A SCHEDULE-BASED TRANSIT NETWORK MODEL

Volume 1 - Main Text

by

C. O. Tong, B.Sc., M.Sc.

A thesis presented in fulfilment of the requirements for the

the degree of Doctor of Philosophy

Monash University

Department of Civil Engineering

1986

TO MY WIFE

CONTENTS

Page No.

CHAPTER 5 MEASUREMENT OF TRANSIT ACCESSIBILITY

Page No.

LIST OF FIGURES

LIST OF TABLES

SUMMARY

This thesis describes a schedule-based transit network model and its applications to trip assignment, accessibility measurement and passenger information systems.

The thesis first shows how a time-dependent computer network description can be obtained from a route timetable database. It then describes various techniques developed for network generation: these include the selection of time period; routes and stops represented. Techniques for network plotting on microcomputers have also been developed. As the network model used is different from conventional headway-based transit models, some new techniques of network modelling can be developed.

A major part of the research is involved in the development and testing of new minimum path algorithms. The new algorithms are able to find time-dependent minimum paths. Thus, the path generated shows for a given starting time from the origin the clock time of the path as it progresses through various stops in the network. Alternatively, an arrival time at the destination can be specified and a minimum path found starting from the destination and retreating back to the origin. Different criteria can also be set for finding the minimum path: apart from minimizing trip time, an algorithm can be used to find an optimal path with minimum weighted trip time where the weightings applied to the various components of the weighted trip time function are user supplied. Furthermore, it can also find minimum paths between any two points with user

specified map coordinates, where these points are not necessarily stops in the network.

The network model with associated minimum path algorithms was first applied to trip assignment. Since the network is time-dependent, various dynamic assignment procedures can be constructed from the model. Based on these new techniques, two assignment sub-models have been developed. The first can be used in the assignment of multi-interval zone to zone trip matrices whereas the second can be used in the assignment of variable demand station to station flows onto the network. Apart from link-based flows, the outputs from both sub-models can produce vehicle-based flows printed on a timetable format and can hence be applied directly to the evaluation of transit schedules.

Secondly, using the model, a method of measuring temporal transit accessibility between zones has also been developed. A conventional gravity type expression is used in the measure, however, the characteristic of the model lies in its ability to show how the accessibility of travelling between zones changes during different times of the day. One important application of the model is that it can be applied to the evaluation of route schedules. A case study is presented to demonstrate the method used. There then follows a discussion of various issues related to transit accessibility which require further research. These are mainly concerned with the problems of including other factors into the accessibility measure, which include passenger arrival patterns at the origin station; network overloading and "idle-time".

Finally, the model has been used in the development of computer generated transit travel information systems. The research is concerned with travel information systems in which computers are used to assist travellers to find an optimal path within a complex transit network. The new minimum path algorithms which have been developed for the model are found to be suitable for these systems.

A trial study using the existing rail, tram and government operated bus networks in Melbourne, Australia has shown that it is feasible to use the software developed to provide a computer generated transit travel information system for a complex multi-modal urban transit network in a large city.

Included in the Appendices are (1) determination of the average distance between two cells, (2) route schedules for a hypothetical transit network, (3) a step by step illustration of the optimal path algorithm and (4) a glossary of terms used in the thesis.

A User's Manual of the programs developed in the model is provided in Volume 2 of the thesis.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

...[redacted].......

C. O. Tong.

ACKNOWLEDGEMENTS

The author wishes to thank his supervisors: Dr. A.J. Richardson and Dr. W. Young, for giving him many original ideas on the initiation of the research; guidance during the development of the project and advice in the preparation of the thesis.

CHAPTER 1

COMPUTER APPLICATIONS TO THE PLANNING OF TRANSIT SYSTEMS

CHAPTER 1

COMPUTER APPLICATIONS TO THE PLANNING OF TRANSIT SYSTEMS

1.1 Introduction

The comprehensive planning, operation and management of a transit system involves various tasks. Firstly, long-range planning is required to estimate future travel demand, this forms the basis for determining the optimal provision of system infrastructure, such as rolling stock, track and stations. Secondly, the efficient operation of the system in the medium-range requires optimal scheduling of vehicles and crew. Thirdly, short-range planning involves the monitoring of existing operations and the detailed planning and evaluation of short-term improvements, such as changes in routes and schedules. Further, the provision of adequate guidance information to travellers is also another important task of the transit operator.

Computers can often assist in the various planning tasks mentioned and in recent years there has been much research on the development of computer packages for performing these tasks. For example, transit models are commonly used in long- and short-range planning of transit systems. The most commonly used techniques are the formulation of the transit system into a network model and the development of algorithms to find minimum paths in these models (Last and Leak, 1976). Computers can also assist in scheduling (Wren, 1980) and in the provision of travel information to travellers (Pickett, 1980).

This chapter first discusses in general the various computer applications relating to transit systems and then proceeds to describe in more detail the scope and objectives of the present research.

1.2 Long-Range Planning of Transit Systems

As early as in the 1960s, computers have already been used in land use transportation studies in the long-range planning of transit systems. For example, in the London Transportation Study (Tresidder et. al. 1968) network models were used in the planning and evaluation of transit systems.

As shown in Figure 1.1, the land use transportation planning process often consists of the following three groups of models:

(a) demand models;

(b) highway supply models; and

(c) transit supply models.

These models are inter-related and provide an analytical tool for the planner to study the interactions between land use, transport demand and transport supply. The objectives of the planning process are to test and evaluate alternative transport networks and land use plans and to obtain optimal solutions.

Supply models consist of network models to represent the supply characteristics of the network (such as travel speed and capacity), and algorithms for finding minimum paths in these networks. From these models, the travelling time or cost between zones within the study area can be determined. Since most cities are provided with both highway and transit

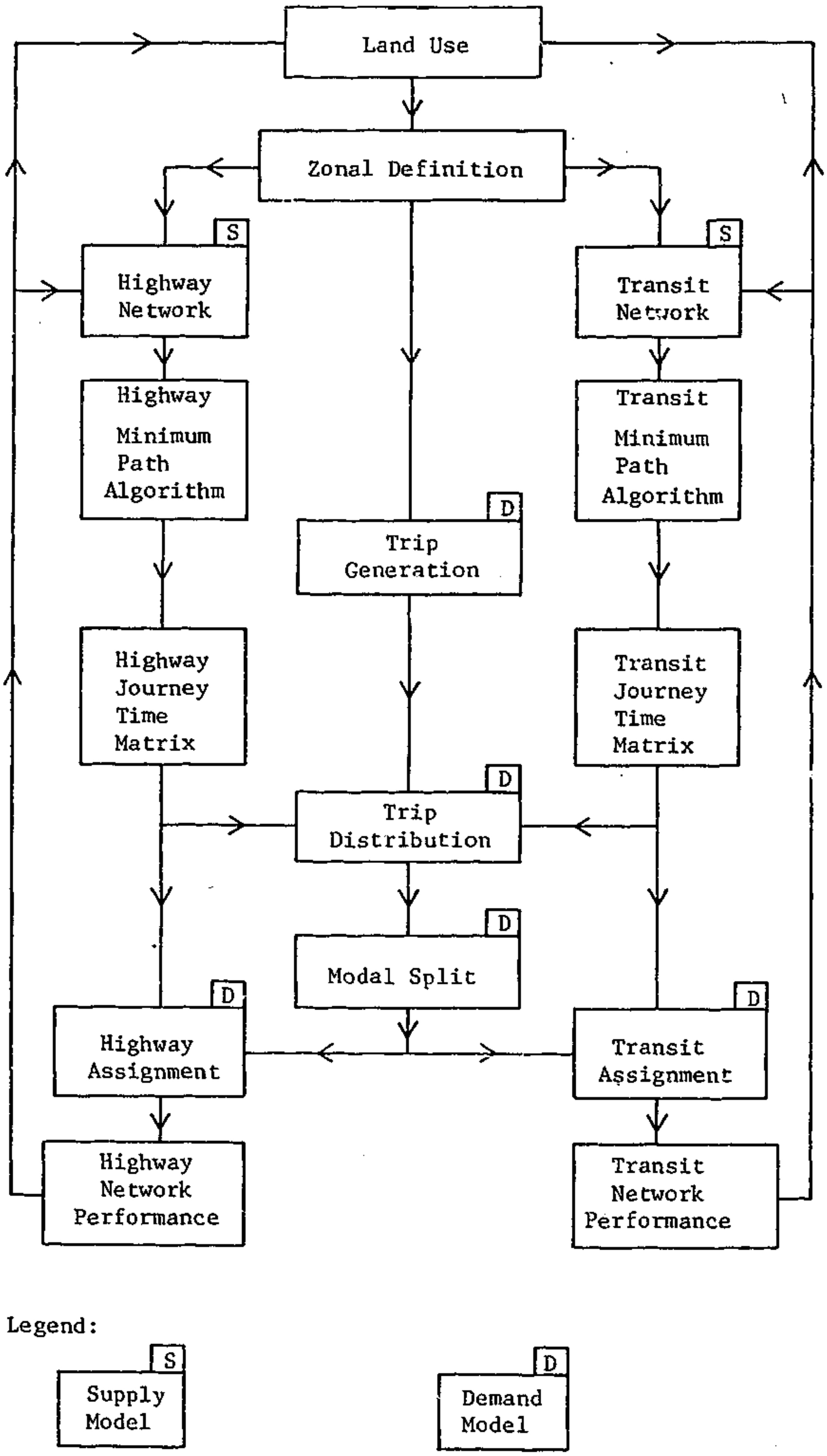


Figure 1.1 The land use transportation planning process

networks, it is necessary to have separate highway and transit type supply models to simulate these networks.

Demand models use the land use characteristics of each zone and the level of service provided by transport networks to predict the demand for travel. Trip generation, distribution and modal split are traditionally the three successive steps used to determine the demand for movement between zones by different modes. Assignment models are then used to determine the passenger flows on the highway and transit networks from the predicted inter-zonal movement.

The development of transit network models, minimum path algorithms and assignment models are reviewed in more detail in Chapters 2, 3 and 4 respectively. These reviews show that in the past, planning was focused on highway networks rather than transit networks. The reason was that travel by public transport was not considered by planners as an important mode of travel. According to data collected by Thomson (1977) and Yago (1984) the number of public transport trips per person in most North American and European cities has shown a steady decline within the period from 1950 to 1970. Both authors pointed out that the cheap price of owning a motor car and steeply rising public transport fares were the major factors contributing to the decline in public transport patronage. In most cities, buses were the major transit mode. As buses operate in the highway network and bus flows were only a small proportion of total flows, it was not necessary to have very accurate transit models. The main objective was to obtain an optimal highway network consistent with predicted demand and therefore transit models were only important as input to modal split models for filtering out transit trips from total trips predicted from the demand models. Transit assignment models were usually quite simple, and only used as a rough assessment of the requirements of bus routes and frequencies.

Further, land use transportation studies were more frequently used for long-range than short-range planning. Highways have a long design life and once built their capacity (road widths and junction layouts) cannot be adjusted easily, therefore, the highway planner's task is often concentrated on long-range rather than short-range planning . In transit network planning, emphasis was therefore mainly on the estimation of maximum bus numbers operating in various road sections and the allocation of space for transit stops and terminals. It was only necessary to have a broad assessment of load/capacity ratios of transit networks whereas detailed planning of transit routes and schedules could be deferred to a later stage for the transit operator to consider. As highway networks often have design lives of twenty years or more, it is not necessary to have very accurate models for long-range planning.

1.3 Short-Range Planning of Transit Systems

In recent years, transport authorities have become increasingly aware of the need to provide improved transit services. A report by Kirby et. al. (1979) showed that financial assistance provided by the U.S. Government to public transport services has grown to several billions of dollars per year. A report of the Second Conference on Mass Transportation in Asia held in Singapore in 1984 showed that many Asian cities such as Hong Kong, Singapore, Taipei and Seoul are planning to upgrade their mass transit systems. The result is that the transit systems in many large cities have or will soon become extensive, multi-modal and are often either partially or wholly segregated from the highway network. As the transit system becomes more complex, there is a need for better models which can be used for the short-range planning and evaluation of these systems. For example, models are required for predicting the behaviour of travellers resulting from minor changes in transit services, and the evaluation of costs and benefits of such changes.

Detailed planning and evaluation of transit systems is a complex process because it involves the consideration of a large number of variables such as routes, stops, capacity, schedules, fares and operation costs. Apart from the use of assignment models for evaluating the balance between supply and demand there is also a need for the use of accessibility models to evaluate the level of service provided to different groups of travellers. In recent years, some transit models have been developed for this particular need. An example is TRANSEPT (Last and Leak, 1976) developed by the Local Government Operations Research Unit (LGORU) of Britain for the evaluation of short-term changes in bus networks. This model is however still limited in the scope of application to transit planning because it does not include schedule-based network modelling. It is considered that there is a need for further development in this area so that a wider range of transit models are available.

The models described are often sophisticated and hence are suited more for use by transport engineers or planners rather than transit operators. On the other hand, there are other types of transit models which have been developed mainly for use by transit operators. Examples are interactive graphic models developed by Rapp and Gehner (1976), and BUSDES (Bowyer, 1983) developed by the Australian Road Research Board. These models are intended to be user-friendly, require simple input data and provide quick results for the testing and evaluation of alternative bus routes. The outputs are graphic and hence more meaningful and easy to interpret.

In spite of the availability of various types of computer software, Chua (1984) stated that more than 70 per cent of British bus operators still use manual methods for the planning of bus routes. The reason may be that it is only in the recent three to four years that computer facilities have become cheap and hence many existing bus operators are still not familiar with the computer environment. So they would still prefer to rely on personal experience and rule of thumb methods for the planning of bus routes. The recent development of cheaper and more powerful computers would make computers more available to these people. When coupled with the availability of suitable software, a much larger proportion of transit operators may adopt computers to assist them.

1.4 Transit Service Scheduling

While the initial planning stage of a transit system involves a rough estimate of routes and frequencies, the final planning stage requires the working out of timetables and assignment of vehicles and crews onto various schedules. Vehicle and crew scheduling is often a tedious and time-consuming exercise because it requires finding an optimal solution to a situation where there are a large number of factors to be considered. Such factors include the temporal variations of passenger demand, labour agreements, operation constraints and fleet size. It is not surprising therefore to find that computer models have been developed to assist transit operators in the task.

Comprehensive reviews on this subject has been provided by Wren (1980), Schmidt and Knight (1980) and it is beyond the scope of the present thesis to describe them in detail. As reported by Schmidt and Knight, in 1980, 40 per cent of transit operators in the U.S. and Canada were already adopting computer aids for scheduling.

One potential advantage of using computer-aided scheduling is that it establishes a database of route schedules in the computer. This information could in fact be utilised for various planning tasks although at present very few transit models have been developed for this purpose. For example, network models usually use route headways as input instead of route schedules. As will be discussed in the next section, a route schedule database is also required in models for generating travel information.

1.5 Passenger Information Systems

Every transit authority provides some travel information service to its customers. Apart from the distribution and display of route timetables, very often travel information offices are set up at strategic locations within the transit system to deal with passengers' enquiries. In recent years, research (e.g. Pickett, 1980) has been conducted on the generation of passenger information by computer. One popular concept is to place a computer terminal at a travel information booth so that a passenger could input travel requirements such as the origin, destination and starting time of the trip and then the computer would output the required travel information.

As will be discussed in more detail in Chapter 6, a computerized passenger information system can be very useful to the public transport authority. While it may not always be desirable to adopt a completely automatic guidance system which completely dispenses with the use of human guidance, the system could help to reduce staff costs and eliminate human error in the interpretation of route timetables. Therefore, with the computer technology which is now available and if suitable software is developed, then computerized passenger information systems will most certainly be more widely adopted in the near future.

1.6 Objectives and Scope of Research

The above discussions show that there are several areas in computer applications to transit planning which require further research.

Firstly, although there has already been much interest in the development of computer software for transit planning, previous researchers have, however, often confined their work on one particular aspect of the various applications while few have studied the inter-relationships between the various applications. The result is that although computer packages have been developed for the various tasks, there are few that can perform all the tasks required by a transit authority. There is often difficulty in switching from one package to another to perform different tasks due to incompatibility of input and output data.

Secondly, most existing transit network models are suited more for long-range than short-range planning. There is a need for the development of transit network models for short-range planning.

Thirdly, recent advances in computer technology have increased the potential applications of computers in the establishment of passenger information systems. There is a need for further research in this area and particularly in the development of software for generating travelling information.

The main objective of the present research is to develop a schedule-based transit network model. The model contains a package of computer programs which uses route schedules as the main input for various planning tasks. As route schedules are a major characteristic of most transit systems, a schedule-based model could therefore increase the scope of application of the model. Thus it can be used for transit route planning as well as for other tasks such as generating travel information or scheduling. Furthermore, it is found from the present research that a schedule-based model can provide many particular features of network modelling which cannot be achieved in a conventional headway-based model and is hence particularly suitable for short-range planning of transit systems.

Some new techniques of network modelling have been developed. For example, a technique has been developed for finding paths between two points defined by their map co-ordinates which are not necessarily nodes in the network. The method used is to adopt a searching procedure in which all transit stops located within accessible distance from the origin or destination are evaluated and access links to potential access stops are generated. In the model, the walk mode for access is assumed, however, the possibility of including other access modes is discussed in Section 2.8.

The scope of the present research can be illustrated by reference to Figure 1.2. Firstly, a method for constructing a schedule-based network description is developed. The second task is to develop algorithms for finding minimum paths in the schedule-based network. Finally, the potential applications of the network model in the following three areas were investigated:

(a) assignment modelling;

(b) accessibility modelling; and

(c) travel information systems.

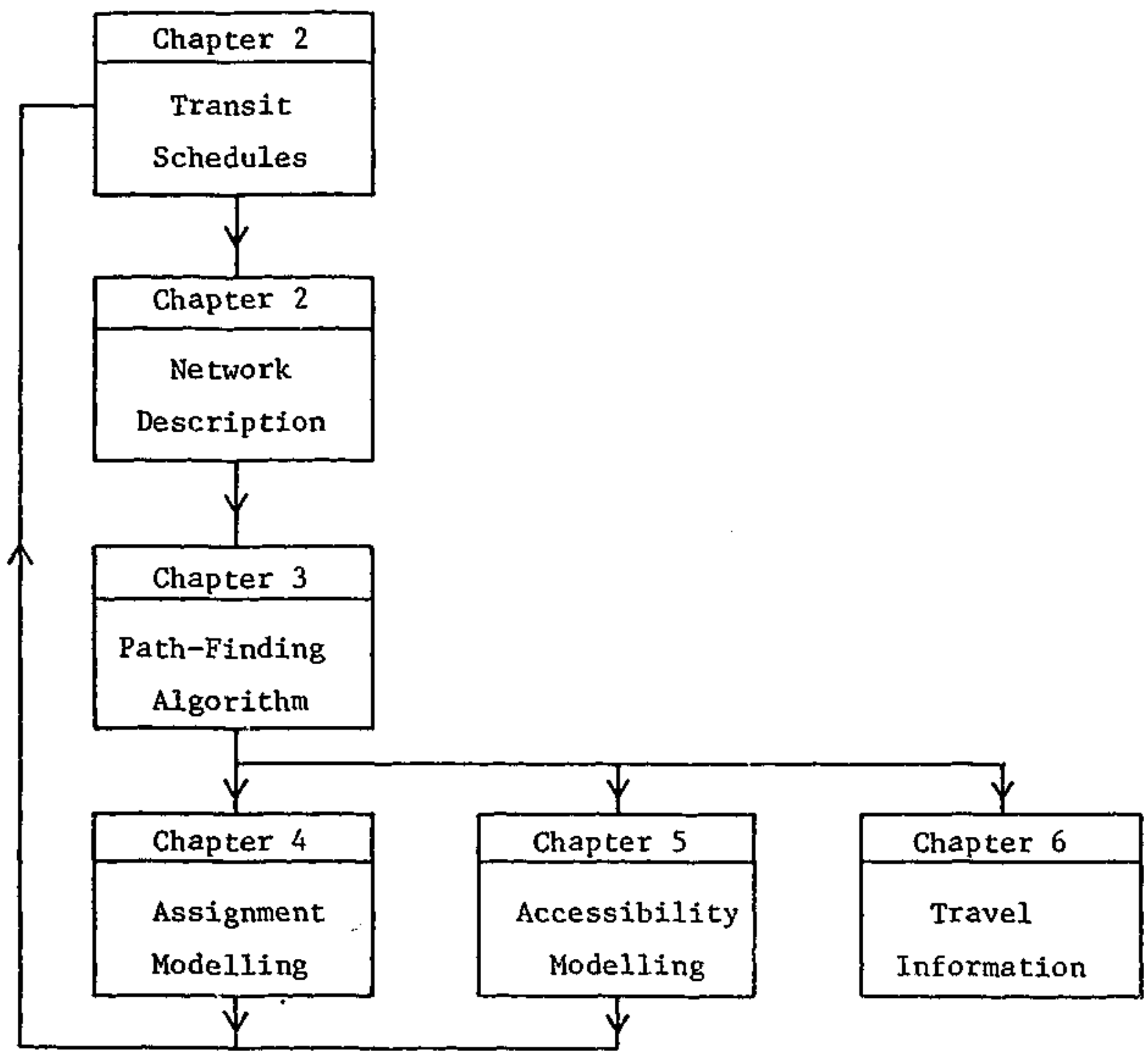


Figure 1.2 Scope of research

The operation sequence of the various modelling activities in the network model is illustrated in Figure 1.2. Thus, starting with a given set of transit schedules, a network description can be generated. With the network description and associated path finding algorithms the schedules can be evaluated either by assignment results or by accessibility measurement. The evaluation could lead to changes in transit schedules and hence a reiteration of the modelling process. Apart from planning tasks, the model can be used for providing travel information.

Figure 1.2 also shows the general structure of the thesis in terms of identifying various Chapters of the thesis used for describing different activities in the model.

CHAPTER 2

A SCHEDULE-BASED TRANSIT NETWORK DESCRIPTION

CHAPTER 2

A SCHEDULE-BASED TRANSIT NETWORK DESCRIPTION

2.1 Introduction

A transit system has often been described as a network because it provides an inter-connected mesh of serviced routes. For this reason, many network analysis techniques can be applied to the planning and evaluation of transit systems. For example, algorithms for finding minimum paths in networks can be used to determine the route choice and travel time of transit trips. Such techniques form the basis of many types of transit planning models.

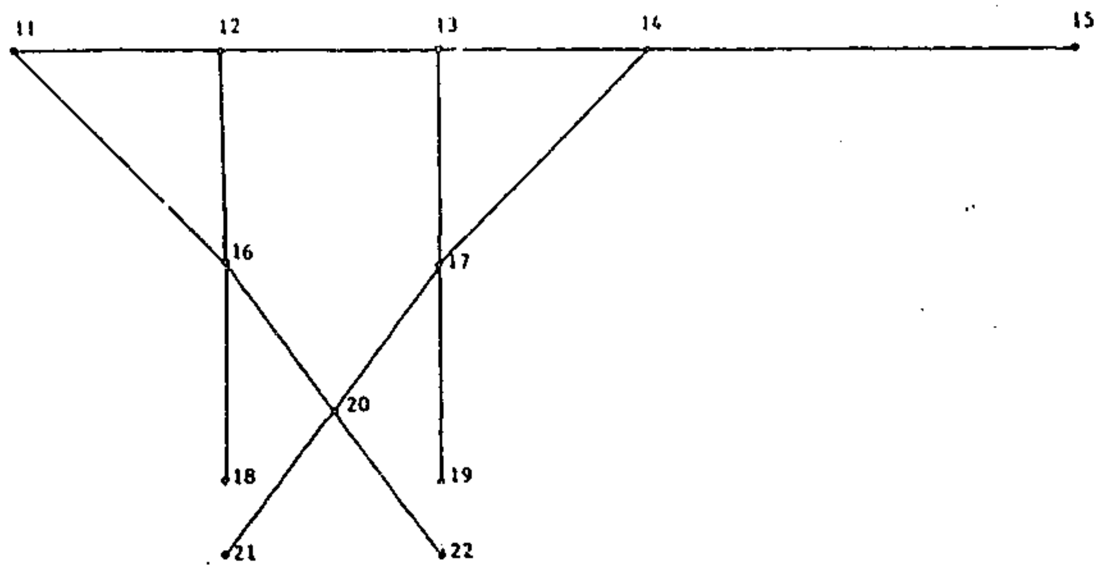
However, in order that a transit system can be modelled as a network, it is necessary to specify in mathematical terms its operational characteristics. The ability to establish an accurate and detailed network description is therefore an essential element in many transit models.

