Factors Influencing Bus Network Design

by

Zhen Xiang Kenneth Loh

B.B.M., B.S., Singapore Management University, 2011

Submitted to the Department of Civil and Environmental Engineering in partial fulfilment of the requirements for the degree of

Master of Science in Transportation

at the

Massachusetts Institute of Technology

February 2014

© 2014 Massachusetts Institute of Technology. All rights reserved

Signature of Author .................................................................

Department of Civil and Environmental Engineering

January 17, 2014

Certified by .................................................................

Cynthia Barnhart

Associate Dean and Ford Professor of Engineering

Thesis Supervisor

Accepted by .................................................................

Heidi M. Nept

Chair, Departmental Committee for Graduate Students
Factors Influencing Bus Network Design

by

Zhen Xiang Kenneth Loh

Submitted to the Department of Civil and Environmental Engineering
on January 17, 2014 in partial fulfilment of the
requirements for the degree of
Master of Science in Transportation

Abstract

Bus network design and frequency setting, the highest level subproblems in the bus planning process, have long-term impacts on bus network performance. Improving network performance not only improves the attractiveness of public transport and thus ridership, but cost-effectiveness as well because public transport experiences increasing returns to scale. In practice, solution approaches rely heavily on the experience and intuition of human planners, possibly guided by solutions obtained through optimization techniques. Optimization is not applied in isolation due to problem complexity and computational intractability, which makes exact solutions for areas larger than a small neighbourhood difficult to compute.

In this thesis, we first review some recent proposals to solve the bus network design and frequency setting problem using optimization methods. We solve a simplified version of the problem on a small network to demonstrate the feasibility of a decomposition approach in which we generate routes algorithmically and frequencies using optimization. Next, we propose a more sophisticated methodology to examine the impacts on network performance of various design criteria, such as route length and number of routes. We describe our implementation of a parameterized route generation algorithm, generate a variety of route networks, and then perform trip assignments using origin-destination data from a major city. We then determine the performance of these networks by comparing total travel time, waiting time, and number of transfers required over different networks and on a benchmark (real-world) network.

We found that average route length and total network length are the most important criteria for determining network performance. We also found that in the generated networks, reducing total travel time came at the cost of increasing the average number of transfers per trip.

Thesis Supervisor: Cynthia Barnhart
Title: Associate Dean and Ford Professor of Engineering
Acknowledgements

First and foremost, I would like to thank my thesis advisor Professor Cynthia Barnhart for her incredible support, guidance and patience. Without a doubt, it is the opportunity to work with and learn from you that gives my MIT degree its value and I am extremely grateful for the knowledge and inspiration you have shared with me.

I would like to thank my academic advisor Professor Amedeo Odoni for helping me ease into graduate student life at MIT, guidance in deciding on a thesis topic and advisor and most importantly, the advice not to walk out into the Boston winter with wet hair.

I would like to thank Professor Byung Joon Park, who taught me at the Singapore Management University, for inspiring me to major in Operations Management (leading to my interest in transportation) and showing me the “dark side of business” with a wicked wit, sense of humour and piercing insight.

I would like to thank the Land Transport Authority of Singapore for providing the scholarship to pursue this degree at MIT and supporting my research with data access.

I would like to thank Krishna, classmate of many transportation courses, colleague, virtual co-author of this work and friend. Your willingness to discuss research with me was a major source of ideas and of great help to this research but your friendship is valued over all else. I would also like to thank all my fellow graduate students in the MST programme for the excellent company and memories. Ivo, Julius, Siavash, Brian, Brendan, Wei and the rest of the CEE soccer team, it is always an honour and my pleasure to share the football field with you guys.

I would like to thank the bros, Karthick, Sheng Rong, Siah Hong and Marcus, for being the best friends one could ask for. I would also like to thank Inez for her unwavering support, encouragement and patience.

Finally, I would like to thank my parents for bringing me up well and giving me the opportunity and support to get to where I am today. I am also incredibly lucky to have amazing sisters Lorraine and Porky, and cousin (more like older brother) Jia Yong as family.
Table of Contents

List of Figures ................................................................................................................. 9
List of Tables ....................................................................................................................... 10
1. Introduction ....................................................................................................................... 12
  1.1 Bus Planning .................................................................................................................... 13
  1.2 Bus Network Design and Frequency Setting Problem ....................................................... 15
  1.3 Model Formulation Review ............................................................................................... 16
    1.3.1 Wan and Lo ............................................................................................................... 16
    1.3.2 Fan and Machemehl ................................................................................................. 17
    1.3.3 Shimamoto, Schmöcker and Kurauchi ....................................................................... 18
    1.3.4 Summary .................................................................................................................. 18
  1.4 Optimization Methods Review ......................................................................................... 19
    1.4.1 Wan and Lo ............................................................................................................... 19
    1.4.2 Fan and Machemehl ................................................................................................. 19
    1.4.3 Shimamoto, Schmöcker and Kurauchi ....................................................................... 21
  1.5 Research Objectives ....................................................................................................... 22
  1.6 Outline of Thesis ............................................................................................................ 22
2. Preliminary Case Study ...................................................................................................... 25
  2.1 Problem Input and Formulation ....................................................................................... 25
  2.2 Route Generation ............................................................................................................. 26
  2.3 Route Generation Algorithm ............................................................................................ 27
  2.4 Bus Route Network Graph ............................................................................................... 28
  2.5 Trip Assignment ............................................................................................................... 30
  2.6 Frequency Setting ............................................................................................................ 31
  2.7 Implementation ............................................................................................................... 31
  2.8 Road Network of Case Study .......................................................................................... 31
  2.9 Results ............................................................................................................................ 34
  2.10 Computational Performance .......................................................................................... 36
  2.11 Conclusion ..................................................................................................................... 37
3. Proposed Methodology ..................................................................................................... 39
  3.1 Design Criteria ................................................................................................................ 39
    3.1.1 Route Overlap ......................................................................................................... 39
3.1.2 Route Length.............................................................................................................................. 40
3.1.3 Network Connectivity ................................................................................................................ 40
3.2 Network Design and Frequency Setting ....................................................................................... 41
3.3 Trip Assignment ............................................................................................................................ 41
3.4 Order of Investigation ................................................................................................................... 41
4. Route Network Design .................................................................................................................... 44
  4.1 Single Route Generation ................................................................................................................ 45
    4.1.1 Required Inputs and Preprocessing ...................................................................................... 45
    4.1.2 Selection of Node Pair .......................................................................................................... 46
    4.1.3 Generation and Selection of Route Skeleton to be Expanded ............................................ 47
    4.1.4 Route Expansion ................................................................................................................ 47
    4.1.5 Cycle Removal .................................................................................................................. 48
  4.2 Network Generation by Repeated Single Route Generation ........................................................ 49
  4.3 Network Generation by Set Covering .......................................................................................... 50
    4.3.1 Modification of the SRG Algorithm ..................................................................................... 50
    4.3.2 Optimization Problem Formulation ...................................................................................... 51
    4.3.3 Formulation 1 .................................................................................................................. 51
    4.3.4 Formulation 2 .................................................................................................................. 53
  4.4 Frequency Setting .......................................................................................................................... 53
5. Trip Assignment .................................................................................................................................... 56
6. Data Sources ........................................................................................................................................... 59
  6.1 Bus Stop Data ................................................................................................................................ 60
  6.2 Road Network Data ...................................................................................................................... 60
  6.3 Origin-Destination Matrices ......................................................................................................... 61
  6.4 Existing Singapore Bus Network ................................................................................................ 62
7. Results and Discussion ....................................................................................................................... 64
8. Conclusions and Future Directions .................................................................................................. 72
Appendices .................................................................................................................................................. 75
References ................................................................................................................................................... 78
List of Figures

1-1. Subproblems of the bus planning process ................................................................. 14
2-1. Diagram of generalized model of network ............................................................... 26
2-2. Example of route network graph with two routes ................................................ 29
2-3. Diagram of road network for a 2 x 2 block neighbourhood .................................. 32
2-4. Road graph for the 2 x 2 block neighbourhood ..................................................... 33
2-5. Graph of total travel time v. Number of routes in route network for 2 x 2 block neighbourhood ...... 34
2-6. Best performing network ......................................................................................... 35
2-7. Graph of total travel time v. Number of routes in route network for 2 x 4 block neighbourhood ...... 36
3-1. Simple examples of a destination-oriented network and a direction-oriented network ............. 40
List of Tables

7-1. Results for Single Design Criterion Networks ................................................................. 66
7-2. Results for Dual Criteria Networks .................................................................................. 67
7-3. Results for Networks Generated by Set Covering ............................................................ 69
1. Introduction

A well-developed public transportation system in cities is the key to achieving mobility for urban citizens in an environmentally sustainable fashion. In addition to its primary benefit of improving mobility, public transportation is seen to have many secondary benefits, including reducing road congestion, air pollution and fuel consumption by competing with car use; and improving social equity by offering mobility to those (such as children or the physically handicapped) unable to use or afford a car. The two major public transportation modes of bus and rail compete with the automobile, with the primary trade-off being between mobility and cost.

Automobiles usually offer the convenience of availability and the ability to get the traveller door-to-door, while public transportation typically requires the traveller to get to a stop/station on his own and then wait to use the public transportation system. Automobile use, however, has a high financial fixed cost to the user and inflicts negative externalities on other members of society through noise, air pollution and inefficient consumption of natural resources. The urban fabric may also be of a form that encourages automobile use, as is the case in many suburban areas of the United States. Thus, to compete for ridership from car-owners, public transportation has to offer comparable levels of mobility and convenience as that of automobiles, which can be extremely costly to the public transport operator, especially without sufficient ridership.

Another important distinguishing characteristic between public and private transportation is the effectiveness to volume (or ridership) response, also referred to as returns to scale. As car volumes increase, unless road capacity is increased by building new or expanding existing roadways and highways, congestion builds up, reducing traffic flow speeds on roads and affecting the travel experience for car users. Even when road capacity is increased, the improvement results in more traffic volume, which might negate some of the travel time improvements [8]. Public transportation however, experiences increasing returns to scale, due to network effects and cost-effectiveness. As the public transportation network increases in size, it becomes more attractive to travellers because its scope of service increases. Furthermore, as ridership increases, the high fixed costs of infrastructure can be spread out amongst a larger number of customers, reducing the average cost each traveller pays to use the service.

Buses are the main mode of public transportation in many cities due to their low capital costs and flexibility in operation. The relatively lower capital cost of buses is because they do not require much dedicated infrastructure beyond terminals and stops and can use most existing roads. Trains, in contrast, require rails and dedicated right-of-way which takes up land space in addition to requiring terminals and
stops. The costs of rail systems can accelerate rapidly as well if the rails are to be located underground, also known as a ‘subway’ or ‘metro’ system. Buses have greater operational flexibility because routes can be relatively easily redrawn by getting buses to follow a different path on the road network while rerouting a rail system would involve clearing land or digging the path for new rails. Similarly, train stations tend to be larger and costlier projects than bus stops.

Similar to other major cities, the majority of Singapore’s public transportation trips are on bus, rather than rail. In 2012, buses had an average daily ridership of 3,481,000 passenger-trips while the Mass Rapid Transit (heavy subway with several elevated sections above ground) system had average daily ridership of 2,525,000 passenger-trips.¹ This will increase as Singapore’s population is projected to continue to grow due to immigration, reaching 6.5 to 6.9 million by 2030.² To accommodate the additional load on the transportation system, Singapore’s public transport agency and regulator, the Land Transport Authority (LTA) has set a target of increasing public transport’s mode share to 75% of all trips during the morning and evening peak periods.³

Achieving these goals will require improving the level of service of buses, reducing wait times and travel time variability to improve the mode’s reliability. To this end, the Bus Service Enhancement Programme (BSEP) was introduced in 2012. The thrust of this programme is to increase the bus fleet by 800 buses (about 20% of the current bus fleet size) and increase service availability by increasing frequency on existing routes as well as introducing new bus routes. In light of this development, to improve the effectiveness and cost-efficiency of this programme, it is extremely relevant to examine the problem of bus network design.

