-cut algorithm and Li et. al [23] provides a review of different algorithms used to tackle OVRP using a set of eight large-scale problems to compare the algorithms' performance. The experiments found that approaches based on adaptive large neighbourhood search, record-to-record travel and tabu search performed well.

## Multi-Depot Open Vehicle Routing Problem (MDOVRP)

A bus network typically consists of more than one depot from which buses may depart. Multiple depots adds another parameter to the VRP (and the OVRP) and this is addressed by the Multi-Depot Vehicle Routing Problem (MDVRP) and the MultiDepot Open Vehicle Routing Problem (MDOVRP) where the vehicle will depart from its starting point of one of the possible depots and will complete its route at the last delivery point. Tarantilis and Kiranoudis (2002) [33] initially proposed the MDOVRP (also referred to as the OMDVRP) as an issue that was faced by a Greek industry that distributed fresh meat from depots to customers. They presented a new metaheuristic algorithm called List-Based Threshold Accepting (LBTA) algorithm. They performed a case study on a real-life distribution problem and found the algorithm provided sufficient results leading to the distribution company adopting the algorithm for daily operations.

Liu et. al (2014) [24] presents a mixed integer programming (MIP) mathematical formation for the MDOVRP as well as a hybrid genetic algorithm for solving the MDOVRP. With a similar approach, Lalla-Ruiz et. al (2016) [18] present a new mixed integer programming formulation for the MDOVRP, using Liu et. al (2014) [24] for a benchmark comparison. They find that their approach is performs better than the approach of Liu et. al, providing new optimal solutions for certain problem instances and performing in a quicker time frame. 1

|  | Buses start \& finish route at different locations | Considers multiple available buses at different starting positions | Performance |
| :---: | :---: | :---: | :---: |
| VRP |  |  |  |
| Rego (2001) \& Groer et. al (2011) | X | X | Master-slave, parallel processing. |
| OVRP |  |  |  |
| Li et. al (2007) | $\checkmark$ | X | Approaches based on adaptive large neighbourhood search, record-to-record travel and tabu search perform best |
| MDOVRP |  |  |  |
| Liu et. al (2014) | $\checkmark$ | $\checkmark$ |  |
| Lalla-Ruiz et. al (2016) | $\checkmark$ | $\checkmark$ | Performs better than Liu et. al (2014) provides new optimal solutions and performs quicker. |

Figure 2.2: State Of the Art: Routing Problem Algorithm Evaluatiosn.

## Chapter 3

## Design

### 3.1 Introduction

### 3.1.1 What is a 'reserve bus'?

A reserve bus is simply a bus that is not operating and is not associated with a specific route or timetable but is waiting as a backup or 'reserve' vehicle. The vehicle is capable of operating and should be prepared to be deployed into operation at any time.

### 3.1.2 What is unique about this proposed use of the reserve bus?

The reserve bus will be held at a certain location until it is required for a route in need of extra passenger relief. This will typically be when a scheduled bus reaches its
capacity unexpectedly and cannot accept passengers at future stops along its route. This forces waiting passengers at future bus stops to wait for the next scheduled bus and by doing so increases their waiting time at the bus stop and also increases their overall travel time.

Rather than simply routing the reserve bus to the start of a route that is in high demand, the reserve bus is dynamically routed to the specific area of the route that is experiencing a high demand due to a large number of waiting passengers. This reduces the time taken for the reserve bus to relieve passengers in the congested area as it does not waste time operating on the segment of the route prior to the congested area.

### 3.1.3 How does the reserve bus know when to begin operating?

When a bus reaches capacity it notifies the bus organisation that it has reached capacity and includes the details of its route and position. A decision must then be made to decide whether or not a reserve bus should be routed to help to relieve waiting passengers, which is covered in more detail in Section 3.4.

Ideally, there will be several reserve buses waiting at different geographic locations spread across the targeted area. The reserve bus with the shortest travel time to the area of demand can then be deployed if deemed beneficial.

### 3.2 Operational Stages

The reserve bus has three distinct operational stages:

- Waiting - the bus is waiting at its designated wait area until it is needed.
- Travelling to Start - the bus is travelling from its designated wait area to the start point of its route.
- Operational - the bus is operating as normal on its route, starting at the specified start point and ending at the final stop in that route, picking up and dropping off passengers along the way.


### 3.2.1 Waiting Stage

The waiting stage consists of the reserve bus waiting at its designated wait area until it is required to operate to relieve waiting passengers. The reserve bus only needs to be told when it needs to depart, what bus route it is following and at which stop is it beginning its operation.

Assuming that the reserve bus operator is a public transport organisation, there could be several bus depots or work areas that would be possible to accommodate the reserve bus as it waits. These depots would have adequate space to hold the reserve bus and they would also contain any maintenance infrastructure required for day-to-day repairs or tweaks on the bus.

There is a cost associated with holding a bus in reserve that is prepared for operation at any time including fuel cost, driver payment and maintenance cost. This cost then multiplies when multiple waiting areas are implemented. This is an important factor for the bus organisation to consider as they must decide if the cost justifies the potential reduction in passenger wait time.

### 3.2.2 Travelling to Start Stage

If the outcome of the decision making process is to deploy the reserve bus to the route in need of extra passenger capacity, the reserve bus will then depart the waiting area and drive to the starting point for its operation. This starting point will not be the typical starting point of the route, but rather the bus stop at which the request for extra capacity was sent from. Due to the fact that this bus is travelling to a different starting point in the route than the usual scheduled bus, the fastest route to the starting point may be different to that typical route a bus driver takes to the first stop.

The fastest route from the reserve bus waiting point to the designated starting point, taking into account any road restrictions for buses, is calculated and provided to the driver of the bus. The bus then departs the waiting area and travels to the starting point. The bus is dead heading, i.e not accepting any passengers, until it reaches the starting point.