This chapter first describes the conventional method of specifying a transit network description by computer. It then describes a new method in which route schedules rather than route headways are used in the network specification. A comparison of the two methods is discussed in Section 2.4. It then proceeds to describe the method for coding network characteristics, the process includes the coding of route timetables (Section 2.5) and node locations (Section 2.7). After all the necessary data have been coded, network specifications with different levels of detail can be generated. These include options to specify particular transit modes, routes and time to be included in the network (Section 2.6); generating additional access links which connect selected points with the network (Section 2.8); increasing the network detail by adding more nodes (Section 2.9); decreasing the network detail by eliminating some nodes (Section 2.10) and the generation of a reversed network (Section 2.11). Finally, the development of software for network graphics is described in Section 2.12.

2.2 Conventional Transit Network Descriptions

Many existing transit network descriptions have been evolved from highway network models. A highway network is constructed by making a network diagram to represent the actual highway configuration. As shown in Figure 2.1, a simple network can be represented by a network diagram. The lines of the network are called links and the intersection points of links called nodes. Each node is assigned a number and a link is then identified by the numbers of the two nodes at its ends. Since a link is normally assumed to be bi-directional, it could be regarded as consisting of two oneway arcs. A network file consisting of an inventory of arc records is then coded. Each arc record contains its start node number; end node number; and other characteristics of the arc such as its travel speed and distance.

Since the transit network of most cities consists of bus routes operating within the highway network, a transit network description can often be obtained from modifications of the corresponding highway network file. The same basic network



Network diagram

Arc records

A-NODE	B-NODE	DISTANCE (km)	SPEED (km/hr)
11	12	5	30
11	16	7	32
12	13	5	30

.
.
.
etc

Route descriptions

Route	A1
Nodes	11, 12, 13, 14, 15
Headway	15 minutes
Route	A2
Nodes	11, 12, 13, 17, 19
Headway	20 minutes

.
.
.
etc

Figure 2.1 A hypothetical network with conventional network descriptions can, initially, be used. Possible changes required are that link speeds may have to be changed to account for the slower movement of buses compared with other traffic. Other additional information may have to be coded, such as transit route descriptions which describe the route headways and nodes passed by various routes.

Such network descriptions can be regarded as headway-based because they assume that each route operates with constant headway and the link speed is constant for all routes using that link. These assumptions are usually acceptable when the network descriptions are used in models for long-range planning of transit systems. The reason is that in these cases, actual operating schedules of the transit system are generally not known and it is not necessary to provide very accurate modelling.

2.3 A Schedule-Based Transit Network Description

The present model adopts a completely different method of forming a network description.

The method used is to generate a network description from route timetables. A route timetable contains information on the list of stops used by the route and scheduled departure times at these stops (or arrival time for the last stop).

The following is a hypothetical example of a route timetable:

Route A2

Stop	11	12	13	17	19
	7:05	7:15	7:25	7:35	7:45
	7:25	7:35	7:45	7:55	8:05
	7:45	7:55	8:05	8:15	8:25
Depart	8:05	8:15	8:25	8:35	8:45
	8:25	8:35	8:45	8:55	9:05
times	8:45	8:55	9:05	9:15	9:25
	9:05	9:15	9:25	9:35	9:45
	9:25	9:35	9:45	9:55	10:05
	9:45	9:55	10:05	10:15	10:25

From the above timetable, it can be seen that a complete run of the vehicle from the starting stop to the end stop is represented by one row of the schedule. A more careful study will show that every row of schedules in the timetable has the same route profile (The space-time trajectory of a vehicle when it is operating along a route is called its route profile.)

A timetable database is coded into the computer by entering the route profile and starting times of different runs of each route. In other words, it is only necessary to enter one row and one column of the matrix of time schedules. Using this database, it is required to generate a transit network description which consists of an inventory of arc records with each record containing the following information:

A-NODE - start node number

B-NODE - end node number

LTIME - time difference between departure time at A-NODE and departure time at B-NODE.

R - number of times a vehicle runs through the arc as recorded

ST(1) - departure time of vehicle from A-NODE in first run

.

ST(J) - departure time of vehicle from A-NODE in Jth run

.

ST(R) - departure time of vehicle from A-NODE in Rth run

RTNO - route number

The information contained in these arc records can be extracted from the timetable database. For example, from the hypothetical timetable shown above the following arc records can be generated:

A-NODE	B-NODE	LTIME	R	ST(1)	ST(2)	....	ST(9)	RTNO
11	12	10	9	7:05	7:25	....	9:45	A2
12	13	10	9	7:15	7:35	....	9:55	A2
13	17	10	9	7:25	7:45	....	10:05	A2
17	19	10	9	7:35	7:55	....	10:15	A2

Apart from transit links as described above, the network may also contain interchange links. These links represent the walking paths made by travellers when changing from one route to another. An interchange link normally connects between two stations which are situated within reasonable access distance from one another, but it could also connect between two platforms in the same station if the station is a large one. Interchange links are generated by coding the necessary information into a data file.

There may also be access links in the network which connect specified points such as trip origins or destinations to nodes in the network. The method of generating these links is described in Section 2.8.

Both interchange links and access links will only have arc records containing information on their A-NODE, B-NODE, LTIME and RTNO. In fact all these arcs have been be assigned a RTNO equal to WALK to distinguish them from transit arcs.

Figure 2.2 illustrates the formation of various links in a transit network.

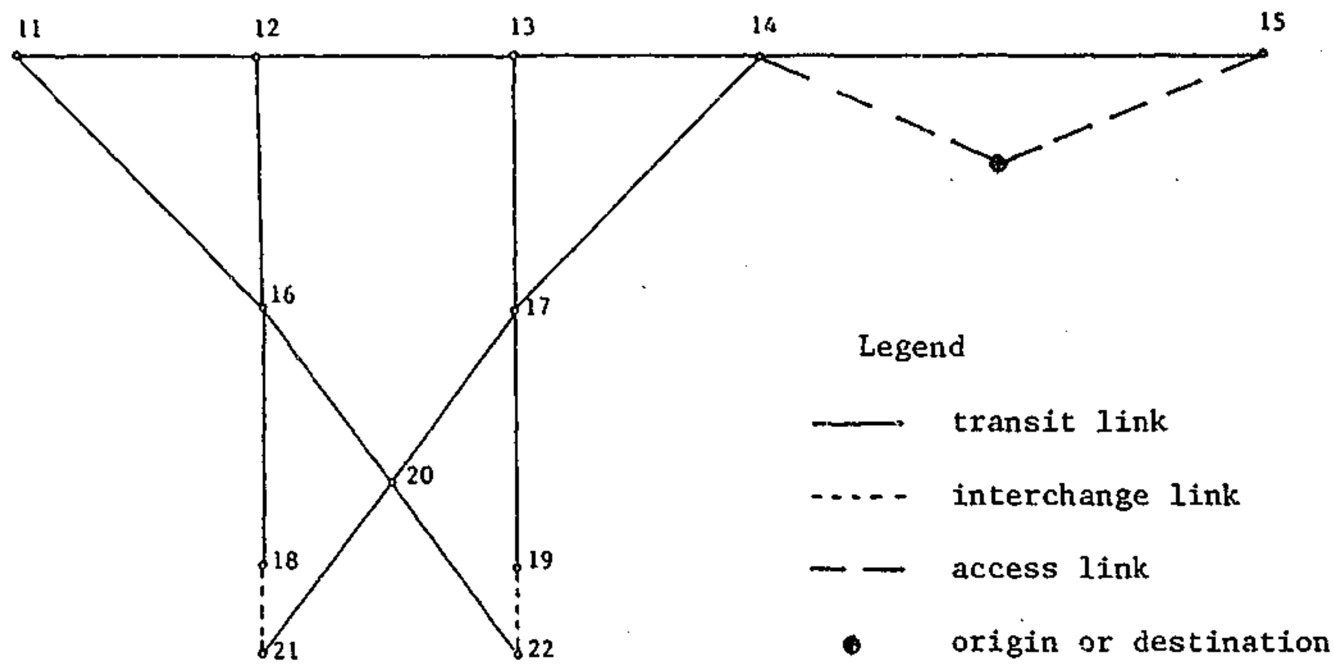


Figure 2.2 Different types of links in a basic network

2.4 Comparison with Conventional Network Description

The proposed network description is different from conventional ones in a number of aspects.

First, the new network representation is schedule-based. It describes vehicle profiles and headways as specified in route timetables, and models the time-dependent movement of vehicles within the transit network.

If the transit system operates to published timetables with reasonable reliability, then the present network description should be more accurate than conventional types and enable the development of improved transit models. For example, when developing minimum path algorithms in this network, it is possible to keep track of the clock time as a path travels through the network. In this way waiting times during route changes can then be calculated based on schedules rather than headways as in conventional models. Minimum paths which are related to specific trip starting or finishing times can also be found.

The ability to find time-dependent minimum paths is an important property of the network which enables the development of more advanced models in traffic assignment, accessibility analysis and route guidance systems. The application of the network in these three areas of transit modelling will be described later in this thesis.

Second, the new network representation is suitable for modelling transit systems which are already in existence and operating to a set of timetables. Existing networks often have irregular schedules: route headways may vary with times of day or day of week and some special services may also operate during specific times. Different routes using the same link may also operate at different speeds. One of the advantages of the new network representation is that it describes routes and schedules accurately. Therefore it is a useful tool for monitoring the operation of an existing transit system or for evaluating proposed short-term changes to an existing transit system. When modelling a hypothetical transit network, or in the planning of a completely new transit system, where detailed operating schedules have not yet been considered, then conventional network models would be more suitable.

Third, the network representation described can be generated directly from a computerized database of route timetables. In conventional network models, it is necessary to code up different sets of data for different networks, such as networks for morning peak, off peak or afternoon peak. In the new system, using the timetable database, it is possible to generate a network covering the schedules of any selected period within any day of week. It is also possible to generate a network for selective routes or modes. The methods for selective network generation are described in Section 2.6.

Finally, apart from providing a network description, the timetable database can also be used for various other purposes, such as the planning and printing of route timetables and scheduling.

2.5 Coding the Route Timetables

In order to operate a schedule-based network model, it is necessary to establish a database of route timetables. The coding of timetables appears to be a simple operation, however, a study on the practicability of establishing a timetable database for Melbourne showed that the process is not as simple as it appears. Approximately 30 man-days were spent on coding all the timetables of the railway network in Melbourne. As will be discussed in the following paragraphs, there are various factors which cause the coding process to be time-consuming and tedious.

Firstly, the schedules of a route may not all follow the same profile as previously described in the hypothetical example. In some of the timetables the schedules were found to be quite irregular and it was necessary to divide the schedules into different groups according to profile and regard each group of schedules as a separate route for coding.

Secondly, although rail timetables show the schedules of vehicle departures at every station, this is not so for bus or tram timetables. In fact, from the bus and tram timetables published by the Melbourne and Metropolitan Transit Authority of Victoria (MTA), it is found that only about 10 to 20 percent of all stops in a bus or tram route are referenced in the timetables. Although it is possible to obtain the departure schedules at every stop by interpolation, for efficiency in computer storage and computations, it is not advisable to do so. The alternative method of overcoming this problem is to build a detailed network when the need arises.

The method is described in Section 2.9.

Thirdly, separate timetables are usually produced for weekdays, Saturdays and Sundays. (Some routes may also have different schedules for different weekdays such as Fridays to cater for shoppers.) The number of timetables required to be coded is hence quite large.

Fourthly, while the timetables for rail, bus and tram routes operated by the Melbourne and Metropolitan Transit Authority of Victoria (MTA) are readily available, it is sometimes not possible to obtain the timetables of bus routes operated by private bus companies.

For the present study, all the weekday services for rail, tram and MTA buses have been coded. Although the weekend services and the private bus system have been excluded from the study, it is considered that the network used is sufficiently large and complex to demonstrate the feasibility of the various applications of the model in a large multi-modal transit system.

Computer programs have been written to facilitate data coding, checking and editing. The method of using these programs is described in Section 1 of the User's Manual.

Recently, it was found that many transit agencies such as the MTA had started coding timetable information on computer for applications in timetable production and conflict resolution. This means that for these agencies a timetable database is already available and the effort of data coding required for operating the model can be saved. What might be required, however, is a program that can convert the database to a format suitable for input to the model.

2.6 Selective Network Generation

The size of the timetable database for the entire urban area of Melbourne is quite large. The transit system has many routes and separate timetables are produced for weekdays and weekends. When solving a particular network problem it is only necessary to select the relevant parts of the database for generating a selective network file. For example, if the network model is to be used in a planning task which is related to a weekday morning peak period, then only the relevant route schedules operating within this period are used in generating the network file. Other selective procedures are also possible. The various options of selective network generation are described below:

(a) Selective mode - only routes of a selected mode, such as train, tram or bus are included in the network. The procedure is quite simple because the timetable database for different modes is in fact coded into different files. For example, in order to generate a network of tram and bus routes, all that is required is to select the relevant timetable data files for input to the network generation program.

(b) Selective route - only selected routes are included in the network. This could be done by editing the timetable data file so that unwanted route timetables are deleted before inputting to the network generation program.

(c) Selective time - only routes and schedules operating within a selected time period are included in the network. In the network generation program, the user is required to input a selected time period, e.g. if a time period between 7 am and 10 am is selected, then during the network process, only schedules falling within the selected time period will be included in the network descriptions.

2.7 Coding the Location of Nodes

The functions of the network model can be extended by coding a node location database. For example, if the locations of the nodes in a network are known, then other functions such as network plotting or finding door-to-door minimum paths can be determined by the model.

In connection with a study on establishing a passenger information system for the city of Melbourne in Australia, a node location database has been established for the transit network of that city. In that study, the location of a node is based on its map co-ordinates according to street maps of that city published by Melway Publishing Pty. Ltd. (1982). Most of the street maps used are of scale 1:20000 and these maps are divided into cells of approximately 400 by 400 metre square. Since the location of a node has been coded according to the map grids of the cell in which it is located in, the location of a node is therefore specified to an accuracy of 400 square metre. Details concerning the coding of node locations are described in Section 2 of the User's Manual.

If a more accurate locational system is required, then it is possible to sub-divide a 400 metre square cell into, say sixteen 100 metre square cells. What is needed is a transparent template with grids of the required spacing that can be used for overlaying on the street maps. The system of referencing the map co-ordinates of a point would also have to be modified. In fact cells of any size can be used depending on the accuracy required.

2.8 Generation of Access Links

Apart from network plotting, the node location database can be used for generating access links from specified points to nodes in the network. Using this method, it is then possible to find minimum paths between points which are not nodes in the network.

Suppose an origin and a destination are specified by inputting their map grids, then as shown in Figure 2.2, access arcs are generated from these points to nodes in the network.

The search for accessible nodes is not a difficult problem. The program first starts from the specified point and searches for all nodes which are located within the same cell or the first ring of adjacent cells. Access links are then generated to connect to these nodes. If no node can be found, then the search is extended outwards to the next ring of cells. The procedure continues until either at least one node is found or the search has extended beyond the fifth ring of cells. The procedure is illustrated in Figure 2.3.

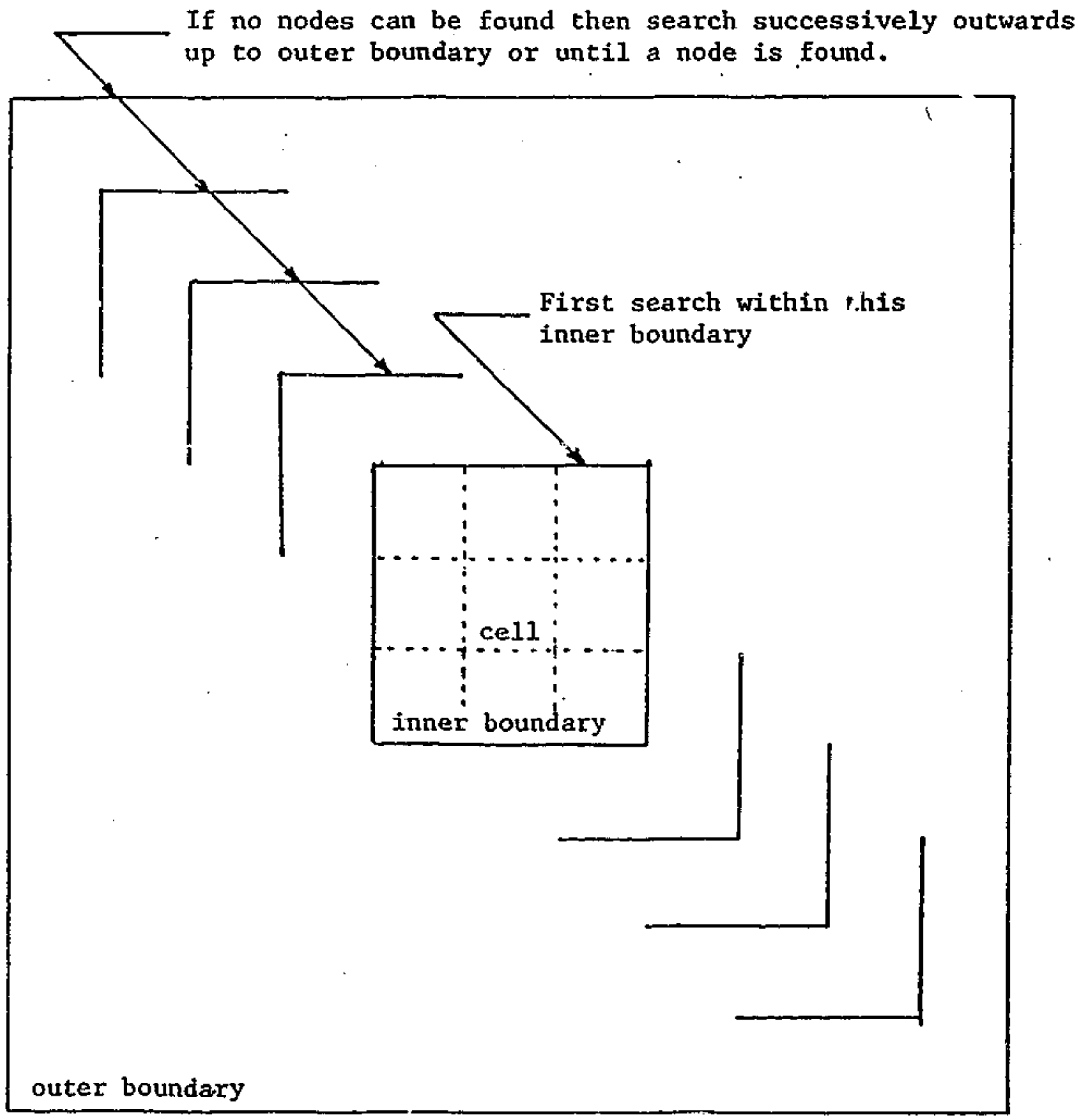


Figure 2.3 Node searching procedure

After the node searching procedure the next step is the calculation of link distance and time. In this case, as the exact location of a point is not known, (only the cell in which a point is located has been specified), the following assumptions have to be made:

(a) A point is assumed to be randomly located within a cell.

(b) Travelling paths between two points are seldom direct. In the case of Melbourne, the road network is based predominantly on a grid pattern oriented in the north-south and east-west direction. Hence it can be assumed that the travelling distance between two points is not its direct distance but approximately equal to the sum of the vertical and horizontal distances between the two points. If this grid pattern road network is rotated clockwise by an angle b then the rectangular distance between the two points is given in Appendix 1B. If the road network is of other forms, then other assumptions on the travelling distance between two points will have to be used.

(c) As shown in Appendix 1A, the average distance between two points assumed to be randomly located in the same cell is equal to 0.33 times the sum of the height and width of the cell; the average distance between two points assumed to be randomly located in different cells is equal to the sum of the vertical and horizontal distances between the centroids of the two cells. However, in the latter case, if the two cells are aligned either vertically or horizontally, the vertical or horizontal distance should not be zero but equal to 0.33 times the height or width of the cell.

(d) The link speed will depend on what type of access mode is specified by the program user. Speeds of 5.5, 10 and 30 km/hr are assumed for access modes of walk, cycle and car respectively. The travel time of each link is then calculated from the link distance and speed.

The option of access by car mode has been included in the program to account for the fact that the search area for potential access stops would otherwise seem unreasonable large if only the walk and cycle access modes were assumed. The search procedure adopted is actually more oriented towards the walk and cycle modes for access because the first search is only within the boundary of nine cells. The subject of access to a transit stop by car has not been fully investigated in the present model. In this case, the search procedure should be modified to cover a wider area. The evaluation of potential access stops would also be more complex. Apart from access time, other factors such as availability of free parking space, difference in fares, the possibility of trip linking, seat availability and other comfort-related considerations may have to be considered. For this reason, only the walk access mode has been assumed in the assignment and accessibility sub-models described in Chapters 4 and 5.

2.9 Building a Detailed Network

Since the network is generated from the timetable database, only those stops of a route which are referenced in timetables are represented as nodes in the network. Experience with coding a timetable database for Melbourne showed that while railway timetables contain schedules of all stops in the rail network the bus and tram timetables in fact only contain schedules of major stops along the route (about 20 % of all stops are included). Although it is possible, by interpolation, to expand the bus and tram timetables to include the schedules at these stops, it is not practicable to do so. To represent each stop as a node in the network would not only increase greatly the required computer storage but could also slow down the processing of programs in the transit model.

In particular cases, some other stops are added to the timetable database. These are stops which have been omitted but considered as essential to basic network modelling. For example, stops used as interchange points between routes, or stops located at points which represent a major change in direction of the route alignment can be added to the network by expanding the timetable database.

A network which is generated directly from the timetable database is called a basic network and those stops which are represented as nodes in the network are called major stops. Those stops which have been omitted from a basic network are called minor stops.

Although a basic network is sufficently accurate for most purposes, however, in some cases, such as the generation of access links, it would be desirable to include other stops for consideration. Especially for tasks related to the generation of passenger information, travellers must be guided to the correct stop for access to or egress from the transit system. Such problems require the building of a detailed network.