1.1 Bus Planning

Ceder and Wilson's [19] framework for bus planning divides the process into 5 parts, as summarized by their table below:

<table>
<thead>
<tr>
<th>Independent Inputs</th>
<th>Planning Activity</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand data</td>
<td><strong>Level A</strong></td>
<td>Route changes</td>
</tr>
<tr>
<td>Supply data</td>
<td>Network Design</td>
<td>New routes</td>
</tr>
<tr>
<td>Route performance indices</td>
<td></td>
<td>Operating strategies</td>
</tr>
<tr>
<td>Subsidy available</td>
<td><strong>Level B</strong></td>
<td>Service frequencies</td>
</tr>
<tr>
<td>Buses available</td>
<td>Setting Frequencies</td>
<td></td>
</tr>
<tr>
<td>Service policies</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current patronage</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demand by time of day</td>
<td><strong>Level C</strong></td>
<td>Trip departure times</td>
</tr>
<tr>
<td>Times for first &amp; last trips</td>
<td>Timetable Development</td>
<td>Trip arrival times</td>
</tr>
<tr>
<td>Running times</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deadhead times</td>
<td><strong>Level D</strong></td>
<td>Bus schedules</td>
</tr>
<tr>
<td>Recovery times</td>
<td>Bus Scheduling</td>
<td></td>
</tr>
<tr>
<td>Schedule constraints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cost structure</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Driver work rules</td>
<td><strong>Level E</strong></td>
<td>Driver schedules</td>
</tr>
<tr>
<td>Run cost structure</td>
<td>Driver Scheduling</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1-1. Subproblems of the bus planning process [19]
From an optimization perspective, one could consider the whole process a single global optimization problem and for optimality, the global problem should be solved. However, due to the complexity and computational intractability of the problem, to the best of our knowledge, this has not been successfully done. Instead, the problem is broken into the five individual subproblems or some combinations thereof. There is a hierarchy to the stages, with the higher levels being long-term decisions, not being revisited for years or decades, while lower levels are shorter-term problems and may be revised as often as weekly or daily. When broken down into the stages shown, we see that the lower stages take as input the output of higher stages, and it is these relationships that prevent us from solving the global problem as a series of independent subproblems.

For the purpose of our research, we have chosen to focus on the highest stage of network design because by its position in the hierarchy, it probably has the most impact on the final performance outcome of the bus network. Also, as the most long-term decision to be made, it seems to be the most deserving of our attention and investment in obtaining the best possible solution. The relatively short-term nature of the later stages appear to lend themselves well to a more experimental approach, where new solution approaches can be tried with less risk and changes easily made if something does not work well.

1.2 Bus Network Design and Frequency Setting Problem

The network design and frequency setting problem aims to find a set of routes and their corresponding frequencies, given data about the demand for trips between different places in the city that the network is expected to serve, as well as data of the road network. Even as an isolated problem, Israeli and Ceder [11] show this to be NP-hard. The objective function for the problem is often multi-objective in nature and has to reflect the trade-off between cost to the operator and benefit to the traveller. Even when the objective function is solely concerned with minimizing the operator’s cost, the objective function is shown to be non-convex [14], making it difficult to solve to optimality. Constraints for the problem may include bus fleet size, required level of service in terms of minimum or maximum route frequencies and level of demand satisfaction.

Despite the difficulty and complexity of the problem, there have been many attempts at solving the problem using optimization. Complete enumeration of solutions is certainly possible, but impractical beyond very small network sizes as the possible combinations of road arcs to make singular routes as well as the combinations of routes that make up the network increase exponentially with road network size (in nodes or arcs). To make the problem computationally tractable, simplifying assumptions are often made.
Lampkin and Saalmans [12] separated route network generation from frequency setting, using a heuristic to perform the former while Mandl [13] assumed a constant headway for all routes, eliminating frequency setting completely. More recently, metaheuristics have been used to approach the problem, such as in Fan and Machemehl's papers of 2006 [6] [7] which used a genetic algorithm and a simulated annealing metaheuristic respectively to solve the problem for relatively small road networks with a few hundred arcs. For a more general survey of the overall bus network planning problem, we refer the reader to Guihaire and Hao [9].

To demonstrate the variety of modelling approaches and optimization methods used to solve the bus network design and frequency setting problem today, we review three recent papers: Wan and Lo [18]; Fan and Machemehl [7]; and Shimamoto, Schmöcker and Kurauchi [15].

1.3 Model Formulation Review

1.3.1 Wan and Lo [18]

Wan and Lo (WL) propose a linear mixed integer formulation of the bus network design and frequency setting where the objective is to minimize the total cost (it is not explicitly stated whose cost, although WL later refer to it as the operator’s cost) required to fully meet the given demands between origin-destination (O-D) pairs.

As is typical of most papers in the area, WL model the transit network as a graph. A set of nodes $N$ represent points of interest on the infrastructure network (e.g. intersections, stops and terminals). Some of these nodes (e.g. bus stops) are associated with a captive trip demand, and constitute the set of origin and destination nodes, which are subsets of $N$. It is possible for the three sets to intersect completely. Non-directed edges are used to represent a specific predetermined path on the physical transport infrastructure (roads in the case of buses) between nodes (from a bus stop to a given intersection, bus stop to bus stop, etc.). Both the node set and the arc sets are finite and non-empty.

Routes are defined as a sequentially ordered set of connected edges. Each route is associated with a specific frequency. Interestingly enough, WL have chosen to model each direction of travel along the route as a composite entity instead of separate routes. This may reduce the number of routes and hence decision variables, but appears to unnecessarily restrict the flexibility of the model and thus routes generated. Furthermore, this implies that each direction is served with identical frequency, again unnecessarily restricting network design. WL justify this constraint by saying it ensures vehicle conservation i.e. each bus
terminal starts and ends the day with the same number of vehicles although this could have been implemented more directly with another set of constraints. Another issue with the formulation is the constraint set to force all routes to be acyclic. Although uncommon, many real-world bus routes have been designed to be cyclic, and to the best of our knowledge, there is no proof that cyclic routes are always dominated by acyclic routes such that cyclic routes need not be considered by the optimization model.

WL introduce route-specific labels at each node that a route passes through to store the node’s position along the route. This allows WL to define a binary indicator variable they call “direct-in-vehicle link indicator variable” to identify whether a particular O-D pair is served by that route or not. This is used to identify the route choice set faced by travelers between a specific O-D pair and distribute them amongst available routes. One weakness of this approach is that there is no attempt to model any sort of passenger behavior in this model; the model is given free rein to distribute the demand between the routes in the choice set to achieve optimality as long as the law of conservation is maintained (no trips disappearing or appearing out of nowhere), which ignores customer preferences for faster/cheaper routes, which will affect the frequency setting for the route such that there is sufficient service to meet demand.

1.3.2 Fan and Machemehl [7]

Fan and Machemehl (FM) use a nonlinear mixed integer formulation with a multi-objective function that aims to minimize a weighted sum of user costs, operator costs and unsatisfied demand costs, i.e. not all demand has to be satisfied. This is one element that distinguishes it from WL and indicates an effort to construct a more sophisticated model that considers more stakeholders. The inclusion of unsatisfied demand seems natural but implies greater data collection efforts as we now need to have not only revealed demand but latent travel demand, which is harder to obtain.

FM use nodes to represent three kinds of entities. Firstly, they may represent centroids of specific zones, from which trips originate or terminate, where these zones could be a single building, group of buildings or some other geographically delineated area. Secondly, nodes may represent road intersections. Finally, “distribution nodes”, lying on roads and allow the centroids to connect to the road network may be bus stops.

Edges represent the use of a particular mode of transportation to travel between two nodes. The distinguishing of off-road centroids from road network nodes and thus, the need for centroid connector arcs implies the existence of walk arcs. These arcs represent the walk from the centroid to the bus stop where the passenger then waits and boards the bus for travel on the road network before disembarking at the relevant distribution node and walking to the destination centroid. This is useful to model because walk
time from the door to the bus stop is often a significant portion of the customer’s total travel time, which may contribute to the objective function value. This also accounts for the possibility of non-captive demand at bus stops, i.e. customers may choose to go to different bus stops depending on which stop offers a better route choice set.

FM define a route as a sequence of nodes, which must be connected by a link representing the transport mode of the route. Unlike WL however, routes traveling in opposite directions appear to be considered as separate route elements in the route set. Examination of FM’s solution methodology later reveals that the requirement for all routes to be acyclic (as in WL) is still enforced during candidate route set generation. Each route is again associated with a frequency.

Interestingly, FM distinguish between direct paths and paths involving transfers (transfer paths), although it is not immediately clear from their formulation of the optimization problem their reason for doing so. The separation appears to play a role only in their trip assignment model, to be discussed later.

1.3.3 Shimamoto, Schmöcker and Kurauchi [15]

Shimamoto, Schmöcker and Kurauchi (SSK) also propose a nonlinear mixed integer formulation with a multi-objective function that incorporates both operator and passenger costs. They do not however consider costs of unsatisfied demand as there is a hard demand satisfaction constraint in their formulation.

Nodes in SSK can be of different types. Firstly, there are origin and destination nodes, which are the sources and sinks of all travel demand (similar to centroids in FM). Stop nodes represent actual physical stops of the transportation service. Boarding and alighting nodes are also defined (note that these are not physical entities but rather points in time) in order to consider dwell time and capacity constraints.

SSK use directed arcs to represent movement between nodes, and arcs are named/defined intuitively. Boarding arcs connect from a stop node to a boarding node while alighting nodes connect from an alighting node to a stop node. Stopping arcs connect an alighting node to a boarding node, whose cost can then be used to represent dwell time. Walk arcs connect origin and destination nodes to stops.

1.3.4 Summary

Of the three papers, this formulation is the most comprehensive in considering the full travel experience of the passenger and accounts for all elements of travel time, which gives us the most precise calculation of travel time (provided precise data can be obtained) and hence, passenger cost functions. The drawback of such a modelling approach is the large number of additional nodes and arcs required relative to FM or WL.
In SSK’s formulation, travellers between any O-D pair have a hyperpath, which is actually defined as a choice set of “attractive” elementary paths between their origin and destination. Each elementary path consists of a sequenced series of nodes and the arcs connecting sequentially neighbouring nodes in the path. Unlike WL, a path includes both “transit arcs” that represent in-vehicle movement as well as walking arcs that represent miscellaneous travel activity such as boarding and alighting. This has the advantage that transfer costs can be explicitly modelled by placing these costs on the boarding/alighting arcs between the alighting node of the first line to the stop node and then to the boarding node of the line being transferred to. Again this is consistent with the implicit intention of modelling all aspects of the customer’s journey.

1.4 Optimization Methods Review

1.4.1 Wan and Lo [18]

The mixed integer formulation lends itself well to solution by commercial optimization software. In their illustrative example, WL used the CPLEX 6.0 mixed integer programming (MIP) solver. According to the CPLEX website at http://www.ibm.com/software/integration/optimization/cplex-optimizer/, CPLEX solves MIPs using either primal or dual variants of the simplex method or the barrier interior point method.

The network used for the demonstration of WL’s formulation in the paper consists of 10 nodes and 19 edges. The resulting mixed integer formulation consisted of 363 binary variables, 30 integer variables and 303 continuous variables for a total of 696 variables. This is 24 times the number of graph elements for a problem where only a maximum of 3 routes are allowed. We observe that the decision variables \( x' \) must increase exponentially with the number of routes considered and the number of arcs in the graph. Hence, this formulation has the potential to explode in size, which may rule out its usefulness for large metropolitan transport networks that typically involve hundreds of routes and tens of thousands of nodes and edges. The actual computing time and computational resources used to solve the network design and frequency setting problem on this example was not given in the paper.

1.4.2 Fan and Machemehl [7]

Because the problem is nonlinear and mixed integer, algorithms that solve to optimality are computationally too expensive (the problem is NP-hard). Thus a metaheuristic procedure known as simulated annealing (SA) is used. Starting from some arbitrary “state” corresponding to a feasible solution to the problem, the algorithm finds the global minimum by probabilistically moving from one state to a
neighbouring state of that state. The key to the algorithm is that occasionally (due to its probabilistic nature), the heuristic allows movement to worse states relative to the current one, which helps the algorithm to escape local optima in order to find the global optimum.

