METHODOLOGY FOR FINDING AN OPTIMAL ROUTE BASED ON CROWDING AND TRAVEL TIME

A Project Report

submitted in partial fulfilment of requirements

for the award of the degree of

BACHELOR OF TECHNOLOGY in

CIVIL ENGINEERING and

MASTER OF TECHNOLOGY in

INTERDISCIPLINARY DATA SCIENCE

by

SAI NIKHIL DONDAPATI

CE15B072

TRANSPORTATION ENGINEERING DIVISION

DEPARTMENT OF CIVIL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

CHENNAI-600036

June 2020

THESIS CERTIFICATE

This is to certify that the thesis titled **Methodology for Finding an Optimal Route based on Crowding and Travel Time**, submitted by **Sai Nikhil Dondapati**, to the Indian Institute of Technology Madras, for the award of **Bachelor of Technology in Civil Engineering and Master of Technology in Interdisciplinary Data Science**, is a bona fide record of the research work done by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Karthik K Srinivasan
Project Guide
Professor
Dept. of Civil Engineering
IIT Madras, 600036

ACKNOWLEDGEMENT

I express my sincere and whole-hearted thanks to my project guide, Dr. Karthik K Srinivasan for his relentless support throughout the project. The implementation of many ideas put forward in this thesis would never have been possible without his guidance. He always inspired me and motivated me to do better. I am forever grateful to him for providing me with a wonderful opportunity to learn and for making me the better person I am today.

I would also like to thank the project review committee members Dr. Gitakrishnan Ramadurai, Dr. Benny Raphael and Dr. Bhargava Rama Chilukuri for their valuable suggestions during review meetings steering this project in the right direction.

I am also grateful to IIT Madras for offering me a very talented peer group, a diverse student community, and a lot of wonderful opportunities to jumpstart my career. My heart holds a special place for all the amazing friends who always supported me and brought serenity and happiness to my soul. I would like to specially mention research scholars Aravinda and Sethu and my batch mates Rahul, Chinmai and Sai Kiran.

Lastly, I would like to thank my parents and my sister for offering me unconditional support and constant encouragement since the day I was born.

ABSTRACT

With the increased urbanization, traffic congestion in Chennai is becoming a major problem. Therefore, it is necessary that more people utilize the extensive public transit system instead of relying on private vehicles or cab services. However, one of the main factors that discourages people from using public transit is overcrowding especially during the peak hours. The Union Transport Ministry reported that Chennai had the most crowded buses in India with 1300 passengers per bus in each direction per day. During peak hours, in some routes, buses are filled with passengers twice the maximum capacity of the vehicle. Therefore, there is a need to provide the passengers with more information about the set of optimal routes which are better in terms of crowding and travel time as a whole. Several studies have shown that the usage of public transportation will be better if people are given real-time updates about various transit factors that affect the comfort of their journey. There is a need to estimate the crowding levels in buses and route-level travel time in a city with no Automatic Passenger Counting machines and limited GPS devices. This study aims to provide a detailed methodology for estimating and predicting crowding, estimating and predicting travel time using ETM data and GPS data. This study also aims to provide a multi-objective optimization algorithm to minimize both crowded duration and travel time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Need for the study:

With the growing population across the globe, environmental pollution and climate change are two major problems that needs to be addressed. Emissions from vehicles contribute significantly to air pollution and noise pollution. Therefore, public transit plays an important role in mitigating pollution.

However, with the increased urbanization and growing population, in-vehicle crowding and congestion only got worse. Crowding in buses and metros is a discouraging factor for people to use public transportation services instead of using personal vehicles or cab services.

When the crowding in buses increase, the buses cannot be operated according to the schedule which will adversely affect the arrival times of buses. Irregular arrival times will result in increased waiting time which will further increase crowding and that leads to more frequent bus bunching which is a much more serious problem. Average waiting times increases with the bus bunching problem and travel time reliability cannot be obtained.

The perceived waiting times can be moderated and passenger satisfaction can be improved significantly if real time crowding information is made available to the passengers. With increased density of standing passengers, passenger safety and theft security also get compromised.

Therefore, there is a need to study crowding in public transportation systems in order to enhance traveller's satisfaction and encourage the use of public transport. This study aims to provide a detailed methodology for estimating and predicting crowding, estimating and predicting travel time using ETM data and GPS data. This study also aims to provide a multi-objective optimization algorithm to minimize both crowded duration and travel time.

1.2 Problem Statement:

Improving the current crowding situation in buses by estimating and predicting crowding levels on a given route.

Finding an optimal route in terms of crowing and travel time for a given O-D pair.

1.3 Objectives & Scope:

This study mainly has the following objectives.

- To formulate the shortest path optimisation problem (in terms of crowding) by deciding the decision variables, constraints and objective function.
- Provide detailed methodology for crowding estimation, crowding prediction, travel time estimation, and illustrate the methodology for one selected route
- To develop a Multi-objective optimization algorithm to find optimal path(s) in terms of crowding and travel time.

Travel time prediction is outside the scope of this study.

1.4 Structure of Thesis:

This thesis is organized as follows:

Chapter 1 gives the general overview of the project and the importance of conducting this study. This chapter gives the motivation behind this study and lists the specific objectives of the project.

Chapter 2 reviews the literature dealing with various aspects of the study and discusses how different studies dealt with the topics in consideration. It also discusses the differences in this study compared to other studies on similar cases.

Chapter 3 gives the Overview of methodology for this study. Chapter 4 discusses the extraction of ETM and GPS data.

Chapter 5 provides a detailed methodology for implementation of crowding estimation and prediction. It also provides an illustrative application for both crowding estimation and prediction.

Chapter 6 provides a detailed methodology for implementation of travel time estimation and prediction.

Chapter 7 provides a detailed methodology for implementation of Multi-objective optimization for crowding and travel time. It also provides an illustrative application of the Multi-objective optimization algorithm developed in this study. The findings of this study are summarised in Chapter 8.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature on Defining Crowding

Before modelling the crowding effects, there is a need to define crowding in public transportation systems.

Whelan and Crockett (2009) estimated the crowding multiplier for railway services as a function of either the load factor or the number of passengers standing per square metre.

Load factor was defined as the ratio of the total number of passengers inside a vehicle to the seating capacity in the vehicle.

Oldfield and Bly (1988) used the nominal capacity of a vehicle to measure the load factor.

However, the load factor concept doesn't accurately capture the discomfort of passengers standing in a vehicle. Standing passengers density is a good indicator of crowding discomfort as it is independent of size or capacity of a vehicle.

Yunqi Zhang et al. (2019) defined Passenger Standing Density Evaluation Standards as shown in Table 2.1.

In the Transit Capacity and Quality of Service Manual, Classification of crowding service quality is given as shown in Table 2.2.

K.W. Axhausen & B. Theler (2013) conducted a survey in Zurich to study the crowding perception among the passengers. Three bus lines (32, 80, 912) that have a capacity problem in certain parts of their course were selected. The bus stops are towards the start and end of these stretches.

In total 505 surveys were distributed (275 at line 32, 157 at line 80 and 73 at line 912) in the course of three late afternoons in third week of November 2012. Three weeks later 215 useable returns had been received. A response rate of 43% was recorded.

The survey included the following details:

- Trip details (time, boarding and alighting stop)
- Personal details (age, gender, season ticket ownership, frequency of bus use, purpose of the current trip, presence of luggage)
- Assessment of the utilisation of the bus at alighting and the reasons for this assessment; assessment of the whole trip
- Choice of the “full” bus among a set of images
- Stated preference exercise asking if the bus was full given the three images shown; each representing a third of the trip.
- Stated choice bus route exercise varying in-vehicle time, degree of utilisation, seat for the passenger, headway
- Space for open comments

The survey has shown that frequent and experienced users develop a higher tolerance for the conditions of their journeys, probably both to reduction in their cognitive dissonance as well as they might not have alternatives. The older and younger users, even if frequent users are not so tolerant.

While the younger users might currently have no alternative their generally increasing income trajectory opens these up in the future. Public transport operators make a mistake when they take them for granted in the future in services with a strong share of such user, e.g. buses and street cars to schools and universities. The older users regain their alternatives, certainly through fewer schedule pressures in their lives.

Service Level	Comfort Level	Standing Passenger Density (people/m^2)	Description
A	Very Comfortable	3	Satisfy the passenger's psychological comfort requirements and the comfort space is just in contact.
B	Comfortable	5	Passengers can move slightly in the vehicle to meet body comfort requirements and the passenger's body will come into contact.
C	Generally Crowded	6	Passenger's body can maintain a standing position, and the passengers can be in contact with each other without squeezing, which can satisfy the basic space.
D	Crowded	7	Passenger's body remains squeezed and have a sense of crowding, but there is no safety issue.
E	Very Crowded	7.5	Passengers squeeze each other and feel uncomfortable. They may cause safety problems.
F	Unbearable	9	Passengers need to break through the seat area and squeeze into the seat area. It is extremely crowded and unbearable. In addition, boarding and alighting bus become difficult. It is an extreme situation.

Table 2.1 Passenger Standing Density Evaluation Standards

QOS	Loading Frequency	Description
A	0-0.5	Passengers can randomly choose seats
B	0.5-0.8	Passengers can appropriately choose seats
C	0.8-1	All passengers have seats but they are less selective
D	1-1.25	20% passengers need to stand, but passengers still have personal space
E	1.25-1.5	There are only 1/3 passengers need to stand, some of them have contact, and there is pressure.
F	>1.5	Passengers are obviously crowded and have a strong sense of oppression

Table 2.2 Classification of crowding service quality

2.2 Literature on Crowding effect on in-vehicle time

If the number of people in a bus or train is less, then all the passengers are able to find a seat, transfer of passengers at bus stations or train stations is smooth and any passenger disruptions which result in unexpected delays are rare. But with the increase in the number of passengers, there comes a situation when passengers need to stand. With this, difficulty arises for the passengers to board or alight from a vehicle thereby increasing the riding time due to friction or crowding effects among passengers.

Lin & Wilson (1992) estimate dwell time models for light rail trains in the Massachusetts and find a statistically significant friction effect between passengers alighting and those standing at stations to board, and between passengers boarding and those that are standing inside trains. Non-linear dwell time models found to fit the observed data.

In the case of buses, models that show how crowding levels increase boarding and alighting times have been empirically estimated using data from several cities around the world including Santiago de Chile (Gibson et al. (1997)), Chicago (Milkovits (2008)), Dhaka (Katz and Garrow (2012)), Vancouver (Fletcher and El-Geneidy (2013)) and Sydney (Tirachini (2013)).

Milkovits (2008) finds that dwell time increases with the square of the number of standees inside a bus, multiplied by the total number of passengers boarding and alighting at a bus stop. Along these lines, Fletcher and El-Geneidy (2013) find that the crowding effect increases dwell times once 60% of the occupancy of vehicles is reached.

Studies have also shown that crowding inside buses cause more problem to alighting than for boarding. Fernandez (2011) performed laboratory experiments with a full-size bus model in London and found out that the average alighting times increase exponentially and average boarding times increase linearly as a function of density of passengers inside the vehicle (from 1 to 6 passengers per square metre).

In particular, average boarding and alighting times are lower than 1.9 s/pax for densities lower than 4 pax/m^2, however with a density of 6 pax/m^2, average boarding time is 2 s/pax but average alighting time escalates to 5.9 s/pax, explained by the difficulties of alighting passengers walking among too many standees.

On the other hand, the study of Tirachini (2013) uses empirical data from buses in Sydney and estimates that average boarding and alighting times increase 0.34 and 0.56 seconds per passenger, respectively when there are passengers standing in the bus aisle relative to uncrowded conditions.

Katz and Garrow (2012) found out that bus design factors (e.g., front seating area, placement of doors, fare collection system) influence the amount of people that stand near doors, which has a larger impact on increasing dwell times than the number of passengers standing in aisles.

For example, on buses with two doors with one door at the front, having the second door at the middle of the bus significantly increases the crowding effects due to standees than having the second door at the back of the bus.

The limited capacity of bus stops and train stations may also represent a problem if a large volume of passengers needs to be handled at the same time, particularly in those stations in which many bus services stop. In such cases, some passengers may take longer to reach a door to board a vehicle if several other people are standing in his/her way, or obstructing

his/her line of sight to sign and approach an incoming bus (TRB (2003), Jaiswal et al. (2007).

Passengers inside buses may also face difficulties leaving a vehicle if the station is crowded. Lin and Wilson (1992) estimate the marginal friction effect between passengers alighting and those standing at stations to board, while Gibson et al. (1997) in Santiago de Chile and Jaiswal et al. (2010) in Brisbane find that the boarding time per passenger also depends on how congested is the platform at bus stations.

2.3 Literature on Crowding effect on waiting time

When the number of passengers is low relative to the capacity of a public transport route, users are able to board the first vehicle that arrives at the station. But when the occupancy rate is high, probability of the buses or trains in some sections running with full capacity increases. When this happens, some passengers cannot board the bus and have to wait for the next one. This increases the waiting time and also adds to the discomfort of the travellers.

Oldfield and Bly (1988) proposed that average waiting time is related not only to the headway (the inverse of bus frequency), but also to the occupancy rate or crowding level in an additive or multiplicative way.

Spiess and Florian (1989) considered that the travel cost per link is a function of the passenger flow, to internalise the fact that waiting time and in-vehicle comfort may be a function of how many passengers use the service.

Cominetti and Correa (2001) and Cepeda et al. (2006) model waiting time as inversely proportional to the effective frequency, which is a function of the actual frequency that decreases with the occupancy rate of buses upstream of a bus stop.

Kurauchi et al. (2003) uses assignment model to study passenger's behaviour regarding what line or service to use, and therefore, be more prone to choose routes in which occupancy levels are lower, as a way to reduce the chance of failing to board a bus.

In real-world applications, the increase in waiting time due to capacity constraints has been considered in the estimation of public transport load and demand in large scale scenarios

including London (Department of Transport (1989), Maier (2011)), Winnipeg, Stockholm and Santiago de Chile (Florian et al. (2005)), Los Angeles and Sydney (Davidson et al. (2011)) and San Francisco (Zorn et al. (2012)).

Another serious problem because of high occupancy and increased waiting time is bus bunching. When a bus gets full and more passengers wait for next bus, the next bus gets delayed because of increased loading. This increases the headway with respective to the next bus ahead and reduces the headway with respective to the next bus behind resulting in bus bunching. This phenomenon gets amplified as buses advance along the route.

2.4 Literature on Crowding effect on travel time reliability

When the occupancy of buses or trains approaches capacity, there might be an increase in both waiting and in-vehicle times. The inherent randomness of public transport demand, however, makes those delays difficult to predict.

When occupancy rates are always low, users know that they will board the first bus that approaches their stops; nevertheless, when the occupancy rate is high, passengers do not know for sure if the next bus will have spare capacity or will be full, implying having to wait for at least another bus, i.e., there might be an increase in waiting time.

This is a source of unpredictability of travel times, which adds to the generalised cost of travel beyond an increase in average waiting time, because a higher variability in travel times is negatively valued by travellers as shown by the growing body of research on the valuation of travel time variability and reliability (e.g., Senna (1994), Bates et al. (2001), Bhat and Sardesai (2006), Li et al. (2010), Börjesson et al. (2012)).

A second issue worth of note is the likely relationship between high occupancy levels and the occurrence of incidents at bus stops or train stations, which is a source of unexpected delays that affect the service performance and reliability.

A common example of this situation is the case of passengers blocking the closing of doors in trains in order to enter a crowded carriage, thereby introducing an extra delay in the process of closing doors (that might include several seconds for safety reasons).

2.5 Literature on Crowding effect on well-being

The impacts of the crowding phenomenon on passengers' health and wellbeing is extremely complex to analyse. Attached to the discomfort of sharing a limited space with several people are multiple physical and psychological factors that intervene in the perception of crowding and its effects.

Travellers in crowded conditions experienced increased anxiety (Cheng (2010)), stress and feeling of exhaustion (Lundberg (1976), Mohd Mahudin et al. (2011)), perceptions of risk to personal safety and security (Cox et al. (2006), Katz and Rahman (2010)), feelings of invasion of privacy (Wardman and Whelan (2011)), propensity to arrive late at work (Mohd Mahudin et al. (2011)) and a possible loss in productivity for passengers that work while sitting on a train (Fickling et al. (2008), Gripsrud and Hjorthol (2012)).

Cantwell et al. (2009) found out that crowding is a significant source of dissatisfaction for public transport users in Ireland, by using a stated choice experiment in which respondents had to choose between rail and bus alternatives with different levels of crowding.

Cheng (2010), who finds by means of a psychometric method, the Rasch model, that crowding is the factor that causes the most anxiety in rail commuters in Taiwan.

Using data from Kuala Lumpur, Mohd Mahudin et al. (2011) found that commuters with greater levels of stress and exhaustion attributed to crowding, reported more somatic symptoms like headaches, tension, stiff muscles and sleeplessness.

The propensity to be late at work is found to be a spill over effect of rail crowding, in cases in which passengers have to let an overcrowded train pass (or decide to do so in hope that the next train will be less crowded, sometimes unsure of the exact time the next train will arrive).

Mohd Mahudin et al. (2012) presents the most comprehensive empirical study on the psychological dimensions of rail crowding to date. The experience of passenger crowding is characterised by three different psychological components.

- evaluation of the psychosocial aspects of the crowded situation (including the items unpleasant, disturbing, cluttered, chaotic, dense, disorderly and confining)
- evaluation of the ambient environment of the crowded situation (including the items hot, smelly, stuffy and noisy)
- affective reactions to the crowded situation (including the items irritable, frustrated, tensed, distracted, stressful, hindered, restricted, uncomfortable and squashed).

The authors conclude that the link between rail passenger crowding and the negative outcomes is mediated by affective feelings of crowdedness.

We can therefore conclude that the existing empirical data and related models show the detrimental effect of crowding on travelling comfort and general wellbeing, which in turn is expected to influence travel decisions such as mode, route and departure time.

2.6 Literature on Crowding effect on route and bus choice

The disutility of standing aboard public transport vehicles may influence bus and route choice when passengers have multiple alternatives to complete a trip.

Sumalee et al. (2009), Leurent and Liu (2009), Hamdouch et al. (2011) and Schmöcker et al. (2011), estimated the probability of getting a seat both when boarding a bus, and once on board if a passenger has to stand at the beginning of his/her trip.

Passengers choose departure time and route according to their perceived travel disutility, which includes the probability of getting a seat (or failure to do so) as a key attribute.

Numerical applications show that perceived seat availability may have a significant influence on departure time, route choice and bus choice.

Leurent and Liu (2011) found out that the predicted passenger load in the Paris metro is reduced by around 30 percent when applying a model with different seat/stand disutilities, relative to a model that does not distinguish sitting from standing.

Raveau et al. (2011) show that the occupancy rate of trains is significant in explaining route choice in the metro network of Santiago de Chile, and that the effect is non-linear; it increases for very high occupancy rates when users perceive that, apart from the discomfort effect of crowding, there are higher chances of not being able to board the first train

Kim et al. (2009) use stated choice data from Seoul to estimate the probability of passengers waiting for a second bus if they are provided with real time information on the occupancy level of the next bus arriving at a bus stop; with results showing that the availability of seats increases the probability of a passenger to choose boarding an arriving bus.

In other words, up to a point, some passengers are willing to trade waiting time for an expected higher chance of getting a seat while travelling.

2.7 Literature on Crowding effect on optimal public transit supply and fare

Crowding as a factor that affects the users' utility and generalised cost of travelling has been recognised by several authors in the analysis of public transport pricing and supply policy (Jansson (1979), Kraus (1991), Jansson (1993), Arnott and Yan (2000), Huang (2002), Pedersen (2003), Pels and Verhoef (2007), Parry and Small (2009)).

The basic idea is that when a person boards a bus or a train, they may impose a crowding externality on everyone else on board, which is especially noticeable when there are passengers standing.

Therefore, the crowding externality raises the marginal social cost of travelling, thus increasing the optimal bus or rail fare, which is obtained as the difference between total marginal cost and average users cost on first best pricing (Tisato (1998), Jara-Díaz and Gschwender (2005)).

In the economic literature of public transport, it is usually proposed that when users' waiting time cost is included in the total cost function of public transport services, the marginal cost pricing rule does not cover operator cost due to the positive effect of increasing frequency in reducing waiting time for users (Mohring (1972), Turvey and Mohring (1975), Jansson (1979)).

Mohring (1972)'s well-known square root formula, which states that an increase in demand is optimally met by a less than proportional increase in supply. Therefore, as demand grows there is an increase in the occupancy rate or load factor inside vehicles, that is, an increase in density, and possibly, crowding effects.

In summary, the acknowledgement of a crowding externality on the valuation of travel time and on travel time itself might have significant effects on the design of a public transport system, particularly in terms of the capacity provided to serve demand.

When the crowding cost is ignored, policy makers may choose to provide a transport capacity that is just enough to meet demand, in which buses would be full (or close to full if a safety level of spare capacity is defined by design) in the most loaded sections of a route.

Nevertheless, when the crowding cost is considered in the design stage of a route, it should be optimal to provide a greater service frequency and bus capacity in order to reduce the occupancy levels inside vehicles, and consequently improve the quality of travelling (Jara Díaz and Gschwender (2003)).

2.8 Literature on factors effecting crowding

For high frequency bus services, passenger arrival times are known to be well approximated by a Poisson process. Under such conditions, the expected number of passengers boarding a bus at a given stop is proportional to the headway to the preceding bus. Headway may therefore be used as a proxy for passenger load.

Erik Jenelius (2019) used headway and headway square of the target vehicle at the K most recent stops to make crowding prediction at the target stop.

Crowding function also should include the effects of historical patterns related to the time of day, day of the week and month of the year.

If the buses are highly crowded during peak hours, more people might be waiting at the bus stop which will increase the access time and that leads to even more crowding. Therefore, peak

hours crowding can be better predicted if access time is also considered. Similarly egress time increases with more crowding.

2.9 Literature on crowding models

Erik Jenelius (2019) in his study addressed the bus crowding prediction based on real time vehicle location and passenger count data in Stockholm city. In order to provide real time crowding information, online automatic passenger counting (APC) machines were installed in few buses. Real Time Crowding Information (RTCI) was provided based on Historical APC data and Automatic Vehicle location (AVL) data.

Lasso Regression was used by Erik Jenelius to predict the crowding in Stockholm buses along a high frequency bus line by taking various combinations of historic, AVL and APC data.
In another study conducted by Yunqi Zhang (2019) to determine the bus crowding coefficient based on passenger flow forecasting, Yunqi Zhang used RBF Neural Network along with historic crowding data.

Another research paper by Zhang and Erik Jenelius studies the impact of providing real time crowding information. During a pilot study at Stockholm metro station, the RTCI data for each car of the metro was made available to the passengers through visual display and speakers.
The impact of the crowding information provided is evaluated with the help of video analysis (by noticing passenger's attention), traveller surveys, and in-vehicle passenger load data.

RTCI reduced the share of passengers boarding the most crowded car by 4.3% points for the trains that were crowded on arrival and also increased the share of passengers boarding the least crowded car by 4.1%.
The results indicate that RTCI may be a useful technology for public by helping them deal with crowding and also the metro operating agencies by increasing the utilization of train space and reducing crowding.

Alejandro Tirachini (2013) used Multinomial logit and Error component models to study the significance of crowding on various factors important for travellers' comfort and satisfaction.

2.10 Literature on Formulating & Solving Multi-objective shortest path problems

The general Multi Objective problem requiring the optimization of N objectives may be formulated as follows:

Minimize $\bar{y} = \bar{F}\ (\bar{x}) = [\ f_1(\bar{x}), f_2(\bar{x}), f_3(\bar{x}), \dots, f_N(\bar{x})]^T$

Subject to $g_j(\bar{x}) \leq 0\ , j = 1, 2, \dots, M$

Where $\bar{x} = \left[\ x_1, x_2, \dots\dots, x_p\right]^T \epsilon\ \Omega$

$\bar{y}$ is the objective vector, the $g_j's$ represent the constraints and $\bar{x}$ is a P-dimensional vector representing the decision variables within a parameter space Ω.

The space spanned by the objective vectors is called the objective space and the subspace of the objective vectors that satisfies the constraints is called the feasible space.

Bicriterion shortest path problem with a general nonadditive cost seeks to optimize a combination of two path costs. There are number of emerging transportation applications in which obtaining a shortest path based on multiple criteria is necessary ranging from in-vehicle routing to the regional infrastructural planning.

Peng Chen & Yu Nie (2013) have tried to balance two attributes of the paths under consideration while valuing one of the attributes linearly. They minimized $P_1^k + h(P_2^k)$ where h is a nonlinear function and P_i^k denotes the i^{th} property of the path k. The general nonlinear function was approximated with a piecewise linear counterpart and then each linear subproblem was solved sequentially. A specialized algorithm was developed to solve the subproblems, by making use of the efficient path set to update the upper and lower bounds of the original problem. They showed that the optimal path must be efficient if the nonlinear cost function is concave.

