TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO: T09/072

Optimal Route Computation for Public Transport with Minimum Travelling Time & Travel Cost: A Case Study of Pokhara City

by

Rabin Thapa

A THESIS

SUBMITTED TO THE DEPARTMENT OF CIVIL ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE IN TRANSPORTATION ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

LALITPUR, NEPAL

NOVEMBER, 2018

COPYRIGHT

TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PULCHOWK CAMPUS
DEPARTMENT OF CIVIL ENGINEERING

The undersigned certify that they have read and recommended to the Institute of Engineering for acceptance, a final thesis entitled "**Optimal Route Computation for Public Transport with Minimum Travelling Time & Travel Cost: A Case Study of Pokhara City**" submitted by Rabin Thapa (2072/MST/259) in partial fulfillment of the requirements for the degree of Master of Science in Transportation Engineering, Nepal is a record of works carried out by him under my supervision and guidance.

Supervisor

_____

Jagat Kumar Shrestha, PhD
Associate Professor
IOE, Pulchowk Campus, Nepal

External Examiner

_____

Pradeep Kumar Shrestha, PhD
Lecturer
IOE, Pulchowk Campus, Nepal

Program Coordinator

_____

Anil Marsani
M.Sc. Program in Transportation Engineering
Department of Civil Engineering
IOE, Pulchowk Campus, Nepal

Date: _____

ABSTRACT

In road networks, it is imperative to discover a shortest way to reach the final destination. When an individual is new to a place, lots of time is wasted in finding the destination. With the advancement of technology, various navigation applications have been developed for guiding private vehicles, but few are designed for public transportation.

The main aim of this study is to find the possible shortest path in terms of minimum time and cost to reach specific destination for an individual. It needs an appropriate algorithm to search the shortest path.

This study uses Dijkstra's algorithm in Microsoft Visual Studio C++ 2017 to find the shortest path with respect to minimum travel time and travel cost. Public transportation network of Pokhara city is selected for the case study of this research. The results of this analysis indicates that when the "time" impedance was used by the algorithm, it generated the shortest path between the origin and destination along with the path to be followed. This study formulates a framework for generating itinerary for passengers in a transit network that allows the user to find the optimal path with minimum travel time and cost.

Key words: Dijkstra's algorithm, Optimal path, Public transport, Route computation, Shortest path

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Jagat Kumar Shrestha, who was tremendously helpful and offered invaluable assistance, guidance and enlightenment and allowed me the open space to work in my own way. Apart from the thesis work, his practical sense of approach in teaching helped me a lot to better understand some key aspects of Operations Research.

I would like to thank Department of Civil Engineering, Institute of Engineering, Pulchowk Campus for providing the opportunity to carry out my thesis work. I would also like to express my heartfelt gratitude to Mr. Anil Marsani, Program Coordinator of Master of Science in Transportation Engineering for his cooperation, criticisms and valuable suggestions which boosted me up to carry out the research work.

I would also like to express my gratefulness to Pokhara-Lekhnath Metropolitan City Office, Statistics Office Kaski (Central Bureau of Statistics) and respective Bus Service Providers (Pokhara-Lekhnath) for providing me guidance with necessary information and data required for my thesis work.

I would also like to express my profound gratitude to Er. Anand Gupta, Er. Arjun Adhikari, Er. Sanat Acharya and Er. Madhav Pokhrel for their constant support and guidance.

Special thanks to my senior brother Er. Purusottam Baral and all my graduate friends, especially Er. Niraj Bohora and Er. Durga Nanda Gupta for sharing the literature and invaluable assistance. I would also like to thank all my colleagues, friends and well-wishers from my bottom of my heart.

Last but not the least, I wish to express my love and gratitude to my beloved family members who have always been there for me at my every ups and downs and I am blessed to have their unconditional love, endless support and encouragement which motivated and revived me to pursue this degree.

Rabin Thapa

2072/MST/259

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CBS	Central Bureau of Statistics
G	Graph of network
GA	Genetic Algorithm
GIS	Geographic Information System
i	Nodes number (1, 2, ….)
j	Junction
L	The set of m directed links
m	Total number of links between the nodes
min.	Minimum
MRT	Mass Rapid Transit
MSPP	Multi-criteria Shortest Path Problem
n	Total number of nodes in the network
N	The set of n nodes
O-D	Origin-Destination
PDA	Personal Digital Assistance
PLMC	Pokhara Lekhnath Metropolitan City
SPPs	Shortest Path Problems
TdBiSP	Time-dependent Bi-criteria Shortest Path Problem
VNS	Variable Neighborhood Search

CHAPTER 1. INTRODUCTION

1.1 Background

Public transportation plays an important role in urban areas to carry huge no. of passengers which helps to reduce traffic congestion. Nowadays, passengers in urban public transport systems do not only seek a shorter travelling time but they also ask for optimizing other criteria such as cost and effort (O. Dib, 2016). Time being one of the most valuable assets, people want to use their time in optimal way. Past researchers have mostly been focusing on planning bus stop locations to address travel demands and to improve transit accessibility (Wiransinghe and Ghoneim, 1981; Chien and Qin, 2004; Bagloee and Ceder, 2011) including the consideration of local topography (Furth and SanClemente, 2006). With the already planned and existing public transportation network, a guidance and decision support information system is necessary to facilitate its use. An accurate and detailed public transport journey planner system can provide multi-modal public transport information on the internet, supports public access for pre-trip planning and optimal route searching. Users can search their optimal routes based on their own preferences such as shortest travelling time, least cost, least transfer, least walking distance or even a preferred mode of transport (Lilian S.C. Pun-Cheng, 2015).

An efficient routing system is an indispensable tool for any route network. Manual routing of even a small scale route is out of the realm of human capacity and is best left to computers. “Optimization” means finding an alternative with the most cost effective or highest achievable performance under the given constraints by maximizing desired factors and minimizing undesired ones (Lilian S.C. Pun-Cheng, 2015). **Route Optimization is the process of determining the most efficient route.** Route computation, being a branch of path finding, has largely been evolved from the graph theory of spatial computing. From a transportation database, transport routes are modelled as a topological network based on the graph theory. Generally speaking, a graph refers to a collection of vertices or nodes and arcs that connect pairs of nodes. The arc can be a directed or single way only. Nodes represent entities or places with a pair of geographic coordinates. An arc is a physical entity to connect two or more nodes together. It includes roads, bridges, tunnels, and so on. Also, an arc could be uni-directional or bi-directional to indicate the traffic flow along the arc. In the context of

public transportation, a route, being a service with defined directed way provided by public transport operators such as railway, bus and ferry, could be described as a sequence of arcs and nodes. Terminals, bus stops, rail stations and ferry pier are all represented as nodes along a route, whereas any point of interest such as a building or site will be modelled as another set of nodes for representing origin or destination locations (i.e. contact or exchange point where it is possible for people to change from one mode/vehicle to another). In this network model, although it is necessary to model all nodes accurately in space and time, the alignment of an arc can just be virtual (i.e. without following the real alignment of a route on the ground) if its associated journey time and fare information can be provided from a reliable source other than derived by spatial computation (Lilian S.C. Pun-Cheng, 2015).

The main aim of this study is to develop a model for optimization of public transport network in order to facilitate passengers/commuters with efficient itineraries according to their needs and preferences.

1.2 Research Problem:

As in real life, commuters do not only seek short time travels. However, they tend to consider other elements into their journeys such as monetary cost, comfort (quality of mode) and effort (walking distance, number of transfers, waiting time, etc.). Therefore, there is a real need to develop a seamless routing application that provides passengers with efficient itineraries according to their needs and preferences (O. Dib, 2016).

For a long history, Pokhara has relied on bus system. With urbanization and socio-economic sustainable development, the requirement of the urban public transport is increasing. Meanwhile, the significance of public transport planning is emphasized by both the inadequate supply for infrastructure of public transport and the low efficiency of the urban road construction which is far behind the growth of traffic volume. This is the reason the traffic congestion of Pokhara is increasing day by day. Also, the dispersed settlement of people has led to the complicated transit networks. Dispersed location of bus stops and irregular transport route is another reason which causes delay and increase in travel cost of the public transport users. Since, many places in Pokhara cannot be reached using only one means of vehicle (i.e. people are compelled to use

two or more vehicles (buses) to reach their destinations). So, during this entire journey one has to give up extra time and money. As no any city is common to another, every city is unique in itself and has its own identity. So does Pokhara, it has its own nature of transport route network and facilitation of transportation system. To counteract the difficulties faced by commuters of Pokhara city, this study will focus on the development of best model to find the optimum route among the network based on minimum travelling time and travelling cost.

1.3 Objective of the study:

From the perspective of bus travelers, bus trip involves a variety of decision making. The purpose of a bus trip might be different, e.g., to reach destination in least cost, or to take a most comfortable line, or to traverse a sightseeing route. Travelers have to make choices on travel distance, travel time, number of transfers, and monetary cost. The existence of different trip objectives makes bus trip optimization a quite complicated issue. Thus, the research problem goes to how to provide more precise trip guidance for the public bus travelers. The context of this research is the city of Pokhara in Nepal.

The main objective of the study is:

- To develop an optimal route model to find the shortest travel time of public transportation network structure of Pokhara city.
- To find an itinerary of lowest fare (travel cost) for a trip around Pokhara city.

1.4 Scope of the Study:

Pokhara city was chosen for the study of this research. Existing public transport network was considered in the whole city such that a node represented a bus stop/station and link between them represented a transport linkage. The time and cost required to travel from each node to another node were identified respectively. The main variables were travel time and travel cost which governs the entire study. Since, the travel cost required to travel from each respective node to another respective node is same and vice

versa but in case of travel time it is not the same case. So, the travel time was calculated by taking average value among the respective connecting nodes for each different trips.

Then, a geographical network was prepared such that bus stops touching each other formed a link. Square matrix for each node in which time and cost as it's element was prepared. A weighted graph was formed and solved using Dijkstra's Algorithm and an optimal path was achieved. This solution was then rechecked with a different C++ code.

1.5 Assumptions and Limitations of the Study:

- The time required for the user/traveler to reach each origin stops and final point from destination stops (i.e. access and egress time) is not considered i.e. the journey always begins from the origin node and ends at the destination node.
- The location of bus stops was not specified at various nodes (places) so, the usual stopping points were assumed as bus stops.
- Delay caused due to traffic congestion and waiting time at stops were taken as a bulk travel time.
- The case of transfer of vehicle, the number of transfers and the time required during transfer is not studied.
- Since, this research is solely dependent on travel time and travel cost. So, the distance between the nodes has not been incorporated.
- The data for travel time was collected without focusing on the period of time i.e. data was collected both at peak and off-peak hours.
- The seasonal routes connecting to the city are neglected.
- The route operated by Prithvi Bus Service has not been included in the study.
- The speed of the vehicle, condition of road and drivers' aggressiveness has not been taken into account in this study.

1.6 Organization of Thesis:

The entire work has been divided into five chapters and presented as follows:

Chapter 1 includes introduction to the research. Background, research problem, objectives, assumptions, limitations and scope of study are also given in this chapter.

Chapter 2 provides review of relevant literature. The literature reviews are mainly related to the study of optimization of transportation networks and shortest path problem.

Chapter 3 is associated with methodology and data analysis. It provides information regarding study area, collection of necessary data.

Chapter 4 outlines result and discussion. This chapter presents the solutions to find the shortest and cheapest path obtained from this study and brief discussion of them.

Chapter 5 ends with the conclusions and then points out some directions for future research.

CHAPTER 2. LITERATURE REVIEW

Routing is widely studied topic in transport systems but also has a real application in wide range of fields like: energy, military, communication networks. Optimizing connections in transportation networks is a popular problem that arises in many different scenarios, such as car journeys, public transportation, or logistics optimization. People expect computers to assist them with these problems in a comfortable and fast way, whether they are at home, at work, or on a journey. This creates a demand for efficient algorithms to solve these problems. We see that in the existence of many online routing services, mobile navigation devices, and industrial tour planners. The basic concept of a routing algorithm is to model the specific problem in a suitable graph and to compute a shortest path to solve it. For example, routing in public transportation networks is fundamentally different and much more difficult than routing in road networks (Geisberger, 2011). In this chapter, we will first look after the previous studies been carried out with respect to optimization of routes in terms of minimum traveling time and travel cost, and then we will discuss about the methods to solve this problem.

2.1 Previous Studies

Xia Xiaotang studied about the bus trip optimization using GIS (Geographic Information System) of Wuhan city, China whose main objective was to obtain comprehensive optimal bus trip plans. Based on the bus route network model at the directional level, this research develops an improved methodology to generate optimal transit routes. Based on the impedance of travel time, this method incorporates transfer delay into the algorithm and gradually updates the bus route network attributes to realize the loop optimization. Bus trip optimization is a significant component in urban public transport planning. Particularly, the problem of bus transfer needs to be emphasized in order to provide sound trip guidance. This research uses the detailed directional data to explicitly present the bus network features. In this way, the optimization procedure may generate more accurate results to meet the requirements in complicated transport situations. The improved algorithm has been introduced in this research which aims to get the multiple optimal routes. The bus trip optimization based

on travel time impedance is designed to be carried out with a looped procedure. Because general shortest path algorithm can only produce one solution each time, for the purpose to acquire alternative routes, essential changes are made on the network attribute. Finally, experiments are made based on the prototype developed with ArcGIS Engine. Considering passengers' travel psychology, the multiple optimal routes generated by the prototype have the least travel time and minimal transfer times. The results indicate this prototype may provide the attractive and efficient alternative routes for travelers (Xiaotang, 2009).

Dib, Moalic, Manier & Caminada (2016), proposed an advanced heuristic approach whereby a Genetic Algorithm (GA) is combined with a Variable Neighborhood Search (VNS) to solve the Multicriteria Shortest Path Problem (MSPP) in multimodal networks. As transportation modes, they have focused on railway, bus, tram and pedestrian. As optimization criteria, they considered travel time, monetary cost, number of transfers and the total walking time. The proposed approach is compared with the exact algorithm of Dijkstra, as well as, with a standard GA and a pure VNS. Experimental results have been assessed by solving real life itinerary problems defined on the transport network of the city of Paris and its suburbs. Results indicate that the proposed combination GA–VNS represents the best approach in terms of computational time and solutions quality for a real world routing system (O. Dib, 2016).

Pun-Cheng & Chan (2015) conducted a structural analysis of public transport routes in terms of fare and operation patterns. An enhanced route computation algorithm has been proposed in order to provide more reasonable and logical results for different structures. The development and implementation of the programming logic, together with the validation of the enhanced algorithm are also presented. It is found that the traditional approach of selecting closest stops to origin, destination or interchange stops may not satisfy all patterns, especially for cities with a very dense network of public transport stops and for circular routes. To cater for a lot of these special cases, the new approach of stop selection adopts a comparison of the stop sequence within a route with a threshold of commuting behavior. Real cases from a governmental public transportation enquiry system in Hong Kong are extracted for implementation and evaluation; results from which have been proved satisfactory to both system planners and users (Lilian S.C. Pun-Cheng, 2015).

Modesti and Sciomachen (1998) used a linear utility function that incorporates travel time, ticket cost, and inconvenience of transfers. In this paper they have studied the problem of finding Origin-Destination (O-D) shortest paths in urban multimodal transportation networks, aiming at minimizing the overall cost, time and users' discommodity associated with the required paths. They have presented an approach based on the classical shortest path problem on a network representing the urban multimodal transportation system, i.e. the private, the public and the pedestrian modalities. They used an ad hoc utility function for weighing the arcs both with their cost and time and considering at the same time the preference of the users related to all the possible transportation modalities. In particular, a utility measure is presented taking into a proper account the different users' propensities. The proposed approach has been developed for analyzing the urban transportation network of an Italian city (Paola Modesti, 1998).

Hamacher, Ruzika and Tjandra (2006), proposed a backward label-setting algorithm or identifying important solution for all to one multiple criteria time-dependent shortest path. In this paper they generalized the classical shortest path problem in two ways. They considered two objective functions and time dependent data. The resulting problem, called the time-dependent bicriteria shortest path problem (TdBiSP), has several interesting practical applications, but has not gained much attention in the literature. After reviewing relevant literature, they developed a new algorithm for the TdBiSP with non-negative data. Numerical tests show that the superiority of their algorithm compared with an existing algorithm in the literature. Furthermore, they discussed algorithms for the TdBiSP with negative travel times and costs (Horst W. Hamacher, 2006).

Wang (2008) did a study on handling time and fares in a routing algorithm for public transport. Given an origin, destination, and intended departure time, this study proposes two timetable-based algorithms to search for optimal itineraries so that the total travel time for an individual is minimized in a transit network. Itineraries showing the suggested routes with walking access and egress, bus stops, and Mass Rapid Transit (MRT) information had been generated, considering the time to wait for, to transfer between, and to stay in transit vehicles, as well as the time to walk between transit stations. In addition, this study proposes an innovative fare-based algorithm to search for the cheapest itineraries with non-linear fare structure, and an alternative fare model

is specially developed for Taipei transit system. Optimizing trip planning according to time or fare meets the common practices of passengers using transit system in a metropolitan area. For the basic problem where walking is not a means of transfer, a preprocessing algorithm is proposed to construct a specialized time-space network, so that the conventional shortest path algorithms can be directly applied to generate the itinerary of minimum travel time. Also, Wang has introduced a hierarchical timetable-based network in order to solve the problem for the situation when the passengers are allowed to walk from one node to another. But at the end he has used Dijkstra's algorithm and topological ordering algorithm to obtain the quickest path (Wang, 2008).

2.2 Shortest Path Problem

The shortest path problem is the problem of finding a path between two vertices (or nodes) in a graph such that the sum of the weights of its constituent edges is minimized. The weights could be cost, distance, travel time, social economic values, etc. (Adhikari, 2014).

Shortest path algorithms are currently used widely. They are the basis of some problems such as network flow problems, tree problems and other related problems. They decide the minimum cost of travel of the problems production cycle, the shortest path in an electric circuit or the most reliable way (Alija, 2015).

Mathematically, the public transportation network may be expressed as follows: let G = (N, L) be a multimodal public transport network, where N is the set of n nodes and L is the set of m directed links, connecting node i to node j in the network, denoted as L (i, j). Each distinct node (N) represents a specific stop and is assigned a stop ID. Each link L (i, j) has a weight associated with it, which can represent either time or fare required to travel from node i to j. In Figure 2. 1 Mathematical model of graphs , graph G is denoted by (N, L) where N = {1, 2, 3, 4, 5} and L = {(1, 2), (1, 4), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5)}. Figure 2. 1 Mathematical model of graphs shows the mathematical model of graphs.

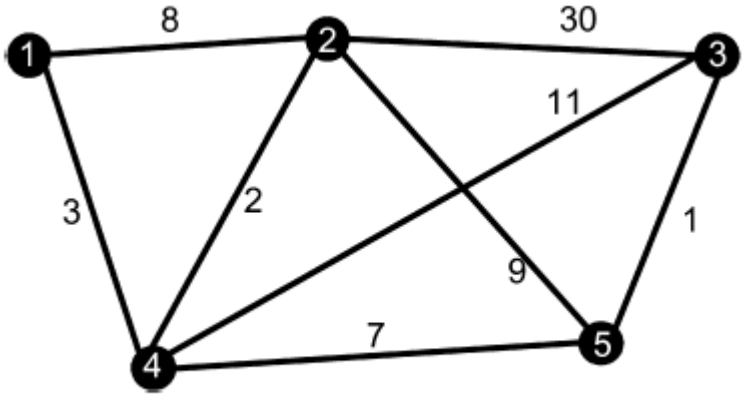


Figure 2. 1 Mathematical model of graphs

For a web-based public transport information system, the difficulty of finding the optimal routes comes from the size and complexity of the network in a modern city as well as the requirement to include user's preferences in the search. Bus stop spacing are planned differently according to the city configuration, people's commuting culture and activity-generated demands (Ibeas et al., 2010; Medina et al., 2013; Ceder et al., 2015), thus limiting the generality of a path finding algorithm. Public transport journey planning is a multi-objective decision making process (Lilian S.C. Pun-Cheng, 2015).

Routing applications whether they arise in transportation area or other domains such as communication networks refer for solving Shortest Path Problems (SPPs). While solving some routing problems can be done in a straightforward manner, computing shortest paths under certain circumstances is not always an easy task. For instance, solving the one-to-one SPP in static networks can be easily accomplished by applying the well-known algorithm of Dijkstra. On the other side, computing multicriteria shortest paths appears to be more difficult especially in large-scale dynamic networks (O. Dib, 2016). Many researches have been conducted to develop the optimal path finding mechanism based on the graph theory (Peng and Huang, 2000; Li and Kurt, 2000). Notably among them is Dijkstra's Algorithm (Dijkstra, 1959). It is a label setting algorithm to figure out the optimal path from a single node to all the others node in a network. Basically, it compares all nodes connected from a specified origin to a destination with its cost of travel (such as distance, time or fare) until the shortest path has been discovered (Lilian S.C. Pun-Cheng, 2015).

2.3 Dijkstra's Algorithm

Shortest path algorithms are central to most network and transportation problems. In Zhan and Noon's research paper, the application of fifteen different algorithms on real road networks indicated that it is worthwhile to apply Dijkstra's algorithm to solve one-to-one or one-to-some shortest path problems. Dijkstra's algorithm, the classical single-source shortest path algorithm, is a greedy algorithm that solves the single-source shortest path problem for a weighted, directed graph where all edge weights are non-negative (Zhan, 1998).

Dijkstra's algorithm is an algorithm named after its developer, Edsger W. Dijkstra, a Dutch computer scientist from Netherlands in 1959. Dijkstra's Algorithm is a chart search algorithm that unravels the single-source shortest path delinquent for a plot with non-negative edge path costs, producing a shortest path tree (Alija, 2015).

Dijkstra's algorithm or variations of it are the most commonly used route finding algorithm for solving the shortest path (Sadeghi-Niaraki, 2011). Dijkstra's algorithm is sometimes called the single-source shortest path because it solves the single-source shortest-path difficulty on a subjective, directed graph (G = V, E) where
V is a set whose elements is called vertices (nodes, junctions, or intersections) and E is a set of ordered pairs of vertices entitled directed edges (arcs or road segments). To find a shortest path from a source 's' vertex or location to a destination location 'd', Dijkstra's algorithm maintains a set S of vertices whose final shortest-path weights from the sources that already been determined. Knowing that "w' is the edge weight, the edge is an ordered pair (u, v) and assuming w (u, v) $\geq$ 0 for each edge (u, v) ϵ E, the algorithm recurrently chooses the vertex u ϵ V – S with the least short-path approximation, adds u to S, and relaxes all edges leaving u (Puthuparampil, 2007).
The Dijkstra algorithm enlarges the node that is at the extreme from the initial node, so it finishes up "stumbling" into the goal node. Just like the breadth-first search, it is certain to find the shortest path (Figure 2. 2 Dijkstra's Search) (Cormen, 2001).

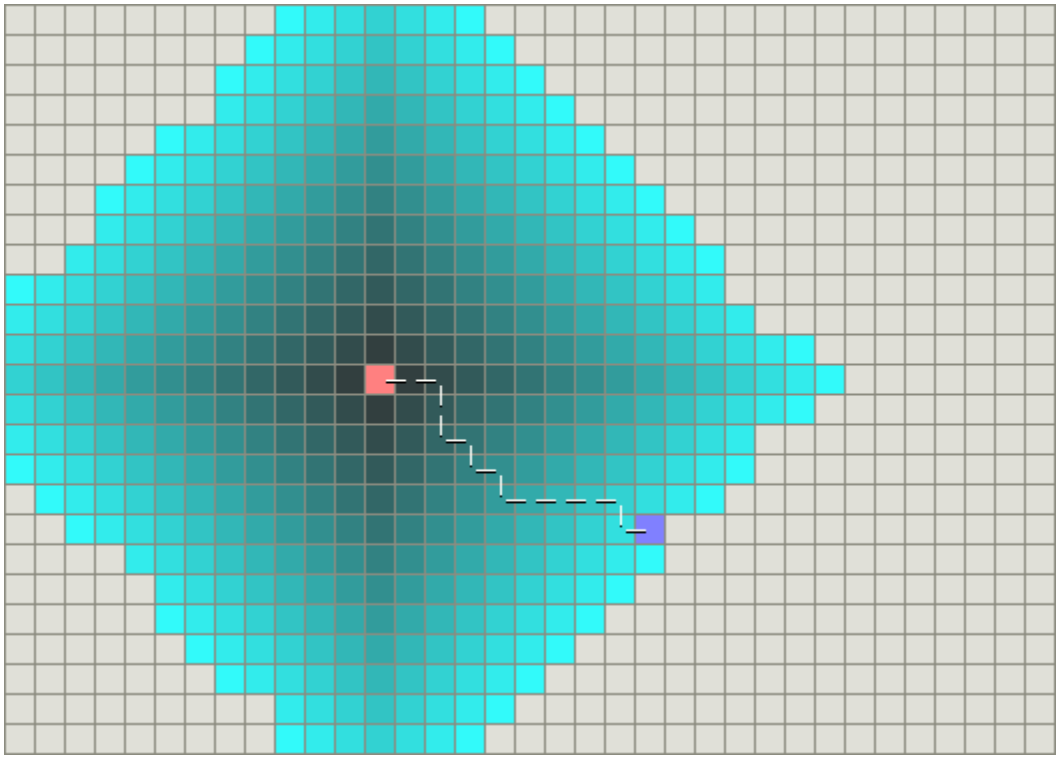


Figure 2. 2 Dijkstra's Search

(Alija, 2015)

From the figure (Figure 2. 3 First step of Dijkstra's Algorithm) shown, the objective is to find the shortest paths from origin to all other nodes. Dijkstra's will assign zero to initial node and node A while assigning infinity to all other nodes that are not visited. It will then assign a value gradually to get smallest value up to the destination, which is node E.

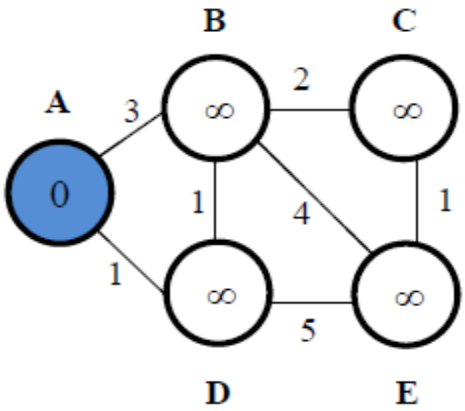


Figure 2. 3 First step of Dijkstra's Algorithm

(Alija, 2015)

Step 1: Node A is set to become current node. Zero is assigned to node A and infinity to all other nodes.

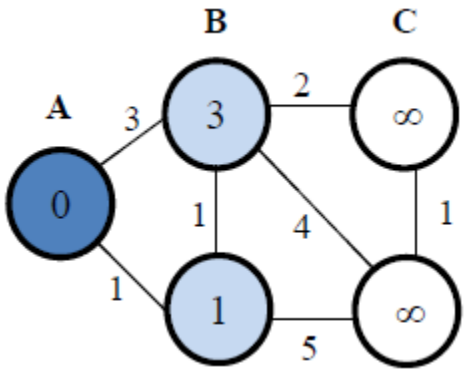


Figure 2. 4 Second step of Dijkstra's Algorithm

(Alija, 2015)

Step 2: Consider all unvisited neighbors and tentative distance will be calculated. Previously recorded value will be replaced since new value less than infinity.

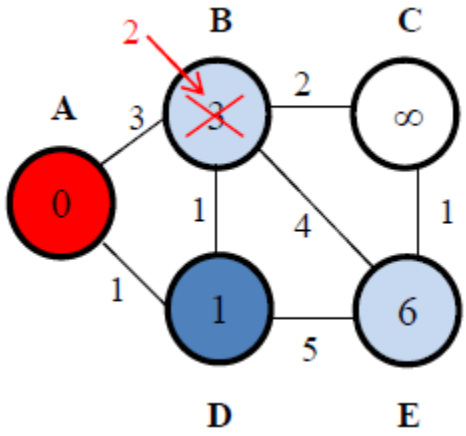


Figure 2. 5 Third step of Dijkstra's Algorithm

(Alija, 2015)

Step 3: Since all neighbors of node A have been taken into account, it is struck as visited and will not be tested again.

The next least distance from node A, node D now will be marked as current node. Its neighboring nodes will be updated with the new minimal distance value.

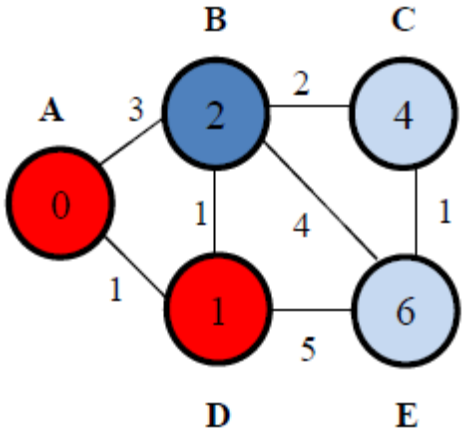


Figure 2. 6 Step 4 of Dijkstra's Algorithm

(Alija, 2015)

Step 4: Since all neighbors of node D have been taken into account, it is marked as visited and will not be checked over.

The next minimal distance from node D, node B will now be marked as current node. Its neighboring nodes will be updated with the new minimal distance value.

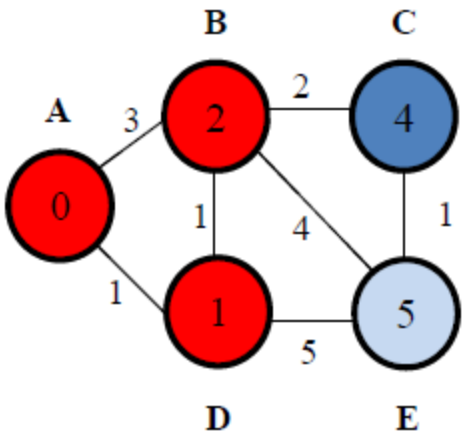


Figure 2. 7 Fifth step of Dijkstra's Algorithm

(Alija, 2015)

Step 5: Since all neighbors of node B have been accounted for, it is marked as visited and will not be tested over.

The next available minimal space from node B, which is node C now will be taken as present node. Its neighboring nodes will be updated with a fresh minimal distance value.