### 3.2.3 Operational Stage

Once the reserve bus reaches the specified starting point, it operates as normal for the remainder of the route (picking up and dropping off passengers).

When it completes the route, it will be re-routed to another area in need of passenger relief if deemed beneficial or else it will return to a waiting area.

### 3.3 Reserve Bus Trigger

In order to determine whether or not a reserve bus would be beneficial to a route, the bus organisation must first be informed where there is a possible requirement for extra passenger relief. This is done by a bus driver sending a simple notification, or 'trigger', to the bus organisation when it has reached capacity, regardless of the stop or progress along the route of the bus.

By instructing the bus driver to report when the bus has reached capacity regardless of any other factors, this significantly reduces the potential for human intervention whereby the driver might decide a reserve bus is not needed for the route.

The trigger sent by the bus driver is sent to the bus administration system with the required details including the bus capacity status, the route the bus is operating on and the current stop from which the trigger is being sent.

Table 3.1: Reserve Bus Trigger Parameters

| Name | Value |
| :--- | :--- |
| Capacity Status | \{boolean\} eg: true |
| Route | \{string\} eg: '77a' |
| Current Stop | \{int\} eg: 391 |

These three pieces of information can be used in combination with the guideline figures outlined in this paper to determine whether it will be beneficial to route any of the available reserve buses to the areas.


Figure 3.1: Reserve Bus Trigger Activity Diagram

### 3.3.1 Trigger Format

Assuming that the buses are not equipped with technology that can track the onboarding and off-boarding of passengers while retaining an accurate passenger count, the only way to determine if a bus is at capacity is by judgement of the bus driver. For this reason, sending the trigger will require manual input from the driver.

There are two formats in which the trigger could be sent: digitally via a mobile application or verbally over the bus radio.

## Smart Phone Application Trigger

A smart phone or tablet running a custom-built application mounted in the bus cockpit would allow the driver to simply push a 'Trigger' button in the application. The application would have the route number that the bus is currently operating pre-populated. This trigger will then be sent directly to a central system with the parameters seen in Table 3.1. The system can pull further information on this bus and route from the bus organisation API.

## 2-Way Radio Trigger

While it is potentially viable to use a smart phone/tablet application in the bus cockpit that acts as an integrated trigger, the presence of a smart phone/tablet or programmable button may not currently be in place in all buses. Also, the development of a smart phone application adds to the implementation cost.

However, most public buses are equipped with two way radios for updates on any local events such as crashes, diversions or for emergency reasons that are capable of communicating with an administrator, typically in an office or depot. The 2-way radio could be used by the driver to send the trigger back to the administrator on duty at the time at the other end of the radio.

### 3.4 Decision Making Process

Upon receiving the notification from a bus in operation of a possible route in need of extra passenger relief, a decision must be made as whether to route a reserve bus to the route or not.

There are four factors to take into consideration when making the decision.

- Percentage of route remaining
- Estimated time until next scheduled bus arrival
- Estimated time until reserve bus arrival
- Estimated reduction in passenger wait time

The decision making process by the bus organisation to decide whether or not to route a reserve bus can be entirely automated so on receiving a reserve bus trigger, a simple script can use the information provided by that bus in combination with information from the bus organisations real time passenger information API to signal the waiting reserve bus.

This process could also very easily be adapted to include an administrator approval stage before the signal is sent to the waiting reserve bus if desired by the bus organisation.

### 3.4.1 Percentage of Route Complete

Assuming the bus drivers are instructed to signal when their bus reaches capacity regardless of location, as mentioned in Section 3.3, the location of the trigger will have to be taken into account during the decision making process. This location is used to determine the percentage of the route that has been completed and thus the percentage of the route that remains for the bus that has reached capacity.

A maximum threshold can be set for what percentage of the trigger route is complete that still validates the use of a reserve bus.

For example, the bus organisation could specify $70 \%$ as the maximum percentage of the route complete, so a reserve bus would still be viable if the trigger comes from a bus that has completed under $70 \%$ of its route. If the trigger comes from a bus that has completed over $70 \%$ of its route, the reserve bus would be deemed not viable and the outcome would be to not route the reserve bus.

### 3.4.2 Estimated Time Until Next Scheduled Bus Arrival

The estimated time of arrival of the next scheduled bus for the route (from which the trigger was sent) to the trigger stop needs to be taken into consideration. Assuming the presence of a real time information system, the estimated arrival time can be taken from this existing architecture.

### 3.4.3 Estimated Time Until Reserve Bus Arrival

The estimated arrival time of the next scheduled bus will be used in a comparison with the estimated arrival time of the reserve bus, which is calculated based on the distance the reserve bus has to travel from the waiting to starting point.

The location that the reserve buses wait at play a huge role in the decision making process. The time taken for a reserve bus to travel from its waiting location to the specified start location must be taken into account. Dependent on the waiting location, the reserve bus could arrive to the specified start location much before, around the same time as or much after the next scheduled bus for that route.

The comparison between the estimated arrival times of the scheduled and reserve buses to the trigger stop is used to estimate (i) if the reserve bus will arrive before the scheduled bus and if so (ii) by how many minutes. This comparison is the one of the key deciding factors in this decision making process.

### 3.4.4 Minimum Reduction in Passenger Wait Time by Reserve Bus

If the reserve bus is estimated to arrive before the scheduled bus, the last factor to consider is the percentage reduction that the reserve bus is estimated to provide for the passengers compared to the next scheduled bus.

A minimum reduction percentage can be defined by the bus organisation, above which the reserve bus will be deemed viable to be deployed.