Dial (1979) proposes an algorithm to construct a set of efficient paths, which are such defined that no other paths provide a better overall cost for any linear combination of the attributes. Dial's algorithm has since been adopted for solving multi class traffic assignment problems that consider heterogeneity in users' valuation of time.

Henig (1985) provides a comprehensive treatment of a bicriterion shortest path problem with continuous monotone functions. Similar to Dial (1979), his algorithm also starts from finding a set of efficient paths. However, he shows that the optimal path is always efficient only if the objective function to be minimised is quasiconcave. For quasiconvex functions, a line search procedure is proposed to locate the efficient path that admits the best upper bound and to further close the gap, a search for K-shortest paths is recommended.

Mirchandani and Wiecek (1993) show that for any monotone quasiconvex function, the optimal path must be a non-dominated path. They have also refined Henig's linear search procedure for this case.

Tsaggouris and Zaroliagis (2004) developed an exact algorithm for the non-additive bicriterion shortest path problem assuming $h(.)$ to be convex and nondecreasing function. Their algorithm consists of two phases. The first solves a Lagrangian relaxation of the original problem, which is equivalent to computing the best efficient path. Phase 1 ends with an optimal solution or a duality gap. If the optimal solution found is not efficient, the second phase of the algorithm closes it using a path enumeration procedure based on branch-and-bound.

Gabriel and Bernstein (1999) propose a feasible direction algorithm for solving the problem that moves around extreme points generated by a Frank-Wolfe type linear subproblem. Whenever the next extreme point fails to improve the upper bound, a heuristic line search procedure, which itself involves solving a series of nontrivial Linear Program.

Peng Chen & Yu Nie (2013) have formulated the bicriterion shortest path problem as:

Consider a directed network $G\ (N, A)$, where N represents a set of nodes and A represents a set of links. Let d_a, τ_a, and u_a be the length, travel time, and monetary cost associated with link $a \in A$ where d_a, τ_a, and u_a are assumed to be nonnegative.

A path for a given origin-destination pair $r - s$ is denoted by using k, and the length, travel time and monetary cost associated with the path k are given by

$$l_k = \Sigma\ \delta_k^a d_a\ , t_k = \Sigma\ \delta_k^a \tau_a\ , c_k = \Sigma\ \delta_k^a u_a\ (for\ a \in A)$$

where $\delta_k^a = 1$ if link a is on path k and 0 if not. The set of all paths that connect an $O - D$ pair $r - s$ is denoted by K.

They tried to minimize $P_1^k + h(P_2^k)$ subject to: $k \in K$ where $P_i^k = \Sigma\, \delta_k^a p_i^a$ are i^{th} cost of traversing path k (here $i = 1,2$), and h is a general continuous function that transforms P_2^k that facilitates the trade-off between the two costs.

The above formulation fits into a variety of applications. For example, if one interprets P_1^k as c_k and P_2^k as t_k, the function h can be considered as an evaluation of travel time in the monetary cost. Another possibility is to consider P_1^k as t_k and P_2^k as l_k, h can be interpreted as distance-based toll measured in the unit of time.

Gabriel and Bernstein (1997) considers a general path cost of the following form

$$g_k = c_k + h(t_k)$$

where c_k is the monetary path cost, h is a nonlinear function that converts time to money. This is a typical example of non-additive traffic equilibrium problem.

Lawphongpanich and Yin (2012) came up with a distance-based congestion pricing model. They assumed that the amount of toll is a nonlinear function of the distance travelled inside tolled areas. Toll was calculated as a function of both time and nonlinear function of the distance.

It involved minimizing $\Sigma_k \left(t_k + T(l_k)\right) f_k$ where $T(l_k)$ is a distance-based toll and the function T takes the following form

$$T(l_k) = \begin{cases} T_1(l_k)\ or\ T_2(l_k) & l_k > 0 \\ 0 & l_k \leq 0 \end{cases} \qquad \begin{aligned} T_1(l_k) &= \max\{a_1 + b_1\, l\,,\ \ a_2 + b_2\, l\} \\ T_2(l_k) &= \min\{a_1 + b_1\, l\,,\ \ a_2 + b_2\, l\} \end{aligned}$$

f_k represents the flow on any path k.

Another interesting problem that involves multiple criteria is arriving at an optimal path prioritizing schedule penalty. When on-time delivery is important, decision makers may choose to impose a penalty cost on both late and early arrivals, which leads to a special instance of optimal path problem.

It involved minimizing $\alpha t_k + h(t_k)$ subject to: k ϵ K

Where $h(t_k) = \begin{cases} \theta(t_0 - t_k) & t_k \leq t_0 \\ \mu(t_k - t_0) & t_k > t_0 \end{cases}$

h(.) represents schedule cost function, and α may be interpreted as value of time

θ and μ represents the unit early and late arrival costs respectively.

To compare candidate solutions to the Multi-Objective problems, the concepts of pareto dominance and pareto optimality are commonly used. A solution belongs to the pareto set if there is no other solution that can improve at least one of the objectives without degradation of any other objective.

Formally, a decision vector $\bar{u} = [u_1, u_2, \ldots, u_p]^T$ is said to pareto dominate the decision vector $\bar{v} = [v_1, v_2, \ldots, v_p]^T$, in a minimization context, if and only if:

$$\forall i \in \{1, \ldots, N\}, f_i(\bar{u}) \leq f_i(\bar{v}), and\ \exists j \in \{1, \ldots, N\}: f_j(\bar{u}) < f_j(\bar{v})$$

In the context of Multi Objective optimization, pareto dominance is used to compare and rank decision vectors: $\bar{u}$ dominates $\bar{v}$ in the pareto sense means that $\bar{F}(\bar{u})$ is better than $\bar{F}(\bar{v})$ for all objectives, and there is at least one objective for which $\bar{F}(\bar{u})$ is strictly better than $\bar{F}(\bar{v})$.

A solution $\bar{a}$ is said to be Pareto optimal if and only if there does not exist another solution that dominates it. In other words, $\bar{a}$ cannot be improved in one of the objectives without adversely affecting at least one other objective. The corresponding objective vector $\bar{F}(\bar{a})$ is called a Pareto dominant vector, or non-inferior or non-dominated vector.

The set of all Pareto optimal solutions is called the Pareto optimal set. The corresponding objective vectors are said to be on the pareto front. It is generally impossible to come up with an analytical expression of the Pareto front.

Sushant Sharma, Satish V. Ukkusuri, and Tom V. Mathew (2011) conducted a study to formulate and solve the multi-objective network design problem with uncertain demand. A formulation was proposed for multi-objective robust network design, and a solution methodology was developed on the basis of a revised fast and elitist nondominated sorting genetic algorithm.

A real medium-sized network was solved to prove efficacy of the model and the results showed better solutions for the multi-objective robust network design problem with relatively less computational effort. It considers the trade-off between two objectives of standard deviation of total system travel time and expected value of total system travel time under demand uncertainty for optimal link-capacity expansion of a network.

Climaco and Martins (2001) propose to use a k-shortest path algorithm to rank paths according to one of the objective functions, and add a dominance test that verifies if each one of the computed paths is non-dominated.

Zhimin and Xianfeng (2003) discuss finding the optimal route with the least travel time between the source and sink nodes in a road network, by considering, the dynamic property of the network. Based on the Dijkstra algorithms, the article puts forward a new algorithm to find the shortest path within the limit of travel time in a dynamic road network. This method can be used to solve the road network with multiple dynamic parameters such as crowd, travel cost, security etc.

Mainali et al. (2008) propose a new dynamic programming method to find the optimal route to the destination considering multiple criteria. Travel time, road type and turn at intersections are considered as criteria for optimization. Q value updating algorithm to find the multi-objective optimal route to the destination. This method has some drawbacks like it is not possible to find the best optimal solution for all users because the preference of the criteria changes with users.

Fan et al. (2009) present a simple algorithm to optimize the multi-objective urban transit routing problem (UTRP). To represent the problem a transport graph which is an undirected graph with nodes as access points and edges as direct links between the nodes are used. A path in the network is called a route and the solution to the problem is defined by a set of routes from origin to destination. In addition, a transit network is constructed such that for each node in the transport network on a route there exists a node in the transit network. Two kinds of edges are present in the transit network: transport edge and transfer edge. Transport edge is between two nodes on the same route and transfer edge corresponds to transfer from one route to another. Two objectives are considered for optimization: min. passenger cost and min. operator cost. The list of feasible and also the best route set for both objectives are found using

the shortest path algorithm like Dijkstra's or Floyd's in the transit network and then the proposed algorithm based on SEAMO algorithm is used to modify an existing route set to produce another feasible route set.

Antsfeld L. and Walsh T. (2012) modifies the original transit algorithm for static, undirected and single mode transport network to a multi-criterion (such as travel time, ticket cost and interchanges) time-expanded network. The authors used a two-layer model: station graph and events graph. The events graph nodes are arrival and departure events of a station and are interconnected by four types of links: departure links, continue links, changing links, waiting links. The station graph nodes are the stations and it has two types of links: station links and walking links. The algorithm consisting of two phases - precomputation and query, is first used to solve a single objective problem of the fastest time. In the first phase, the shortest routes among every pair are precomputed and stored. Using a search algorithm like A*, in the query phase the travel time of all the routes from the origin at time t to destination are analysed and the one with minimal time is chosen as the fastest route. To deal with multi-objectiveness a linear utility function is used to reduce the problem to a single-criterion optimization.

Wen F. & Lin C. (2010) proposes a model with three objectives for the multi-objective route selection problem (MRSP). The article considers the three objectives of minimizing driving distance, minimizing driving time and minimizing driving cost, simultaneously. The solution approach consists of three phases: Pre-processing the traffic network, calculating a Pareto solution set using a multi-objective genetic algorithm, deciding a best-compromised route from the Pareto set according to the driver's preference. The pre-processing method proposed in the article reduces the computation time compared to the traditional methods.

Yuh-Wen Chen et al. (2008) developed a multi-objective geographic information system (GIS) for route selection for the safe transport of nuclear wastes. This article integrates the multi-objective shortest route problem with the actual road-network attributes of GIS instead of a simplified road network to plan an optimal route. The model objectives were to minimize travel time, minimize transportation risk and minimize the exposed population.

2.11 Literature on Crowding Applications

Search engine giant Google has recently introduced a new feature to the Google Maps that now would predict transit crowdedness of buses and trains for passengers. It has been made available in around 200 cities around the globe.

The search engine giant is basing these details on past rides and for and months and has been asking some people who use Maps to provide additional details about the level of crowdedness of their transit trips.

After completing their trips, riders were given four options: many empty seats, few empty seats, standing room only, or cramped standing room only. Now that the company has collected enough data, it has begun offering predictions to customers who use Maps to plan their daily commute.

In addition, Maps is also launching live traffic delays for buses in places where commuters do not already have real-time information direct from local transit agencies. The users would now be able to see if their buses are running late, how long would the delay last and other more accurate travel times based on the live traffic conditions along the route.

Delhi Integrated Multi-Modal Transit System (DIMTS) uses Automatic Vehicle Location system to improve the efficiency, reliability and punctuality of the bus operations in Delhi and ensure optimal deployment of the fleet which would eventually lead to higher commuter satisfaction and improved level of confidence in bus services.

The GPS enabled AVL allows real-time tracking of bus movement and provides information on its location. This information is then used along with other details such as the speed of the bus, the route followed etc to provide the passengers waiting at the bus stops with the expected arrival time of their bus. The AVL also helps in improving the efficiency of bus operation by generating various standard and exception.

DIMTS has implemented a unique self-learning algorithm for accurate calculation of ETA driven by historic travel speed data across road segments and time of day. Passenger Information System boards display traffic information at bus shelters, parking lots, crossroads, etc.

NextBus Delhi is a mobile application providing route information, live occupancy status and estimated time of arrival (exact time of the next bus's arrival with its bus number) for cluster buses in Delhi. The app also gives information regarding bus frequency, bus stop locations, and maps for better visibility to the commuter.

Swiss Federal Railways company SBB recommends users about making a reservation on the trains via SBB mobile application. Based on the historic data about travel demand, the train reservation recommendation is provided. Time of the day, Day of the week, and Month of the year data is used to make a prediction about the crowding level. Three levels of crowding: low, medium and high are shown in the mobile application. Real time location of next bus/train at a station is also made available to the passengers through the mobile application.

In March 2012, Israel-based social transportation app Moovit was launched by Nir Erez as he felt that overcrowding is a critical part of the passenger experience. The application is aimed at taking real-time user feedback on transit and making it available to a wider audience of travellers. By 2013, Moovit was operating in over 30 cities across 3 continents which includes Paris, New York, and New Jersey.

The suggestions are given based on passenger comfort information, operators schedules, live updates given as inputs by passengers, and GPS based vehicle location data. The algorithm developed by Moovit integrates user-generated and operator-provided data (both static and real time whenever available), as well as the company's statistical layer. The statistical layer is provided by app users, and logs information on travel times and distances from previous trips.

Netherlands Railways have also developed a seat availability indicator app, with a particular focus on helping passengers with mobility problems. It generates a symbol on the basis of the length of the train and its historical data on the number of people traveling at that time of the day. Feedback is also collected from the passengers to report on the accuracy of the information provided.

Crowd sensing smartphone apps like CMS, REsen, CommuniSense, UrbanEye use a new sensory paradigm, called war-driving, where volunteers contribute their phones' sensor data that can be used to analyse comfort levels.

Several public bus trip planner systems are available, like the Google Transit, TRipGo, GOTransit etc., which give information on alternative routes, fares, schedules, as well as map-based visualization of the real time traffic information.

Further, some of the personalized bus route recommendation systems like MetroCognition, PATRASH, etc. rely on specialized cards, war driving or explicit user feedback for route database generation. The existing personalized route recommender systems and tour planners, like PaRE, FAVOUR, Routeme, Feeder, Treads, etc. are mostly developed for private cabs and taxis and do not consider the wide sets of comfort parameters as experienced during public bus travels in our daily life, along with the dynamics of the environments as well as personalized choices of the commuters.

2.12 Literature on Route Recommendation Systems

Rohit Verma, Surjya Ghosh, Mahankali Saketh (2018) have developed ComfRide, a smartphone-based application that recommends the most comfortable route based on a user's preference honouring his/her travel time constraint.

Analytics Laboratory at UFCG have also developed a best trip recommender manager. The overview of their system developed is as follows. The Recommendation Service Manager holds the procedure of paralleling multiple requests to the BestTripRecommender.

The Application makes several requests to the Web Service which is responsible for distributing those requests into multiple processes, each one will execute in an individual gate of their Virtual Machine in a parallelized way. These processes receive the requests with the input data and then prediction algorithms are applied.

BestTripRecommender predicts duration of a trip and its number of passengers. The user application provides the input data necessary for the system to generate a prediction and also receives output data from the recommender.

Jiaqui Wang, Yunyao Lu (2018) propose a social aware route recommendation system (SAR), which is designed to recommend proper routes to drivers to help diminish negative mood and fatigue and help improve driving experience. SAR integrates multiple sources of sensing data from smartphones to recommend a proper route to the driver considering the driver's real time mood fatigue, history record and social information. Experiments have been conducted with SAR to demonstrate the system's effectiveness as well as reasonable computation and communication overheads on smartphones.

Garvita Bajaj, Rachit Agarwal (2016) have also developed a route recommendation system for public transit. They have introduced an android based application MetroCognition which recommends a route based on the commuter preferences and also receives feedback from them after their trip. They have defined convenience based on three parameters: time, crowdedness and seat availability.

Data used in Route Recommendation System:

The data collection module of the ComfRide app runs as a background process that collects various sensor data periodically from the smartphone sensors, like accelerometer, gyroscope, compass, periodic GPS, sound sensors etc. Segment specific features like average speed of bus from one stop to another, jerkiness, crowding, average stopping time at a bus stop, etc which vary between route segments are calculated separately for each segment.

BestTripRecommender developed at UFCG Analytics Laboratory uses the following data.
Input data contains the following content:

- Bus route
- Time
- Date
- Stop ID

The Timetable data contains the following fields based on which the trips closest to user passed are returned:

- Bus route
- Day type: days having similar bus schedules are grouped together

- Stop id
- Mean timetable: historically, the mean time at which the bus arrived at the bus stop
- Lower timetable: statistically, the earliest time at which the bus is supposed to arrive at the bus stop
- Upper timetable: statistically, the latest time at which the bus is supposed to arrive at the bus stop
- Trip start: The time at which the bus left the initial stop
- Trip end: The time at which the bus arrived at the final stop

N closest trips of the scheduled timetable passed by the user are returned.

Features are created based on the user and timetable data which contains the following information:

- The week day
- Difference between central time table and previous one
- Difference between central time table and next one
- Floor function applied on time: Grouped timetable

The prediction data contains all the historical data about accomplished bus trips during a certain period of time. It contains the following information:

- Bus route
- Date of trip
- Departure: The time of bus departure from initial stop
- Arrival: The time of bus arrival in final stop
- The week day
- Total passengers
- Time difference between durations of actual trip and the trip that happened before
- Time difference between durations of actual trip and the trip that happened after

The output data contains the most N probable trips that are closest to the user query.
For a recommendation system developed based on the user priorities, user data is also equally important. In the case of ComfRide, a large-scale online survey involving more than 300 participants across the globe was conducted and various factors like total travel time, traffic

congestion, sitting probability, road condition, bus type, number of stops, break journey, which reflect commuter comfort were assessed.

While developing SAR, User information was divided into three parts: Driver Social Context (DSC), Real-time Mood-fatigue Context (RMC) and Time Context (TC), respectively.
We can see that drivers with similar social background tend to share the favour of similar style of routes. Thus, in current stage of SAR, four attributes are chosen: nationality, gender, age and occupation.

Aggressiveness, patience, neutral, happiness, sadness, and anger. Together with fatigue, a seven-dimension vector is formed, with each dimension quantified as float data in the range of 0–1. Every dimension of the vector is the ratio of number of corresponding moods to number of all detected faces during a short period of time.

Time context is the third factor. For instance, if a driver starts his car in 5:00 PM every weekday, perhaps he drives home after work at that time. So, a relaxing route without a lot of traffic may be a great choice for him at that time. However, if someone starts at midnight without any history departure recorded at that time, maybe he/she is in a hurry, and a route of the shortest time may fit his/her situation. In the current stage of SAR, departure time of day is considered as an integer which indicates the number of seconds has elapsed from 00:00 of that day.

Algorithms used in Route Recommendation Systems:

In the case of ComfRide, Dynamic Input / Output Automata (DIOA) based composition model was used that captures both the wide varieties of comfort choices from the commuters and the impact of environment on the comfort parameters. DIOA algorithm ensures that there is no overloading on the system even in case of handling huge data. This algorithm also modifies the model to suit the personalized preference of a commuter after receiving feedback from the commuter.

Based on the historical information and context of the query, DIOA based compositional model identifies the most preferable route. Evaluation of ComfRide, involving 50 participants over 28 routes in a state capital of India, reveals that recommended routes have on average 30%

better comfort level than Google map recommended routes, when a commuter gives priority to specific comfort parameters of her choice.

Considering the factors like possibility of multiple feasible routes between a source-destination pair, possibility of opting for one or more breakpoints during the journey for improving comfort choice, and users' personalized choices on route parameters (like travel time, road condition, crowdedness), the state space of the network graph can increase exponentially. To overcome this problem, they have used DIOA algorithm which can interact with the external environment and internal entities like historical data and context information.

Methodology used in the Route Recommendation Systems:

ComfRide uses Fuzzy set theory-based recommendation technique along with TOPSIS. TOPSIS constructs a feature matrix for various available alternatives and ranks the alternatives based on the distance similarity with the worst alternative in comparison to the best alternative. A Route Comfort Index (χ) is defined as follows. Let ρ be the feature value as computed for a route, and ω is the feature weight obtained from the internal signature. The feature values are calculated from the crowdsourced data collected from the commuters and are normalized over different routes based on TOPSIS feature normalization. Then $\chi = \Sigma \rho * w$. Now, for every route, χ is calculated from the feature matrix, and the route having the highest rank based on TOPSIS distance similarity over χ is recommended. When the user opts for a break journey, the average of route comfort index of all route segments that the commuter follows will be considered.

The ComfRide DIOA consists of two components - the client automaton (runs at the Smartphone app) and the server automaton. DIOA requests the clients to provide a feature ordering and weights based on the personalized choice. The first filter removes unfeasible routes based on the query context (like time of the day, spatial characteristics of the route etc.). The server automaton computes the occurrence density (number of occurrences per km of the route) of all the features of a route. The set of candidate routes includes all the feasible routes between the given s-d pair in the query, which conform to the user's travel time budget. They considered only the routes between the given s-d pair, where the median travel time as obtained from the historical data is not more than the historical median travel time for the least cost route

(the route with minimum travel time) for the s-d pair plus the allowed deviation specified by the commuter.

SAR development involves two major steps. First of all, the candidate route generation method was used to analyse the map to generate the candidate routes and their attributes (e.g. convenience, traffic, pedestrian, beauty). Secondly, user model was developed which is then used to process users and routes into a unified model to make the recommendation process easier.

K shortest routes between the start point and destination point are initially chosen as the candidate routes. A heuristic method, which is introduced is designed to calculate the candidate routes efficiently. Then the attributes are calculated on the chosen routes for further recommendation.

For source V and destination T, the heuristic function was defined as $f(V) = h(V) + g(V)$, where $h(V)$ is the distance from S to V, and $g(V)$ is heuristic that estimates cost from V to T. The five attributes chosen for each route are distance, traffic, convenience, pedestrian, and scenery. The attributes of each route along with Map API were fetched from the internet.

The recommendation system first calculates a predicted route attribute. Then one of routes in the candidate routes which has the closest attribute is selected and return to the SAR application on driver's smartphone. Predicted route attribute is obtained based on the similarities between user's attributes as defined above.

Results obtained from the Route Recommendation Systems:

From a field trial for 2 years over 28 different bus routes in a state capital of India, they have observed that ComfRide recommended routes have on average 30% better comfort level than Google navigation based recommended routes, for various combinations of commuters' priorities to the comfort features.

Using the SAR recommendation system, the experience is the least enjoyable among 80% of the subjects when driving on a randomly assigned route. About 60% of the subjects reported

the most enjoyable experience when driving on their favourite route and 40% gave it to the automatically recommended route.

2.13 Summary of Literature Gaps

Previous studies on crowding estimation in public transit is mainly based on the data obtained from Automatic Passenger Counting machines and Automatic Vehicle Location (GPS). Some studies were also conducted based on the information obtained from transit users. However, this study aims to find out the inherent transit factors that indicate the level of crowding in buses and use that information to predict the level of crowding and recommend the least crowded transit route.

Previously, crowding is included as one of the many objectives like distance, travel time, fare, congestion, etc in order to improve public transportation systems for passengers. Some studies have also included crowding as a constraint for optimizing other main objectives like shortest route and minimum travel time. In order to give suggestions to the public transit operators or provide passengers with travel information, crowding was never considered as the main objective beforehand. This study aims to consider crowding in buses as the main objective function along with travel time and suggest a route for the passengers in which they experience least crowding in buses along that route for a given origin-destination pair. In the previous studies, whenever multi objective shortest path problems were formulated, costs of each arcs were defined quantitatively. Crowding in public transit systems is a very qualitative measure. Therefore, there is a need to formulate and optimize a multi objective shortest path problem which have costs that are qualitative (for example, low, medium, and high crowding in buses).

In order to give suggestions about the least crowded route, we need historic data about crowding information. For real time predictions, Automatic Vehicle Location data and Automatic Passenger count data is necessary. In India, GPS is installed on very few buses which makes it very difficult to make real time predictions. Ticketing information can give historic travel data only up to some extent as usage of smart cards is very limited and proper information is not available about monthly pass holders. In order to develop a robust crowding model, a lot of data is necessary. Keeping Indian scenario in mind, there is a necessity to develop a crowding model which makes decent predictions limiting the assumptions. There are

lot of countries across the globe where public participation is helping the transit agencies or third-party application developers in developing more accurate crowding models. With limited public participation, there is a need to come up with good crowding estimates based on the available data and develop models suitable to Indian conditions.

When there are two non-additive objectives to be minimized for a shortest path problem, studies have shown that Lagrangian Relaxation and Pareto Optimal Non dominated solutions give a set of efficient solutions which are close to optimal solution. However, the basic algorithms are modified according to the requirements of the objective function and constraints. Therefore, there is a need to come up with an algorithm for predicting the least crowded route. The existing algorithms need to be modified accordingly.

Studies have also shown that modelling crowding effects improve the decisions taken by the public transit operators and also the passenger's satisfaction. The applications developed previously only give information about the crowding conditions in buses. They also help the transit agencies in adjusting the schedule according to the demand. This study however aims to predict the least crowded route between a given origin-destination pair. If a mobile application is developed, a user can enter origin and destination details and get the information about which buses to take and which connections to utilize (for transfer from one bus to another). This will also help the transit agencies to figure out which parts of the network need more buses. They can also change the frequency of bus arrival along the highly crowded route.