As illustrated in Figure 2.4, the method of building a detailed network consists of the following three steps:

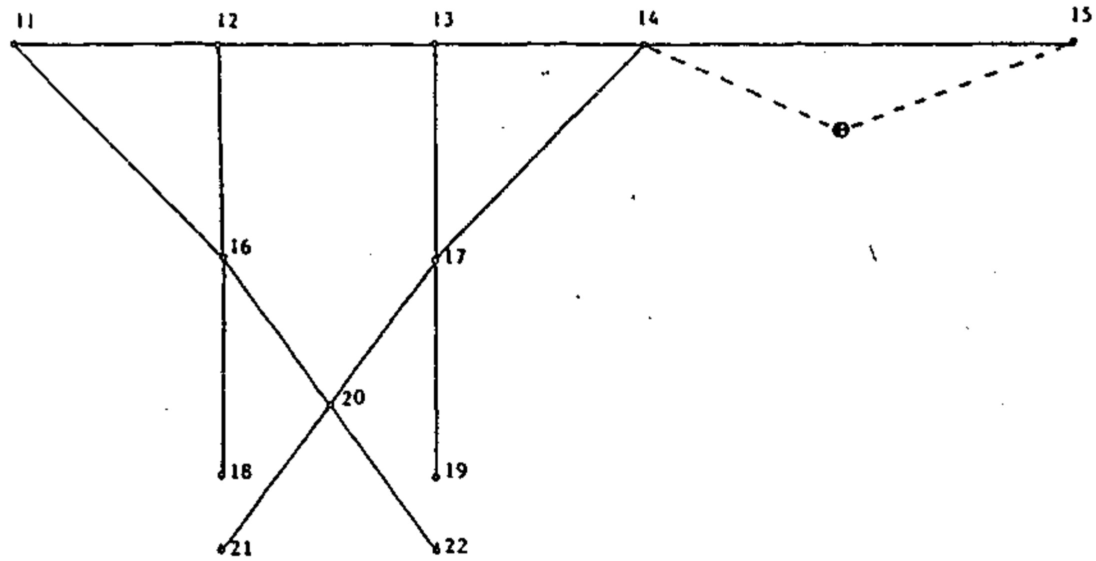
(a) The first step finds access links to nodes as described in Section 2.8.

(b) From the nodes selected, the second stage then looks at those minor stops which are associated with these nodes. A node is in fact a major stop and all minor stops which are located along a link originating from that node are regarded as associated to that node. As several links may originate from a node, there may be several groups of minor stops associated with the same node.

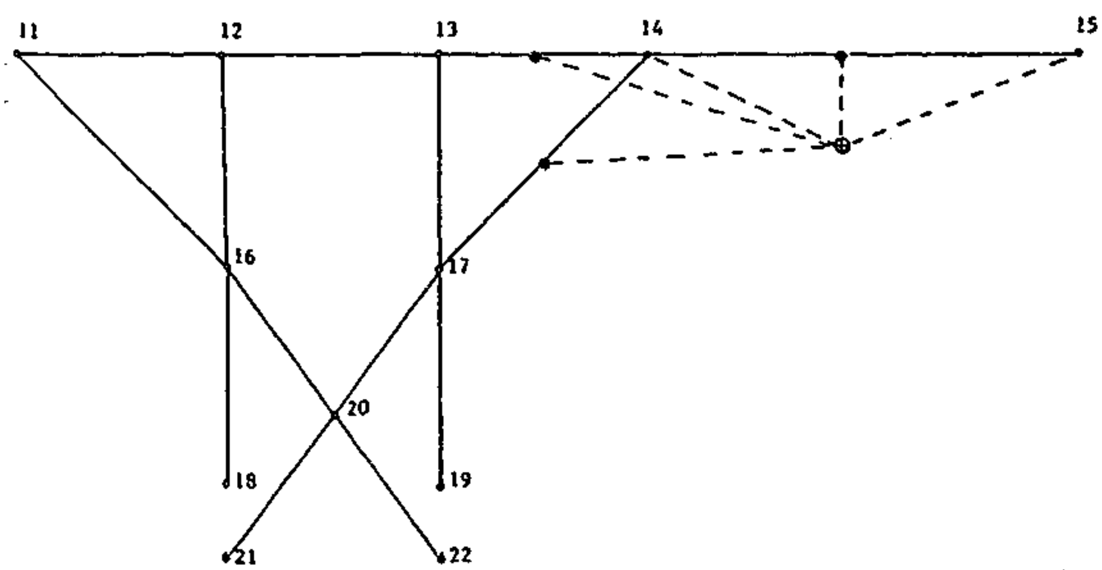
(c) In the third step, each group of minor stops is evaluated according to its distance from the associated node. The stop with the minimum distance is then selected. If this minimum distance is less than the distance of the access link to the associated node, then a new access link connecting to this stop is generated.

The above technique has the advantage that only a small proportion of the total number of minor stops in a transit system is included into the network for any particular problem. A detailed network usually will only have a few more

Step 1 : locate access nodes



Step 2 : evaluate associated minor stops



Step 3 : select minor stops

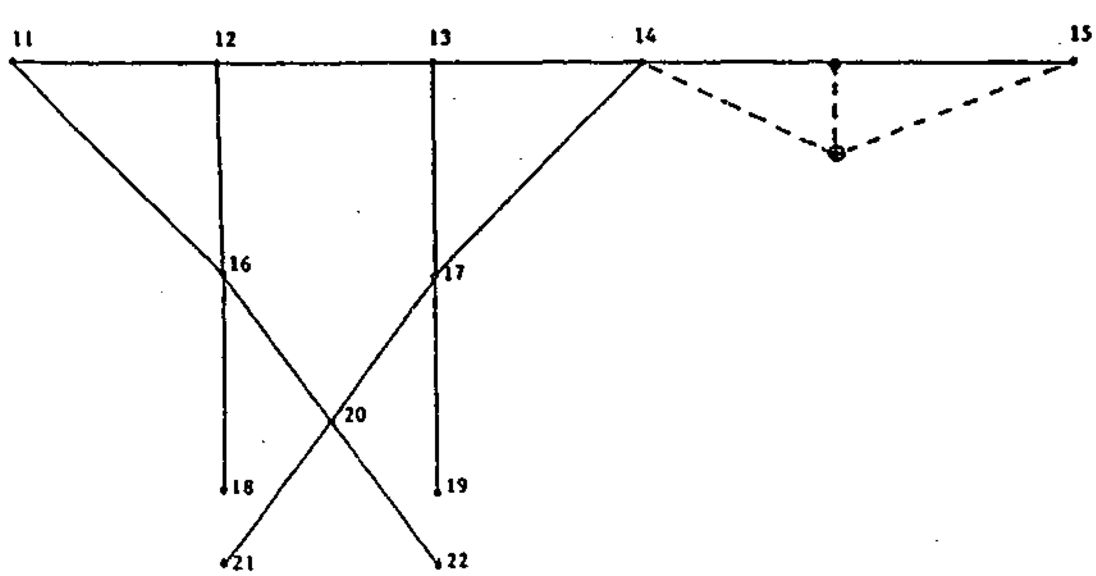


Figure 2.4 Building a detailed network

nodes added as compared to the corresponding basic network.

In addition to a timetable database and a node location database, the method requires the coding of a third database on minor stops. This database does not require coding the map grids of minor stops. However, it is necessary to code a file which specifies the position of minor stops in relation its associated major stops. For example, a record in the data file will contain the following information:

(a) A-NODE of link,

(b) B-NODE of link,

(c) number of minor stops along the link,

(d) the distance of each stop from the A-NODE expressed a percentage of the total link distance,

(e) number of routes using the stop, and

(f) the name of each route.

The coding procedure is described more fully in Section 3 of the User's Manual.

2.10 Building a Sketch Network

Instead of increasing the number of nodes in a network, for some tasks, it may in fact be desirable to reduce the number of nodes in the network. The resulting network is then called a sketch network and the purpose is to speed up the processing of some of the programs in the transit model.

The technique consists of eliminating those nodes in the network which are non-essential. A node is regarded as non-essential if it has no branches and is not the beginning or end of a transit route. When a node is eliminated, the two links connecting to it are combined to form one link. The travel time through this new link is then equal to the sum of the travel time through the two old links. The resulting network will hence have fewer nodes and links.

An example of how to form a sketch network is illustrated in Figure 2.5.

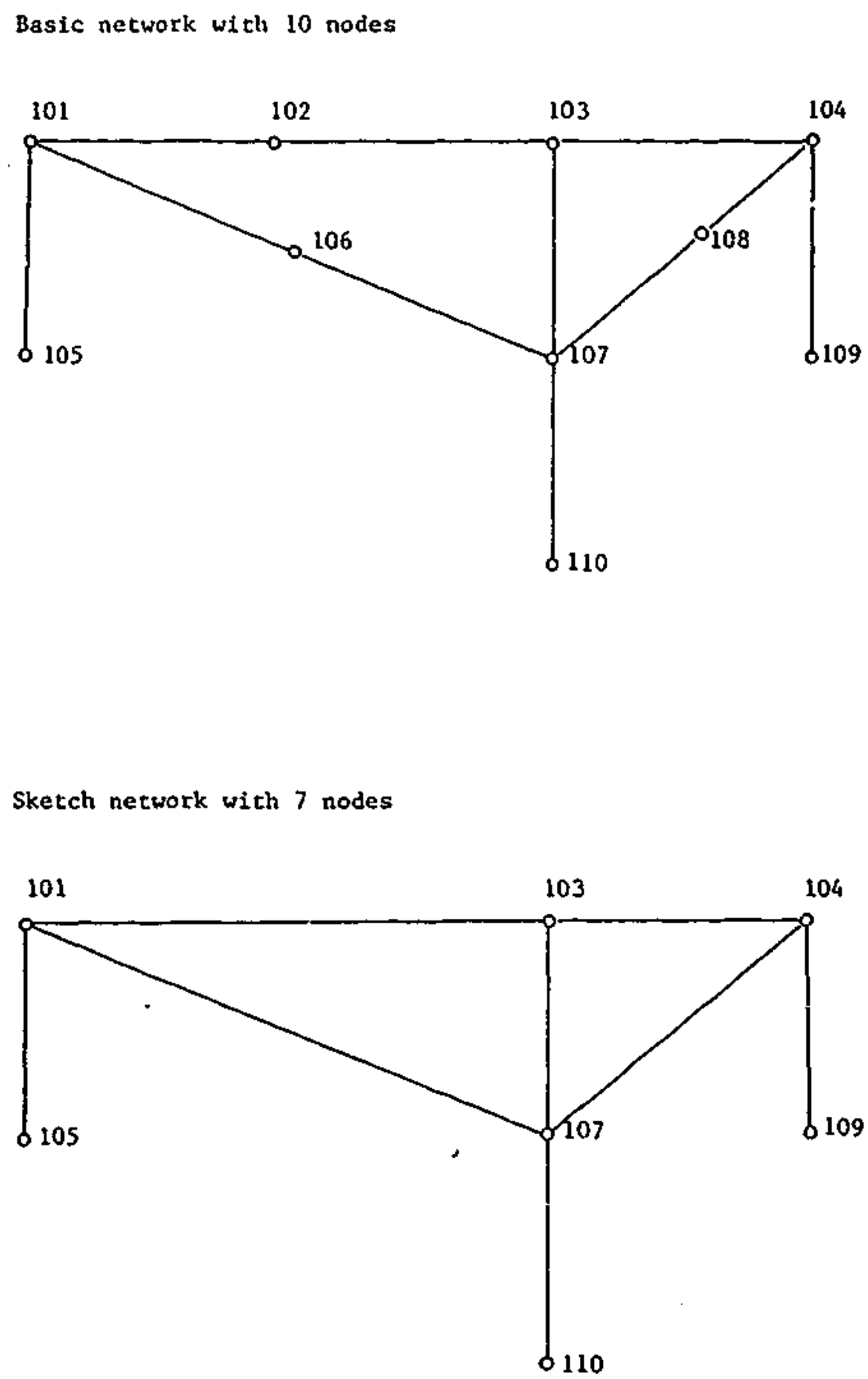


Figure 2.5 Building a sketch network

In programs for finding the minimum path between two nodes, a sketch network can be used instead of a basic network and the same results can be obtained. Since the program processing time is reduced using a sketch network, for tasks which require repetitive processing using the same network, such as the calculation of a trip time matrix between zones, it is advantageous to use this technique.

2.11 Generating a Reversed Network

In some minimum path problems, it is advantageous to be able to backtrack a path from its destination to its origin. For example, when the arrival time rather than starting time of a trip is specified, it is more efficient to start the path-finding algorithm from the destination end and backtrack towards the origin. Another example is when finding minimum paths from several different origins to the same destination, then it is more efficient to search for minimum paths in the reverse direction starting from the destination rather than starting from different origins and searching in the forward direction.

The technique of finding paths in the reverse direction can be done by generating a network with "reversed timetables". This means that the time schedules are reversed by using a time value equal to 24 hours minus the original time schedules. The network is then called a "reversed network". In a reversed network the directions of all transit arcs are also reversed. This means that all arc records are reversed by interchanging their A-NODEs and B-NODEs. The following example shows that moving through a reverse network in the forward direction is analogous to moving through a normal network in the reversed direction.

EXAMPLE: Generating paths in a reversed network.

Given the following arc from node 22 to node 23 of a normal network:

```
        22                            23
         *------------>-------------*
              LTIME = 3 minutes
              ST(1) = 8:00
              ST(2) = 8:10
              ST(3) = 8:24
```

the procedure for reversing the time schedules is shown below:

Departure time at node 22 (M)	Departure time at node 23 (N)	Reverse departure time at node 23 =(24:00 - N)
8:00	8:03	15:57
8:10	8:13	15:47
8:24	8:27	15:33

the following reversed arc is then formed:

```
        22                            23
         *------------<-------------*
               LTIME = 3 minute
               ST(1) = 15:33
               ST(2) = 15:47
               ST(3) = 15:57
```

A path can be traced in the forward direction through a normal network, e.g. if the starting time at node 22 is 8:10, then the departure time at node 23 can be traced through arc 22-23 as 8:10 + 3 = 8:13. A path can also be traced in the backward direction through a reversed network. For example, a starting time at node 23 of 8:13 is reversed to 24:00 - 8:13 = 15:47. Through the reversed arc 23-22, the departure time at node 22 is then equal to 15:47 + 3 = 15:50. The reversed time of 15:50 is 24:00 - 15:50 = 8:10. Hence, moving through a reverse network with reversed schedules is equivalent to back-tracking in a normal network.

2.12 Graphics

Network plotting by computer creates a visual description of the network and this can serve a variety of functions:

(a) It creates a visual description of the network and this could make the results of the network model more meaningful to the user, for example, by plotting out a network the model user can easily detect errors in network coding.

(b) It can be used in the printing or presentation of transit route information to guide travellers.

(c) It can be used in the graphic presentation of planning data, such as the results of trip assignment.

In recent years, several types of computer systems with graphics function have been developed. There is also a great range in the price and capability of various systems. For example, the Tektronix system produces high quality graphics but it is also expensive. On the other hand, some home computers, such as the IBM personal computer are also provided with graphics functions although the capability is lower compared with the more expensive Tektronix system.

In the present study, it was decided to develop the graphics software on an IBM personal computer. As the main objective of developing the software is for applications in passenger information systems, it is considered that the graphics programs must be able to run on microcomputers.

The programs have been developed in the high-resolution graphics mode which means that there are 640 horizontal points and 200 vertical points on the terminal screen. This represents the maximum size of the "window" or physical co-ordinate system that can be viewed at any one time on the screen. There is however an abstract co-ordinate system which is as large as 32768 by 32768 units and the "window" can be shifted to any part of this abstract co-ordinate system.

A graphics program for general network plotting has been developed to perform the following functions:

(a) plot out the entire network;

(b) enlarge a portion of the network; and

(c) plot a particular route of the network and show the location of vehicles running along the route when a time is specified.

Some of the pictures produced by the program are illustrated in Figure 2.6.

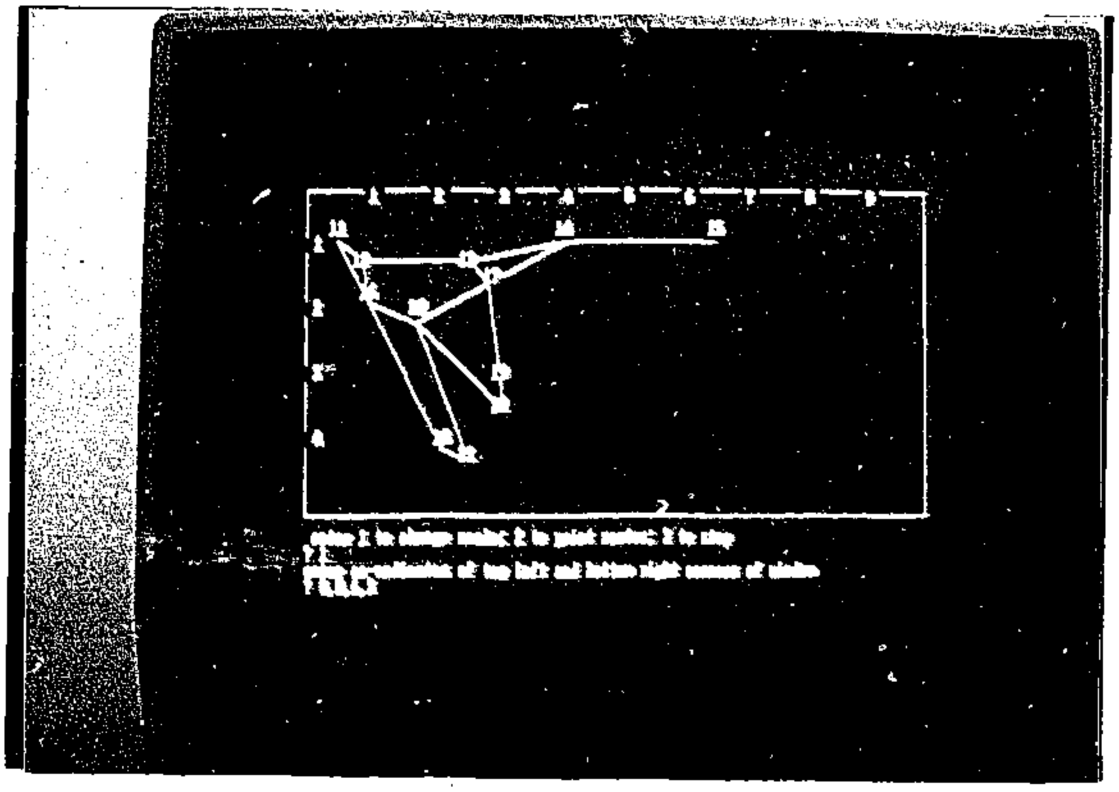


Figure 2.6A Network plotting by microcomputer

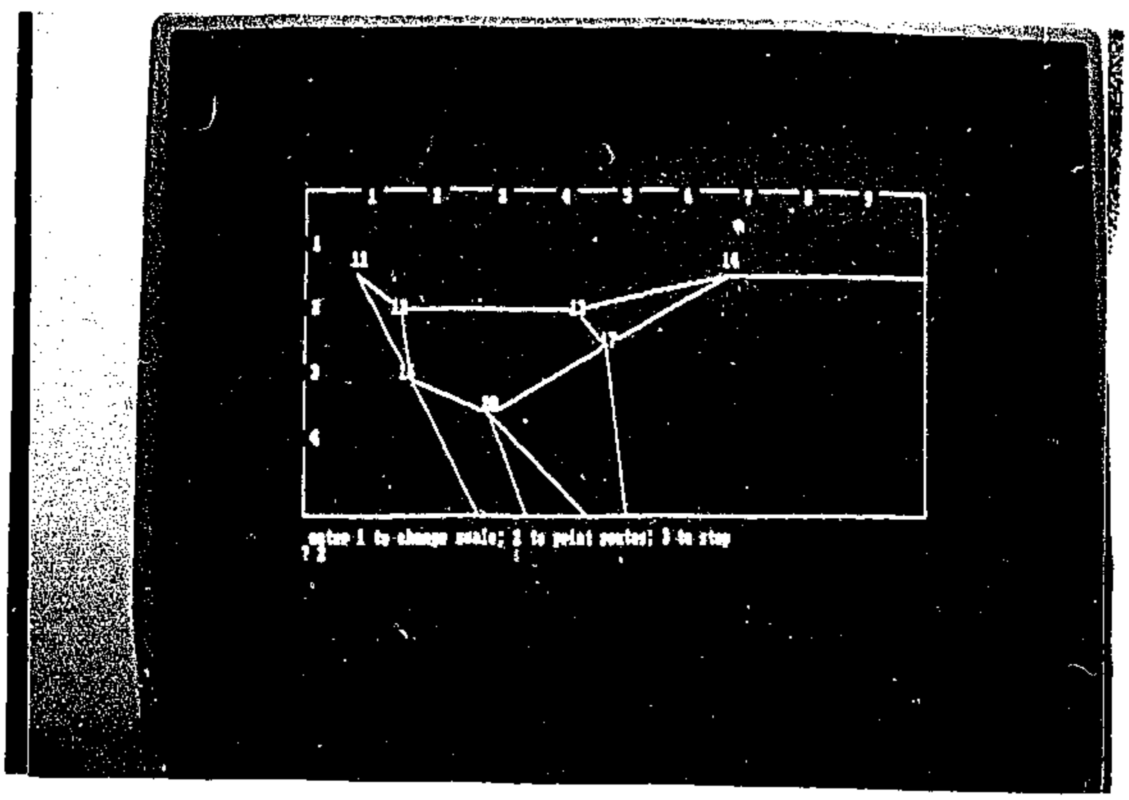


Figure 2.6B "Windowing" technique illustrated

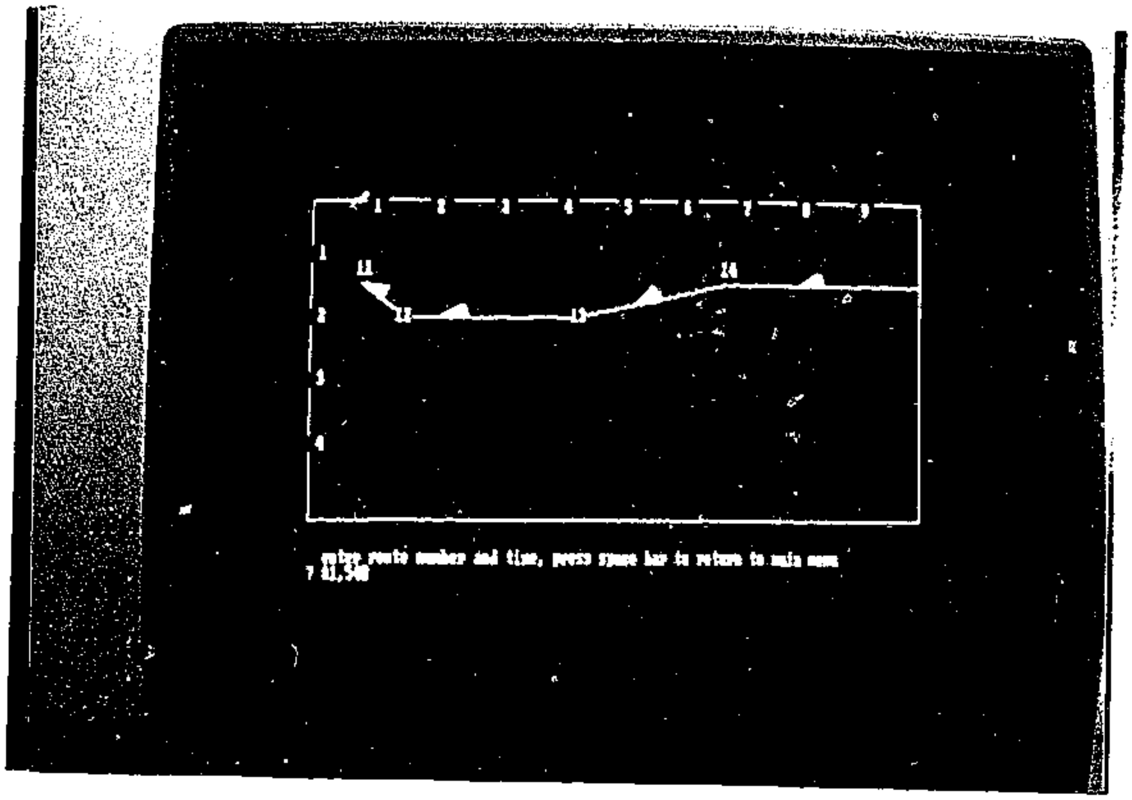


Figure 2.6C Location of vehicles along route A1 at 9:00 am (540 min.)