## Calculating the Estimated Percentage Reduction in Passenger Wait Time

The following formula is used to calculate the estimated percentage reduction in passenger wait time that the reserve bus provides in comparison to the next scheduled bus on the route.

$$
\text { Reduction }_{\text {est }}=\frac{\left(T_{\text {scheduled }}-T_{\text {reserve }}\right)}{T_{\text {schedululed }}} * 100
$$

where:

- Reduction ${ }_{\text {est }}$ is the estimated percent reduction in passenger wait time provided by the reserve bus in comparison to the next scheduled bus.
- $T_{\text {scheduled }}$ is the estimated number of minutes until the scheduled bus reaches the trigger stop.
- $T_{\text {reserve }}$ is the estimated number of minutes until the reserve bus reaches the trigger stop.


### 3.4.5 Example Decision Situations

The situations where a reserve bus is approved for operation are dependent on the thresholds set by the bus organisation for the (i) maximum percentage of the (trigger) route remaining and (ii) the minimum reduction in passenger wait time provided by the reserve bus.

There are four example situations outlined below in Table 3.2 with different values for the percentage of the route completed, the estimated time until the arrival of the scheduled and reserve bus and the estimated reduction in passenger wait time. There is an accompanying table for each example that illustrates under what thresholds a reserve bus would still be viable to dynamically dispatch, where a green square ( / ) illustrates that the use of a reserve bus is viable whereas a red square ( $\mathbf{X}$ ) illustrates the opposite. These tables provide an insight into how the bus organisation can set their own parameters for dispatching a reserve bus to suit different situations.

For example, Table 3.3 highlights the acceptable requirements for Example Situation \#1. If the bus operator has a minimum reduction for passenger wait time of above $60 \%$ $(70 \%, 80 \%$ or $90 \%)$, the reserve bus is not routed in this situation as illustrated by the red squares ( $\mathbf{X}$ ) in the relevant columns. This is because the estimated reduction in this situation is only $60 \%$, as seen in Table 3.2. Similarly, it can be seen that if the bus operator requests that the route can only trigger a reserve bus if it has completed under $10 \%$ of the route then a reserve bus is not viable in this situation as the bus has completed $20 \%$ of the route, as seen in Table 3.2.

## Examples

Table 3.2: Example Reserve Bus Trigger Situations.

|  | \% Route <br> Completed | Est. Arrival <br> Time (Schedule) | Est. Arrival <br> Time (Reserve) | Est. Reduction <br> in Passenger <br> Wait Time |
| :--- | :---: | :---: | :---: | :---: |
| Ex 1 | $20 \%$ | 30 mins | 12 mins | $60 \%$ |
| Ex 2 | $40 \%$ | 15 mins | 10 mins | $\sim 33 \%$ |
| Ex 3 | $60 \%$ | 15 mins | 10 mins | $\sim 33 \%$ |
| Ex 4 | $80 \%$ | 20 mins | 10 mins | $50 \%$ |

Table 3.3: Example 1: Evaluation for operation of reserve bus.

| Ex 1 |  | Minimum Reduction of Passenger Wait Time |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 10\% | 20\% | 30\% | 40\% | 50\% | 60\% | 70\% | 80\% | 90\% |
|  | 90\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
|  | 80\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
| Maximum | 70\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
| \% | 60\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
| of | 50\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
| Route | 40\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
| Completed | 30\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
|  | 20\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X |
|  | 10\% | X | X | X | X | X | X | X | X | X |

Table 3.4: Example 2: Evaluation for operation of reserve bus.

| Ex 2 |  | Minimum Reduction of Passenger Wait Time |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 10\% | 20\% | 30\% | 40\% | 50\% | 60\% | 70\% | 80\% | 90\% |
| $\begin{gathered} \text { Maximum } \\ \% \\ \text { of } \\ \text { Route } \\ \text { Completed } \end{gathered}$ | 90\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 80\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 70\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 60\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 50\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 40\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 30\% | X | X | X | X | X | X | X | X | X |
|  | 20\% | X | X | X | X | X | X | X | X | X |
|  | 10\% | X | X | X | X | X | X | X | X | X |

Table 3.5: Example 3: Evaluation for operation of reserve bus.

| Ex 3 |  | Minimum Reduction of Passenger Wait Time |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | 10\% | 20\% | 30\% | 40\% | 50\% | 60\% | 70\% | 80\% | 90\% |
| $\begin{gathered} \text { Maximum } \\ \% \\ \text { of } \\ \text { Route } \\ \text { Completed } \end{gathered}$ | 90\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 80\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 70\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 60\% | $\checkmark$ | $\checkmark$ | $\checkmark$ | X | X | X | X | X | X |
|  | 50\% | X | X | X | X | X | X | X | X | X |
|  | 40\% | X | X | X | X | X | X | X | X | X |
|  | 30\% | X | X | X | X | X | X | X | X | X |
|  | 20\% | X | X | X | X | X | X | X | X | X |
|  | 10\% | X | X | X | X | X | X | X | X | X |

Table 3.6: Example 4: Evaluation for operation of reserve bus.

| Ex 4 |  | Minimum Reduction of Passenger Wait Time |  |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $90 \%$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $80 \%$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\checkmark$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $70 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $60 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
| Route | $50 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
| Completed | $30 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $30 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $20 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |
|  | $10 \%$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |

## Chapter 4

## Simulation Design

### 4.1 Introduction

In this chapter, the implementation details of the interactive simulation tool built to demonstrate the system are discussed.

The aim of the simulation is to determine whether or not dynamically routing reserve buses from a waiting area to an area in need of passenger relief is beneficial to a bus provider, and if so in what circumstances is it (i) extremely beneficial, (ii) averagely beneficial and (iii) not beneficial.

### 4.2 Data Retrieval

### 4.2.1 Real Time Passenger Information API

The National Transport Authority of Ireland provide a publicly accessible Real Time Passenger Information (RTPI) API [12] that provides a REST interface to retrieve information on real time bus information, timetables and bus stops operated by Dublin Bus.

The API provides several methods of data retrieval with different parameters and results. It was utilised by the simulation tool to fetch route specific information, most importantly the longitude and latitude coordinates of each of the bus stops that make up a given route.