CHAPTER 3
OVERVIEW OF RESEARCH METHODOLOGY

3.1 Flowchart for the overview of Research Methodology

The flowchart for the overview of research methodology for this study is shown in Figure 3.1

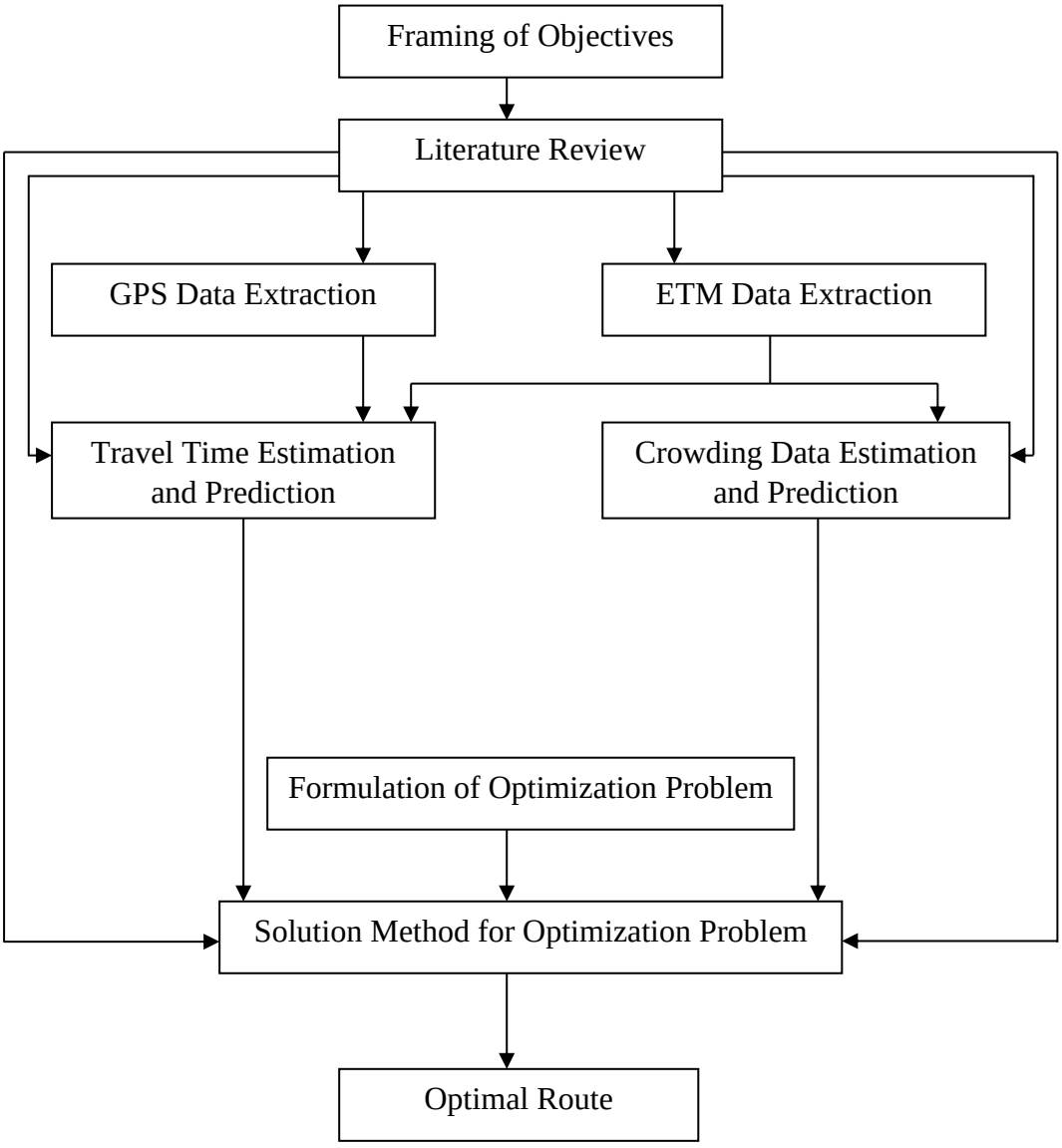


Figure 3.1 flowchart for the overview of research methodology

The Objectives were framed in Chapter 1 and the Literature Review was covered in Chapter 2. The following Chapters provide detailed methodology for Crowding estimation and prediction, Travel time estimation and prediction, and Optimization.

3.2 Overview of Data Extraction

As a first step of the data extraction process, the Raw ETM data obtained from Chennai MTC needs to be cleaned for discrepancies. The cleaned ETM data is then used for extracting the ticket data for a selected route on a given day, or for a selected trip of a particular route on a given day.

The extracted ticket data is then used as an input for Crowding estimation and prediction, Travel time estimation and prediction.

In order to know the crowding level for each link on a given route, the order in which the stages appear for that route needs to be determined.

GPS data is also extracted for some of the routes where we have both ETM data and GPS data available. The extracted GPS data is used to model the relationship between Stage to Stage travel time estimates obtained from ETM data and actual Stage to Stage travel time estimates obtained from GPS data.

3.3 Overview of Crowding Estimation

Using the extracted ETM data, for a given trip, the number of boardings at each stage of all stages belonging to the route of the trip, the number of alightings at each stage of all stages belonging to the route of the trip, and the interchanges between stages of all the links belonging to the route of the trip are estimated.

The processed ticket data along with the data of boardings, alightings, and interchanges is defined as demand data.

The demand data can be estimated for multiple trips for a selected route and on a given day. The estimated demand data is at stage-level. Here, the nodes of a selected route represent the stages and the links connecting the nodes represent the road connecting the two stages.

In order to estimate the demand data at stop-level, we need to distribute the tickets issued at each stage to the corresponding stops at that stage. This allocation of tickets to stops at each stage for a given trip is achieved on the basis of the time-stamps of the tickets issued.

Passengers who travel frequently often use bus passes issued by the Chennai MTC for daily commute. Therefore, it is important that while estimating crowding, we need to also account for pass-holders along with the ticket-holders.

Based on the interchanges between stages for a given trip, a Pass-holder model is developed which estimates the number of pass-holders who are traveling from one stage to the other stage.

Total demand data is calculated by combining the estimates obtained from ETM data and Pass-holder model.

Based on the summary statistics of multiple trips of multiple routes and on multiple days, we defined the crowding thresholds.

For a future trip, the demand at each stage (or stop) is predicted and using the state space equation, the number of people present in the bus in each link is calculated and on the basis of crowding thresholds defined, the crowding on each link of the route for that trip is categorized into low, medium, or high.

3.4 Overview of Crowding Prediction

For a selected route, the demand data is estimated for multiple trips and on multiple days. Based on the estimated demand data, some essential features that define the characteristics of a trip are extracted.

After the Features Extraction, Exploratory Data Analysis is done that identifies the features that has an effect on the number of boardings, number of alightings, and stage-stage interchanges for a given trip. In this step, the correlation between features is also identified.

For a selected route, some demand data of the previous trips is used as training data and the remaining data is used for testing. Machine Learning models are used to train and test and then are fine-tuned to make predictions for a future trip of the selected route.

Once the number of boardings, number of alightings, and the stage-stage interchanges for a future trip of a selected route are predicted, the predicted data is then input into the state space equation and the crowding level in each link is estimated.

3.4 Overview of Travel Time Estimation

Using the extracted ETM data, for a given trip, approximate stage-stage travel time is estimated. For a given trip, for each link of the route belonging to that trip, we need to find out if travel time can be approximately estimated for that link. The reason being that for some trips, tickets are not issued at every stage.

For a given trip, for the links where travel time can be approximately estimated, it is done by finding the time difference between the first ticket issued at the From Stage and To Stage of that link. In this way, for a given trip of a selected route and on a given day, the travel time is estimated for all the links wherever it is possible.

However, the estimates from the ETM data are approximate and cannot be relied upon unless they are corrected for practical purposes. If we have GPS data available for the trip we are interested in, we can estimate the actual travel time between stages.

In order to find the GPS data corresponding to the ETM data of a selected route, we first need to map the ETM and GPS data to find the correct GPS file corresponding to a given route-level ETM data.

Once we have both ETM and GPS data for a given trip, we can model the relationship between the approximate stage-stage travel time estimates from the ETM data and the actual stage-stage travel time estimates from the GPS data using Machine Learning. This will help us in estimating the actual stage-stage travel time for the links of a selected route just with the help of ETM data. For many routes, we do not have GPS data available but we have ETM data. With the help of this model, we can estimate the actual travel times even if we don't have GPS data.

3.5 Overview of Travel Time Prediction:

For a selected route, the stage-stage travel time data is estimated for multiple trips and on multiple days. Based on the extracted ticket data, some essential features that define the characteristics of a trip are extracted. Along with those features, additional features like Dwell time at a stop, and Crowding level in each link of a selected route also has an effect on the stage-stage travel time.

After the Features Extraction, Exploratory Data Analysis is done that identifies the features that has an effect on the stage-stage travel time for all the links of a selected route and for a given trip. In this step, the correlation between features is also identified.

For a selected route, estimated stage-stage travel time data of some previous trips is used as training data and the remaining data is used for testing. Machine Learning models are used to train and test and then are fine-tuned to make stage-stage travel time predictions for a future trip of the selected route.

3.6 Overview of Optimization:

Once we have both the travel time data and crowding level data for each route, we have to find an optimal path in terms of both crowding and travel time between a given origin and destination pair.

As a first step of the Optimization process, single-objective optimization problem was formulated where we find the shortest path in terms of travel time. Based on whether the shortest path performs well in terms of crowding or not, multi-objective optimization problem was formulated where we try to minimize both crowding duration and travel time simultaneously.

The single-objective optimization problem will be solved using the Dijkstra's algorithm while the multi-objective optimization problem will be solved using K-shortest paths-based algorithm to obtain a set of Non-Dominated paths.

As there are three levels of crowding, the crowding coefficients are defined in different ways and ultimately crowding coefficient is brought to 0 or 1. Based on the way crowding coefficient is defined, the multi-objective optimization problem changes. For every multi-objective optimization problem defined, we find a set of Non-Dominated paths.

All the sets of Non-Dominated paths obtained are combined into a set of candidate paths and if there are any dominated paths are remaining in the candidate set, they are removed to get the final set of optimal routes.

CHAPTER 4

EXTRACTION OF ETM DATA AND GPS DATA

4.1 Overview:

In order to estimate crowding using ETM data, the first thing that needs to be done is processing the Raw ETM data obtained from Chennai MTC and extract the essential features that we need to complete this study. Similarly, we also need GPS data to estimate travel time and this chapter covers the detailed process of extracting ETM data and GPS data.

4.2 ETM Data Extraction

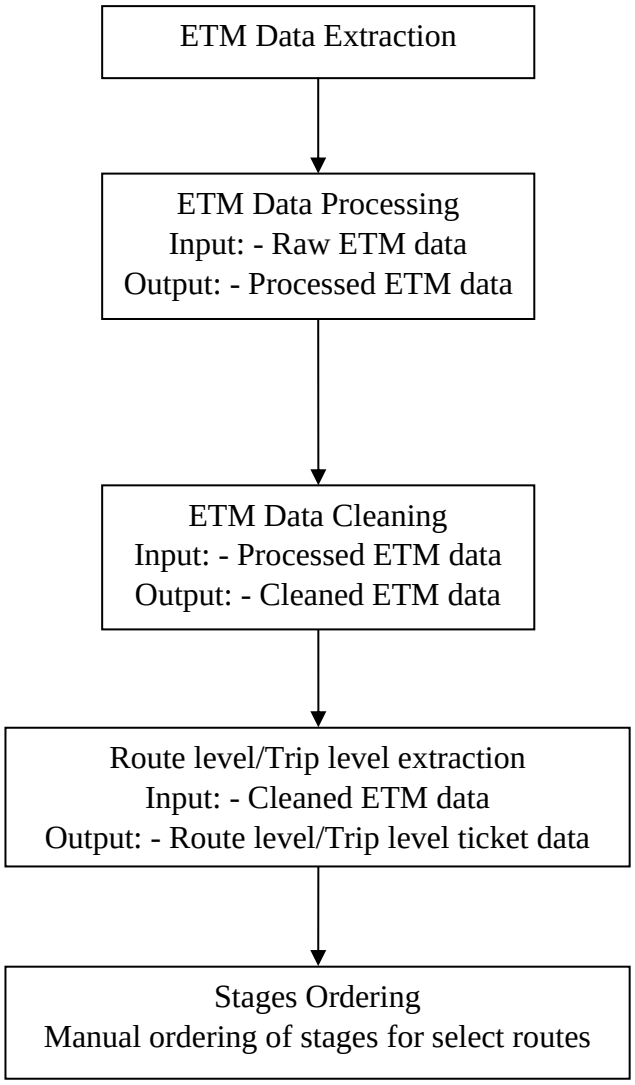


Figure 4.1 Flowchart for ETM Data Extraction

4.2.1 ETM Data Description

As of May 2017, the MTC had a scheduled fleet of 3688 buses and total fleet strength of 3968 buses, on a daily basis carries 6.0 million passengers to and from, which is half the population of Chennai. On March 22, 2016, the Union Transport Ministry reported that Chennai had the most crowded buses in the country with 1300 passengers per bus in each direction per day. During peak hours, in some routes, a bus with capacity to accommodate 80 persons carries twice the number of people due to the extensiveness of the system. It has an operating area of 3,929 square kilometres (1,517 sq mi). MTC has a total of 830 routes.

In the MTC buses, the passenger trip information is recorded using Electronic Ticket Machines (ETMs). At the end of the shift, the conductors transfer data from the ETM device to a central server.

Each data file consists of data on all the trips run in a day. The data from ETM's is comprehensive and gives details listed in Table 4.1. Each row in the data corresponds to a ticket purchased by the passenger.

Table 4.1 Headers in the raw ETM data file

Waybill No.	In Date	Ticket Issued Time	Luggage	Trip Start Date
Depot	In Time	From Stage	Ticket No.	Trip Start Time
Schedule name	Total Amount	To Stage	Trip No.	Trip End Date
ETM Device ID	Adult	Fleet No.	Concession	Trip End time
Out Date	Child	Denomination	Conductor ID	Source
Out Time	Ticket Issued Date	Ticket Type	Driver ID	Destination

Each ETM Ticket records the boarding and alighting stage of the passenger. A Stage includes the bus stops within an approximately 2 km stretch. On the basis of the information about boarding and alighting stage of the passenger, demand at each stage is estimated.

The ETM Ticket also records the time at which the ticket is issued on the basis of which approximate travel time between stages is estimated.

The following sections describe the methodology for ETM data extraction.

4.2.2 ETM Data Processing

In order to complete this study, we need only selected fields from the Raw ETM data. The required fields listed in Table 4.2, are extracted from the raw ETM data.

Table 4.2 Headers in the processed ETM data file

Date	ETM Device ID	From Stage	Trip No.	Source
Depot	Adult	To Stage	Trip Start Time	Destination
Schedule Name	Child	Fleet No.	Trip End Time	Ticket Issued Time

In this step, the input file is Raw ETM data while the output file generated is Processed ETM data.

4.2.3 ETM Data Cleaning

Once we have the Processed ETM data file, we need to look out for discrepancies if there are any and make the necessary corrections.

The processed ETM data file is cleaned for errors in stage names and for discrepancies in Trip Start Times, Trip End Times, and Ticket Issued Times. For some routes, the stage names are not used consistently or sometimes the stage name is wrong. In such cases, the stage names are corrected manually and are used commonly for all the trips belonging to that route. The time-stamps are corrected by checking the difference between Trip Start Time and First Ticket Issued time and also by checking the difference between Trip End Time and Last Ticket Issued time. After looking out for these discrepancies, suitable corrections are applied.

In this step, the input file is Processed ETM data while the output file generated is Cleaned ETM data.

4.2.4 Route Level/ Trip Level ETM Data Extraction

Once we have the Cleaned ETM data file, we can extract the ticket data for selected routes. Every bus has a unique field ‘Schedule Name’ which includes the information about the Route Number, Schedule ID, Shift, and Service Type. So, on a given day, for each route, we can extract the ticket data from the Cleaned ETM data file.

On a given day, for a selected route, multiple trips are made between origin and destination. A trip is defined as a single run between an origin-destination pair. So, the ticket data can also be extracted from the Cleaned ETM data file for one single trip.

In this step, the input is Cleaned ETM data file and the output is Route Level/Trip Level ticket data.

4.2.5 Stages Ordering:

For a selected route and a given schedule, there are multiple stages in the ETM data. However, we might or might not see all the stages between an origin-destination pair in the ticket data because ticket purchases might not happen at all the stages on a given day. Therefore, we need to have the stage list that reflects the on-ground truth. In order to estimate crowding or travel time between stages, we need to know the correct stage order especially for estimating the interchanges. Therefore, for some selected routes and schedules, stages are manually arranged in a sequential order. The stages are plotted in Google Maps and the correct order of stages is obtained.

Since some of the trips might not be covering all the stages of a given route, we have to take the superset of all the stages of all possible origin destination pairs of that route and manually arrange them in an order. In this step, we are manually arranging the stages in a sequential order.

4.3 GPS Data Extraction

4.3.1 GPS Data Description

GPS data is obtained from probe vehicles which are buses of the Chennai MTC. The frequency of capturing vehicle location data is 10 seconds. One hundred fifteen probe vehicles have been fitted with GPS devices.

There are two types of GPS devices fitted on MTC buses. The first set of devices are provided by GLOBEES vendor (GLB files) and the second set of devices are provided by a vendor from Kanpur (KNP files).

The headers of the GLB files are provided in Table 4.3 and the headers of the KNP files are provided in Table 4.4.

Table 4.3 Headers in the GLB GPS data file

ID	Track_Time	Longitude	Latitude
Speed	Status	Device ID	Long_Dir
Lat_Dir	Checksum	Direction	Fetch_Time

Table 4.4 Headers in the KNP GPS data file

IMEINO (Device ID)	Latitude	Longitude	Speed	Device Time
Server Time				

4.3.2 Trip Level GPS Data Extraction

Each GPS data file contains the latitude, longitude coordinates of the bus in which the GPS device is installed for a selected route on a given day.

The first step is to break down the entire data file for each day into multiple trips data. Since we have the ETM data in trips, it is important to correctly extract the GPS data for individual trips for a selected route on a given day.

The extraction of GPS data is outside the scope of this study.

4.4 Summary

This chapter provides a detailed description of ETM data and GPS data. Detailed methodology of extracting and processing ETM data has also been discussed. Now that we have the Route Level/Trip Level ETM data, we can use it to estimate and predict crowding. We will also use Route Level/Trip Level ETM data for estimating travel time approximately and use the GPS data for understanding the relationship between travel time estimates of ETM data and GPS data. The next chapter provides a detailed methodology for implementing crowding estimation and prediction.

CHAPTER 5

IMPLEMENTATION AND ILLUSTRATIVE APPLICATION OF CROWDING ESTIMATION AND PREDICTION

5.1 Overview:

In this chapter, we provide the detailed methodology for estimating crowding in MTC buses using the extracted ETM data. We also provide a detailed methodology for predicting crowding on selected routes based on the historical crowding estimates on these selected routes. The flow charts including the input, output, and process of all the steps involved in crowding estimation and prediction are depicted in this chapter. We also demonstrate an illustrative application of crowding estimation and prediction for the route '19B'.

The flowchart for the steps involved in this chapter are shown in Figure 5.1.

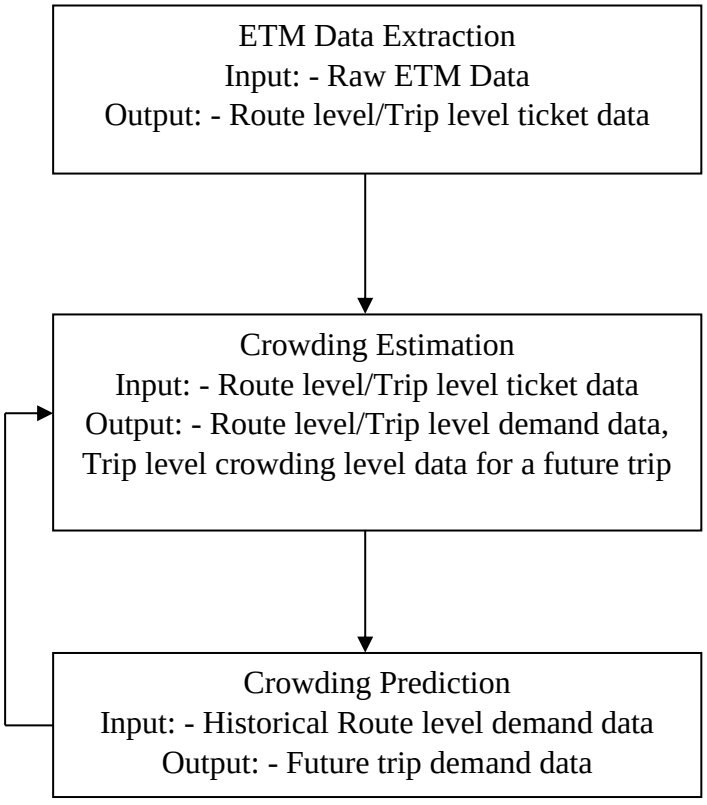


Figure 5.1 Overview Flowchart for Crowding Estimation and Prediction

5.2 Methodology flowchart for Crowding Estimation

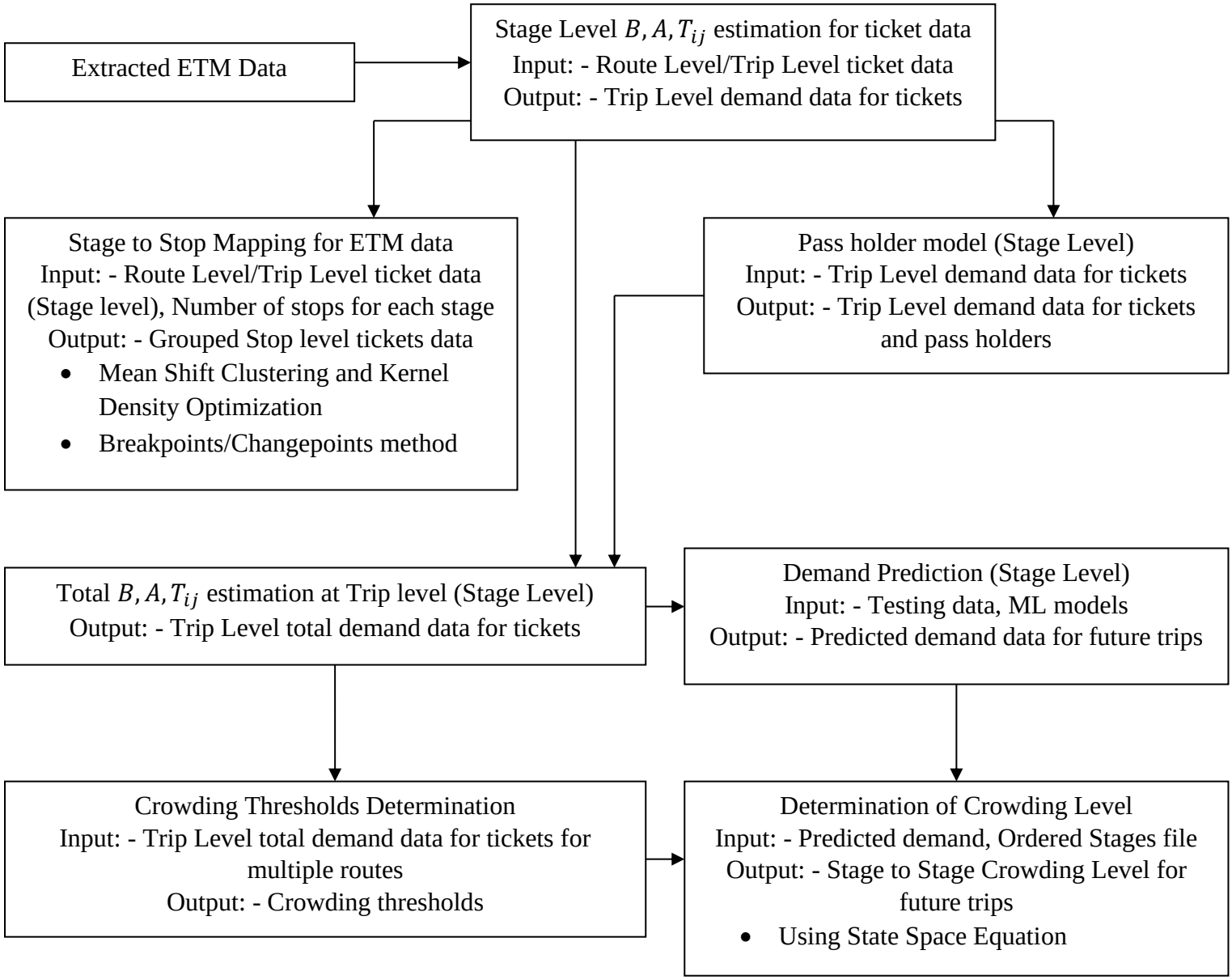


Figure 5.2 Methodology flowchart for Crowding Estimation

5.2.1 Methodology for Stage Level demand data estimation

Using the extracted route level ticket data, the first step in crowding estimation is generating stage level demand data for all the trips for the selected route on a given day. On a given day, for a selected route, there may be multiple origins and destinations. If we select a particular route, and then fix the origin and destination, we get trip level data. On a given day, there are multiple bus trips between origin and destination of a selected route.