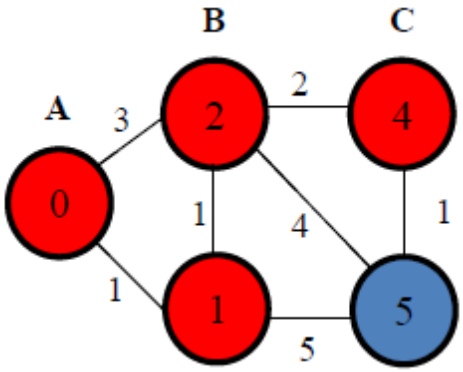


Figure 2. 8 Step 6 of Dijkstra's Algorithm

(Alija, 2015)

Step 6: Meanwhile, all neighbours of node C have been taken into account, it is marked as visited and will not be checked.

The next shortest distance from node C is node E, which will be chosen as current node. Since all the nodes have been visited, the shortest route from node A to node E is found.

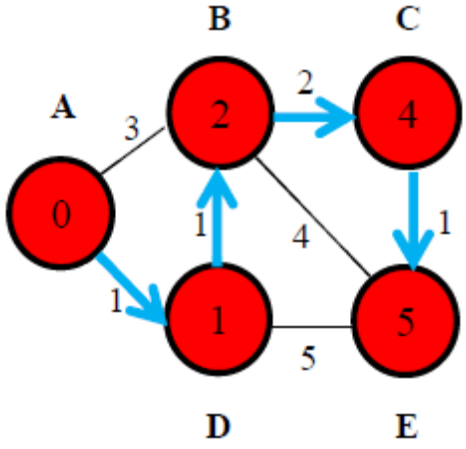


Figure 2. 9 Shortest path by Dijkstra's Algorithm

The shortest distance from Node A to Node E is:

A → D → B → C → E

(Alija, 2015)

CHAPTER 3. METHODOLOGY AND DATA ANALYSIS

3.1 Study Area

Pokhara city is selected for the case study of this research. Pokhara is a metropolitan area and the largest city of Nepal in terms of area. It is the provincial capital of Province number 4, Gandaki Pradesh according to recent federal restructuring of Nepal and headquarter of Kaski District. It is the center of the western part of Nepal with 1,05,630 households and 4,02,995 populations (CBS, 2011). The elevation ranges from 505 m (Kotre) to 2650 m (Armala) above sea level. The total area of Pokhara Lekhnath Metropolitan City (PLMC) is 464.24 square kilometers (PLMC, 2017) which represents 23.01 percent area of the Kaski district and 0.31 percent area of the country. The average temperature in PLMC ranges from 7^{o}c minimum to 31^{o}c with an annual rainfall of 3800 mm. Pokhara Lekhnath lies within mid hill of western Nepal. This city is situated on the lap of Annapurna Himalayan Range. It is surrounded by green hills. There are 9 lakes, 10 caves, Seti river gorges, several holy places, aesthetic viewpoints which are playing vital role to make city gregarious. It is divided into 33 wards (Figure 3. 1 Location Map of Study Area). PLMC is also known as a multi-cultural city (R.P Pokharel, 2018).

Since, Pokhara is in the rapid verge of immergence so the city is developing rapidly. Pokhara has annual population growth rate of 5.6%. In 2013, 45% population of Pokhara used private vehicles (Shakya, 2013). To achieve the anticipated social and economic development of a city, transportation system plays a decent role. Public transportation is a key component of the nation's economy and quality of life. It helps to bring a better quality of life by reducing traffic congestion, reducing pollution, and offering citizens a way to travel within an urban area. Public transportation provides people with mobility, freedom, and accessibility to job opportunities, education, health care, and community resources.

Not to forget, Pokhara is known for its beauty. It is also recognized as one of the major destinations for adventure tourism in Nepal. The visitors will certainly find such new places worth visiting. Such new places will further whet up the tourists' interest in more exotic places around Pokhara. The number of tourists visiting PLMC has been

increasing from 2,30,000 in 2010 to 3,01, 200 tourists in 2016 (DoT, 2016). Similarly, the growth of domestic tourists visiting PLMC is also in increasing trend. This indicates that there will be more number of passengers (locals and visitors) whose specific interest to visit specific place has to be responded by the public transportation. So, the tourists visiting Pokhara definitely want their time and money to be used efficiently. Therefore, this study will also be fruitful to those public transport users who focuses on the minimum travel time and cost.

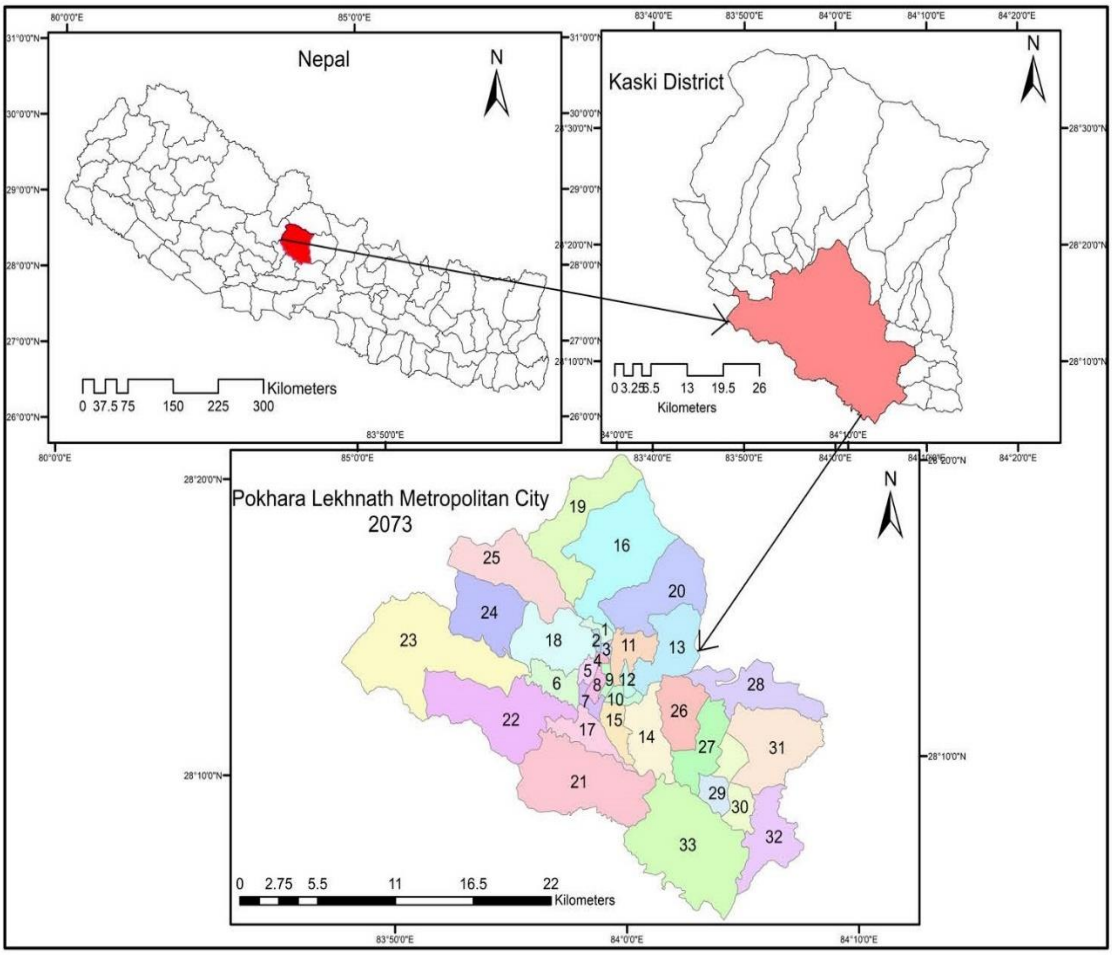


Figure 3. 1 Location Map of Study Area

(R.P Pokharel, 2018)

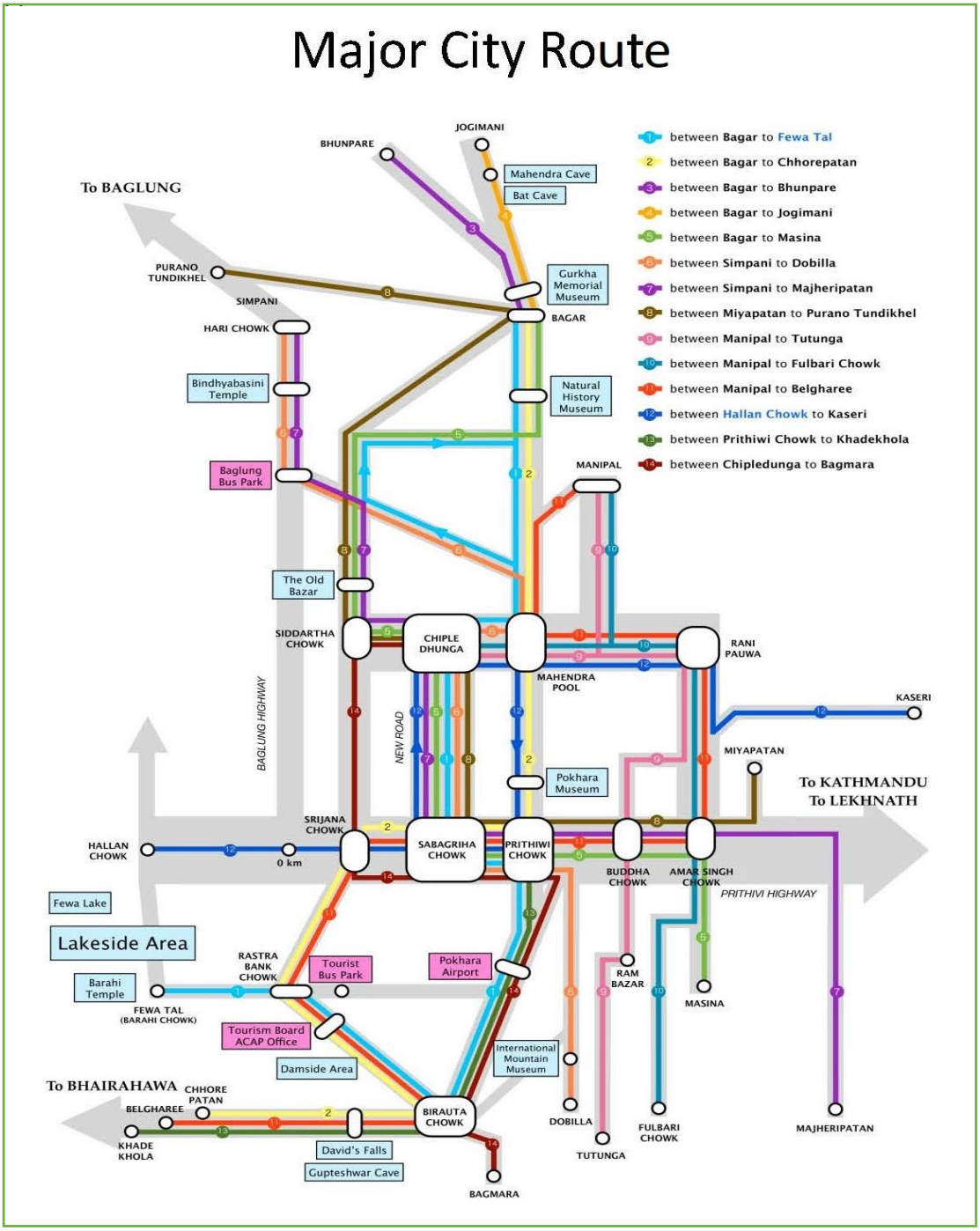


Figure 3. 2 Major Public Transportation route of Pokhara

(Paudel)

3.2 Data Collection

This research is based on both primary as well as secondary data sources. For the purpose of analysis of optimal route computation, primary data regarding travel time has to be collected. Since, the travel time of any public transportation system (of Pokhara city) has not been noted till date. So, the time taken to travel from one node to another node was individually observed using stop watch and noted respectively.

Also, we need the vehicle fare required to travel from one stop to another stop. Since, the travel cost from one stop to other stops is different and managed scientifically as far as possible by the local bodies. The travel cost (fare) is determined by private bus service providers by close coordination with the municipality. There are currently four City Bus Service Providers in Pokhara city, namely:

1. Pokhara City Bus Service
2. Lekhnath City Bus Service
3. City Micro Bus Service
4. Prithvi Bus Service

And there are respective routes determined for the respective bus service providers in the city. So, to travel different parts of the city require all the four service providers' travel cost. Those data were obtained from the respective bus service providers.

Other necessary data regarding Pokhara city were obtained from the reports of Pokhara-Lekhnath Municipality (PLMC).

Households and other demographic data were obtained from the reports of Central Bureau of Statistics, Nepal (CBS).

The bus routes of respective bus service providers and collected data are summarized in **Annex 1**.

3.3 Methodology

To conduct this research, the methodology has been divided into two main stages. The following figure (Figure 3. 3 Methodological framework Figure 3. 3 Methodological framework

) gives an overview of the methodology followed in the present research. The stages of methodology are discussed and highlighted one by one.

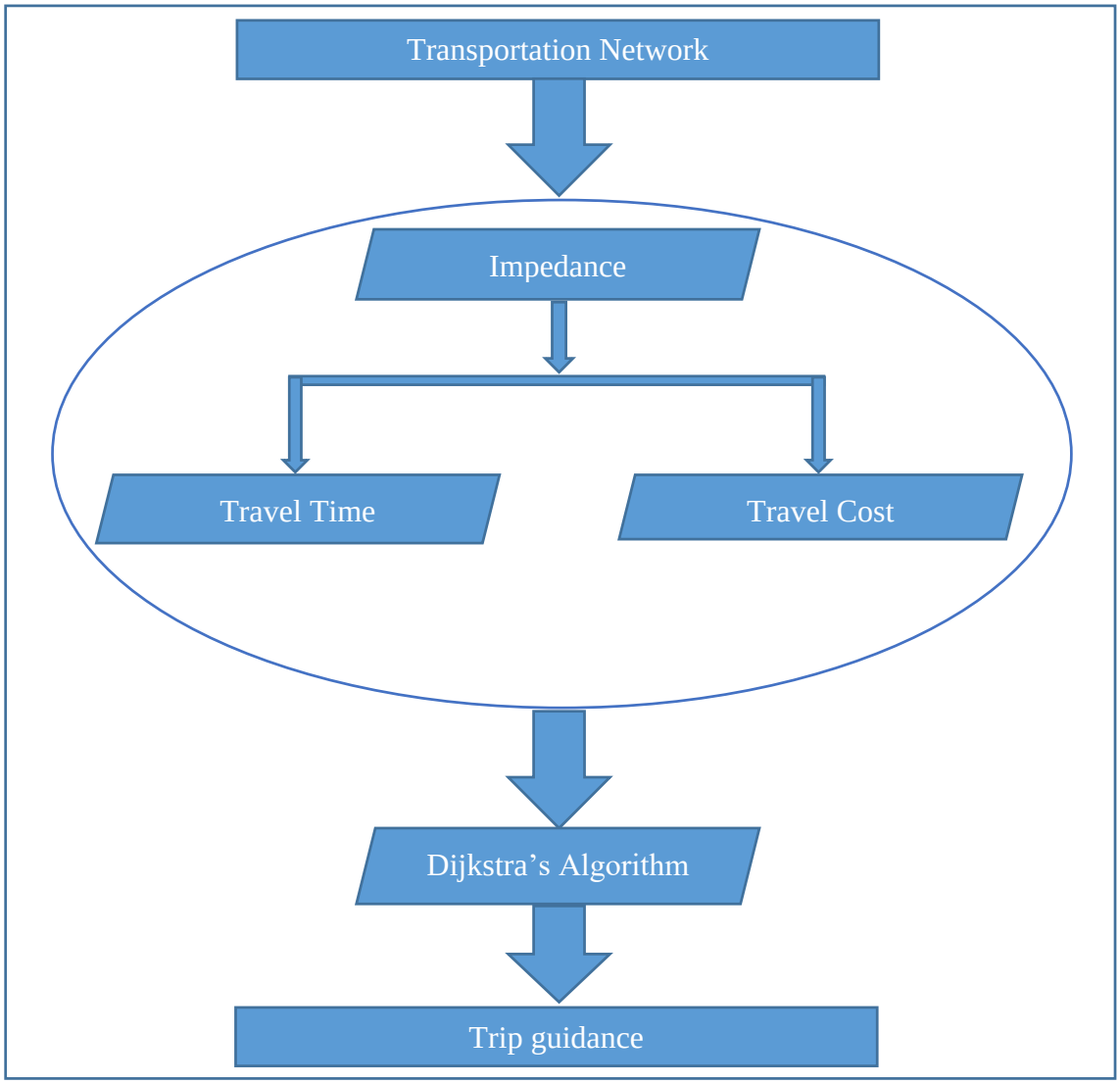


Figure 3. 3 Methodological framework

Stage One: Identifying the impedance of bus trip

First of all, the existing public transportation network of Pokhara was taken into consideration such that a node represented a bus stop/station and link between them represented a transport linkage. Since, the main variables of this research were travel time and travel cost. So, the field observation was carried out in order to note down the time taken to travel from each node to another node of each bus routes around Pokhara city with the help of stop watch. The field work for data collection was carried out from

December 2017 to January 2018. Similarly, the travel fare data was collected from the respective bus service providers.

Stages Two: Analyzing the data and find the optimal bus route

During the bus trips, the travelers may prefer shortest travel distance and time including minimum bus fare. This has led to two different strategies in searching optimal bus trips, one is minimum travel cost oriented and another is shortest travel time oriented. As soon as the raw data were collected, they need to be processed. So, the time and cost required to travel from each respective node to others were classified. Since, the travel cost required to travel from each respective node to another respective node is same and vice versa but in case of travel time the case is not the same. Thus, the travel time was calculated by taking average value among the respective connecting nodes for each different trips. Once the travel time and cost were identified, a geographical network was prepared such that bus stops touching each other formed a link. Square matrix for each node in which time and cost as it's element was prepared in MS Excel. A weighted graph was formed. The elements of matrix were imported to Microsoft Visual Studio 2017 as the input for each node. The bus trip optimization based on time and cost impedance was carried out with a looped procedure. With the help of Pseudocode for Dijkstra's Algorithm using C++ an optimal path in terms of minimum travel time and travel cost were achieved for each run. This solution was then matched with the existing network.

3.3.1 Printing Paths in Dijkstra's Shortest Path Algorithm

First of all, a SPT (shortest path tree) is generated with given source as root. Two sets are maintained such that one set contains vertices included in shortest path tree, other set includes vertices not yet included in shortest path tree. At every step of the algorithm, a vertex is found which is in the other set (set of not yet included) and has a minimum distance from the source. The use of this algorithm only finds the shortest distance but do not print the paths so some improvements has to be done in the pseudocode so that it displays the shortest distance along with printing path.

Below are the detailed steps used in Dijkstra's algorithm to find the shortest path from a single source vertex to all other vertices.

Algorithm Steps:

1) Create a set sptSet (shortest path tree set) that keeps track of vertices included in shortest path tree, i.e., whose minimum distance from source is calculated and finalized. Initially, this set is empty.

2) Assign a distance value to all vertices in the input graph. Initialize all distance values as INFINITE. Assign distance value as 0 for the source vertex so that it is picked first.

3) While sptSet doesn't include all vertices

a) Pick a vertex u which is not there in sptSet and has minimum distance value.

b) Include u to sptSet.

c) Update distance value of all adjacent vertices of u. To update the distance values, iterate through all adjacent vertices. For every adjacent vertex v, if sum of distance value of u (from source) and weight of edge u-v, is less than the distance value of v, then update the distance value of v.

The idea is to create a separate array parent[]. Value of parent[v] for a vertex v stores parent vertex of v in shortest path tree. Parent of root (or source vertex) is -1. Whenever we find shorter path through a vertex u, we make u as parent of current vertex. Once we have parent array constructed, we can now print the path using below recursive function. (GeeksforGeeks, n.d.)

```
void printPath(int parent[], int j)
{
    // Base Case : If j is source
    if (parent[j]==-1)
        return;

    printPath(parent, parent[j]);

    printf("%d ", j);
}
```

Figure 3. 4 Recursive function to print path

(GeeksforGeeks, n.d.)

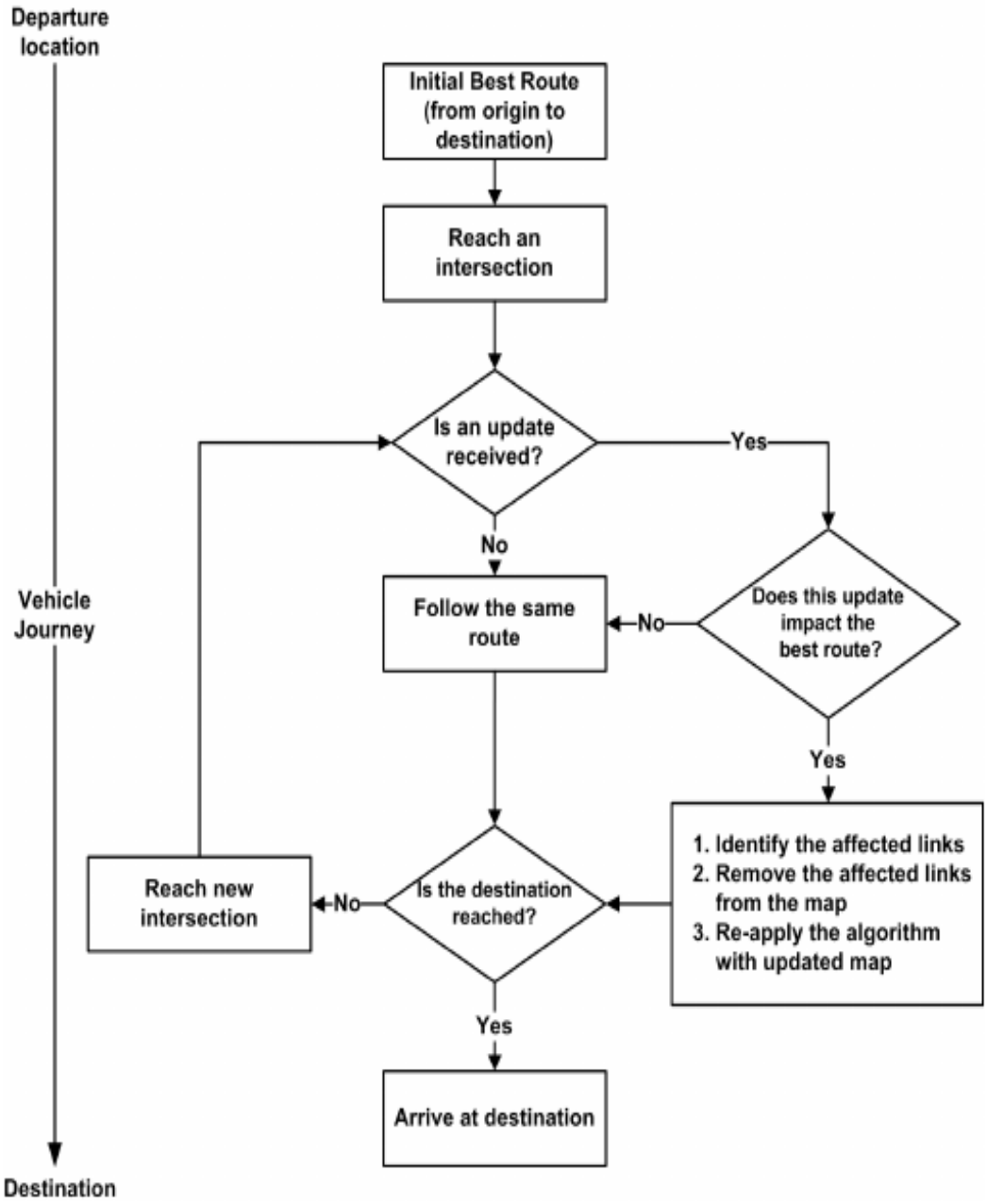


Figure 3. 5 Flowchart illustrating the best route during the vehicle's journey

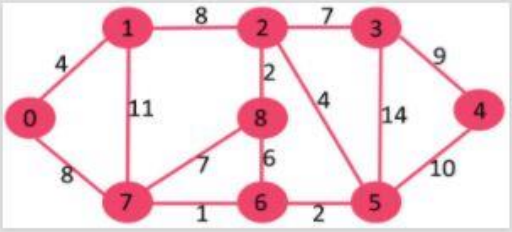
(Vi Ngoc-Nha Tran, 2012)

3.3.2 Numerical Example

A numerical example is used to demonstrate the effectiveness and implementation of the algorithm to be used for this study.

For example, consider below graph and starting point (origin/source) as 0.

For example, consider below graph and source as 0,



Output should be

Vertex	Distance	Path
0 -> 1	4	0 1
0 -> 2	12	0 1 2
0 -> 3	19	0 1 2 3
0 -> 4	21	0 7 6 5 4
0 -> 5	11	0 7 6 5
0 -> 6	9	0 7 6
0 -> 7	8	0 7
0 -> 8	14	0 1 2 8

Figure 3. 6 Example of Transport Network

(GeeksforGeeks, n.d.)

After running the pseudocode for Dijkstra's algorithm using C++, the following result was obtained.

Vertex	Distance	Path
0 -> 1	4	0 1
0 -> 2	12	0 1 2
0 -> 3	19	0 1 2 3
0 -> 4	21	0 7 6 5 4
0 -> 5	11	0 7 6 5
0 -> 6	9	0 7 6
0 -> 7	8	0 7
0 -> 8	14	0 1 2 8

3.4 Data Analysis

3.4.1 Ranking of Bus Stops

For the optimization of any road network, it should consist of ranking of nodes so that it will be easier to conduct and present the outcomes of research work. The bus stops (i.e. nodes) of Pokhara were ranked as follows:

Rank	Bus Stop	Rank	Bus Stop
0	Bagar	34	Titepani
1	Mahendrapool	35	Puranchaur
2	Chipedhunga	36	Raikar
3	Prithvi Chowk	37	Maidan
4	Birauta	38	Gufa
5	Lakeside	39	Riverdale School
6	Srijana Chowk	40	Jogimadi
7	Rastrabank Chowk	41	Tupke
8	Chorepatan	42	Ghattekunna
9	Hallanchowk	43	Chauthe
10	Zero KM	44	Bhotechautara
11	Sitaldevi	45	Majheripatan
12	Bajhapatan	46	Bagar Ward Office
13	Kahunkhola Chowk	47	Miyapatan
14	Sajha	48	Manipal
15	Kaseri	49	Ban Campus
16	Simpani	50	Tutunga
17	Harichowk	51	Belghari
18	Baglung Buspark	52	Bijaypur
19	Himalayan Eye Hospital	53	Buddhi Bazar
20	Int'l Mountain Museum	54	Arghaun Chowk
21	Mahatgauda	55	Tal Chowk
22	Amarsingh Chowk	56	Sisuwa
23	Rambazar	57	Begnas Tal
24	Kaji Pokhari	58	PU Chowk
25	Nirajan Chowk	59	Tamukh
26	Chinedanda	60	Satmuhane
27	Khalte Masina	61	Rupa Tal
28	Sundarfeed	62	Janata Ko Chautara
29	Engineering Campus	63	Phedi
30	Lamachaur	64	Rithepani
31	Akalaa Devi Temple	65	NEA Office, Malepatan
32	Bhunpare	66	WR Hospital
33	Besichowk	67	Saint Mary's School

Table 3. 1 Ranking of Bus Stops

3.4.2 Optimization of Transport Network Based on Minimum Travel Time

For the preparation of the network, bus stops touching each were considered as the nodes and the geographical connection between nodes (i.e. bus stops) were considered as their links. The weights of the link between each node were the time required to travel to the respective nodes. The optimization was carried out for the nodes touched by the buses of respective bus service providers and was dealt individually.

1. Pokhara City Bus Service:

Pokhara City Bus Service provides its service throughout the city and covers most of the parts of the city. It runs its bus service in 16 different routes with touching 52 nodes.