The GET request utilised in the simulation is the method used to retrieve details for a specific route as outlined below, where rtpiserver (Real Time Passenger Information server) is the address of the server which in this case was provided by Dublinked (data.dublinked.ie/cgi-bin/rtpi/).

| Method | http://[rtpiserver]/routeinformation?routeid=[route]\&operator=[operator]\&format=[format] |
| :---: | :---: |
| Description | Function to retrieve route details defined in the system |
| Parameters | - route - required, requested route id <br> - operator - required, operator id <br> - format - optional, valid values: xml, json; if not provided JSON format is used by default. |
| Response | List of routes and bus stops defined in the system in JSON or XML format. <br> JSON: <br> \{ |
|  | ```"errorcode" : "[Error code]", "errormessage" : "[Additional error details]", "numberofresults" : [Number of results], "route" : "[Route]", "timestamp" : "[Date/time information returned]", "results" : [{ "operator" : "[Operator name]", "origin" : "[Origin name]", "origintranslated" : "[translated origin name]", "destination" : "[Destination name]", "destinationtranslated" : "[Translated destination name]", "lastupdated" : "[Last update date/time for this record]", "stops" : [{ "stopid" : "[Stop Id]", "displaystopid" : "[Stop Identifier to display for the end user]", "shortname" : "[Stop Name]", "shortnamelocalized" : "[Translated Stop Name]", "fullname" : "[Full Name]", "fullnamelocalized" : "[Translated Full Name]", "latitude" : "[latitude]" "longitude" : "[longitude]" }, ...] }, ...] }``` |

Figure 4.1: Dublin Bus Real Time Information GET Request Example

Following the format specified in Figure 4.1, a GET request for information on the Dublin Bus route number 1 in JSON format would look as follows:
https://[rtpiserver]/routeinformation?routeid=1<br>\&operator=bac<br>\&format=json

This GET request will return an array of results which contains an array of the bus stops included in the route number 1 from start to finish. The entry for each stop contains additional information such as the stop identifier, the name of the stop, the longitude and latitude coordinates of the stop along with some additional information.

```
{
    "fullnamelocalized": "",
    "displaystopid": "381",
    "latitude": "53.324195",
    "operators": [
        {
            "routes": [
            "1",
            "47"
            ],
            "name": "bac"
        }
    ],
    "longitude": "-6.212296944",
    "stopid": "381",
    "shortnamelocalized": "",
    "shortname": "Park Avenue",
    "fullname": "Park Avenue"
}
```

Listing 4.1: Stop information of first stop for Dublin Bus route number 1 from the route information GET request results.

When simulating the reserve bus, the directions and route for the reserve bus from its waiting position to its starting position must be calculated. Also, even though the coordinates of each of the stops that make up a bus route can be retrieved, the route
between each of the stops is not provided. Thus the route between each of the stops must be calculated. The precise latitude and longitude coordinates of the location of each bus stop is thus incredibly valuable information as we can use these coordinates to calculate our routes using Graphhopper.

### 4.2.2 Directions via Graphhopper API

The Graphhopper Directions API [15] is a fast directions API with worldwide data from OpenStreetMap. The API can provide routing between two locations (with several required stops in between start and destination nodes) with useful attributes such as turn-by-turn instructions, total distance and total estimated time. The API can also take the vehicle type as a parameter in the GET request to restrict the directions to routes that are only acceptable for certain vehicle types, like in this case a bus.

The reason that Graphhopper was chosen as the source of directions rather than an alternate directions API, such as the Google Maps API, was due to their additional route optimisation API that specialises in solving vehicle routing problems that would be potentially useful for expanding work on this simulation tool. The route optimisation API considers a range of business constraints such as driver breaks and multiple vehicles.

The GET request to the Graphhopper API follows the following format:

```
https://graphhopper.com/api/1/route
    ?point={coordinate_points}&vehicle={vehicle_type}
        &key={api-key}&type={format}&points_encoded={boolean};
```

where the parameters are:

- point - A point is made up of a latitude and longitude value: (latitude,longitude). Multiple points may be specified and at least two points must be specified.
- vehicle - The vehicle for which the route should be calculated. The type of vehicle puts restrictions on roads based on vehicle height, width, weight, number of axes, etc.
- type - The resulting format of the route (json/gsx).
- points_encoded - If false the coordinates in point are returned as array using the order [lon,lat,elevation] for every point. If true the coordinates will be encoded as string leading to less bandwith usage which requires extra handling to decode.

The Graphhopper API will then return the result route that starts at the first point and ends at the last, making sure any other specified points are also visited along the way. The result is a JSON file with turn-by-turn directions, the distance for each movement and the precise coordinates of each of the points along the route. These values can then be used to route the reserve buses from their waiting point to their designated start point in the fastest manner possible.

### 4.3 Front End Libraries

The simulation tool built utilises two front end libraries, Leaflet and D3.js. These two libraries are used to provide a visual animation of the bus simulations on an interactive map.

### 4.3.1 Leaflet

Leaflet is an open source JavaScript library for building web applications with interactive maps [21]. It is designed with simplicity, performance and usability in mind meaning it puts little stress on the browser presenting the map.

The decision to use Leaflet was driven by a number of factors. First of all, it is a lightweight library with a plentiful amount of associated plug-ins allowing for flexibility in development and layering on top of the map element. Also, Leaflet can use map data from OpenStreetMap which eases the integration of the Graphhopper API for precise routing.

With use of the plug-in Leaflet.D3SvgOverlay, D3.js elements can be easily drawn on to the Leaflet map. This extends the ability to layer customised D3 visual elements on top of the interactive map.

### 4.3.2 OpenStreetMap

The implementation of Leaflet uses OpenStreetMap as the source of the map. OpenStreetMap is a collaborative mapping project to create a free editable map of the world [29]. It is built and maintained by a community of mappers providing vast amounts of data for roads, railways, buildings, shops, etc. across the world. OpenStreetMap is open data and is free to use.