The trip level ticket data contains the headers listed in Table 5.1

Table 5.1 Headers of trip level ticket data

Date	ETM Device ID	From Stage	Trip No.	Source
Depot	Adult	To Stage	Trip Start Time	Destination
Schedule Name	Child	Fleet No.	Trip End Time	Ticket Issued Time

In this study, the demand is estimated in terms of the number of people getting into the bus at a stage defined as the number of boardings (B), the number of people getting down from the bus at a stage defined as the number of alightings (A), and the number of people traveling from stage i to stage j defined as interchanges T_{ij}. In order to estimate the demand data for any given trip, we need to estimate the interchanges T_{ij} because the number of boardings B, and the number of alightings A can be estimated from the interchanges T_{ij} using the following equations 5.1 and 5.2

$$B_i = \Sigma_j T_{ij} \qquad (Eq. 5.1)$$

$$A_j = \Sigma_i T_{ij} \qquad (Eq. 5.2)$$

$$0 \leq i < j \leq maximum\ number\ of\ stages\ in\ the\ route$$

The number of boardings at a stage i is equal to the sum of all interchanges between stages where 'FromStage' is stage i and the number of alightings at stage j is equal to the sum of all interchanges between stages where 'ToStage' is stage j.

Initially, for each trip of a selected route, the number of boardings and the number of alightings at every stage belonging to that route has been estimated. The number of boardings at stage i is equal to the sum of 'Adult' and 'Child' columns of all the rows in trip level ticket data where the 'FromStage' is stage i. Similarly, the number of alightings at stage i is equal to the sum of 'Adult' and 'Child' columns of all the rows in trip level ticket data where 'ToStage' is stage j. The stages for the route are listed manually because for all the trips, tickets might not be issued

at every stage. Therefore, for estimating the demand at all the stages for a given route, we need an ordered stages file.

Sample boardings and alightings extraction at stage level for '19B' route between 'KELAMBAKKAM' and 'T. NAGAR' for one trip of one particular schedule on 3rd November, 2016 is depicted in Figure 5.3

	Schedule_name	Trip_no	Source	Destination	Trip_start_time	stage_name	number_of_boardings	number_of_alightings
0	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	KELAMBAKKA	23	0
1	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	HINDUSTAN	2	0
2	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	CHE.COMPAN	2	0
3	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	CHURCH	6	1
4	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	SIPCOT	0	0
5	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	NAVALUR	3	2
6	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	CHEMMANCHE	0	0
7	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	KUMARAN NG	8	1
8	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	SHOZHINGAN	1	2
9	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	KARAPAKKAM	3	2
10	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	MOOTAIKARA	0	0
11	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	THORAPPAKK	2	2
12	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	THORAIPAKK	0	0
13	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	KANDANCHAV	2	1
14	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	SRP TOOLS	2	0
15	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	M.K. CHAVA	7	1
16	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	ANNA UNIVE	0	8
17	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	WPTC	0	3
18	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	CONCORDE	0	0
19	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	SAIDAPET	0	6
20	19B-B1-AS-MAC	4	KELAMBAKKA	T.NAGAR	09:22:56	T.NAGAR	0	32

Figure 5.3 Sample boardings and alightings of a '19B' trip

Similarly, the boardings and alightings are estimated for multiple trips belonging to multiple routes and on multiple days.

In the next step, for each trip of a selected route, the interchanges between stages belonging to that route are estimated. A matrix containing stage to stage interchanges has been generated.

Sample interchanges between stages for '19B' route between 'KELAMBAKKAM' and 'T. NAGAR' for one trip of one particular schedule on 3rd November, 2016 is shown in Figure 5.4

	ANNA UNI	CHEMMAI	CHURCH	HINDUSTA	KANDANC	KARAPAKK	KELAMBAI	KUMARAN	M.K. CHA	NAVALUR	SAIDAPET	SHOZHINC	SRP TOOL	T.NAGAR	THORAPP/	WPTC	CHE.COMI
ANNA UNI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHEMMAI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHURCH	0	0	0	0	0	5	0	0	0	0	0	0	2	0	2	6	0
HINDUSTA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KANDANC	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
KARAPAKK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
KELAMBAI	3	3	2	3	1	0	0	0	0	0	0	2	2	5	0	2	0
KUMARAN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M.K. CHA\	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
NAVALUR	0	0	0	0	1	0	0	0	0	0	0	0	2	0	2	2	0
SAIDAPET	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SHOZHINC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
SRP TOOL!	0	0	0	0	0	0	0	0	0	0	2	0	0	3	0	0	0
T.NAGAR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
THORAPP/	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WPTC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHE.COMI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CONCORD	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MOOTAIK	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SIPCOT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
THORAIPA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 5.4 Sample interchanges of a ‘19B’ trip

Similarly, the interchanges between stages are estimated for multiple trips belonging to multiple routes and on multiple days.

In this step, the input is Route level/Trip level ticket data and the output is Trip level demand data for tickets.

5.2.2 Methodology for Pass-holders data estimation

Chennai MTC issues bus passes to passengers who use MTC buses for daily commute. The demand estimated from ETM data is therefore not sufficient to accurately depict the crowding levels in buses. Especially during the peak hours, it is important to estimate the number of pass-holders in the bus at any point of the trip in order to accurately estimate the crowding level in buses.

It is difficult to estimate the number of pass-holders boarding the bus at each stage without detailed on field surveys. Even if we know the number of pass-holders boarding the bus at each stage, we cannot track the destination of these pas-holders which restrains us from estimating the number of pass-holders alighting at each stage.

However, we can come up with some models that capture the number of interchanges between stages T_{ij} coming from the pass-holders. One of the commonly used models to estimate the number of trips between an origin and destination pair is Gravity Model.

Based on the productions and attractions of each stage pair, we can estimate the number of pass-holders traveling between the stages as a percentage of the interchange’s estimates coming from the ETM data for the selected stage pair.

In this study, we demonstrate the crowding prediction for '19B' and for the sake of simplicity, we assume that the number of pass-holders traveling from stage i to stage j is equal to forty percent of the number of ticket-holders traveling from stage i to stage j.

The number of interchanges estimates coming from ETM data is added to the number of interchanges coming from the Pass-holders model to get the total demand data for a selected route or trip on a given day.

In this step, the input is trip level demand data for tickets and the output is total demand data.

5.2.3 Methodology for Stage to Stop Mapping using ETM data

Chennai MTC issues tickets to passengers from one stage to another stage. However, in reality passengers board the bus at a stop and travel from one stop to another stop. A stage includes the bus stops within an approximately 2 km stretch. Therefore, it is important that we estimate the demand at stop level so that predictions can be made that reflect the reality.

More importantly, we get travel time estimates from GPS data which will be from one stop to another stop. In order to use both ETM data and GPS data to come up with more accurate estimates, we need to estimate the demand at stop level.

Additionally, the transit network contains nodes which represent the stops but not stages. Therefore, we need to estimate the crowding level on the links connecting two stops but not two stages.

Therefore, a hierarchical algorithm is proposed in order to estimate the demand at stop level using ETM data.

Hierarchical Algorithm for Stage to Stop Mapping

Using the extracted Trip level ticket data, we try to assign the tickets at each stage to the stops corresponding to that stage for a given trip of a selected route with the help of the Hierarchical Algorithm as described below.

Input: Trip level ticket data at stage level

Step 1: For each stage belonging to the route of the given trip, check whether there are tickets issued at that stage. If yes, proceed to the next step. If there are no tickets issued at that stage, stage to stop mapping is not necessary for that stage.

Step 2: Check if the tickets issued at a stage are uniformly distributed based on the time stamps of the tickets issued.

If the tickets are uniformly distributed, compute the cut-offs in terms of travel time based on the lengths between stops and assign the tickets to stops.

If the tickets are not uniformly distributed, proceed to the next step.

Step 3: If the tickets are not uniformly distributed, try one of the following methods:

- Mean Shift Clustering (number of groups not defined)
- Breakpoint method/Changepoint method (number of groups pre-defined)

If the number of groups obtained from either of the above two methods match the number of stops corresponding to that stage, we use that method to assign the tickets to each stop for that stage.

Step 4: Repeat the first three steps for all the stages in the given trip and assign all the tickets to stops to obtain the stop level ticket data.

Output: Trip level ticket data at stop level

Illustrative example of Stage to Stop Mapping

One trip of '19B' between 'KELAMBAKKAM' and 'T. NAGAR' on 3rd November, 2016 is used for demonstrating the stage to stop mapping. The ticket issued time data for all the tickets of the trip is shown in Figure 5.5

Based on the time stamps of the tickets issued, time till the issue of each ticket since departure is calculated and plotted as depicted in Figure 5.6

For each of the unique stages of this trip, Mean Shift Clustering was tried with different bandwidths. Mean Shift Clustering did not work for any of the stages because the time stamps of ticket issued are too close. However, if there are sufficient number of tickets belonging to a stage, Mean Shift Clustering will identify the clusters in the time stamps of ticket issued.

	Stage	Ticket_issued_time
0	KELAMBAKKA	15:42:30
1	KELAMBAKKA	15:40:30
2	KELAMBAKKA	15:40:19
3	KELAMBAKKA	15:39:59
4	HINDUSTAN	15:45:44
5	HINDUSTAN	15:45:38
6	HINDUSTAN	15:45:22
7	HINDUSTAN	15:45:07
8	HINDUSTAN	15:44:29
9	HINDUSTAN	15:44:26
10	CHURCH	15:53:00
11	NAVALUR	15:55:45
12	NAVALUR	15:55:41
13	NAVALUR	15:55:30
14	NAVALUR	15:54:54
15	NAVALUR	15:54:48
16	NAVALUR	15:54:28
17	NAVALUR	15:53:56
18	CHEMMANCHE	15:58:56
19	CHEMMANCHE	15:58:17
20	CHEMMANCHE	15:58:10
21	WPTC	16:37:09
22	WPTC	16:36:47

Figure 5.5 Time stamps of tickets issued at each stage

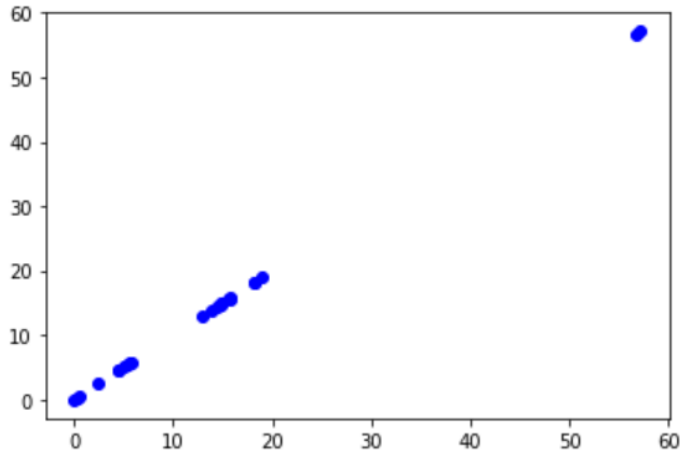


Figure 5.6 Plot of time stamps of tickets issued at each stage

Similarly, Changepoint method is tried for the stages ‘KELAMBAKKAM’ and ‘NAVALUR’. Figures 5.7 and 5.8 depicts the time since departure for all the tickets belonging to the stages ‘KELAMBAKKAM’ and ‘NAVALUR’ respectively.

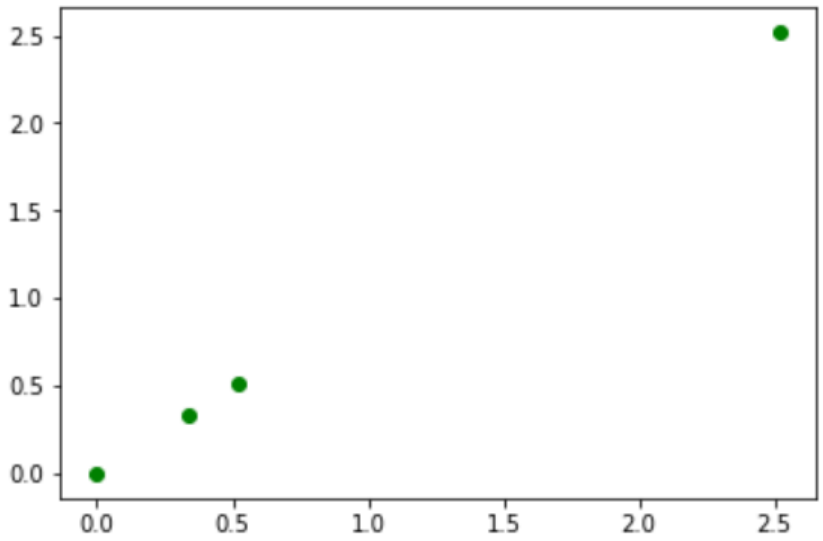


Figure 5.7 Plot of time stamps of tickets issued at stage ‘KELAMBAKKAM’

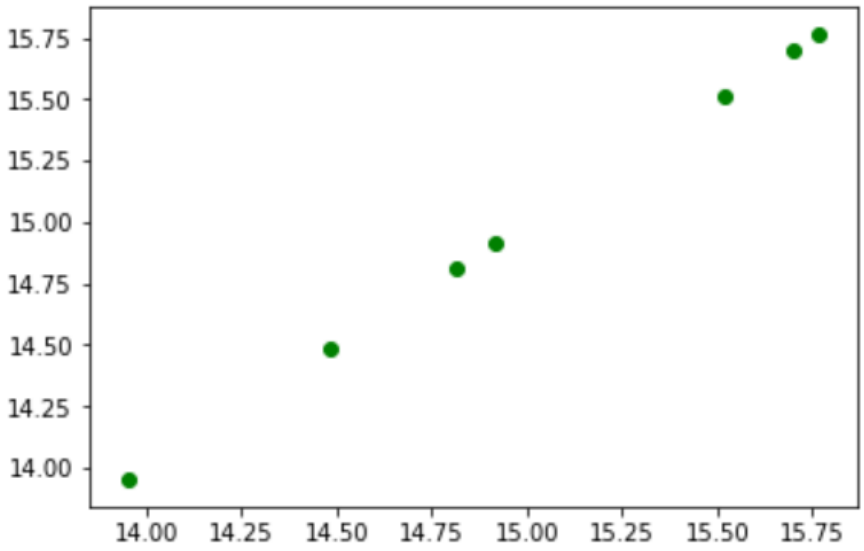


Figure 5.8 Plot of time stamps of tickets issued at stage ‘NAVALUR’

The Changepoint method could not identify any clusters for the stage ‘KELAMBAKKAM’ but could identify two clusters for the stage ‘NAVALUR’ as depicted in Figure 5.9

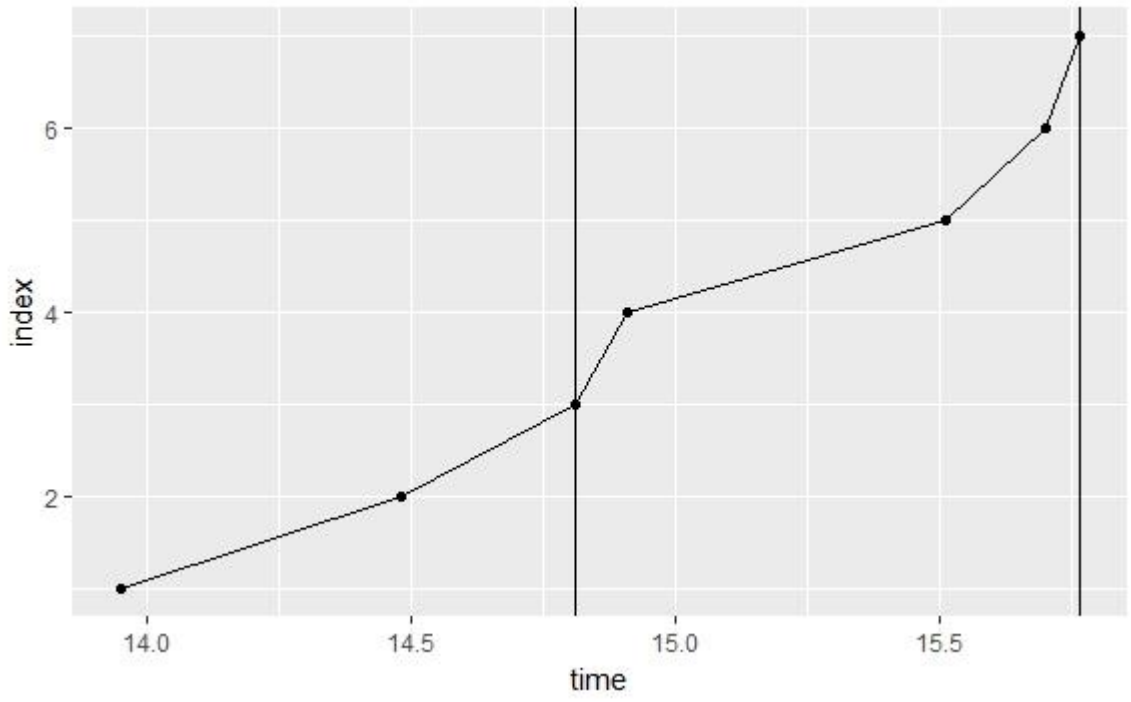


Figure 5.9 Groups of tickets obtained by the changepoint function in R Studio

It can be seen in the above figure that the changepoint function grouped the first three tickets together and the next four tickets together.

We can therefore conclude that if there are large number of tickets at a given stage, we can assign tickets at each stage to the stops corresponding to that stage using the method described in this section to obtain stop level ticket data.

Ultimately our goal is to be able to estimate stop level demand data in order to find an optimal route for a given origin-destination pair. Therefore, it is important that the hierarchical algorithm is automated and improvised so that it can be implemented in real-time estimation.

In this study, we estimate and predict the crowding data only at stage level and we limit ourselves to the methodology for stage to stop mapping.

In this step, the input is Trip level ticket data at stage level and the output is Trip level ticket data at stop level.

5.2.4 Methodology for determining Crowding Thresholds

Based on the literature review, it is clear that crowding in public transit is a very subjective measure which is quantified in different ways in various studies.

Chennai MTC buses have a seating capacity of 48. The maximum capacity of these buses is 72 which includes 24 standing passengers. We can say that the bus is low crowded if the occupancy in the bus is less than or equal to 48, the bus is medium crowded if the occupancy in the bus is greater than 48 and less than or equal to 60 and the bus is high crowded if the occupancy in the bus is greater than 60.

It is important to note that the thresholds discussed above must be corresponding to the total occupancy, which is the sum of occupancy because of ticket-holders and occupancy because of pass-holders.

In order to come up with crowding thresholds which reflects the ground truth, we first need to extract the boardings and alightings at each stage for multiple trips belonging to multiple routes and on multiple days. Based on the boardings and alightings at each stage, the occupancy in the bus from one stage to another stage is estimated (Deatiled methodology is mentioned in the next section).

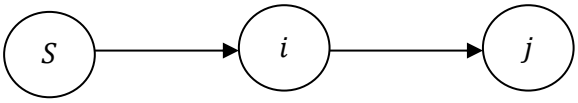
From the summary statistics of occupancy in buses for multiple trips belonging to multiple routes and on multiple days, crowding thresholds can be defined.

5.2.5 Methodology for Determining Crowding Level

From the trip level demand data estimated, we know the boardings at each stage, alightings at each stage, and the interchanges between stages for a selected route on a given day.

We can now determine the occupancy in the bus between two stages in the following way:

Let S represent the Source bus stage node, i, j represent the intermediate bus stage nodes.



If we know the number of passengers who got into the bus at the Source bus stage node, we will know the crowding estimate on the link $S - i$

Occupancy on the link $S - i$ = Crowd on the link $S - i$ = Number of boardings at the Source bus stage node = $B_s = O_{Si}$

Similarly, if we know the number of passengers getting down at the bus stage node i and the number of passengers getting inside at the bus stage node i, we will know the crowding estimate on the link $i - j$ from the following equation:

Occupancy on the link $i - j$ = Crowd on the link $i - j = B_S +$ Number of passengers getting inside the bus at the bus stage node $i -$ Number of passengers getting down at the bus stage node i

$$O_{ij} = O_{Si} + B_i - A_i \ (Eq. 1)$$

For a given trip, in the similar way, the occupancy on each link of the entire route can be estimated and can be classified into low, medium, or high crowding based on the crowding thresholds defined.

$(Eq. 1)$ is known as the state space equation. Using the state space equation, we have determined the crowding level for a historical '19B' trip.

Illustrative example of determining crowding level

Multiple trips of '19B' between 'KELAMBAKKAM' and 'T. NAGAR' on 3rd November, 2016 are used for demonstrating the methodology for determining the crowding level.

Based on the boardings and alightings at each stage and the state space equation, occupancy in each link from one stage to the next stage is estimated and then each link is classified into low, medium, or high level of crowding.