To generate the candidate routes for the network, FM use Dijkstra's [5] label-setting shortest path algorithm and Yen's [20] kth-shortest path algorithm on every centroid (O-D) pair to generate a set of routes that covers all centroids. The resulting paths are then filtered through user-defined minimum and maximum route length constraints to give a final set of candidate routes, from which a combination will be picked to form a candidate solution for the problem.

As mentioned earlier, both frequency setting and a trip assignment model is necessary to distribute the trips across the multiple routes/transfer routes that serve a given O-D pair in order to obtain the final flows on each edge and hence compute the objective function value to evaluate the quality of the solution. FM have integrated the two into what they call a “Network Analysis Procedure” (NAP). Given a candidate solution set of routes, the NAP assigns an initial set of route frequencies. It then assigns trips for each O-D pair to the routes available, favouring direct paths (using only one bus route) to those involving transfers in an all-or-nothing fashion. This is where the separation of direct paths and transfer paths plays a role in the optimization. This is a clear weakness in the algorithm because in reality we would expect that if there are sufficient cost (in time, money etc.) savings, people would prefer to take a transfer path rather than a direct path.

The frequencies are then recalculated (because each vehicle has limited capacity) such that sufficient vehicles serve the route to meet the demand assigned to it. These frequencies are then checked against the initial frequencies with which the trip assignment model was run. This is one iteration of the NAP, and the NAP runs iteratively until route frequencies converge sufficiently. With a consistent set of flows and frequencies, the objective function value can then be computed.

The overall simulated annealing algorithm runs as follows. The candidate set of routes is determined using the procedure described above. The SA then arbitrarily picks a combination of routes from the candidate set and passes it off to the NAP subprocedure. The NAP returns a set of frequencies for the candidate solution and its objective function value. The SA then constructs a neighbourhood solution by making a small change to the route set and gets the NAP to evaluate this neighbouring state as well. If the neighbouring state is better, the SA takes it as the local optimum. If the neighbouring state is worse, the SA probabilistically accepts it as the local optimum and repeats the whole process of generating a neighbouring state and evaluating it. This is done until a computational budget is exceeded or the rate of improvement in the objective function value is sufficiently small.
The obvious weakness of such a metaheuristic algorithm is that optimality cannot be guaranteed. However, this is typically acceptable because the solution space is too large and computationally expensive to explore and we may be satisfied with a solution that is “good enough”. The strength of such a metaheuristic approach is clearly the savings in time and computational resources required. Another strength of the metaheuristic procedure is its flexibility in parameters. Different rates of “cooling” (which affect the probability of accepting a worse solution) and stopping conditions can be chosen, which may give different results and allow user control of computation time.

The key finding of this paper is that SA is better than another commonly used metaheuristic algorithm known as the genetic algorithm for a given computational budget and/or target convergence rate in the objective function value.

1.4.3 Shimamoto, Schmöcker and Kurauchi [15]

Similar to FM, SSK use a metaheuristic algorithm to solve the problem. Again, the three subprocedures involve route generation, frequency setting and trip assignment.

While FM used some shortest path algorithms to generate the k-shortest paths and used those paths as routes, SSK uses a genetic algorithm to generate routes. In the genetic algorithm, a gene contains the value of the next node in the route. Over multiple iterations, the algorithm creates a variety of routes as “mutation” occurs and values of the genes change. Thus there is no guarantee as to the straightness of routes. This allows the creation of very convoluted routes which might be undesirable for operational reasons and displeasing to passengers.

SSK also deal with frequency in an indirect manner. Unlike FM where the frequency variable is set directly, SSK use a genetic algorithm to randomly assign a number of vehicles to each route. The number of vehicles is divided by the travel time of the route to obtain the frequency. This has the advantage of maintaining vehicle numbers as integers, unlike the case where frequencies are set directly, which may give non-integer values for the number of vehicles required. This is easier to interpret and operationalize.

Finally, trip assignment is done by probabilistically distributing trips amongst all the arcs in a hyperpath. Where passengers can decide between boarding one bus service or another, the probability is proportionate to the frequency of service to reflect the assumption that passengers board the first arriving vehicle that can get them to their destination (whether directly or with transfers). This assignment model has the weakness that it assumes passengers know the full network well enough to be able to enumerate their choice set perfectly. Furthermore, passengers may not want to take certain services if those services have high load factors or take longer travel times.
The most interesting aspect of this methodology is that it allows for the consideration of what SSK have termed the “common lines problem” where multiple routes can serve the demand for a given O-D pair. The toy example used in the paper demonstrates how consideration of this problem results in radically different network structure/strategy. Without consideration of the problem, most ‘road arcs’ are only served by one route while with common lines, the main travel corridor is served by multiple routes that branch off to service minor travel corridors as appropriate.

1.5 Research Objectives

With all the assumptions and problem size limitations imposed when using an optimization approach, it is not difficult to imagine that such approaches would be difficult to implement in real-life situations. For example, the Singapore road network data we used in this paper had over 6000 arcs. Many cities continue to rely on human planners to design their bus networks, with or without the assistance of modelling tools such as TransCAD or EMME. Solutions obtained using optimization models such as those mentioned above may be used as ‘guides’ but planners often end up modifying them using their experience and intuition. Such intuition may take the form of heuristics, such as “longer routes are better” or “having a larger number of direct routes is better than less but more circuitous routes”. In other words, implicit judgements are being made about the importance and effectiveness of network design criteria such as route length, route count and route directness.

It is these design criteria and their influence on network performance that we are interested in examining in this thesis. Instead of attempting to innovate a new optimization approach to solve the problem to (or close to) optimality, we would like to examine the effect on network performance of varying the influence of these design criteria in the route network generation process. Our goal is to gain general insights into the problem and characteristics of “better” bus networks.

1.6 Outline of Thesis

In Chapter 2, we describe a preliminary case study and demonstrate the feasibility of solving the two subproblems of network design and frequency setting sequentially, using an algorithm for the former and optimization for the latter. We detail our computational experience applying the approach to a small network. With the lessons learnt, we propose, in Chapter 3, a more sophisticated methodology to investigate the effect of various design criteria by using a parameterized algorithm to generate route networks. In Chapter 4, we elaborate on our route network generation approaches in greater detail, while in Chapter 5, we describe some possible trip assignment methods and our choice of trip assignment
procedure to evaluate the route networks we generate. In Chapter 6, we describe the sources and preprocessing of data used in our research. We state the numerical results and discuss them in Chapter 7, and summarize our conclusions and propose directions for future work in Chapter 8.
2. Preliminary Case Study

In this case study, we aim to prove the feasibility of a decomposition approach for solving the network design and frequency setting problem. We describe the nature of our solution strategy and implement it on a small problem. Analyses of results as well as computational performance are described.

The overall objective of the problem is to determine the set of routes (which we refer to as a route network) and their associated frequencies that minimize the total travel time. We consider two components of travel time in this problem: link traversal time and waiting time at the bus stop. We formulate the model with the aim of capturing the trade-off between the desirable goals of maximizing the number of routes (hence increasing the directness of routes and reducing in-vehicle travel time) and maximizing frequency (reducing the waiting time), for a given bus fleet size.

2.1 Problem Input and Formulation

We model the given road network as a directed graph $G (N, A)$ consisting of a set $N$ of nodes and a set $A$ of directed arcs.

All nodes in set $N$ represent bus stops, a subset $T$ of which are bus terminals, where buses must start and end their routes.

Each arc $ij$ in this graph represents a feasible road connection between bus stops $i$ and $j$ that does not pass another bus stop that is not $i$ or $j$. We account for the underlying road structure (non-driver side turns etc., no of traffic lights encountered etc.) in the cost of that arc.

We also assume we are given a full origin-destination (O-D) matrix, where associated with each O-D pair $h$ is a travel demand $w_h$ from origin $o_h$ to destination $d_h$ in trips/time period.

Each arc in the graph has a constant cost (traversal time) of $c_{ij}$.

The decision variables are routes and their frequencies.

Each route $r$ is a sequenced set of nodes $\{n_1, n_2, n_3, ..., n_m\}$, or equivalent arcs $\{a_{n_1n_2}, a_{n_2n_3}, a_{n_3n_4}, ..., a_(n_mn_1)\}$ where the first and last node can only be terminals (may be the same terminal).

Each route $r$ has an associated frequency $f_r$. 

25
2.2 Route Generation

We define a feasible route set as one where all bus stops are visited at least once by a route in the route set and all terminals have at least one route in the route set that starts from it and one route in the route set that terminates at it.

Originally, a minimum cost network circulation problem was considered as a possible method to generate a feasible route set. The advantage of such a formulation is its problem structure allows the network simplex to be used as the solution algorithm, which is faster than regular simplex due to the ease of selecting the pivot point for the algorithm. The network would look as follows:

![Diagram of generalized model of network](image)

Figure 2-1. Diagram of generalized model of network

We split each bus stop \( s \) into two nodes \( s \) and \( s' \) and draw an arc from \( s \) to \( s' \) with a lower bound on flow of 1, this forces the bus stop to be visited at least once. The costs of all arcs are 0 except for the one from the sink to the source, which has a cost of 1.
The disadvantage is that the routes generated are not immediately apparent. By conservation of flow, we can decompose the solution to a set of cycles, each representing the route a bus takes around the network. However, there are multiple ways we can decompose the solution. One possible decomposition algorithm is to start at the source node, and always move along the adjacent arc with the greatest flow until we reach the sink node, removing one unit of flow as we move along the arcs. This path through the flow network is a route. To minimize the number of routes generated, we traverse the path repeatedly until all flow has been removed from one of its constituent arcs. We then restart the algorithm to pick out a new route.

Another disadvantage becomes apparent: the network circulation formulation, with its strict structure, does not let us formulate certain constraints on routes, such as maximum and minimum route lengths. We must account for these constraints in the decomposition process.

For this project, we decided to use an algorithmic approach that generates the most direct routes (with respect to terminals) while ensuring all bus stops are visited by at least one route. This algorithm uses the one-to-one shortest path as a subproblem, which is a special case of network flow/linear problems with very efficient and well-established solution algorithms, such as Dijkstra’s algorithm and the Bellman-Ford algorithm.

### 2.3 Route Generation Algorithm

We control the algorithm using three lists. The first contains all terminals that do not yet have a route that originates from it (unprocessed-out list). The second contains all terminals that do not yet have a route that ends at it (unprocessed-in list). The final list contains all stops that are not included in any route yet (unprocessed stops list). To ensure all stops are visited, by the end of the algorithm, all these lists should be empty.

1. Choose an unprocessed-out terminal at random.
2. Find the shortest paths to every other terminal in the unprocessed-in list.
3. Add the longest of these paths (in the event of a draw, pick one of the drawn paths at random) to the route set.
4. We delete the origin node of the added path from the unprocessed-out terminals list and delete the destination node of the added path from the unprocessed-in terminals list.
5. Delete all the stops visited by the added path from the unprocessed stops list.
6. Repeat from step 1 until the unprocessed-out list is empty.

7. Choose an unprocessed-in terminal at random.

8. Find the shortest paths from all other terminals to it.

9. Add the shortest of these paths (in the event of a draw, pick one of the drawn paths at random) to the route set.

10. Delete the destination node of the added path from the unprocessed-in terminals list.

11. Delete all the stops visited by the added path from the unprocessed stops list.

12. Repeat from step 7 until the unprocessed-in list is empty.

13. Choose an unprocessed stop at random.

14. Find the shortest paths into the stop from all terminals.

15. Find the shortest paths out of the stop to all terminals.

16. Choose the shortest in-path and the shortest out-path and merge them to form the new route to be added to the route set.

17. Delete all the stops visited by the added path from the unprocessed stops list.

18. Repeat from step 13 until the unprocessed stop list is empty.

At step 3, we choose the longest terminal-to-terminal path to ensure connectivity, that is, that there is some intersection of routes somewhere in the road network, avoiding a disjoint route network.

2.4 Bus Route Network Graph

For a given route network, whether an existing one, or one generated via optimization/algorithmically, we need to determine its performance. In this study, we use one performance metric - total travel time of all passengers, although other performance measures are possible, such as equity in distribution of travel times experienced by individual passengers, operating costs, proportion of trips requiring one, two or more transfers etc.