### 4.3.3 D3.js

D3.js is a JavaScript library for producing data visualisations with HTML, SVG and CSS [5]. D3 was chosen in combination with Leaflet, and the Leaflet plug-in D3SvgOverlay, to aid in creating a visual, data-driven simulation of the scheduled buses as they traverse their route, pick up \& drop off passengers and to aid in the visual demonstration of the routing and operation of the reserve bus.

The bus route information and routing data that is retrieved can be visually plotted onto the Leaflet map. D3 is used to draw all of the overlaying objects on the Leaflet map, including:

- The bus stops \& number of waiting passengers - static object
- The buses (scheduled \& reserve) - moving objects
- The progress path of the buses along their routes - moving object


Figure 4.2: Simulation Objects Created With D3

Figure 4.2(a) shows the visual representation of four bus stops (in blue) and the number
of passengers waiting at those boss stops（in red）．Figure 4.2 （b）shows the path（grey） taken by the scheduled bus（yellow）as it manoeuvres its route．

## 4．4 Running The Simulation

Each simulation consists of several stages／aspects：first scheduled bus operating，second scheduled bus operating and the reserve bus operating（if triggered）．If a reserve bus is triggered，the point where the bus is triggered is recorded and highlighted in red on the map．The time taken for the next scheduled bus on that route to arrive at the trigger point and the time for the reserve bus to arrive at the trigger point is then recorded． These times are then displayed on the information section of the web application for comparison．

Three front－end input fields were created in the web application to allow the user running the simulation to enter a Dublin Bus Route，the volume of waiting passengers and the volume of traffic（see Figure 4．3）．The user can also select the starting point of the reserve bus by placing a marker on the interactive map（see Figure 4．4）．

## Dublin Bus Route：

enter route，eg：1，ani
List of possible routes
can be found here．

## Level of Traffic：


\＃of Waiting Passengers：
ษ ダ ザษ

Figure 4．3：Example of User Input：Bus Route，Passenger Volume and Traffic Volume


Figure 4.4: Example of User Input: Selecting Reserve Bus Starting Point

### 4.4.1 Random Passenger Generation

The web application has an input field for the volume of passengers that will be waiting at the various bus stops for the specified route (see Figure 4.3). This input value will then set the range of the number of passengers to be randomly generated at each stop.

- Low - 0 to 2 passengers per stop.
- Medium - 3 to 6 passengers per stop.
- High - 7 to 11 passengers per stop.

The number of passengers waiting per stop is calculated prior to the first scheduled bus departing. Each stop will have a number of passengers within the selected range waiting. This number is displayed on the interactive map beside the respective bus stop.

When a bus arrives at a stop, a random number of passengers will offload and then each waiting passenger will board the bus if there is space. The application keeps track of the buses current on-board passengers. As a bus departs from a bus stop, the updated number of passengers waiting at that stop will be displayed (zero if all passengers have boarded successfully).

### 4.4.2 Traffic Weight

The final input field in the web application is the selection of the volume of traffic in the simulation. Similar to the passenger volume, there are three severities of traffic. The traffic weight acts as a multiple that is factored into the calculation of the bus travel duration.

- Low - traffic weight is 1 (normal).
- Medium - traffic weight is 1.2 .
- High - traffic weight is 1.5 .


### 4.4.3 Bus Travel Speed

To reduce the time cost for running simulations, the simulation is not run in real time. Thus, a formula is needed to calculate the duration of the bus travel speed between points in the simulation.

$$
t=\text { dist } * \text { speed } * \text { traffic }
$$

where

- $\mathbf{t}$ is the duration of bus travel speed
- dist is the distance to be travelled
- speed is the amount of milliseconds required for a bus to travel one kilometre in the simulation
- traffic is the traffic weight


### 4.4.4 Second Scheduled Bus

A second scheduled bus operates on the same route after a certain time interval. This is to represent the bus that follows the first scheduled bus by approximately twenty minutes in a real-time timetable scenario.

### 4.4.5 Reserve Bus Trigger

During the simulation, if the first scheduled bus reaches capacity it will send a signal out stating that it has reached capacity and include the details of (i) what stop it is currently at and (ii) the route it is operating on. From this point, two timers are started: timer $_{\text {res }}$ and timer sch, a timer each for the reserve bus and the second scheduled bus respectively.

The reserve bus then receives its calculated route from its waiting point to the stop where the signal came from (via Graphhopper API). The reserve bus object follows this
calculated route, taking the amount of time determined by the travel speed algorithm (see Section 4.4.3). When the reserve bus reaches the starting point (where the signal came from), timer $_{r e s}$ is stopped. Similarly, when the second scheduled bus reaches the starting point, timer sch is stopped. These timers can now be compared and results can be gathered.

## Chapter 5

## Evaluation

### 5.0.1 Approach

In order to evaluate the implementation of the proposed reserve bus method, a range of simulations were performed. Five Dublin Bus routes were simulated multiple times, running the simulation with a combination of low, medium and high volumes of passengers and traffic, with the reserve bus starting from seven different locations across Dublin.

## Simulated Routes

With the aim of the simulation being to determine in what situations, if any, a dynamically routed reserve bus would reduce passenger wait time in comparison to a normal scheduled bus, routes of different lengths and locations are simulated. Five Dublin Bus routes were chosen based on the geographic position of the stops in the route in an attempt to provide a diverse number of locations where a reserve bus may be routed
to across Dublin. These routes are:

- 1 - departing from Sandymount, ending in Santry.
- 44-departing from Enniskerry, ending in DCU (Whitehall).
- 45a - departing from Kilmacanogue, ending in Dun Laoghaire.
- 66 - departing from Maynooth, ending in Merrion Square.
- 77a - departing from Citywest, ending in Ringsend.