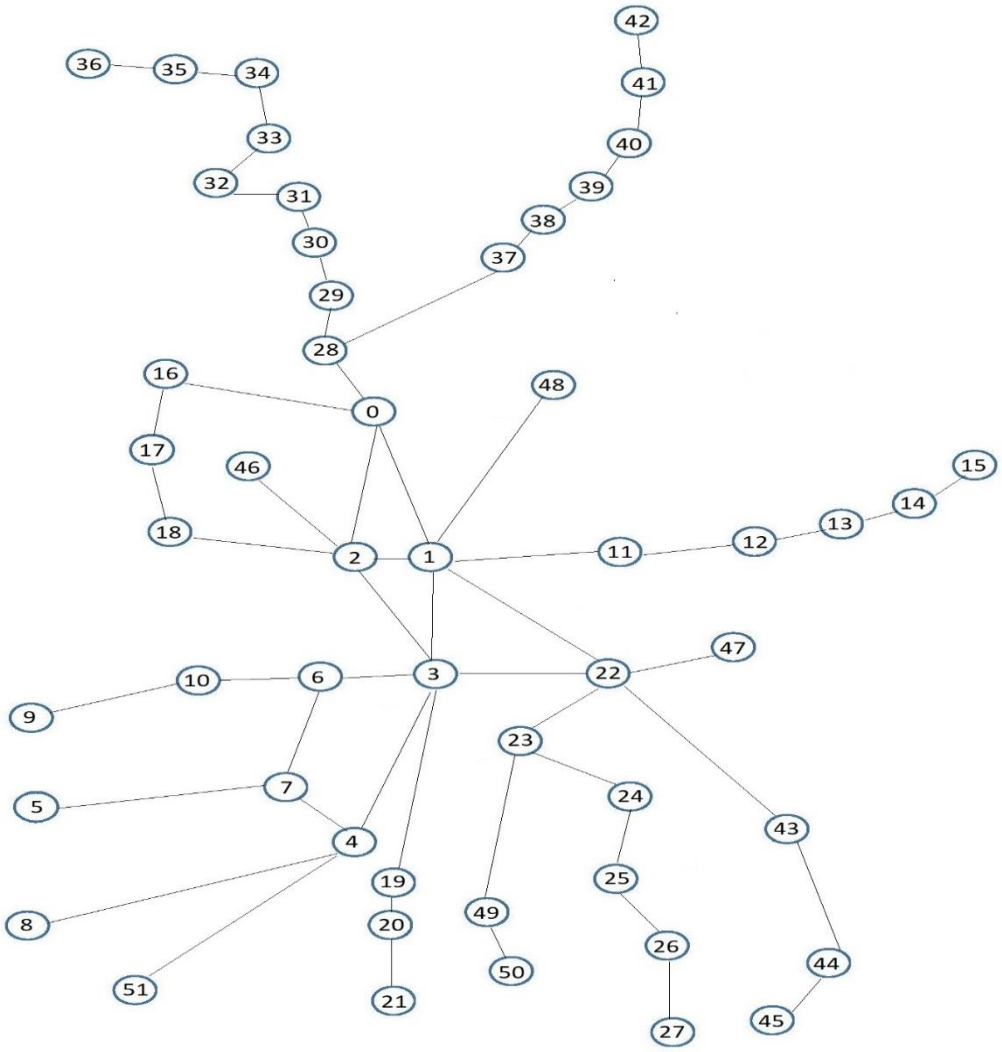


Figure 3. 7 Geographical Network showing the connection between different nodes touched by Pokhara City Bus Service

Table 3. 2 Network in mathematical terms

From		To					
0	Bagar	1	2	16	28		
1	Mahendrapool	0	2	3	11	22	48
2	Chipedhunga	0	1	3	18	46	
3	Prithvi Chowk	1	2	4	6	19	22
4	Birauta	3	7	8	51		
5	Lakeside	7					
6	Srijana Chowk	3	7	10			
7	Rastrabank Chowk	4	5	6			
8	Chorepatan	4					
9	Hallanchowk	10					
10	Zero KM	6	9				
11	Sitaldevi	1	12				
12	Bajhapatan	11	13				
13	Kahunkhola Chowk	12	14				
14	Sajha	13	15				
15	Kaseri	14					
16	Simpani	17					
17	Harichowk	16	18				
18	Baglung Buspark	2	17				
19	Himalayan Eye Hospital	3	20				
20	Int'l Mountain Museum	19	21				
21	Mahatgauda	20					
22	Amarsingh Chowk	1	3	23	43	47	
23	Rambazar	22	24	49			
24	Kaji Pokhari	23	25				
25	Nirajan Chowk	24	26				
26	Chinedanda	25	27				
27	Khalte Masina	26					
28	Sundarfeed	0	29	37			
29	Engineering Campus	28	30				
30	Lamachaur	29	31				
31	Akalaa Devi Temple	30	32				
32	Bhunpare	31	33				
33	Besichowk	32	34				
34	Titepani	33	35				
35	Puranchaur	34	36				
36	Raikar	35					
37	Maidan	28	38				
38	Gufa	37	39				
39	Riverdale School	38	40				
40	Jogimadi	39	41				
41	Tupke	40	42				

42	Ghattekunna	41					
43	Chauthe	22	44				
44	Bhotechautara	43	45				
45	Majheripatan	44					
46	Bagar Ward Office	2					
47	Miyapatan	22					
48	Manipal	1					
49	Ban Campus	23	50				
50	Tutunga	49					
51	Belghari	4					

Weights of link:

Travel time is the weight between the respective nodes which were calculated by taking the average values among the respective connecting nodes and are shown in the Table 3. 3 Network with link weights and destination nodes below:

Table 3. 3 Network with link weights and destination nodes

From		To					
0	Bagar	9.43 **1**	15.98 **2**	4.38 **16**	4.54 **28**		
1	Mahendrapool	9.43 **0**	1.57 **2**	15.73 **3**	8 **11**	13.4 **22**	11.11 **48**
2	Chipedhunga	15.98 **0**	1.57 **1**	7.38 **3**	9.17 **18**	14.83 **46**	
3	Prithvi Chowk	15.73 **1**	7.38 **2**	15.41 **4**	3.32 **6**	21.61 **19**	10.56 **22**
4	Birauta	15.41 **3**	5 **7**	6.26 **8**	10.32 **51**		
5	Lakeside	6.04 **7**					
6	Srijana Chowk	3.32 **3**	4.76 **7**	2.25 **10**			
7	Rastrabank Chowk	5 **4**	6.04 **5**	4.76 **6**			
8	Chorepatan	6.26 **4**					
9	Hallanchowk	5.69 **10**					
10	Zero KM	2.25 **6**	5.69 **9**				
11	Sitaldevi	8 **1**	7.54 **12**				

12	Bajhapatan	7.54 **11**	6 **13**				
13	Kahunkhola Chowk	6 **12**	7.14 **14**				
14	Sajha	7.14 **13**	5.88 **15**				
15	Kaseri	5.88 **14**					
16	Simpani	2.62 **17**					
17	Harichowk	2.62 **16**	10 **18**				
18	Baglung Buspark	9.17 **2**	10 **17**				
19	Himalayan Eye Hospital	21.61 **3**	3.87 **20**				
20	Int'l Mountain Museum	3.87 **19**	4.06 **21**				
21	Mahatgauda	4.06 **20**					
22	Amarsingh Chowk	13.4 **1**	10.56 **3**	7.09 **23**	7.96 **43**	13.03 **47**	
23	Rambazar	7.09 **22**	6.54 **24**	7.29 **49**			
24	Kaji Pokhari	6.54 **23**	6.17 **25**				
25	Nirajan Chowk	6.17 **24**	5.27 **26**				
26	Chinedanda	5.27 **25**	13.26 **27**				
27	Khalte Masina	13.26 **26**					
28	Sundarfeed	4.54 **0**	3.75 **29**	9.78 **37**			
29	Engineering Campus	3.75 **28**	3.37 **30**				
30	Lamachaur	3.37 **29**	9.72 **31**				
31	Akalaa Devi Temple	9.72 **30**	3.14 **32**				
32	Bhunpare	3.14 **31**	3.28 **33**				
33	Besichowk	3.28 **32**	2.84 **34**				

34	Titepani	2.84 **33**	7.04 **35**				
35	Puranchaur	7.04 **34**	6.51 **36**				
36	Raikar	6.51 **35**					
37	Maidan	9.78 **28**	4.23 **38**				
38	Gufa	4.23 **37**	9.8 **39**				
39	Riverdale School	9.8 **38**	8 **40**				
40	Jogimadi	8 **39**	5.3 **41**				
41	Tupke	5.3 **40**	4.4 **42**				
42	Ghattekunna	4.4 **41**					
43	Chauthe	7.96 **22**	3.22 **44**				
44	Bhotechautara	3.22 **43**	5.66 **45**				
45	Majheripatan	5.66 **44**					
46	Bagar Ward Office	14.83 **2**					
47	Miyapatan	13.03 **22**					
48	Manipal	11.11 **1**					
49	Ban Campus	7.29 **23**	12.47 **50**				
50	Tutunga	12.47 **49**					
51	Belghari	10.32 **4**					

The weighted graph was constructed with the help of the data from Table 3. 3 Network with link weights and destination nodes.

The weighted graph is shown in Figure 3. 8 Weighted graph between the nodes

.

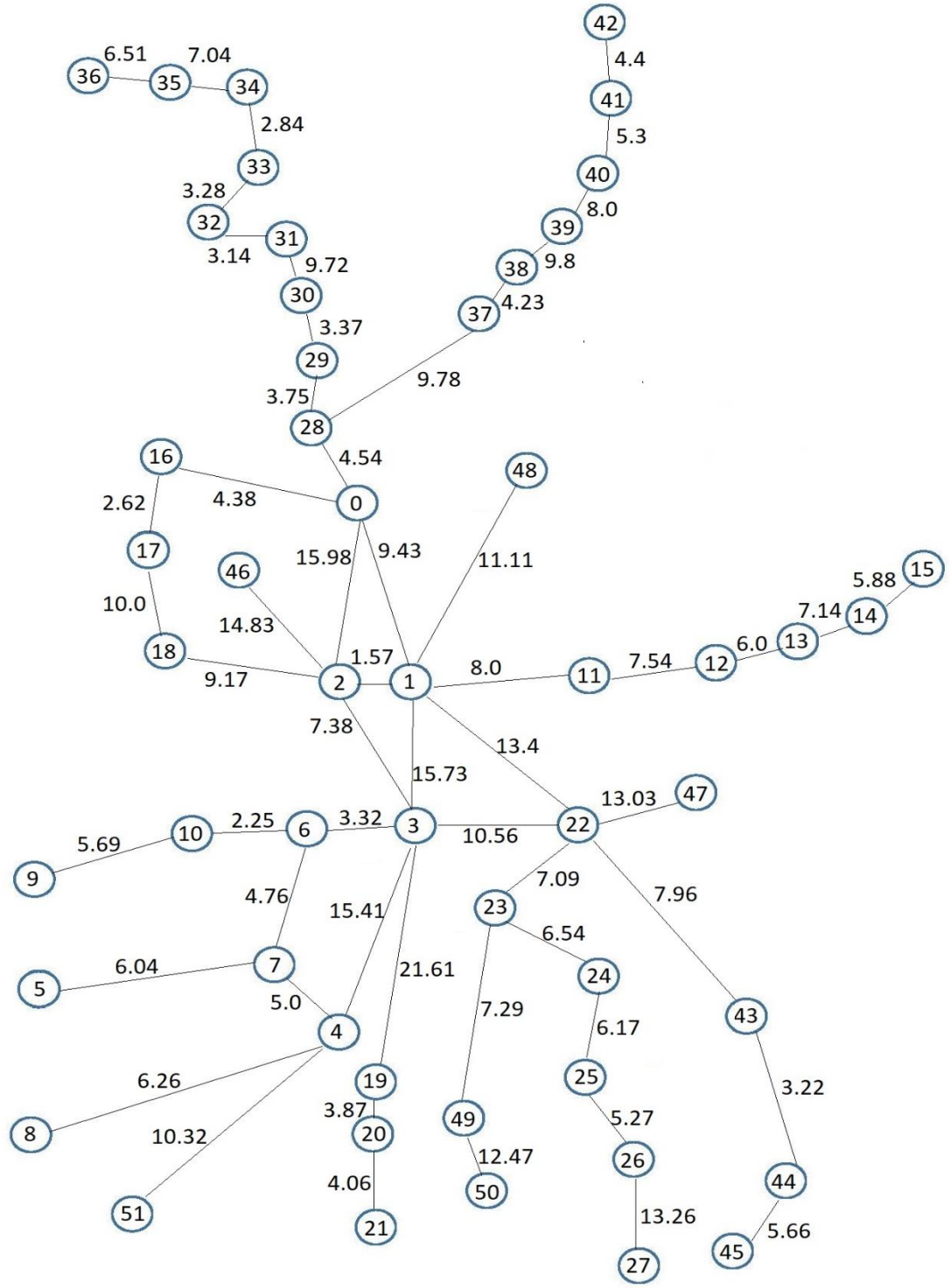


Figure 3. 8 Weighted graph between the nodes

Since, the commuter can start their journey from any node/bus stop so the matrix for each node to be considered as starting point was formed.

Table 3. 4 Network with link weights and destination nodes in matrix form

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0.00	9.43	15.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.38	0.00	0.00	0.00
1	9.43	0.00	1.57	15.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	15.98	1.57	0.00	7.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.17	0.00
3	0.00	15.73	7.38	0.00	15.41	0.00	3.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21.61
4	0.00	0.00	0.00	15.41	0.00	0.00	0.00	5.00	6.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	3.32	0.00	0.00	0.00	4.76	0.00	0.00	2.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	5.00	6.04	4.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	6.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	2.25	0.00	0.00	5.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.54	0.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.00	0.00	7.14	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.14	0.00	5.88	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.88	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.62	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	13.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	14.83	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.32
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.62	0.00	10.00	0.00
18	0.00	0.00	9.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00
19	0.00	0.00	0.00	21.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.87
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	13.40	0.00	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	4.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	4.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	4.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	7.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	7.09	0.00	6.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	6.54	0.00	6.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	6.17	0.00	5.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	5.27	0.00	13.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	13.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.78	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.75	0.00	3.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.37	0.00	9.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.72	0.00	3.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.14	0.00	3.28	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.28	0.00	2.84	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	7.96	0.00	0.00	0.00	13.03	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.29	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	14.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	11.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	10.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.84	0.00	7.04	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.04	0.00	6.51	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.51	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.23	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.23	0.00	9.80
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.80	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	7.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	13.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	7.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	5.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	5.30	0.00	4.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	4.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	3.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	3.22	0.00	5.66	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	5.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.47	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.47	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Once the matrix is formed, it was then optimized using Dijkstra's Algorithm in Microsoft Visual Studio 2017 (C++ code) which is incorporated in the **Annex 2** and result is shown in **Chapter 4**.

2. Lekhnath City Bus Service:

Lekhnath City Bus Service provides its service focusing on major parts of Lekhnath and connects to Pokhara. It runs its bus service in 5 different routes with touching 19 nodes.

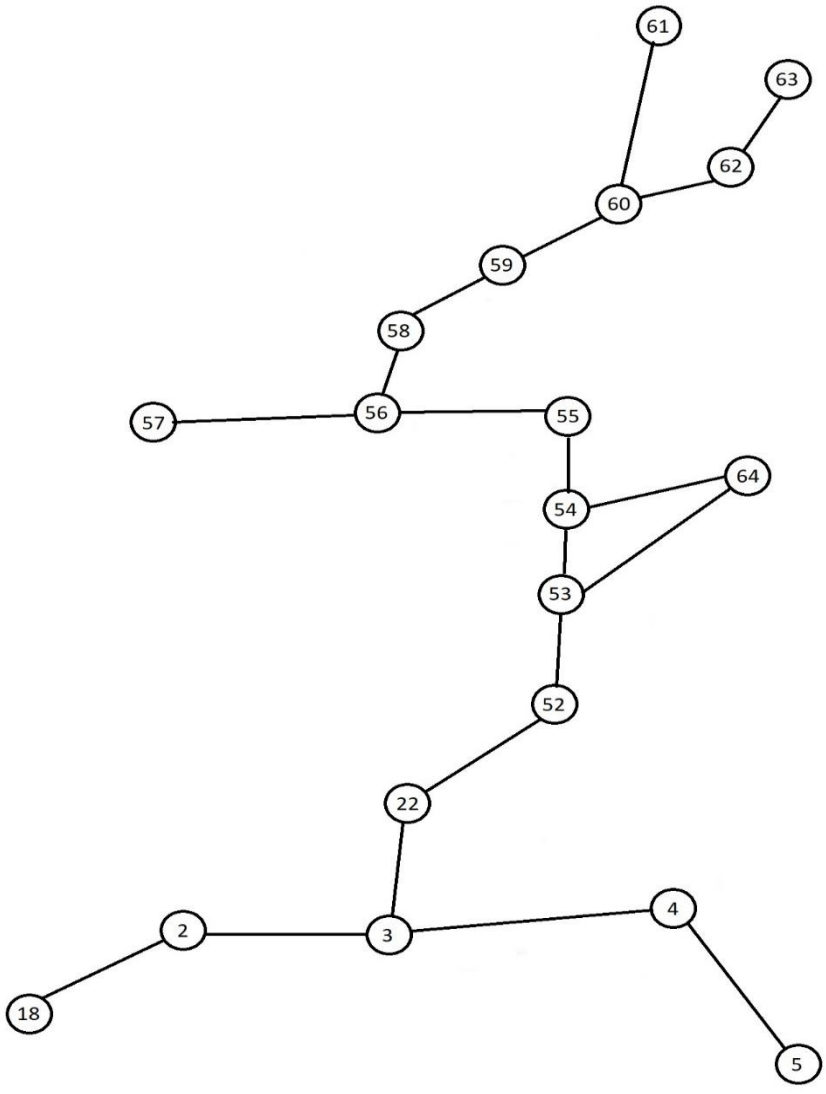


Figure 3. 9 Geographical Network showing the connection between different nodes touched by Lekhnath City Bus Service

Table 3. 5 Network in mathematical terms

From		To		
2	Chipedhunga	3	18	
3	Prithvi Chowk	2	4	22
4	Birauta	3	5	
5	Lakeside	4		
18	Baglung Buspark	2		
22	Amarsingh Chowk	3	52	
52	Bijaypur	22	53	
53	Buddhi Bazar	52	54	64
54	Arghaun Chowk	53	55	64
55	Tal Chowk	54	56	
56	Sisuwa	55	57	58
57	Begnas Tal	56		
58	PU Chowk	56	59	
59	Tamukh	58	60	
60	Satmuhane	59	61	62
61	Rupa Tal	60		
62	Janata Ko Chautara	60	63	
63	Phedi	62		
64	Rithepani	53	54	

Weights of link:

Travel time is the weight between the respective nodes which were calculated by taking the average values among the respective connecting nodes and are shown in the Table 3. 6 Network with link weights and destination nodes below:

Table 3. 6 Network with link weights and destination nodes

From		To		
2	Chipedhunga	9.01 **3**	4.86 **18**	
3	Prithvi Chowk	9.01 **2**	7.54 **4**	9.55 **22**
4	Birauta	7.54 **3**	9.48 **5**	
5	Lakeside	9.48 **4**		
18	Baglung Buspark	4.86 **2**		
22	Amarsingh Chowk	9.55 **3**	11.24 **52**	
52	Bijaypur	11.24 **22**	4.73 **53**	
53	Buddhi Bazar	4.73 **52**	4.74 **54**	15.4 **64**
54	Arghaun Chowk	4.74 **53**	5 **55**	8.75 **64**
55	Tal Chowk	5 **54**	4.85 **56**	
56	Sisuwa	4.85 **55**	4.89 **57**	7.53 **58**
57	Begnas Tal	4.89 **56**		
58	PU Chowk	7.53 **56**	4.55 **59**	
59	Tamukh	4.55 **58**	4.5 **60**	
60	Satmuhane	4.5 **59**	6.34 **61**	5.27 **62**
61	Rupa Tal	6.34 **60**		
62	Janata Ko Chautara	5.27 **60**	14.58 **63**	
63	Phedi	14.58 **62**		
64	Rithepani	15.4 **53**	8.75 **54**	

The weighted graph was constructed with the help of the data from Table 3. 6 Network with link weights and destination nodes.

The weighted graph is shown in Figure 3. 10 Weighted graph between the nodes

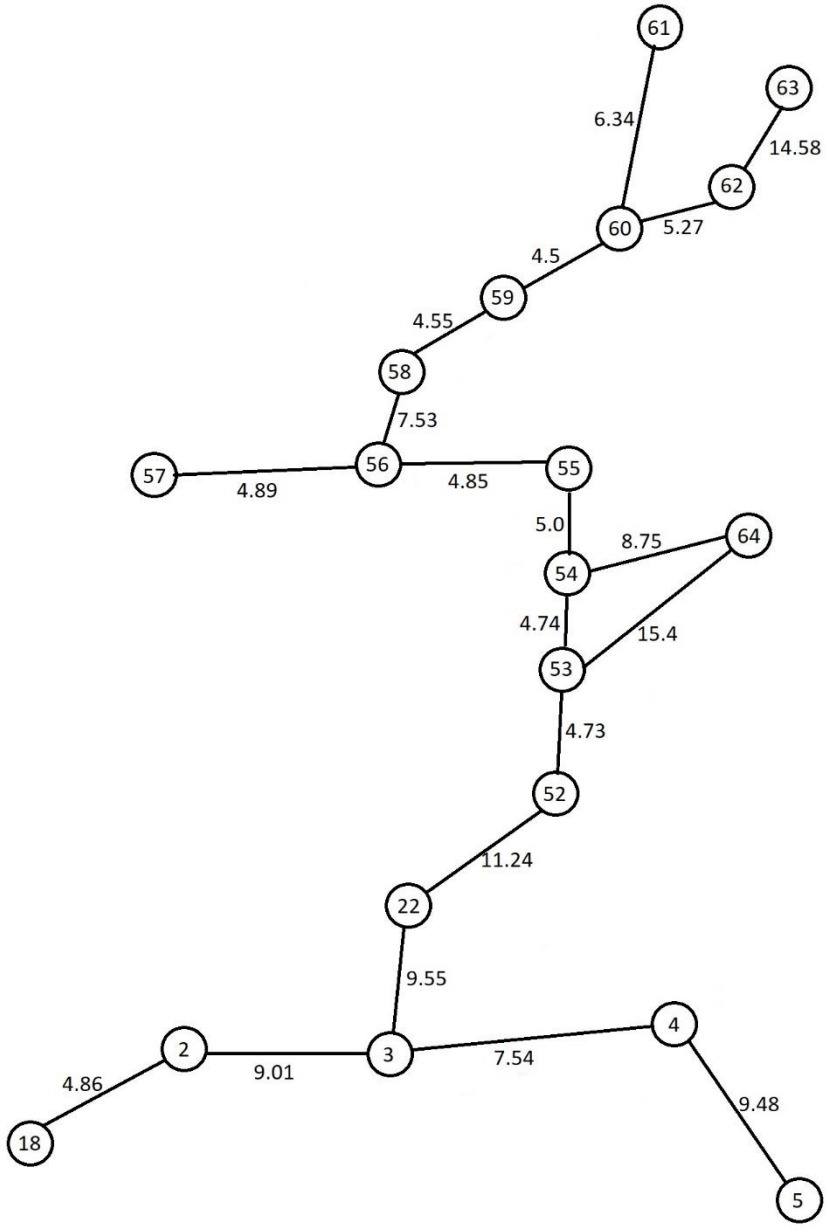


Figure 3. 10 Weighted graph between the nodes

Since, the commuter can start their journey from any node/bus stop so the matrix for each node to be considered as starting point was formed.

Table 3. 7 Network with link weights and destination nodes in matrix form

	2	3	4	5	18	22	52	53	54	55	56	57	58	59	60	61	62	63	64
2	0.00	9.01	0.00	0.00	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	9.01	0.00	7.54	0.00	0.00	9.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	7.54	0.00	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	9.55	0.00	0.00	0.00	0.00	11.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00	0.00	11.24	0.00	4.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
53	0.00	0.00	0.00	0.00	0.00	0.00	4.73	0.00	4.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.74	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.75
55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	4.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.85	0.00	4.89	7.53	0.00	0.00	0.00	0.00	0.00	0.00
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.53	0.00	0.00	4.55	0.00	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.55	0.00	4.50	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.50	0.00	6.34	5.27	0.00	0.00
61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.34	0.00	0.00	0.00	0.00
62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.27	0.00	0.00	14.58	0.00
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.58	0.00	0.00
64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40	8.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Once the matrix is formed, it was then optimized using Dijkstra's Algorithm in Microsoft Visual Studio 2017 (C++ code) and result obtained are incorporated in **Chapter 4**.

3. City Micro Bus Service:

City Micro Bus Service provides its service on major parts of Pokhara. It runs its bus service in 6 different routes with touching 18 nodes.

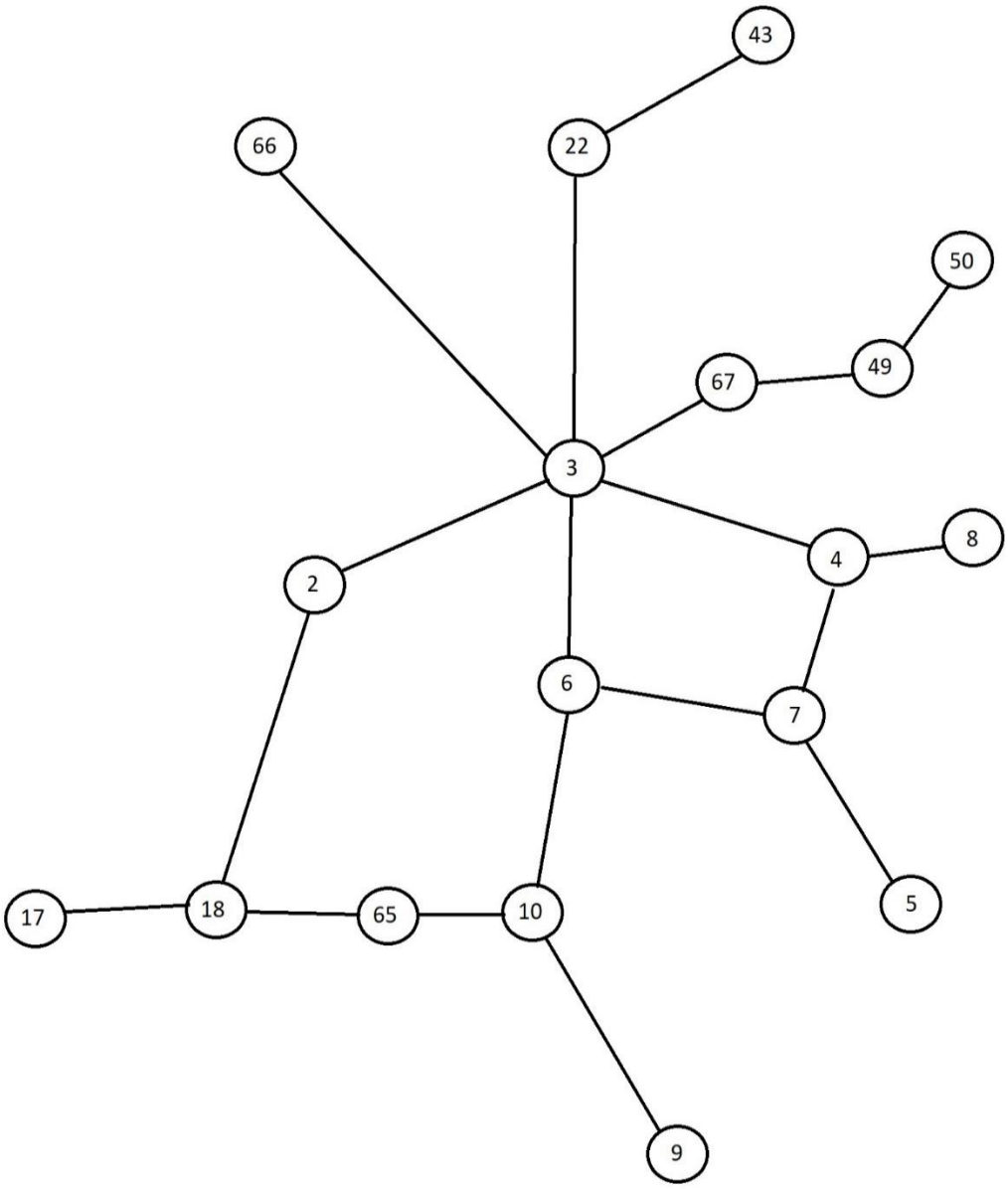


Figure 3. 11 Geographical Network showing the connection between different nodes touched by City Micro Bus Service

Table 3. 8 Network in mathematical terms

From		To					
2	Chipedhunga	3	18				
3	Prithvi Chowk	2	4	6	22	66	67
4	Birauta	3	7	8			
5	Lakeside	7					
6	Srijana Chowk	3	7	10			
7	Rastrabank Chowk	4	5	6			
8	Chorepatan	4					
9	Hallan Chowk	10					
10	Zero KM	6	9	65			
17	Hari Chowk	18					
18	Baglung Buspark	2	17	65			
22	Amarsingh Chowk	3	43				
43	Chauthe	22					
49	Ban Campus	50	67				
50	Tutunga	49					
65	NEA Office, Malepatan	10	18				
66	WR Hospital	3					
67	Saint Mary's School	3	49				

Weights of link:

Travel time is the weight between the respective nodes which were calculated by taking the average values among the respective connecting nodes and are shown in the Table 3. 9 Network with link weights and destination nodes below:

Table 3. 9 Network with link weights and destination nodes

From		To					
2	Chipedhunga	10.04 **3**	6.93 **18**				
3	Prithvi Chowk	10.04 **2**	13.06 **4**	4.22 **6**	11.77 **22**	13.05 **66**	18.54 **67**
4	Birauta	13.06 **3**	4.6 **7**	4.87 **8**			
5	Lakeside	6.47 **7**					
6	Srijana Chowk	4.22 **3**	5.46 **7**	5.06 **10**			
7	Rastrabank Chowk	4.6 **4**	6.47 **5**	5.46 **6**			
8	Chorepatan	4.87 **4**					
9	Hallan Chowk	7.73 **10**					
10	Zero KM	5.06 **6**	7.73 **9**	6.02 **65**			
17	Harichowk	6.72 **18**					
18	Baglung Buspark	6.93 **2**	6.72 **17**	7.71 **65**			
22	Amarsingh Chowk	11.77 **3**	10.05 **43**				
43	Chauthe	10.05 **22**					
49	Ban Campus	12.87 **50**	6.24 **67**				
50	Tutunga	12.87 **49**					
65	NEA Office, Malepatan	6.02 **10**	7.71 **18**				
66	WR Hospital	13.05 **3**					
67	Saint Mary's School	18.54 **3**	6.24 **49**				

The weighted graph was constructed with the help of the data from Table 3. 9 Network with link weights and destination nodes.

The weighted graph is shown in Figure 3. 12 Weighted graph between the nodes

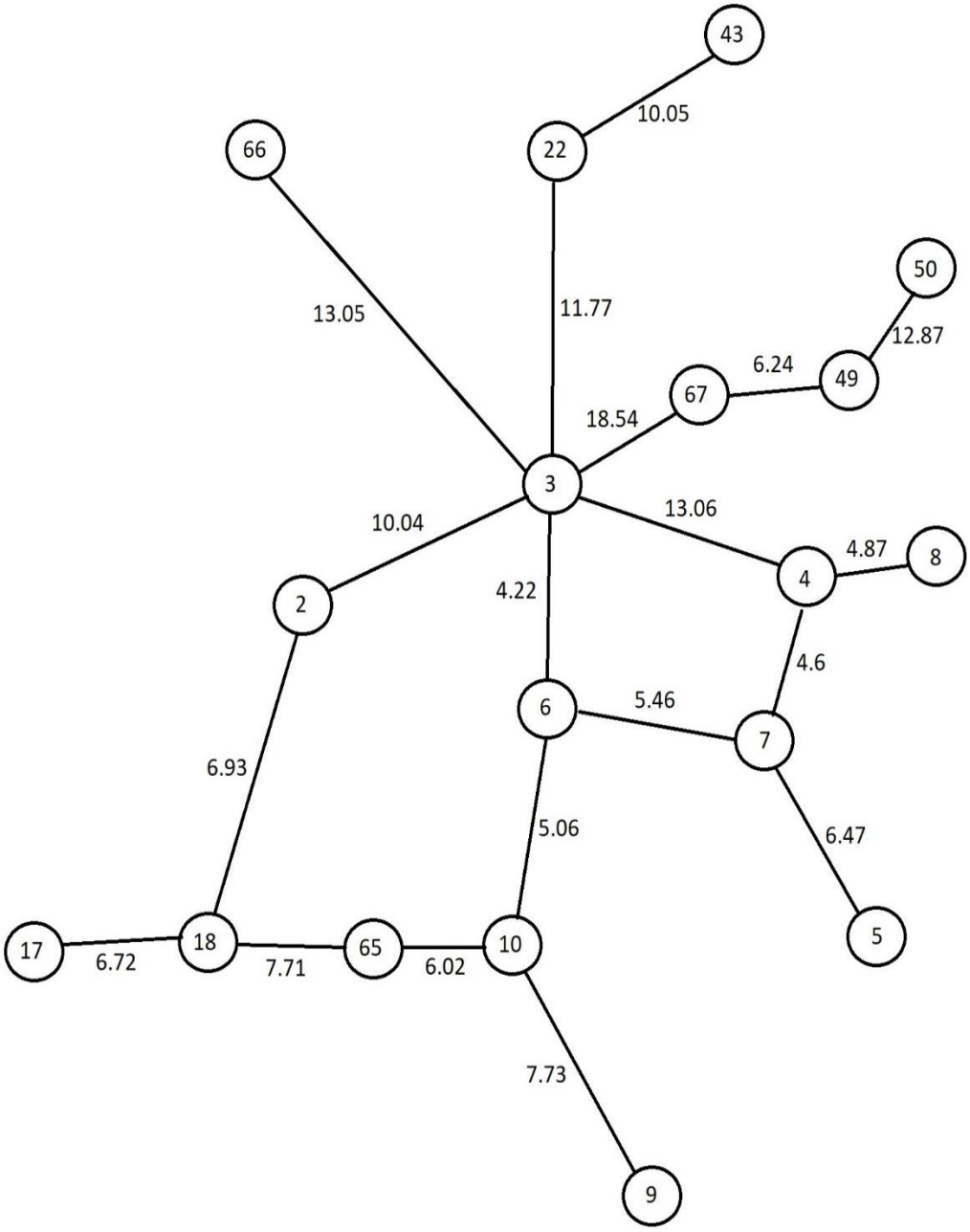


Figure 3. 12 Weighted graph between the nodes

Since, the commuter can start their journey from any node/bus stop so the matrix for each node to be considered as starting point was formed.