CHAPTER 3

TIME-DEPENDENT MINIMUM PATH ALGORITHMS

CHAPTER 3

TIME-DEPENDENT MINIMUM PATH ALGORITHMS

3.1 Review of Minimum Path Algorithms for Transit Networks

3.1.1 Introduction

Whilst minimum path algorithms are essential elements of a network model, they are often difficult to describe and comprehend. In order to learn a new algorithm or compare alternative ones, it is often necessary to apply the algorithms to actual examples or write computer programs incorporating the algorithms. Further, the correctness of an algorithm is often difficult to prove and errors may only be discovered through repeated testing of the algorithm with actual networks. For these reasons, it is often difficult to provide a comprehensive review on the subject.

However, good reviews have been provided by Dreyfus (1968) and Potts and Oliver (1972). According to Dreyfus, many authors had often overlooked previous reported results and there were many instances where recently reported results were inferior to older ones. Also, there had been cases of erroneous procedures being reported in the literature. Therefore, before proceeding with the development work on algorithms, particular care has been taken to review previous work where possible not to overlook the work reported by others.

This chapter first describes various types of minimum path problems that are applicable to transportation networks and the types of algorithms that have been suggested for solving them. In particular, previous research on those problems which are more closely related to the present scope of research, such as finding the best path based on various criteria and second best paths in scheduled networks are reviewed in more detail. Since the review showed that there had been little previous research in this area, it was decided to investigate the feasibility of developing new minimum path algorithms for scheduled networks. The results of the research are reported and the new algorithms which have been developed are described and illustrated with the aid of a worked example.

3.1.2 Criteria for Determining the Minimum Path

Different criteria can be used for determining the minimum path. As described in Newell (1980), the commonly used ones are trip time; trip distance; or a combination of the various attributes of a trip to form a generalised cost function. (For convenience, within the context of this study, these three types of minimum paths are called the quickest, shortest, and optimal path respectively.)

The criteria used for determining the minimum path can be represented by an objective function which expresses the separation between two nodes in a network as a function of one or more variables. The type of objective function used can have a great influence on the suitability of the algorithm. Bellman (1958) made an important contribution to the subject by showing that there are some objective functions which obey the Principle of Optimality and others which do not. The principle states:

"An optimal policy (series of decisions) has the property that whatever the preceding decisions or the initial state has been, the next decisions form an optimal policy for the situation that existed after the preceding decisions."

So if the Principle of Optimality holds, then once a minimum path to any node in the network is found, every other minimum path which passes through this node will use the same path from the origin up to that node. It is therefore possible to proceed in a step by step manner, first finding minimum paths to nodes which are closer to the origin and then proceed gradually from these nodes outwards to find minimum paths to other nodes.

The minimum paths found also have the "unique back node" property, which means that the minimum paths from one origin to all destinations can be traced by knowing the node behind each destination node along the minimum path. Algorithms which make use of this principle are often called "tree building algorithms" because the minimum paths formed from the origin to other nodes resemble a tree. Examples of commonly used algorithms belonging to this category are those invented by Dijkstra (1959) and Moore (1959).

In a minimum path problem, the objective function is used to define the separation from origin to destination of a path through the network. This separation can usually be expressed in terms of either distance, time or generalized cost. If it is also possible to use this objective function to define the separation of any link in the network, and each link separation thus defined is independent of the path through the link, and the separation of a path is equal to the sum of the separations of links making up the path, then the Principle of Optimality holds. However, if an objective function is used in which the separation of a link is variable and dependent on the path through the link, then the principle will not hold.

Bonsall (1976) has shown that transit networks with complex fare structures (e.g. the fare charged is not proportional to the distance travelled) do not obey the Principle of Optimality. Of the three types of objective junctions mentioned in the last section, quickest and shortest paths obey the principle whereas optimal paths do not. The reason for this is that in an optimal path, the separation may be considered as comprising journey time, waiting time and the inconvenience cost of route changes. Hence the total separation of a path cannot be obtained by summing up the separation of individual links making up the path.

When the Principle of Optimality does not hold, the minimum path has no "unique back node" property. This means that the minimum path to an intermediate node is not necessarily the minimum path to the destination node and hence it is not known which is the path to an intermediate node until the minimum path to the destination node has been found. So in this type of minimum path problem the algorithms invented by Dijkstra (1959) or Moore (1959) cannot be used. This problem has long been recognized and algorithms for finding optimal paths in transit networks have been reported by Dial (1967); Hewing and Hoffman (1969) and Clercq (1972). Further, Bonsall (1976) reported an algorithm for finding the optimal path in a transit network with complex fare structures. All these algorithms, however, cannot be applied to scheduled transit networks.

3.1.3 Scheduled Networks

A transit trip is affected by the scheduled nature of transit networks. Firstly, transit vehicles are not available on call and hence the transit trip must include an allowance for waiting time at the origin. Secondly, when changing from one route to another, allowance must be made for the co-ordination of the timetables on each route. Thus a route transfer waiting time must be included in the trip. Thirdly, because of the nature of fixed-schedule transit operations, the minimum path may vary throughout the day as the timetables on each of the routes are varied in accordance with passenger demand on the routes. The minimum path travel time will therefore be time-dependent. For these reasons, there is a need to model transit networks as scheduled networks.

In spite of the advantages of a schedule-based network, most previous researchers have often modelled transit networks as headway-based rather than schedule-based networks. For example, the following passage is quoted from Dial (1967):

> "The program could assume an actual bus schedule as input. But in so doing it would be imposing a hardship on both itself and on the planner. First, it would require the planner to produce these schedules for systems proposed for use 20 years hence- an exercise of questionable value. Second, the minimum time through a fixed schedule of stops varies with the time of departure from the home node. To find the minimum time over all departure times is an exercise of considerable worth, but only to computer suppliers, insofar as long-range transit planning is concerned."

It is interesting to note that Dial has specifically referred to the use of the models only for long-range planning in which case it would be appropriate to use a headway-based model. However, when it is required to use the models for other tasks, such as short-range planning or the generation of passenger information, then it is necessary to develop models with scheduled networks.

Dreyfus (1968) has included scheduled networks in his review of minimum path algorithms. He mentioned an algorithm developed by Cooke and Halsey (1966) which deals with this type of problem but suggested that a Dijkstra type algorithm can in fact be adopted for finding the quickest path in scheduled networks and work more efficiently than the Cooke and Halsey algorithm.

Other papers which have referred to the use of minimum path algorithms in scheduled networks are mostly connected with software developed for passenger information systems. For example, Pickett (1980) mentioned the development of computer software for finding the quickest, cheapest and easiest (i.e. with the least number of interchanges) path in scheduled networks. While the need for the development of more advanced path-finding algorithms, such as optimal path algorithms which minimizes a generalised cost function, is generally acknowledged, no such algorithms have been reported.

3.1.4 Tree Building and Matrix Algorithms

A common way of categorizing minimum path algorithms is the number of paths which can be found from one iteration of the algorithm. In one iteration, a single path algorithm finds a minimum path from one origin to one destination; a tree building algorithm can find minimum paths from one origin to all destinations in one iteration and a matrix algorithm can find minimum paths between all origins and destinations in the network. Obviously, a algorithm with lower path-finding capability can always substitute that of a higher one. For example, a tree building algorithm can be used as a matrix algorithm by repeating the operation for different origins. Although in this case it may then not be as efficient compared to a matrix algorithm.

Whilst all the algorithms mentioned previously in this Chapter are tree building algorithms some matrix algorithms for finding quickest paths have been reported by Floyd (1962) and Dantzig (1966). These two algorithms have been reviewed by Dreyfus (1968) who said that if a Dijkstra type algorithm is used repeatedly instead of applying these algorithms, then the computing time will be increased by a factor of approximately one and a half.

3.1.5 Second Best Paths

All the algorithms mentioned above can only be used to find the best path but not the second best or in the case of several equally best paths, only one is found. The first algorithm developed to find the second best paths has been reported by Hoffman and Pavley (1959). The algorithm analyses deviations from the best path for finding the second best. Modifications to the Hoffman and Pavley algorithm have been suggested by Bellman and Kalaba (1960) and Dreyfus (1968). The procedures described are of necessity tedious and lengthy and it is not possible to find from the research literature any report on the application of these algorithms to transport network planning models.

When generating paths to provide travel guidance information there is sometimes a need to find second best paths. The algorithm must also be efficient if it is required to generate the travel information in the interactive mode.

3.1.6 Summary

The review shows that minimum path algorithms for conventional networks have been well developed and there is probably little scope or need for further improvement. However, there has been little research on minimum path algorithms for scheduled transit networks and especially when weighted trip time instead of trip time is used as the objective function. Further, there is a need to develop an efficient algorithm to find the second (or third, etc.) paths in these networks.

3.2 Scope and Objective

The objective of the research is to develop minimum path algorithms for scheduled networks. Two types of minimum path algorithms have been developed: the first is used to find the quickest path and the second to find an optimal path with minimum weighted trip time.

These algorithms can be seen to have several applications in the development of transit planning models. First, they can be used in assignment modelling and have resulted in the development of more accurate assignment procedures. Second, using the algorithms, it is possible to show how journey time is related to the starting time of a journey (or the required arrival time), and thus are valuable tools in the analysis of the temporal variations in accessibility for trips made by transit systems. Finally, they can be used to provide travellers with travel information, such as the minimum path within a transit network, and route boarding times along the path.

3.3 Quickest Path Algorithm

3.3.1 Introduction

The first task of the study is to develop a quickest path algorithm for scheduled transit networks. It is known that some of the commonly used quickest path algorithms for non-scheduled networks, such as those invented by Dijkstra (1959) and Moore (1959), can be modified for use in schedule networks. Based on these algorithms, computer software was developed for this study to determine the quickest paths in scheduled transit networks. The software was tested using the rail, tram and government operated bus networks of Melbourne, Australia and found to give correct results. Preliminary comparison showed Dijkstra's algorithm to be slightly more efficient than Moore's. It was decided therefore to adopt the Dijkstra type algorithm in the software.

3.3.2 Dijkstra's Algorithm

The principle used in Dijkstra's algorithm will first be described, as this forms the basis of the new quickest path algorithm developed. In a Dijkstra type algorithm, all nodes in the network are assigned a label which contains the following two quantities:

(a) the arrival time at the node by the path considered; and

(b) the previous node along the path, which is called the back node.

The algorithm is essentially a node labelling procedure, which fills or updates node labels, starting from the origin and moving progressively to nodes further away from the origin. An empty node label is always filled when a path to the node is found. However, a node with a filled label is updated only if a quicker alternative path to the node is found. When it is certain that the minimum path to a node is found, its label is declared permanent. All other labels which have not yet been declared permanent are regarded as temporary. In the first step, the label of the origin node is filled by setting its arrival time to zero and there is no back node. This label is then declared permanent. Then, for every node which is linked to the origin node, the arrival time is calculated and its label filled. In this case, the back node is the origin node. In the second step, all temporary labels are compared and the one with the earliest arrival time declared permanent. After a node is permanently labelled, those nodes which are linked to it and have a temporary label are compared and their labels either filled or updated where appropriate. The procedures in this second step are then iterated. As only one label will be declared permanent at any one step, so it will take N steps to complete the algorithm, where N is the number of nodes in the network. Once the algorithm is completed, it is then possible to find the quickest path from the origin node to all other nodes in the network.

3.3.3 A Time-Dependent Quickest Path Algorithm

With some modifications, a Dijkstra type algorithm can also be used to find quickest paths in time-dependent scheduled transit networks. The required modifications are described in the following paragraphs.

Firstly, instead of starting at zero time at the origin node, the actual clock time is used. The arrival time at various nodes in the network will be recorded as the actual clock time.

Secondly, the length of an arc is dependent on the arrival time at the beginning of the arc. Hence, in order to determine the shortest travel time from a node A to another node B which is adjacent to A, the procedure is to search through the network file and find all arcs which have their A-NODE equal to node A and their B-NODE equal to node B, and then for each of these arcs in turn read their ST(J)s and obtain the first ST(J) which is greater than the arrival time at node A. The arrival time at node B when travelling by this arc is then equal to the appropriate ST(J) plus the LTIME of the arc. The arc which gives the earliest arrival time at node B is chosen as the quickest path from node A to B.

Lastly, when a route transfer is made at a node, then a transfer time is added to the arrival time at the node. It is possible to have different transfer times for different nodes or different types of nodes, e.g. nodes at rail stations can have different transfer times from nodes at tram or bus stops. The reason is that the walking times between services at railway stations are usually longer than those at tram or bus stops.

Variation of the transfer time at a node could also be used to account for the unpunctuality or unreliability of services arriving at a node. A simple approach might be to ascribe a level of "lateness" on a route-specific, mode-specific or area-specific basis and to apply this constant "lateness" to the "arrival mode" in each transfer situation. This approach is based on the observation that services usually run late rather than early. Due to lack of actual data, this subject was not studied in greater detail. However, with minor modifications, the quickest path algorithm developed can allow for variable transfer times.

3.3.4 Different Types of Quickest Path Problems

The quickest path algorithm described is a tree building algorithm, hence one execution of the algorithm will find the quickest paths for all journeys from a single origin node to all other nodes in the network. It can of course be used to find the quickest path from a single origin node to a single destination node in a single execution. In this case the computation will stop once that destination node is reached, and the algorithm will nevertheless have calculated the minimum journey times to all other nodes which have a shorter journey time than the specified destination node.

With some modifications the algorithm can also be used to find the quickest paths from all nodes to a single destination. With this type of problem, the "reversed timetable" technique can be used. The method, as described in Chapter 2, is to modify the network file so that all routes are assumed to be running in the reverse direction and time schedules are reversed by replacing all times by a value equal to 24 hours minus the scheduled time. In this way a "one origin to all destinations" type problem is converted to an "all origins to one destination" type problem. It is interesting to note that with a normal network the quickest path found is based on a given starting time from the origin. However, if a reversed network is used, then the quickest paths found are based on a given arrival time at the destination. This reversed timetable procedure has a number of useful applications. For example, it could be used for planning trips to participate in activities requiring arrival at specific times, such as going to a concert or a business appointment. This procedure is also an essential part of the optimal path algorithm described in Section 3.4.

The use of transfer time to account for unpunctuality could also be achieved in the reversed network. However, in this case it is the "lateness" of the "pseudo-departure mode" which needs to be considered.

If it is required to find quickest paths from all nodes to all other nodes, then this can be obtained by N executions of the algorithm where N is equal to the number of nodes in the network. For quickest paths which are not time-dependent, then as discussed in Section 3.1.4, a Floyd type algorithm (Floyd, 1962) has been found to be more efficient. However, it cannot be used if quickest paths are time-dependent because in such cases the Principle of Optimality will not hold when the algorithm is dealing with paths starting from more than one origin at the same time.

3.4 Optimal Path Algorithm

3.4.1 Introduction

The above section has described the development of a quickest path algorithm for a scheduled transit network. While such an algorithm is considered to have many useful applications in transit network models, it is commonly acknowledged that route choice may not depend purely on trip time but on other factors such as the number of route changes, waiting time, preference for certain modes and travel cost. The generalised cost concept can be used to account for these factors by assigning weightings to the various components of journey time and cost. There has already been much previous research on the various methods that can be used for obtaining a set of weights for the generalised cost function. A brief review of this subject is given in Section 4.1.3.1.

This section will describe the development of an optimal path algorithm for transit networks. The algorithm adopts an objective function in which the weighted trip time consists of the following four components:

(a) in-vehicle time,

(b) waiting time,

(c) walking time, and

(d) route change penality (expressed in terms of time units).

Each component can be weighted differently to form a total weighted trip time. Travel costs have been excluded from the function because it is found the fare structures for transit systems are often complex and variable. The use of season tickets, concessions for students and for travel during off-peak hours can introduce much variability to travel costs. The other problem is that even if a minimum cost algorithm is developed for a particular type of fare structure, it could become out-dated as soon as the fare structure changes. Past experience show that transit operators do change fare structures quite frequently. It is acknowledged, however, that travel cost is often an important considration in route selection. The development of an optimal path algorithm that can minimise generalised cost is considered to be highly desirable when it is to be applied to the planning of a specific transit system and when the fare structure is known.

3.4.2 New Algorithm

As mentioned in Section 3.1, although optimal path algorithms for transit networks have already been developed by others, these are generally not applicable to scheduled transit networks. It was therefore decided in this study to develop a new optimal path algorithm for transit networks.

Generally, weighted trip time is not a linear function of time. This means that weighted trip time does not increase uniformly with trip time. For example, walking time and waiting time normally have a higher weighting than time spent travelling in other modes. The weighted time function may also be stepped because whenever a route change is made there may be a sudden increase in weighted time due to the reluctance of travellers to change route. In these cases the Principle of Optimality will not hold and the algorithm will hence become more complex.

The new algorithm adopts a path generation procedure. Starting from the origin node, various paths are generated according to a set of rules. These rules help to restrict the number of paths that can be generated. After all feasible paths are generated, the optimal one is then selected.

The following three rules has been used to guide the path generation procedure:

(a) identification of inaccessible nodes;

(b) definition of efficient path; and

(c) path elimination.

These rules will be explained in detail in the following three sub-sections.

As the number of paths generated could be quite large, path storage is another important element of the algorithm. The special tree structure used for the storing of these paths is described in Section 3.4.6.

In order to explain in detail the procedures adopted in the algorithm, a step by step description of the algorithm is shown in Appendix 3.

3.4.3 Identification of Inaccessible Nodes

The algorithm is concerned with finding the optimal path from an origin node to a destination node. Inaccessible nodes are defined as those nodes in the network which have no chance of being included in the optimal path.

Given a starting time at the origin, a quickest path algorithm can be used to find the quickest path from origin to destination. The weighted time of this quickest path is then calculated, using weights for the various components of the journey as specified by the program user. This weighted time is then assumed as real time and taken as the latest permissible arrival time at the destination. This means that it is permissible for an optimal path to be slower than others in real time provided that it is more attractive in other respects. So if a node cannot be included in any path which goes from origin to destination within a time period between the starting time and "weighted finish time", the node is considered inaccessible.

The method used to identify inaccessible nodes is quite straight-forward. From the output of the quickest path algorithm, the "earliest arrival time" at every intermediate node in the network when starting from the origin at the specified starting time is known. The quickest path algorithm is executed a second time using the "reversed timetable" technique and a specified "weighted finish time" at the destination. From the output the "latest departure time" at every intermediate node so as to arrive at the destination not later than the "weighted finish time" is known. If a node has its "earliest arrival time" later than its "latest departure time", then this node is considered as inaccessible. It is obvious that the method will not work if the weighting coefficients of the weighted time function are less than unity. This condition, however, is not commonly found in reported studies of actual traveller behaviour. unnecessary

By identifying inaccessible nodes in the network, the number of alternative paths that can be generated is greatly reduced.

3.4.4 Definition of Efficient Path

During the tree-building process some conditions are set so that only efficient paths are generated. An efficient path is normally defined as one which does not backtrack. The condition as set by Dial (1971) is that as the path progresses from node to node, it always moves further from the origin and closer to the destination. (Either distance or journey time can be used to denote the meaning of closer or further.) This condition can be relaxed so that a path is considered efficient if it either moves further away from the origin or moves closer to the destination. The above definitions can work quite well for highway networks, however this is not so for transit networks. The reason is that transit routes are often much less direct, due to the restriction of available transit routes and transit schedules and the desire to avoid waiting, walking and transfer times. In order to reduce the chance of missing out on a path that seems indirect and yet is the optimal, it is necessary to set the condition for path generation as unrestrictive as possible provided that the condition is sufficient to prevent looping.

After some experimentation with the Melbourne transit network, two possible definitions for an efficient path were considered. These are described below:

(a) Every alternative path begins with the initial condition that as the path progresses from node to node it must always move further from the origin, (the meaning of further is defined by "earliest arrival time"). When a path cannot proceed any further with this condition, the condition is changed so that the path should then always be moving closer to the destination, (as defined by "latest departure time").

(b) Every alternative path begins with the initial condition that as the path progresses from node to node, it must always move further from the origin. After the weighted time of a path has exceeded half of the weighted time of the quickest path then the condition is changed so that the path should then always be moving closer to the destination.

The first definition was found to be less restrictive than the second and hence could generate a larger number of paths for consideration. However, it may also cause the generation of some looping paths. One type of looping which is often detected with this definition is that when waiting time is weighted higher than walking time and there is a long wait before boarding at the first stop, a path which walks to and fro between the origin and first transit stop is generated. In this way, waiting time is converted to walking time and results in a lower weighted trip time. This type of path, although actually within the criteria set, may not be admissible especially when used as guidance information for travellers.

The second definition is more restrictive than the first but it eliminates the looping error described. Although the chance of missing the optimal path is slightly higher, this is considered to be more suitable when used in path-finding software of passenger information systems.

3.4.5 Path Elimination

As the algorithm progresses from node to node, starting from the origin, both the arrival time and weighted time at each node are computed and stored. As discussed previously, when there are alternative paths to an intermediate node, paths cannot be discarded simply because they arrive later or have a larger weighted time than the optimal. A simple example will illustrate this, suppose one path arrives at an intermediate node at 9:00 am and another path arrives at the same node at 9:10 am. Both paths must be retained for further consideration since if the next vehicle departure at the node is at 9:12 am then the second path will require a smaller waiting time and hence could result in a smaller total weighted trip time. Hence, it is not possible to discard a path even if both its arrival time and weighted time are found to be greater than another, due to the nature of the weighted time function which usually has a large weighting on waiting time and route changes.

Nevertheless, there are some criteria which can be set for the elimination of paths to intermediate nodes, and these are described below:

(a) If two paths arrive at the same node at the same clock time, then the path with the larger weighted time is discarded.

(b) If the arrival time at a node is later than the "latest departure time" of the node, the path is discarded.

(c) On arrival at a node, the weighted trip time consumed by the path up to this point is computed. This is subtracted from the total maximum weighted trip time budgeted for the trip to obtain the remaining weighted trip time budget. If this is either less than the quickest trip time from the intermediate node to the destination or less than the "weighted finish time" minus the "latest departure time" at the node, then the path is discarded.

The above three conditions are incorporated into the algorithm for the elimination of paths to intermediate nodes. These conditions are found to occur quite frequently in actual transit networks and hence its adoption resulted in an increase in the efficiency of the algorithm.

3.4.6 Tree Structure

The algorithm builds a tree not in the normal sense of representing optimal paths from one origin to all destinations. Rather, the tree represents alternative paths from one origin to one destination. From this tree, it is then possible to re-construct the optimal path by searching for the end node with the minimum weighted time and backtracking along the tree until the path reaches the origin. In fact, if the path elimination function is taken out from the algorithm, then the tree can also be used to backtrack any required number of paths ranked according to weighted time.

3.4.7 Different Types of Optimal Path Problems

One execution of the algorithm developed finds the optimal path from one origin to one destination with a specified starting time. In order to find the optimal path from one origin to all destinations, it is necessary to repeat the algorithm for each destination.

By using the reversed network technique, the algorithm can also be modified to find the optimal path from one origin to one destination with a specified arrival time.