Figure 5.1: The Five Simulated Routes

## Reserve Bus Starting Locations

In section 3.3, the idea of using bus depots for the reserve bus starting locations is proposed. Following this suggestion, the simulations performed utilised the location of seven Dublin Bus depots [11] listed below. The letter in brackets beside the depot location is the letter used to represent the specific depots in the results figures found in this chapter.

- Clontarf (A)
- Conyngham Road (B)
- Donnybrook (C)
- Harristown (D)
- Phibsborough (E)
- Ringsend (F)
- Summerhill (G)


Figure 5.2: Dublin Bus Depots for Reserve Bus Starting Points.

Thus, each route will be simulated with a reserve bus starting from each of the seven bus depot locations for each combination of the passenger and traffic volume parameters. The simulated reserve bus will wait at its designated waiting point until it receives the trigger from a bus that reaches full capacity.

## Key Values

There are some key values to be recorded for each simulation to be evaluated.

- The starting location of the reserve bus
- The route number
- The distance from the trigger stop to the starting location of the reserve bus
- The time taken for the second scheduled bus to arrive at the trigger stop
- The time taken for the reserve bus to arrive at the trigger stop


### 5.0.2 Results

Tables and graphs have been used to illustrate the results in a clear fashion where appropriate. Due to the number of waiting passengers that are randomly generated, within the range permitted by the volume parameter, varying slightly per simulation, each route simulation was run ten times and then the average time taken and the average distance from the trigger stop to the starting location of the reserve bus was recorded.

The graphs provide a visual representation of the passenger wait time (in minutes) from the point of the reserve bus trigger to the (i) scheduled bus arrival and (ii) the reserve bus arrival for each of the seven reserve bus starting points. The average distance, as the crow flies, from the starting points to the trigger point is displayed beside the location symbol in the graph information.

Firstly, running simulations for each of the bus routes 1, 44, 45a, 66 and 77 a with low and medium passenger volume results in no reserve bus being requested. This result is the expected outcome as the simulation application tool was built to be able to simulate any bus route under normal circumstances. So, in any route simulation with low or medium passenger volume, the buses performed their routes from start to finish without reaching capacity and requiring a reserve bus. When a simulation is run with a high volume of passengers, the reserve bus is always triggered regardless of the route.

## Traffic Volume

In the simulation implementation, the volume of traffic can be selected as low, medium or high as previously described. This volume of low, medium or high correlates to a numerical value, 1.0, 1.2 or 1.5 respectively. This acts as a factor in the bus travel time duration calculation as described in Section 4.4.3.

The volume of traffic used in this simulation does not significantly alter the time difference between the scheduled bus and the reserve bus. This is because each bus in the simulation uses the same formula to determine the travel duration of its route. When the traffic volume was set to high, both the reserve bus and the schedule bus travelled 1.5 times slower along their route. Thus, there is no significant change in the reduction in passenger wait time when comparing the two buses.

In hindsight, this was an oversight during the design process. For future work, if the traffic volume could be changed during the simulation (i) for one of the buses or (ii) depending on the area, then this could perhaps produce some more insightful results.

## Passenger Wait Time Reduction

The "passenger wait time" in the simulation is the amount of time a passenger waits at a bus stop after being refused entry to a bus (due to capacity) until he boards a bus at that stop (for the same route). With this definition, the time taken for both the second scheduled bus and the reserve to travel to the trigger stop can be compared to determine which bus resulted in a lower passenger wait time.

Since the scheduled bus does not vary its starting position like the reserve bus, its passenger wait times remain fairly consistent across simulations. However, different starting positions for the reserve bus drastically effect its passenger wait times due to the variation in distance the reserve bus must travel from the starting position to the trigger point.

Figure 5.3 displays the results for a simulation of route 1 with a heavy volume of passengers and a medium volume of traffic. It can be seen that the reserve bus reaches the trigger stop faster than the next scheduled bus when it starts in any of the locations except for Harristown (D). The reserve bus provides significant reductions in passenger wait time when starting in Phibsborough (E) of $61 \%$, Summerhill (G) of $69 \%$ and Ringsend (F) of $83 \%$ while it provides a reduction of $27 \%$ when starting in Clontarf (A), $44 \%$ when started in Donnybrook (C) and $47 \%$ when started in Conyngham Road (B).

Figure 5.7 displays the results for a simulation of route 77 a with a heavy volume of passengers and a medium volume of traffic. It illustrates results that differ quite significantly from the simulation results of route 1 . In this simulation, the reserve bus arrives after the next scheduled bus for every reserve bus starting location. This means that any passengers who were left waiting at a stop due to a full capacity bus have been picked up by the second scheduled bus before the reserve bus even arrives. In this case, the scheduled bus results in a lower passenger wait time by at least $33 \%$ (at Conyngham Road(B)) and up to $65 \%$ (at Harristown (D)).

Similar to the results of the route 77a simulation, the simulation for routes 44, 45a and 66 all resulted in disappointing results for the reserve bus performance, as seen in Figure 5.4, 5.5 and 5.6 respectively.

Route 1 (medium traffic): Passenger wait time from reserve bus trigger to bus arrival.


Figure 5.3: Results: Route 1

Route 44 (medium traffic): Passenger wait time from reserve bus trigger to bus arrival.


Figure 5.4: Results: Route 44

Route 45a (medium traffic): Passenger wait time from reserve bus trigger to bus arrival.


Figure 5.5: Results: Route 45a

Route 66 (medium traffic): Passenger wait time from reserve bus trigger to bus arrival.


Figure 5.6: Results: Route 66

Route 77a (medium traffic): Passenger wait time from reserve bus trigger to bus arrival.


Figure 5.7: Results: Route 77a

### 5.0.3 Evaluation

Firstly, from the results gathered from the simulations shown in figures in section 5.0.2, it can be seen that the scheduled bus maintains a consistent value in the range of 0.48 0.52 for the average passenger wait time. This is an expected result and is because the scheduled bus is departing from the same starting point in each simulation at the same time interval after the previously scheduled bus. The small variation is due to the amount of time added to the trip by passengers boarding and departing, which depends on the number of passengers which is randomly generated within a range.