For this illustrative example, the number of pass-holders is not taken into account and the following assumption is made:

- If the crowd on link connecting two stages is less than or equal to 30, the crowding level on that link is defined as low
- If the crowd on link connecting two stages is greater than 30 and less than or equal to 40, the crowding level on that link is defined as medium
- If the crowd on link connecting two stages is greater than 40, the crowding level on that link is defined as high

The sample plots generated are shown in Figure 5.10, Figure 5.11, and Figure 5.12

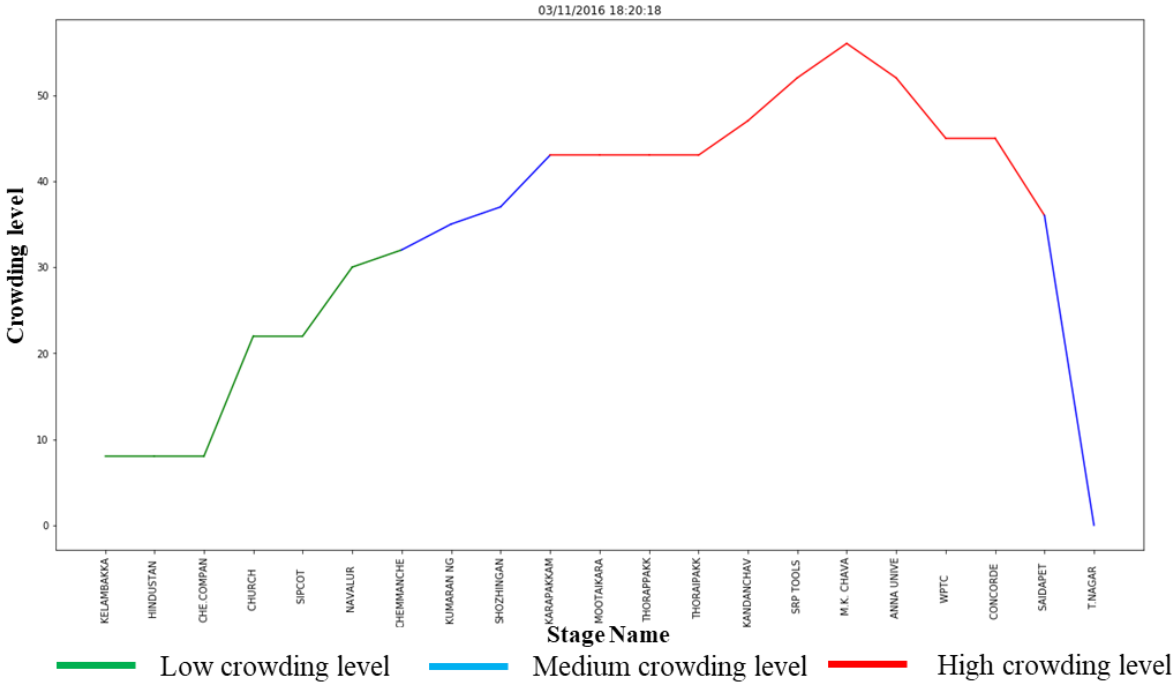


Figure 5.10 19B evening 6:20 PM trip crowding level on 3-11-2016

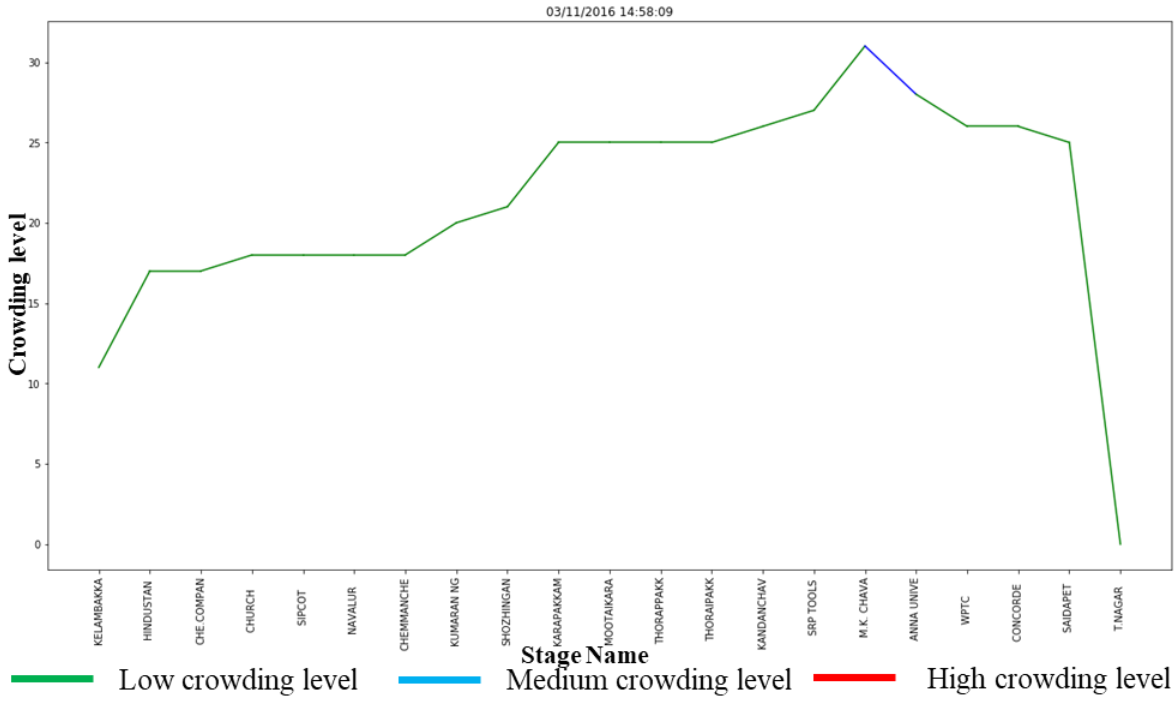


Figure 5.11 19B afternoon 3:00 PM trip crowding level on 3-11-2016

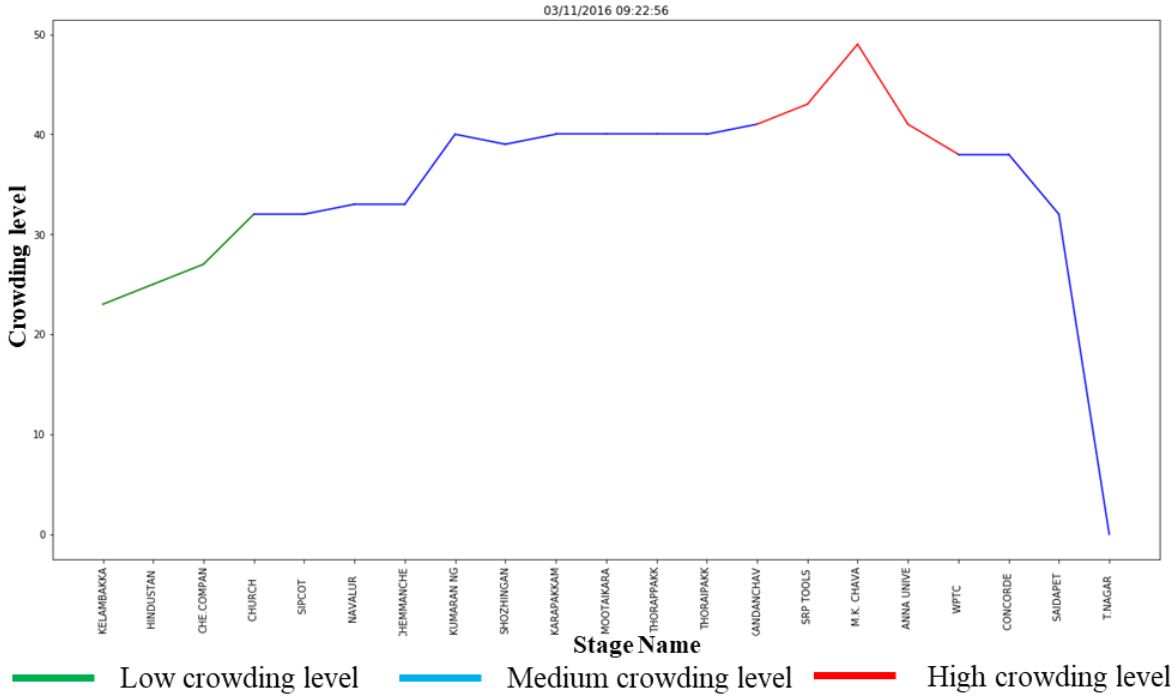


Figure 5.12 19B morning 9:20 AM trip crowding level on 3-11-2016

Suppose that we want to determine the crowding level for all the links of a given route for a future trip, we need to apply the state space equation to the predicted demand data and then determine the crowding level.

In this study, we will be predicting the number of interchanges between stages and then estimate the boardings and alightings at each stage which will then be input to the state space equation and the output is the predicted crowding level for each link of a given route. Detailed methodology for demand prediction is described in the next section.

In this step the input is historical trip level demand data or real time demand data, and ordered stages file. In this step, the output is the crowding level for all the links belonging to that route.

In the next section, we will discuss the methodology for Crowding Prediction.

5.3 Methodology flowchart for Crowding Prediction

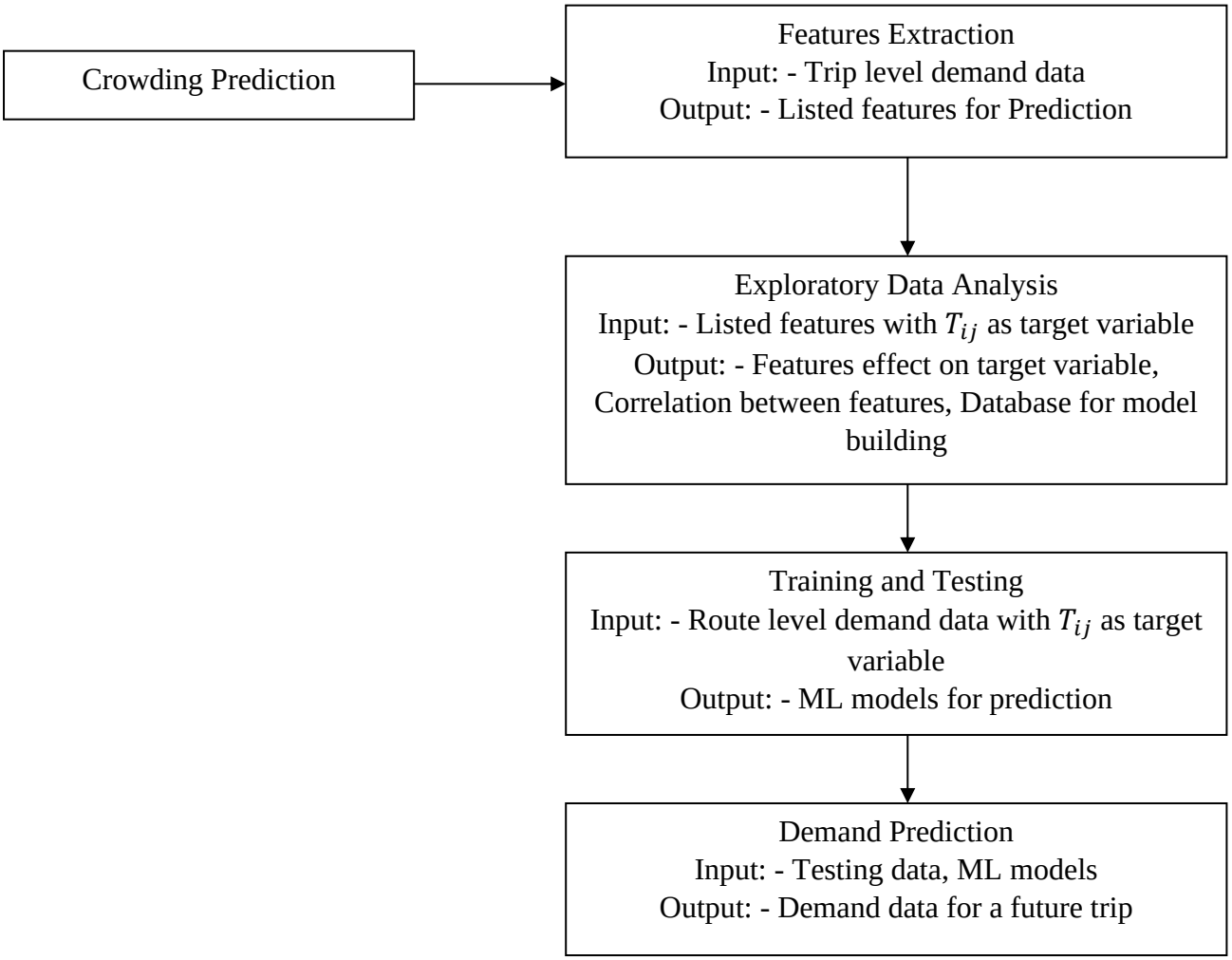


Figure 5.13 Flowchart for Crowding Prediction

5.3.1 Methodology overview for Crowding Prediction

For a future trip, if we predict the number of boardings and the number of alightings for each stage of the route, we can estimate the occupancy in each link of the route. However, the occupancy estimates are only for the ticket-holders. In order to estimate the number of pass-holders traveling from one stage to another stage, we have to first estimate the number of interchanges between stages from the ticket data.

Therefore, in this study, we predict the interchanges T_{ije} between two stages i and j for a given trip on a given day coming from the ticket data and then estimate the interchanges T_{ijp} for pass-holders as a percentage of the interchanges T_{ije}. We then calculate the total occupancy on each link and then predict the crowding level on each link using the state space equation described in the methodology for crowding estimation.

Factors affecting the number of boardings at a stop:

Based on the literature review, the following factors have been identified that affect the number of boardings at a stop.

- Time of Day
- Day of Week
- Month of Year
- Dwell time at the stop
- Travel time between previous bus stop and current bus stop
- Bus stop proximity to a prominent location
- Directional changes

Factors affecting the number of alightings at a stop:

Based on the literature review, the following factors have been identified that affect the number of alightings at a stop.

- Time of Day
- Day of Week
- Month of Year
- Dwell time at a stop
- Number of boardings at the source bus stop
- Travel time between previous bus stop and current bus stop

- Proximity to a prominent location
- Directional changes

In this study, we predict the number of interchanges for a trip belonging to one particular route based on the number of interchanges for the same trip on previous days.

Detailed methodology for interchanges prediction is described in the next sections.

5.3.2 Methodology for Interchanges prediction

Part A: Based on previous days interchanges:

For a selected route and a given schedule, there are multiple trips between an origin-destination pair on a given day. Also, the time at which the first trip of a particular schedule can alter across days. Therefore, the combination of schedule, run number, and time at which the trip started, we get a unique trip.

The number of interchanges between each stage pair of a unique trip obviously depends on the interchanges between the same stage pairs of the same trips on previous days. The response variable is the interchanges for a selected trip on current day while the explanatory variables are the interchanges for the same trip on previous days.

Before we describe the detailed methodology for interchanges prediction, let us look at the notations commonly used for prediction. The notations used are shown in Table 5.2

Table 5.2 Notations used in interchanges prediction

o: Origin	des: Destination	i: Boarding Stage
j: Alighting Stage	e: Ticket data	p: Pass data
B: Boardings	A: Alightings	T_{ij}: Interchanges from i to j
r: Route number	q: Run number	d: Day number
s: Schedule index (a, b, c, etc.)	t: Time within day	$q^1, d^1, s^1, etc.$ are indices for independent variables
$D: d_{max}$ (Number of previous days used for prediction)		
Q_r: Maximum number of runs on route r		

Suppose that we want to predict the number of interchanges of a given stage pair i, j for a selected route, run and schedule on a given day based on the interchanges of same run and schedule for the same route on first seven days as baseline explanatory variables, the response variable and explanatory variables are as shown in Table 5.3

Table 5.3 Response variable and Explanatory variables for interchanges prediction A

Y (Response Variable)	X (Explanatory Variables)
$Number\ of\ interchanges\ T_{ijsqtde}$	$Interchanges\ for\ same\ o-des\ for\ same\ schedule\ on\ first\ 7\ days$ $T_{ijs^1q^1t^11e}, T_{ijs^1q^1t^12e}, T_{ijs^1q^1t^13e}, \ldots . T_{ijs^1q^1t^17e}$
	$Day\ of\ Week_{ijsqtd}$ (Binary Variable)
	$Boarding\ Stage\ name_{ijsqtd}$ (Binary Variable)
	$Alighting\ Stage\ name_{ijsqtd}$ (Binary Variable)
	Other Explanatory Variables
	$Service\ type_{ijqstd}$ (Binary Variable) (if multiple schedules are used)
	$Time\ of\ Day_{ijsqtd}$ (Binary Variable) (1-hour intervals)
	$combination\ of\ s, q, and\ t\ represents\ unique\ run$
	$1 < q < Q_r$ ($\max runs\ on\ a\ day$)

All the explanatory variables which are not numerical are represented as binary variables by an array of 0's and 1's. For example, Monday is represented as [1,0,0,0,0,0,0] and Tuesday is represented as [0,1,0,0,0,0,0]. The final dataset used for prediction will have all these arrays as columns out of which one column will be kept as a baseline and will not be used for prediction.

Both linear (Linear Regression) and non-linear (Artificial Neural Networks) models can be developed that gives us the functional relationship between the explanatory variables and the response variable.

Part B: Based on previous runs on the same day:

The number of interchanges between each stage pair of a unique trip also depends on the interchanges between the same stage pairs of the previous trips on the same day.

Suppose that the for all the trips on a given day, the interchanges are less on that day, we can say that there is more chance that the interchanges will be less for the current trip. Similarly, if

the last two trips get cancelled for some reason, there is more chance that the interchanges for the current trip will be higher than usual. Therefore, it is important to know the functional relation between the interchanges of the previous trips and the interchanges of the current trip on the same day.

The response variable is the interchanges for a selected trip on a given day while the explanatory variables are the interchanges for the last few trips on the same day.

Suppose that we want to predict the number of interchanges of a given stage pair i, j for a selected route, run and schedule on a given day based on the interchanges of previous three runs for the same route on the same day as explanatory variables, the response variable and explanatory variables are as shown in Table 5.4

Table 5.4 Response variable and Explanatory variables for interchanges prediction B

Y (Response Variable)	X (Explanatory Variables)
$Number\ of\ interchanges\ T_{ijsqtde}$	$Interchanges\ for\ same\ o-des\ for\ previous\ 3\ trips\ on\ same\ day$ $T_{ijs-1q-1t-1e}, T_{ijs-2q-2t-2e}, T_{ijs-3q-3t-3e}$
	$Day\ of\ Week_{ijsqtd}$ (Binary Variable)
	$Boarding\ Stage\ name_{ijsqtd}$ (Binary Variable)
	$Alighting\ Stage\ name_{ijsqtd}$ (Binary Variable)
	Other Explanatory Variables
	$Service\ type_{ijqstd}$ (Binary Variable) (if multiple schedules are used)
	$Time\ of\ Day_{ijsqtd}$ (Binary Variable) (1-hour intervals)
	$combination\ of\ s, q, and\ t\ represents\ unique\ run$
	$1 < q < Q_r$ ($\max runs\ on\ a\ day$)

Both linear (Linear Regression) and non-linear (Artificial Neural Networks) models can be developed that gives us the functional relationship between the explanatory variables and the response variable.

Part C: Based on occupancy in the last few links of same run:

The number of interchanges between each stage pair of a unique run also depends on the occupancies on the previous links of the same run.

Suppose that the occupancies on all the previous links of the same run are low, then we can say that there is more chance that the interchanges for the current link will be low. Similarly, there is more chance that the interchanges for the links down the line will be low. Therefore, it is important to know the functional relation between the interchanges of the current link and the occupancies of the previous links for the same run.

Suppose that we want to predict the number of interchanges of a given stage pair $i, i+1$ for a selected route, run and schedule on a given day based on the occupancies of previous three links for the same run as explanatory variables, the response variable and explanatory variables are as shown in Table 5.5

Table 5.5 Response variable and Explanatory variables for interchanges prediction C

Y (Response Variable)	X (Explanatory Variables)
$Number\ of\ interchanges\ T_{ii+1sqtde}$	$Occupancies\ for\ same\ run\ on\ previous\ 3\ links$ $O_{i-1isqtd}, O_{i-2i-1sqtd}, O_{i-3i-2sqtd}$
	$Day\ of\ Week_{ii+1sqtd}$ (Binary Variable)
	$Boarding\ Stage\ name_{ii+1sqtd}$ (Binary Variable)
	$Alighting\ Stage\ name_{ii+1sqtd}$ (Binary Variable)
	Other Explanatory Variables
	$Service\ type_{ii+1qstd}$ (Binary Variable) (if multiple schedules are used)
	$Time\ of\ Day_{ii+1sqtd}$ (Binary Variable) (1-hour intervals)
	$combination\ of\ s, q, and\ t\ represents\ unique\ run$
	$1 < q < Q_r$ ($\max runs\ on\ a\ day$)

Both linear (Linear Regression) and non-linear (Artificial Neural Networks) models can be developed that gives us the functional relationship between the explanatory variables and the response variable.

Once we have the predicted interchanges for a future trip, we estimate the crowding level for each link of a selected route using the methodology described in Section 5.2

In the next section, we will see an illustrative application of Crowding Prediction.

5.4 Illustrative application of Crowding Prediction

We will illustrate the crowding prediction for '19B' route. The main explanatory variables used for prediction are the previous days interchanges.

We have initially extracted the interchanges for '19B' route between 'SAIDAPET' and 'KELAMBAKKAM' for all the trips from May 1st 2019 till May 31st 2019. We have fixed the time period as between 5:00 pm and 7:00 pm for all days. The first trip during this time period is assumed to be the same trip every day.

The interchanges between two stages of this trip for the first seven days are used as explanatory variables, the interchanges between two stages of the same trip for the next seven days are used as target variables for the training data and the interchanges between two stages of this trip for the last seven days are used as target variables for the testing data.

Additional explanatory variables included are Boarding Stage, Alighting Stage, and Day of Week. The target variable and response variables are shown in Table 5.6

Table 5.6 Response Variable and explanatory variables for sample prediction

Y (Response Variable)	X (Explanatory Variables)
$Number\ of\ interchanges\ T_{ijsqtde}$	$Interchanges\ for\ same\ o-des\ for\ same\ schedule\ on\ first\ 7\ days$ $T_{ijs^1q^1t^11e}, T_{ijs^1q^1t^12e}, T_{ijs^1q^1t^13e}, \ldots . T_{ijs^1q^1t^17e}$
	$Day\ of\ Week_{ijsqtd}$ (Binary Variable)
	$Boarding\ Stage\ name_{ijsqtd}$ (Binary Variable)
	$Alighting\ Stage\ name_{ijsqtd}$ (Binary Variable)

The sample dataset created for training and testing is shown in Figure 5.14 and Figure 5.15

Results Obtained:

Linear Regression model gave a testing score of 0.32 and a training score of 0.18. Multi-layer Perceptron Artificial Neural Network is overfitting on the training data. Random Forest Regressor is not giving good training score or testing score.

Mean Square Error obtained by the best Linear Regression model is 0.554.

Td	T1	T2	T3	T4	T5	T6	T7	B1	B2	B3	B4	B5
0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	2	0	0	0	0	0	1	0	1	0	0	
0	0	1	2	2	3	1	4	0	1	0	0	
1	2	0	0	0	2	5	1	0	1	0	0	
1	0	2	0	0	1	1	2	0	1	0	0	
1	1	0	0	0	1	2	2	0	1	0	0	
5	6	5	1	1	4	10	5	0	1	0	0	
1	2	1	0	0	2	0	0	0	1	0	0	
0	0	2	0	0	1	1	1	0	1	0	0	
1	3	3	2	2	0	0	1	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	2	0	1	0	0	
1	2	3	0	0	2	0	0	0	1	0	0	

used training data

Figure 5.14 Sample training data part 1

A19	A20	A21	A22	A23	A24	A25	A26	A27	A28	A29	Monday	Tuesday	W
0	0	0	0	0	0	0	1	0	0	0	0	0	
0	0	0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	0	0	0	0	

Figure 5.15 Sample training data part 2

Conclusions:

As we extracted the data for only one particular trip, we have very limited data compared to the number of features we have considered. We need to extract the interchanges for more trips and repeat this exercise to get ample amount of training data. Also, it is suggestible to use time-series data analysis in order to better predict the number of interchanges for a future trip.

As we have noted that Machine Learning models failed to capture the patterns in the time-series data, LSTM neural networks have to be tried for prediction.

5.5 Summary

This chapter provides a detailed methodology for crowding estimation and prediction. We have seen the methodology and extraction process for estimating boardings, alightings, and interchanges for a selected route. We have then discussed a hierarchical algorithm and machine learning application to map the stops to stages and estimate the demand at stop level. Estimating the crowding level on each link of a given route using the state space equation has also been discussed. We have also seen detailed methodology for building a historical model to predict the number of interchanges between stages and an illustrative application of crowding prediction. In the next chapter, we will discuss the detailed methodology for estimating and predicting travel time using ETM and GPS data.

CHAPTER 6

IMPLEMENTATION OF TRAVEL TIME ESTIMATION AND PREDICTION

6.1 Overview

In this chapter, we provide the detailed methodology for estimating travel time for MTC buses using the extracted ETM data. We will then provide a methodology for finding the functional relationship between the approximate travel time estimates from the ETM data to that of actual travel time estimates obtained from GPS data. Lastly, we will provide a detailed methodology for predicting stage to stage travel time on selected routes in real-time. The flow charts including the input, output, and process of all the steps involved in travel time estimation and prediction are depicted in this chapter.