The total travel time is calculated as the sum of in-vehicle travel time and waiting time. While in-vehicle travel time varies with the stops at which the passenger boards or alights, average waiting time is experienced exactly once per route. This presents a modelling challenge which will be discussed shortly. To obtain the total travel time by all passengers, we have to determine the number of passengers who
used each arc, or the “flow” on each arc, and sum the products of the flow on each arc and the cost on the arc.

Due to the different nature of the problem, we cannot reuse the earlier road network graph. Instead we construct a new bus route network graph (route graph), $G'(N', A')$, on which we will perform the trip assignment. The set $N'$ of nodes includes all the nodes $N$ in the original graph. We then go through the routes in set $R$, and add a “route node” for each stop visited by each route, linking them by “route arcs” which are added to $A'$ and have a cost $c_{ij}$ equal to their corresponding arc in $A$. We also add one boarding and one disembarking arc between each route node and its corresponding stop/terminal with costs equal to $f_i/2$ and 0 respectively. Such a graph $G'$ lets us capture the waiting time as well as allowing transfers between routes.

![Example of route network graph with two routes](image)

Figure 2-2. Example of route network graph with two routes

Figure 2-2 is an example of what the route graph would look like for two routes that have the same sequence of stops. Blue nodes are actual bus stops/terminals. Black nodes and arcs are route nodes and arcs respectively. Route arcs are kept distinct so we can check that capacity constraints are satisfied for each route. Red arcs are boarding and disembarking arcs, this allows us to capture waiting times. The dashed blue arcs do not exist in the route graph, but their cost values are used to cost the corresponding route arcs.
To determine the cost of boarding arcs, we would need the frequency of the route. Once known, assuming random arrival of passengers to a bus stop with uniform distribution, the expected waiting time to the next bus on a particular route is the frequency of that route divided by 2. We could try to solve for frequencies and passenger flows simultaneously, however this results in a multi-commodity transportation problem with a nonlinear objective function when we try to minimize total travel time.

The number of commodities, or distinct O-D pairs, \( |H| = |N|^2 - |N| \) grows exponentially. Even the smallest test network used in our experiments, a 2 by 2 block grid road network, has \( |N| = 26 \) and \( |H| = 650 \), presenting tractability issues in the large nonlinear problem.

### 2.5 Trip Assignment

We thus decompose the problem further into trip assignment and frequency setting. Given that in the interest of maximizing social welfare, we try to meet all travel demand. It therefore seems reasonable to perform a bus fleet assignment that provides sufficient capacity on all routes to meet the flows expected on the route network, which we would determine in a separate trip assignment model. Once the number of buses assigned to each route is found, we can calculate frequency of a route by taking the time period of the model and dividing that by the number of buses assigned that route. Because this bus fleet assignment requires the flows on each arc, we should perform the trip assignment first.

Assuming that there is some desired minimum frequency on all bus networks, we can initialize the costs of boarding arcs using this minimum frequency and perform the trip assignment. Again, the intuitive formulation for trip assignment as an optimization problem is a multi-commodity transportation problem with decision variables \( x_{ijh} \), the amount of flow on arc \( ij \) for O-D pair \( h \). We encounter the curse of dimensionality, as the number of O-D pairs and hence decision variables considered increases exponentially with the size of the network we model.

We thus consider an alternative solution technique which again involves an algorithm using the shortest path problem as a subproblem. For each O-D pair, we find the shortest path from the origin to destination stop and assign the full travel demand as the flow on all the arcs constituting this shortest path. This form of trip assignment is commonly known as an ‘all-or-nothing’ assignment. We process all the O-D pairs, cumulatively adding to the flow on each arc. Clearly, we are ignoring capacity constraints and assume that sufficient buses will be assigned to the routes to make these shortest path trips possible. While we still have to consider \(|H|\) distinct O-D pairs, the shortest path problem is extremely quick to solve.
2.6 Frequency Setting

Frequency setting is a relatively simple linear program where the decision variables are $n_r$, the number of buses to assign to route $r$ per minute. We have capacity constraints; the number of buses assigned must be enough to handle the assigned flow to the arcs. Secondly, the minimum frequency constraints, which is in effect a lower bound for the decision variables $n_r$. The input to this part of the problem is the given bus fleet size. The vector of costs is intentionally left at 0 as we only care about feasibility and want all buses to be used, which should improve frequency and thus minimize total travel time (our ultimate objective).

Having solved for the frequencies, we realize that this may affect the optimal solution to the trip assignment problem, hence we iterate between the two models until convergence is observed in the solved values of route frequencies.

2.7 Implementation

All the above algorithms and optimization problems were coded in MATLAB R2012b Windows 64-bit version, with the Bioinformatics Toolbox which includes a shortest path solver. The default algorithm used by the solver function is Dijkstra’s algorithm.

2.8 Road Network of Case Study

To test the accuracy and efficiency of our problem solution strategy, a small network was constructed based on the following hypothetical road network:
Figure 2-3. Diagram of road network for a $2 \times 2$ block neighbourhood

According to the rule by which we draw arcs, which is to only draw arcs between stops that are feasible on the route network, without passing another bus stop, the following road graph is produced:
Figure 2-4. Road graph for the 2 x 2 block neighbourhood

The O-D matrix was generated using MATLAB's inbuilt random integer function randi with uniform distribution on the interval [1 40], i.e., we assume between 1 to 40 boardings/disembarkings occur at a bus stop per hour. This is divided by 25, to obtain the number of trips for a given O-D pair. The demand values are given in Appendix 1.

To obtain sufficient results for analysis, we solve the problem a total of 50 times with the same road network and demand. This is subdivided into 5 runs of 10 experiments so we can average out any errors in timing the computation using MATLAB’s inbuilt Profiler tool. The other parameters of interest were as follows:

1. BUSCAP = 45; maximum number of passengers per bus
2. BUSFLEET = 60; maximum number of buses on the road per hour
3. MINFREQ = 1000; minimum frequency in minutes
2.9 Results

We plot the total travel time on each of the networks generated versus the number of routes in that route network:

Figure 2-5. Graph of total travel time v. Number of routes in route network for 2 x 2 block neighbourhood

We observe a wide variance in travel times. The minimum was 6962 minutes versus a maximum of 7785 minutes, a difference of 823 minutes, or 11.8% of the best time. This was all for the same bus fleet size and demand distribution, which confirms that network design has a major role to play in determining travel times incurred by passengers.

It is also interesting to note that total travel time decreased very strongly with the total number of routes. We observe that the worst performing 8-route network was still better than the best performing 9-route network. The same relationship applies between 9-route and 10-route networks.
The best performing network is illustrated in Figure 2-6 below:

Figure 2-6. Best performing network
For completeness, we present the results of the 50 experiment run on a 2 by 4 block grid.

This time, the difference in performance between route networks with fewer routes and those with more is less extreme. There is significant overlap and the difference between the best and worst performers is only 7.16%.

2.10 Computational Performance

All 5 runs were performed in a time of between 7.985 seconds and 8.758 seconds with an average of 8.517 seconds. From the profiler results in Appendix 2, we observe that most of this time is spent on the trip assignment portion of the problem. This is no surprise as the majority of the shortest path subproblems to be solved are mostly due to this section rather than the route generation algorithm.

To get an empirical idea of how the computational requirements might scale with the problem, we extended the size of the problem to a 2 by 4 block road grid network. This results in a total of 46 stops versus 26 in the earlier test problem. We made 1 run of 50 experiments, which took 246.963 seconds to solve. We conclude that again that this is due mostly to the trip assignment subproblem which has increased from ~8 seconds to 48.72 seconds, a factor of about 6 times. Meanwhile, the route generation subproblem time has only increased from 0.118 seconds to 0.15 seconds, a factor of 1.27 times.
Knowing that trip assignment is the most resource-intensive subproblem suggests that we should redesign our overall strategy to avoid having to iterate on the trip assignment subproblem and obtain a set of frequencies in some other manner.

2.11 Conclusion

We have demonstrated that such a decomposition approach to solving the bus network design and frequency solution is feasible, although computationally intensive due to the iterative application of optimization to solve for frequencies. It is also difficult to isolate the effects of network design versus frequency setting on the performance of the network. Hence, we are justified in separating the two subproblems when generating our route networks to test the effects of the various design criteria. The trip assignment procedure may also be overly simplistic as it does not account for the possibility of alternative paths from origin to destination, and we will require a more sophisticated trip assignment procedure for determining the effects on network performance of various design criteria. Generating the route network using optimization may give the optimal answer, but does not give us much insight into how the various design characteristics of the network affect performance. We can only conclude that there seems to be some correlation between fewer routes and better network performance.
3. Proposed Methodology

Our proposed approach uses a route network generation process where design criteria can be parameterized so that the effect of the various criteria on the route network generated can be quantified and controlled.

3.1 Design Criteria

3.1.1 Route Overlap

It is intuitive that overlapping routes must be in some sense, inefficient, because additional buses are being assigned to roads that already have bus service. This inefficiency is exacerbated by the characteristic positive returns to scale of ridership for public transportation. Generally, as ridership on a route increases, the route becomes more cost-effective because the high fixed costs can be better distributed. Furthermore, as more buses are allocated to the route, waiting time is reduced, making the route even more attractive and creating a positive feedback loop. Thus, as far as possible, we want to consolidate as much ridership as possible onto a few routes rather than distributing demand amongst many routes.

However, overlap may be inevitable due to topological features, such as bottlenecks or at areas near bus terminals, where routes must diverge or converge. High route overlap may be considered a distinguishing feature of destination-oriented networks, as opposed to direction-oriented networks. It is also, by necessity, highly correlated with transfer-averse networks, which are route networks that try as far as possible to serve origin-destination pairs with direct routes, to minimize the number of transfers required by travellers. The network’s properties would affect network connectivity, as defined below. An example of these networks can be seen in Figure 3-1 below:
3.1.2 Route Length

Route length necessarily affects the number of routes in the route network, because longer routes cover more roads and thus fewer routes will be required in the network to serve the same set of stops. Longer routes are also a likely consequence of destination-oriented or transfer-averse networks. This is necessary because the focus on reducing transfers necessitates distant origin-destination pairs to be connected by one route instead of two or more. Route length may also be an indirect proxy for route circuity, because an emphasis on more direct routes reduces the deviation from the shortest path that routes may take.

There is some relation between route overlap and route length, however the direction of correlation is not clear. Because longer routes typically mean the route network as a whole requires fewer routes, reducing the likelihood of overlap. On the other hand, longer routes mean that it is more likely for any given route to include a road arc that another route might already include.

3.1.3 Network Connectivity

We interpret network connectivity quantitatively by the number of transfer opportunities available on the network. Thus, the more bus stops there are visited by more than one route, and the more routes available at these stops, the more transfer opportunities there are and hence, the network is better-connected. Another aspect of connectivity that could be measured is the spatial distribution of bus stops where transfers can be made. These may of course include terminals, which usually have the highest concentration of transfer opportunities because all routes have to start and end at terminals. Transfer
opportunities could be highly concentrated at few locations, as in a hub and spoke network, or evenly distributed (both in volume and spatially) in a grid network. This spatial measure of connectivity was not investigated in this work.

3.2 Network Design and Frequency Setting

We propose an algorithm that generates single feasible bus routes. The definition of a feasible route can be found in section 4.1. This algorithm first selects an origin-destination pair that the route will serve. It then generates a list of route skeletons, which are all the combinations of start and end terminals with the origin and destination pair inserted in between, ignoring those that are infeasible due to violation of maximum route length and circuitry constraints. The skeletons are then scored based on a weighted sum of various measures of the design criteria we decided on above, and the best one is chosen. These weights are the parameters of the algorithm we can vary to control the final route network produced. The chosen route skeleton is expanded into a feasible route by repeated nodal insertion. Again, a weighted sum is used to score the candidate nodes to decide which should be inserted. After the route network has been generated, if necessary, an additional frequency setting heuristic is used to propose a feasible set of frequencies for the routes. Ideally we would be able to optimize the frequencies as well, and techniques to do so should be substituted for the heuristic in future work.

3.3 Trip Assignment

In order to obtain performance measures for the networks, we next run a trip assignment, using historical trip data, on the networks generated in order to obtain performance measures for the networks. The following measures were chosen: total travel time, in-vehicle travel time, waiting time and number of boardings.