3.5 Worked Example

3.5.1 Network

A worked example is used to illustrate the procedures used in the quickest path and optimal path algorithms. The example assumes a hypothetical transit network as shown in Figure 3.1. There are a total of 10 routes operating in the network and their schedules are shown in Appendix 2.

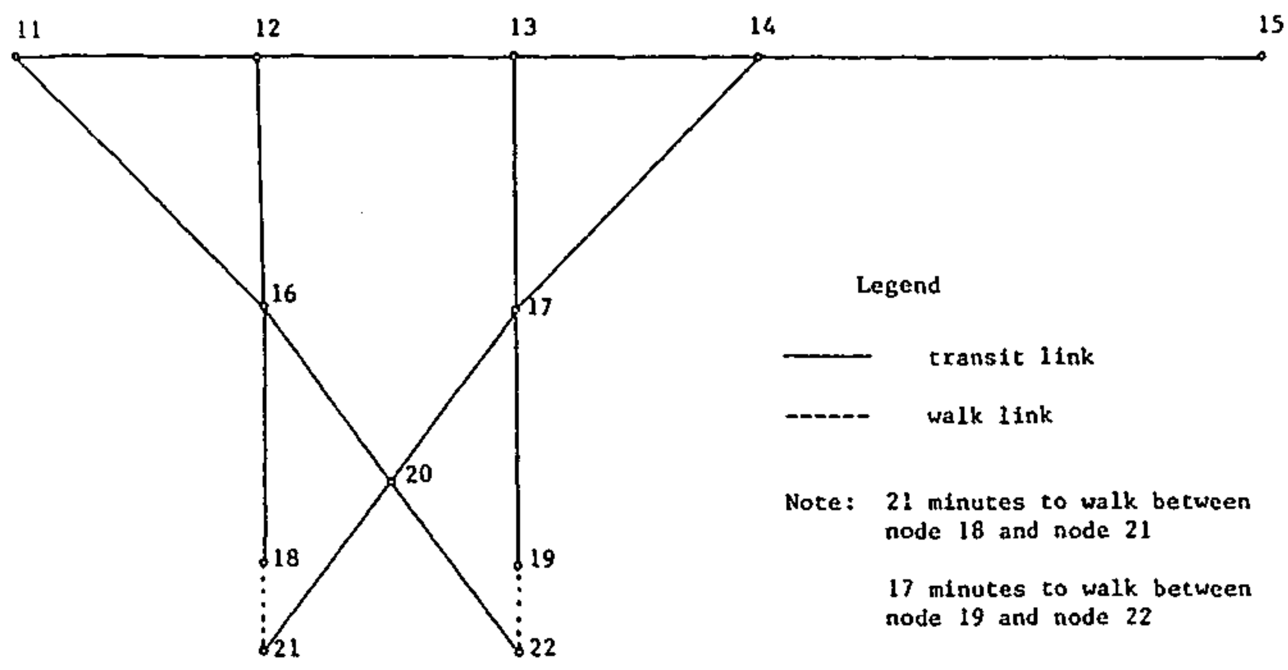


Figure 3.1 Hypothetical transit network

3.5.2 Finding the Quickest Path

Using the data available, a network file is generated. Suppose node 21 is chosen as the origin and the starting time of the trip is 7:15 am, then one execution of the quickest path algorithm will output the following tree:

Destination node	Previous node	Earliest arrival time	Route used at arrival
11	16	7 54	A5
12	11	8 10	A1
13	17	8 5	B2
14	17	7 54	A4
15	14	8 20	A1
16	20	7 39	A5
17	20	7 39	A4
18	21	7 36	WALK
19	22	7 57	WALK
20	21	7 29	A4
21	0	7 15	
22	20	7 40	B5

From this tree the quickest path from node 21 to any other node in the network can be backtracked, for example the quickest path from node 21 to node 12 is given below:

Node	Arrive time	Depart time	Route used at arrival
21	7:15	7:19	
20	7:29	7:29	A4
16	7:39	7:39	A5
11	7:54	8:00	A5
12	8:10	--	A1

The above shows how to find the quickest path from node 21 to node 12. However, the path may not be an optimal path for a particular traveller because it involves 2 route changes. Suppose it is assumed that the weightings placed on walking time and waiting time are both equal to 2 and that each route change is equivalent to a penalty of 5 minutes, then the weighted time of the quickest path shown above is calculated as follows:

```
in vehicle time          = 45 minutes
walking time             =  0 minutes
waiting time             = 10 minutes
number of route changes  = 2
weighted trip time       = 45 + 10x2 + 2x5 = 75 minutes
```

3.5.3 Finding the Optimal Path

In order to find the optimal path from node 21 to node 12, the quickest path algorithm is executed using the "reversed timetable" technique for a "weighted finish time" of 8:30 am. This will calculate the latest departure time at all nodes in order to arrive at the destination at 8:30 am. The results will be used for the elimination of inaccessible nodes. As explained in Section 3.4.3, a node is considered to be inaccessible if the arrival time at the node is later than the departure time at the node. The procedure is illustrated below:

Node	Earliest arrival time when starting from node 21 at 7:15 am	Latest departure time to arrive at node 12 not later than 8:30 am	Is node access-ible ?
11	7:54	8:17	yes
12	8:10	8:30	yes
13	8:05	8:20	yes
14	7:54	8:10	yes
15	8:20	7:50	no
16	7:39	8:02	yes
17	7:39	7:55	yes
18	7:36	7:52	yes
19	7:57	7:45	no
20	7:29	7:44	yes
21	7:15	7:34	yes
22	7:40	7:34	no

In this case only 3 nodes are eliminated and the resulting network is shown in Figure 3.2.

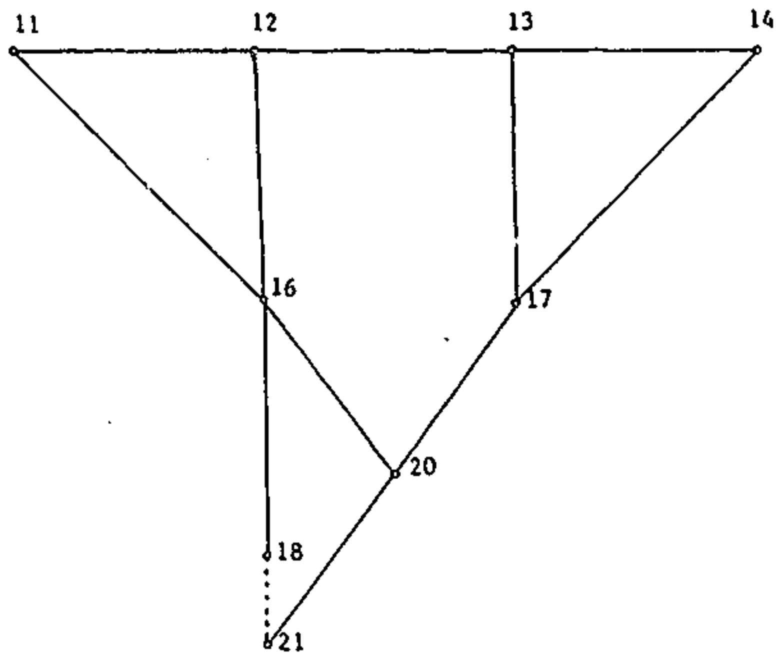


Figure 3.2 Simplified transit network

In this case the network is not greatly simplified. However, if the network is larger then the reduction will generally be significant.

The algorithm then builds a tree in the form shown below:

Node no.	Arrive time	Weighted time	Route used at arrival	Record number	Pointer
21	7:15	-5		1	
18	7:36	37	WALK	2	1
20	7:29	18	A4	3	1
16	7:39	33	A3	4	3
17	7:39	28	A4	5	3
11	7:54	48	A5	6	4
14	7:54	43	A4	7	5
12	8:10	75	A1	8	6
13	8:05	60	B1	9	7
12	8:15	70	B1	10	9

The method of tree building and data storage is described as follows: every time a path is extended to a new node, a new record is added to the tree. Each record contains the following information:

(a) the node number;

(b) the arrival time at the node;

(c) the weighted time of the path on arrival;

(d) the route used to arrive at the node;

(e) the record number; and

(f) a pointer which shows the record number of the preceding node in the path.

When the tree-building process is completed, alternative paths can be traced from the path by means of backtracking. The optimal path is obtained by searching through the tree to find all records with node numbers equal to the destination node, and selecting the one which has the smallest weighted time. Starting from this record, the optimal path can be backtracked by following the pointers in successive records. The arrows drawn in the tree above show how the optimal path can be obtained by backtracking. The optimal path from node 21 to node 12 is then shown below:

Node	Arrive time	Depart time	Route used at arrival
21	7:15	7:19	
20	7:29	7:29	A4
17	7:39	7:39	A4
14	7:54	7:55	A4
13	8:05	8:05	B1
12	8:15		B1

in-vehicle time = 55 min.
waiting time = 5 min.
walking time = 0 min.
route changes = 1

weighted time = 55 + 5x2 + 1x5
= 70 min.

The second best path can also be backtracked from the tree and it can easily be seen that it is the same as the quickest path found previously. In this case, other paths ranked according to weighted time cannot be backtracked from the tree because the weighted time of these paths has exceeded 75 minutes. If it is required to obtain these other paths, then it is necessary to rerun the quickest path program with reversed timetable using a larger weighted time.

3.6 Package of Programs

The above example can only illustrate the basic functions of the algorithms. As discussed, many other functions can be performed from these algorithms on different types of networks. The various algorithms described have been incorporated into a package of programs. Using this package, various path-finding problems can be solved. A User's Manual to these programs is contained in Volume 2 of this thesis.

CHAPTER 4

IMPROVED TRANSIT ASSIGNMENT TECHNIQUES

CHAPTER 4

IMPROVED TRANSIT ASSIGNMENT TECHNIQUES

4.1 Evolution of Trip Assignment Methods

4.1.1 Introduction

Trip assignment is an essential stage in the transport planning process. It consists of the prediction of travellers' route choices and the allocation of trips onto such routes. Over the past thirty years, trip assignment methods have evolved from simple manual procedures to the development of complicated algorithms performed by computer. The literature on assignment is extensive and it is not intended to provide a detailed review in this thesis. Rather, a brief account of the evolution of assignment methods will be given. Detailed reviews can be found in Martin et.al. (1963), Van Vliet and Dow (1979).

In the past, transport planners have been more concerned with the upgrading of highway networks rather than transit networks (Lamb, 1970). As a result, research has been focused on highway rather than transit assignment. A literature search has shown that the number of publications on highway assignment is much larger than those on transit assignment. Since the two topics are related, however, a brief account of the evolution of both types of assignment is provided in this section.

4.1.2 Highway Assignment

4.1.2.1 Diversion Curve Method

The earliest researchers on highway assignment were mainly concerned with the factors that influence a traveller's route choice. For example, surveys carried out by the U.S. California Division of Highways (Moskowitz, 1956) had shown that travel time and distance were the most significant factors which influenced the motorist's route choice. Using these survey results, diversion curves such as those shown in Figure 4.1 could be constructed and used as a basis for allocating flows to alternative routes.

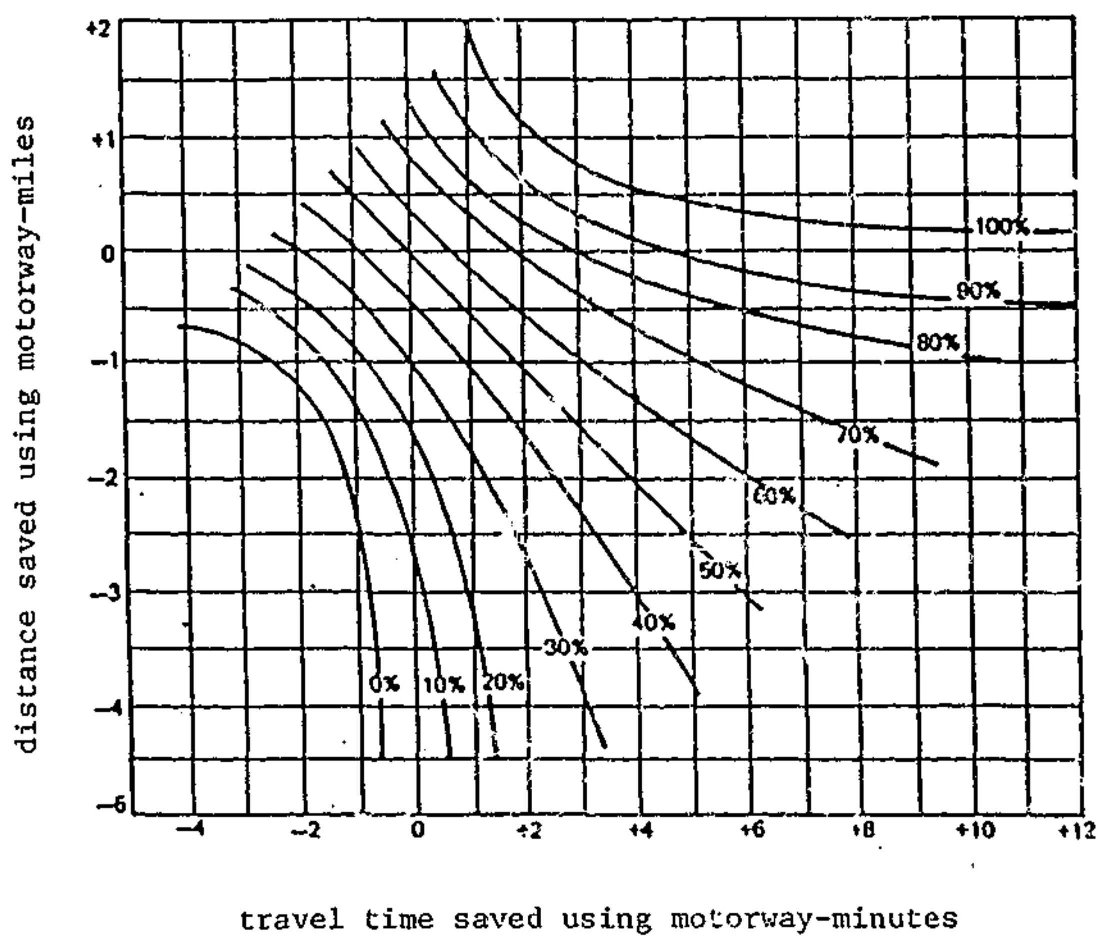


Figure 4.1 Travel time and distance saved diversion curve (Moskowitz, 1956)

This diversion curve method is suited mainly for estimating the amount of traffic that would be attracted to a proposed new freeway and when the number of alternative routes to be compared is small. If it is required to evaluate a complex urban highway network, where the number of alternative routes for traversing the network is large, this method of assignment which required the enumeration of alternative paths would need very lengthy computations.

The various types of assignment methods that were subsequently developed were greatly influenced by the capabilities of algorithms that were available for finding minimum paths in networks. The minimum path algorithm devised by Dijkstra (1959), for example, can be used very efficiently for finding the minimum paths from one node to all other nodes in the network. However, when it is required to find not only the minimum, but the second, and succeeding minimum paths, or to find all equal minimum paths when more than one is present, other more lengthy and inefficient procedures are required. For this reason, most assignment methods used in practice only require finding the minimum path between each origin-destination pair.

4.1.2.2 All-Or-Nothing Assignment

The "all-or-nothing" assignment method is so called because all traffic between an origin-destination pair is assigned to the minimum path. Trip time is normally used as the quantity to be minimized although other factors such as distance and cost or a combination of these factors can also be adopted

(Lai and Van Vliet, 1979).

A problem which is often encountered in assignment procedures is that the flow and speed of a link are interrelated. However, before the assignment, link flows are not known and hence link speeds can only be obtained based on some prediction of link flows. After one iteration of the assignment, a set of link flows are obtained and it is then possible to re-adjust the link speeds based on the new flows and perform a second iteration of the assignment. This iterative process also helps to divert flows from congested links to less congested ones. This process is described as "capacity restraint" assignment (Smock, 1962).

This approach can be viewed as an equilibrium solution between supply of and demand for road space. It is important to obtain a convergent process. This means that speed and flow values should stabilize as a result of successive iterations. Various heuristic procedures have been proposed for accelerating the converging process. These include total or partial loading of flows onto the network during each iteration (Martin and Manheim, 1965); or random assignment in which a flow related speed adjustment is made after assignment to each randomly selected O-D pair (Schneider, 1956). However, as discussed in Van Vliet and Dow (1979), there is no guarantee of a convergent solution in these types of assignment procedures.

The assumption underlying the "all-or-nothing" methods is not strictly correct because survey results have demonstrated that not all travellers choose the minimum path (Moskowitz, 1956). This may result from the differences between "measured" and "perceived" values of travel time and the effect of other unquantifiable factors such as comfort and environmental conditions on different routes. The method is therefore likely to result in an over estimate of flows along minimum paths for uncongested networks. For congested networks, although "capacity restraint" can be used to provide a more even distribution of flows among alternative paths, as discussed earlier, the method does not guarantee a convergent solution, and the procedure of obtaining resultant flows is heuristic and hence not guaranteed to be consistent with observed behaviour (Huber, Boutwell and Witheford, 1968).

Nevertheless, the following advantages make this method commonly used at present:

(a) Only the minimum path is required and hence the algorithm is relatively simple;

(b) The results can be used as a broad indication of the performance of the highway network; and

(c) It can make allowance for congestion effects on route choice.

4.1.2.3 Stochastic Methods

A third category of network assignment methods which are designed to account for the non-optimal behaviour of travellers has also been developed. These are often called stochastic assignment methods. Within this category, the methods proposed by Burrell (1968) and Dial (1971) are the most well known.

In Dial's method, two passes of a minimum path algorithm are used: a forward pass starting from the origin and a backward pass starting from the destination. From these, the minimum travel time from the origin to any node and from any node to the destination can be found. This information is then used in determining the probability of a node being used and hence the ratio in which flows are apportioned between various paths. A logit type model (Dial, 1971) was used for determining the apportioning factor. A rule was also set for deciding whether a path is reasonable or not in order to prevent path looping and backtracking. The method is computationally efficient because the assignment and path finding processes are combined and it is not necessary to enumerate paths.

The most important feature of Dial's method is that it makes efficient use of the information available from a minimum path algorithm. Dijkstra's algorithm, for example, can be used to calculate the minimum travel time (or distance or cost, depending on the criterion used) from the origin to every node in the network as well as the minimum path to the destination.

Although Dial's method is quite frequently used in practice, the method however contained two inconsistencies (Burrell, 1976 and Daganzo, 1977). First, as with other logit type models, it assumes that all alternatives are equally independent. Alternative paths, however, may not be equally independent due to the possibility of different degrees of path over-lapping. Second, Dial's definition of a reasonable path is not fail-safe as this may sometimes wrongly classify reasonable paths as unreasonable.

Burrell's method assumes that link times are not single valued but can vary according to an assigned distribution. Thus, for every origin, a set of link times is randomly generated according to the given distribution, and the minimum paths to all other destinations found. Another set of link times are then generated for another origin. When there are a large number of origin-destination pairs, the effect would be a more even distribution of flows among different links in the network. Even if the number of origin-destination pairs is small, a satisfactory result can still be obtained by generating more than one set of link times for each origin and iterating the assignment procedure. This would however increase the computation effort substantially.

The major drawback of both Dial's and Burrell's methods is that "capacity restraint" would again have to be used to model the effect of road congestion on route choice. A convergent solution is not guaranteed in these processes (Van Vliet and Dow, 1979).

4.1.2.4 Equilibrium Methods

The inability of the previously described methods to guarantee convergent solutions, especially for congested networks, has led to the development of "equilibrium methods" of assignment. These methods consist of defining the equilibrium criterion and finding convergent equilibrium solutions.

The most commonly used equilibrium criterion is a condition known as Wardrop's first principle (Wardrop, 1952). This assumes a user optimized equilibrium condition in which all alternative paths in a network are fully utilised so that no individual user can reduce his travel time by changing to another route. Although Beckmann et.al. (1956) had pointed out that finding a solution to this assumed equilibrium condition is a straightforward optimization problem, it was not until many years later that algorithms for finding the equilibrium solution (e.g. Dafermos, 1971; LeBlanc et.al., 1974 and Nguyen, 1974) became available. The advantages of equilibrium methods are that they are not heuristic but based on the assumption of optimized user behaviour and a convergent solution is also guaranteed. One major disadvantage, however, is that these methods cannot account for variation of user perceived travel time or costs (Van Vliet, 1976).

4.1.2.5 Dynamic Assignment

Yagar (1971) was one of the earliest to study the feasibility and application of dynamic assignment techniques. In the method which he proposed, a number of different trip matrices were assigned successively to the road network. These trip matrices represent the average inter-zonal flows of consecutive time slices. Trips were assigned as usual to minimum time paths, however, congested links were identified and the formation and dissipation of queues were accounted for both in the travel time calculation and assignment processes. By treating assignment as a quasi-dynamic process, it is then possible to account for the effect of road bottlenecks on the distribution of traffic. He used the method for assigning trips to a freeway corridor in California, U.S.A.

Another reported development in the area of dynamic assignment is CONTRAM, a traffic assignment and queueing model for urban road networks developed by the TRRL in Britian (Leonard et.al., 1978). The model assumes time-varying flow demands and the traffic movement through the network is represented by grouping vehicles into packets. Each packet is assigned to its minimum journey time route, taking into account delays and queues at junctions.

SATURN, a simulation and assignment model developed in the University of Leeds in Britian (Van Vliet, 1982), is similar to CONTRAM in that it is also quasi-dynamic. It however adopts more detailed simulation techniques of the movement of traffic through signalized intersections.

4.1.3 Transit Assignment

4.1.3.1 Generalised Cost of Travel

As in highway assignment, the earliest research on transit assignment (Von Cube, 1958) was related to the various factors influencing route choice. The study consisted of a survey conducted in Toronto, Canada on the usage of different alternative transit modes. Results from the survey showed that of the various factors that might influence route choice, travel time is much more significant than travel distance, whereas other factors related to the characteristics of the transit system, such as number of transfers, vehicle headways, dependability, comfort and traffic conditions along the route are also significant. Another finding from the study was that route choice behaviour during peak periods may be quite different from off-peak periods.

Since then, the fact that travel time is not the only factor influencing route choice has been well recognised. This has led to the use of the concept of generalised cost which assumes that the influence of the various factors can be quantified and summed to a single value which is expressed either in time or money units. McIntosh and Quarmby (1970) were among the first to conduct detailed calibration studies to obtain coefficients for the generalised cost function. Their results were widely adapted for transport planning studies conducted in Britian. Following their work, there has been much continuing research particularly on the valuation of travelling time. (DeSerpa, 1971; Goodwin, 1974; Dalvi and Daly, 1976). Another related area of research is the use of psychological techniques to obtain attitudinal values of travel time (Dobson, 1975; Horowitz, 1981).

4.1.3.2 Path-Finding Algorithms

Another important element required in transit assignment is the development of path-finding algorithms. As the determinant of transit route choice is often generalised cost rather than travel time, it is required to develop algorithms for finding paths with minimum generalised cost rather than travel time. As discussed in more detail in Chapter 3, these algorithms are much more complicated than minimum time algorithms. Nevertheless, such algorithms have been constructed. Examples are an algorithm developed by the U.S. Department of Housing and Urban Development (Dial, 1965) and another algorithm called Transitnet developed by Freeman, Fox, Wilbur Smith and Associates (Hewing and Hoffman, 1969.) which was first used for the London Transport Study. These algorithms however are based on transit networks which are assumed to be time-independent and not schedule-based. As will be discussed in the next section, more advanced assignment models can be developed from minimum path algorithms which are based on time-dependent and schedule-based transit networks.