The simulations performed provided an insight into the the heavy influence that the required distance the reserve bus must travel has on the reduction in passenger wait time.

- As described in Section 4.4.3, the travel time for the simulated buses is calculated by $t=$ dist $*$ speed $*$ traffic. The speed and traffic factor remain the same for both the scheduled bus and the reserve bus. Thus, the only factor that varies is the distance. Due to the reserve bus starting at seven different locations for each route simulation, the reserve bus experiences a significant fluctuation in required distance to be travelled within the simulation. A larger distance to be travelled results in a larger travel time.
- A formula to estimate the expected passenger wait time for passengers waiting for a reserve bus can be deduced by analysing the relationship between the distance from the reserve bus starting location to the trigger stop and the passenger wait time from the results.

$$
T=\text { distance } * W_{k m}
$$

where

- $T$ is the expected passenger wait time until reserve bus arrives (in mins).
- distance is distance from the trigger point to the reserve bus starting location (in kilometres)
- $W_{k m}$ is the passenger wait time per kilometre of distance between reserve bus starting point and trigger stop (in mins).

The results of the simulations can be used to determine the average $W_{k m}$, see Table 5.1.

$$
W_{k m}=\frac{P W T}{D}
$$

where

- PWT is the total passenger wait time (in mins).
- $D$ is the distance from the trigger point to the reserve bus starting location (in kilometres).

Table 5.1: Calculating the average $W_{k m}$.

|  | $W_{k m}$ for each Reserve Bus Starting Points |  |  |  |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | A | B | C | D | E | F | G | Average $W_{k m}$ |
| Route 1 | 0.10 | 0.08 | 0.09 | 0.09 | 0.08 | 0.08 | 0.09 | 0.09 |
| Route 44 | 0.10 | 0.10 | 0.10 | 0.10 | 0.09 | 0.10 | 0.09 | 0.10 |
| Route 45a | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 | 0.10 |
| Route 66 | 0.09 | 0.10 | 0.10 | 0.11 | 0.10 | 0.10 | 0.10 | 0.10 |
| Route 77a | 0.08 | 0.08 | 0.09 | 0.08 | 0.09 | 0.09 | 0.09 | 0.09 |
| Overall Average $W_{k m}$ |  |  |  |  |  |  |  | 0.096 |

- With the overall average $W_{k m}$ for the simulations calculated as 0.096 , we can conclude that for each kilometre travelled by the reserve bus in the simulation, the passenger wait time will be increased by approximately 0.096 .
- A reserve bus that has to travel approximately 5 kilometres to the trigger stop will result in passengers waiting approximately $0.48 \mathrm{mins}(5 * 0.096)$. However, with the scheduled bus averaging a passenger wait time of 0.50 minutes (in the simulation), the reserve bus is only providing a $4 \%$ reduction in passenger wait time.
- A $20 \%$ reduction in passenger wait time would require the reserve bus to be within approximately 4.16 kilometres of the trigger point, resulting in passengers waiting for approximately 0.40 minutes (in the simulation), a $40 \%$ reduction would require the reserve bus to be within approximately 3.125 kilometres, etc.

These results show that dynamically routing the reserve bus can reduce passenger wait time in the area in need of extra passenger capacity by a significant percentage ( $20 \%+$ ) when it is waiting to be deployed from an area less than approximately 4 kilometres away. When the reserve bus is waiting more than 5 kilometres away from the trigger
point requesting the extra passenger capacity, the reserve bus does not provide any reduction in passenger wait time.

## Chapter 6

## Conclusion

In this dissertation, I proposed a method of dynamically dispatching and routing reserve buses to bus routes in need of extra passenger capacity in order to reduce passenger wait times. This method is primarily focused on reducing passenger wait times during events where a scheduled bus reaches capacity and, as a result, is forced to refused entry to passengers waiting at future stops along this route. The dynamic routing of the reserve bus to the area of the route in need functions during unexpected large influxes of passengers that may fill a normal scheduled bus to capacity.

A simulation web application tool was developed to run route simulations with real bus route info from Dublin Bus. The web application plots the stops from any real Dublin Bus route onto an interactive map and simulates a bus travelling along the route, picking up and dropping off passengers. In the event that there's a large, unexpected influx of passengers and the bus reaches capacity, it signals a trigger for a reserve bus. The reserve bus is then simulated travelling to the start point and operating along that route. The travel time for the reserve bus to the trigger point is recorded, as is the travel time for the next scheduled bus to that point. These values can then be compared and, along with additional information from the simulation, a decision can
be made to whether or not the reserve bus was beneficial in that scenario.

Results gathered from the simulations ran show that the use of a dynamically reserve bus can reduce passenger wait time, significantly so in some situations. However, the percentage reduction in passenger wait time is proportional to the distance the reserve bus must travel to reach the trigger point. The reserve bus must be within approximately 5 kilometres just to perform as well as the scheduled bus. To reduce passenger wait times by over $20 \%$ the bus must be within approximately 4 kilometres of the trigger point. Although passenger wait time may be reduced, these distances are quite limiting for an urban transportation bus network.

## Chapter 7

## Further Work

This dissertation outlines a method to potentially reduce bus passenger wait times during times of high passenger influx on the bus network. However, there are further questions to answer and potential avenues for further work on this proposal.

### 7.1 Capacity Prediction

One of the main issues with the currently proposed method is the short radius that the reserve bus must be within in order to reduce passenger wait time. This is because the reserve bus may spend significant time travelling from its waiting point to the area in need of the bus. If, instead of the normal bus signalling when it reaches $100 \%$ capacity, the bus could signal when it reaches $80 \%$ or $90 \%$ capacity, the model could be slightly adjusted to attempt to predict if the bus will reach full capacity and if so a reserve bus could be routed into operation before the first bus even reaches $100 \%$ capacity. The reserve bus could also be routed a few stops ahead of where the current bus is in a pre-emptive action.