Table 3. 10 Network with link weights and destination nodes in matrix form

	2	3	4	5	6	7	8	9	10	17	18	22	43	49	50	65	66	67
2	0.00	10.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	10.04	0.00	13.06	0.00	4.22	0.00	0.00	0.00	0.00	0.00	0.00	11.77	0.00	0.00	0.00	0.00	13.05	18.54
4	0.00	13.06	0.00	0.00	0.00	4.60	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	6.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	4.22	0.00	0.00	0.00	5.46	0.00	0.00	5.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	4.60	6.47	5.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	5.06	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	6.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	0.00	7.71	0.00	0.00
22	0.00	11.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.05	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.05	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	6.24
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	0.00	0.00
65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	7.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00
66	0.00	13.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
67	0.00	18.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.24	0.00	0.00	0.00	0.00

Once the matrix is formed, it was then optimized using Dijkstra's Algorithm in Microsoft Visual Studio 2017 (C++ code) and the result obtained is shown in **Chapter 4**.

3.4.3 Optimization of Transport Network Based on Minimum Travel Cost

In order to optimize the network based on minimum travel cost, the procedure will all remain same except the weights between the nodes had been replaced by travel cost instead of travel time. The weighted graph and matrix was introduced. The optimization in terms of min. travel fare was carried out for the nodes touched by the buses of respective bus service providers and was dealt individually.

CHAPTER 4. RESULT AND DISCUSSION

After the formation of matrix, it was optimized using pseudocode for Dijkstra's Algorithm with printing path in Microsoft Visual Studio 2017 (C++ code) and following results were obtained. Here, in the result box, the "vertex" indicates the origin to destination node where as "travel time" indicates the time required to travel from origin to destination and "path" indicates the nodes to be followed so as to reach the desired destination. For example, to travel from node "0" to node "5", the shortest path is: 0 → 1 → 2 → 3 → 6 → 7 → 5 and it takes 32.5 min.

Optimization based on minimum travel time for routes of:

1. Pokhara City Bus Service:

Result along with printing path:

From Node 0 (Bagar):

```
Vertex          Travel Time          Path
0 -> 1          9.430000             0 1
0 -> 2          11.000000            0 1 2
0 -> 3          18.380000            0 1 2 3
0 -> 4          31.460000            0 1 2 3 6 7 4
0 -> 5          32.500000            0 1 2 3 6 7 5
0 -> 6          21.700000            0 1 2 3 6
0 -> 7          26.460000            0 1 2 3 6 7
0 -> 8          37.720000            0 1 2 3 6 7 4 8
0 -> 9          29.640000            0 1 2 3 6 10 9
0 -> 10         23.950000            0 1 2 3 6 10
0 -> 11         17.430000            0 1 11
0 -> 12         24.970000            0 1 11 12
0 -> 13         30.970000            0 1 11 12 13
0 -> 14         38.110000            0 1 11 12 13 14
0 -> 15         43.990000            0 1 11 12 13 14 15
0 -> 16         4.380000             0 16
0 -> 17         7.000000             0 16 17
0 -> 18         17.000000            0 16 17 18
```

```
0 -> 19          39.990000            0 1 2 3 19
0 -> 20          43.860000            0 1 2 3 19 20
0 -> 21          47.920000            0 1 2 3 19 20 21
0 -> 22          22.830000            0 1 22
0 -> 23          29.920000            0 1 22 23
0 -> 24          36.460000            0 1 22 23 24
0 -> 25          42.630000            0 1 22 23 24 25
0 -> 26          47.900000            0 1 22 23 24 25 26
0 -> 27          61.160000            0 1 22 23 24 25 26 27
0 -> 28          4.540000             0 28
0 -> 29          8.290000             0 28 29
0 -> 30          11.660000            0 28 29 30
0 -> 31          21.380000            0 28 29 30 31
0 -> 32          24.520000            0 28 29 30 31 32
0 -> 33          27.800000            0 28 29 30 31 32 33
0 -> 34          30.640000            0 28 29 30 31 32 33 34
0 -> 35          37.680000            0 28 29 30 31 32 33 34 35
0 -> 36          44.190000            0 28 29 30 31 32 33 34 35 36
0 -> 37          14.320000            0 28 37
0 -> 38          18.550000            0 28 37 38
0 -> 39          28.350000            0 28 37 38 39
0 -> 40          36.350000            0 28 37 38 39 40
0 -> 41          41.650000            0 28 37 38 39 40 41
0 -> 42          46.050000            0 28 37 38 39 40 41 42
0 -> 43          30.790000            0 1 22 43
0 -> 44          34.010000            0 1 22 43 44
0 -> 45          39.670000            0 1 22 43 44 45
0 -> 46          25.830000            0 1 2 46
0 -> 47          35.860000            0 1 22 47
0 -> 48          20.540000            0 1 48
0 -> 49          37.210000            0 1 22 23 49
0 -> 50          49.680000            0 1 22 23 49 50
0 -> 51          41.780000            0 1 2 3 6 7 4 51
```

https://ide.geeksforgeeks.org/Ae0FfSI558

The optimal path in terms of min. time for any trip starting from node (2-51) can be calculated similarly and the respective results are shown in **Annex 4**.

For the validation of result provided by Microsoft Visual Studio after inputting the inputs using C++ code to solve Dijkstra's algorithm, the same inputs were inputted in a different C++ code to solve Dijkstra's algorithm (without showing path) and the same results were obtained at the end which is enclosed as below.

Result without showing path:

From Node 0 (Bagar):

```
Vertex          Travel Time
1                9.430000
2                11.000000
3                18.380000
4                31.460000
5                32.500000
6                21.700000
7                26.460000
8                37.720000
9                29.640000
10               23.950000
11               17.430000
12               24.970000
13               30.970000
14               38.110000
15               43.990000
16               4.380000
17               7.000000
18               17.000000
19               39.990000
20               43.860000
21               47.920000
22               22.830000
23               29.920000
```

```
24                36.460000
25                42.630000
26                47.900000
27                61.160000
28                4.540000
29                8.290000
30                11.660000
30                11.660000
31                21.380000
32                24.520000
33                27.800000
34                30.640000
35                37.680000
36                44.190000
37                14.320000
38                18.550000
39                28.350000
40                36.350000
41                41.650000
42                46.050000
43                30.790000
44                34.010000
45                39.670000
46                25.830000
47                35.860000
48                20.540000
49                37.210000
50                49.680000
51                41.780000
```

https://ide.geeksforgeeks.org/ecqRK4Jrl2

2. Lekhnath City Bus Service:

Result along with path:

From Node 2 (Chipledhunga):

```
Vertex          Travel Time          Path
0 -> 1          9.010000             0 1
0 -> 2          16.550000            0 1 2
0 -> 3          26.030000            0 1 2 3
0 -> 4          4.860000             0 4
0 -> 5          18.560000            0 1 5
0 -> 6          29.800000            0 1 5 6
0 -> 7          34.530000            0 1 5 6 7
0 -> 8          39.270000            0 1 5 6 7 8
0 -> 9          44.270000            0 1 5 6 7 8 9
0 -> 10         49.120000            0 1 5 6 7 8 9 10
0 -> 11         54.010000            0 1 5 6 7 8 9 10 11
0 -> 12         56.650000            0 1 5 6 7 8 9 10 12
0 -> 13         61.200000            0 1 5 6 7 8 9 10 12 13
0 -> 14         65.700000            0 1 5 6 7 8 9 10 12 13 14
0 -> 15         72.040000            0 1 5 6 7 8 9 10 12 13 14 15
0 -> 16         70.970000            0 1 5 6 7 8 9 10 12 13 14 16
0 -> 17         85.550000            0 1 5 6 7 8 9 10 12 13 14 16 17
0 -> 18         48.020000            0 1 5 6 7 8 18
```

https://ide.geeksforgeeks.org/DlOcrQMolr

As we have ranked the nodes according to our preferences. In order to comply to the numbering that we have provided in Table 3. 1 Ranking of Bus Stops, the numbering of nodes in the result has to be converted because for each running of code, the code takes each individual origin as node “0” and other nodes in ascending order.

Name of Node	Node No.																		
	Original (As per Map)	Shown in result after running the Code for each runs																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Chipedhunga	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Prithvi Chowk	3	1	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Birauta	4	2	2	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Lakeside	5	3	3	3	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Baglung Buspark	18	4	4	4	4	0	5	5	5	5	5	5	5	5	5	5	5	5	5	5
Amarsingh Chowk	22	5	5	5	5	5	0	6	6	6	6	6	6	6	6	6	6	6	6	6
Bijaypur	52	6	6	6	6	6	6	0	7	7	7	7	7	7	7	7	7	7	7	7
Buddhi Bazar	53	7	7	7	7	7	7	7	0	8	8	8	8	8	8	8	8	8	8	8
Arghaun Chowk	54	8	8	8	8	8	8	8	8	0	9	9	9	9	9	9	9	9	9	9
Tal Chowk	55	9	9	9	9	9	9	9	9	9	0	10	10	10	10	10	10	10	10	10
Sisuwa	56	10	10	10	10	10	10	10	10	10	10	0	11	11	11	11	11	11	11	11
Begnas Tal	57	11	11	11	11	11	11	11	11	11	11	11	0	12	12	12	12	12	12	12
PU Chowk	58	12	12	12	12	12	12	12	12	12	12	12	12	0	13	13	13	13	13	13
Tamukh	59	13	13	13	13	13	13	13	13	13	13	13	13	13	0	14	14	14	14	14
Satmuhane	60	14	14	14	14	14	14	14	14	14	14	14	14	14	14	0	15	15	15	15
Rupa Tal	61	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	0	16	16	16
Janata Ko Chautara	62	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	0	17	17
Phedi	63	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	0	18
Rithepani	64	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	18	0

Table 4. 1 Conversion of numbering of nodes for each run

Result after transforming the node no. as per of map:

```
Vertex          Travel Time          Path
2 -> 3          9.010000             2 3
2 -> 4          16.550000            2 3 4
2 -> 5          26.030000            2 3 4 5
2 -> 18         4.860000             2 18
2 -> 22         18.560000            2 3 22
2 -> 52         29.800000            2 3 22 52
2 -> 53         34.530000            2 3 22 52 53
2 -> 54         39.270000            2 3 22 52 53 54
2 -> 55         44.270000            2 3 22 52 53 54 55
2 -> 56         49.120000            2 3 22 52 53 54 55 56
2 -> 57         54.010000            2 3 22 52 53 54 55 56 57
2 -> 58         56.650000            2 3 22 52 53 54 55 56 58
2 -> 59         61.200000            2 3 22 52 53 54 55 56 58 59
2 -> 60         65.700000            2 3 22 52 53 54 55 56 58 59 60
2 -> 61         72.040000            2 3 22 52 53 54 55 56 58 59 60 61
2 -> 62         70.970000            2 3 22 52 53 54 55 56 58 59 60 62
2 -> 63         85.550000            2 3 22 52 53 54 55 56 58 59 60 62
                                     63
2 -> 64         48.020000            2 3 22 52 53 54 64
```

The optimal path in terms of min. time for any trip starting from other remaining nodes can be calculated similarly and the respective results are shown in **Annex 4**.

3. City Micro Bus Service:

Result along with path:

From Node 2 (Chipledhunga):

```
Vertex          Travel Time          Path
0 -> 1          10.040000            0 1
0 -> 2          23.100000            0 1 2
0 -> 3          26.190000            0 1 4 5 3
0 -> 4          14.260000            0 1 4
0 -> 5          19.720000            0 1 4 5
0 -> 6          27.970000            0 1 2 6
0 -> 7          27.050000            0 1 4 8 7
0 -> 8          19.320000            0 1 4 8
0 -> 9          13.650000            0 10 9
0 -> 10         6.930000             0 10
0 -> 11         21.810000            0 1 11
0 -> 12         31.860000            0 1 11 12
0 -> 13         34.820000            0 1 17 13
0 -> 14         47.690000            0 1 17 13 14
0 -> 15         14.640000            0 10 15
0 -> 16         23.090000            0 1 16
0 -> 17         28.580000            0 1 17
```

https://ide.geeksforgeeks.org/slOmPDci4p

As we have named the nodes according to our preferences so the numbering of nodes in the result has to be converted so that it comply to our numbering.

Name of Node	Node No.																	
	Original (As per Map)	Shown in result after running the Code for each runs																
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Chipedhunga	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Prithvi Chowk	3	1	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Birauta	4	2	2	0	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Lakeside	5	3	3	3	0	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Srijana Chowk	6	4	4	4	4	0	5	5	5	5	5	5	5	5	5	5	5	5	5
Rastrabank Chowk	7	5	5	5	5	5	0	6	6	6	6	6	6	6	6	6	6	6	6
Chorepatan	8	6	6	6	6	6	6	0	7	7	7	7	7	7	7	7	7	7	7
Hallan Chowk	9	7	7	7	7	7	7	7	0	8	8	8	8	8	8	8	8	8	8
Zero KM	10	8	8	8	8	8	8	8	8	0	9	9	9	9	9	9	9	9	9
Hari Chowk	17	9	9	9	9	9	9	9	9	9	0	10	10	10	10	10	10	10	10
Baglung Buspark	18	10	10	10	10	10	10	10	10	10	10	0	11	11	11	11	11	11	11
Amarsingh Chowk	22	11	11	11	11	11	11	11	11	11	11	11	0	12	12	12	12	12	12
Chauthe	43	12	12	12	12	12	12	12	12	12	12	12	12	0	13	13	13	13	13
Ban Campus	49	13	13	13	13	13	13	13	13	13	13	13	13	13	0	14	14	14	14
Tutunga	50	14	14	14	14	14	14	14	14	14	14	14	14	14	14	0	15	15	15
NEA Office, Malepatan	65	15	15	15	15	15	15	15	15	15	15	15	15	15	15	15	0	16	16
WR Hospital	66	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16	0	17
Saint Mary's School	67	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	17	0

Table 4. 2 Conversion of numbering of nodes for each run

Result after transforming the node no. as per of map:

Vertex	Travel Time	Path
2 -> 3	10.040000	2 3
2 -> 4	23.100000	2 3 4
2 -> 5	26.190000	2 3 6 7 5
2 -> 6	14.260000	2 3 6
2 -> 7	19.720000	2 3 6 7
2 -> 8	27.970000	2 3 4 8
2 -> 9	27.050000	2 3 6 10 9
2 -> 10	19.320000	2 3 6 10
2 -> 17	13.650000	2 18 17
2 -> 18	6.930000	2 18
2 -> 22	21.810000	2 3 22
2 -> 43	31.860000	2 3 22 43
2 -> 49	34.820000	2 3 67 49
2 -> 50	47.690000	2 3 67 49 50
2 -> 65	14.640000	2 18 65
2 -> 66	23.090000	2 3 66
2 -> 67	28.580000	2 3 67

The optimal path in terms of min. time for any trip starting from other remaining nodes can be calculated similarly and the respective results are shown in **Annex 4**.

Optimization based on minimum travel cost for routes of:

Here, in the result box, the “vertex” indicates the origin to destination node where as “fare” indicates the travel cost required to travel from origin to destination and “path” indicates the nodes to be followed so as to reach the desired destination. For example, to travel from node “0” to node “10”, the cheapest path is: 0 → 1 → 10 and it costs NRs. 29. In this case, it also means that there is no any direct connection (bus route) between node “0” and node “10”, so node “10” can only be reached via node “1” which requires a transfer of vehicle at node “1”.

1. Pokhara City Bus Service:

Result showing along with path:

From Node 0 (Bagar):

```
Vertex          Fare          Path
0 -> 1          13            0 1
0 -> 2          13            0 2
0 -> 3          16            0 3
0 -> 4          22            0 4
0 -> 5          23            0 5
0 -> 6          16            0 6
0 -> 7          22            0 7
0 -> 8          23            0 8
0 -> 9          34            0 6 9
0 -> 10         29            0 1 10
0 -> 11         26            0 2 11
0 -> 12         31            0 2 12
0 -> 13         32            0 2 13
0 -> 14         37            0 2 14
0 -> 15         47            0 2 15
0 -> 16         13            0 16
0 -> 17         26            0 16 17
0 -> 18         26            0 16 18
0 -> 19         32            0 3 19
0 -> 20         35            0 3 20
```

```
0 -> 21          42              0 2 21
0 -> 22          19              0 22
0 -> 23          20              0 23
0 -> 24          24              0 24
0 -> 25          28              0 25
0 -> 26          29              0 26
0 -> 27          32              0 27
0 -> 28          13              0 28
0 -> 29          13              0 29
0 -> 30          14              0 30
0 -> 31          22              0 31
0 -> 32          26              0 32
0 -> 33          27              0 33
0 -> 34          28              0 34
0 -> 35          32              0 35
0 -> 36          36              0 36
0 -> 37          14              0 37
0 -> 38          19              0 38
0 -> 39          27              0 39
0 -> 40          28              0 40
0 -> 41          37              0 41
0 -> 42          41              0 42
0 -> 43          32              0 2 43
0 -> 44          41              0 22 44
0 -> 45          44              0 22 45
0 -> 46          26              0 2 46
0 -> 47          32              0 3 47
0 -> 48          28              0 1 48
0 -> 49          33              0 23 49
0 -> 50          36              0 23 50
0 -> 51          35              0 4 51
```

https://ide.geeksforgeeks.org/EzbgooNWqi

Similarly, the results of the nodes touched by Lekhnath and City Micro Bus Services were obtained and incorporated in the **Annex 4**.

CHAPTER 5. CONCLUSIONS AND FUTURE RESEARCH

This chapter concludes the thesis by summing up the contributions made in Section 5.1, and then suggesting some potential directions for future research in Section 5.2.

5.1 Summary and Contributions

With the advancement of technology, various navigation applications have been developed for guiding private vehicles, but very few are designed for public transportation. This study acts as a remedy to overcome this complication in case of public transportation of Pokhara.

The main aim of this thesis is to develop a model that generates itinerary for passengers / commuters in a public transportation network assuming that a passenger's objective is to travel with minimum travel time and travel cost. So as to find the optimal route, the main data regarding travel time and cost of travel were collected. The data obtained were processed. Weighted graph and matrix for each starting node was constructed and analyzed with the help of Microsoft Visual Studio 2017 software using C++ code for Dijkastra's algorithm which also provides the path to be followed. The code was run for each time for each starting node with its own respective inputs. Separate analysis was carried out in case of time and cost. The results of this analysis indicates that when the "time" impedance was used by the algorithm, it generated the shortest path between the origin and destination along with the path to be followed. Similarly, when the "cost" impedance was used, it generated the cheapest route. The results were cross checked with a different C++ code to solve the algorithm. From both the codes, the output obtained was same. The results obtained were completely unique and distinct for each individual node which was considered as starting point. Since, this research was able to develop a model to find an optimal path so the research objectives were met.

Though it is a challenge to practically implement the results of this study as the navigation practice is yet to be widely accepted by public in finding the location within Nepal but nevertheless it can definitely play a significant role in the field of optimization of public transport network in near future. It is believed that the methodology can be implemented in other cities as well to optimize the network as per

their needs. Also, the nodes of the public transit network of Pokhara city touches almost all of the touristic destinations in the town.

5.2 Future Research

This type of study relating to optimization of public transport network has never been conducted before in case of Pokhara city so, this study is one of the first if not, it is the first attempt to study and find the optimal transit route in terms of minimum travel time and cost. Due to time and resources constraints, there are some limitations in the study. Still lot more could be done in this type of study. Followings are some of the recommendations for future research:

i. To make this study more practical other parameters such as speed of vehicle, condition of road, etcetera could have been considered.
ii. The research could be further extended by figuring out the factors like: transfer of vehicle, walking time, no. of transfers, etcetera.
iii. The factors influencing travel time like: delays at bus stops, traffic conditions, weather conditions, etc. could be focused which would provide a clearer picture of the study and comply with the realistic situation.
iv. The study based on isolated travel time at peak and off-peak hours could help to answer the ambiguity of commuters.

REFERENCES

Adhikari, P. (2014). *Optimization of Road Network: A Case Study of Makwanpur District on Agriculture and Construction Material Perspective.* Department of Civil Engineering, Pulchowk Campus.

Alija, A. S. (2015). *Analysis of Dijkstra's and A* Algorithm to find the shortest path.* University of Tun Hussein Onn Malaysia Master's Thesis.

CBS, C. B. (2011). *National Population and Housing Census 2011.* Central Bureau of Statistics, Kathmandu.

Cormen, T. L. (2001). *Single-source shortest Paths : Introduction to algorithms.* 2nd ed. Cambridge, MA: MIT Press,581-635.

DoT, D. o. (2016). Nepal Tourism Statistics, 2016.

GeeksforGeeks. (n.d.). *GeeksforGeeks A computer science portal for geeks*. Retrieved from Geeksforgeeks A computer science portal for geeks: https://www.geeksforgeeks.org/

Geisberger, R. (2011). *Advanced Route Planning in Transportation Networks.* Karlsruhe Institute of Technology Doctoral Thesis.

Horst W. Hamacher, S. R. (2006). Algorithms for time-dependent bicriteria shortest path problems. *Discrete Optimization 3 (2006) 238–254*.

Lilian S.C. Pun-Cheng, A. W. (2015). Optimal route computation for circular public transport routes with differential fare structure. *Travel Behaviour and Society 3 (2016) 71–77*.

O. Dib, L. M.-A. (2016). (2016). An advanced GA-VNS combination for multicriteria route planning in public transit networks. *Expert Systems With Applications 72 (2017) 67–82*.

Paola Modesti, A. S. (1998). A utility measure for finding multiobjective shortest paths in urban multimodal transportation networks. *European Journal of Operational Research 111 (1998) 495-508*.

Paudel, K. B. (n.d.). Situation of Urban Transportation facility in Pokhara Sub-Metropolitan City. Pokhara, Nepal.

PLMC. (2017). *Pokhara Metropolitan City*. Retrieved from http://www.pokharamun.gov.np/

Puthuparampil, M. (2007). *Report Dijkstra's Algorithm [online]*. Unpublished Presentation, Computer Science Department, New York University. Available from:http://www.cs.nyu.edu/courses/summer07/G22.2340001/Presentations/Puthuparampil.pdf. (October, 2014).

R.P Pokharel, N. R. (2018). Open space: Typology and distribution in Pokhara Lekhnath metropolitan city. *The Geographical Journal of Nepal*.

Sadeghi-Niaraki, A. V. (2011). *Real world representation of a road network for route planning in GIS.* Expert Systems with Applications, 38 (10), 11999-12008.

Shakya, S. D. (2013). "Transportation Network of Pokhara Sub-Metropolitan City." . Pokhara: Proceedings of IOE Graduate Conference, Vol. 1 168-174.

Vi Ngoc-Nha Tran, S. D. (2012). A comparative study of vehicles' routing algorithms for route planning in smart cities.

Wang, J.-J. (2008). *Optimal paths based on time and fares in transit networks.* Taipei: National Cheng Kung University Master Thesis.

Xiaotang, X. (2009). *Bus Trip Optimization at Directional Level in GIS.* Wuhan city, China: International Institute for Geo-information Science and Earth Observation Master Thesis.

Zhan, F. B. (1998). *Shortest Path Algorithms: An Evaluation Using Real Road Networks.* Transportation Science, 32(1), 65-73.

ANNEX 1. COLLECTED DATA

The bus routes around Pokhara city served by different bus service providers are as follows:

S. N.	Bus Routes
1	Bagar-Lakeside
2	Bagar-Chorepatan
3	Bagar-Lamachaur
4	Bagar-Gufa
5	Bagar-Ghattekuna
6	Bagar-Puranchaur-Raikar
7	Bagar-Bhunpare
8	Bagar-Khaltemasina
9	Simpani-Chauthe-Majheripatan
10	Simpani-Mahatgauda
11	Bagar (Ward No. 1)-Miyapatan-Chauthe
12	Manipal-Tutunga
13	Chorepatan-Ghattekuna
14	Chorepatan-Raikar
15	Hallanchowk-Kahun Khola-Kaseri
16	Manipal-Davis Fall-Belghari
17	Manipal-Kahundanda (Seasonal)
18	Prithvi Chowk-Shanti Stupa-Kafal danda (Seasonal)

Table A1. 1 Bus Routes of Pokhara City Bus Service

S. N.	Bus Routes
1	Baglung Buspark-Begnas Tal
2	Baglung Buspark-Rupa Tal
3	Baglung Buspark-Phedi
4	Baglung Buspark-Rithepani
5	Lakeside-Begnas Tal

Table A1. 2 Bus Routes of Lekhnath City Bus Service

S. N.	Bus Routes
1	Hari Chowk-Western Regional Hospital
2	Hari Chowk-Chauthe
3	Hari Chowk-Chorepatan
4	Chipledhunga-Hallan Chowk
5	Chipledhunga-Lakeside (Barahi Chowk)
6	Chipledhunga-Tutunga

Table A1. 3 Bus Routes of City Micro Bus Service

S. N.	Bus Routes
1	Prithvi Chowk-Kande

Table A1. 4 Bus Routes of Prithvi Bus Service

Travel Time for various routes of Pokhara City Bus Services

Date: 2074/09/29 (Saturday) Time: 1:29 pm

Route: Bagar-Lakeside Bus No: Ga. 1. Kha 3444

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Bagar	7	-	-	-	-	-	
2	Mahendrapool	3	4	4	51	-	-	4.85
3	Chipledhunga	2	0	1	10	-	-	1.17
4	Prithvi Chowk	17	3	4	8	9	46	4.13
5	Birauta	0	5	6	22	-	-	16.13
6	Rastrabank Chowk	2	1	2	57	-	-	2.95
7	Lakeside (Barahi Chowk)	-	18	4	26	-	-	4.43
			Sum	21	174	9	46	33.67
			Total	33	40			

Table A1. 5 Travel time of Bagar-Lakeside route

Date: 2074/09/29 (Saturday) Time: 2:19 pm

Route: Lakeside-Bagar Bus No: Ga. 1. Kha 3444

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Lakeside (Barahi Chowk)	25	-	-	-	-	-	
2	Rastrabank Chowk	1	0	7	39	-	-	7.65
3	Birauta	18	0	2	42	15	23	2.70
4	Prithvi Chowk	1	22	13	6	-	-	28.48
5	Chipledhunga	0	6	4	55	-	-	4.92
6	Mahendrapool	12	1	1	14	4	13	1.23
7	Bagar	-	28	10	47	-	-	15.00
			Sum	37	203	19	36	59.98
			Total	59	59			

Table A1. 6 Travel time of Lakeside-Bagar route

Date: 2074/09/29 (Saturday) Time: 3:27 pm

Route: Bagar-Chorepatan Bus No: Ga. 1. Kha 3527

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Bagar	10	-	-	-	-	-	
2	Mahendrapool	19	4	5	0	10	57	5.00
3	Prithvi Chowk	2	19	5	19	3	35	16.27
4	Srijana Chowk	1	0	1	25	-	-	5.00
5	Rastrabank Chowk	1	6	3	24	-	-	3.40
6	Birauta	0	1	2	23	-	-	2.38
7	Chorepatan	-	3	4	36	-	-	4.60
			Sum	20	127	13	92	36.65
			Total	36	39			

Table A1. 7 Travel time of Bagar-Chorepatan route

Date: 2074/09/29 (Saturday) Time: 4:08 pm

Route: Chorepatan-Bagar Bus No: Ga. 1. Kha 3527

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chorepatan	7	-	-	-	-	-	
2	Birauta	0	0	5	36	-	-	5.60
3	Rastrabank Chowk	2	5	2	20	-	-	2.33
4	Srijana Chowk	0	2	3	39	-	-	3.65
5	Prithvi Chowk	48	0	1	32	16	26	1.53
6	Mahendrapool	4	27	10	15	6	11	26.68
7	Bagar	-	27	8	55	-	-	15.10
			Sum	29	197	22	37	54.90
			Total	54	54			

Table A1. 8 Travel time of Chorepatan-Bagar route

Date: 2074/10/6 (Saturday) Time: 11:07 am

Route: Chorepatan-Raikar Bus No: Ga. 1. Kha 5004

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chorepatan	6	-	-	-	-	-	
2	Birauta	18	3	8	56	5	14	8.93
3	Prithvi Chowk	6	8	10	24	-	-	15.63
4	Mahendrapool	5	10	8	41	-	-	8.68
5	Bagar	25	12	7	52	5	10	7.87
6	Sundarfeed	3	0	2	18	-	-	7.47
7	Engineering Campus	0	4	4	10	-	-	4.17
8	Lamachaur	10	12	2	25	-	-	2.42
9	Akalaa Devi Temple	3	6	10	32	-	-	10.53
10	Bhunpare	2	4	3	56	-	-	3.93
11	Besi Chowk	5	6	4	20	-	-	4.33
12	Titepani	3	10	3	1	-	-	3.02
13	Puranchaur	0	8	7	32	-	-	7.53
14	Raikar	-	3	6	2	-	-	6.03
			Sum	74	369	10	24	90.55
			Total	90	33			