We have chosen to treat the operator-side costs as a constraint in our model by fixing the resource availability of all route networks. This resource is bus-minutes, the total amount of time that all the buses in the fleet can spend in total on the road. The amount of bus-minutes thus defines the relationship between the frequency of the route and its length.

3.4 Order of Investigation

We propose two different approaches to route network generation. The first generates routes one at a time until some stopping condition is reached while the second uses a pre-generation all-or-nothing trip
assignment to generate lower bounds for a covering problem on the road network. The two approaches were chosen because, while the effects of the design criteria were more transparent in the first, it has certain failures such as ignoring network effects that the second approach might handle better. We use both approaches while varying the relative weights of the design criteria in the skeleton/node scoring function to generate a variety of route networks. These approaches are detailed in Chapter 4. We then run our implementation of Spiess and Florian's [16] trip assignment algorithm, first on the existing Singapore bus network as a benchmark, and all the route networks generated by our two approaches.
4. Route Network Design

For the purpose of this approach, we define the objective of the route network design stage of the overall transportation planning problem to be as follows: given an origin-destination matrix of trips demanded between bus stops and a graph representing the road network that can be traversed by buses to travel from one bus stop to another, determine a set of feasible bus routes that ensures all trips in the origin-destination (O-D) matrix can be served. The definition of a feasible bus route is left to section 3.2 on Single Route Generation.

Note that our problem definition differs from that commonly used in the literature in some significant ways. First, we require that all trips be served. We choose to do this because we are using historical trip data to generate the O-D matrices. These data represent trips that people made and want to make on the existing public transportation system. Thus, any network we design as an alternative to the existing system should at minimum be required to serve these trips so that existing users would not be forced to the socially less desirable alternative of private car use. An advantage of the requirement to serve all trips in the O-D matrix on the route network is that it provides a convenient stopping condition when designing the route network by repeated single route generation.

Also, we have assumed that all demand for trips originate at bus stops and that such demand is captive. This allows us to ignore the problem of defining centroids for zones of demands and defining valid walking arcs from centroids to bus stops. We then do not have to consider questions such as whether a given bus stop should be served by the route network, or how people might choose to start their trips at different bus stops depending on which routes supply those stops. Because the focus of this research is on the route network design, we believe the benefits of reductions in data required justify the problem simplification.

It should also be noted that the routes designed by the algorithms in this section are non-express services, i.e. they stop at all stops in their itinerary. If the possibility of express routes are to be considered, extra directed arcs should be inserted (with the appropriate in-vehicle travel times) between the stops that are to be considered for express routes. The decision of which stops should be considered for express routes is left to the network designer and is not considered by our model or algorithm. For an example of such work, we refer the reader to Chiraphadhanakul [17].

Two distinct approaches to network design were used in this research. The first is to generate routes one at a time until some stopping condition is achieved, and the set of routes generated up to this point is the output route network of the algorithm. The second is to generate a large set of candidate...
routes, more than we estimate we would need in the route network, and solve some variations of an optimization problem to decide the routes to be included in the route network. These two approaches are further elaborated on below.

4.1 Single Route Generation

Both approaches depend on an algorithm to generate individual feasible bus routes, which we call the Single Route Generation (SRG) algorithm. To be feasible, a route must begin and end at valid start and end terminals respectively, every sequential pair of stops in the route’s itinerary must be connected by a road arc, and the route’s length and circuity should not exceed the designer-determined maximum allowable route length and circuity.

Our SRG algorithm is based mainly on that proposed by Baba [3], which in turn combines aspects of the two differing route generation approaches of Lampkin and Saalmans [12] and Baaj and Mahmassani [2]. Lampkin and Saalmans introduce the idea of route skeletons, consisting of four nodes where the first and last nodes are the start and end terminal of the route. They then insert nodes into the “gaps” between the nodes in the route skeleton until the route is completed, the choice of which depends on some criteria deemed to be “desirable” in a “good” route network. Baaj and Mahmassani use a shortest (or close to shortest) path as a skeleton before performing nodal insertion/replacement on the skeleton according to a user-chosen operating cost and passenger service trade-off strategy.

Our chosen approach to single route generation reflects the objective of this research, which is to determine the relative importance of differing design criteria and principles. To do so, how the various design criteria influence the route generation should be clear and quantifiable. Our SRG algorithm proceeds as follows:

1. select a pair of nodes to be connected by the route,
2. generate all feasible route skeletons for the pair of nodes,
3. select a skeleton for expansion, and
4. expand the skeleton into a feasible route through repeated nodal insertion.

4.1.1 Required Inputs and Preprocessing

The list of all bus stops to be served is required and it must be known whether each bus stop is a valid start and/or end terminal for routes. The geographical location of stops is optional, but may aid in visualization of the resulting bus route network and further spatial analysis. To reduce computational time
during route expansion, we define nodes which have exactly one arc in and one arc out as “through
nodes” and combine adjacent “through nodes” to form “segments”, because once a through node is
inserted into a route, its neighbours must be inserted as well. Segments and nodes are treated equally for
all intents and purposes.

The list of all roads that are traversable by bus is the next requirement. Each road arc has a from
node and a to node, representing the bus stops it connects if included in a route. The in-vehicle travel time
for the arc must be known.

We also require an O-D matrix of all trips that the route network is required to serve; this is
named ‘demand’ in the code. A duplicate of this matrix, ‘unmetDemand’, is also kept. This matrix will be
used to keep track of unserved trips as routes are generated. As routes are generated, all O-D pairs that are
connected by the network have their corresponding entry in the unmetDemand matrix set to 0.

A matrix of minimum travel times between all nodes is also required for the algorithm. This was
obtained by using MATLAB’s graphallshortestpaths function on the road graph. This is used to estimate
the lower bound of the length of a route that is still under expansion. The difference between this lower
bound and the length of the route with a prospective node inserted is also used as part of the score of the
node when deciding which node will be inserted next into the route under expansion.

There are two design parameters. First, the maximum allowed route length, where route length is
the sum of all the in-vehicle travel time on arcs between the stops that make up the stop itinerary of the
route. The other is the maximum allowable route circuity, where circuity is defined as the ratio of the
route’s length to the shortest path distance on the road graph between the start and end terminals of the
route.

Last, we are free to decide and vary the weights allocated to the various criteria used to score the
route skeletons during skeleton selection and nodes during route expansion. Together with the design
parameters, these weights will allow us to explore the question of which design criteria matter in
designing route networks.

4.1.2 Selection of Node Pair

One of key requirements for a feasible route network is to ensure that all trips in the O-D matrix can be
served. It therefore makes sense to design a mechanism in the route generation algorithm that ensures the
route generated will fulfil at least the trips between one O-D pair. Instead of leaving things to the
“chance” that the final route after route expansion will connect the pair of nodes, we include the nodes in
the route skeleton from the start. Thus, in the worst-case, we can ensure all O-D pairs are served by generating one route for every O-D pair. A route skeleton has a minimum size of two stops (when the node pair selected is a start and an end terminal).

4.1.3 Generation and Selection of Route Skeleton to be Expanded

Once the node pair to be connected has been determined, all feasible route skeletons for the node pair are generated by combining every possible combination of start and end terminal with the segment(s) of which the nodes in the pair are components of. We can determine a lower bound for the length of the final route by summing the shortest paths between each of the four nodes currently in the route skeleton. If this lower bound is above the maximum allowed route length, the skeleton is immediately discarded.

Each route skeleton is then given a total score, which is the weighted sum of the design criteria we have chosen to examine.

1. To score demand, we sum up all the trips between all the nodes in the route skeleton by referring to the O-D demand matrix. These are all the trips that the route would be able to serve on its own, i.e. without transfers.
2. The next criteria is unmet demand, which is calculated the same way as demand, except the values used for trips served are taken from the unmet demand O-D matrix. The resulting value is the number of unserved trips the route being constructed will serve without transfers.
3. Length is measured by the difference between the maximum allowed length and the lower bound of the total route length with the candidate node inserted. This lower bound is calculated by using the minimum travel time matrix to obtain the minimum travel time between sequential nodes in the route skeleton’s current itinerary and summing them. We do this to ensure that shorter route lengths contribute positively to the overall score, which preserves the non-negativity of the score. Non-negativity is required for the route generation approach elaborated in section 4.3.

The skeleton with the highest total score is selected for route expansion.

4.1.4 Route Expansion

The first step in route expansion is to take the skeleton in its current state and determine where there are “gaps” for nodal insertion. A gap is defined as a pair of nodes which appear sequentially in the stop itinerary of the route skeleton that are not connected in the road graph, hence the possibility of inserting a node (or its segment) to “close” the gap.
For each gap, all nodes are checked for feasibility against the maximum allowed route length and
circuity and then feasible nodes are evaluated by calculating the feasible node’s score. Similar to scoring
skeletons, the node’s total score is a weighted sum of the following criteria:

1. To score demand using the demand O-D matrix, we sum up all the trips between all the nodes in the
route skeleton before the gap to the candidate node and from the node to stops in the itinerary after
the gap by referring to the demand O-D matrix. These are the additional trips that inserting the node
would enable the route to serve without transfers.

2. The next criterion is unmet demand, which is calculated in the same manner as demand, except the
values used for trips served are taken from the unmet demand O-D matrix.

3. Length is measured by the difference between the maximum allowed length and the lower bound of
the total route length with the candidate node inserted. This lower bound is calculated by using the
minimum travel time matrix to obtain the minimum travel time between sequential nodes in the route
skeleton’s current itinerary and summing them.

4. Delay is the difference between the lower bound on the route length before and after the insertion of
the candidate node, multiplied by the number of trips being made from stops before the gap being
considered to stops after the gap.

5. The last scoring component is the number of routes already generated that included the candidate
node.

From all the candidate node-gap combinations, the combination with the highest total score
chosen and the appropriate candidate node inserted into that gap.

This process of finding gaps, scoring all nodes for all gaps and picking one node-gap combination
for insertion is repeated until the route is connected, that is, that all sequential pairs of stops in the route’s
itinerary are connected by route arcs and no gaps exist.

4.1.5 Cycle Removal

During route expansion, it is possible that a node chosen for insertion is already on the route under
construction. This creates a cycle, which is generally undesirable because it would cause additional travel
(delay) for passengers travelling from stops before the cycle in the route itinerary to stops after the cycle.
Thus when a repeat is detected, all the nodes that are part of the cycle, except for one of the instances of
the repeated node, are deleted. Deletion is not performed, however, if part of the cycle includes either the
origin or destination node of the O-D pair that was selected at the beginning of the SRG process as that
would result in a route that does not serve the trips between the O-D pair.
4.2 Network Generation by Repeated Single Route Generation

In our first network generation method, we repeatedly run the SRG algorithm, generating one route at a time until some stopping condition is reached. As the definition of a feasible route network is one that connects all O-D pairs that have trips in the O-D matrix, this is a logical stopping condition to use. To track which O-D pairs are connected by the routes thus far, a subgraph of the road graph is generated after every single run of the SRG, using the road arcs used by at least one of the routes generated thus far. We can then use the Hao-Kocur algorithm for shortest paths [10] with the origin node as the root node and check the distance label of the destination node to determine if the two are connected.

It seems reasonable to try to generate routes that can fulfill the most trip demand first. This is achieved by selecting the O-D pair not yet connected with the highest unsatisfied trip demand. The O-D demand matrix is read and any O-D pairs with non-zero entries are stored as O-D pair objects in a maximum binary heap. Choosing the next O-D pair for which to generate a route is done by performing a heap deletion, checking if the deleted pair is connected in the current road subgraph, using the Hao-Kocur algorithm, and repeating this process until a O-D pair is obtained that is not connected.

In all cases where scoring is performed, the candidate skeleton/node with the highest score is used for the next step in the SRG algorithm. As routes are generated, the heap of O-D pairs gets smaller as more O-D pairs are connected in the network; the route network is complete when the heap is empty.

This approach to network generation has the advantage of making the effects of the various design criteria easily manipulated by adjusting the relative weights assigned to each design criterion in scoring the skeletons or nodes. It is also deterministic; hence it is easy to error check the network generation process by running the algorithm again on the same inputs as a previous run and comparing the outputs.