The flowchart for the steps involved in this chapter are shown in Figure 6.1

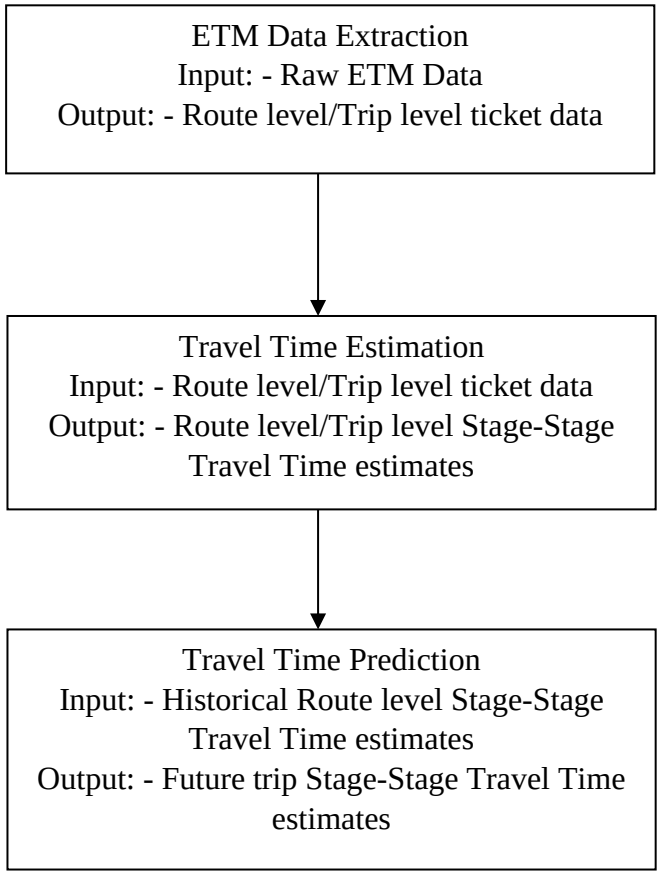


Figure 6.1 Flowchart for travel time estimation and prediction

6.2 Methodology flowchart for Travel Time Estimation

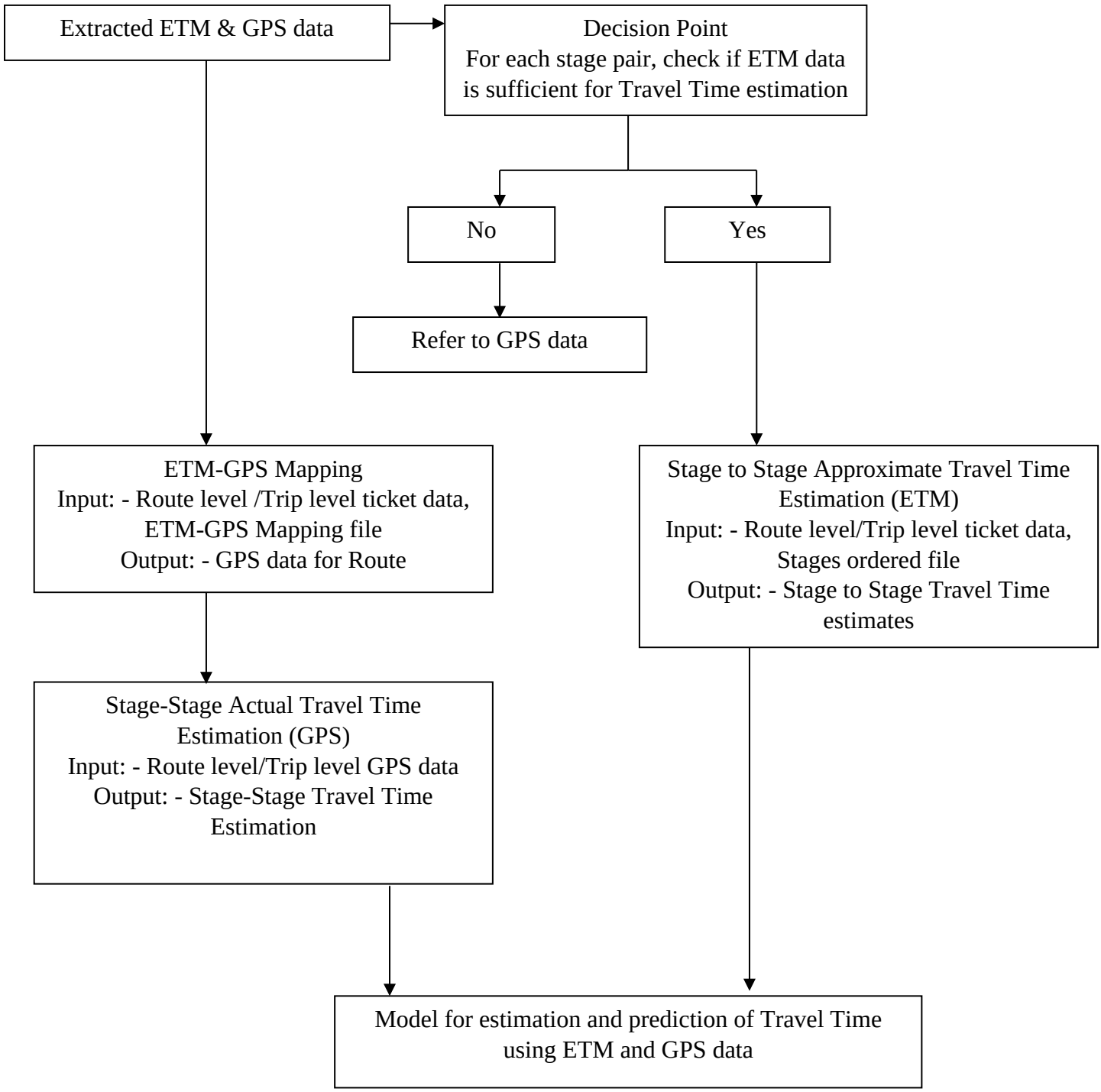


Figure 6.2 Methodology flowchart for Travel Time Estimation

6.2.1 Methodology for Stage to Stage Travel Time Estimation using ETM data

The first step in travel time estimation is generating stage to stage approximate travel time estimates using the extracted route level ticket data. It is assumed that the timestamp of the first ticket issued at a stage is equal to the arrival time of the bus at that stage. The approximate travel time between two stages is the difference between the arrival time of the bus at those two stages.

On a given day, for a selected route, there may be multiple origins and destinations. If we select a particular route, and then fix the origin and destination, we get trip level data. On a given day, there are multiple bus trips between origin and destination of a selected route.

The trip level ticket data contains the headers listed in Table 6.1

Table 6.1 Headers of trip level ticket data

Date	ETM Device ID	From Stage	Trip No.	Source
Depot	Adult	To Stage	Trip Start Time	Destination
Schedule Name	Child	Fleet No.	Trip End Time	Ticket Issued Time

In this study, we estimate the approximate travel time between stages only if there are tickets issued at both 'FromStage' and 'ToStage'. The approximate travel time between a stage pair i, j is defined only if $j > i$. The approximate travel time between a stage pair i, j is defined as -1 if $j < i$. The approximate travel time between a stage pair i, j is defined as -2 if there are no tickets issued at either stage i or stage j.

Initially for each trip of a selected route, the approximate travel time between stages is estimated. The stages for the route are listed manually because for all the trips, tickets might not be issued at every stage. Therefore, for estimating the approximate travel time between stages for a given route, we need an ordered stages file of that route.

Sample stage to stage approximate travel time estimates for '19B' route between 'KELAMBAKKAM' and 'T. NAGAR' for one 'UP' trip on 3rd November, 2016 is depicted in Figure 6.3

	KELAMBAI	HINDUSTA	CHE.COMI	CHURCH	SIPCOT	NAVALUR	CHEMMAI	KUMARAN	SHOZHINC	KARAPAKK	MOOTAIK	THORAPP/	THORAIPA	KANDANC	SRP TOOL	M.K. CHA	ANNA UNI	WPTC	CONCORD	SAIDAPET	T.NAGAR
KELAMBAI	0	4.45	-2	13.0167	-2	13.95	18.1833	-2	-2	-2	-2	-2	-2	-2	-2	50.75	-2	56.8	-2	-2	-2
HINDUSTA	-1	0	-2	8.56667	-2	9.5	13.7333	-2	-2	-2	-2	-2	-2	-2	-2	46.3	-2	52.35	-2	-2	-2
CHE.COMI	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
CHURCH	-1	-1	-1	0	-2	0.93333	5.16667	-2	-2	-2	-2	-2	-2	-2	-2	37.7333	-2	43.7833	-2	-2	-2
SIPCOT	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
NAVALUR	-1	-1	-1	-1	-1	0	4.23333	-2	-2	-2	-2	-2	-2	-2	-2	36.8	-2	42.85	-2	-2	-2
CHEMMAI	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	32.5667	-2	38.6167	-2	-2	-2
KUMARAN	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
SHOZHINC	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
KARAPAKK	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
MOOTAIK/	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
THORAPP/	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2
THORAIPA	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2
KANDANC	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2
SRP TOOL	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2
M.K. CHA	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	6.05	-2	-2	-2
ANNA UNI	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2
WPTC	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2
CONCORD	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2
SAIDAPET	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2
T.NAGAR	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Figure 6.3 Sample stage to stage travel time estimates for a trip between 'KELAMBAKKAM' and 'T. NAGAR'

Sample stage to stage approximate travel time estimates for '19B' route between 'T. NAGAR' and 'KELAMBAKKAM' for one 'DOWN' trip on 3rd November, 2016 is depicted in Figure 6.4

	KELAMBA	HINDUST/	CHE.COM	SIRUCHER	CHURCH	SIPCOT	NAVALUR	CHEMMAI	PERUMBA	NUCLEUS	KUMARAN	SHOZHINC	KARAPAKK	MOOTAIK	THORAIPA	KANDANC	SRP TOOL	WPTC	ANNA UN	CONCORE	SAIDAPET	T.NAGAR
KELAMBA	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
HINDUST/	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHE.COM	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SIRUCHER	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHURCH	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SIPCOT	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
NAVALUR	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHEMMAI	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
PERUMBA	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
NUCLEUS	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
KUMARAN	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SHOZHINC	-2	-2	-2	-2	-2	12.95	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
KARAPAKK	-2	-2	-2	-2	-2	20.6167	-2	-2	-2	-2	-2	7.66667	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
MOOTAIK	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1
THORAIPA	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1
KANDANC	-2	-2	-2	-2	-2	37.95	-2	-2	-2	-2	-2	25	17.3333	-2	-2	0	-1	-1	-1	-1	-1	-1
SRP TOOL	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1
WPTC	-2	-2	-2	-2	-2	45.4167	-2	-2	-2	-2	-2	32.4667	24.8	-2	-2	7.46667	-2	0	-1	-1	-1	-1
ANNA UN	-2	-2	-2	-2	-2	57.7	-2	-2	-2	-2	-2	44.75	37.0833	-2	-2	19.75	-2	12.2833	0	-1	-1	-1
CONCORE	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1
SAIDAPET	-2	-2	-2	-2	-2	62.1833	-2	-2	-2	-2	-2	49.2333	41.5667	-2	-2	24.2333	-2	16.7667	4.48333	-2	0	-1
T.NAGAR	-2	-2	-2	-2	-2	70.3833	-2	-2	-2	-2	-2	57.4333	49.7667	-2	-2	32.4333	-2	24.9667	12.6833	-2	8.2	0

Figure 6.4 Sample stage to stage travel time estimates for a trip between 'T. NAGAR' and 'KELAMBAKKAM'

In this study, for a given route we fix the ordering of the stages belonging to that route. An 'UP' trip on this route is that trip where the origin is same as the first stage or close to the first stage and destination is same as the last stage or close to the last stage. A 'DOWN' trip on this route is that trip where the origin is same as the last stage or close to the last stage and destination is same as the first stage or close to the first stage.

It can be noted from Figure 6.3 and Figure 6.4 that the 'UP' trip have an upper triangular matrix and the 'DOWN' trip have a lower triangular matrix. Since both the 'UP' trip and 'DOWN' trip doesn't have tickets issued at all the stages, we can see that some cells in the Figure 6.3 and Figure 6.4 have -2.

Although, for one particular trip, we might not be getting approximate travel time between stages for all possible stage pairs but if we consider more trips between the same origin and destination pair, we can get the approximate travel time between all possible stage pairs by averaging the estimates over multiple trips. Essentially, for a given stage pair, we are filling the missing estimates of one particular trip with the average of approximate travel time estimates between the same stage pair of all other trips.

Sample stage to stage approximate average travel time estimates for '19B' route between 'KELAMBAKKAM' and 'T. NAGAR' for all 'UP' trips on 3rd November, 2016 is depicted in Figure 6.5

	KELAMBA	HINDUST/	CHE.COM	SIRUCHER	CHURCH	SIPCOT	NAVALUR	CHEMMAI	PERUMBA	NUCLEUS	KUMARAN	SHOZHIN	KARAPAK	MOOTAIK	THORAIP/	KANDANC	SRP TOOL	WPTC	ANNA UN	CONCORI	SAIDAPET	T.NAGAR
KELAMBA	0	5.05079	12.9611	-2	13.0321	11.1767	16.1902	19.6383	-2	-2	21.3386	30.3009	35.1676	40.3024	45.8417	52.4571	61.9788	61.8533	67.8583	80.6278	70.6083	-2
HINDUST/	-1	0	3.16667	-2	6.91061	7.01	11.1167	14.4476	-2	-2	17.4433	24.1278	30.0196	36.3357	42.4222	47.4231	56.0426	58.4433	64.6583	77.725	66.8583	-2
CHE.COM	-1	-1	0	-2	6.95	-2	9.3	12.35	-2	-2	12.7	22.7583	27.45	-2	-2	43.6333	46.6333	-2	-2	61.9833	-2	-2
SIRUCHER	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
CHURCH	-1	-1	-1	-1	0	-2	3.82692	6.82222	-2	-2	11.1515	17.7256	22.7967	-2	-2	40.2119	49.7813	59.2417	-2	69.0333	45.9333	-2
SIPCOT	-1	-1	-1	-1	-1	0	3.90417	-2	-2	-2	9.7	16.3233	24.5708	28.8167	35.2722	42.5458	56.75	59.4167	64.0167	71.1833	72.2167	-2
NAVALUR	-1	-1	-1	-1	-1	-1	0	3.57778	-2	-2	6.27037	13.3583	19.5333	24.9071	31.1083	36.9897	46.5646	47.63	52.975	61.225	55.3417	-2
CHEMMAI	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	2.35	11.1292	16.2944	21.65	28.5917	34.2958	43.875	39.775	-2	-2	-2	-2
PERUMBA	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
NUCLEUS	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2
KUMARAN	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	6.76548	12.9436	18.8733	25.7458	30.3958	39.0926	45.0611	52.2167	56.0722	49.275	-2
SHOZHIN	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5.81071	11.3375	17.7958	23.2917	31.5683	47.125	39.4583	48.7278	53.0667	-2
KARAPAK	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	4.93056	11.1167	17.1808	26.2233	35.95	34.225	46.4917	37.8083	-2
MOOTAIK	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	6.83667	12.8667	25.5833	24.475	32.85	40.0167	41.05	-2
THORAIP/	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	5.22333	14.5667	-2	21.675	33.0333	34.0667	-2
KANDANC	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	7.87	16.3611	17.375	27.1167	21.8417	-2
SRP TOOL	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	7.55	7.10833	16.5944	15.4667	-2
WPTC	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2	-2	-2	-2
ANNA UN	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	7.16667	8.2	-2
CONCORI	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	1.03333	-2
SAIDAPET	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-2
T.NAGAR	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Figure 6.5 Average stage to stage travel time estimates for al trips between 'KELAMBAKKAM' and 'T. NAGAR' on 3-11-2016

Sample stage to stage approximate average travel time estimates for '19B' route between 'T. NAGAR' and 'KELAMBAKKAM' for all 'DOWN' trips on 3rd November, 2016 is depicted in Figure 6.6

	KELAMBA	HINDUST/	CHE.COM	SIRUCHER	CHURCH	SIPCOT	NAVALUR	CHEMMAI	PERUMBA	NUCLEUS	KUMARAN	SHOZHIN	KARAPAK	MOOTAIK	THORAIP/	KANDANC	SRP TOOL	WPTC	ANNA UN	CONCORI	SAIDAPET	T.NAGAR
KELAMBA	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
HINDUST/	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHE.COM	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SIRUCHER	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHURCH	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SIPCOT	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
NAVALUR	-2	-2	-2	-2	-2	2.73333	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
CHEMMAI	-2	-2	-2	-2	-2	5.11667	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
PERUMBA	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
NUCLEUS	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
KUMARAN	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
SHOZHIN	-2	-2	-2	-2	-2	13.2417	14.675	6.79167	-2	-2	6.1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
KARAPAK	-2	-2	-2	-2	-2	20.8208	24.7083	23.2667	-2	-2	16.35	8.3375	0	-1	-1	-1	-1	-1	-1	-1	-1	-1
MOOTAIK	-2	-2	-2	-2	-2	28.3667	-2	-2	-2	-2	-2	16.8333	4.01667	0	-1	-1	-1	-1	-1	-1	-1	-1
THORAIP/	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1
KANDANC	-2	-2	-2	-2	-2	40.6139	37.7958	34.7389	-2	-2	32.0167	25.8583	17.6431	14.3917	-2	0	-1	-1	-1	-1	-1	-1
SRP TOOL	-2	-2	-2	-2	-2	44.8667	41.0833	42.8083	-2	-2	35.0333	29.0278	21.7537	18.8167	-2	3.60513	0	-1	-1	-1	-1	-1
WPTC	-2	49.35	-2	-2	-2	50.1917	58.6444	44.9208	-2	-2	42.0333	39.5486	31.7056	37.1917	-2	13.1917	9.41778	0	-1	-1	-1	-1
ANNA UN	-2	54.45	-2	-2	-2	63.8619	60.4958	59.4167	-2	-2	56.05	47.7929	46.0773	49.4417	-2	25.5396	22.5488	11.6433	0	-1	-1	-1
CONCORI	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1
SAIDAPET	-2	64.2667	-2	-2	-2	70.35	80.9417	67.8944	-2	-2	60.1167	55.0385	51.0083	54.0917	-2	31.4897	27.0064	17.4108	6.74118	-2	0	-1
T.NAGAR	-2	70.3	-2	-2	-2	80.681	77.9042	75.375	-2	-2	69.0667	65.45	62.6833	63.975	-2	41.7222	38.8531	28.1283	16.8545	-2	9.80952	0

Figure 6.6 Average stage to stage travel time estimates for al trips between 'T. NAGAR' and 'KELAMBAKKAM' on 3-11-2016

Similar exercise has also been carried out with 2019 ETM data for '19B' route. Travel time estimates of both 'UP' trips and 'DOWN' trips are recorded in the same file and the trips are distinguished by the column 'Direction'.

In order to estimate the travel time between stages from 2019 ETM data, we need the following files:

1. Route wise cleaned ETM data on a given day.
2. The headers of the cleaned ETM data. The headers are listed in Table 6.2

Table 6.2 Headers of the cleaned ETM data

WayBillNo	Depot	Schedule_name	ETMNO	OutDate
OutTime	InDate	InTime	TotalAmount	Adult
Child	TicketIssuedDate	TicketIssuedTime	FromStage	ToStage
FLEETNO	Denomination	TicketType	Luggage	TicketNo
TripNo	Concession	Cond	Driver	TripStartDate
TripStartTime	TripEndDate	TripEndTime	Source	Destination

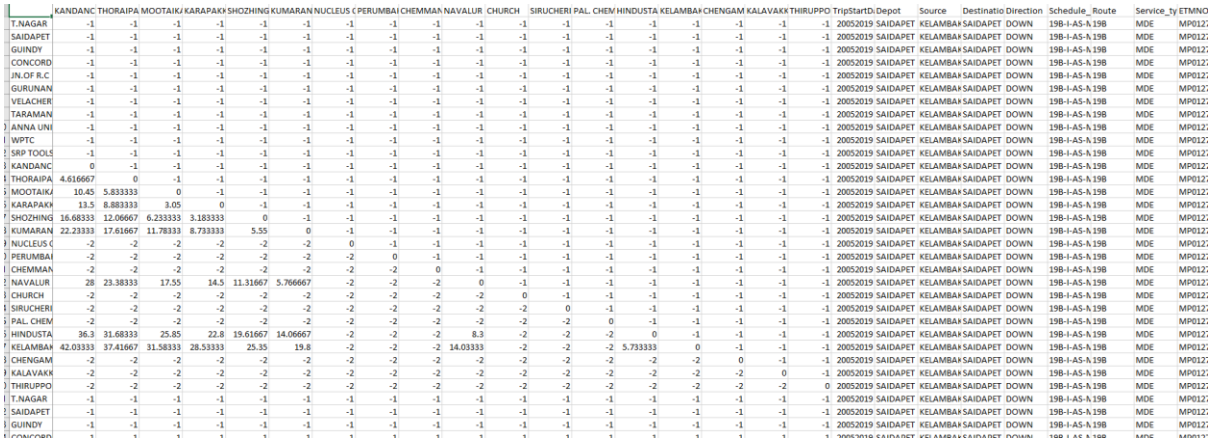
3. Ordered Stage List. The headers are listed in Table 6.3

Table 6.3 Headers of the Stage List

519	570	19B	M70

We input the ordered stage list for the route of our interest.

Sample stage to stage approximate average travel time estimates for '19B' route between 'T. NAGAR' and 'THIRUPPORU' for all trips on 20th May 2019 is depicted in Figure 6.7



	KANDANC	THORAIPA	MOOTAIK/	KARAPAK	SHOZHING	KUMARAN	NUCLEUS (	PERUMBAI	CHEMMAN	NAVALUR	CHURCH	SIRUCHERI	PAL. CHEM	HINDUSTA	KELAMBAI	CHENGAM	KALAVAKK	THIRUPPO	TripStartDa	Depot	Source	Destinatio	Direction	Schedule_Route	Service_ty	ETMNO
T.NAGAR	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
SAIDAPET	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
GUINDY	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
CONCORD	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
JN.OF R.C	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
GURUNAN	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
VELACHER	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
TARAMAN	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
ANNA UNI	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
WPTC	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
SRP TOOLS	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
KANDANC	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
THORAIPA	4.616667	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
MOOTAIK/	10.45	5.833333	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
KARAPAK	13.5	8.883333	3.05	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
SHOZHING	16.68333	12.06667	6.233333	3.183333	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
KUMARAN	22.23333	17.61667	11.78333	8.733333	5.55	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
NUCLEUS (	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
PERUMBAI	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
CHEMMAN	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
NAVALUR	28	23.38333	17.55	14.5	11.31667	5.766667	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
CHURCH	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
SIRUCHERI	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
PAL. CHEM	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
HINDUSTA	36.3	31.68333	25.85	22.8	19.61667	14.06667	-2	-2	-2	8.3	-2	-2	-2	0	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
KELAMBAI	42.03333	37.41667	31.58333	28.53333	25.35	19.8	-2	-2	-2	14.03333	-2	-2	-2	5.733333	0	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
CHENGAM	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
KALAVAKK	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
THIRUPPO	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	-2	0	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
T.NAGAR	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
SAIDAPET	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012
GUINDY	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	20052019	SAIDAPET	KELAMBAI	SAIDAPET	DOWN	19B-I-AS-N	19B	MDE	MP012

Figure 6.7 Sample stage to stage average travel time estimates for '19B' route between 'T. NAGAR' and 'THIRUPPORU' for all trips on 20th May 2019

In this study, we have estimated the approximate travel time between stages if at least one ticket is issued at both 'FromStage' and 'ToStage'. However, the travel time estimates between two stages where the number of tickets issued at both the two stages is very less might not be good estimates at all because in reality, there are only stops and not stages.

The approximate travel time between two stages ideally should be difference between the arrival time of the bus at the first stop corresponding to the first stage and the arrival time of the bus at the first stop corresponding to the second stage. When there are very few tickets, issued at two stages, the ticket issued at one stage might be the one corresponding to the last stop of that stage and the ticket issued at the other stage might be the one corresponding to the first stop of that stage.

Therefore, we will get meaningful travel time estimates between two stages only if we have two distinguishable clusters of tickets close to 'FromStage' and close to 'ToStage'. If we plot the time stamps of all the tickets issued at two consecutive stages and then if we see two distinguishable clusters, we can include both the stages in our analysis. If we do not see two distinguishable clusters, then we use travel time estimates from the GPS data for such stages. The methodology for estimating travel time between two stops using the GPS data is discussed in section 6.2.3.

The decision point shown in the Figure 2 is about the preliminary clustering of timestamps of tickets issued to decide if a stage is to be included in approximate travel time estimation between stages using ETM data.

In this step, the input is Route level/Trip level ticket data and the ordered stages file and the output is stage to stage approximate travel time estimates.

6.2.2 Methodology for ETM-GPS Mapping

The stage to stage travel time estimates obtained from the ETM data are approximate. We therefore need to estimate the travel time between stops/stages using GPS data.

As discussed in Chapter 4, one MTC bus have one GPS device which is identified by the Device ID and the GPS data for all the trips made by the bus on a given day are saved in a CSV file.

On a given day, for a selected route, we have the processed ticket data. Now we need to find which CSV files have the GPS data of all the trips made by buses on the same route on the same day.

A vehicle is identified by its Fleet ID. The data about Kanpur or GLOBEES GPS Device ID installed on a bus with a known Fleet ID is obtained from MTC in a CSV file named 'KNP GLB ETM Mapping'. The headers of the file are listed in Table 6.4

Table 6.4 Headers of 'KNP GLB ETM Mapping file'

ROUTE	DEVICE ID	SERVICE	FLEET ID
DEPOT	COMPANY	Device activeness on a given day	

Among the common features between the processed ticket data and the mapping file, Fleet ID and Depot are used to merge the processed ticket data of a given day and the mapping file.

Sample data of the 2016 mapping file after processing for common features is depicted in Figure 6.8. Sample processed ticket data for 9th Nov 2016 used for mapping ETM and GPS data is depicted in Figure 6.9

For verifying the correctness of the merging, route number is extracted from the 'Schedule_name' column in the ticket data and compared it with the route number obtained from the mapping file.

Because of some discrepancies in the mapping file obtained from the MTC and in the Fleet ID notations in ETM data and the mapping file, very few matches were obtained for November 2016 data.