Table A1. 9 Travel time of Chorepatan-Raikar route

Date: 2074/10/6 (Saturday) Time: 1:04 pm

Route: Raikar-Chorepatan Bus No: Ga. 1. Kha 5004

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Raikar	3	-	-	-	-	-	
2	Puranchaur	6	1	7	14	-	-	7.23
3	Titepani	5	1	6	36	-	-	6.60
4	Besi Chowk	2	0	3	24	-	-	3.40
5	Bhunpare	3	0	3	2	-	-	3.03
6	Akalaa Devi Temple	5	1	2	25	-	-	2.42
7	Lamachaur	8	3	9	22	-	-	9.37
8	Engineering Campus	4	2	4	32	-	-	4.53
9	Sundarfeed	0	3	4	24	-	-	4.40
10	Bagar	6	12	2	47	4	36	2.78
11	Chipledhunga	3	7	8	21	2	12	12.95
12	Prithvi Chowk	4	8	4	29	-	-	6.68
13	Srijana Chowk	0	3	1	32	-	-	1.53
14	Rastrabank Chowk	2	2	4	25	-	-	4.42
15	Birauta	0	4	3	41	-	-	3.68
16	Chorepatan	-	4	4	33	-	-	4.55
			Sum	64	407	6	48	77.58
			Total	77	35			

Table A1. 10 Travel time of Raikar-Chorepatan route

Date: 2074/10/5 (Friday) Time: 9:15 am

Route: Chorepatan-Ghattekuna Bus No: Ga. 1. Kha 6396

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chorepatan	8	-	-	-	-	-	
2	Birauta	16	2	9	26	4	38	9.43
3	Rastrabank Chowk	5	0	3	8	-	-	7.77
4	Srijana Chowk	2	10	6	41	-	-	6.68
5	Prithvi Chowk	5	14	1	42	2	2	1.70
6	Chipledhunga	9	7	5	35	-	-	7.62
7	Bagar	21	10	12	16	5	26	12.27
8	Sundarfeed	4	1	2	25	-	-	7.85
9	Maidan	3	5	9	32	-	-	9.53
10	Gufa	2	6	4	2	-	-	4.03
11	Riverdale School	1	6	8	54	-	-	8.90
12	Jogimadi	2	10	5	8	-	-	5.13
13	Tupke	0	2	5	24	-	-	5.40
14	Ghattekuna	-	5	4	13	-	-	4.22
			Sum	73	326	11	66	90.53
			Total	90	32			

Table A1. 11 Travel time of Chorepatan-Ghattekuna route

Date: 2074/10/5 (Friday) Time: 11:03 am

Route: Ghattekuna-Chorepatan Bus No: Ga. 1. Kha 6396

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Ghattekuna	4	-	-	-	-	-	
2	Tupke	3	0	4	35	-	-	4.58
3	Jogimadi	11	1	5	6	4	15	5.10
4	Riverdale School	5	0	6	32	-	-	10.78
5	Gufa	8	2	10	42	-	-	10.70
6	Maidan	5	3	4	25	-	-	4.42
7	Sundarfeed	7	4	10	2	-	-	10.03
8	Bagar	4	20	4	32	-	-	4.53
9	Mahendrapool	3	6	8	44	-	-	8.73
10	Prithvi Chowk	4	5	5	47	-	-	5.78
11	Birauta	0	5	6	10	-	-	6.17
12	Chorepatan	-	8	4	27	-	-	4.45
			Sum	66	302	4	15	75.28
			Total	75	17			

Table A1. 12 Travel time of Ghattekuna-Chorepatan route

Date: 2074/10/08 (Monday) Time: 2:39 pm
Route: Bagar-Khalte Masina Bus No: Ga. 1. Kha 6368

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Bagar	34	-	-	-	-	-	
2	Chipledhunga	14	10	15	53	3	48	15.88
3	Prithvi Chowk	10	10	6	5	2	8	9.88
4	Amar Singh Chowk	4	12	6	13	5	3	8.35
5	Ram Bazar	8	5	3	16	4	0	8.32
6	Kaji Pokhari	4	6	2	26	-	-	6.43
7	Nirajan Chowk	0	5	6	47	-	-	6.78
8	Chinedanda	0	14	3	16	-	-	3.27
9	Khalte Masina	-	12	8	15	-	-	8.25
			Sum	49	191	14	59	67.17
			Total	67	10			

Table A1. 13 Travel time of Bagar-Khalte Masina route

Date: 2074/10/08 (Monday) Time: 3:58 pm
Route: Khalte Masina-Bagar Bus No: Ga. 1. Kha 6348

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Khalte Masina	8	-	-	-	-	-	
2	Chinedanda	11	1	18	16	-	-	18.27
3	Nirajan Chowk	6	0	7	16	-	-	7.27
4	Kaji Pokhari	11	0	5	33	-	-	5.55
5	Ram Bazar	13	8	6	39	-	-	6.65
6	Amar Singh Chowk	13	7	4	53	4	31	4.88
7	Prithvi Chowk	8	20	5	9	-	-	9.67
8	Chipledhunga	13	11	11	1	-	-	11.02

9	Bagar	-	36	17	7	-	-	17.12
			Sum	73	174	4	31	80.42
			Total	80	25			

Table A1. 14 Travel time of Khalte Masina-Bagar route

Date: 2074/10/02 (Tuesday) Time: 12:36 pm

Route: Simpani-Majheripatan Bus No: Ga. 1. Kha 6893

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Simpani	4	-	-	-	-	-	
2	Hari Chowk	12	1	2	52	3	51	2.87
3	Baglung Bus Park	2	3	8	43	-	-	12.57
4	Chipledhunga	8	8	10	54	3	26	10.90
5	Prithvi Chowk	7	6	5	12	4	33	8.63
6	Amarsingh Chowk	3	4	7	2	2	12	11.58
7	Chauthe	2	9	4	52	-	-	7.07
8	Bhotechautara	0	3	2	32	-	-	2.53
9	Majheripatan	-	4	5	14	-	-	5.23
			Sum	43	261	12	122	61.38
			Total	61	23			

Table A1. 15 Travel time of Simpani-Majheripatan route

Date: 2074/10/02 (Tuesday) Time: 1:52 pm

Route: Majheripatan-Simpani Bus No: Ga. 1. Kha 6893

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Majheripatan	3	-	-	-	-	-	
2	Bhotechautara	4	0	6	5	-	-	6.08
3	Chauthe	6	3	3	54	3	15	3.90
4	Amarsingh Chowk	7	4	5	36	4	35	8.85
5	Prithvi Chowk	6	6	8	51	3	33	13.43
6	Chipledhunga	8	7	4	56	6	2	8.48
7	Bagar	2	11	15	38	-	-	21.67
8	Simpani	-	5	4	23	-	-	4.38
			Sum	45	263	16	85	66.80
			Total	66	48			

Table A1. 16 Travel time of Majheripatan-Simpani route

Date: 2074/10/08 (Monday) Time: 12:23 pm

Route: Simpani-Mahatgauda Bus No: Ga. 1. Kha 6992

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Simpani	4	-	-	-	-	-	
2	Harichowk	10	2	2	26	3	27	2.43
3	Baglung Buspark	10	2	10	9	-	-	13.60
4	Chipledhunga	4	9	9	44	-	-	9.73
5	Prithvi Chowk	7	8	4	28	9	19	4.47
6	Himalayan Eye Hospital	3	6	7	52	-	-	17.18
7	Int'l Mountain Museum	0	4	3	6	-	-	3.10
8	Mahatgauda	-	7	3	29	-	-	3.48
			Sum	38	194	12	46	54.00
			Total	54	0			

Table A1. 17 Travel time of Simpani-Mahatgauda route

Date: 2074/10/08 (Monday) Time: 1:23 pm

Route: Mahatgauda-Simpani Bus No: Ga. 1. Kha 6340

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Mahatgauda	3	-	-	-	-	-	
2	Int'l Mountain Museum	7	0	4	39	-	-	4.65
3	Himalayan Eye Hospital	16	0	4	38	15	23	4.63
4	Prithvi Chowk	8	13	10	40	5	34	26.05
5	Chipledhunga	3	11	4	46	-	-	10.33
6	Baglung Buspark	0	2	6	52	-	-	6.87
7	Harichowk	0	10	3	52	-	-	3.87
8	Simpani	-	1	2	34	-	-	2.57
			Sum	33	301	20	57	58.97
			Total	58	58			

Table A1. 18 Travel time of Mahatgauda-Simpani route

Date: 2074/10/09 (Tuesday) Time: 9:36 am

Route: Bagar Ward Office-Miyapatan Bus No: Ga. 1. Kha 6804

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Bagar Ward Office	18	-	-	-	-	-	
2	Chipledhunga	11	13	13	52	3	42	13.87
3	Prithvi Chowk	8	10	5	58	3	54	9.67
4	Amarsingh Chowk	8	7	4	46	1	21	8.67
5	Miyapatan	-	15	11	54	-	-	13.25
			Sum	33	210	7	117	45.45
			Total	45	27			

Table A1. 19 Travel time of Bagar Ward Office-Miyapatan route

Date: 2074/10/09 (Tuesday) Time: 11:02 am

Route: Miyapatan-Bagar Ward Office Bus No: Ga. 1. Kha 6804

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Miyapatan	11	-	-	-	-	-	
2	Amarsingh Chowk	10	6	12	48	2	14	12.80
3	Prithvi Chowk	11	10	5	58	1	58	8.20
4	Chipledhunga	8	9	5	48	3	24	7.77
5	Bagar Ward Office	-	15	12	23	-	-	15.78
			Sum	34	177	6	96	44.55
			Total	44	33			

Table A1. 20 Travel time of Miyapatan-Bagar Ward Office route

Date: 2074/10/11 (Thursday) Time: 11:45 am

Route: Manipal-Tutunga Bus No: Ga. 1. Kha 3826

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Manipal	11	-	-	-	-	-	
2	Mahendrapool	4	7	9	38	4	23	9.63
3	Amarsingh Chowk	9	5	9	17	3	56	13.67
4	Rambazar	8	7	5	31	-	-	9.45
5	Ban Campus	3	6	5	56	-	-	5.93
6	Tutunga	-	10	11	21	-	-	11.35
			Sum	39	163	7	79	50.03
			Total	50	2			

Table A1. 21Travel time of Manipal-Tutunga route

Date: 2074/10/11 (Thursday) Time: 1:30 pm

Route: Tutunga-Manipal Bus No: Ga. 1. Kha 7059

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Tutunga	7	-	-	-	-	-	
2	Ban Campus	5	2	13	35	2	25	13.58
3	Rambazar	17	4	6	14	-	-	8.65
4	Amarsingh Chowk	8	8	5	42	4	32	5.70
5	Mahendrapool	6	14	9	12	4	8	13.73
6	Manipal	-	15	8	46	-	-	12.90
			Sum	41	149	10	65	54.57
			Total	54	34			

Table A1. 22 Travel time of Tutunga-Manipal route

Date: 2074/10/11 (Thursday) Time: 9:27 am

Route: Manipal-Belghari Bus No: Ga. 1. Kha 5284

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Manipal	9	-	-	-	-	-	
2	Mahendrapool	8	7	8	51	4	53	8.85
3	Amarsingh Chowk	9	6	10	39	2	51	15.53
4	Prithvi Chowk	8	10	7	29	6	52	10.33
5	Birauta	2	4	3	48	3	21	10.67
6	Belghari	-	9	9	1	-	-	12.37
			Sum	37	168	15	177	57.75
			Total	57	45			

Table A1. 23 Travel time of Manipal-Belghari route

Date: 2074/10/11 (Thursday) Time: 10:30 am

Route: Belghari-Manipal Bus No: Ga. 1. Kha 5284

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Belghari	5	-	-	-	-	-	
2	Birauta	16	2	8	16	4	25	8.27
3	Rastrabank Chowk	11	1	4	23	-	-	8.80
4	Srijana Chowk	0	8	5	39	-	-	5.65
5	Prithvi Chowk	9	14	6	10	2	35	6.17
6	Amarsingh Chowk	8	13	7	25	2	5	10.00
7	Mahendrapool	12	10	8	34	3	11	10.65
8	Manipal	-	13	9	52	-	-	13.05
			Sum	47	199	11	76	62.58
			Total	62	35			

Table A1. 24 Travel time of Belghari-Manipal route

Date: 2074/10/04 (Thursday) Time: 1:42 pm

Route: Hallanchowk-Kaseri Bus No: Ga. 1. Kha 3421

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Hallanchowk	10	-	-	-	-	-	
2	Zero KM	1	1	5	47	-	-	5.78
3	Srijana Chowk	1	0	1	7	-	-	1.12
4	Prithvi Chowk	2	1	1	55	-	-	1.92
5	Chipledhunga	6	2	2	23	-	-	2.38
6	Mahendrapool	38	6	2	18	4	13	2.30
8	Sitaldevi	0	2	6	55	-	-	11.13
9	Bajhapatan	0	12	8	33	-	-	8.55
10	Kahun Khola Chowk	2	25	0	40	-	-	0.67
11	Sajha	0	4	7	52	-	-	7.87
12	Kaseri	-	7	5	35	-	-	5.58
			Sum	37	365	4	13	47.30
			Total	47	18			

Table A1. 25 Travel time of Hallanchowk-Kaseri route

Date: 2074/10/04 (Thursday) Time: 2:34 pm

Route: Kaseri-Hallanchowk Bus No: Ga. 1. Kha 3826

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Kaseri	3	-	-	-	-	-	
2	Sajha	2	0	6	10	-	-	6.17
3	Kahun Khola Chowk	6	2	6	24	7	56	6.40
4	Bajhapatan	1	0	3	24	-	-	11.33
5	Sitaldevi	6	2	6	32	-	-	6.53
7	Mahendrapool	11	8	4	53	5	29	4.88
8	Prithvi Chowk	15	15	6	8	4	43	11.62
9	Srijana Chowk	0	0	0	41	2	12	5.40
10	Zero KM	0	3	1	11	-	-	3.38
11	Hallanchowk	-	14	5	36	-	-	5.60
			Sum	37	239	18	140	61.32
			Total	61	19			

Table A1. 26 Travel time of Kaseri-Hallanchowk route

Travel Time for various routes of Lekhnath City Bus Services

Date: 2074/10/09 (Tuesday) Time: 12:40 pm

Route: Baglung Bus Park- Rupa Tal Bus No: Ga. 1. Kha 4771

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Baglung Bus Park	7	-	-	-	-	-	
2	Chipledhunga	7	3	4	3	1	56	4.05
3	Prithvi Chowk	15	3	6	55	4	49	8.85
4	Amar Singh Chowk	12	2	7	17	3	30	12.10
5	Bijaypur	2	5	9	13	-	-	12.72
6	Buddhi Bazar	1	2	3	5	-	-	3.08
7	Arghaun Chowk	2	5	4	21	-	-	4.35
8	Tal Chowk	3	4	4	11	1	31	4.18
9	Sisuwa	6	2	3	19	6	15	4.83
10	PU Chowk	0	6	4	37	-	-	10.87
11	Tamukh	2	1	3	55	-	-	3.92
12	Satmuhane	1	9	4	21			4.35
13	Rupa Tal	-	16	6	51	-	-	6.85
			Sum	57	308	15	181	80.15
			Total	80	9			

Table A1. 27 Travel time of Baglung Buspark-Rupa Tal route

Date: 2074/10/09 (Tuesday) Time: 2:20 pm

Route: Rupa Tal-Baglung Bus Park Bus No: Ga. 1. Kha 4771

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Rupa Tal	4	-	-	-	-	-	
2	Satmuhane	5	0	5	50	-	-	5.83
3	Tamukh	1	0	4	46	2	25	4.77
4	PU Chowk	9	2	3	15	-	-	5.67
5	Sisuwa	3	4	5	50	-	-	5.83
6	Tal Chowk	5	2	3	36	1	54	3.60
7	Arghaun Chowk	8	3	3	52	3	26	5.77
8	Buddhi Bazar	7	2	4	4	4	29	7.50
9	Bijaypur	5	3	3	0	-	-	7.48
10	Amar Singh Chowk	2	8	8	52	-	-	8.87
11	Prithvi Chowk	2	7	8	16	-	-	8.27
12	Chipledhunga	0	14	7	48	-	-	7.80
13	Baglung Bus Park	-	6	5	12	-	-	5.20
			Sum	58	381	10	134	76.58
			Total	76	35			

Table A1. 28 Travel time of Rupa Tal-Baglung Buspark route

Date: 2074/10/10 (Wednesday) Time: 11:23 am

Route: Baglung Bus Park- Begnas Tal Bus No: Ga. 1. Kha 3511

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Baglung Bus Park	9	-	-	-	-	-	
2	Chipledhunga	8	4	4	36	2	12	4.60
3	Prithvi Chowk	13	5	7	24	5	14	9.60
4	Amar Singh Chowk	10	4	7	32	4	2	12.77
5	Bijaypur	4	6	10	8	-	-	14.17
6	Buddhi Bazar	2	4	2	28			2.47
7	Arghaun Chowk	3	6	4	5	-	-	4.08
8	Tal Chowk	4	8	4	32	1	22	4.53
9	Sisuwa	2	6	3	10	-	-	4.53
10	Begnas Tal	-	12	4	59	-	-	4.98
			Sum	45	234	12	50	61.73
			Total	61	44			

Table A1. 29 Travel time of Baglung Buspark-Begnas Tal route

Date: 2074/10/10 (Wednesday) Time: 12:34 pm

Route: Begnas Tal-Baglung Bus Park Bus No: Ga. 1. Kha 4384

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Begnas Tal	12	-	-	-	-	-	
2	Sisuwa	5	2	4	9	-	-	4.15
3	Tal Chowk	6	5	3	9	-	-	3.15
4	Arghaun Chowk	2	0	4	11	4	15	4.18
5	Buddhi Bazar	3	2	3	46	3	48	8.02
6	Bijaypur	4	3	2	36	-	-	6.40
7	Amar Singh Chowk	10	6	13	12	-	-	13.20
8	Prithvi Chowk	2	15	7	27	-	-	7.45
9	Chipledhunga	0	8	5	26	-	-	5.43
10	Baglung Bus Park	-	3	6	8	-	-	6.13
			Sum	47	184	7	63	58.12
			Total	58	7			

Table A1. 30 Travel time of Begnas Tal-Baglung Buspark route

Date: 2074/10/12 (Friday) Time: 10:21 am

Route: Baglung Bus Park- Phedi Bus No: Ga. 1. Kha 3914

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Baglung Bus Park	8	-	-	-	-	-	
2	Chipledhunga	5	5	4	48	2	36	4.80
3	Prithvi Chowk	8	1	6	15	1	18	8.85
4	Amar Singh Chowk	11	1	7	28	1	58	8.77
5	Bijaypur	5	2	7	11	-	-	9.15
6	Buddhi Bazar	8	2	3	29	-	-	3.48
7	Arghaun Chowk	1	2	4	45	-	-	4.75
8	Tal Chowk	5	8	3	53	4	21	3.88
9	Sisuwa	4	3	3	31	1	10	7.87
10	PU Chowk	2	12	5	52	-	-	7.03
11	Tamukh	0	2	3	34	-	-	3.57
12	Satmuhane	1	8	4	6			4.10
13	Janata Ko Chautara	1	4	5	40			5.67
14	Phedi	-	9	12	18	-	-	12.30
			Sum	66	410	9	143	84.22
			Total	84	13			

Table A1. 31 Travel time of Baglung Buspark-Phedi route

Date: 2074/10/12 (Friday) Time: 12:08 pm

Route: Phedi-Baglung Bus Park Bus No: Ga. 1. Kha 3914

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Phedi	6	-	-	-	-	-	
2	Janata Ko Chautara	5	0	16	51	-	-	16.85
3	Satmuhane	7	2	4	53	-	-	4.88
4	Tamukh	4	1	4	46	2	25	4.77
5	PU Chowk	9	3	2	38	-	-	5.05
6	Sisuwa	5	4	6	24	-	-	6.40
7	Tal Chowk	6	5	4	52	2	36	4.87
8	Arghaun Chowk	3	3	3	26	-	-	6.03
9	Buddhi Bazar	4	2	3	47	3	57	3.78
10	Bijaypur	3	5	3	26	-	-	7.38
11	Amar Singh Chowk	3	8	10	27	-	-	10.45
12	Prithvi Chowk	4	11	10	14	-	-	10.23
13	Chipledhunga	2	10	8	42	-	-	8.70
14	Baglung Bus Park	-	7	3	51	-	-	3.85
			Sum	76	497	7	118	93.25
			Total	93	15			

Table A1. 32 Travel time of Phedi-Baglung Buspark route

Date: 2074/10/12 (Friday) Time: 2:25 pm

Route: Baglung Bus Park- Rithepani Bus No: Ga. 1. Kha 4225

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Baglung Bus Park	11	-	-	-	-	-	
2	Chipledhunga	4	8	5	36	3	10	5.60
3	Prithvi Chowk	9	7	7	32	2	8	10.70
4	Amar Singh Chowk	6	3	6	13	2	16	8.35
5	Bijaypur	4	5	8	19	-	-	10.58
6	Buddhi Bazar	5	4	2	38	-	-	2.63
7	Arghaun Chowk	4	3	3	57	-	-	3.95
8	Rithepani	-	13	8	45	-	-	8.75
			Sum	39	240	7	34	50.57
			Total	50	34			

Table A1. 33 Travel time of Baglung Buspark-Rithepani route

Date: 2074/10/12 (Friday) Time: 12:08 pm

Route: Rithepani-Baglung Bus Park Bus No: Ga. 1. Kha 4225

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Rithepani	8	-	-	-	-	-	
2	Buddhi Bazar	6	2	15	24	3	15	15.40
3	Bijaypur	4	1	3	53	-	-	7.13
4	Amar Singh Chowk	1	5	8	41	-	-	8.68
5	Prithvi Chowk	2	4	7	39	-	-	7.65
6	Chipledhunga	0	5	6	21	-	-	6.35
7	Baglung Bus Park	-	3	4	37	-	-	4.62
			Sum	43	215	3	15	49.83
			Total	49	50			

Table A1. 34 Travel time of Rithepani-Baglung Buspark route

Date: 2074/10/13 (Saturday) Time: 12:19 pm

Route: Lakeside- Begnas Tal Bus No: Ga. 1. Kha 3499

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Lakeside	5	-	-	-	-	-	
2	Birauta	3	2	10	36	-	-	10.60
3	Prithvi Chowk	10	4	8	39	3	21	8.65
4	Amar Singh Chowk	7	1	8	10	2	33	11.52
5	Bijaypur	5	4	10	24	-	-	12.95
6	Buddhi Bazar	1	3	2	1	-	-	2.02
7	Arghaun Chowk	2	2	3	15	-	-	3.25
8	Tal Chowk	3	9	3	51	2	17	3.85
9	Sisuwa	3	5	3	58	-	-	6.25
10	Begnas Tal	-	9	5	23	-	-	5.38
			Sum	52	257	7	71	64.47
			Total	64	28			

Table A1. 35 Travel time of Lakeside-Begnas Tal route

Date: 2074/10/13 (Saturday) Time: 2:02 pm

Route: Begnas Tal-Lakeside Bus No: Ga. 1. Kha 3499

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Begnas Tal	8	-	-	-	-	-	
2	Sisuwa	4	1	5	2	-	-	5.03
3	Tal Chowk	7	2	3	42	3	2	3.70
4	Arghaun Chowk	2	1	4	33	-	-	7.58
5	Buddhi Bazar	3	2	2	58	2	11	2.97
6	Bijaypur	1	0	3	2	-	-	5.22
7	Amar Singh Chowk	4	7	11	36	-	-	11.60
8	Prithvi Chowk	1	12	8	21	-	-	8.35
9	Birauta	0	2	6	26	-	-	6.43
10	Lakeside	-	3	8	21	-	-	8.35
			Sum	50	241	5	13	59.23
			Total	59	14			

Table A1. 36 Travel time of Begnas Tal-Lakeside route

Travel Time for various routes of City Micro Bus Services

Date: 2074/10/01 (Monday) Time: 10:12 am

Route: Hari Chowk-Western Regional Hospital Bus No: Ga. 1. Ja 3341

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Hari Chowk	7	-	-	-	-	-	
2	Baglung Bus Park	7	2	9	5	-	-	9.08
3	NEA Office, Malepatan	4	6	9	10	-	-	9.17
4	Zero KM	6	6	3	10	-	-	3.17
5	Srijana Chowk	1	3	1	25	-	-	1.42
6	Prithvi Chowk	6	7	6	25	4	50	6.42
7	WR Hospital	-	7	10	45	-	-	15.58
			Sum	38	120	4	50	44.83
			Total	44	50			

Table A1. 37 Travel time of Harichowk-WR Hospital route

Date: 2074/10/01 (Monday) Time: 11:03 am

Route: Western Regional Hospital-Hari Chowk Bus No: Ga. 1. Ja 3341

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	WR Hospital	12	-	-	-	-	-	
2	Prithvi Chowk	10	8	10	31	4	17	10.52
3	Srijana Chowk	4	2	1	52	2	48	6.15
4	Zero KM	3	1	1	53	5	26	4.68
5	NEA Office, Malepatan	2	2	3	26	-	-	8.87
6	Baglung Bus Park	2	5	6	15	-	-	6.25
7	Hari Chowk	-	15	4	23	-	-	4.38
			Sum	25	200	11	91	40.85
			Total	40	51			

Table A1. 38 Travel time of WR Hospital-Harichowk route

Date: 2074/10/01 (Monday) Time: 11:56 am

Route: Hari Chowk-Chauthe Bus No: Ga. 1. Ja 4556

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Hari Chowk	5	-	-	-	-	-	
2	Baglung Bus Park	6	3	8	49	-	-	8.82
3	Chipledhunga	6	5	5	1	5	12	5.02
4	Prithvi Chowk	5	4	7	15	4	58	12.45
5	Amarsingh Chowk	3	4	6	39	2	12	11.62
6	Chauthe	-	9	8	39	-	-	10.85
			Sum	34	143	11	82	48.75
			Total	48	45			

Table A1. 39 Travel time of Harichowk-Chauthe route

Date: 2074/10/01 (Monday) Time: 1:05 pm

Route: Chauthe-Hari Chowk Bus No: Ga. 1. Ja 4556

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chauthe	5	-	-	-	-	-	
2	Amarsingh Chowk	8	4	9	15	4	9	9.25
3	Prithvi Chowk	5	6	7	46	3	21	11.92
4	Chipledhunga	7	5	7	36	4	35	10.95
5	Baglung Bus Park	4	4	4	22	-	-	8.95
6	Hari Chowk	-	10	4	23	-	-	4.38
			Sum	31	142	11	65	45.45
			Total	45	27			

Table A1. 40 Travel time of Chauthe-Harichowk route

Date: 2074/10/02 (Tuesday) Time: 10:32 am
Route: Hari Chowk-Chorepatan Bus No: Ga. 1. Ja 6324

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Hari Chowk	8	-	-	-	-	-	
2	Baglung Bus Park	4	3	9	13	-	-	9.22
3	Chipledhunga	9	7	4	6	4	56	4.10
4	Prithvi Chowk	3	4	6	10	4	12	11.10
5	Srijana Chowk	2	1	1	19	3	56	5.52
6	Rastra Bank Chowk	1	4	3	35	-	-	7.52
7	Birauta	2	5	2	16	-	-	2.27
8	Chorepatan	-	5	4	42	-	-	4.70
			Sum	29	141	11	124	44.42
			Total	44	25			

Table A1. 41 Travel time of Harichowk-Chorepatan route

Date: 2074/10/02 (Tuesday) Time: 11:22 am
Route: Chorepatan-Hari Chowk Bus No: Ga. 1. Ja 6324

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chorepatan	6	-	-	-	-	-	
2	Birauta	2	0	5	2	3	10	5.03
3	Rastra Bank Chowk	4	4	2	26	-	-	5.60
4	Srijana Chowk	0	3	3	24	1	48	3.40
5	Prithvi Chowk	6	4	1	5	5	5	2.88
6	Chipledhunga	7	5	6	57	6	2	12.03
7	Baglung Bus Park	5	6	3	36	-	-	9.63
8	Hari Chowk	-	8	4	25	-	-	4.42
			Sum	24	175	15	65	43.00
			Total	43	0			

Table A1. 42 Travel time of Chorepatan-Harichowk route

Date: 2074/10/14 (Sunday) Time: 9:42 am

Route: Chipledhunga-Hallanchowk Bus No: Ga. 1. Ja 4539

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chipledhunga	7	-	-	-	-	-	
2	Prithvi Chowk	7	5	7	13	-	-	7.22
3	Srijana Chowk	3	2	2	18	4	36	2.30
4	Zero KM	4	3	3	9	3	12	7.75
5	Hallan Chowk	-	11	4	39	-	-	7.85
			Sum	16	79	7	48	25.12
			Total	25	7			

Table A1. 43 Travel time of Chipledhunga-Hallanchowk route

Date: 2074/10/14 (Sunday) Time: 10:23 am

Route: Hallanchowk-Chipledhunga Bus No: Ga. 1. Ja 4539

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Hallan Chowk	6	-	-	-	-	-	
2	Zero Km	4	3	7	36	3	58	7.60
3	Srijana Chowk	3	2	2	26	-	-	6.40
4	Prithvi Chowk	7	5	2	4	4	29	2.07
5	Chipledhunga	-	10	7	21	-	-	11.83
			Sum	18	87	7	87	27.90
			Total	27	54			

Table A1. 44 Travel time of Hallanchowk-Chipledhunga route

Date: 2074/10/14 (Sunday) Time: 11:55 am

Route: Chipledhunga-Lakeside Bus No: Ga. 1. Ja 6633

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chipledhunga	7	-	-	-	-	-	
2	Prithvi Chowk	6	4	5	7	5	32	5.12
3	Birauta	7	7	6	59	4	46	12.52
4	Rastrabank Chowk	1	2	3	21	-	-	8.12
5	Lakeside	-	8	4	54	-	-	4.90
			Sum	18	141	9	78	30.65
			Total	30	39			

Table A1. 45 Travel time of Chipledhunga-Lakeside route

Date: 2074/10/14 (Sunday) Time: 12:38 pm

Route: Lakeside-Chipledhunga Bus No: Ga. 1. Ja 6633

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Lakeside	4	-	-	-	-	-	
2	Rastrabank Chowk	2	1	8	2	-	-	8.03
3	Birauta	6	2	2	24	4	12	2.40
4	Prithvi Chowk	7	4	9	24	4	29	13.60
5	Chipledhunga	-	12	6	19	-	-	10.80
			Sum	25	69	8	41	34.83
			Total	34	50			