The main problem we would expect with this approach is that because it generates routes one at a time, it is unable to take network effects into account well when generating the network. Furthermore, because unserved demand is a design criterion, the final route network is very sensitive to the order in which O-D pairs are picked. Another possible issue is once an O-D pair is connected in the network by a path requiring transfers; the algorithm does not consider generating a direct route for that O-D pair even if a direct route might save a lot of travel time for those trips. To address these issues, we devised a second network generation approach.
4.3 Network Generation by Set Covering

In this approach, we use the SRG algorithm to generate a large number of routes greater than the number we think we will actually need in the final route network. We call this set of routes the candidate route set. We then solve a variation of the set covering problem to determine a subset of candidate routes that will be the route network. A notable difference from the network generation approach above is that this one attempts to solve the problem at a network level, considering network effects.

To determine which road arcs need to have bus service, we perform a shortest-paths all-or-nothing trip assignment for every O-D pair. The resulting arc flow volumes therefore are a reflection of how travellers would get from their origins to destinations if they had cars, aggregated by arc. These volumes are used in the optimization problem as minimum constraints to ensure sufficient bus resources are allocated to routes to provide bus capacity on these arcs.

4.3.1 Modification of the SRG Algorithm

To ensure sufficient variety of routes in the candidate route set, the SRG is modified in one significant way. When selecting route skeletons or candidate nodes, instead of choosing the highest scoring skeleton or node, selection of the skeleton or node for the next step in the SRG process is randomized. The score of all the candidates is summed and each candidate is given a probability of being selected equal to its score over the sum of all candidates’ scores. This is the reason we modified the measure of length to ensure the total score is always non-negative.

As we are going to be solving a set covering problem over the road arcs, we must ensure that for every road arc, there exists in the candidate route set at least one route that includes it to make the set covering problem feasible. Hence we also have to modify the O-D selection stage of the SRG algorithm. After the heap of O-D pairs that need to be connected has been emptied, subsequent iterations of the algorithm will select from the road arcs that have not been used thus far.

The stopping condition of the repeated SRG algorithm is also no longer when the heap of unconnected O-D pairs is empty, because we need additional iterations of the SRG algorithm to generate routes for the road arcs. Instead we set the stopping condition to be a target number of routes we want in the candidate route set. When the list of unused road arcs is also exhausted, the SRG algorithm selects O-D pairs randomly, with a probability proportional to the relative trips demanded between that O-D pair over the sum of all trips demanded between all O-D pairs.
4.3.2 Optimization Problem Formulation

We have two alternate formulations, one that minimizes the total length of the route network and a second one that attempts to maximize the allocation of bus-minutes to road arcs with the most volume of flow. The problem is solved on a directed graph $G (N, A)$ consisting of a set $N$ of nodes and a set $A$ of directed arcs. $R$ is the choice set of all routes generated using the modified SRG algorithm, as described in section 4.3.1 above.

Parameters:

- $\delta_a$, has value 1 if arc $a$ is part of route $r$, 0 otherwise.
- $v_a$ is the volume of flow (from the shortest paths all-or-nothing trip assignment) on each arc $a$.
- $l_r$ is the length (in in-vehicle travel time minutes) of route $r$.
- $B$ is the capacity of a single bus vehicle. We assume this to be 90 persons.
- $T$ is the length of the time period for which we are optimizing the network. In our case, because we are using 5 to 7 pm, $T = 120$ minutes.
- $M$ is the maximum available bus-minutes for allocation. Based on the existing Singapore bus network data from GTFS, we calculated $M = 363,929.74$ bus-minutes.

Decision variables:

- $m_r$ is the number of bus-minutes allocated to route $r$.
- $f_r$ is the frequency (number of buses per minute) of route $r$ and is related to $m_r$ by the following equation:
  \[ f_r = \frac{m_r}{Tl_r} \]
- $y_r$ is a binary variable with value 1 if route $r$ is included in the route network, 0 otherwise.

4.3.3 Formulation 1

Maximize:

\[ \sum_{a \in A} (v_a \sum_{r \in R} \delta_{ar} f_r) \]
subject to:

\[ \sum_{r \in R} B \delta_{ar} f_r T \geq v_a, \forall a \in A \]  \hspace{1cm} (1)

\[ \sum_{r \in R} T l_r f_r = M \]  \hspace{1cm} (2)

\[ f_r \geq 0.1 y_r, \forall r \in R \]  \hspace{1cm} (3)

\[ f_r \leq 0.5 y_r, \forall r \in R \]  \hspace{1cm} (4)

\[ y_r \in \{0,1\}, \forall r \in R \]  \hspace{1cm} (5)

The objective function is to maximize the product of bus frequency on an arc and the flow volume (from the all-or-nothing trip assignment) on that arc summed over all arcs. The idea is to reduce waiting times for the most-heavily used arcs.

Constraint 1 ensures that there is sufficient capacity on the arc for the volume of flow expected on that arc.

Constraint 2 ensures all bus-minutes available are used.

Constraint 3 and 4 set the frequency to achieve a minimum and maximum headway for routes of 2 and 10 minutes respectively. The minimum headway value is equal to the minimum headway in the existing Singapore network, while the maximum value of 10 minutes is taken from the Service Provision Standards dictated by the Public Transport Council of Singapore.\(^4\)

Constraint 5 defines \( y_r \) as binary variables.

The advantage of this formulation is that it simultaneously selects the routes to be included in the route network and the frequencies for those routes. The disadvantage is as a mixed integer problem, it could prove computationally costly to solve to optimality.

---

\(^4\) Public Transport Council, 'Quality of Service (QoS) Standards for Basic Bus Services', http://www.ptc.gov.sg/FactsAndFigures/QOS.htm
4.3.4 Formulation 2

Minimize:

\[ \sum_{r \in R} l_r y_r \]

subject to:

\[ \sum_{r \in R} \delta_{ar} y_r \geq b_a, \forall a \in A \tag{6} \]

\[ y_r \in \{0, 1\}, \forall r \in R \tag{7} \]

In this formulation, the objective function minimizes the total length of the route network.

\( b_a \) is 1 if \( v_a > 0 \) and 0 otherwise. Hence, constraint 6 ensures that all arcs with flow are included in at least one route in the final route network.

This formulation is clearly much simpler than the previous; however like Network Generation by Repeated SRG, it requires route frequencies to be determined separately. One advantage of this formulation is that it is equivalent to creating a route network where all bus routes are to have the same frequency with the maximum constant frequency, if that is a desired design objective.

4.4 Frequency Setting

Where necessary, frequency is set by solving an optimization problem using MATLAB’s fmincon function using the following function options:

1. Algorithm: 'interior-point'
2. MaxFunEvals: 500000
3. TolFun: 1e-3

We introduce a new parameter \( v_r \), which is the total number of trips between O-D pairs that can be found on route \( r \). The problem is defined as follows:

Minimize:

\[ \sum_{r \in R} \frac{v_r}{2f_r} \]

subject to:
The objective function is constructed to minimize the waiting time experienced by travellers that could be served by the route without transfers.

Constraint 8 ensures sufficient buses are allocated to each route to provide capacity for the total volume of trips on the route.

Constraint 9 ensures that all bus-minutes available are used. C however is set to be some arbitrarily large number such that there are sufficient bus-minutes for constraint 8 to be satisfied. To make the solution comparable to other networks, we then scale down the number of bus-minutes allocated to each route by the factor $M/C$, where $M$ was defined earlier in this section.

Such a formulation minimizes the correct variable, which is waiting time, rather than frequencies. However, the problem is no longer linear, and is thus computationally costly to solve.

\[
TB_{fr} \geq v_r, \forall r \in R
\]  

\[
\sum_{r \in R} Tbf_r = C
\]
5. Trip Assignment

In order to evaluate the performance of the route networks we have generated, we need to perform a trip assignment. Trip assignment generally involves assigning each traveller a path through a graph representing the public transport network, and aggregating flows of travellers on each arc in order to determine total travel time, and possibly other performance measures on the network. The path assigned to the traveller is determined according to assumptions made on traveller behaviour. Typically, it is assumed that the traveller tries to minimize the total disutility incurred during the trip. Mathematically, this disutility can take the form of a linearly weighted sum of factors, such as (but not limited to) in-vehicle travel time, waiting time, walking time, ticket fare and transfer penalty.

The most basic of trip assignment algorithms is the shortest path all-or-nothing trip assignment. For each origin-destination (O-D) pair, the shortest path on the graph is found and all trips between that O-D pair are assigned as flow volumes onto the arcs that make up the shortest path. While this may model traveller behaviour for car-users reasonably well, assuming no congestion or road speeds independent of traffic volume, it is not sufficiently realistic for public transport trip assignment. The key difference with buses is that there may be multiple feasible paths from the origin to the destination and the traveller’s decision may vary based on various factors such as which bus service arrives first, or their knowledge of the bus schedule.

Dial [4] came up with an algorithm to assign all travellers between a given O-D pair to the path in the route network with the minimum expected travel time. This model for trip assignment assumes constant in-vehicle travel times and independently and exponentially distributed headways for all bus routes. This algorithm is therefore able to account for the different wait times possible due to routes having differing frequencies. When multiple paths have equal expected travel time, the algorithm then splits the total number of trips between the paths proportionally to the routes’ frequencies. This split is known in the literature as the frequency share rule. This algorithm is unrealistic because it fails to consider a situation where the traveller may arrive at a bus stop shortly before a bus with long headways has arrived and chooses to board it because it has a shorter travel time to the destination (given that waiting time was realized to be close to 0) than another bus route with short headways that has not yet arrived at the bus stop.

Andreasson [1] algorithm can be seen as an attempt to address this by making it more likely for the model to assign trips to more than one route by introducing the idea of a choice set for travellers between a given O-D pair. All trips are split among the routes in the choice set according to the frequency
share rule. The choice set is determined by adding routes with in-vehicle travel time from the origin to destination shorter than the minimum amongst the sum of the full headway and in-vehicle travel time of each route already in the choice set. This accounts for exactly the situation we have described in the previous paragraph.

Spiess and Florian [16] reuse this idea of a choice set, calling it the “set of attractive routes”. The key traveller behaviour assumption in this model is that travellers, at every stop (whether their origin or a transfer stop), will always board the first bus that arrives that is part of the set of attractive routes for their stop to the destination. It is assumed that they know what this set is a priori. This allows allocation of trip volumes by the frequency share rule. This leads to a definition of attractive routes as routes which by their inclusion in the set, reduce the expected total travel time for the traveller. This setup allows Spiess and Florian to develop a polynomial-time algorithm that they prove is user-optimal under the assumptions of traveller knowledge. The polynomial time algorithm is a key reason for our adoption of this trip assignment model as we are going to be working with a large network of bus stops (4584) and hundreds of routes. The traveller behaviour assumption also does not seem too onerous.

The Spiess and Florian trip assignment algorithm has been implemented in the transport planning software EMME/2 and subsequent versions, which are widely used in many cities around the world. This gives further support to our use of this algorithm as the results will be widely accepted and comparable. Because we did not have the license for the software, we implemented the algorithm in Java using binary heaps where possible to reduce computational time. For a more in-depth description of the Spiess and Florian algorithm, we would refer the reader to the 1989 paper.

With the use of additional node labels, we are able to not only output expected total travel time, but expected waiting time and expected number of boardings (or transfers) and assess our networks based on these measures. It is also possible to determine these performance measures for individual O-D pairs, however, due to the large number of O-D pairs and memory constraints; our implementation requires that the user specify for which destination nodes he/she would like pair-specific performance measures.
6. Data Sources

Data for this research was obtained through the Singapore-MIT Alliance for Research and Technology from the Land Transport Authority of Singapore and consists of two datasets, Google Transit Feed Specification (GTFS) data and EZ-Link data.

The General Transit Feed Specification is a format for public transportation schedules developed by Google to standardize the data shared by public transport agencies and operators. Full details of the specification can be found at https://developers.google.com/transit/gtfs/. As some data files specified are optional, the files included in the dataset obtained can be found in Appendix I. Of these files, the two required for this research were stops and stop_times.txt. The GTFS data was used to generate a “road graph” with roads traversed by existing bus routes as arcs between nodes which represent bus stops in the existing network.