4.1.3.3 Assignment Methods

Compared to highway assignment, assignment techniques for transit networks are more difficult to develop. There are at least two additional problems related to transit assignment procedures. Firstly, traditional transit networks are not scheduled-based, hence waiting times are assumed to be a function of the average route headway. In an all-or-nothing assignment, when there is a choice between alternative routes, all trips may then be assigned to the route with the highest frequency. However, in actual behaviour, the route used will be probabilistic, depending on which vehicle arrives first at the station. (The word "station" used within the context of this thesis refer to either a tram stop, bus stop or railway station.)

Secondly, the procedure for applying "capacity restraint" to congested networks is also quite different compared to those applicable to highway networks. When a transit route is overloaded, it is necessary to take into account the order in which passengers are loaded and unloaded at different stops along the route. Transit networks generally do not have speed-flow relationships as characterised in highway networks. When demand exceeds supply, queues are formed at the stations, and the waiting time at stations, rather than the in-vehicle time of a traveller increases.

The TRANSEPT bus model developed by the Local Government Operational Research Unit in Britian (Last and Leak, 1976) uses assignment techniques which try to account for the above problems. A simulation technique is used to calculate the probabilities of overloading along different sections of a transit route and the results are then used to determine waiting times. Dial's method of apportioning trips to alternative paths is used while waiting times and route demands are estimated iteratively until they stabilise.

While TRANSEPT is an isolated example of research into the development of more sophisticated transit assignment models, very few similar efforts have been reported. Most of the more widely used transit assignment packages, such as the UTPS system of the Urban Mass Transit Administration of the U.S. Department of Transportation do not have similar capabilities.

As reported by Lai and Van Vliet (1979), after a survey of the use of assignment techniques by local authorities in the U.K., it was found that only 25 per cent of public transport operators use the results of transit assignment for planning purposes. This could be attributed to the following reasons:

(a) unsatisfactory state of development of transit assignment models;

(b) high costs of using the models, both in terms of computer capacity and processing time; and

(c) large amount of output produced which required manual inspection and evaluation.

4.1.3.4 The Need for Further Research on Transit Assignment Methods

Research on transit assignment has been slow with few details reported in the literature. This can be attributed to the fact that in the past, the investment by most countries in highway facilities was very much larger than the investment in public transport facilities. The result was that transit assignment techniques were less well developed. Very often transit assignment models were adapted from highway assignment models with minor alterations. The following extract from Lamb (1970) could perhaps reflect the level of development of transit assignment techniques during this time:

> Very little importance was attached to public transport in the earliest transportation studies and where public transport networks were included they were described in terms which made it possible to use minimum path tree build (sic) algorithm which had already been used for the road networks. It will be appreciated that this approach cannot give wholly satisfactory results because of the fundamental difference in the nature of road and public transport journeys.

The previous section has shown that in many aspects transit assignment is more complex than highway assignment. Inadequate research has lead to the poor level of development of transit assignment models. However, it needs to be stressed that a good assignment model is a useful tool for the testing and evaluation of transit systems. The optimization of a highway network requires consideration of road alignment, road widths and junction layouts. These network elements are generally time-independent, thus it is not possible to vary the supply according to the change in demand during different periods of the day or during the duration of the design life of the road network. In contrast, many transit network elements such as routes and headways are time-dependent. In fact, it is essential in the optimization process to adjust the time-dependent capacity of a transit network in accordance with the predicted demand so as to maximize the operating efficiency of the transit system.

There are many problems related to transit assignment which warrant further research. Firstly, more research is required on modelling the behaviour of transit travellers during the route choice process. When making route choice decisions, the traveller often has to evaluate the various attributes of the route such as travel time, fares, reliability of services and comfort. As argued by Gray (1978), it is an over simplification of the choice process to assume that the effects of the various attributes can be summed in a way as described by a generalised cost function. Other more advanced models such as non-compensatory models have been suggested for modelling the human choice behaviour, (Young et.al., 1982) and applied to mode choice problems (Young and Richardson, 1981). The feasibility of applying these models to transit route choice should be investigated.

Secondly, more research is warranted on the development of path-finding algorithms. Although algorithms had been developed for finding paths with minimum generalised cost, as discussed in Chapter 3, these algorithms all perform similar functions and often require large computer storage. As the path-finding algorithm is at the heart of the assignment model, more advanced assignment models can only be obtained from more advanced path-finding algorithms.

Thirdly, more research is required on assignment procedures. As discussed in the previous section, the treatment of waiting times and capacity restraint are problems often encountered, and not satisfactorily solved, in most assignment models. There is also little research on the convergence of iterative assignment procedures for congested transit networks. Research is needed on the development of assignment procedures which can take into account the time varying demand and supply of a transit system.

4.2 Scope Of Present Research

As described in the previous section, there are several areas where research on transit assignment is needed. It is not possible to undertake research in all these areas in one project and the scope of the present chapter is described below.

This chapter describes some new techniques for transit assignment. Its development originates partly from previous research on a time-dependent network sub-model as described in Chapters 2 and 3 of this thesis. Some particular features of the new techniques are first discussed and the chapter then proceeds to describe two assignment sub-models which are based on the new techniques developed. Worked examples are given to illustrate the methods used in the two sub-models.

4.3 New Assignment Techniques

4.3.1 A Schedule-Based Transit Model

The assignment sub-models developed form part of an integrated transit model whose structure is shown in Figure 4.2. The model is described as schedule-based because a computerized database of transit timetables is used as the main input to the model.

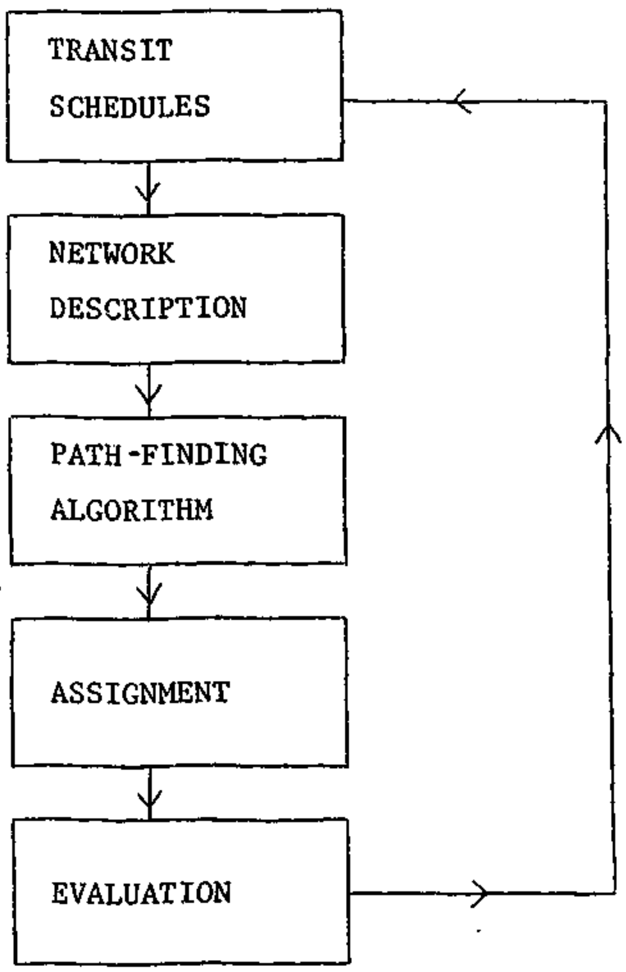


Figure 4.2 A schedule-based transit model

There are several advantages in adopting a schedule-based model. Firstly, the network generated from the model is time-dependent, hence, a minimum path tree built from the network not only shows the links used but also the clock times of travelling through the links. This important feature has enabled the development of dynamic assignment techniques which will be discussed in more detail in the next section.

Secondly, in conventional headway-based transit models, the waiting times of a trip cannot be realistically estimated when service headways are large or irregular, or when route schedules are co-ordinated at interchanges. By adopting a schedule-based model, such problems are avoided. This is because in such a model, waiting times are calculated from service schedules. In conventional transit models, when there is a choice of route running on parallel paths, all trips may be assigned to the route with the smallest headway. In a schedule-based model, those routes with larger headways may also have a chance of being selected as waiting times are determined from route schedules.

Finally, using the model, it is possible to evaluate in more detail alternative transit networks, such as the effects of irregular route schedules or alternative ways of route scheduling.

4.3.2 Dynamic Assignment Procedure

Conventional assignment procedures are performed for specific periods, the most common being a.m. peak, off peak and p.m. peak. Within each period, a number of simplifying assumptions are made:

(a) the traveller's route choice is time-independent;

(b) travel demand and supply remain constant; and

(c) a steady state condition is reached so that flows from different origin-destination pairs using the same link can be added.

A dynamic assignment procedure eliminates the need for the above simplifying assumptions. The result is that it provides more advanced and detailed modelling. The procedure originates from the development of a time-dependent network as described in Chapter 2. As the network is schedule-based, the data structure of a network link is designed such that each record for a link contains a number of fields, with each field corresponding to each vehicle that passes through the link at a given time as specified by route schedules. When a time-dependent minimum path is generated, the flow along the path is allocated to the appropriate links and appropriate field within each link record.

The above assignment procedure is analogous to assignment to vehicles, and hence can be described as a vehicle-based rather than link-based assignment procedure.

4.3.3 Multipath Assignment Procedure

A stochastic multipath assignment procedure is used. The procedure is related to a minimum path algorithm developed and described in Chapter 3. The algorithm finds an optimal path through the network for a specified origin-destination pair by minimizing a weighted time function consisting of four components:

(a) in vehicle time;

(b) waiting time;

(c) walking time; and

(d) time penalty for a route change.

By assuming that the weights attached to each component are not constant but vary for different transit user populations according to some distributions, alternative optimal paths can be generated. This procedure is analogous to the method proposed by Burrell (1968) for highway networks. The difference is that while Burrell's method assumes that link times are perceived differently by different groups of motorists, this method assumes that the coefficients of the weighted time function are perceived differently by different groups of transit users.

Compared to Burrell's method, one advantage of the present method is that the random components are attached to origin-destination pairs and not to individual links. This means that once a set of weights is determined for an O-D pair, all link weighted times are determined from this set of weights.

The remaining question then is what types of distribution functions should be used for the various coefficients of the weighted time function. Although it is quite natural to assume that different transit users perceive weighted travel time differently, very little previous research had been addressed to this problem. Bruzelius (1979) suggested that a normal distribution would be suitable. In the present model a similar distribution to that shown in Figure 4.3 is used. The means and variances assumed for these distributions are hypothetical. However, if required, actual values based on results of attitudinal surveys can be obtained (e.g. Dalvi and Daly, 1976). Random selection of values from a normal distribution can be usually be obtained by sub-routines in computer packages. Alternatively, values of random normal numbers can be read off from statistical tables (Rand Corporation, 1955).

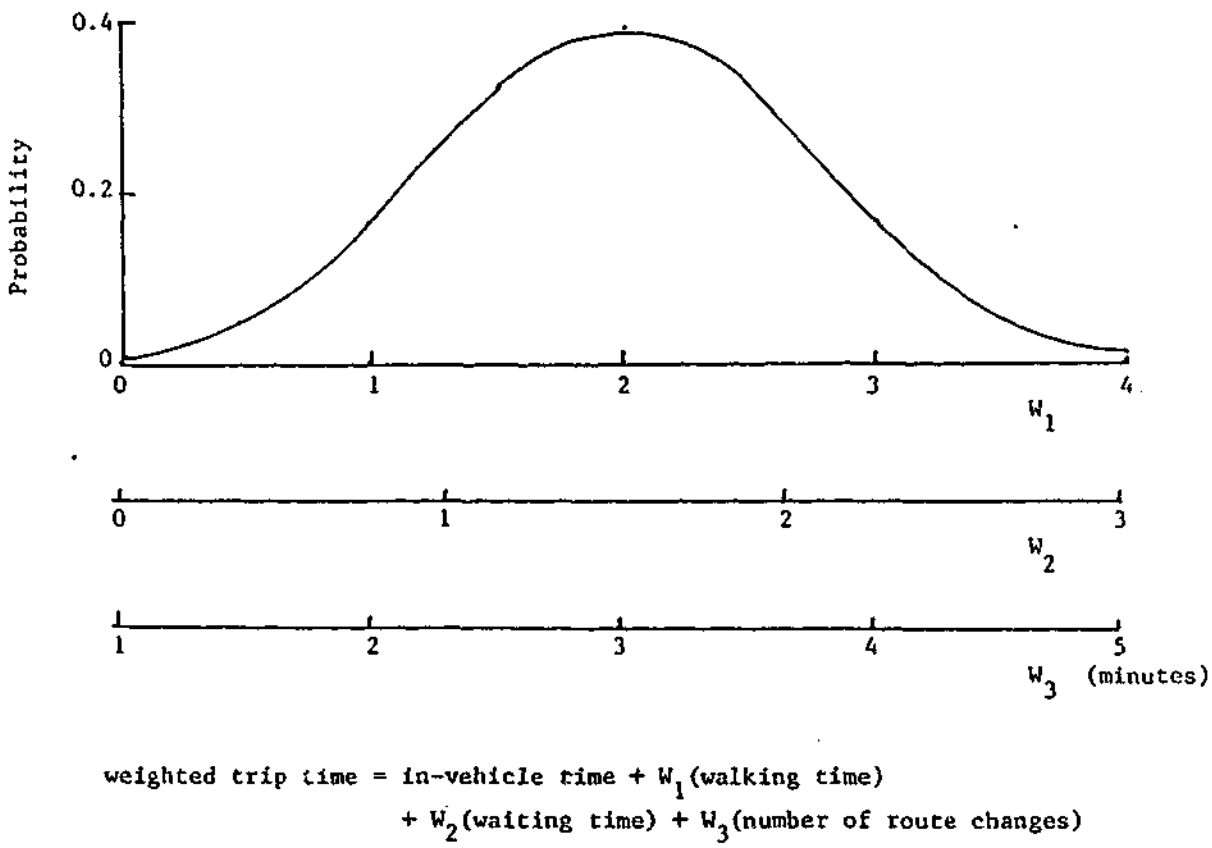


Figure 4.3 Distribution of weights in the weighted trip time function

4.4 Multi-interval Assignment Sub-model

Using the techniques described in the previous section, it should be possible to develop a fully dynamic transit assignment program. However, compared to conventional assignment programs, such a program would require more detailed input on trip flow matrices and much longer computer time. For practical considerations, a semi-dynamic assignment program has also been developed. The advantage of this program is that it requires less detailed input and computation effort as compared to a fully dynamic one and yet is more accurate and detailed compared to conventional assignment programs. In fact, the computation effort and accuracy of modelling can be adjusted in the program by varying the number of intervals used.

This latter sub-model is described as multi-interval because the assignment procedure is repeated for a number of consecutive time-intervals. A hypothetical example is used to illustrate the methods used. The example assumes a transit network shown in Figure 4.4.

There are 6 consecutive time intervals used in the assignment:

(a) 7:00 am to 7:19 am
(b) 7:20 am to 7:39 am
(c) 7:40 am to 7:59 am
(d) 8:00 am to 8:19 am
(e) 8:20 am to 8:39 am
(e) 8:40 am to 8:59 am

As input to the program, it is necessary to have zonal trip matrices corresponding to each time interval. Such matrices can be obtained either by survey or from other stages of the transport modelling process. In the latter case, it is

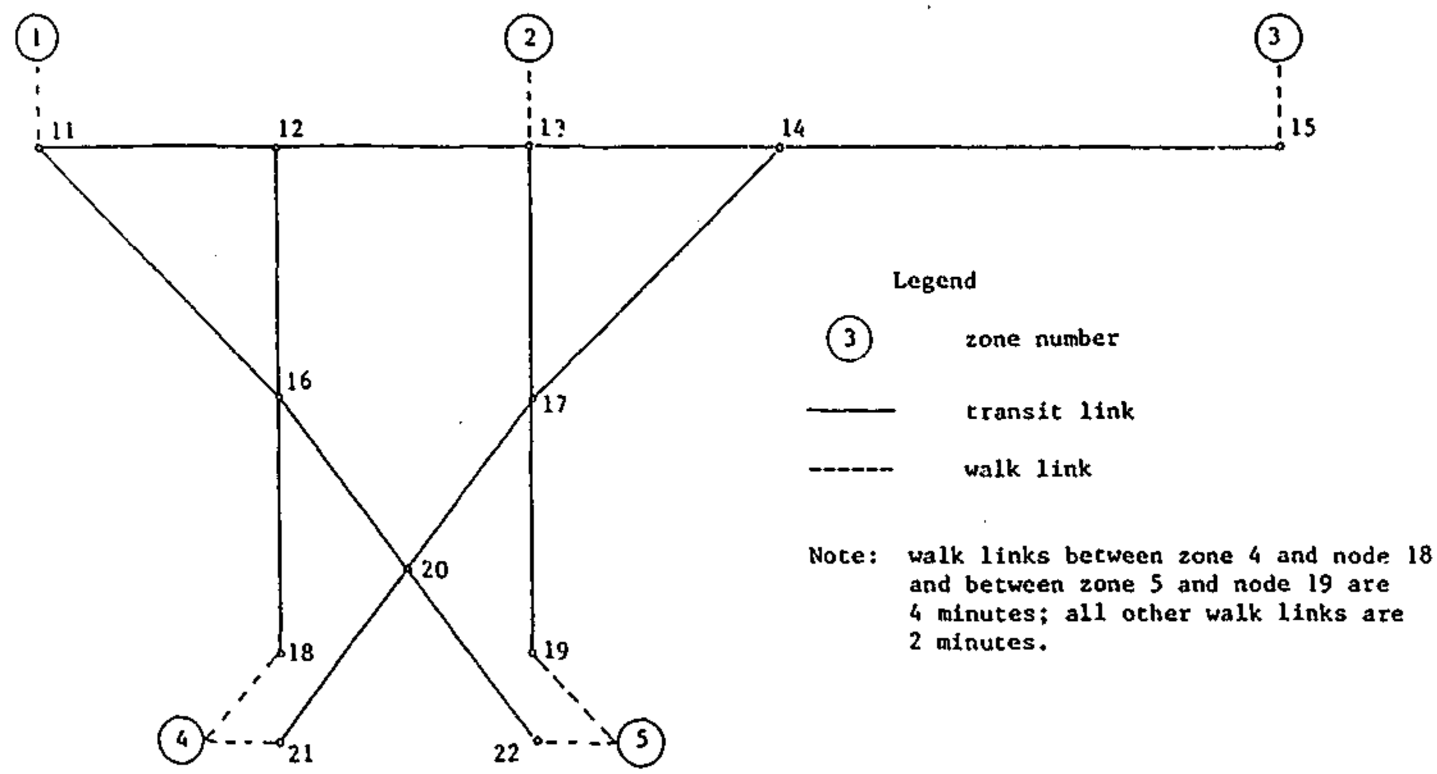


Figure 4.4 Hypothetical zonal transit network

necessary to modify the trip generation, distribution and modal split sub-models so that they are multi-interval as well. Multi-interval zonal trip matrices used in the example are hypothetical and shown in Table 4.1.

The assignment procedure used is briefly represented by the flow chart shown in Figure 4.5. For each origin zone, a random starting time within the chosen time interval is generated and the time-dependent optimal path to various destinations are found from the minimum path algorithm. The optimal paths from one origin to all destinations are found based on a weighted time function with weighting coefficients which are assumed to be not constants but normally distributed among the population of transit users (see Figure 4.3). Thus different weighting coefficients are used for different O-D pairs in the generation of optimal paths.

For more accurate results, the above procedure is iterated 4 times. Thus for each origin zone, 4 random starting times within each time interval are selected. Since there are a total of 6 time intervals and 4 destinations for each of the 5 origins, a total of (4 x 6 x 4 x 5 = 480) paths have been generated during the assignment procedure. An example of some of the paths generated is shown in Table 4.2.

The optimal paths generated are time-dependent, i.e. the departure time at the beginning of every link along a path is known. The data structure of the network links are modified such that each link record has a number of fields, with each field corresponding to one time interval used in the program.

Table 4.1 Multi-interval zone to zone trip matrices

For trips starting within period 7:00 am to 7:19 am

	1	2	3	4	5
1	-	60	80	80	60
2	60	-	60	60	60
3	80	80	-	80	80
4	80	60	80	-	60
5	60	60	80	60	-

For trips starting within period 7:20 am to 7:39 am

	1	2	3	4	5
1	-	80	100	100	80
2	80	-	80	80	80
3	100	100	-	100	100
4	100	80	100	-	80
5	80	80	100	80	-

For trips starting within period 7:40 am to 7:59 am

	1	2	3	4	5
1	-	100	120	120	100
2	100	-	100	100	100
3	120	120	-	120	120
4	120	100	120	-	100
5	100	100	120	100	-

For trips starting within period 8:00 am to 8:19 am

	1	2	3	4	5
1	-	120	100	100	120
2	120	-	120	120	120
3	100	100	-	100	100
4	100	120	100	-	120
5	120	120	100	120	-

For trips starting within period 8:20 am to 8:39 am

	1	2	3	4	5
1	-	80	100	100	80
2	80	-	80	80	80
3	100	100	-	100	100
4	100	80	100	-	80
5	80	80	100	80	-

For trips starting within period 8:40 am to 8:59 am

	1	2	3	4	5
1	-	60	80	80	60
2	60	-	60	60	60
3	80	80	-	80	80
4	80	60	80	-	60
5	60	60	80	60	-

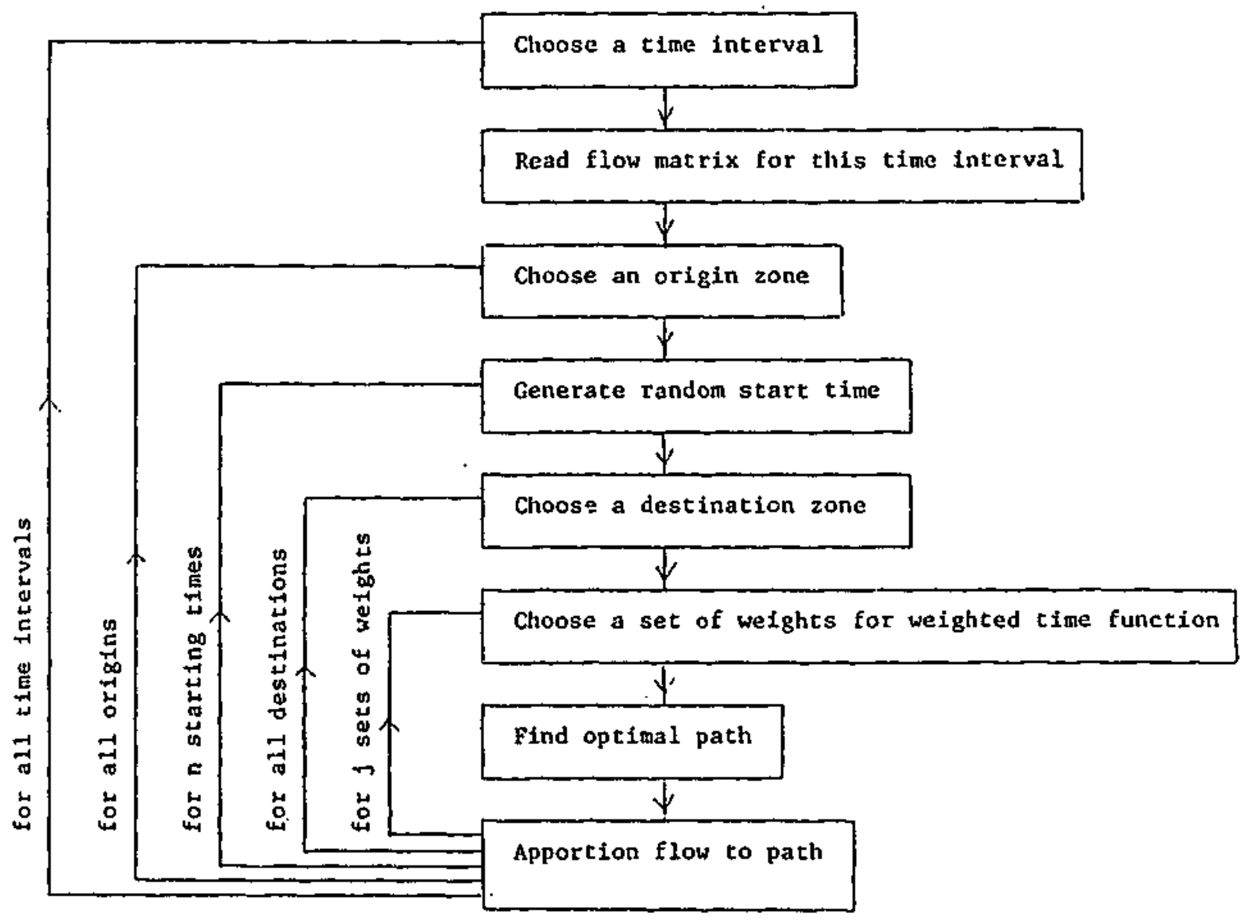


Figure 4.5 A multi-interval assignment sub-model

Table 4.2 Some paths generated in multi-interval assignment

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:15	A1	9
12	7:25	7:25	A1	17
13	7:35	7:35	WALK	19
2	7:37			