### 7.2 Passenger Crowd Information Retrieval/Estimation

One of the difficulties with this project was determining when the reserve bus would be dispatched into operation. As the proposal stands currently, a manual trigger is required from the driver of a bus that reaches full capacity. As mentioned in Section 7.1, if the bus capacity could be predicted this would allow for more time for the reserve bus to be routed to the area in need and it could be done before the first bus has even reached capacity.

Similarly, if more information could be obtained on the crowds of passengers waiting at bus stops, problematic stops with unusually large waiting passenger numbers could be flagged pro-actively. Computer vision could be performed on the video source from a bus' on-board camera to measure the crowd size. Similarly, the volume of mobile phones present at bus stops could be measured to estimate the crowd size. Crowd sourcing information from passengers is another method which could be performed via the bus organisations mobile phone app or something along the lines of a Twitter or Facebook chat bot.

## Bibliography

[1] J. J. Bartholdi and D. D. Eisenstein. A self-coordinating bus route to resist bus bunching. Transportation Research Part B: Methodological, 46(4):481-491, 2012.
[2] R. Bellman. On a routing problem. Quarterly of applied mathematics, 16(1):87-90, 1958.
[3] J. Brandão. A tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research, 157(3):552-564, 2004.
[4] A. Ceder and N. H. Wilson. Bus network design. Transportation Research Part B: Methodological, 20(4):331-344, 1986.
[5] D3. D3 js.
[6] C. Daganzo. Schedule instability and control. Daganzo, CF Fundamentals of Transportation and Transportation Operations, Elsevier, New York, NY, pages 304-309, 1997.
[7] C. F. Daganzo. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. Transportation Research Part B: Methodological,

43(10):913-921, 2009.
[8] C. F. Daganzo and J. Pilachowski. Reducing bunching with bus-to-bus cooperation. Transportation Research Part B: Methodological, 45(1):267-277, 2011.
[9] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. Management science, 6(1):80-91, 1959.
[10] Dublin Bus. Dublin bus passenger numbers 2015. https: //www.dublinbus.ie/News-Centre/General-News-Archive/ Dublin-Bus-reports-passenger-numbers-reached-122-million-in-2015/, 2016. Accessed: 2017-05-6.
[11] Dublin Bus. Dublin bus fleet information. http://www.dublinbus.ie/About-Us/ Dublin-Bus-Fleet/, 2017. Accessed: 2017-05-6.
[12] Dublin Bus. Dublin bus real time passenger information. https://www. dublinbus.ie/RTPI/, 2017. Accessed: 2017-05-6.
[13] X. J. Eberlein, N. H. Wilson, and D. Bernstein. The holding problem with realtime information available. Transportation science, 35(1):1-18, 2001.
[14] B. L. Golden, S. Raghavan, and E. A. Wasil. The vehicle routing problem: latest advances and new challenges, volume 43. Springer Science \& Business Media, 2008.
[15] Graphhopper. Graphhopper api.
[16] C. Groër, B. Golden, and E. Wasil. A parallel algorithm for the vehicle routing problem. INFORMS Journal on Computing, 23(2):315-330, 2011.
[17] M. D. Hickman. An analytic stochastic model for the transit vehicle holding problem. Transportation Science, 35(3):215-237, 2001.
[18] E. Lalla-Ruiz, C. Expósito-Izquierdo, S. Taheripour, and S. Voß. An improved formulation for the multi-depot open vehicle routing problem. OR spectrum, 38(1):175-187, 2016.
[19] G. Laporte. The vehicle routing problem: An overview of exact and approximate algorithms. European journal of operational research, 59(3):345-358, 1992.
[20] G. Laporte and Y. Nobert. Exact algorithms for the vehicle routing problem. North-Holland Mathematics Studies, 132:147-184, 1987.
[21] Leaflet. Leaflet js.
[22] A. N. Letchford, J. Lysgaard, and R. W. Eglese. A branch-and-cut algorithm for the capacitated open vehicle routing problem. Journal of the Operational Research Society, 58(12):1642-1651, 2007.
[23] F. Li, B. Golden, and E. Wasil. The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers \& Operations Research, 34(10):2918-2930, 2007.
[24] R. Liu, Z. Jiang, and N. Geng. A hybrid genetic algorithm for the multi-depot open vehicle routing problem. OR spectrum, 36(2):401-421, 2014.
[25] T. L. Magnanti. Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. Networks, 11(2):179-213, 1981.
[26] MTA. Mta real time information.
[27] M. M. Nesheli, A. Ceder, and V. A. Gonzalez. Real-time public-transport operational tactics using synchronized transfers to eliminate vehicle bunching. IEEE Transactions on Intelligent Transportation Systems, 17(11):3220-3229, 2016.
[28] G. F. Newell and R. B. Potts. Maintaining a bus schedule. In Australian Road Research Board (ARRB) Conference, 2nd, 1964, Melbourne, volume 2, 1964.
[29] OpenStreetMap. Openstreetmap.
[30] S. Raff. Routing and scheduling of vehicles and crews: The state of the art. Computers \& Operations Research, 10(2):6369117149195-67115147193211, 1983.
[31] C. Rego. Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. Parallel Computing, 27(3):201-222, 2001.
[32] D. Sariklis and S. Powell. A heuristic method for the open vehicle routing problem. Journal of the Operational Research Society, 51(5):564-573, 2000.
[33] C. Tarantilis and C. Kiranoudis. Distribution of fresh meat. Journal of Food Engineering, 51(1):85-91, 2002.
[34] Transport for London. Transport for london real time information. https://tfl. gov.uk/modes/buses/live-bus-arrivals, 2017. Accessed: 2017-05-6.