The EZ-Link card is a contactless stored-value card system mainly used for payment of fares on Singapore’s public transportation systems because April 2002. Because travellers are required to both “tap in” and “out” when paying for public transport services with the card, the data collected on trips made include both the boarding and alighting stop. The dataset consists of all public transportation trips made by stored value cards (EZ-Link) and concession passes on buses and trains from 12 pm to 2 am on the following days: 2011-02-22; 2011-04-23; 2011-04-28; 2011-05-09; 2011-05-17; 2011-05-18; 2011-05-26; 2011-05-27; 2011-05-28. The data fields found in the data set are detailed in Appendix II. Each row of data represents a single “ride” (trip segment) on either a bus or train. A “journey” (trip) may consist of multiple rides, provided the rides meet the transfer conditions for Singapore’s distance-based fare system. The conditions are as follows:

- Pay with an EZ-Link or NETS Flashpay stored value card;
- Make up to 5 transfers within a single journey, with a 45-minute allowance between each transfer;
- Take up to 2 hours to complete your journey;
- Enter and exit the train network only once in a journey; and
- Do not take the same bus service number more than once in a journey.  

---

According to LTA, there were 3,988,000 daily passenger-journeys on average per day in 2011. As an estimate of the representativeness of the EZ-Link data, we note that there were 3,435,212 rides in the data set for 26 May 2011, which gives an estimate of 86.1% of journeys being recorded in the EZ-Link data. Ideally, we would have the data of all trips including those paid for by cash, but because of the difficulty of recording the trips paid for by cash, the EZ-Link data was used to generate the origin-destination (O-D) matrices for our research.

6.1 Bus Stop Data

From stops.txt, we were able to extract the names, codes and geographical locations (latitude and longitude in decimal degrees) of all bus stops in the Singapore bus network. From stop_times.txt we were able to determine the stop itineraries for bus routes. The first and last stop of every route were then assumed to be valid start and end termini for routes because we do not have further information on which stops are actual bus terminals in reality. This bus stop data was collated into one data table with five fields, namely:

1. stop code;
2. whether the stop was a start terminal;
3. whether it was an end terminal;
4. the latitude; and
5. longitude of the stop.

The stops were sorted by stop code and their resulting positions in the table were used as a unique identification number (ID) primary key for each stop. The final bus stop data consists of 4584 nodes, of which 52 are only valid start terminals, 43 are only valid end terminals and 41 are both valid start and end terminals for routes.

6.2 Road Network Data

The road network was inferred from the bus itineraries in stop_times.txt. MATLAB was used to parse the data file and all sequential pairs of stops in a given bus route are inferred to be road arcs. The difference in the scheduled time of arrival for a bus at the two stops that define an arc is then taken to be the in-vehicle travel time on that road arc. Because there were multiple itineraries for a given bus route and also

---

arcs that were traversed by more than one bus route, arcs with multiple costs were found. These were averaged out to obtain the final values used in our research. The road arcs data table has three fields:

1. the ID number of the ‘from’ node;
2. ID number of the ‘to’ node; and
3. in-vehicle travel time on that arc.

A notable fault with the data is the coarseness of travel time measurements. The smallest unit of measurement of scheduled inter-stop arrival times, which we take as arc in-vehicle travel times, is a minute. As Singapore’s bus stops are relatively densely located, the rounding errors could prove to be significant. This is slightly mitigated by the averaging of travel times for arcs with multiple travel times.

Some arcs had travel times of 0 minutes; these were all set to an arbitrary non-zero cost of 0.1 minutes as MATLAB’s sparse function, used to reduce the memory space required to store the road arc data, ignores dense matrix entries with a value of 0.

It would be best to have data for the full road network and hence all possible arcs that connect a bus stop to another without passing by an intermediate stop. Instead, because we are inferring the road network from the stop itineraries, the ‘road arcs’ we have inferred are only those that are served by the existing bus network. Depending on the road coverage of the existing network, using the inferred subset of roads may be a significant constraint on the solution space of the route networks to be designed. Determining the actual percentage of roads that are covered by Singapore’s bus network was determined to be beyond the scope of research. The final road arc data consists of 6531 arcs.

6.3 Origin-Destination Matrices

The O-D matrices used in this research were obtained by parsing the EZ-Link data using MATLAB. As we are interested in designing the bus network, Rapid Transit System (RTS) trips were ignored by the parser. Because the EZ-Link data only covers the period from 12 pm to 2 am, the only peak we could use is the PM peak. To ensure fair comparison with the existing Singapore bus network, we use the PM peak as defined by the frequency settings for the network found in frequencies.txt of the GTFS data, which is between 5 pm and 7 pm. The parsed rides were then sorted by Journey ID. We assume that the boarding stop of the first ride of the journey and the alighting stop of the last ride of the journey are the “true” origin and destination of the traveller.

The assumption about journeys raises several considerations about the data. Firstly, there might be a significant number of journeys that are mixed-mode, using both bus and trains. This is a problem because we are now designing a bus network to service trips that passengers would normally partially
travel using train (which can cover long distances rapidly and with low travel time variance). It would be interesting to redo this study with trip data consisting of bus-only trips. Also, some journeys were found to start and end at the same bus stop. These trips are probably best explained by travellers who are leaving their point of origin to run quick errands and then returning to their origins within 45 minutes. The EZ-Link system then treats such trips as a single journey due to the transfer conditions mentioned earlier. Because such round-journeys would have no effect in a trip assignment, we ignore them when constructing the O-D matrices.

Upon generation of the O-D matrices, we noted that there were some nodes in the O-D matrices that were not connected in the road graph obtained from GTFS stop_times bus itinerary data. Upon examination, we found that these bus stops were located in the Tuas industrial area of Singapore, and concluded that these must have been recently built bus stops, i.e. there was a time discrepancy between the EZ-Link, GTFS stops data and GTFS stop_times data. The unconnected stops and associated trips were removed from the O-D matrices and bus stop data array.

6.4 Existing Singapore Bus Network

The existing Singapore bus network was constructed based on the GTFS data. The stop itineraries were determined from the stop_times file using MATLAB to parse the data. As the Singapore network has routes whose itineraries vary at different times of the day, we used the itinerary active during the PM peak of 5 pm to 7 pm. Bus routes that did not have frequencies during this period were concluded to be inactive and thus not part of the network that we would use to benchmark the performance of our algorithmically-generated routes which are designed for PM peak travel. Although Singapore numbers bus routes that travel in opposite directions with the same service number, we treat them as separate routes for the purposes of this research. The existing Singapore bus network during PM peak periods was found to consist of 401 bus routes with minimum, maximum and average headways of 2, 50 and 10.03 minutes respectively. The total route network length was found to be 17355.17 minutes. The minimum, maximum and average route lengths were 8, 120.35 and 43.28 minutes respectively. These numbers can also be found in Table 7-1.
7. Results and Discussion

For network generation by repeated SRG, the key parameters of maximum route length and maximum route circuitry were set to 180 minutes and 2.0 respectively. A trip assignment on the route networks generated was then performed using trip data for 26 May 2011.

We first generated networks using only one design criterion. The results are summarized in Table 7-1. We observe that the route network designed to minimize route lengths has the lowest total travel time, followed by the network designed according to unmet demand only. Only these two design criteria have total travel times comparable to the benchmark set by the existing network, found in the first row of Table 7-1.

It is interesting to note that most of the difference in total travel time performance arises from the difference in waiting time on the network. While total in-vehicle time of all travellers does vary, the variance is on the order of a million minutes. The network with the largest amount of total waiting time has almost 6 times as much waiting time as the network with the smallest amount.

However, we note that all the networks generated are unable to match the benchmark for boardings. While a trip on the existing network requires an average of 1.65 boardings, i.e. between one and two transfers, trips on all the networks we generated required more than 2 boardings on average.

The worst performing network was the one that used the number of routes as the sole design criterion. This results in a network where transfer opportunities are concentrated at one point, forcing travellers to travel via the hub and incurring extra travel time.

Because unmet demand and length were the two most effective criteria at reducing total travel time on networks individually, we decided to explore if a combination of the two would be more effective. We also explored the relative weights of the two criteria that would minimize total travel time on the network. The results are summarized in Table 7-2.

We found that a relative weight of 1 to 10 for the weights for unmet demand and length minimized the total travel time. We note that this resulted in the route network with the lowest total route length even though it did not have the lowest average route length. Again, differences in performance seem to be mostly due to differences in waiting time. The total waiting time experienced on the route network correlates with the total length of routes in the network. This makes intuitive sense because the lower the network length, for a given fleet size, more buses can make repeat trips on routes more often, reducing headway. The next two route networks generated were attempts to determine if the demand
criteria can substitute for unmet demand and whether delay can substitute for length to produce similarly well-performing networks. The results show that this is not possible.
<table>
<thead>
<tr>
<th>Demand</th>
<th>Unmet Demand</th>
<th>Length</th>
<th>Delay</th>
<th>Routes</th>
<th>No. of Routes</th>
<th>Total Route Dist.</th>
<th>Total Route Dist.</th>
<th>Avg. Headway</th>
<th>Total Travel Time</th>
<th>Total Waiting Time (WT)</th>
<th>Total Boarding Time</th>
<th>Total In-Veh. Time (IVT)</th>
<th>WT %</th>
<th>IVT %</th>
<th>Avg. Boardings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing Network</td>
<td>401</td>
<td>17355.17</td>
<td>43.28</td>
<td>10.03</td>
<td>8709720</td>
<td>3433614</td>
<td>775434</td>
<td>5276106</td>
<td>39.42</td>
<td>60.58</td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing Network (Frequencies Set by Algorithm)</td>
<td>401</td>
<td>17355.17</td>
<td>43.28</td>
<td>6.40</td>
<td>7381910</td>
<td>2228636</td>
<td>785104</td>
<td>5153274</td>
<td>30.19</td>
<td>69.81</td>
<td>1.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>472</td>
<td>48276.68</td>
<td>102.2</td>
<td>16.36</td>
<td>14960236</td>
<td>8923989</td>
<td>1113653</td>
<td>6036247</td>
<td>59.65</td>
<td>40.35</td>
<td>2.37</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>318</td>
<td>24048.31</td>
<td>75.62</td>
<td>8.27</td>
<td>9465467</td>
<td>4065272</td>
<td>1055367</td>
<td>5400195</td>
<td>42.95</td>
<td>57.05</td>
<td>2.25</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>807</td>
<td>12568.69</td>
<td>15.57</td>
<td>4.55</td>
<td>8444889</td>
<td>2220012</td>
<td>1051649</td>
<td>6224877</td>
<td>26.29</td>
<td>73.71</td>
<td>2.24</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>619</td>
<td>48400.80</td>
<td>78.19</td>
<td>16.13</td>
<td>15579104</td>
<td>8747530</td>
<td>1100978</td>
<td>6831574</td>
<td>56.15</td>
<td>43.85</td>
<td>2.34</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>625</td>
<td>58708.82</td>
<td>93.93</td>
<td>19.77</td>
<td>18401407</td>
<td>11324050</td>
<td>1174992</td>
<td>7077357</td>
<td>61.54</td>
<td>38.46</td>
<td>2.50</td>
</tr>
</tbody>
</table>