WEIGHTED TRIP TIME IS

20 + 1.5 X 4 + 1.9 X 0 + 4 X 0
= 26.1 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:15	A1	9
12	7:25	7:25	A1	17
13	7:35	7:35	A1	22
14	7:45	7:45	A1	25
15	8: 5	8: 5	WALK	27
3	8: 7			

WEIGHTED TRIP TIME IS

50 + 2.8 X 4 + 1.9 X 0 + 4 X 0
= 61.4 MIN

FLOW ASSIGNED TO LAST PATH = 80

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:20	B5	12
16	7:35	7:35	B5	32
20	7:45	7:45	B4	43
21	7:55	7:55	WALK	45
4	7:57			

WEIGHTED TRIP TIME IS

35 + 2.4 X 4 + 1.5 X 5 + 4 X 1

= 56.1 MIN

FLOW ASSIGNED TO LAST PATH = 80

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:20	B5	12
16	7:35	7:35	B5	32
20	7:45	7:45	B5	44
22	7:55	7:55	WALK	47
5	7:57			

WEIGHTED TRIP TIME IS

35 + 2.3 X 4 + 1.1 X 5 + 2 X 0
= 49.8 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
2	7:19	7:19	WALK	2
13	7:21	7:25	B2	20
12	7:35	7:35	B2	15
11	7:45	7:45	WALK	8
1	7:47			

WEIGHTED TRIP TIME IS

20 + 1.0 X 4 + 1.5 X 4 + 4 X 0
= 30.1 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
2	7:19	7:19	WALK	2
13	7:21	7:35	A1	22
14	7:45	7:45	A1	25
15	8: 5	8: 5	WALK	27
3	8: 7			

WEIGHTED TRIP TIME IS

30 + 3.4 X 4 + 1.8 X 14 + 3 X 0
= 69.0 MIN

FLOW ASSIGNED TO LAST PATH = 60

NOTE:

WEIGHTED TRIP TIME = IN-VEHICLE TIME + W_1 (WALKING TIME) + W_2 (WAITING TIME) + W_3 (NUMBER OF ROUTE CHANGES)

When assigning flows onto links, the flow will be allocated to the appropriate field in the link record depending on the time at which the flow passes through the link. Thus, as a path which is generated at one interval may proceed onto other time intervals, the assignment process is multi-interval.

Results from the assignment can be output either in a link-based or vehicle-based format. If a link-based format is used, then the output will consist of a table showing link flows at different time intervals. The assignment results obtained are shown in Table 4.3.

When using the assignment results, the effect of time-lag must be considered. This is because trips which start before but end within the specified assignment period have been excluded from the assignment. In order to account for this time-lag effect, the results obtained for some of the earlier intervals should be discarded. For example, if trip times are not expected to exceed 40 minutes, then the assignment results for the first two intervals are discarded as being under estimated. Alternatively, it is possible to account for this time lag effect by including earlier time intervals in the assignment.

If a vehicle-based output option is chosen, then the assignment results will be output in the form of vehicle flows printed in a timetable format. In this case when there is more than one vehicle of the same route passing through the same link in one time interval, it is assumed that the flows are equally divided among the vehicles, thus yielding an average loading in each vehicle. The advantage of using this format is

Table 4.3 Results of multi-interval assignment

A-NODE	B-NODE	ROUTE NO.	FLOW IN INTERVAL NOS. 1	2	3	4	5	6
1	11	WALK	280	360	440	440	360	280
2	13	WALK	240	320	400	480	320	240
3	15	WALK	320	400	480	400	400	320
4	18	WALK	0	50	72	110	0	40
4	21	WALK	280	225	410	372	360	240
5	19	WALK	123	145	130	190	200	75
5	22	WALK	136	155	330	255	155	185
11	1	WALK	0	0	260	215	460	535
11	12	A1	90	65	105	300	280	20
11	12	A3	0	45	85	25	0	20
11	12	A2	15	15	50	30	10	30
11	16	B5	35	195	25	190	320	0
12	11	B3	0	0	0	110	50	0
12	11	B1	0	0	60	172	235	75
12	11	B2	0	60	65	57	60	50
12	13	A2	15	15	50	42	10	30
12	13	A1	0	90	170	220	270	170
12	16	A3	0	45	0	190	25	40
13	2	WALK	0	65	155	315	495	565
13	12	B2	0	60	85	57	60	50
13	12	B1	0	60	145	112	305	250
13	14	A1	0	200	173	300	400	210
13	17	A2	0	120	125	280	505	75
14	13	B1	0	60	480	375	165	410
14	15	A1	0	60	100	569	380	605
14	17	B4	0	20	110	50	140	300
15	3	WALK	0	0	60	100	569	380
15	14	B1	80	540	460	220	700	240
16	11	A5	0	140	90	45	275	140
16	12	B3	0	0	0	122	110	0
16	18	A3	0	20	25	190	50	0
16	20	B5	0	120	110	25	305	180
17	13	B2	0	123	290	207	240	285
17	14	A4	0	156	0	140	405	75
17	19	A2	0	60	125	180	295	75
17	20	B4	0	60	20	110	495	55
18	4	WALK	0	20	25	0	190	50
18	16	B3	0	0	122	110	0	0
19	5	WALK	0	0	60	125	180	295
19	17	B2	0	123	145	130	290	135
20	16	A5	0	140	135	135	140	235
20	17	A4	0	156	362	170	235	350
20	21	B4	0	60	160	270	325	410
20	22	B5	0	75	85	275	225	170
21	4	WALK	0	0	220	135	135	735
21	20	A4	280	180	277	550	360	140
22	5	WALK	0	0	160	150	125	395
22	20	A5	136	135	140	465	125	175

that it facilitates the evaluation of transit schedules. The results obtained are shown in Table 4.4. Again, the effects of time lag must be considered in the interpretation of the results. Furthermore, since the assignment period is from 7 am to 9 am, all transit vehicles scheduled after 9 am have not been assigned any flow and this explains why the lower portions of the timetabled flows are zero.

4.5 Dynamic Assignment Sub-Model

While the multi-interval assignment program described in the previous section is applicable to strategic planning of transit networks, a dynamic assignment program which has also been developed is suited for cases which require more accurate modelling, such as the monitoring of existing transit networks or the evaluation of the effects of minor changes to existing transit networks and schedules.

A hypothetical example with a transit network shown in Figure 4.6 will be used to illustrate the methods used.

The input to the program consists of multi-interval station to station trip matrices. The reason for using station to station rather than zone to zone trip matrices is that the former can be obtained more accurately by conducting surveys at stations. There is no restriction on the size of each interval corresponding to the trip matrices, although smaller time intervals would result in more accurate assignment results. A suitable size would be between 10 to 20 minutes depending on the variability of passenger flow rates. In the present example, due to the lack of data, the same multi-

Table 4.4 Results of multi-interval assignment in timetable format

ROUTE NUMBER A1

NODE	11	12	13	14	15
	45	0	100	60	
	45	90	100	100	
	65	85	173	284	
	105	85	300	284	
	150	220	200	380	
	150	135	200	302	
	140	135	105	302	
	140	85	105	0	
	20	85	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B1

NODE	15	14	13	12	11
	80	60	60	60	
	270	240	145	86	
	270	240	112	86	
	460	375	152	235	
	110	82	152	37	
	110	82	125	37	
	350	205	125	0	
	350	205	0	0	
	240	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A2

NODE	11	12	13	17	19
	15	15	120	60	
	15	15	125	125	
	50	50	280	180	
	30	42	505	295	
	10	10	75	75	
	30	30	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B2

NODE	19	17	13	12	11
	0	0	60	60	
	123	123	85	65	
	145	290	57	57	
	130	207	60	60	
	290	240	50	50	
	135	285	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A3

NODE	11	12	16	18
	0	0	20	
	45	45	25	
	85	190	190	
	25	25	50	
	20	40	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B3

NODE	18	16	12	11
	0	0	0	
	0	0	0	
	122	122	110	
	110	110	50	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A4

NODE	21	20	17	14
	140	0	78	
	140	156	78	
	180	181	0	
	277	181	140	
	275	170	202	
	275	235	202	
	360	175	75	
	140	175	0	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B4

NODE	14	17	20	21
	0	30	60	
	10	30	160	
	10	20	135	
	110	110	135	
	50	247	325	
	70	247	410	
	70	55	0	
	300	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A5

NODE	22	20	16	11
	68	0	70	
	68	140	70	
	135	67	90	
	140	67	45	
	232	135	137	
	232	140	137	
	125	117	140	
	175	117	0	
	0	0	0	

ROUTE NUMBER B5

NODE	11	16	20	22
	35	60	75	
	97	60	85	
	97	110	137	
	25	25	137	
	190	152	225	
	160	152	170	
	160	180	0	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

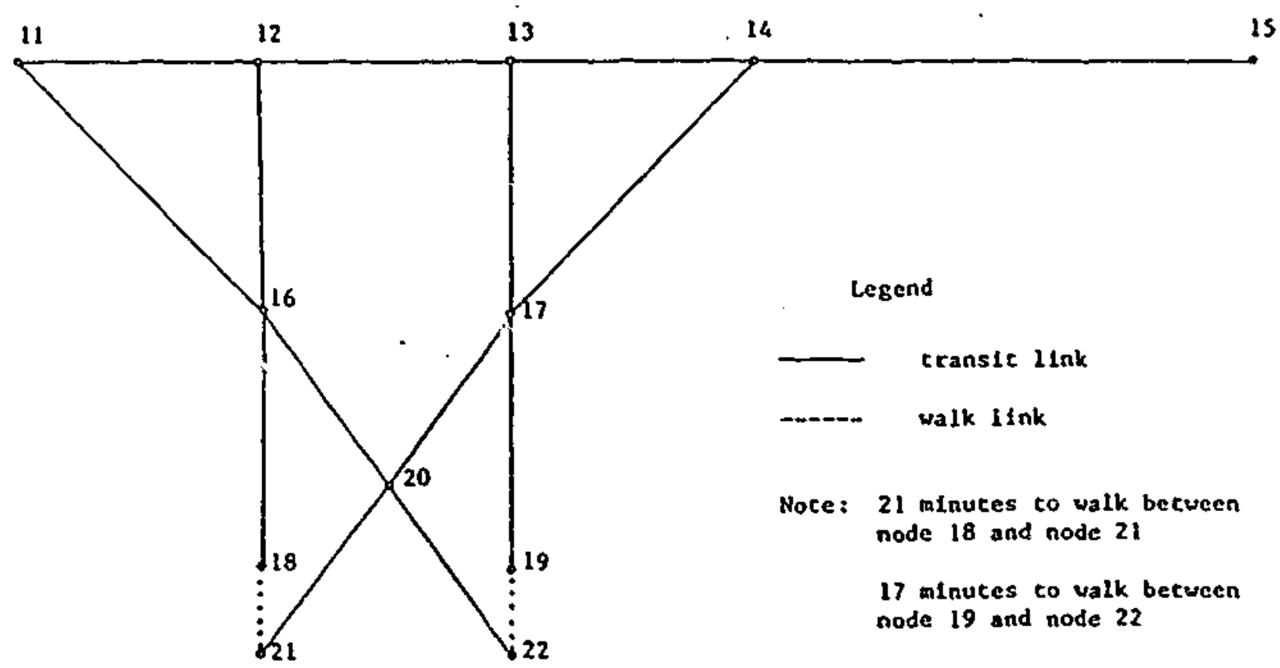


Figure 4.6 Hypothetical transit network

interval matrices used in the previous model will be used and it is assumed that the 5 zones represented in the matrix are replaced by 5 stations, as represented in the network by the following nodes: 11, 13, 15, 21 and 22. Thus although there are 12 nodes in the network, the assignment is only made for the 5 nodes specified. In practice, more detailed trip matrices with larger number of nodes can be used.

The iterative assignment procedure is applied to each origin-destination station pair. The procedure is illustrated in Figure 4.7 and is briefly described in the following paragraphs.

An assignment time period from 7 am to 9 am is specified. For every iteration applied to an O-D station pair, the procedure starts at the beginning of the assignment period. As time progresses, an assignment is made whenever three conditions are satisfied:

(a) a vehicle departs from the origin station;

(b) the vehicle departs on an optimal path for the specified O-D station pair; and

(c) at least 5 passengers have accumulated at the station during the time interval between the last assignment and the present assignment.

The last condition is used to control the accuracy of the assignment results as well as the computation time and the minimum number is specified by the program user. If the minimum number of passengers is set to one, then all relevant vehicles are accounted for.

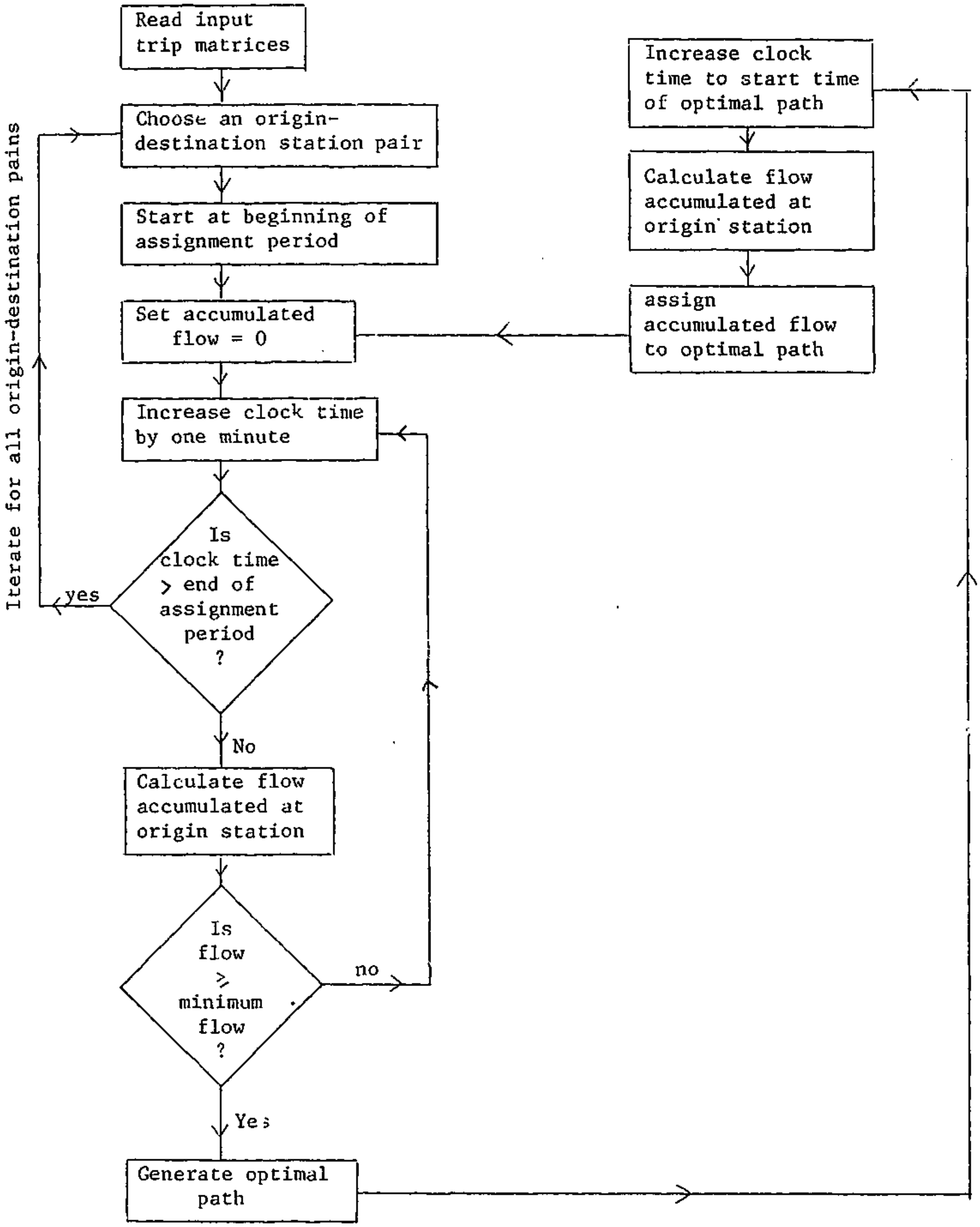


Figure 4.7 A dynamic assignment sub-model

The method is illustrated by means of an example in Figure 4.8. The passenger flow rates at the origin station for the first 30 minutes of the assignment period are shown. The passenger flow to be assigned to each path is determined from the given flow rate and the time interval between the starting time of successive optimal paths. Five persons are set as the minimum number of persons for making an assignment. The first optimal path generated marks the beginning point of the assignment. The 2nd path is actually not generated because at this point in time less than 5 persons have accumulated at the origin station for the intended trip. The 3rd path is therefore the next path to be generated and 9 persons are assigned to this path. The 4th path is the next to be generated and 8 persons are assigned to this path. This assignment procedure continues until the end of the assignment period is reached.

The weighting coefficients of the weighting time function are randomly generated for each path from distributions as assumed in Figure 4.3. The assignment procedure is hence stochastic. An example of some of the paths generated during the assignment process is shown in Table 4.5. For large transit networks, the number of paths generated in the assignment procedure is normally very large so that it would not be necessary to iterate the procedure in order to obtain consistent results.

The assignment is vehicle-based. Each link record is designed to contain a number of fields, with each field

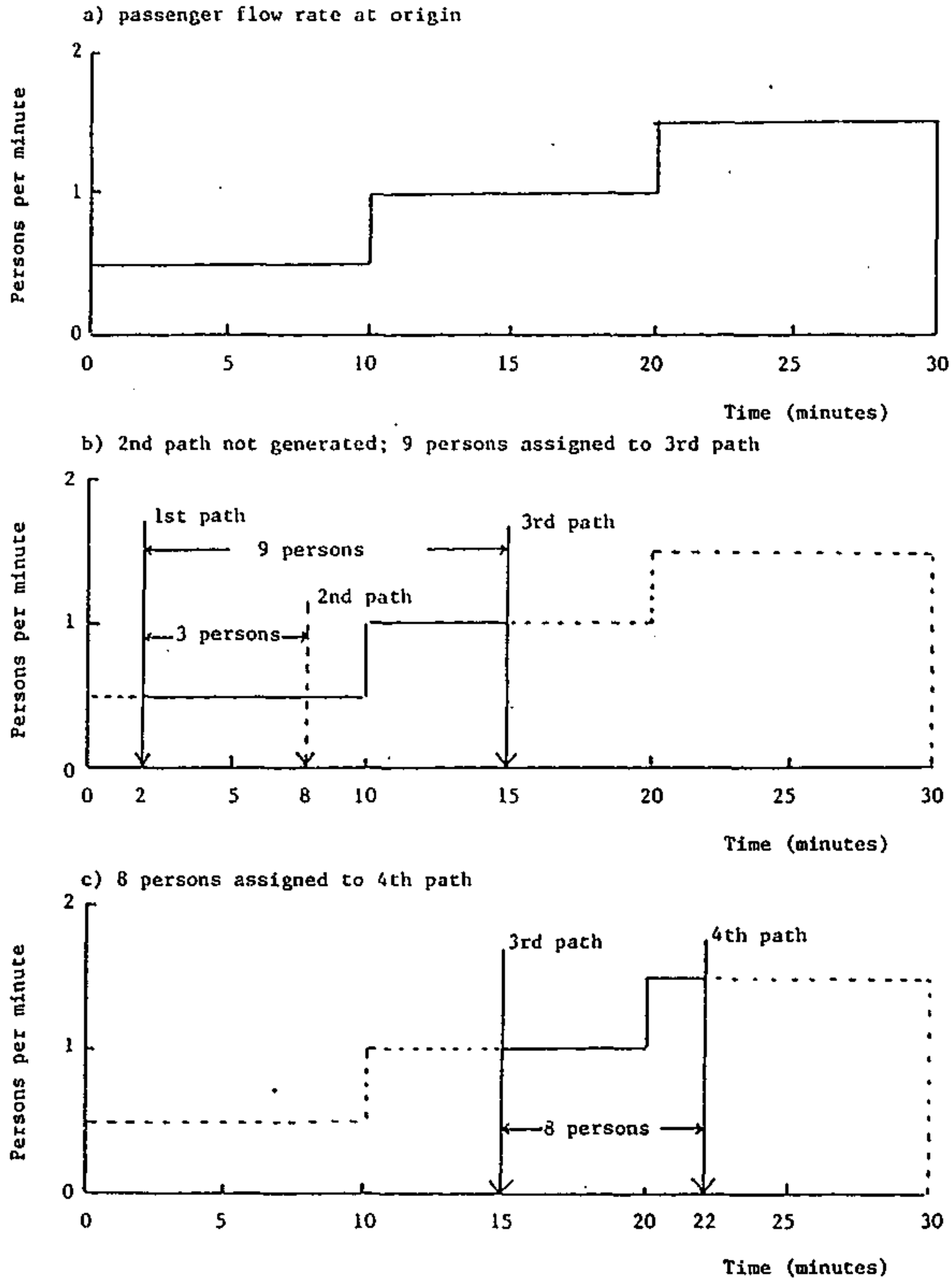


Figure 4.8 Dynamic assignment to optimal paths

Table 4.5 Some paths generated in dynamic assignment

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 2	7: 2	A3	2
12	7:12	7:12	A3	10
16	7:22	7:22	A3	21
18	7:32	7:32	WALK	28
21	7:34			

WEIGHTED TRIP TIME IS

30 + 2.6 X 2 + .9 X 0 + 4 X 0
= 35.1 MINUTES

FLOW ASSIGNED TO LAST PATH = 8

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 4	7: 5	B5	4
16	7:20	7:22	A3	21
18	7:32	7:32	WALK	28
21	7:34			

WEIGHTED TRIP TIME IS

25 + 2.2 X 2 + 1.2 X 3 + 3 X 1
= 36.0 MINUTES

FLOW ASSIGNED TO LAST PATH = 12

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 7	7:20	B5	4
16	7:35	7:35	B5	22
20	7:45	7:45	B4	33
21	7:55			

WEIGHTED TRIP TIME IS

35 + 1.7 X 0 + 1.4 X 13 + 2 X 1
= 55.2 MINUTES

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7:21	7:27	A3	2
12	7:37	7:37	A3	10
16	7:47	7:47	A3	21
18	7:57	7:57	WALK	28
21	7:59			

WEIGHTED TRIP TIME IS

30 + 2.0 X 2 + 2.4 X 6 + 2 X 0
= 48.3 MINUTES

FLOW ASSIGNED TO LAST PATH = 35

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7:28	7:35	B5	4
16	7:50	7:50	B5	22
20	8: 0	8: 0	B4	33
21	8:10			

WEIGHTED TRIP TIME IS

35 + 3.0 X 0 + .9 X 7 + 4 X 1
= 45.0 MINUTES

FLOW ASSIGNED TO LAST PATH = 40

NOTE:

WEIGHTED TRIP TIME

= IN-VEHICLE TIME + W_1 (WALKING TIME)

+ W_2 (WAITING TIME)

+ W_3 (NUMBER OF ROUTE CHANGES)

corresponding to each vehicle that passes through the link at a given time as specified by route schedules. The flows are allocated to the appropriate links and the appropriate field in each link record. The resultant vehicle-based link flows are shown in Table 4.6. However, it is more convenient to convert the results in a timetable format and this is shown in Table 4.7. As in the case of the multi-interval assignment sub-model, the results are also affected by time lag effects.