Table A1. 46 Travel time of Lakeside-Chipledhunga route

Date: 2074/10/15 (Monday) Time: 10:58 am

Route: Chipledhunga-Tutunga Bus No: Ga. 1. Ja 3339

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Chipledhunga	6	-	-	-	-	-	
2	Prithvi Chowk	8	4	9	13	6	12	9.22
3	Saint Mary's School	3	2	13	45	-	-	19.95
4	Ban Campus	2	3	3	56	-	-	3.93
5	Tutunga	-	10	12	42	-	-	12.70
			Sum	37	156	6	12	45.80
			Total	45	48			

Table A1. 47 Travel time of Chipledhunga-Tutunga route

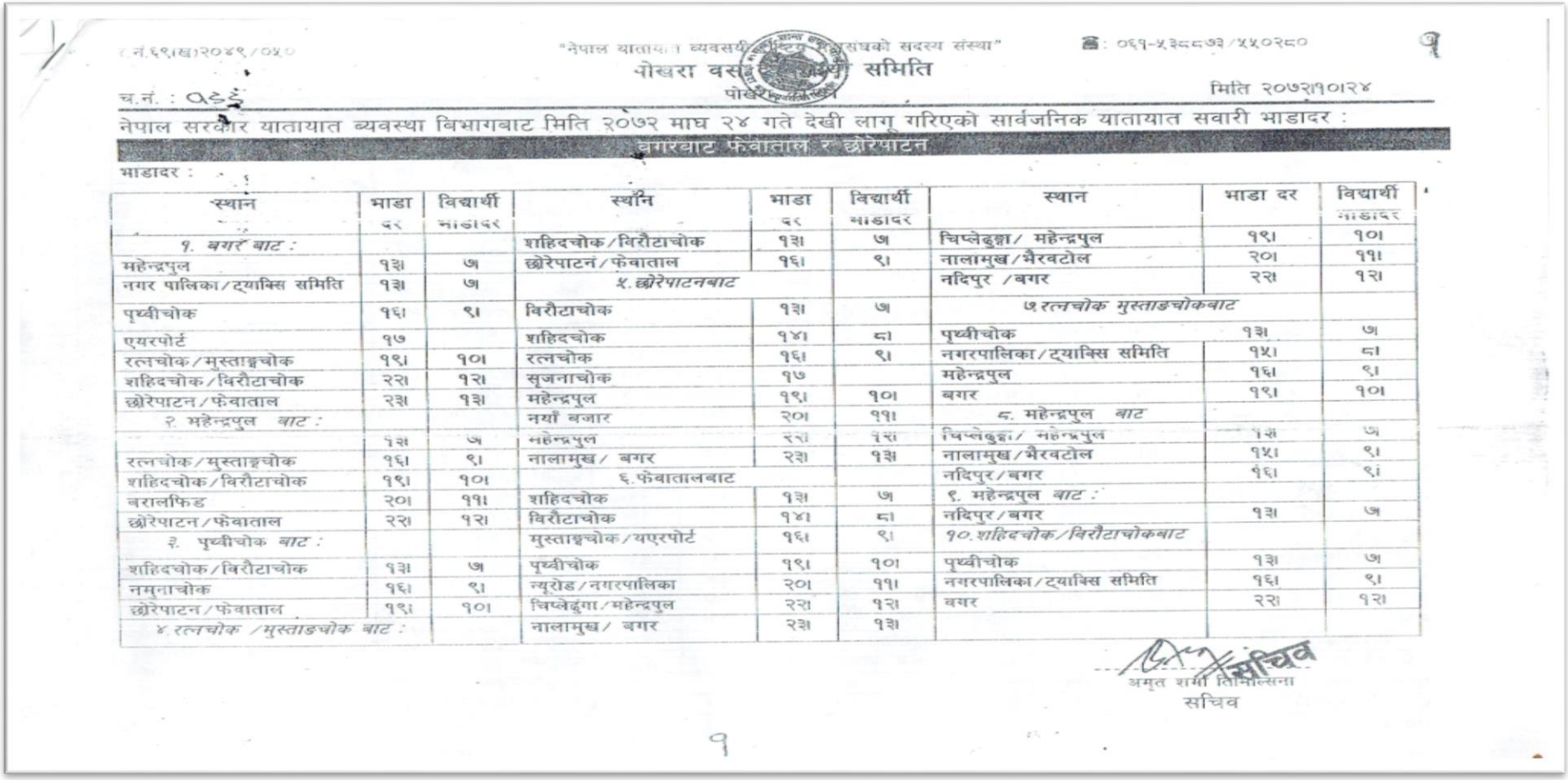
Date: 2074/10/15 (Monday) Time: 11:22 am

Route: Tutunga-Chipledhunga Bus No: Ga. 1. Ja 3339

S. N.	Stop	No. of Passengers		Travel Time		Waiting Time		Total Travel Time
		Entering	Leaving	min	sec	min	sec	min
1	Tutunga	4	-	-	-	-	-	
2	Ban Campus	3	1	13	2	4	12	13.03
3	Saint Mary's School	4	2	4	21	-	-	8.55
5	Prithvi Chowk	11	8	17	8	4	38	17.13
6	Chipledhunga	-	11	5	2	-	-	9.67
			Sum	39	33	8	50	48.38
			Total	48	23			

Table A1. 48 Travel time of Tutunga-Chipledhunga route

Travel Cost of Pokhara City Bus Service



"नेपाल यातायात व्यवसायी [illegible] संघको सदस्य संस्था"

पोखरा वस [illegible] समिति

फोन: ०६१-५३८८७३/५५०२८०

च.नं. : ०६६

मिति २०७२।१०।२४

नेपाल सरकार यातायात व्यवस्था विभागबाट मिति २०७२ माघ २४ गते देखी लागू गरिएको सार्वजनिक यातायात सवारी भाडादर :

बगरबाट फेवाताल र छोरेपाटन

भाडादर :

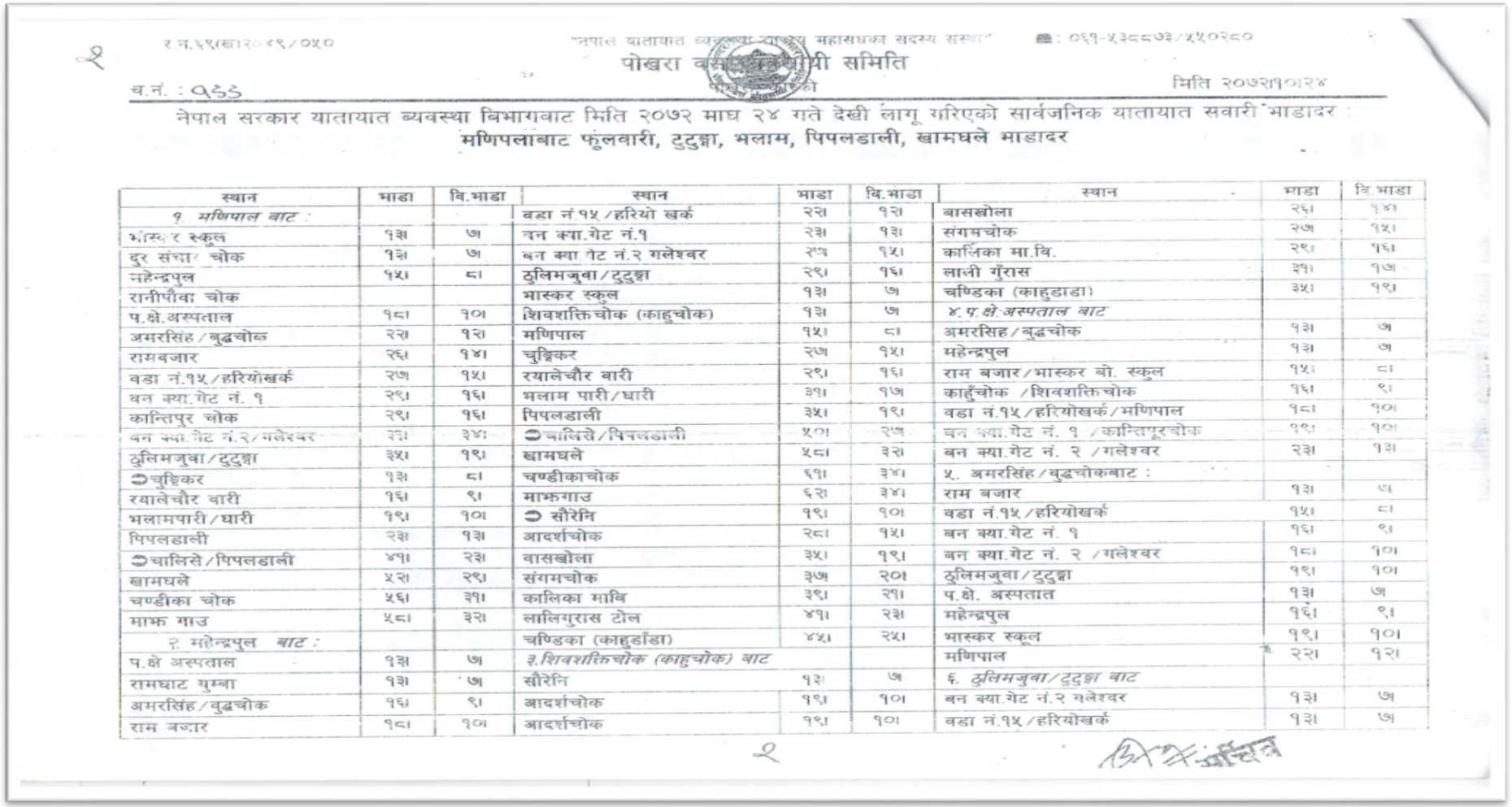
स्थान	भाडा दर	विद्यार्थी भाडादर	स्थान	भाडा दर	विद्यार्थी भाडादर	स्थान	भाडा दर	विद्यार्थी भाडादर
१. बगर बाट :			शहिदचोक/विरौटाचोक	१३।	७।	चिप्लेढुङ्गा/ महेन्द्रपुल	१९।	१०।
महेन्द्रपुल	१३।	७।	छोरेपाटन/फेवाताल	१६।	९।	नालामुख/भैरवटोल	२०।	११।
नगर पालिका/ट्याक्सि समिति	१३।	७।	५.छोरेपाटनबाट			नदिपुर /बगर	२२।	१२।
पृथ्वीचोक	१६।	९।	विरौटाचोक	१३।	७।	७.रत्नचोक मुस्ताङचोकबाट		
एयरपोर्ट	१७		शहिदचोक	१४।	८।	पृथ्वीचोक	१३।	७।
रत्नचोक/मुस्ताङचोक	१९।	१०।	रत्नचोक	१६।	९।	नगरपालिका/ट्याक्सि समिति	१५।	८।
शहिदचोक/विरौटाचोक	२२।	१२।	सृजनाचोक	१७		महेन्द्रपुल	१६।	९।
छोरेपाटन/फेवाताल	२३।	१३।	महेन्द्रपुल	१९।	१०।	बगर	१९।	१०।
२. महेन्द्रपुल बाट :			नयाँ बजार	२०।	११।	८. महेन्द्रपुल बाट		
	१३।	७।	महेन्द्रपुल	२२।	१२।	चिप्लेढुङ्गा/ महेन्द्रपुल	१३।	७।
रत्नचोक/मुस्ताङचोक	१६।	९।	नालामुख/ बगर	२३।	१३।	नालामुख/भैरवटोल	१५।	९।
शहिदचोक/विरौटाचोक	१९।	१०।	६.फेवातालबाट			नदिपुर/बगर	१६।	९।
बरालफिड	२०।	११।	शहिदचोक	१३।	७।	९. महेन्द्रपुल बाट :		
छोरेपाटन/फेवाताल	२२।	१२।	विरौटाचोक	१४।	८।	नदिपुर/बगर	१३।	७।
३. पृथ्वीचोक बाट :			मुस्ताङचोक/यएरपोर्ट	१६।	९।	१०.शहिदचोक/विरौटाचोकबाट		
शहिदचोक/विरौटाचोक	१३।	७।	पृथ्वीचोक	१९।	१०।	पृथ्वीचोक	१३।	७।
नम्नाचोक	१६।	९।	न्यूरोड/नगरपालिका	२०।	११।	नगरपालिका/ट्याक्सि समिति	१६।	९।
छोरेपाटन/फेवाताल	१९।	१०।	चिप्लेढुंगा/महेन्द्रपुल	२२।	१२।	बगर	२२।	१२।
४.रत्नचोक /मुस्ताङचोक बाट :			नालामुख/ बगर	२३।	१३।			

सचिव

अमृत शर्मा तिमिल्सिना

सचिव

Figure A1. 1 Travel cost of Bagar-Lakeside and Bagar-Chorepatan route



र न.६९/का२०६९/०५०

"नेपाल यातायात व्यवस्था [illegible] महासंघका सदस्य संस्था" ☎ : ०६१-५३८८७३/५५०२८०

पोखरा [illegible] समिति

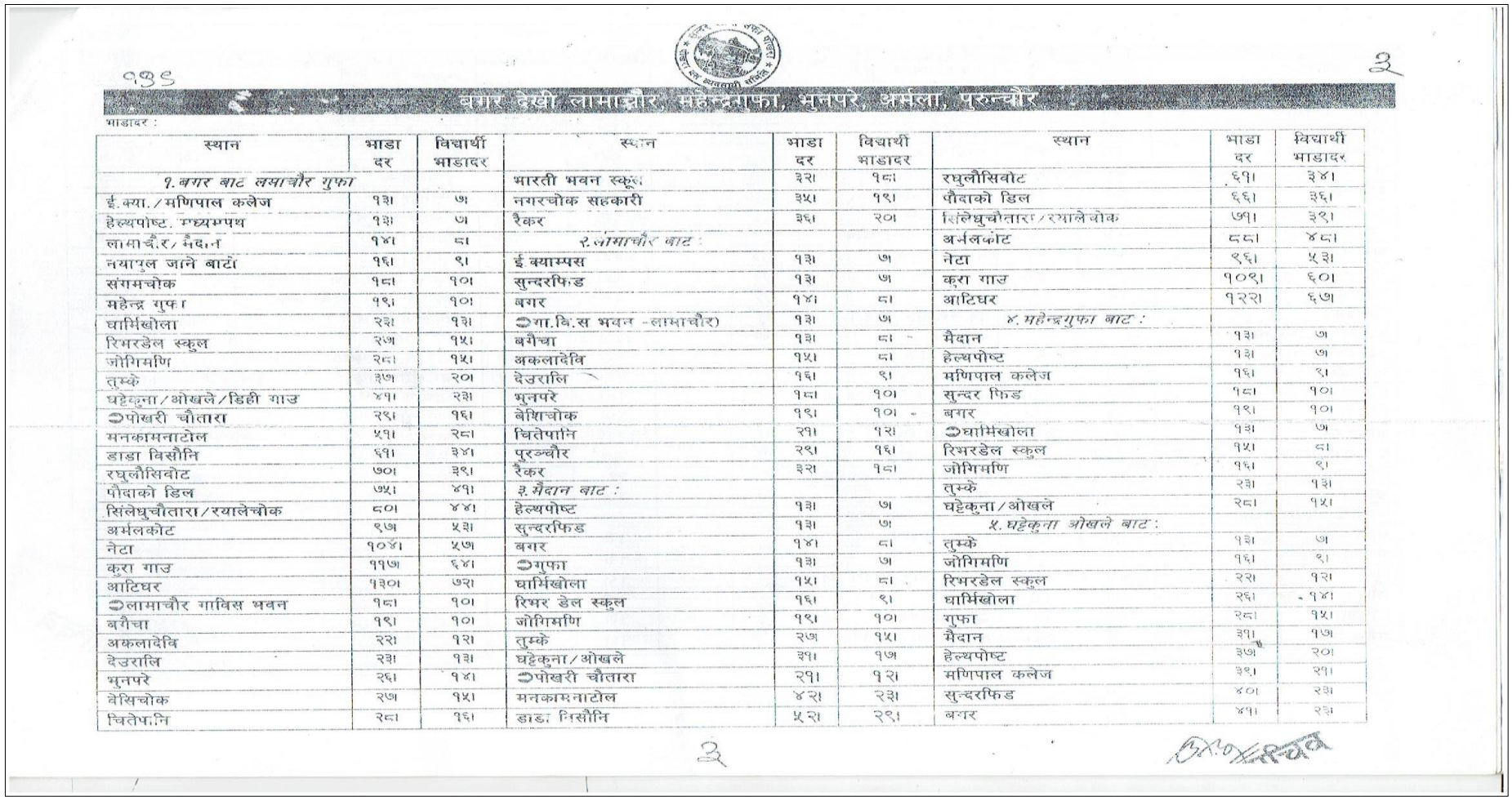
च.नं. : ९३६ मिति २०७२।१०।२४

नेपाल सरकार यातायात व्यवस्था बिभागबाट मिति २०७२ माघ २४ गते देखी लागू गरिएको सार्वजनिक यातायात सवारी भाडादर :

मणिपलाबाट फुलवारी, टुटुङ्गा, भलाम, पिपलडाली, खामघले भाडादर

स्थान	भाडा	वि.भाडा	स्थान	भाडा	वि.भाडा	स्थान	भाडा	वि.भाडा
१. मणिपाल बाट :			वडा नं.१५/हरियो खर्क	२२।	१२।	बासखोला	२६।	१४।
भास्कर स्कुल	१३।	७।	बन क्या.गेट नं.१	२३।	१३।	संगमचोक	२७।	१५।
दुर संचार चोक	१३।	७।	बन क्या.गेट नं.२ गलेश्वर	२७।	१५।	कालिका मा.वि.	२९।	१६।
महेन्द्रपुल	१५।	८।	ठुलिमजुवा/टुटुङ्गा	२९।	१६।	लाली गुँरास	३१।	१७।
रानीपौवा चोक			भास्कर स्कुल	१३।	७।	चण्डिका (काहुडाडा)	३५।	१९।
प.क्षे.अस्पताल	१८।	१०।	शिवशक्तिचोक (काहुचोक)	१३।	७।	*४.प.क्षे.अस्पताल बाट*		
अमरसिंह/बुद्धचोक	२२।	१२।	मणिपाल	१५।	८।	अमरसिंह/बुद्धचोक	१३।	७।
रामबजार	२६।	१४।	चुङ्गिकर	२७।	१५।	महेन्द्रपुल	१३।	७।
वडा नं.१५/हरियोखर्क	२७।	१५।	रयालेचौर बारी	२९।	१६।	राम बजार/भास्कर बो. स्कुल	१५।	८।
बन क्या.गेट नं. १	२९।	१६।	भलाम पारी/घारी	३१।	१७।	काहुँचोक /शिवशक्तिचोक	१६।	९।
कान्तिपुर चोक	२९।	१६।	पिपलडाली	३५।	१९।	वडा नं.१५/हरियोखर्क/मणिपाल	१८।	१०।
बन क्या.गेट नं.२/गलेश्वर	३१।	३४।	➲चालिसे/पिपलडाली	५०।	२७।	बन क्या.गेट नं. १ /कान्तिपूरचोक	१९।	१०।
ठुलिमजुवा/टुटुङ्गा	३५।	१९।	खामघले	५८।	३२।	बन क्या.गेट नं. २ /गलेश्वर	२३।	१३।
➲चुङ्गिकर	१३।	८।	चण्डीकाचोक	६१।	३४।	५. अमरसिंह/बुद्धचोकबाट :		
रयालेचौर बारी	१६।	९।	माझगाउ	६२।	३४।	राम बजार	१३।	७।
भलामपारी/घारी	१९।	१०।	➲ सौरेनि	१९।	१०।	वडा नं.१५/हरियोखर्क	१५।	८।
पिपलडाली	२३।	१३।	आदर्शचोक	२८।	१५।	बन क्या.गेट नं. १	१६।	९।
➲चालिसे/पिपलडाली	४१।	२३।	बासखोला	३५।	१९।	बन क्या.गेट नं. २ /गलेश्वर	१८।	१०।
खामघले	५२।	२९।	संगमचोक	३७।	२०।	ठुलिमजुवा/टुटुङ्गा	१९।	१०।
चण्डीका चोक	५६।	३१।	कालिका माबि	३९।	२१।	प.क्षे. अस्पताल	१३।	७।
माझ गाउ	५८।	३२।	लालिगुरास टोल	४१।	२३।	महेन्द्रपुल	१६।	९।
२. महेन्द्रपुल बाट :			चण्डिका (काहुडाँडा)	४५।	२५।	भास्कर स्कूल	१९।	१०।
प.क्षे अस्पताल	१३।	७।	*३.शिवशक्तिचोक (काहुचोक) बाट*			मणिपाल	२२।	१२।
रामघाट गुम्बा	१३।	७।	सौरेनि	१३।	७।	*६. ठुलिमजुवा/टुटुङ्गा बाट*		
अमरसिंह/बुद्धचोक	१६।	९।	आदर्शचोक	१९।	१०।	बन क्या.गेट नं.२ गलेश्वर	१३।	७।
राम बजार	१८।	१०।	आदर्शचोक	१९।	१०।	वडा नं.१५/हरियोखर्क	१३।	७।

Figure A1. 2 Travel cost of Manipal-Tutunga route

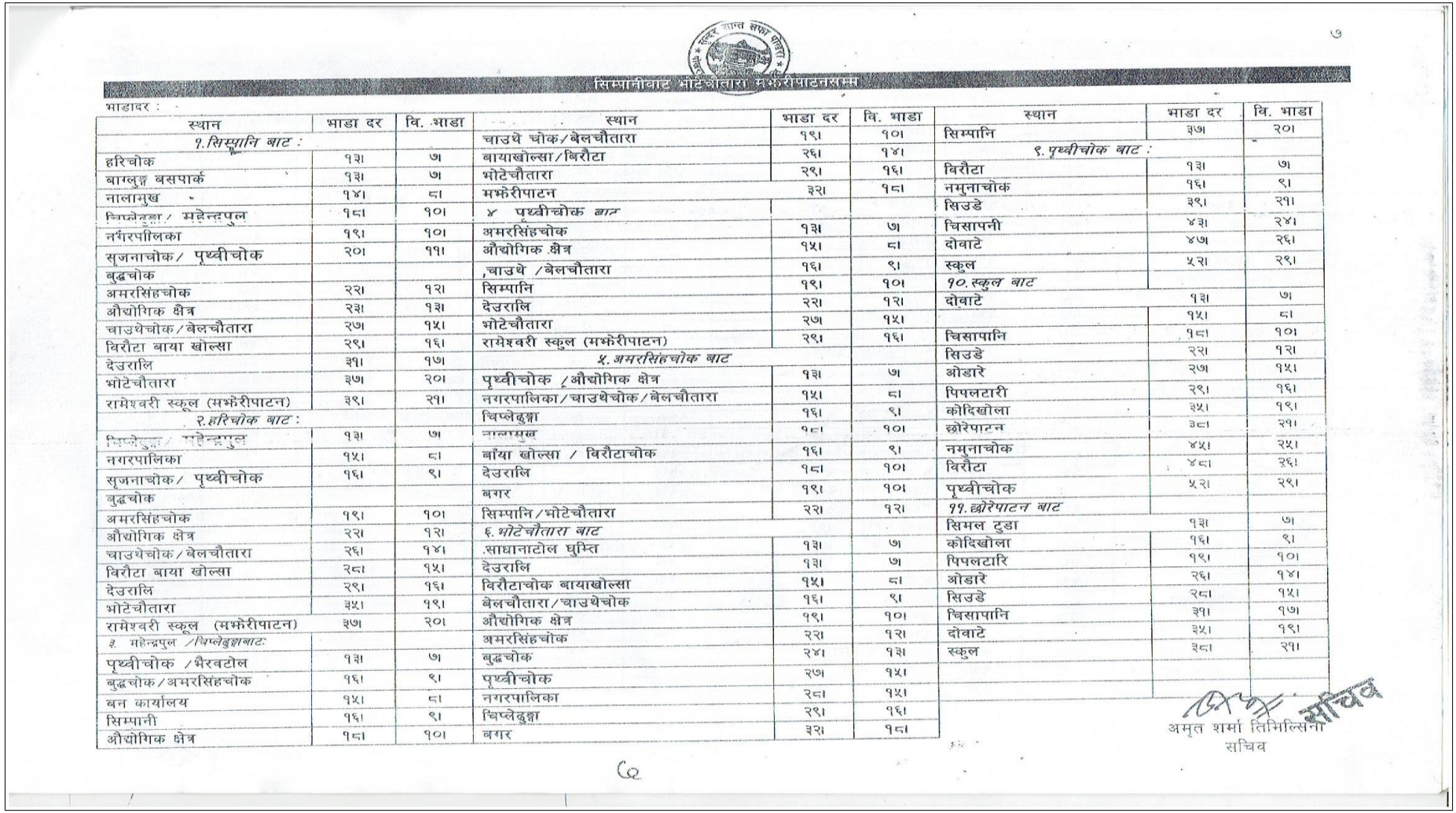


बगर देखी लामाचौर, महेन्द्रगुफा, भुनपरे, अर्मला, पुरन्चौर

भाडादर :

स्थान	भाडा दर	विद्यार्थी भाडादर	स्थान	भाडा दर	विद्यार्थी भाडादर	स्थान	भाडा दर	विद्यार्थी भाडादर
१.बगर बाट लमाचौर गुफा			भारती भवन स्कूल	३२।	१८।	रघुलौसिवोट	६१।	३४।
ई.क्या./मणिपाल कलेज	१३।	७।	नगरचोक सहकारी	३५।	१९।	पौदाको डिल	६६।	३६।
हेल्थपोष्ट, मध्यमपथ	१३।	७।	रैकर	३६।	२०।	सिलेधुचौतारा/रयालेचोक	७१।	३९।
लामाचौर/मैदान	१४।	८।	*२.लामाचौर बाट :*			अर्मलकोट	८८।	४८।
नयापुल जाने बाटो	१६।	९।	ई.क्याम्पस	१३।	७।	नेटा	९६।	५३।
संगमचोक	१८।	१०।	सुन्दरफिड	१३।	७।	कुरा गाउ	१०९।	६०।
महेन्द्र गुफा	१९।	१०।	बगर	१४।	८।	आटिघर	१२२।	६७।
घार्मिखोला	२३।	१३।	⊃गा.वि.स भवन -लामाचौर)	१३।	७।	*४.महेन्द्रगुफा बाट :*		
रिभरडेल स्कुल	२७।	१५।	बगैचा	१३।	८।	मैदान	१३।	७।
जोगिमणि	२८।	१५।	अकलादेवि	१५।	८।	हेल्थपोष्ट	१३।	७।
तुम्के	३७।	२०।	देउरालि	१६।	९।	मणिपाल कलेज	१६।	९।
घट्टेकुना/ओखले/डिही गाउ	४१।	२३।	भुनपरे	१८।	१०।	सुन्दर फिड	१८।	१०।
⊃पोखरी चौतारा	२९।	१६।	बेसिचोक	१९।	१०।	बगर	१९।	१०।
मनकामनाटोल	५१।	२८।	चितेपानि	२१।	१२।	⊃घार्मिखोला	१३।	७।
डाडा सिसौनि	६१।	३४।	पुरन्चौर	२९।	१६।	रिभरडेल स्कुल	१५।	८।
रघुलौसिवोट	७०।	३९।	रैकर	३२।	१८।	जोगिमणि	१६।	९।
पौदाको डिल	७५।	४१।	*३.मैदान बाट :*			तुम्के	२३।	१३।
सिलेधुचौतारा/रयालेचोक	८०।	४४।	हेल्थपोष्ट	१३।	७।	घट्टेकुना/ओखले	२८।	१५।
अर्मलकोट	९७।	५३।	सुन्दरफिड	१३।	७।	*५.घट्टेकुना ओखले बाट :*		
नेटा	१०४।	५७।	बगर	१४।	८।	तुम्के	१३।	७।
कुरा गाउ	११७।	६४।	⊃गुफा	१३।	७।	जोगिमणि	१६।	९।
आटिघर	१३०।	७२।	घार्मिखोला	१५।	८।	रिभरडेल स्कुल	२२।	१२।
⊃लामाचौर गाविस भवन	१८।	१०।	रिभर डेल स्कुल	१६।	९।	घार्मिखोला	२६।	१४।
बगैचा	१९।	१०।	जोगिमणि	१९।	१०।	गुफा	२८।	१५।
अकलादेवि	२२।	१२।	तुम्के	२७।	१५।	मैदान	३१।	१७।
देउरालि	२३।	१३।	घट्टेकुना/ओखले	३१।	१७।	हेल्थपोष्ट	३७।	२०।
भुनपरे	२६।	१४।	⊃पोखरी चौतारा	२१।	१२।	मणिपाल कलेज	३९।	२१।
बेसिचोक	२७।	१५।	मनकामनाटोल	४२।	२३।	सुन्दरफिड	४०।	२३।
चितेपानि	२८।	१६।	डाडा सिसौनि	५२।	२९।	बगर	४१।	२३।

Figure A1. 3 Travel cost of Bagar-Lamachaur, Bagar-Gufaand Bagar-Bhunpare route



सिम्पानीबाट भोटेचौतारा मझेरीपाटनसम्म

भाडादर :