Out[8]:

	Fleet_ID	Depo_name	Route_number	Device_number
0	I1880	TAMBARAM	PP66	868324020318031
1	I2417	SAIDAPET	5A	865733021573027
2	I0028	AYANAVARAM	48C	865733021572730
3	I2164	AMBATTUR	70A	865733021572409
4	I2824	T NAGAR	5A	865733021572391
5	I2521	MADHAVARAM	62	865733021572383
6	I0742	ENNORE	56N	865733021572367
7	I3004	ADYAR	29C	865733021572367
8	I2189	VADAPALANI	25G	865733021572359
9	I2432	VADAPALANI	25G	865733021572342
10	I0573	ADYAR	91	865733021572334
11	I3008	THIRUVANMIYUR	102	865733021572326
12	I1350	PERAMBUR	29C	865733021572318
13	I1251	ADYAR	91	865733021572300
14	I2248	THIRUVANMIYUR	19B	865733021572292
15	I2986	ENNORE	56N	865733021572268
16	I2305	ADYAR	29C	865733021572268
17	J0772	ADYAR	91	865733021572250
18	I0819	KUNDRATHUR	88K	865733021572243
19	I2393	BASIN BRIDGE	109	865733021572227
20	I0580	KK NAGAR	M70	865733021572219
21	I2228	CENTRAL	21G	865733021572201
22	I3006	ADYAR	47A	865733021572193
23	I1131	KUNDRATHUR	88K	865733021572169
24	I2176	TONDIARPET	56P	865733021572169
25	I0575	ADYAR	95	865733021572151
26	I2725	ADYAR	102	865733021572144
27	I1317	ADYAR	91	865733021572128
28	I3259	KK NAGAR	M70	865733021572110
29	I0886	CENTRAL	54	865733021572094

Figure 6.8 Sample data of the 2016 mapping file

	Date	Depo_name	Schedule_name	Fleet_ID	Source	Destination	Trip_Start_Time	Trip_End_Time	First_Ticket_Time	Last_Ticket_Time	ETM_NO
0	09/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	T.NAGAR	TAMBARAM W	22:40:07	23:26:07	22:51:23	23:02:36	MP02595A
1	09/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	TAMBARAM W	T.NAGAR	21:28:00	22:37:00	21:43:06	22:20:17	MP02595A
2	09/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	T.NAGAR	TAMBARAM W	20:14:22	21:24:22	20:21:21	21:17:37	MP02595A
3	09/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	N.G.O COLO	T.NAGAR	19:47:00	20:11:00	20:00:04	20:03:52	MP02595A
4	10/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	TAMBARAM W	T.NAGAR	01:18:00	02:58:00	01:45:48	02:43:19	MP02595A
5	10/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	T.NAGAR	TAMBARAM W	00:41:54	01:16:54	00:42:29	01:01:51	MP02595A
6	09/11/2016	ADAMBAKKAM	5ANS-C-PN-MNS	I2384	TAMBARAM W	T.NAGAR	23:28:00	00:39:00	00:02:37	00:23:27	MP02595A

Figure 6.9 Sample processed ticket data for 9th Nov

The sample output of the process described above for 9th November, 2016 is depicted in Figure 6.10

In this step, the input is Route/trip level ticket data and ETM-GPS mapping file, and the output is the corresponding GPS file of a selected Fleet ID on a selected route and on a given day. Now that we know which GPS file to refer, we will discuss the methodology to estimate the travel time between stops/stages using GPS data.

Schedule	Fleet_ID	Route	Device_number	match
5E-A-PN-MNS	I2725	102	8.65733E+14	FALSE
102-F1-AS-MAC	I2692	91	8.65733E+14	FALSE
M05-B-AM-MDE	J0394	21L	8.65733E+14	FALSE
102-E-AS-MAC	I2725	102	8.65733E+14	TRUE
505-A-AS-MEX	I0704	120	8.6217E+14	FALSE
120-I-AM-MEX	I2981	120	8.6217E+14	TRUE
23C-U-PM-MDE	I3136	23C	8.65733E+14	TRUE
48C-B-PM-MOR	I1362	48C	8.65733E+14	TRUE
120-C-AM-MDE	I3158	120	8.65733E+14	TRUE
120-A-AM-MDE	I3154	120	8.65733E+14	TRUE
109ET-H-AS-MDE	I2330	109	8.65733E+14	TRUE
159A-L1-AS-MDE	I0291	159A	8.6217E+14	TRUE
109-C1-AS-MDE	I3023	109	8.65733E+14	TRUE
56NEX-M-AS-MDE	I2040	56N	8.65733E+14	TRUE
56N-B-PM-MOR	I0160	56N	8.65733E+14	TRUE
56N-K-AS-MOR	I1306	56N	8.65733E+14	TRUE
56N-I-PM-MOR	I2986	56N	8.65733E+14	TRUE
26-C-AS-MDE	I2102	26	8.65733E+14	TRUE
P66-Q-PM-MDE	I1883	PP66	8.6217E+14	FALSE
M70-Y-AS-MDE	I2377	M70	8.65733E+14	TRUE
6D-S-PM-MEX	I0733	6D	8.65733E+14	TRUE
109-M-AS-MDE	I2247	109	8.65733E+14	TRUE
109-L-AS-MDE	I2225	109	8.65733E+14	TRUE
M1GS-A-AP-MOR	I0894	6D	8.65733E+14	FALSE
109-J-AS-MDE	I2363	109	8.65733E+14	TRUE

Figure 6.10 Sample output of the ETM-GPS mapping

6.2.3 Methodology for Stage to Stage Travel Time Estimation using GPS data

The actual stage to stage travel time estimation is done for a particular trip of a selected route on a given day. However, the GPS data file of a given Device ID used on a given Route ID (Device ID and Route ID data is obtained from MTC) on a given day have the GPS data of all the trips made by the bus with that Device on that route and on the selected day. Therefore, our first step is to break down the full-day data into trips data.

A trip is identified based on whether the destination stop has been reached or not. The arrival time of the bus at each stop is recorded when the bus enters a 100 m or 50 m buffer zone around the corresponding latitude and longitude coordinates of the stop.

The stop to stop actual travel time of a given stop pair i, j is calculated as the difference between the arrival time of the bus at stop i and the arrival time of the bus at stop j. The stage to stage actual travel time is then calculated as the difference between the arrival time of the bus at the first stop corresponding to first stage and the arrival time of the bus at the first stop corresponding to second stage.

In this step, the input is GPS data file corresponding to a selected route on a given day and the latitude, longitude coordinates of the bus stop locations of that route and the output is stop to stop travel time estimates or stage to stage travel time estimates.

The Stage to Stage travel time estimates obtained from GPS data are much more accurate than the estimates obtained from ETM data. Therefore, we hypothesize that we can build some Machine Learning models to capture the functional relationship between the travel time estimates obtained from the ETM data and the travel time estimates obtained from the GPS data. We discuss more on this hypothesis in the next section.

6.2.4 Methodology for estimating stage to stage travel time using both ETM and GPS data

For a given trip, now we can get the stage to stage travel time estimates obtained from both ETM and GPS data.

If we can capture the functional relationship between the travel time estimates obtained from the ETM data and the travel time estimates obtained from the GPS data, we can estimate the actual travel time between two stages just from the ETM data. In this way, we can completely calculate more accurate travel time between stages without using GPS data. We have GPS devices installed only on selected routes of the Chennai MTC transit network. However, we have ETM data on almost all of the routes of the Chennai MTC transit network. Installing GPS devices on all the buses is very expensive and if we are able to obtain travel time estimates just from ETM data, it would be more beneficial to all the stakeholders involved.

If we take multiple trips of a selected route and on multiple days and estimate the approximate travel time between all stage pairs from ETM data and actual travel time between all stage pairs from GPS data, we can create a dataset where the dependent variable is Stage to Stage GPS data and independent variable is Stage to Stage ETM data.

We build machine learning models to obtain $tt_{ii+1_{gps}} = f(tt_{ii+1_{etm}})$ where $tt_{ii+1_{gps}}$ is the actual travel time estimate between two consecutive stages and $tt_{ii+1_{etm}}$ is the approximate travel time estimate between the same stage pair.

Suppose that there are 10 stages on a given route. Assume that we know the approximate and actual travel time between all stages i and $i+1$ where $1 \leq i \leq 8$. Now my training data will contain the approximate travel time as dependent variable and actual travel time as independent variable for the first 9 links of that route. Assuming that we know the approximate travel time

for the 10^{th} link, we will predict the actual travel time for the 10^{th} link of that route using the machine learning model trained on the first 9 links.

Using this idea, we can estimate the actual travel time between stages of those routes where GPS data is completely missing from the models developed on the routes where we have both ETM and GPS data.

In this step, the input is travel time estimates from ETM and GPS data and the output is the functional relationship between travel time estimates obtained from ETM and GPS data.

In the next section, we will discuss the methodology for Travel time prediction of a future trip and the methodology for Travel time prediction in real-time.

6.3 Methodology flowchart for Travel Time Prediction

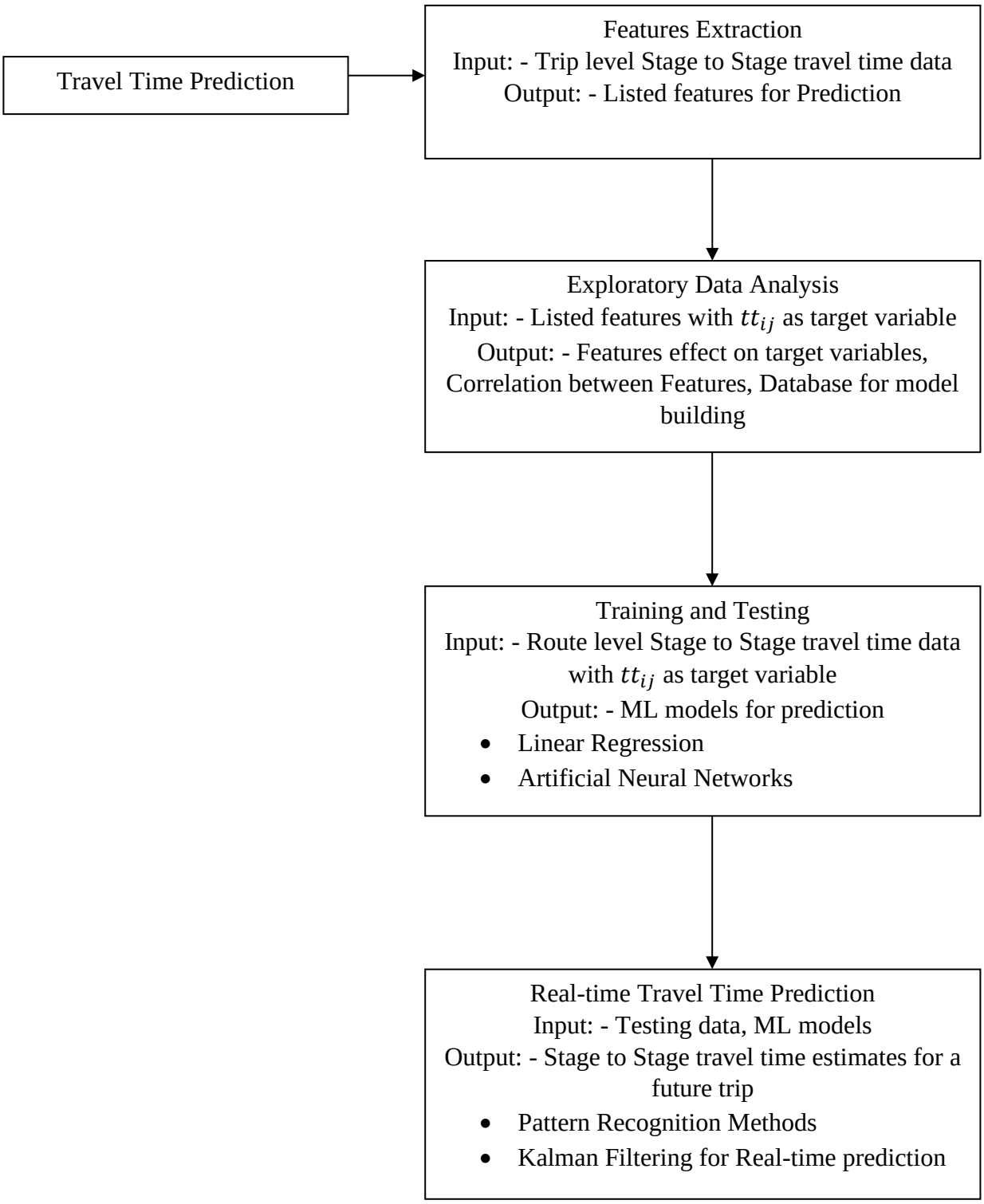


Figure 6.11 Methodology flowchart for Travel Time Prediction

6.3.1 Methodology for Travel Time Prediction

We have discussed the methodology for crowding prediction in Chapter 5 and we have seen that the main explanatory variables used for crowding prediction are historical estimates of the target variable.

In this study, we propose a similar methodology for travel time prediction. There are some additional explanatory variables like Dwell time and Occupancy which better explain the travel time between a given stage pair for a given trip.

We describe the methodology for predicting the travel time tt_{ij} between two stages for a given trip on a given day.

Factors affecting the travel time between two stops

Based on the literature review, the following factors have been identified that affect the travel time between two stages.

- Time of day
- Day of Week
- Month of Year
- Dwell time at the first stop
- Occupancy in the bus
- Directional changes

If there are more people boarding and alighting at a stop, then the dwell time at the stop increases which will in turn increase the travel time between two stops. Therefore, the dwell time at a stop will be an important explanatory variable for travel time prediction.

Based on the literature review, it is also clear that as buses get more crowded, the travel time between the origin and destination increases. Likewise, as the travel time increases, the buses get more crowded. Therefore, occupancy in the bus will also be an important explanatory variable for travel time prediction.

Methodology for Stage to Stage Travel Time Prediction based on the estimates of previous days:

For a selected route and a given schedule, there are multiple trips between an origin-destination pair on a given day. Also, the time at which the first trip of a particular schedule can alter across

days. Therefore, the combination of schedule, run number, and time at which the trip started, we get a unique trip.

The travel between each stage pair of a unique trip obviously depends on the travel time between the same stage pairs of the same trips on previous days. The response variable is the travel time for a selected trip on current day while the explanatory variables are the travel times for the same trip on previous days.

We use the same notations that we used for crowding prediction in Chapter 5.

Suppose that we want to predict the travel time of a given stage pair i, j for a selected route, run and schedule on a given day based on the travel times of same run and schedule for the same route on first seven days as baseline explanatory variables, the response variable and explanatory variables are as shown in Table 6.5

Table 6.5 Response variable and Explanatory variables for travel time prediction A

Y (Response Variable)	X (Explanatory Variables)
$Travel\ Time\ tt_{ijsqtd}$	$Interchanges\ for\ same\ o-des\ for\ same\ schedule\ on\ first\ 7\ days$ $tt_{ijs^1q^1t^11}, tt_{ijs^1q^1t^12}, tt_{ijs^1q^1t^13}, \ldots . tt_{ijs^1q^1t^17}$
	$Day\ of\ Week_{ijsqtd}$ (Binary Variable)
	$Boarding\ Stage\ name_{ijsqtd}$ (Binary Variable)
	$Alighting\ Stage\ name_{ijsqtd}$ (Binary Variable)
	Other Explanatory Variables
	$Service\ type_{ijqstd}$ (Binary Variable) (if multiple schedules are used)
	$Time\ of\ Day_{ijsqtd}$ (Binary Variable) (1-hour intervals)
	$Occupancy_{ijsqtd}$
	$Dwell\ time_{isqtd}$
	$combination\ of\ s, q, and\ t\ represents\ unique\ run$
	$1 < q < Q_r$ ($\max runs\ on\ a\ day$)

All the explanatory variables which are not numerical are represented as binary variables by an array of 0's and 1's. For example, Monday is represented as [1,0,0,0,0,0,0] and Tuesday is

represented as [0,1,0,0,0,0,0]. The final dataset used for prediction will have all these arrays as columns out of which one column will be kept as a baseline and will not be used for prediction.

Both linear (Linear Regression) and non-linear (Artificial Neural Networks) models can be developed that gives us the functional relationship between the explanatory variables and the response variable.

Similarly, we can use the travel time estimates from the previous runs on the same day as explanatory variables for prediction. We use the methodology already discussed in Chapter 5 for crowding prediction.

Kalman Filtering for real-time travel time prediction:

Kalman filtering was used in several studies for travel time prediction in real-time. Once we have a historical travel time prediction model, we can update the historical model as we get more and more measurements in real-time using Kalman Filtering.

Demonstration of travel time prediction is outside the scope of this study.

6.4 Summary

This chapter provides a detailed methodology for travel time estimation and prediction. We have seen the approximate travel time estimates obtained from the ETM data and then we discussed the methodology for obtaining actual travel time estimates from the GPS data. We also discussed the machine learning application to obtain the functional relationship between actual travel time estimates from GPS data and approximate travel time estimates from ETM data. We then provided a detailed methodology for predicting the travel time between two stages for a future trip of a selected route and on a given day.

Now that we have seen the methodologies for obtaining the crowding and travel time on each link of a given route, in the next chapter, we will discuss the detailed methodology for finding an optimal route for a given origin and destination pair and the implementation of the hierarchical algorithm developed on a sample network.

CHAPTER 7

IMPLEMENTATION AND ILLUSTRATIVE APPLICATION OF MULTI-OBJECTIVE OPTIMIZATION FOR CROWDING AND TRAVEL TIME

7.1 Overview

In this chapter, we will discuss the various formulations of the shortest path problem with various combinations of crowding, travel time, and distance as objectives. We will then discuss the methodology for solving the shortest path problems formulated. A hierarchical minimum crowding path algorithm is developed where we optimize for both crowded duration and travel time. Lastly, we provide an illustrative application of the hierarchical minimum crowding path algorithm on a sample network.

7.2 Formulating the shortest path problem

In this study, we have defined both Single Objective formulations and Multi Objective formulations to find an optimal route between an origin-destination pair.

We have considered the following basic objectives for the shortest path problem:

- Crowded duration as the total trip time on crowded links
- Crowded length as the total path length on crowded links
- Crowded fraction in terms of travel time as the ratio of cumulative crowded duration to that of total trip duration
- Crowded fraction in terms of length as the ratio of cumulative crowded length to that of total path length
- Standing duration as the time from origin until first edge when seat becomes available

If our objective is to minimize crowded duration or crowded length, it might become a very simplistic formulation. We miss out routes that might be that might be taking longer time than usual or are lengthy on which passengers experience crowding for very small fractions of total travel time or total length of the route. For passengers who give more preference to experience least crowding, longer duration routes and lengthier routes cannot be ruled out.

Minimizing the crowded fraction in terms of travel time or in terms of length might be a better way of formulating the shortest path problem. If we consider crowded fraction in terms of travel

time as our objective function, we might select a route that takes too long to reach a destination which we do not want. However, if we consider crowded fraction in terms of length, passengers who prefer traveling longer distances to avoid crowding in public transit can benefit.

If we define crowding as the standing duration, it might capture the real discomfort of the passengers but during the peak hours, buses on all the routes might be running full and because of which the route with minimum standing duration is also the route with minimum travel time. So, considering the standing duration alone will not help us find an optimal solution in terms of crowding and travel time.

The best formulation ultimately should capture the discomfort of the passengers. Standing duration, probability of seat availability at the time of boarding, convenience to manoeuvre while inside the bus are other important factors that needs to be considered.

Additionally, we have also considered the following multi-objective variants:

- Least crowded duration among the tied minimum travel time routes
  The advantage of using this formulation is that we minimize the travel time and also crowded duration up to some extent. But what if there is only one route with minimum travel time which is highly crowded, and also what if all the routes that take least travel time are very crowded. This formulation might not always give us more information about crowding for the transit users.

- Least travel time among tied minimum crowded duration paths
  The advantage of using this formulation is that we are minimizing the crowded duration and also the travel time. But what if all the routes that have least crowded duration take very long time or are very lengthy and not be feasible, and also what if there is only one path that have the least crowded duration and if it's a very lengthy route for it to be a recommended path.

- Minimize weighted sum of travel time and crowding duration with pre-assigned weights for travel time and crowded duration
  Slight advantage of this formulation compared to that of the previous ones is that we can give more weightage to the parameters that are important for transit users based on the inputs received from them. However, for this formulation to work well in real time,

we need inputs from users. So, arriving at the pre-assigned weights is a challenging task as it involves public participation.

- Minimize travel time subject to crowded duration less than crowded duration threshold
  The major advantage of this formulation is that we can minimize the crowded duration to a great extent if we set a good threshold but such routes might not be possible at all in real time (during peak hours) and even though there exists routes as such, the travel time may still be very high for it to be a feasible route

- Minimize crowded duration subject to travel time less than travel time duration threshold
  The major advantage of this formulation is that we can minimize the travel time duration but all the routes which have travel time less than threshold travel time might perform very poorly with respective to crowded duration and the crowding problem might go unaddressed

- Multi objective optimization of both crowded duration and travel time (pareto optimal solutions)
  As it is nearly impossible or (impossible in some cases) to obtain a solution that minimizes all the objectives, the best we can do is to attain at a set of efficient solutions that remain non-dominated by the rest of the paths. Pareto optimal solutions is the best we can do if we consider both crowded duration and travel time. The ultimate aim of this study is to come up with a pareto based optimal solution.

7.3 Solving the shortest path problem

Consider a directed network $G = (N, A)$ with an arc cost c_{ij} associated with each arc $(i, j)\ \epsilon\ A$. The directed network G has two distinct nodes s, d called origin and destination respectively.

The nodes represent the bus stops and the arcs represent the road connecting two bus stops. Initially, we will consider two levels of crowding, low and high.

If the crowding on a bus for a particular arc (i, j) is low, then we define c_{ij} as 0. If the crowding on a bus for a particular arc (i, j) is high, then we defined c_{ij} as 1.

The length of an arc connecting two nodes i, j is defined as l_{ij} and it represents the distance between two bus stops. The travel time for a bus connecting two nodes i, j is defined as tt_{ij}.

If the objective function is crowded duration, then the formulation is as follows:

$$\text{Minimize } \Sigma_p \, c_{ij} tt_{ij} \qquad \text{(Eq. 7.1)}$$

such that c_{ij} is 0 or 1

where p is any directed path that connects origin and destination

If the objective function is crowded length, then the formulation is as follows:

$$\text{Minimize } \Sigma_p \, c_{ij} l_{ij} \qquad \text{(Eq. 7.2)}$$

such that c_{ij} is 0 or 1

where p is any directed path that connects origin and destination

Eq. (7.1) and Eq. (7.2) can be solved using Dijkstra's shortest path algorithm.

The objective function for crowded fraction in terms of length is defined as follows:

$$\text{Minimize } \frac{\Sigma_p \, c_{ij} \, l_{ij}}{\Sigma_p \, l_{ij}} \qquad \text{(Eq. 7.3)}$$

such that c_{ij} is 0 or 1

where p is any directed path that connects origin and destination

The objective function for crowded fraction in terms of travel time is defined as follows:

$$\text{Minimize } \frac{\Sigma_p \, c_{ij} \, tt_{ij}}{\Sigma_p \, tt_{ij}} \qquad \text{(Eq. 7.4)}$$

such that c_{ij} is 0 or 1

where p is any directed path that connects origin and destination

In Eq. 7.3 and Eq. 7.4, there are two decision variables are that drive the objective function which are not additive. Hence the objective function cannot be minimized using the standard Dijkstra's shortest path algorithm. Therefore, we need to use multi objective shortest path algorithms to obtain a set of efficient paths for formulations in Eq. 7.3 and Eq. 7.4.

If our objective is to find the least crowded duration among the tied minimum travel time routes, then we need to do the following:

First, we need to find the shortest paths in the network where arc costs are defined as t_{ij} . Then a new network needs to be formed with only the shortest paths obtained. Then we need to minimize $\Sigma_p \, c_{ij} tt_{ij}$ to get the least crowded route among the minimum travel time routes.

If our objective is to find the least travel time among tied minimum crowded duration paths, the we need to do the following:

First, we need to find the least crowded duration routes by minimizing $\Sigma_p \, c_{ij} t_{ij}$. Then a new network needs to be formed with only the shortest paths obtained. Then we need to minimize $\Sigma_p \, tt_{ij}$ in the new network to get the least travel time path among tied minimum crowded duration paths.

If our objective is to minimize the weighted sum of travel time and crowding duration with pre-assigned weights for travel time and crowded duration, then we need to do the following:

We need to solve the shortest path problem whose objective function is defined as $\Sigma_p \, W_1 * c_{ij} tt_{ij} + W_2 * tt_{ij}$ where W_1 and W_2 are the weights associated with crowded duration and travel time respectively.

Based on the literature review, it is clearly that the commonly used multi objective shortest path algorithms that are used for variety of purposes are as follows:

- Non dominated pareto optimal solutions method
- Lagrangian Relaxation method

7.4 Minimum Crowding Path Algorithm

In this study, we have developed a hierarchical algorithm for obtaining the non-dominated pareto optimal solutions where we optimize for both crowded duration and travel time.

Consider a directed network $G = (N, A)$. The directed network G has two distinct nodes s, t called origin and destination respectively.

We define the following directed networks:

$G1$ as a subnetwork of G having only low crowded links of the directed network G.

$G2$ as a subnetwork of G having low and medium crowded links of the directed network G.

$G3$ is the same as the directed network G.

We will find the shortest travel time path in $G1$, $G2$, and $G3$ using Dijkstra's Algorithm to get $tt1^*$, $tt2^*$, and $tt3^*$ respectively.