Table 7-1. Results for Single Design Criterion Networks
<table>
<thead>
<tr>
<th>Demand</th>
<th>Unmet Demand</th>
<th>Length</th>
<th>Delay</th>
<th>Routes</th>
<th>No. of Routes</th>
<th>Total Route Dist.</th>
<th>Average Route Dist.</th>
<th>Average Headway</th>
<th>Total Travel Time</th>
<th>Total Waiting Time (WT)</th>
<th>Total Boardings</th>
<th>Total In-Veh. Time (IVT)</th>
<th>WT %</th>
<th>IVT %</th>
<th>Avg. Boardings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Network</td>
<td>401</td>
<td>17355.17</td>
<td>43.28</td>
<td>10.03</td>
<td>8709720</td>
<td>3433614</td>
<td>775434</td>
<td>5276106</td>
<td>39.42</td>
<td>60.58</td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Existing Network (Frequencies Set by Algorithm)</td>
<td>401</td>
<td>17355.17</td>
<td>43.28</td>
<td>6.40</td>
<td>8709720</td>
<td>3433614</td>
<td>775434</td>
<td>5276106</td>
<td>39.42</td>
<td>60.58</td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1000</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>333</td>
<td>22248.21</td>
<td>66.81</td>
<td>9.19</td>
<td>9560487</td>
<td>4089923</td>
<td>1054928</td>
<td>5470565</td>
<td>42.78</td>
<td>57.22</td>
<td>2.25</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>322</td>
<td>21457.24</td>
<td>66.64</td>
<td>8.81</td>
<td>9491490</td>
<td>4010603</td>
<td>1059885</td>
<td>5480887</td>
<td>42.25</td>
<td>57.75</td>
<td>2.26</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>340</td>
<td>17997.80</td>
<td>52.93</td>
<td>6.72</td>
<td>8645438</td>
<td>3163890</td>
<td>1065128</td>
<td>5481548</td>
<td>36.60</td>
<td>63.40</td>
<td>2.27</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>416</td>
<td>14014.28</td>
<td>33.69</td>
<td>6.86</td>
<td>8361441</td>
<td>2706618</td>
<td>1088854</td>
<td>5654823</td>
<td>32.37</td>
<td>67.63</td>
<td>2.32</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>590</td>
<td>11627.34</td>
<td>19.71</td>
<td>4.54</td>
<td>7964302</td>
<td>2054862</td>
<td>1054666</td>
<td>5904400</td>
<td>25.80</td>
<td>74.20</td>
<td>2.24</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>783</td>
<td>12332.29</td>
<td>15.75</td>
<td>4.48</td>
<td>8408348</td>
<td>2191229</td>
<td>1058449</td>
<td>6217119</td>
<td>26.06</td>
<td>73.94</td>
<td>2.25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>806</td>
<td>12533.99</td>
<td>15.60</td>
<td>4.55</td>
<td>859518</td>
<td>2233181</td>
<td>1055584</td>
<td>6226337</td>
<td>26.40</td>
<td>73.60</td>
<td>2.25</td>
</tr>
<tr>
<td>0</td>
<td>1000</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>328</td>
<td>23561.80</td>
<td>71.83</td>
<td>8.17</td>
<td>9459071</td>
<td>4002701</td>
<td>1054856</td>
<td>5456370</td>
<td>42.32</td>
<td>57.68</td>
<td>2.24</td>
</tr>
<tr>
<td>0</td>
<td>100</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>328</td>
<td>23075.11</td>
<td>70.35</td>
<td>10.25</td>
<td>9838981</td>
<td>4348346</td>
<td>1052835</td>
<td>5490635</td>
<td>44.20</td>
<td>55.80</td>
<td>2.24</td>
</tr>
<tr>
<td>0</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>365</td>
<td>22276.27</td>
<td>61.03</td>
<td>8.38</td>
<td>9198916</td>
<td>3798412</td>
<td>1019629</td>
<td>5394104</td>
<td>41.33</td>
<td>58.67</td>
<td>2.17</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>450</td>
<td>197377.72</td>
<td>43.86</td>
<td>7.15</td>
<td>8575083</td>
<td>3255722</td>
<td>988103</td>
<td>5319361</td>
<td>37.97</td>
<td>62.03</td>
<td>2.10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>554</td>
<td>18939.16</td>
<td>34.19</td>
<td>7.85</td>
<td>8704477</td>
<td>3332846</td>
<td>998092</td>
<td>5371631</td>
<td>38.29</td>
<td>61.71</td>
<td>2.12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>558</td>
<td>18637.89</td>
<td>33.40</td>
<td>6.74</td>
<td>8418073</td>
<td>3020951</td>
<td>995628</td>
<td>5397122</td>
<td>35.89</td>
<td>64.11</td>
<td>2.12</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>561</td>
<td>18760.73</td>
<td>33.28</td>
<td>6.73</td>
<td>8434923</td>
<td>3022843</td>
<td>996556</td>
<td>5412080</td>
<td>35.84</td>
<td>64.16</td>
<td>2.12</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>505</td>
<td>34359.26</td>
<td>68.04</td>
<td>11.99</td>
<td>12585170</td>
<td>6672486</td>
<td>1141887</td>
<td>5912684</td>
<td>53.02</td>
<td>46.98</td>
<td>2.43</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>742</td>
<td>18782.14</td>
<td>25.31</td>
<td>9.10</td>
<td>9847097</td>
<td>3719755</td>
<td>1144321</td>
<td>6127342</td>
<td>37.78</td>
<td>62.22</td>
<td>2.44</td>
</tr>
</tbody>
</table>

Table 7-2. Results for Dual Criteria Networks
Of all the route networks generated, only one was able to outperform the existing Singapore bus network. To determine if this was due to the topological shape of the network or the frequency setting process, we reassigned frequencies on the Singapore network using our method described in section 4.4, keeping the route itineraries the same. The modified Singapore network outperformed all other networks, including the unmodified Singapore network. This is despite it not having the lowest total route length or average route distance. It also managed to keep the average number of boardings under 2.

We hypothesize that network generation by repeated single route generation underperforms the benchmark due to its failure to consider network effects.

For network generation by optimization, we generated a choice set of 2000 routes. This number was chosen because of memory constraints affecting the maximum size of the constraint matrix and the computational intractability of the optimization problems of larger sizes. The SRG parameters were set as follows:

- Demand: 0
- Unmet demand: 1
- Length: 10
- Delay: 0
- Number of routes: 0
- Maximum route length: 100
- Maximum route circuity: 1.2

We noted earlier that shorter route lengths correlate with better network performance. Therefore, we reduced the maximum route length and circuity. We assigned two different sets of frequencies to the route network generated using formulation 1. The first was equal frequencies on all routes, while the second set was generated using the method described in section 4.4. We also assigned two different sets of frequencies to the route network generated using formulation 2. The first was the solution given by the optimization solver, the second was also generated using our method. The results of the trip assignment on these networks are summarized in the following table:
<table>
<thead>
<tr>
<th>Network</th>
<th>Total Route Dist</th>
<th>Avg. Route Dist</th>
<th>Avg. Headway</th>
<th>Total Travel Time</th>
<th>Total Waiting Time (WT)</th>
<th>Total Boarding Time</th>
<th>Total In-Veh. Time (IVT)</th>
<th>WT %</th>
<th>IVT %</th>
<th>Avg. Boardings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Network</td>
<td>17355.17</td>
<td>43.28</td>
<td>10.03</td>
<td>8709720</td>
<td>3433614</td>
<td>775434</td>
<td>5276106</td>
<td>39.42</td>
<td>60.58</td>
<td>1.65</td>
</tr>
<tr>
<td>Existing Network (Frequencies Set by Algorithm)</td>
<td>17355.17</td>
<td>43.28</td>
<td>6.40</td>
<td>7381910</td>
<td>2228636</td>
<td>785104</td>
<td>5153274</td>
<td>30.19</td>
<td>69.81</td>
<td>1.67</td>
</tr>
<tr>
<td>Formulation 2 (Equal Frequencies)</td>
<td>23585.91</td>
<td>64.27</td>
<td>7.78</td>
<td>9368069</td>
<td>4252129</td>
<td>1093503</td>
<td>5115940</td>
<td>45.39</td>
<td>54.61</td>
<td>2.33</td>
</tr>
<tr>
<td>Formulation 2 (Frequencies Set by Algorithm)</td>
<td>23585.91</td>
<td>64.27</td>
<td>8.10</td>
<td>9270044</td>
<td>4148213</td>
<td>1090774</td>
<td>5121831</td>
<td>44.75</td>
<td>55.25</td>
<td>2.32</td>
</tr>
<tr>
<td>Formulation 1</td>
<td>30189.01</td>
<td>67.99</td>
<td>8.78</td>
<td>11064453</td>
<td>5911524</td>
<td>1107899</td>
<td>5152929</td>
<td>53.43</td>
<td>46.57</td>
<td>2.36</td>
</tr>
</tbody>
</table>

Table 7-3. Networks Generated by Set Covering
Unexpectedly, the networks generated using the second approach performed worse than earlier networks and the benchmark network. We hypothesize this is due to the limited size of the choice set, which in turn was constrained by computational resources and solution time.

We have found that route lengths and total route network distance have the greatest effect on the total travel time performance of route networks. Reducing route overlap also improves network performance. Even the best performing network generated only improved total travel time slightly at the cost of greatly increased number of boardings required.
8. Conclusions and Future Directions

In this thesis, a methodology for examining the relative importance of various design criteria in designing bus networks was proposed and implemented. We were motivated to do so by the prospect of gaining greater insight into bus route network performance and using such insight to design better performing route networks, whether heuristically or by optimization. Review of the literature and a preliminary case study convinced us that optimization methods, while powerful, did not give sufficient answers as to what made certain networks better than others. We were interested in determining which criteria are more important for better performing networks and the relative importance of these criteria to one another.

We first implemented a single route generation (SRG) procedure that transparently uses quantifiable measures of the design criteria of interest to generate routes. This SRG was then used in two distinct network generation approaches to generate route networks for experimentation. The first approach generated single routes repeatedly until all origin-destination pairs were connected. The second approach first generates a choice set of routes, then selects a subset of these routes as the route network using a couple of variations of a set covering optimization problem. We hypothesized that the second approach would be necessary in order to consider network effects.

We found that minimizing route length and route overlap (measured as unmet demand) were the two most important and effective design criteria. Some of the route networks we generated using these two criteria were able to marginally outperform the benchmark in total travel time but at the cost of inconveniencing many travellers with additional transfers. This suggests that while our algorithms may be able to generate closer-to-optimal solutions, when confronted with a single objective, human planners have the edge in designing to achieve multiple objectives.

We also demonstrated the effectiveness of only reallocating route frequencies without restructuring the network, and showed that the modified benchmark network outperformed the unmodified one. This suggests that public transport operators or regulators may be able to improve network performance by simply adjusting frequencies, which may be politically more feasible than redrawing routes. Our findings also lent credence to the importance of solving the network design problem from the network level and incorporating frequency setting in the solution method.

Further work could be done using better road data to expand the solution space and obtain solutions we might have missed. Also, this work could be repeated with the metro rail system modelled to take that popular mode into account. With more computational resources and time, it should also be possible to use the set covering approach to network generation with a far larger choice set to obtain
solutions of better quality. If other design criteria could be conceptualized and measured, further research to include those criteria could also be done. Finally, it would be of great interest to adapt this study for networks served by vehicles of different passenger capacities.
Appendix I: List of files in GTFS bus data set for Singapore

1. agency.txt
2. calendar.txt
3. calendar_dates.txt
4. frequencies.txt
5. routes.txt
6. shapes.txt
7. stop_times.txt
8. stops.txt
9. trips.txt
# Appendix II: Data fields of EZ-Link data

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journey ID</td>
<td>The unique number for a journey.</td>
</tr>
<tr>
<td>Card ID</td>
<td>The coded number for the stored value card used.</td>
</tr>
<tr>
<td>Passenger Type</td>
<td>The card type the commuter used – Adult, Senior Citizen and Child (including students).</td>
</tr>
<tr>
<td>Travel Mode</td>
<td>The transport mode of the ride – Bus or RTS.</td>
</tr>
<tr>
<td>Srvc_number</td>
<td>Bus service number if it is a bus ride; NULL for RTS ride.</td>
</tr>
<tr>
<td>Direction</td>
<td>Direction of bus route if it is a bus ride; NULL for RTS ride.</td>
</tr>
<tr>
<td>Boarding_STOP_STN</td>
<td>Boarding bus stop for a bus ride, or station of entering rail system for a RTS ride.</td>
</tr>
<tr>
<td>Alighting_STOP_STN</td>
<td>Alighting bus stop for a bus ride, or station of leaving rail system for a RTS ride.</td>
</tr>
<tr>
<td>Ride start date</td>
<td>The date of a ride started. The field is NULL if the commuter did not tap for alighting.</td>
</tr>
<tr>
<td>Ride start time</td>
<td>The time of a ride started. The field is NULL if the commuter did not tap for alighting.</td>
</tr>
<tr>
<td>Ride distance</td>
<td>The ride distance in km. The field is NULL if the commuter did not tap for alighting.</td>
</tr>
<tr>
<td>Ride time</td>
<td>The time interval (minutes) between the boarding and the alighting of a ride. The field is NULL if the commuter did not tap for alighting.</td>
</tr>
<tr>
<td>FarePaid</td>
<td>Fare paid for the ride.</td>
</tr>
<tr>
<td>Transfer_number</td>
<td>The transfer sequence number of a journey.</td>
</tr>
</tbody>
</table>
References