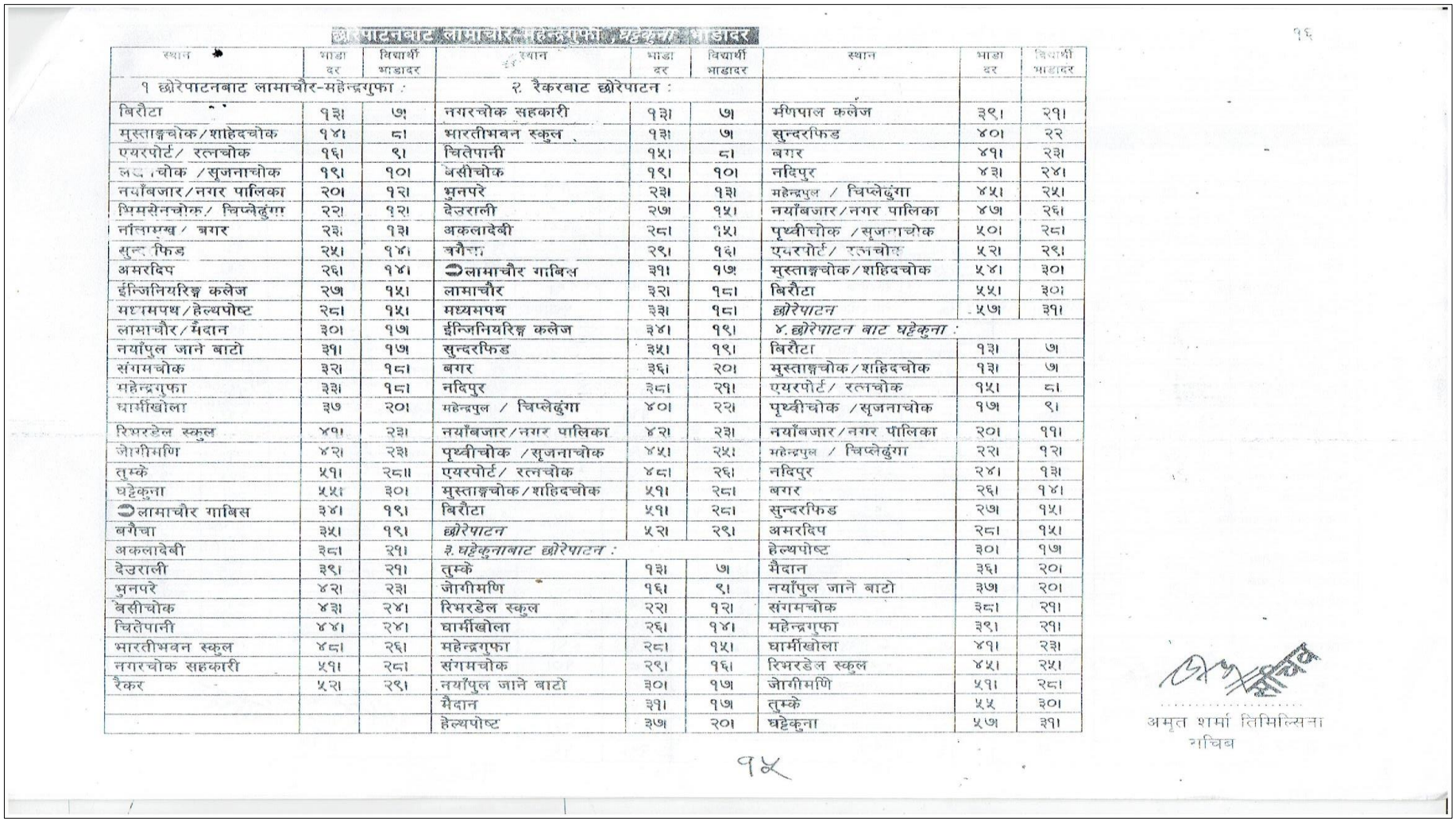
स्थान	भाडा दर	वि. भाडा
१.सिम्पानि बाट :		
हरिचोक	१३।	७।
बारलुङ्ग बसपार्क	१३।	७।
नालामुख	१४।	८।
चिप्लेढुङ्गा / महेन्द्रपुल	१८।	१०।
नगरपालिका	१९।	१०।
सृजनाचोक/ पृथ्वीचोक	२०।	११।
बुद्धचोक		
अमरसिंहचोक	२२।	१२।
औद्योगिक क्षेत्र	२३।	१३।
चाउथेचोक/बेलचौतारा	२७।	१५।
विरौटा बाया खोल्सा	२९।	१६।
देउरालि	३१।	१७।
भोटेचौतारा	३७।	२०।
रामेश्वरी स्कूल (मझेरीपाटन)	३९।	२१।
२.हरिचोक बाट :		
चिप्लेढुङ्गा / महेन्द्रपुल	१३।	७।
नगरपालिका	१५।	८।
सृजनाचोक/ पृथ्वीचोक	१६।	९।
बुद्धचोक		
अमरसिंहचोक	१९।	१०।
औद्योगिक क्षेत्र	२२।	१२।
चाउथेचोक/बेलचौतारा	२६।	१४।
विरौटा बाया खोल्सा	२८।	१५।
देउरालि	२९।	१६।
भोटेचौतारा	३५।	१९।
रामेश्वरी स्कूल (मझेरीपाटन)	३७।	२०।
३. महेन्द्रपुल /चिप्लेढुङ्गाबाट:		
पृथ्वीचोक /भैरवटोल	१३।	७।
बुद्धचोक/अमरसिंहचोक	१६।	९।
बन कार्यालय	१५।	८।
सिम्पानी	१६।	९।
औद्योगिक क्षेत्र	१८।	१०।

स्थान	भाडा दर	वि. भाडा
चाउथे चोक/बेलचौतारा	१९।	१०।
बायाखोल्सा/विरौटा	२६।	१४।
भोटेचौतारा	२९।	१६।
मझेरीपाटन	३२।	१८।
४ पृथ्वीचोक बाट		
अमरसिंहचोक	१३।	७।
औद्योगिक क्षेत्र	१५।	८।
,चाउथे /बेलचौतारा	१६।	९।
सिम्पानि	१९।	१०।
देउरालि	२२।	१२।
भोटेचौतारा	२७।	१५।
रामेश्वरी स्कूल (मझेरीपाटन)	२९।	१६।
५. अमरसिंहचोक बाट		
पृथ्वीचोक /औद्योगिक क्षेत्र	१३।	७।
नगरपालिका/चाउथेचोक/बेलचौतारा	१५।	८।
चिप्लेढुङ्गा	१६।	९।
नालामुख	१८।	१०।
बाँया खोल्सा / विरौटाचोक	१६।	९।
देउरालि	१८।	१०।
बगर	१९।	१०।
सिम्पानि/भोटेचौतारा	२२।	१२।
६.भोटेचौतारा बाट		
साधानाटोल घुम्ति	१३।	७।
देउरालि	१३।	७।
विरौटाचोक बायाखोल्सा	१५।	८।
बेलचौतारा/चाउथेचोक	१६।	९।
औद्योगिक क्षेत्र	१९।	१०।
अमरसिंहचोक	२२।	१२।
बुद्धचोक	२४।	१३।
पृथ्वीचोक	२७।	१५।
नगरपालिका	२८।	१५।
चिप्लेढुङ्गा	२९।	१६।
बगर	३२।	१८।

स्थान	भाडा दर	वि. भाडा
सिम्पानि	३७।	२०।
९. पृथ्वीचोक बाट :		
विरौटा	१३।	७।
नमुनाचोक	१६।	९।
सिउडे	३९।	२१।
चिसापनी	४३।	२४।
दोवाटे	४७।	२६।
स्कुल	५२।	२९।
१०.स्कुल बाट		
दोवाटे	१३।	७।
	१५।	८।
चिसापानि	१८।	१०।
सिउडे	२२।	१२।
ओडारे	२७।	१५।
पिपलटारी	२९।	१६।
कोदिखोला	३५।	१९।
छोरेपाटन	३८।	२१।
नमुनाचोक	४५।	२५।
विरौटा	४८।	२६।
पृथ्वीचोक	५२।	२९।
११.छोरेपाटन बाट		
सिमल टुडा	१३।	७।
कोदिखोला	१६।	९।
पिपलटारि	१९।	१०।
ओडारे	२६।	१४।
सिउडे	२८।	१५।
चिसापानि	३१।	१७।
दोवाटे	३५।	१९।
स्कुल	३८।	२१।

अमृत शर्मा तिमिल्सिना
सचिव

Figure A1. 4 Travel cost of Simpani- Majheripatan route



छोरेपाटनबाट लामाचौर-महेन्द्रगुफा :

स्थान	भाडा दर	विद्यार्थी भाडादर
बिरौटा	१३।	७।
मुस्ताङ्गचोक/शाहेदचोक	१४।	८।
एयरपोर्ट/ रत्नचोक	१६।	९।
लक्ष्मीचोक /सुजनाचोक	१९।	१०।
नयाँबजार/नगर पालिका	२०।	१२।
भिमसेनचोक/ चिप्लेढुंगा	२२।	१२।
नौलाख्ख / बगर	२३।	१३।
सुन्दर फिड	२५।	१४।
अमरदिप	२६।	१४।
ईन्जिनियरिङ्ग कलेज	२७।	१५।
मध्यमपथ/हेल्थपोष्ट	२८।	१५।
लामाचौर/मैदान	३०।	१७।
नयाँपुल जाने बाटो	३१।	१७।
संगमचोक	३२।	१८।
महेन्द्रगुफा	३३।	१८।
घार्मीखोला	३७	२०।
रिभरडेल स्कुल	४१।	२३।
जोगीमणि	४२।	२३।
तुम्के	५१।	२८॥
घट्टेकुना	५५।	३०।
लामाचौर गाबिस	३४।	१९।
बगैचा	३५।	१९।
अकलादेबी	३८।	२१।
देउराली	३९।	२१।
भुनपरे	४२।	२३।
बेसीचोक	४३।	२४।
चितेपानी	४४।	२४।
भारतीभवन स्कुल	४८।	२६।
नगरचोक सहकारी	५१।	२८।
रैकर	५२।	२९।

२. रैकरबाट छोरेपाटन :

स्थान	भाडा दर	विद्यार्थी भाडादर
नगरचोक सहकारी	१३।	७।
भारतीभवन स्कुल	१३।	७।
चितेपानी	१५।	८।
बेसीचोक	१९।	१०।
भुनपरे	२३।	१३।
देउराली	२७	१५।
अकलादेबी	२८।	१५।
बगैचा	२९।	१६।
गाबिस लामाचौर	३१।	१७
लामाचौर	३२।	१८।
मध्यमपथ	३३।	१८।
ईन्जिनियरिङ्ग कलेज	३४।	१९।
सुन्दरफिड	३५।	१९।
बगर	३६।	२०।
नदिपुर	३८।	२१।
महेन्द्रपुल / चिप्लेढुंगा	४०।	२२।
नयाँबजार/नगर पालिका	४२।	२३।
पृथ्वीचोक /सुजनाचोक	४५।	२५।
एयरपोर्ट/ रत्नचोक	४८।	२६।
मुस्ताङ्गचोक/शाहिदचोक	५१।	२८।
बिरौटा	५१।	२८।
छोरेपाटन	५२।	२९।

३. घट्टेकुनाबाट छोरेपाटन :

स्थान	भाडा दर	विद्यार्थी भाडादर
तुम्के	१३।	७।
जोगीमणि	१६।	९।
रिभरडेल स्कुल	२२।	१२।
घार्मीखोला	२६।	१४।
महेन्द्रगुफा	२८।	१५।
संगमचोक	२९।	१६।
नयाँपुल जाने बाटो	३०।	१७।
मैदान	३१।	१७।
हेल्थपोष्ट	३७।	२०।

स्थान	भाडा दर	विद्यार्थी भाडादर
मणिपाल कलेज	३९।	२१।
सुन्दरफिड	४०।	२२
बगर	४१।	२३।
नदिपुर	४३।	२४।
महेन्द्रपुल / चिप्लेढुंगा	४५।	२५।
नयाँबजार/नगर पालिका	४७।	२६।
पृथ्वीचोक /सुजनाचोक	५०।	२८।
एयरपोर्ट/ रत्नचोक	५२।	२९।
मुस्ताङ्गचोक/शहिदचोक	५४।	३०।
बिरौटा	५५।	३०।
छोरेपाटन	५७।	३१।

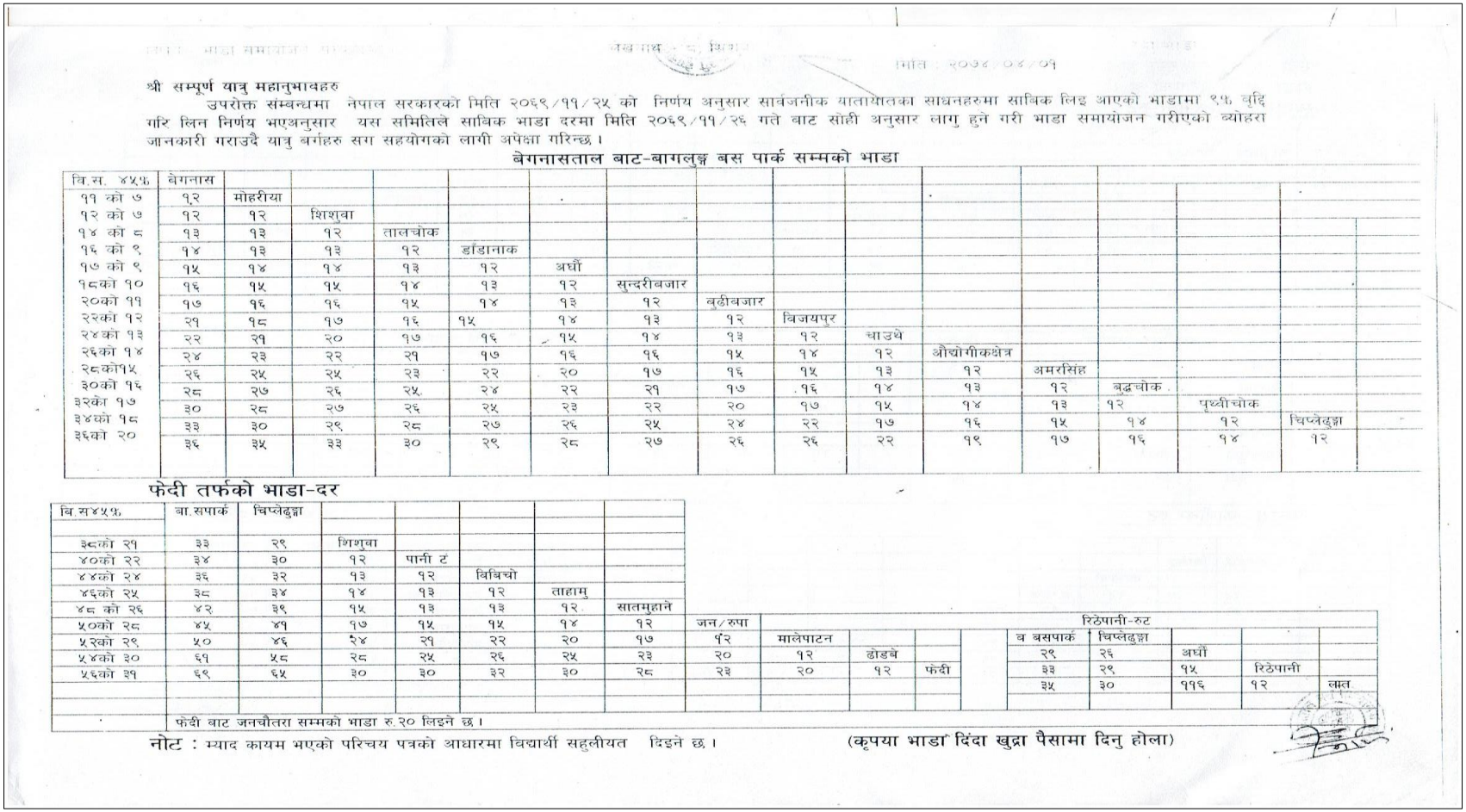
४. छोरेपाटन बाट घट्टेकुना :

स्थान	भाडा दर	विद्यार्थी भाडादर
बिरौटा	१३।	७।
मुस्ताङ्गचोक/शहिदचोक	१३।	७।
एयरपोर्ट/ रत्नचोक	१५।	८।
पृथ्वीचोक /सुजनाचोक	१७।	९।
नयाँबजार/नगर पालिका	२०।	११।
महेन्द्रपुल / चिप्लेढुंगा	२२।	१२।
नदिपुर	२४।	१३।
बगर	२६।	१४।
सुन्दरफिड	२७।	१५।
अमरदिप	२८।	१५।
हेल्थपोष्ट	३०।	१७।
मैदान	३६।	२०।
नयाँपुल जाने बाटो	३७।	२०।
संगमचोक	३८।	२१।
महेन्द्रगुफा	३९।	२१।
घार्मीखोला	४१।	२३।
रिभरडेल स्कुल	४५।	२५।
जोगीमणि	५१।	२८।
तुम्के	५५	३०।
घट्टेकुना	५७	३१।

अमृत शर्मा तिमिल्सिना
सचिव

Figure A1. 5 Travel cost of Chorepatan-Ghattekuna and Chorepatan-Raikar route

Travel Cost of Lekhnath City Bus Service



श्री सम्पूर्ण यात्रु महानुभावहरु

उपरोक्त सम्बन्धमा नेपाल सरकारको मिति २०६९/११/२५ को निर्णय अनुसार सार्वजनीक यातायातका साधनहरुमा साबिक लिइ आएको भाडामा ९% वृद्दि गरि लिन निर्णय भएअनुसार यस समितिले साबिक भाडा दरमा मिति २०६९/११/२६ गते बाट सोही अनुसार लागु हुने गरी भाडा समायोजन गरीएको व्योहरा जानकारी गराउदै यात्रु बर्गहरु सग सहयोगको लागी अपेक्षा गरिन्छ ।

बेगनासताल बाट-बागलुङ्ग बस पार्क सम्मको भाडा

वि.स. ४५%	बेगनास															
११ को ७	१२	मोहरीया														
१२ को ७	१२	१२	शिशुवा													
१४ को ८	१३	१३	१२	तालचोक												
१६ को ९	१४	१३	१३	१२	डाँडानाक											
१७ को ९	१५	१४	१४	१३	१२	अर्घौ										
१८को १०	१६	१५	१५	१४	१३	१२	सुन्दरीबजार									
२०को ११	१७	१६	१६	१५	१४	१३	१२	बुढीबजार								
२२को १२	२१	१८	१७	१६	१५	१४	१३	१२	बिजयपुर							
२४को १३	२२	२१	२०	१७	१६	१५	१४	१३	१२	चाउथे						
२६को १४	२४	२३	२२	२१	१७	१६	१६	१५	१४	१२	औद्योगीकक्षेत्र					
२८को१५	२६	२५	२५	२३	२२	२०	१७	१६	१५	१३	१२	अमरसिंह				
३०को १६	२८	२७	२६	२५	२४	२२	२१	१७	१६	१४	१३	१२	बुद्धचोक			
३२को १७	३०	२८	२७	२६	२५	२३	२२	२०	१७	१५	१४	१३	१२	पृथ्वीचोक		
३४को १८	३३	३०	२९	२८	२७	२६	२५	२४	२२	१७	१६	१५	१४	१२	चिप्लेढुङ्गा	
३६को २०	३६	३५	३३	३०	२९	२८	२७	२६	२६	२२	१९	१७	१६	१४	१२	

फेदी तर्फको भाडा-दर

वि.स४५%	बा.सपार्क	चिप्लेढुङ्गा														
३८को २१	३३	२९	शिशुवा													
४०को २२	३४	३०	१२	पानी ट												
४४को २४	३६	३२	१३	१२	बिचिचो											
४६को २५	३८	३४	१४	१३	१२	ताहामु										
४८ को २६	४२	३९	१५	१३	१३	१२	सातमुहाने									
५०को २८	४५	४१	१७	१५	१५	१४	१२	जन/रुपा	मालेपाटन				ब बसपार्क	चिप्लेढुङ्गा		
५२को २९	५०	४६	२४	२१	२२	२०	१७	२०	१२	ढोडबे			२९	२६	अर्घौ	
५४को ३०	६१	५८	२८	२५	२६	२५	२३	२३	२०	१२	फेदी		३३	२९	१५	रिठेपानी
५६को ३१	६९	६५	३०	३०	३२	३०	२८	२३					३५	३०	११९	१२

रिठेपानी-रुट

फेदी बाट जनचौतरा सम्मको भाडा रु.२० लिइने छ ।

नोट : म्याद कायम भएको परिचय पत्रको आधारमा विद्यार्थी सहुलीयत दिइने छ ।

(कृपया भाडा दिंदा खुद्रा पैसामा दिनु होला)

Figure A1. 6 Travel cost of Baglung Buspark-Begnas Tal, Phedi, Rupa Tal and Rithepani route

Travel Cost of Taxi Micro Bus Service

ट्याक्सी व्यवसाय समिति पोखराको नगर माईक्रोबस भाडा तालिका-२०७३/०९/२७ देखि लागू हुने

हरिचोकबाट

	पुरा भाडा	सहुलियत
बास्तुङ्गबसपार्क	१३	७
प्रस्याङ्ग	१५	८
चिप्लेढुङ्गा	१६	९
नगरपालिका	१८	९
पृथ्वीचोक	२१	१२
मालेपाटन	१७	९
जिरो.कि.मी	१९	१०
बुद्धचोक	२३	१३
गण्डकी अस्पताल	२४	१३
अमरसिंहचोक	२४	१३
शिवचोक	२४	१३
औद्योगिकक्षेत्र	२५	१४
चाउथे	२६	१४
सूचनाचोक	२१	१२
रत्नचोक	२४	१३
राष्ट्रबैकचोक	२५	१४
बिरौटाचोक	२६	१४
बरालफिड	२७	१५
छोरेपाटन	२८	१५

राष्ट्रबैकचोकबाट

	पुरा भाडा	सहुलियत
बिरौटाचोक	१३	८
बरालफिड	१४	८
छोरेपाटन	१४	८

चिप्लेढुङ्गाबाट

	पुरा भाडा	सहुलियत
पृथ्वीचोक	१३	७
सृजनाचोक	१५	८
रत्नचोक	१८	९
राष्ट्रबैकचोक	१९	१०
बिरौटाचोक	२१	१२
बरालफिड	२३	१३
छोरेपाटन	२४	१३
बुद्धचोक	१७	९
अमरसिंहचोक	१८	९
शिवचोक	१९	१०
औद्योगिकक्षेत्र	२३	१३
चाउथे	२४	१३
बास्तुङ्गबसपार्क	१३	८
विन्ध्यवासिनी	१५	८
हरिचोक	१५	८
जौबारी	१६	९

छोरेपाटनबाट

	पुरा भाडा	सहुलियत
बिरौटाचोक	१३	७
राष्ट्रबैकचोक	१४	८
रत्नचोक	१६	९
पृथ्वीचोक	१९	१०
नगरपालिका	२१	१२
चिप्लेढुङ्गा	२४	१३
प्रस्याङ्ग	२४	१३
बास्तुङ्गबसपार्क	२६	१४
विन्ध्यवासिनी	२८	१५
हरिचोक	२८	१५

पृथ्वीचोकबाट

	पुरा भाडा	सहुलिय
चिप्लेढुङ्गा	१३	७
प्रस्याङ्ग	१७	९
बास्तुबसपार्क	१८	१०
विन्ध्यवासिनी	२१	११
हरिचोक	२१	११
रत्नचोक	१३	८
राष्ट्रबैकचोक	१४	८
बिरौटाचोक	१५	८
बरालफिड	१७	९
छोरेपाटन	१९	१०
सृजनाचोक	१३	७
जिरो.कि.मी	१३	७
मालेपाटन	१५	८
बुद्धचोक	१३	७
अमरसिंहचोक	१४	८
शिवचोक	१५	८
औद्योगिकक्षेत्र	१७	९
चाउथे	१९	१०
गण्डकी अस्पताल	१५	८

जिरो.कि.मी.बाट

	पुरा भाडा	सहुलियत
पृथ्वीचोक	१३	७
बुद्धचोक	१८	९
गण्डकी अस्पताल	१९	१०
मालेपाटन	१३	७
प्रस्याङ्ग	१४	८
बास्तुबसपार्क	१६	९
विन्ध्यवासिनी	१९	१०
हरिचोक	१९	१०

बिरौटाचोकबाट

	पुरा भाडा	सहुलियत
राष्ट्रबैकचोक	१३	७
रत्नचोक	१३	७
सृजनाचोक	१४	८
पृथ्वीचोक	१५	८
नगरपालिका	१९	१०
चिप्लेढुङ्गा	२१	१२
प्रस्याङ्ग	२३	१३
बास्तुङ्गबसपार्क	२४	१३
विन्ध्यवासिनी	२६	१४
हरिचोक	२६	१४

गण्डकी अस्पतालबाट

	पुरा भाडा	सहुलियत
बुद्धचोक	१३	७
पृथ्वीचोक	१५	८
सृजनाचोक	१७	९
जिरो.किमी	१९	१०
मालेपाटन	२२	१२
प्रस्याङ्ग	२३	१३
बास्तुङ्गबसपार्क	२४	१३
विन्ध्यवासिनी	२५	१४
हरिचोक	२५	१४

अमरसिंहचोकबाट

	पुरा भाडा	सहुलियत
बुद्धचोक	१३	७
पृथ्वीचोक	१४	८
नगरपालिका	१६	९
चिप्लेढुङ्गा	१८	९
बास्तुङ्गबसपार्क	२३	१३
विन्ध्यवासिनी	२४	१३
हरिचोक	२४	१३
जौबारी	२५	१४
औद्योगिकक्षेत्र	१३	७
चाउथे	१५	९

चाउथेबाट

	पुरा भाडा	सहुलियत
औद्योगिकक्षेत्र	१३	७
अमरसिंहचोक	१५	८
बुद्धचोक	१८	९
पृथ्वीचोक	१९	१०
नगरपालिका	२१	११
चिप्लेढुङ्गा	२४	१३
बास्तुङ्गबसपार्क	२५	१४
विन्ध्यवासिनी	२६	१४
हरिचोक	२६	१४
जौबारी	२९	१६

जौबारीबाट

	पुरा भाडा	सहुलियत
हरिचोक	१३	७
विन्ध्यवासिनी	१४	८
बास्तुङ्गबसपार्क	१५	८
चिप्लेढुङ्गा	१६	९
नगरपालिका	१९	१०
पृथ्वीचोक	२३	१३
बुद्धचोक	२४	१३
अमरसिंहचोक	२५	१४
शिवचोक	२५	१४
औद्योगिकक्षेत्र	२६	१४
चाउथे	२९	१६

ट्याक्सी व्यवसाय समिति

युवराज पटेल

नोट: विद्यार्थी, जेष्ठनागरिकलेनका असक्त धाईते, जेष्ठ नागरिक र अपाङ्गहरुलाई परिचय पत्रको आधारमा ४५ प्रतिशत सहुलियतको भाडादर लागू हुने छ यदि कसैले नक्कली परिचय पत्र देखाई सहुलियत मागेमा आवश्यक कारवाही गरिने छ।

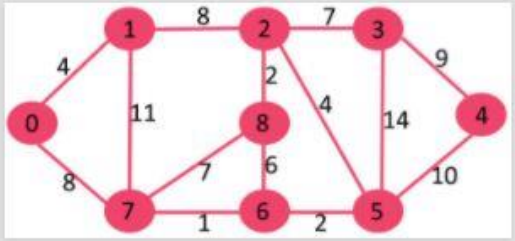
सम्पर्क नं. ०६१-५२१४४७, ५२४०११, ५२६८९०, ५२६८९१

Figure A1. 7 Travel cost of Hari Chowk-WR Hospital, Chorepatan, Chauthe route

ANNEX 2. PSEUDOCODE FOR ALGORITHM

Pseudocode for Dijkstra's Algorithm using C++ (example)

For example, consider below graph and source as 0,



Output should be

Vertex	Distance	Path
0 -> 1	4	0 1
0 -> 2	12	0 1 2
0 -> 3	19	0 1 2 3
0 -> 4	21	0 7 6 5 4
0 -> 5	11	0 7 6 5
0 -> 6	9	0 7 6
0 -> 7	8	0 7
0 -> 8	14	0 1 2 8

```
// A C / C++ program for Dijkstra's single source shortest
// path algorithm. The program is for adjacency matrix
// representation of the graph.
#include <stdio.h>
#include <limits.h>

// Number of vertices in the graph
#define V 9

// A utility function to find the vertex with minimum distance
// value, from the set of vertices not yet included in shortest
// path tree
int minDistance(int dist[], bool sptSet[])
{
    // Initialize min value
    int min = INT_MAX, min_index;

    for (int v = 0; v < V; v++)
        if (sptSet[v] == false && dist[v] <= min)
            min = dist[v], min_index = v;

    return min_index;
}

// Function to print shortest path from source to j
// using parent array
void printPath(int parent[], int j)
{
    // Base Case : If j is source
    if (parent[j]==-1)
        return;

    printPath(parent, parent[j]);

    printf("%d ", j);
}
```

```
// A utility function to print the constructed distance
// array
int printSolution(int dist[], int n, int parent[])
{
    int src = 0;
    printf("Vertex\t  Distance\tPath");
    for (int i = 1; i < V; i++)
    {
        printf("\n%d -> %d \t\t %d\t\t%d ", src, i, dist[i], src);
        printPath(parent, i);
    }
}

// Funtion that implements Dijkstra's single source shortest path
// algorithm for a graph represented using adjacency matrix
// representation
void dijkstra(int graph[V][V], int src)
{
    int dist[V];  // The output array. dist[i] will hold
                  // the shortest distance from src to i

    // sptSet[i] will true if vertex i is included / in shortest
    // path tree or shortest distance from src to i is finalized
    bool sptSet[V];

    // Parent array to store shortest path tree
    int parent[V];

    // Initialize all distances as INFINITE and stpSet[] as false
    for (int i = 0; i < V; i++)
    {
        parent[0] = -1;
        dist[i] = INT_MAX;
        sptSet[i] = false;
    }

    // Distance of source vertex from itself is always 0
    dist[src] = 0;

    // Find shortest path for all vertices
    for (int count = 0; count < V-1; count++)
    {
        // Pick the minimum distance vertex from the set of
        // vertices not yet processed. u is always equal to src
        // in first iteration.
        int u = minDistance(dist, sptSet);

        // Mark the picked vertex as processed
        sptSet[u] = true;

        // Update dist value of the adjacent vertices of the
        // picked vertex.
        for (int v = 0; v < V; v++)

            // Update dist[v] only if is not in sptSet, there is
            // an edge from u to v, and total weight of path from
            // src to v through u is smaller than current value of
            // dist[v]
            if (!sptSet[v] && graph[u][v] &&
                dist[u] + graph[u][v] < dist[v])
            {
                parent[v]  = u;
                dist[v] = dist[u] + graph[u][v];
            }
    }

    // print the constructed distance array
    printSolution(dist, V, parent);
}

// driver program to test above function
int main()
{
    /* Let us create the example graph discussed above */
    int graph[V][V] = {{0, 4, 0, 0, 0, 0, 0, 8, 0},
```

```
                        {4, 0, 8, 0, 0, 0, 0, 11, 0},
                        {0, 8, 0, 7, 0, 4, 0, 0, 2},
                        {0, 0, 7, 0, 9, 14, 0, 0, 0},
                        {0, 0, 0, 9, 0, 10, 0, 0, 0},
                        {0, 0, 4, 0, 10, 0, 2, 0, 0},
                        {0, 0, 0, 14, 0, 2, 0, 1, 6},
                        {8, 11, 0, 0, 0, 0, 1, 0, 7},
                        {0, 0, 2, 0, 0, 0, 6, 7, 0}
                       };

    dijkstra(graph, 0);

    return 0;
}
```

(GeeksforGeeks, n.d.)