If $tt1^* \leq tt2^*$ and $tt1^* \leq tt3^*$, it means that there is a path between origin and destination that is not having any crowded links and it is also the path with least travel time in the directed network G. It means that we have obtained our optimal path

If $tt2^* < tt1^*$ and $tt2^* \leq tt3^*$, we will formulate a Multi-objective formulation (MOF 1) where we optimize $G2$ for crowded duration and travel time using K-shortest paths-based algorithm to get the Non-Dominated Set 1. In this case, we define the crowding coefficient as 0 for low and 1 for medium. Crowded duration is the travel time on medium crowded links in the directed network $G2$.

If $tt3^* < tt2^*$ and $tt3^* < tt1^*$, we will formulate two Multi-objective formulations (MOF 2 and MOF 3) where we optimize $G3$ for crowded duration and travel time using K-shortest paths-based algorithm to get Non-Dominated Set 2 and Non-Dominated Set 3

For Multi-objective formulation 2, we define the crowding coefficient as 0 for low and medium and 1 for high. Crowded duration 1 is the travel time on high crowded links in the directed network $G3$.

For Multi-objective formulation 3, we define the crowding coefficient as 0 for low and 1 for medium and high. Crowded duration 2 is the travel time on medium and high crowded links in the directed network $G3$.

We will then form a candidate set by combining Non-Dominated Set 2 and Non-Dominated 3. We get the final pareto-optimal solution set by removing the dominated paths from the candidate set in terms of travel time, medium crowded duration, and high crowded duration.

The flowchart for the hierarchical algorithm is depicted in Figure 7.1

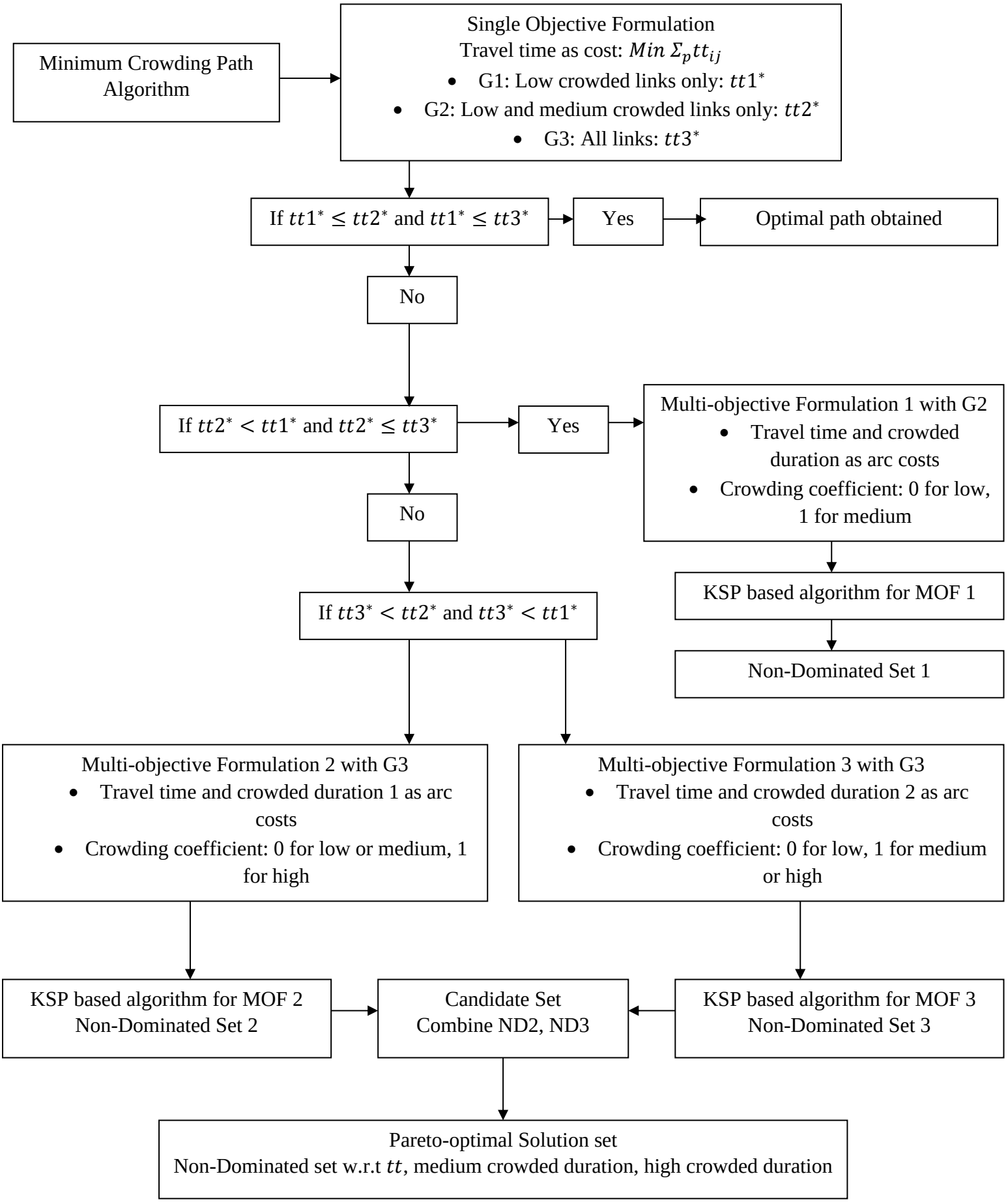


Figure 7.1 Flowchart for Minimum Crowding Path Algorithm

7.5 KSP based algorithm for obtaining Pareto-optimal solution set

We describe the methodology for obtaining a Pareto-optimal solution set using a K-shortest paths-based algorithm.

Consider a directed network $G = (N, A)$. The directed network G has two distinct nodes s, d called origin and destination respectively.

Let travel time and crowded duration be the two costs associated with each arc.

The travel time of an arc (i, j) is defined as tt_{ij} and the crowded duration of an arc (i, j) is defined as C_{ij}.

We find an optimal travel time path between s, d by minimizing $\Sigma_p tt_{ij}$ for the directed network G. Let the travel time for the obtained path be tt^*. Let the crowded duration of the optimal travel time path be C.

Similarly, we find an optimal crowded duration path between s, d by minimizing $\Sigma_p C_{ij}$ for the directed network G. Let the crowded duration for the obtained path be C^*. Let the travel time of the optimal crowded duration path be tt.

Now we start from the path (C^*, tt) and reach (C, tt^*) by increasing crowded duration in every iteration. We will add a new path to the solution set if the new path is not dominated by the previous path or same as the previous path. If the new path obtained is better than the previous paths in the solution set, we remove the previous paths which get dominated by the new path and add the new path to the solution set. We repeat this process until we reach (C, tt^*) and the final solution set becomes the Non-Dominated Pareto optimal solution set.

Similarly, we can also start from (C, tt^*) and reach (C^*, tt) by increasing the travel time in every iteration.

We find the Non-Dominated Set 1, 2, and 3 using this method for Multi-objective Formulation 1, Multi-objective Formulation 2, and Multi-objective Formulation 3 respectively.

In this study, we have obtained the three best optimal solutions from the Non-Dominated set by taking the minimum of the Euclidean distance between each path and ideal path

The flowchart for a Multi-objective optimization problem using K-shortest paths-based algorithm for obtaining the Pareto-optimal solution set for is depicted in Figure 7.2

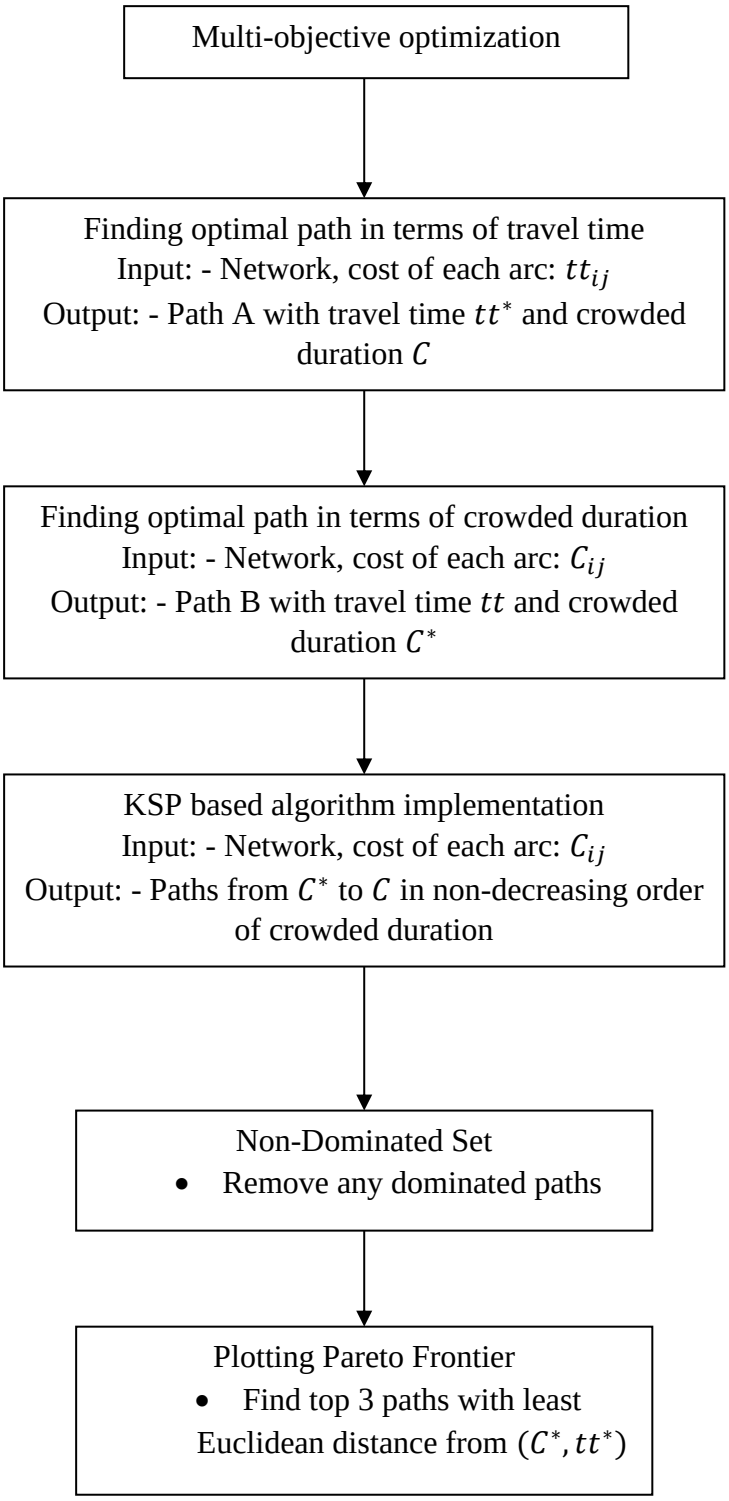


Figure 7.2 Flowchart of KSP based algorithm for finding pareto-optimal solutions

7.6 Illustrative application of the minimum crowding path algorithm

We have implemented the hierarchical minimum crowding path algorithm on a sample network.

We have considered three levels of crowding low, medium, and high. The crowding cost is 0 for low crowded link, 1 for medium crowded link, and 2 for high crowded link.

We have considered travel time for each link as any real number between 3 and 15 (inclusive of both).

We have generated the crowding cost and travel time cost using uniform distribution. Network structure was taken from already existing open source networks from GitHub and we created a sample network as depicted in Figure 7.3

```
<NUMBER OF ZONES> 24
<NUMBER OF NODES> 24
<FIRST THRU NODE> 1
<NUMBER OF LINKS> 76
init_node,term_node,crowding cost,travel time
1,2,2,10.800295256779876
1,3,1,4.244680039740395
2,1,1,6.260987316461521
2,6,1,12.311922088885815
3,1,2,14.90550127868568
3,4,2,14.667944640321245
3,12,2,6.008687088898412
4,3,1,4.496093867364162
4,5,0,13.520532613893277
4,11,0,5.401839384149734
5,4,0,13.007428158666116
5,6,2,7.027937901708494
5,9,2,10.54016664725658
6,2,1,4.825056718459292
6,5,2,7.368673233000282
6,8,1,14.470881529056854
7,8,0,4.018561040805409
7,18,1,12.709695513537202
8,6,2,3.2945009206047766
8,7,0,5.368301198442975
8,9,0,9.756340503881402
8,16,2,6.687031108961779
9,5,2,9.535886323426313
9,8,1,12.260301966138261
9,10,1,7.667770304473619
10,9,0,10.983968821685181
10,11,1,12.422172458430882
10,15,2,7.597467716744481
10,16,2,6.0905006481251185
10,17,0,14.09309317810438
11,4,0,8.655814790141935
11,10,1,14.992510089633981
11,12,2,3.9516401003167956
11,14,1,8.334836542238195
```

Figure 7.3 Sample network

We have implemented the hierarchical minimum crowding path algorithm on the sample network. The results obtained are depicted in Figure 7.4, Figure 7.5 and Figure 7.6

```
Enter origin:1
Enter destination:16
1 or 16 not present in g1!
No path between 1 and 16 in g2!
tt1 =  -1 , tt2 =  -2 , tt3 =  34.852432462044185
cd_max =  0 , cd1_max =  15.615242332669808 , cd2_max =  34.852432462044185
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 13, 24, 23, 22, 21, 20, 18, 16], 81.70373792593617, 24.130555622292434, 9.107644190706223]
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 11, 10, 17, 16], 48.08482744945122, 19.237190129374376, 14.754544141972465]
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 11, 10, 16], 34.852432462044185, 19.237190129374376, 15.615242332669808]
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 13, 24, 23, 22, 15, 10, 9, 8, 7, 18, 16], 105.95837644415045, 60.98958093848573, 6.008687088898412]
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 11, 10, 9, 8, 7, 18, 16], 74.32474315256397, 48.447731447891144, 9.52474168454469]
optimal path, travel time, medium crowded duration, high crowded duration
[[1, 2, 6, 8, 7, 18, 16], 59.90163942554401, 43.73304297032117, 10.800295256779876]
best optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 11, 10, 16], 34.852432462044185, 19.237190129374376, 15.615242332669808]
best optimal path, travel time, medium crowded duration, high crowded duration
[[1, 3, 12, 11, 10, 17, 16], 48.08482744945122, 19.237190129374376, 14.754544141972465]
best optimal path, travel time, medium crowded duration, high crowded duration
[[1, 2, 6, 8, 7, 18, 16], 59.90163942554401, 43.73304297032117, 10.800295256779876]
```

Figure 7.4 Output for sample network with origin as node '1' and destination as node '16'

Note that $tt = -1$ if either origin node or destination node is not present in the network and $tt = -2$ if there is no path for the origin-destination pair entered by the user.

```
Enter origin:5
Enter destination:20
No path between 5 and 20 in g1!
No path between 5 and 20 in g2!
tt1 =  -2 , tt2 =  -2 , tt3 =  36.445481022125676
cd_max =  0 , cd1_max =  28.777710717652052 , cd2_max =  36.445481022125676
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 4, 11, 14, 15, 22, 21, 20], 64.8557193344124, 8.334836542238197, 5.100787366977213]
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 4, 11, 14, 15, 19, 20], 42.48496780568225, 8.334836542238197, 15.740863720628205]
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 9, 10, 15, 19, 20], 36.445481022125676, 7.667770304473624, 28.777710717652052]
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 4, 11, 12, 13, 24, 23, 22, 21, 20], 86.4717774997809, 15.645331743710743, 3.9516401003167956]
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 6, 8, 7, 18, 20], 46.94181461785122, 27.18057704259406, 14.392936376814184]
optimal path, travel time, medium crowded duration, high crowded duration
[[5, 6, 8, 16, 18, 20], 41.235612171786315, 14.470881529056857, 26.764730642729457]
best optimal path, travel time, medium crowded duration, high crowded duration
[[5, 4, 11, 14, 15, 19, 20], 42.48496780568225, 8.334836542238197, 15.740863720628205]
best optimal path, travel time, medium crowded duration, high crowded duration
[[5, 6, 8, 16, 18, 20], 41.235612171786315, 14.470881529056857, 26.764730642729457]
best optimal path, travel time, medium crowded duration, high crowded duration
[[5, 6, 8, 7, 18, 20], 46.94181461785122, 27.18057704259406, 14.392936376814184]
```

Figure 7.5 Output for sample network with origin as node '5' and destination as node '20'

```
Enter origin:2
Enter destination:18
2 or 18 not present in g1!
tt1 =  -1 , tt2 =  44.86080032992284 , tt3 =  39.154597883857946
cd_max =  39.492499131479875 , cd1_max =  12.371794265915273 , cd2_max =  39.154597883857946
optimal path, travel time, medium crowded duration, high crowded duration
[[2, 1, 3, 12, 13, 24, 23, 22, 21, 20, 18], 83.72418140355639, 26.15099909991266, 9.107644190706223]
optimal path, travel time, medium crowded duration, high crowded duration
[[2, 1, 3, 12, 11, 14, 15, 22, 21, 20, 18], 69.57581793415098, 18.84050389844011, 17.724486153329714]
optimal path, travel time, medium crowded duration, high crowded duration
[[2, 6, 8, 16, 18], 39.154597883857946, 26.782803617942673, 12.371794265915273]
optimal path, travel time, medium crowded duration, high crowded duration
[[2, 6, 8, 7, 18], 44.86080032992284, 39.492499131479875, 0]
best optimal path, travel time, medium crowded duration, high crowded duration
[[2, 6, 8, 16, 18], 39.154597883857946, 26.782803617942673, 12.371794265915273]
best optimal path, travel time, medium crowded duration, high crowded duration
[[2, 6, 8, 7, 18], 44.86080032992284, 39.492499131479875, 0]
best optimal path, travel time, medium crowded duration, high crowded duration
[[2, 1, 3, 12, 11, 14, 15, 22, 21, 20, 18], 69.57581793415098, 18.84050389844011, 17.724486153329714]
```

Figure 7.6 Output for sample network with origin as node ‘2’ and destination as node ‘18’

If a user enters origin and destination, we return all the optimal paths in the Non-Dominated set and also the best three in terms of travel time, medium crowded duration, and high crowded duration.

7.7 Summary

In this chapter, we have seen the various formulations of the shortest path problem with various combinations of crowding, travel time, and distance as objectives. We have then discussed the methodology for solving the shortest path problems formulated. We have then discussed the hierarchical minimum crowding path algorithm that is developed in this study where we have optimized for both crowded duration and travel time. We have also seen an illustrative application of the hierarchical minimum crowding path algorithm on a sample network. If we obtain the crowding and travel time data in real-time, we can use the hierarchical minimum crowding path algorithm developed to recommend optimal transit routes to users in real-time.

CHAPTER 8

SUMMARY AND FUTURE WORK

In this study, we have put forward detailed methodology for crowding estimation and prediction, travel time estimation and prediction and Multi-objective optimization. We have extracted and processed ETM data for different routes, schedules, and service types. Using the extracted ETM data, we have estimated boardings, alightings, and stage to stage interchanges for multiple trips of selected routes and on multiple days. We have then discussed the methodology for estimating the number of pass-holders in a particular trip.

The boardings and alightings actually happen at a stop in reality. Therefore, there is a need to estimate the demand at stop-level. We have developed a hierarchical algorithm for mapping the stages to stops for a particular trip. Using the crowding estimates obtained, we have described the methodology for estimating the crowding level in each link of a given route and we have also shown an illustrative application for the same.

We have then discussed the detailed methodology for predicting stage to stage interchanges for a future trip based on the interchanges for the same trip on previous days, based on the interchanges of the last few trips on the same day. We have also seen how we can predict the number of interchanges of a selected stage pair based on the occupancies on the previous links of the same trip. An illustrative application of interchanges prediction using Machine Learning was also shown in this study. Once we have the interchanges for a future trip, we have provided the methodology for obtaining the occupancy on each link of the entire route for that trip.

Using the ETM data, we have also estimated the stage to stage approximate travel times for multiple trips of selected routes and on multiple days. We have also discussed the methodology for extracting the stage to stage actual travel times from the GPS data. We have also described the process of deciding on whether ETM data is sufficient for travel time estimation or not. Detailed methodology for applying Machine Learning to understand the functional relationship between the approximate travel time estimates and actual travel time estimates has also been discussed. In this study, we have also put forth the detailed methodology for travel time prediction using ETM data.

We have discussed several shortest path problem formulations based on crowding, travel time and length. We have also seen the methodology for solving the shortest path problems formulated. In this study, we have developed a novel hierarchical minimum crowding path algorithm which is a Multi-objective optimization algorithm to optimize for crowded duration and travel time. The Multi-objective optimization algorithm uses Yen's K-shortest paths algorithm to obtain the Non-Dominated Pareto-optimal Solution set. We have also implemented the algorithm developed in this study on a sample network with sample crowding and travel time data.

There is a lot of scope for future work on this project. The crowding estimation from the ETM data can be done for multiple routes and on multiple days to create a large historical dataset. A novel pass-holder model needs to be developed that works for the Chennai MTC buses. In this study, we have estimated the demand at stage level. The hierarchical algorithm for mapping the stops to stages need to be automated and implemented to get the demand at stop level. The crowding prediction machine learning model developed in this study gave poor results mainly because the data is less and it is time-series based. Deep Learning LSTM time-series based models can be developed which will give better predictions.

The historical offline models can be further improved using Kalman Filtering to make predictions in real-time. In this study, we haven't implemented travel time prediction models. Further work can be done on predicting the actual travel time between stages using only ETM data. The Multi-objective optimization algorithm developed in this study need to be made much more efficient for a real-time application. Also, the algorithm can be implemented on a transit network which is different from a general graph. A real-time application can be developed by integrating the predicted crowding level, predicted travel time and the Multi-objective optimization algorithm.

REFERENCES

K. Dziekan and K. Kottenhoff, “Dynamic at-stop real-time information displays for public transport: effects on customers,” Transportation Research Part A, vol. 41, pp. 489–501, 2007.

Erik Jenelius, “Data-Driven Bus Crowding Prediction Based on Real-Time Passenger Counts and Vehicle Locations”.

Yunqi Zhang “Determination of Bus Crowding Coefficient Based on Passenger Flow Forecasting”.

KFH Group, Transit Capacity and Quality of Service Manual, TRB,Washington, DC, USA, 3th edition, 2013.

A. Ceder, “Public Transit Planning and Operation: Theory, Modelling and Practice”, Butterworth-Heinemann Ltd, 2007.

Alejandro Tirachini, David A. Hensher, John M. Rose, “Crowding in public transport systems: effects on users, operation and implications for the estimation of demand”

Patrick Ngatchou, Anahita Zarei, and M. A. El-Sharkawi, “Pareto Multi Objective Optimization”.

Sushant Sharma, Satish V. Ukkusuri, and Tom V. Mathew, “Pareto Optimal Multiobjective Optimization for Robust Transportation Network Design Problem”.

J.C.N. Climaco, E.Q.V. Martins, “A bicriterion shortest path algorithm European Journal of Operational Research”.

“When is a bus full? A study of perception” by B. Theler & K.W. Axhausen

Puong, A. (2000). “Dwell time model and analysis for the MBTA Red Line”.

Mohd Mahudin, N. D., T. Cox and A. Griffiths (2012), “Measuring rail passenger

crowding: Scale development and psychometric properties”.

Mohring, H. (1972), “Optimization and scale economies in urban bus transportation”.

Oldfield, R. H. and P. H. Bly (1988), “An analytic investigation of optimal bus size”.

Peng Chen, Yu Nie, “Bicriterion shortest path problem with a general nonadditive cost”.

Antsfeld L. & Walsh T. (2012). “Finding multi-criteria optimal paths in multi-modal public transportation networks using the transit algorithm” 19th Intelligent Transport Systems World Congress, ITS 2012

Fan L., Mumford C. L. and Evans D. (2009). “A Simple Multi-Objective Optimization Algorithm for the Urban Transit Routing Problem” IEEE Congress on Evolutionary Computation (CEC 2009)

Mainali M.K., Shimada K., Mabu S., Hirasawa K. (2008). “Multi-objective optimal route search for road networks by dynamic programming” SICE Annual Conference, Japan

Wen F. & Lin C. (2010). “Multiobjective Route Selection Model and Its Solving Method Based on Genetic Algorithm” International Journal of Information Systems for Logistics and Management, 5(2), 1-8

Yuh-Wen Chen, Chi-Hwang Wang, Sain-Ju Lin (2008). “A multi-objective geographic information system for route selection of nuclear waste transport” Omega, 36(3), 363-372

Zhimin, A.W. & Xianfeng, B.L. (2003). “The model and algorithm for finding the optimal route in a dynamic road network” Proceedings of Intelligent Transportation Systems, vol. 2, 1495–1498

Rohit Verma, Surjya Ghosh, Mahankali Saketh, “ComfRide: A Smartphone based System for Comfortable Public Transport Recommendation”

Jiaqi Wang, Yunyao Lu, “A Social-aware route recommendation system for intelligent transportation”

Garvita Bajaj, Rachit Agarwal, “Towards Building Real-Time, Convenient Route Recommendation System for Public Transit”