Pseudocode for Dijkstra's Algorithm with printing path using C++ (for Min. travel time of Pokhara Bus Service Route Node 0 as starting node)

```
// C++ program for Dijkstra's single
// source shortest path algorithm.
// The program is for adjacency matrix
// representation of the graph.
#include <iostream>
#include <climits>
using namespace std;

// Number of vertices
// in the graph
#define V 52

// A utility function to find the
// vertex with minimum distance
// value, from the set of vertices
// not yet included in shortest
// path tree
int minDistance(double dist[],
                                        bool sptSet[])
{

        // Initialize min value
        int min = INT_MAX, min_index;

        for (int v = 0; v < V; v++)
                if (sptSet[v] == false &&
```

```
                                        dist[v] <= min)
                                    min = dist[v], min_index = v;
        return min_index;
}

// Function to print shortest
// path from source to j
// using parent array
void printPath(int parent[], int j)
{

        // Base Case : If j is source
        if (parent[j] == - 1)
                return;

        printPath(parent, parent[j]);

        printf("%d ", j);
}

// A utility function to print
// the constructed distance
// array
void printSolution(double dist[], int n,
                                                int parent[])
{
        int src = 0;
        printf("Vertex\t Travel Time\tPath");
        for (int i = 1; i < V; i++)
        {
                printf("\n%d -> %d \t\t %f\t\t%d ",
                                                src, i, dist[i], src);
                printPath(parent, i);
        }
}

// Funtion that implements Dijkstra's
// single source shortest path
// algorithm for a graph represented
```

```
// using adjacency matrix representation
void dijkstra(double graph[V][V], int src)
{

        // The output array. dist[i]
        // will hold the shortest
        // distance from src to i
        double dist[V];

        // sptSet[i] will true if vertex
        // i is included / in shortest
        // path tree or shortest distance
        // from src to i is finalized
        bool sptSet[V];

        // Parent array to store
        // shortest path tree
        int parent[V];

        // Initialize all distances as
        // INFINITE and stpSet[] as false
        for (int i = 0; i < V; i++)
        {
                parent[0] = -1;
                dist[i] = INT_MAX;
                sptSet[i] = false;
        }

        // Distance of source vertex
        // from itself is always 0
        dist[src] = 0;

        // Find shortest path
        // for all vertices
        for (int count = 0; count < V - 1; count++)
        {
                // Pick the minimum distance
                // vertex from the set of
                // vertices not yet processed.
```

```
                        // u is always equal to src
                        // in first iteration.
                        int u = minDistance(dist, sptSet);

                        // Mark the picked vertex
                        // as processed
                        sptSet[u] = true;

                        // Update dist value of the
                        // adjacent vertices of the
                        // picked vertex.
                        for (int v = 0; v < V; v++)

                                // Update dist[v] only if is
                                // not in sptSet, there is
                                // an edge from u to v, and
                                // total weight of path from
                                // src to v through u is smaller
                                // than current value of
                                // dist[v]
                                if (!sptSet[v] && graph[u][v] &&
                                        dist[u] + graph[u][v] < dist[v])
                                {
                                        parent[v] = u;
                                        dist[v] = dist[u] + graph[u][v];
                                }
        }

        // print the constructed
        // distance array
        printSolution(dist, V, parent);
}

// Driver Code
int main()
{
        // Let us create the example
        // graph discussed above
double graph[V][V] = {
```

{0,9.43,15.98,0,0,0,0,0,0,0,0,0,0,0,0,4.38,0,0,0,0,0,0,0,0,0,0,4.54,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{9.43,0,1.57,15.73,0,0,0,0,0,0,0,8,0,0,0,0,0,0,0,0,0,0,0,13.4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11.11,0,0,0},

{15.98,1.57,0,7.38,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9.17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,14.83,0,0,0,0,0},

{0,15.73,7.38,0,15.41,0,3.32,0,0,0,0,0,0,0,0,0,0,0,0,0,21.61,0,0,10.56,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,15.41,0,0,0,5,6.26,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10.32},

{0,0,0,0,0,0,0,6.04,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,3.32,0,0,0,4.76,0,0,2.25,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,5,6.04,4.76,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,6.26,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,5.69,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,2.25,0,0,5.69,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,8,0,0,0,0,0,0,0,0,0,7.54,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,7.54,0,6,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,6,0,7.14,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,7.14,0,5.88,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,5.88,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.62,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.62,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,9.17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,21.61,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.87,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.87,0,4.06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4.06,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,13.4,0,10.56,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.09,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.96,0,0,0,13.03,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.09,0,6.54,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.29,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6.54,0,6.17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6.17,0,5.27,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5.27,0,13.26,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13.26,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{4.54,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.75,0,0,0,0,0,0,9.78,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.75,0,3.37,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.37,0,9.72,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9.72,0,3.14,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.14,0,3.28,0,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.28,0,2.84,0,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2.84,0,7.04,0,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.04,0,6.51,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,6.51,0,0,0,0,0,0,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9.78,0,0,0,0,0,0,0,4.23,0,0,0,0,0},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4.23,0,9.8,0,0,0,0,0,0,0,0},

```
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,9.8,0,8,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,8,0,5.3,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5.3,0,4.4,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4.4,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.96,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.22,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3.22,0,5.66,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5.66,0,0,0,0,0,0,0},
{0,0,14.83,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,13.03,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,11.11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,7.29,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,12.47,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,12.47,0,0},
{0,0,0,0,10.32,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
        };

        dijkstra(graph, 0);
system("pause");
        return 0;
}
```

Using this code all other routes were optimized separately for time and cost impedance.

ANNEX 3. MATRIX FORMATION

The geographical network of public transport route connecting the interlinked nodes and the weights between them has to be converted into matrix form which actually is the input for pseudocode. The matrix has to be formed for both time and cost and for each routes served by service providers.

In case of Pokhara City Bus Service, it runs its bus service in 16 different routes with touching 52 nodes. Therefore, a square matrix of 52x52 has to be formed. The table below has shown the network with link weights (time) and destination nodes in matrix form of Pokhara City Bus Service for node 0. But, the commuter can start their journey from any node/bus stop so the matrix for each node (1-51) to be considered as starting point has to be formed which is quite a tedious job. Similarly, the matrix for weights as cost should also be formed for cost optimization.

.

Table A3. 1 1Network with link weights and destination nodes in matrix form of Pokhara City Bus Service for Node 0

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0.00	9.43	15.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.38	0.00	0.00	0.00
1	9.43	0.00	1.57	15.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	15.98	1.57	0.00	7.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.17	0.00
3	0.00	15.73	7.38	0.00	15.41	0.00	3.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	21.61
4	0.00	0.00	0.00	15.41	0.00	0.00	0.00	5.00	6.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	3.32	0.00	0.00	0.00	4.76	0.00	0.00	2.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	5.00	6.04	4.76	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	6.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	2.25	0.00	0.00	5.69	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.54	0.00	6.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.00	0.00	7.14	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.14	0.00	5.88	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.88	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.62	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	13.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	14.83	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.32
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.62	0.00	10.00	0.00
18	0.00	0.00	9.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00	0.00	0.00
19	0.00	0.00	0.00	21.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.87
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	13.40	0.00	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	4.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	3.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	4.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	4.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	7.09	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
23	0.00	0.00	7.09	0.00	6.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24	0.00	0.00	0.00	6.54	0.00	6.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	6.17	0.00	5.27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	5.27	0.00	13.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	13.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.78	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.75	0.00	3.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.37	0.00	9.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.72	0.00	3.14	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.14	0.00	3.28	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.28	0.00	2.84	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	0.00	7.96	0.00	0.00	0.00	13.03	0.00	0.00	0.00	0.00
23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.29	0.00	0.00
24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
33	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	14.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	11.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	10.32	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.84	0.00	7.04	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.04	0.00	6.51	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.51	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.78	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.23	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.23	0.00	9.80
39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	9.80	0.00
40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.00
41	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	7.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	13.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	7.29	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

	40	41	42	43	44	45	46	47	48	49	50	51
34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
39	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
40	0.00	5.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
41	5.30	0.00	4.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
42	0.00	4.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	3.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00
44	0.00	0.00	0.00	3.22	0.00	5.66	0.00	0.00	0.00	0.00	0.00	0.00
45	0.00	0.00	0.00	0.00	5.66	0.00	0.00	0.00	0.00	0.00	0.00	0.00
46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.47	0.00
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.47	0.00	0.00
51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

In case of Lekhnath City Bus Service, it runs its bus service in 5 different routes with touching 19 nodes. Therefore, a square matrix of 19x19 has to be formed. The table below has shown the network with link weights (time) and destination nodes in matrix form of Lekhnath City Bus Service for node 3 and 4, similarly the matrix for all other nodes (5, 18, 22, 52-64) and for cost too has to be formed which is quite a tedious job.

	3	2	4	5	18	22	52	53	54	55	56	57	58	59	60	61	62	63	64
3	0.00	9.01	7.54	0.00	0.00	9.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	9.01	0.00	0.00	0.00	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	7.54	0.00	0.00	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	9.55	0.00	0.00	0.00	0.00	0.00	11.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00	0.00	11.24	0.00	4.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
53	0.00	0.00	0.00	0.00	0.00	0.00	4.73	0.00	4.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.74	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.75
55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	4.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.85	0.00	4.89	7.53	0.00	0.00	0.00	0.00	0.00	0.00
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.53	0.00	0.00	4.55	0.00	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.55	0.00	4.50	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.50	0.00	6.34	5.27	0.00	0.00
61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.34	0.00	0.00	0.00	0.00
62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.27	0.00	0.00	14.58	0.00
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.58	0.00	0.00
64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40	8.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A3. 2 Network with link weights and destination nodes in matrix form of Lekhnath City Bus Service for Node 3

	4	2	3	5	18	22	52	53	54	55	56	57	58	59	60	61	62	63	64
4	0.00	0.00	7.54	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	9.01	0.00	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	7.54	9.01	0.00	0.00	0.00	9.55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	9.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	4.86	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
22	0.00	0.00	9.55	0.00	0.00	0.00	11.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
52	0.00	0.00	0.00	0.00	0.00	11.24	0.00	4.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
53	0.00	0.00	0.00	0.00	0.00	0.00	4.73	0.00	4.74	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40
54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.74	0.00	5.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	8.75
55	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.00	0.00	4.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.85	0.00	4.89	7.53	0.00	0.00	0.00	0.00	0.00	0.00
57	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.89	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
58	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.53	0.00	0.00	4.55	0.00	0.00	0.00	0.00	0.00
59	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.55	0.00	4.50	0.00	0.00	0.00	0.00
60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.50	0.00	6.34	5.27	0.00	0.00
61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.34	0.00	0.00	0.00	0.00
62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.27	0.00	0.00	14.58	0.00
63	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.58	0.00	0.00
64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	15.40	8.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table A3. 3 Network with link weights and destination nodes in matrix form of Lekhnath City Bus Service for Node 4

In case of City Micro Bus Service, it runs its bus service in 6 different routes with touching 18 nodes. Therefore, a square matrix of 18x18 has to be formed. The table below has shown the network with link weights (time) and destination nodes in matrix form of City Micro Bus Service for node 3 and 43, similarly the matrix for all other nodes (4-10, 17, 18, 22, 49, 50, 65-67) and for cost too has to be formed.

	3	2	4	5	6	7	8	9	10	17	18	22	43	49	50	65	66	67
3	0.00	10.04	13.06	0.00	4.22	0.00	0.00	0.00	0.00	0.00	0.00	11.77	0.00	0.00	0.00	0.00	13.05	18.54
2	10.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	13.06	0.00	0.00	0.00	0.00	4.60	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	6.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	4.22	0.00	0.00	0.00	0.00	5.46	0.00	0.00	5.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	4.60	6.47	5.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	5.06	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	6.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	0.00	7.71	0.00	0.00
22	11.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.05	0.00	0.00	0.00	0.00	0.00
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.05	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	6.24
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	0.00	0.00
65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	7.71	0.00	0.00	0.00	0.00	0.00	0.00	0.00
66	13.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
67	18.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.24	0.00	0.00	0.00	0.00

Table A3. 4 Network with link weights and destination nodes in matrix form of City Micro Bus Service for Node 3

	43	2	3	4	5	6	7	8	9	10	17	18	22	49	50	65	66	67
43	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.05	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	10.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.93	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	10.04	0.00	13.06	0.00	4.22	0.00	0.00	0.00	0.00	0.00	0.00	11.77	0.00	0.00	0.00	13.05	18.54
4	0.00	0.00	13.06	0.00	0.00	0.00	4.60	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	6.47	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
6	0.00	0.00	4.22	0.00	0.00	0.00	5.46	0.00	0.00	5.06	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	4.60	6.47	5.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	4.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	5.06	0.00	0.00	7.73	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	0.00
17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	0.00	0.00
18	0.00	6.93	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.72	0.00	0.00	0.00	0.00	7.71	0.00	0.00
22	10.05	0.00	11.77	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	6.24
50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.87	0.00	0.00	0.00	0.00
65	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.02	0.00	7.71	0.00	0.00	0.00	0.00	0.00	0.00
66	0.00	0.00	13.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
67	0.00	0.00	18.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	6.24	0.00	0.00	0.00	0.00

Table A3. 5 Network with link weights and destination nodes in matrix form of City Micro Bus Service for Node 43

ANNEX 4. OUTPUT OF PSEUDOCODE

For optimization based on time:

1. Pokhara City Bus Service:

Result along with printing path:

From Node 1 (Mahendrapool):

```
Vertex          Travel Time          Path
1 -> 0          9.430000             1 0
1 -> 2          1.570000             1 2
1 -> 3          8.950000             1 2 3
1 -> 4          22.030000            1 2 3 6 7 4
1 -> 5          23.070000            1 2 3 6 7 5
1 -> 6          12.270000            1 2 3 6
1 -> 7          17.030000            1 2 3 6 7
1 -> 8          28.290000            1 2 3 6 7 4 8
1 -> 9          20.210000            1 2 3 6 10 9
1 -> 10         14.520000            1 2 3 6 10
1 -> 11         8.000000             1 11
1 -> 12         15.540000            1 11 12
1 -> 13         21.540000            1 11 12 13
1 -> 14         28.680000            1 11 12 13 14
1 -> 15         34.560000            1 11 12 13 14 15
1 -> 16         13.810000            1 0 16
1 -> 17         16.430000            1 0 16 17
1 -> 18         10.740000            1 2 18
1 -> 19         30.560000            1 2 3 19
1 -> 20         34.430000            1 2 3 19 20
1 -> 21         38.490000            1 2 3 19 20 21
1 -> 22         13.400000            1 22
1 -> 23         20.490000            1 22 23
1 -> 24         27.030000            1 22 23 24
1 -> 25         33.200000            1 22 23 24 25
1 -> 26         38.470000            1 22 23 24 25 26
```

```
1 -> 27          51.730000            1 22 23 24 25 26 27
1 -> 28          13.970000            1 0 28
1 -> 29          17.720000            1 0 28 29
1 -> 30          21.090000            1 0 28 29 30
1 -> 31          30.810000            1 0 28 29 30 31
1 -> 32          33.950000            1 0 28 29 30 31 32
1 -> 33          37.230000            1 0 28 29 30 31 32 33
1 -> 34          40.070000            1 0 28 29 30 31 32 33 34
1 -> 35          47.110000            1 0 28 29 30 31 32 33 34 35
1 -> 36          53.620000            1 0 28 29 30 31 32 33 34 35 36
1 -> 37          23.750000            1 0 28 37
1 -> 38          27.980000            1 0 28 37 38
1 -> 39          37.780000            1 0 28 37 38 39
1 -> 40          45.780000            1 0 28 37 38 39 40
1 -> 41          51.080000            1 0 28 37 38 39 40 41
1 -> 42          55.480000            1 0 28 37 38 39 40 41 42
1 -> 43          21.360000            1 22 43
1 -> 44          24.580000            1 22 43 44
1 -> 45          30.240000            1 22 43 44 45
1 -> 46          16.400000            1 2 46
1 -> 47          26.430000            1 22 47
1 -> 48          11.110000            1 48
1 -> 49          27.780000            1 22 23 49
1 -> 50          40.250000            1 22 23 49 50
1 -> 51          32.350000            1 2 3 6 7 4 51
```

From Node 5 (Lakeside):

```
Vertex          Travel Time          Path
5 -> 0          32.500000            5 7 6 3 2 1 0
5 -> 1          23.070000            5 7 6 3 2 1
5 -> 2          21.500000            5 7 6 3 2
5 -> 3          14.120000            5 7 6 3
5 -> 4          11.040000            5 7 4
5 -> 6          10.800000            5 7 6
5 -> 7          6.040000             5 7
5 -> 8          17.300000            5 7 4 8
5 -> 9          18.740000            5 7 6 10 9
5 -> 10         13.050000            5 7 6 10
5 -> 11         31.070000            5 7 6 3 2 1 11
5 -> 12         38.610000            5 7 6 3 2 1 11 12
5 -> 13         44.610000            5 7 6 3 2 1 11 12 13
5 -> 14         51.750000            5 7 6 3 2 1 11 12 13 14
5 -> 15         57.630000            5 7 6 3 2 1 11 12 13 14 15
5 -> 16         36.880000            5 7 6 3 2 1 0 16
5 -> 17         39.500000            5 7 6 3 2 1 0 16 17
5 -> 18         30.670000            5 7 6 3 2 18
5 -> 19         35.730000            5 7 6 3 19
5 -> 20         39.600000            5 7 6 3 19 20
5 -> 21         43.660000            5 7 6 3 19 20 21
5 -> 22         24.680000            5 7 6 3 22
5 -> 23         31.770000            5 7 6 3 22 23
5 -> 24         38.310000            5 7 6 3 22 23 24
5 -> 25         44.480000            5 7 6 3 22 23 24 25
5 -> 26         49.750000            5 7 6 3 22 23 24 25 26
5 -> 27         63.010000            5 7 6 3 22 23 24 25 26 27
5 -> 28         37.040000            5 7 6 3 2 1 0 28
5 -> 29         40.790000            5 7 6 3 2 1 0 28 29
5 -> 30         44.160000            5 7 6 3 2 1 0 28 29 30
5 -> 31         53.880000            5 7 6 3 2 1 0 28 29 30 31
5 -> 32         57.020000            5 7 6 3 2 1 0 28 29 30 31 32
```

```
5 -> 33          60.300000          5 7 6 3 2 1 0 28 29 30 31 32 33
5 -> 34          63.140000          5 7 6 3 2 1 0 28 29 30 31 32 33
                                    34
5 -> 35          70.180000          5 7 6 3 2 1 0 28 29 30 31 32 33
                                    34 35
5 -> 36          76.690000          5 7 6 3 2 1 0 28 29 30 31 32 33
                                    34 35 36
5 -> 37          46.820000          5 7 6 3 2 1 0 28 37
5 -> 38          51.050000          5 7 6 3 2 1 0 28 37 38
5 -> 39          60.850000          5 7 6 3 2 1 0 28 37 38 39
5 -> 40          68.850000          5 7 6 3 2 1 0 28 37 38 39 40
5 -> 41          74.150000          5 7 6 3 2 1 0 28 37 38 39 40 41
5 -> 42          78.550000          5 7 6 3 2 1 0 28 37 38 39 40 41
                                    42
5 -> 43          32.640000          5 7 6 3 22 43
5 -> 44          35.860000          5 7 6 3 22 43 44
5 -> 45          41.520000          5 7 6 3 22 43 44 45
5 -> 46          36.330000          5 7 6 3 2 46
5 -> 47          37.710000          5 7 6 3 22 47
5 -> 48          34.180000          5 7 6 3 2 1 48
5 -> 49          39.060000          5 7 6 3 22 23 49
5 -> 50          51.530000          5 7 6 3 22 23 49 50
5 -> 51          21.360000          5 7 4 51
```

2. Lekhnath City Bus Service:

From Node 3 (Prithvi Chowk):

Vertex	Travel Time	Path
3 -> 2	9.010000	3 2
3 -> 4	7.540000	3 4
3 -> 5	17.020000	3 4 5
3 -> 18	13.870000	3 2 18
3 -> 22	9.550000	3 22
3 -> 52	20.790000	3 22 52
3 -> 53	25.520000	3 22 52 53
3 -> 54	30.260000	3 22 52 53 54
3 -> 55	35.260000	3 22 52 53 54 55
3 -> 56	40.110000	3 22 52 53 54 55 56
3 -> 57	45.000000	3 22 52 53 54 55 56 57
3 -> 58	47.640000	3 22 52 53 54 55 56 58
3 -> 59	52.190000	3 22 52 53 54 55 56 58 59
3 -> 60	56.690000	3 22 52 53 54 55 56 58 59 60
3 -> 61	63.030000	3 22 52 53 54 55 56 58 59 60 61
3 -> 62	61.960000	3 22 52 53 54 55 56 58 59 60 62
3 -> 63	76.540000	3 22 52 53 54 55 56 58 59 60 62 63
3 -> 64	39.010000	3 22 52 53 54 64

From Node 4 (Birauta):

Vertex	Travel Time	Path
4 -> 2	16.550000	4 3 2
4 -> 3	7.540000	4 3
4 -> 5	9.480000	4 5
4 -> 18	21.410000	4 3 1 18
4 -> 22	17.090000	4 3 22
4 -> 52	28.330000	4 3 22 52
4 -> 53	33.060000	4 3 22 52 53

```
4 -> 54          37.800000          4 3 22 52 53 54
4 -> 55          42.800000          4 3 22 52 53 54 55
4 -> 56          47.650000          4 3 22 52 53 54 55 56
4 -> 57          52.540000          4 3 22 52 53 54 55 56 57
4 -> 58          55.180000          4 3 22 52 53 54 55 56 58
4 -> 59          59.730000          4 3 22 52 53 54 55 56 58 59
4 -> 60          64.230000          4 3 22 52 53 54 55 56 58 59 60
4 -> 61          70.570000          4 3 22 52 53 54 55 56 58 59 60 61
4 -> 62          69.500000          4 3 22 52 53 54 55 56 58 59 60 62
4 -> 63          84.080000          4 3 22 52 53 54 55 56 58 59 60 62
                                    63
4 -> 64          46.550000          4 3 22 52 53 54 64
```

3. City Micro Bus Service:

From Node 3 (Prithvi Chowk):

```
Vertex           Travel Time        Path
3 -> 2           10.040000          3 2
3 -> 4           13.060000          3 4
3 -> 5           16.150000          3 6 7 5
3 -> 6           4.220000           3 6
3 -> 7           9.680000           3 6 7
3 -> 8           17.930000          3 4 8
3 -> 9           17.010000          3 6 10 9
3 -> 10          9.280000           3 6 10
3 -> 17          23.690000          3 2 18 17
3 -> 18          16.970000          3 2 18
3 -> 22          11.770000          3 22
3 -> 43          21.820000          3 22 43
3 -> 49          24.780000          3 67 49
3 -> 50          37.650000          3 67 49 50
3 -> 65          15.300000          3 6 10 65
3 -> 66          13.050000          3 66
3 -> 67          18.540000          3 67
```

From Node 5 (Lakeside):

```
Vertex          Travel Time          Path
5 -> 2          26.190000            5 7 6 3 2
5 -> 3          16.150000            5 7 6 3
5 -> 4          11.070000            5 7 4
5 -> 6          11.930000            5 7 6
5 -> 7          6.470000             5 7
5 -> 8          15.940000            5 7 4 8
5 -> 9          24.720000            5 7 6 10 9
5 -> 10         16.990000            5 7 6 10
5 -> 17         37.440000            5 7 6 10 65 18 17
5 -> 18         30.720000            5 7 6 10 65 18
5 -> 22         27.920000            5 7 6 3 22
5 -> 43         37.970000            5 7 6 3 22 43
5 -> 49         40.930000            5 7 6 3 17 49
5 -> 50         53.800000            5 7 6 3 17 13 50
5 -> 65         23.010000            5 7 6 8 65
5 -> 66         29.200000            5 7 6 3 66
5 -> 67         34.690000            5 7 6 3 67
```

For optimization based on fare:

1. Pokhara City Bus Service:

Result along with printing path:

From Node 1 (Mahendrapool):

```
Vertex          Fare          Path
1 -> 0          13            1 0
1 -> 2          13            1 2
1 -> 3          13            1 3
1 -> 4          19            1 4
1 -> 5          22            1 5
1 -> 6          13            1 6
1 -> 7          19            1 7
1 -> 8          22            1 8
```

```
1 -> 9          22          1 9
1 -> 10         16          1 10
1 -> 11         13          1 11
1 -> 12         18          1 12
1 -> 13         19          1 13
1 -> 14         24          1 14
1 -> 15         34          1 15
1 -> 16         31          1 2 16
1 -> 17         26          1 2 17
1 -> 18         26          1 2 18
1 -> 19         29          1 3 19
1 -> 20         32          1 3 20
1 -> 21         40          1 3 21
1 -> 22         16          1 22
1 -> 23         18          1 23
1 -> 24         29          1 3 24
1 -> 25         29          1 22 25
1 -> 26         31          1 22 26
1 -> 27         34          1 22 27
1 -> 28         13          1 28
1 -> 29         14          1 29
1 -> 30         20          1 30
1 -> 31         27          1 31
1 -> 32         32          1 32
1 -> 33         33          1 33
1 -> 34         34          1 34
1 -> 35         37          1 35
1 -> 36         40          1 36
1 -> 37         20          1 37
1 -> 38         24          1 38
1 -> 39         32          1 39
1 -> 40         33          1 40
1 -> 41         40          1 41
1 -> 42         45          1 42
1 -> 43         29          1 3 43
1 -> 44         38          1 22 44
```

```
1 -> 45          41              1 22 45
1 -> 46          26              1 2 46
1 -> 47          29              1 3 47
1 -> 48          15              1 48
1 -> 49          23              1 49
1 -> 50          29              1 50
1 -> 51          29              1 51
```

From Node 5 (Lakeside):

```
Vertex           Fare            Path
5 -> 0           23              5 0
5 -> 1           22              5 1
5 -> 2           22              5 2
5 -> 3           19              5 3
5 -> 4           14              5 4
5 -> 6           26              5 7 6
5 -> 7           13              5 7
5 -> 8           27              5 7 8
5 -> 9           37              5 3 9
5 -> 10          32              5 3 10
5 -> 11          35              5 3 11
5 -> 12          40              5 1 12
5 -> 13          41              5 1 13
5 -> 14          46              5 1 14
5 -> 15          54              5 3 15
5 -> 16          38              5 3 16
5 -> 17          35              5 3 17
5 -> 18          33              5 3 18
5 -> 19          35              5 3 19
5 -> 20          38              5 3 20
5 -> 21          46              5 3 21
```

```
5 -> 22          27              5 7 22
5 -> 23          34              5 3 23
5 -> 24          35              5 3 24
5 -> 25          36              5 3 25
5 -> 26          38              5 3 26
5 -> 27          41              5 3 27
5 -> 28          35              5 2 28
5 -> 29          36              5 2 29
5 -> 30          37              5 0 30
5 -> 31          45              5 0 31
5 -> 32          49              5 0 32
5 -> 33          50              5 0 33
5 -> 34          51              5 0 34
5 -> 35          55              5 0 35
5 -> 36          59              5 0 36
5 -> 37          37              5 0 37
5 -> 38          42              5 7 38
5 -> 39          50              5 0 39
5 -> 40          51              5 0 40
5 -> 41          60              5 7 41
5 -> 42          64              5 7 42
5 -> 43          35              5 3 43
5 -> 44          46              5 3 44
5 -> 45          48              5 3 45
5 -> 46          35              5 3 46
5 -> 47          35              5 3 47
5 -> 48          37              5 1 48
5 -> 49          43              5 7 22 49
5 -> 50          46              5 7 22 50
5 -> 51          27              5 5 51
```

2. Lekhnath City Bus Service:

From Node 3 (Prithvi Chowk):

Vertex	Fare	Path
3 -> 2	12	3 2
3 -> 4	12	3 4
3 -> 5	14	3 5
3 -> 18	14	3 18
3 -> 22	13	3 22
3 -> 52	20	3 52
3 -> 53	20	3 53
3 -> 54	23	3 54
3 -> 55	17	3 55
3 -> 56	27	3 56
3 -> 57	30	3 57
3 -> 58	31	3 55 58
3 -> 59	33	3 55 59
3 -> 60	35	3 55 60
3 -> 61	40	3 55 61
3 -> 62	40	3 55 62
3 -> 63	57	3 55 63
3 -> 64	29	3 64

From Node 5 (Lakeside):

Vertex	Fare	Path
5 -> 2	26	5 3 2
5 -> 3	14	5 3
5 -> 4	12	5 4
5 -> 18	28	5 3 18
5 -> 22	17	5 22
5 -> 52	17	5 52
5 -> 53	26	5 53
5 -> 54	28	5 54

```
5 -> 55          30              5 55
5 -> 56          33              5 56
5 -> 57          36              5 57
5 -> 58          39              5 52 58
5 -> 59          41              5 52 59
5 -> 60          43              5 52 60
5 -> 61          47              5 52 61
5 -> 62          47              5 52 62
5 -> 63          68              5 56 63
5 -> 64          35              5 52 64
```

3. City Micro Bus Service:

From Node 3 (Prithvi Chowk):

```
Vertex           Fare            Path
3 -> 2           13              3 2
3 -> 4           15              3 4
3 -> 5           22              3 5
3 -> 6           13              3 6
3 -> 7           14              3 7
3 -> 8           19              3 8
3 -> 9           17              3 9
3 -> 10          13              3 10
3 -> 17          21              3 17
3 -> 18          18              3 18
3 -> 22          14              3 22
3 -> 43          19              3 43
3 -> 49          20              3 49
3 -> 50          22              3 50
3 -> 65          15              3 65
3 -> 66          15              3 66
3 -> 67          17              3 67
```

From Node 4 (Birauta):

Vertex	Fare	Path
4 -> 2	21	4 2
4 -> 3	15	4 3
4 -> 5	17	4 5
4 -> 6	14	4 6
4 -> 7	13	4 7
4 -> 8	13	4 8
4 -> 9	31	4 6 9
4 -> 10	27	4 6 10
4 -> 17	26	4 17
4 -> 18	24	4 18
4 -> 22	29	4 3 22
4 -> 43	34	4 3 43
4 -> 49	35	4 3 49
4 -> 50	37	4 3 50
4 -> 65	29	4 6 65
4 -> 66	30	4 2 66
4 -> 67	32	4 2 67