Compared to conventional transit assignment methods, the two sub-models described in this chapter would require more detailed data input and processing time. These disadvantages could mostly be overcome by improved methods of data collection such as electronic detection and recording of passenger movements through station gates and the availability of more powerful computers. However, for hypothetical schemes there may be a problem in obtaining forecasts of passenger demand at the level of detail required by the present sub-models.

If the two sub-models are to be applied to the Melbourne network consisting of rail, tram and government operated bus services, the required program capacity is estimated to be about 150 K bytes. Running the multi-interval assignment program with 25 zones, 6 time intervals and 4 iterations would require a CPU time of approximately 52 hours. Running the dynamic assignment program with 30 by 30 station to station flow matrices would require a CPU time of approximately 75 hours.

Compared with conventional transit assignment programs, the two sub-models developed would require more detailed data input and CPU time. The advantage of the sub-models is that for short-range planning of transit systems, the results of from the sub-models can be used for the detailed evaluation of changes in transit services.

Table 4.6 Results of dynamic assignment

A-NODE	B-NODE	ROUTE NO.	VEHICLE-BASED LINK LOADINGS												
11	12	A1	0	90	90	112	165	102	124	108	73	0	0	0	0
11	12	A3	8	35	107	60	59	0	0	0					
11	12	A2	15	55	32	30	20	31	0	0	0				
11	16	B5	27	105	80	70	143	105	115	56	0	0	0		
12	11	B3	0	24	90	62	0	0	0						
12	11	B1	60	85	115	175	98	104	0	0	0	0			
12	11	B2	80	60	64	20	6	0	0	0					
12	13	A2	15	55	32	35	20	31	0	0	0				
12	13	A1	0	90	90	112	165	104	124	162	73	0	0	0	0
12	16	A3	8	35	192	60	87	0	0						
13	12	B2	80	75	64	20	6	0	0	0					
13	12	B1	60	155	115	175	126	104	96	0	0	0	0		
13	14	A1	60	168	159	337	180	110	104	96	0	0	0	0	
13	17	A2	160	140	260	220	272	0	0	0					
14	13	B1	60	156	270	170	210	180	150	120	0	0	0	0	
14	15	A1	60	152	279	337	300	280	274	0	0	0	0		
14	17	B4	0	0	20	130	75	170	130	90	0	0	0		
15	14	B1	80	240	300	340	280	240	240	240	200	0	0	0	0
16	11	A5	28	108	89	45	159	115	140	0	0				
16	12	B3	0	24	97	116	48	0	0						
16	18	A3	20	35	182	75	0	0	0						
16	20	B5	15	105	80	70	153	90	115	0	0	0	0		
17	13	B2	14	102	347	172	190	233	0	0	0				
17	14	A4	30	165	0	120	170	170	122	0	0	0	0		
17	19	A2	80	125	180	195	225	0	0	0					
17	20	B4	6	95	20	145	155	201	130	0	0	0	0		
18	16	B3	0	28	97	121	57	0	0	0					
18	21	WALK		413											
19	17	B2	20	105	132	73	136	135	0	0	0				
19	22	WALK		970											
20	16	A5	28	104	89	45	159	110	131	63	0				
20	17	A4	42	165	215	219	250	150	220	137	0	0	0	0	
20	21	B4	57	199	129	164	318	257	0	0	0	0			
20	22	B5	72	104	109	209	194	152	0	0	0	0			
21	18	WALK		343											
21	20	A4	56	210	237	222	317	209	274	175	0	0	0	0	
22	19	WALK		631											

Table 4.7 Results of dynamic assignment in timetable format

ROUTE NUMBER A1

NODE	11	12	13	14	15
	0	0	60	60	
	90	90	168	152	
	90	90	159	279	
	112	112	337	337	
	165	165	180	300	
	102	104	110	280	
	124	124	104	274	
	108	162	96	0	
	73	73	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B1

NODE	15	14	13	12	11
	80	60	60	60	
	240	156	155	85	
	300	270	115	115	
	340	170	175	175	
	280	210	126	98	
	240	180	104	104	
	240	150	96	0	
	240	120	0	0	
	200	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A2

NODE	11	12	13	17	19
	15	15	160	80	
	55	55	140	125	
	32	32	260	180	
	30	35	220	195	
	20	20	272	225	
	31	31	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B2

NODE	19	17	13	12	11
	20	14	80	80	
	105	102	75	60	
	132	347	64	64	
	73	172	20	20	
	136	190	6	6	
	135	233	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A3

NODE	11	12	16	18
	8	8	20	
	35	35	35	
	107	192	182	
	60	60	75	
	59	87	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B3

NODE	18	16	12	11
	0	0	0	
	28	24	24	
	97	97	90	
	121	116	62	
	57	48	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A4

NODE	21	20	17	14
	56	42	30	
	210	165	165	
	237	215	0	
	222	219	120	
	317	250	170	
	209	150	170	
	274	220	122	
	175	137	0	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B4

NODE	14	17	20	21
	0	6	57	
	0	95	199	
	20	20	129	
	130	145	164	
	75	155	318	
	170	204	257	
	130	130	0	
	90	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A5

NODE	22	20	16	11
	32	28	28	
	149	104	108	
	170	89	89	
	180	45	45	
	250	159	159	
	255	110	115	
	195	131	140	
	127	63	0	
	0	0	0	

ROUTE NUMBER B5

NODE	11	16	20	22
	27	15	72	
	105	105	104	
	80	80	109	
	70	70	209	
	143	153	194	
	105	90	152	
	115	115	0	
	56	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

CHAPTER 5

MEASUREMENT OF TRANSIT ACCESSIBILITY

CHAPTER 5

MEASUREMENT OF TRANSIT ACCESSIBILITY

5.1 Introduction

In recent years, a large number of research papers have been published on accessibility (E.g. Weibull, 1976; Ben-Akiva and Lerman, 1978; Pirie 1979; and Koenig, 1980). The result is that a wide spectrum of accessibility measures has been suggested. Surprisingly, however, very little previous research has been directed to transit accessibility, especially related to the benefits that can be derived from a given transit system. One important application of this type of accessibility measure is in the evaluation of transit systems. The advantage is that it provides a sound theoretical basis for economic evaluation, resulting in a more equitable distribution of transport services to different groups of people. There are other potential applications as well: for example, it can be used to determine the accessibility provision of a transit system to various areas and hence fix the level of transit based land rates levied on properties located in these areas. It also affects facility and home location and hence can be applied to urban development modelling. It is a determinant of travel choice, and hence can be applied to transport choice modelling, such as in trip generation, distribution and modal split sub-models.

Transit accessibility is much more difficult to measure than auto accessibility. Firstly, it must be closely related to the performance of a given transit system. Since transit services are not available on call but run according to fixed schedules, the accessibility of a transit user will vary according to what time the user wishes to start from the origin or arrive at the destination. Transit accessibility is thus temporal and dependent on route schedules. The index used must therefore be sensitive to these factors. Secondly, a transit trip often involves walking, waiting and interchanging in addition to travelling on vehicles. A suitable algorithm which can calculate accurately these various components of a transit trip based on route schedules and for different starting or arrival times will be an essential part of a model developed for the measurement of transit accessibility.

At the current state of development, there is an obvious need for further research into the question of what is an appropriate transit accessibility measure and the development of computer models that can provide an efficient computation tool.

The present study has suggested a few improvements in the measurement of transit accessibility. These are demonstrated in the form a case study described in Section 5.3. The study cannot be considered as comprehensive as there are still many outstanding problems related to the measurement of transit accessibility. The need for further research on these problems are discussed in Section 5.4.

While several previous authors have provided reviews on accessibility (Ingram, 1971; Morris et.al. 1978; Pirie, 1979), they were however concerned mainly with auto accessibility while transit accessibility was generally not discussed in detail. For this reason, a review of transit accessibility is provided in the next section.

5.2 Review Of Transit Accessibility

5.2.1 Level of Service Measure

The level of service provided by public transport has often been wrongly termed accessibility. Apart from the ease of travel, accessibility should also include the spatial distribution of opportunities relative to the origin. Hence level of service alone cannot be regarded as accessibility.

The Baltimore Study (Wilbur Smith and Associates, 1964) used the following level of service index:

$$\text{transit service index of zone } i = \sum_r f_{ri}$$

where

f_{ri} = the frequency of transit service r available in zone i

In the London Traffic Study (Greater London Council, 1966), the following level of service indices were used but found to be not a significant factor in determining trip generation:

$$\text{Rail level of service for zone } i = \sum_r N_{ri} / a_i$$

$$\text{Bus level of service for zone } i = \sum_r M_{ri} / a_i$$

where

N_{ri} = the number of trains during the off-peak period stopping at station r in zone i

M_{ri} = the number of buses during the off-peak period stopping at station r in zone i

a_i = the area of zone i (square miles).

Leake and Huzayyin (1979) suggested an index of the following form:

$$A_i = (\Sigma_r f_{ri} l_{ri}) / a_i$$

where

A_i = the transit level of service of zone i

l_{ri} = the length of route r passing through zone i

and the other variables are the same as those used in previous equations used in this Section.

In all the above cases, level of service was used in the modelling of trip generation. The selection of the expression for level of service is either arbitrary or dependent on which expression could give a better fit to observed data when used as an independent variable in trip generation models. If the spatial distribution of destinations can be incorporated into these expressions to obtain an accessibility index, then it is likely that the accuracy of these trip generation models could be improved.

Dallal (1980) used the walking time from home to the nearest bus stop plus an average waiting time as a measure of transit level of service and produced level of service contours for a small district inside London. An analysis was then made to assess the correlation between the household level of service, car ownership and modal choice. Again, this shows that very crude accessibility measures have often been used in various types of transport models.

5.2.2 Cumulative-Opportunities Measure

Accessibility is often equated to the number of opportunities that can be reached from the origin within a specified travel time. For example, the accessibility of a zone for work trips can be defined as the number of work places that can be reached within a specified time from the zone centroid. Wachs and Kumagai (1973) used an index of this type to measure the auto accessibility index to employment in Los Angeles, United States. They used a similar method to measure the transit accessibility to health care centres. It is interesting to note that they actually use published transit timetables and manual procedures to calculate trip times, assuming that all transit trips start at noon of a typical weekday and the initial waiting time is zero. With the model developed in the presented study, the required information can be generated by computer. It should also be possible to include a wider range of starting times and provide schedule-based estimates of initial waiting times in the analysis.

This type of accessibility measure is simple in concept and hence easy to understand. The main disadvantage however is that the measure is sensitive to the cut off point on travel time assumed in the measure.

5.2.3 Gravity Type Measure

Hansen (1959) used the concept of accessibility in the modelling of land development and suggested the following gravity type measure for accessibility:

$$A_i = \sum_j D_j / f(c_{ij})$$

where

A_i = accessibility of site i

D_j = opportunities at site j

$f(c_{ij})$ = a function of the cost of travel between i and j.

The above expression is analogous to Newton's Law of Gravity Attraction and is hence termed a gravity measure. Dalvi and Martin (1976) used a similar expression to measure the auto accessibility of zones in London using data from the 1962 London Travel Survey. Their study showed that accessibility measurements could be influenced by zonal configuration used and the values of parameters used in the travel cost function.

Such a measure can also be used in the measurement of transit accessibility. Pike et.al. (1976) in the Telford Transportation Study have used the following expression:

$$A = -\frac{1}{b} \ln \frac{\sum_i P_i \sum_j (e^{-b c_{ij}}) E_j}{\sum_i P_i \sum_j E_j}$$

where

b = sensitivity coefficient,

P_i = the number of persons in a particular potential user group in zone i,

E_j = a measure of the attractiveness of zone j,

c_{ij} = the generalised cost of travel from zone i to zone j.

The above expression can be considered as a modification of the Hansen type index by a logarithmic transformation. The study showed that the index is sensitive to changes in trip cost but much less sensitive to changes in b. Since it is often difficult to estimate the value of b accurately, the index is considered quite suitable for the evaluation of alternative transit systems.

The paper however did not describe details on how the minimum cost paths between zones were determined and there was no mention of the time of day on which the accessibility measurement was based.

5.2.4 Time-Space Measure

Time-space measures of accessibility emphasize the temporal constraints of activities and travel facilities. For example, shops only open during certain times of the day and schedules restrict the time of travel of a transit trip. It also relaxes the restrictive assumption that trips must always start from the location in which accessibility is being considered, and includes other trips such as trips returning to the location under analysis or trip chaining behaviour.

Lenntorp (1976) developed a model for measuring time-space measures of accessibility. The model actually considers the opening times of various facilities; transit service schedules; the average time spent on various activities and typical activity schedules of various types of people. He recognised the huge demand on data and analysis effort of the model and considered the method to be only applicable to small towns and small planning projects. This suggests the need for the development of computer algorithms to assist in the analysis.

Burns (1979) has further developed the idea of time-space measures to represent the transportation, temporal and spatial components of accessibility. He developed diagrams of time-space prisms which represents the relationship between time, space and the ease of travel. The same concept can be applied to transit accessibility.

While time-space measures of accessibility appears to be an attractive and useful concept, especially when it is desired to account for trip chaining behaviour, it is rarely applied in practice. This may be due to the fact that without the development of computer models to assist in its measurement, new concepts are difficult to put into practical use.

5.2.5 Logit Model Type Measure

In economic terms, the accessibility of a place i could be inferred as the expected benefit that a person in i can derive from available transport facilities for travelling to various places. Several authors (Williams, 1977; Daly and Zachary, 1978; Ben-Akiva and Lerman, 1979) have shown that using this definition the accessibility of site i can then be given by the following expression:

$$A_i = \ln \sum_j \exp V_{ij}$$

where

V_{ij} = utility of the trip from i to j.

Although a transit accessibility index of this type could provide a sound theorectical basis for the evaluation of transit systems, it also has some disadvantages. Firstly, the choice set used in the model may have to include choices over various dimensions, such as choice over destinations, modes and time of travel. This would require a nested logit model and calibrations to obtain the various sensitivity coefficients used in the model. An index of this type is hence more difficult to obtain compared to other types, such as gravity type accessibility. Secondly, the index is derived from the logit model of transport demand and hence represents a demand accessibility measure. On the other hand, if the index is derived purely from the supply characteristics of the transit system, then a supply accessibility measure is obtained. As explained in more detail in Section 5.4.2, a demand type accessibility may not be suitable for evaluation.

5.3 A Case Study

5.3.1 Introduction

The review showed that while there are many different ways of defining transit accessibility, those which have been used in transport studies are usually simple measures. This can be attributed to the lack of research on the development of transit accessibility models.

The present study is concerned with the development of transit accessibility models. The techniques developed will be demonstrated in the form of a case study using a hypothetical transit network shown in Figure 5.1.

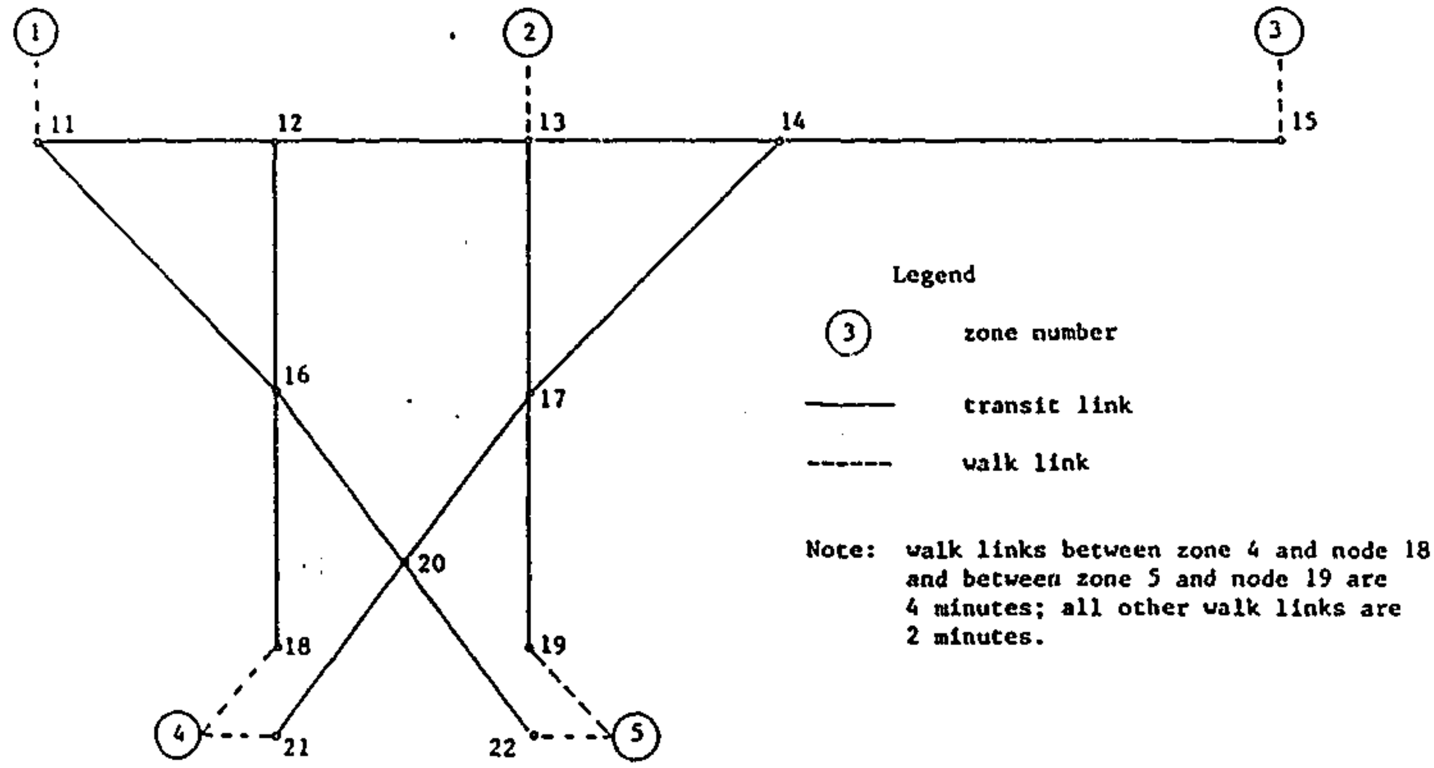


Figure 5.1 Zonal transit network

The following modified gravity type accessibility measure is used:

$$A_i = -\frac{1}{b} \ln \frac{\sum_j D_j e^{-b c_{ij}}}{\sum_j D_j},$$

where

A_i = accessibility of zone i for travelling to work,

D_j = number of jobs at zone j,

c_{ij} = travel cost from zone i to zone j and

b = sensitivity coefficient.

A gravity type expression has been chosen because it is well established among transport planners. A logarithmic transformation is used because it is found to reduce the sensitivity of accessibility to the value of b. This is a desirable feature since b cannot normally be estimated accurately. As discussed in Section 5.2.3, a similar expression has been used by Pike et al. (1976) for the evaluation of alternative bus schemes in the Telford Transportation Study.

The expression has a negative sign because it is related to the cost of travel. For example, an increas in the cost of travel will increase the absolute value of accessibility but decrease its actual value.

An example will be used to show how to measure the accessibility of zone 4 for travelling to work by the network and schedule provided. The zonal distribution of jobs has been assumed as follows:

Zone	Jobs
1	500
2	200
3	100
5	50

5.3.2 How to Construct a Weighted Trip Time Verses Time of Day Graph

The most important parameter to be estimated in the accessibility expression is the trip cost. In the present study, it is assumed to be represented by a weighted trip time consisting of the following components:

(a) walking time (access modes by bicycle or car have been excluded in the analysis, see Section 2.8);

(b) waiting time;

(c) in-vehicle time; and

(d) route change.

The weightings applied to the first three components are 1.9, 2.5 and 1 respectively (same as those used in Pike et al., 1976) and a route change is arbitarily assumed to have an extra time penalty of 5 minutes.

Using the path-finding algorithm described in Chapter 3, it is possible to find an optimal path with minimum weighted trip time to a specified destination when given a starting time from the origin. As the algorithm can be iterated for different starting times, it is therefore possible to obtain the variation of weighted trip time with starting time.

Data for plotting a weighted trip time versus time of day graph can be obtained by repeating the path-finding algorithm for different starting times. The procedure consists of finding all trip starting times in which the initial waiting time is zero. This can easily be calculated from the route schedules. However, when there are several stations which are accessible from the origin and frequent vehicle departures from these stations, a large number of starting times will have to be considered. For example, in the case of zone 4, it is necessary to consider all times at which there is a transit vehicle departing from nodes 21 or 18 and there are a total of 64 different departure times in a day. It is therefore necessary to have a computerized procedure so that the data can be generated automatically.

From the data generated, the weighted trip time corresponding to each starting time is plotted out as troughs on the graph. This is because that at these points the initial waiting time is zero. This means that at these starting times a traveller would arrive at an origin station and immediately board a vehicle. However, if the traveller arrives slightly later, he would miss the vehicle and his waiting time would then be a maximum. Therefore at the same x-coordinate of a trough, a crest is also formed. The position of this crest is obtained by calculating the waiting time for the next vehicle departure. When time progresses from one vehicle departure time to another, the initial waiting time reduces from a maximum value to zero. Since the weighting for waiting time is set to 2.5, the slope of the graph is either vertical or -2.5. For this reason, it is only necessary to find the troughs of the graph. By plotting a vertical line and a line of slope equal to -2.5 from each trough, the graph can be completed.

Not every vehicle departure at an origin station can provide an optimal path with minimum weighted trip time. Sometimes a traveller may decide not to board the first vehicle that is departing from the station but wait for another vehicle departing at a later time but on a different route. In this case, a trough and crest on the graph will converge to the same point. These points will become apparent during the graph plotting process.

Examples of this types of graphs for trips from zone 4 to the other zones are shown in Figures 5.2A, B, C and D.

From these graphs, the average weighted trip time from zone 4 to the other zones for any given period of time can be obtained. This is calculated from the area under the graph for this time period divided by the length of the time period. For example the average weighted trip time for two selected periods are shown in Table 5.1.

From zone 4 to zone	Average weighted trip time (minutes) 8 am to 9 am	daily
1	61	68
2	75	86
3	100	110
5	55	77

Table 5.1 Trip time from zone 4 to other zones

5.3.3 Determination of Temporal Accessibility

Using the average weighted trip times obtained for different time periods, the temporal accessibility for these time

Figure 5.2A Variation of weighted trip time with starting time: zone 4 to zone 1

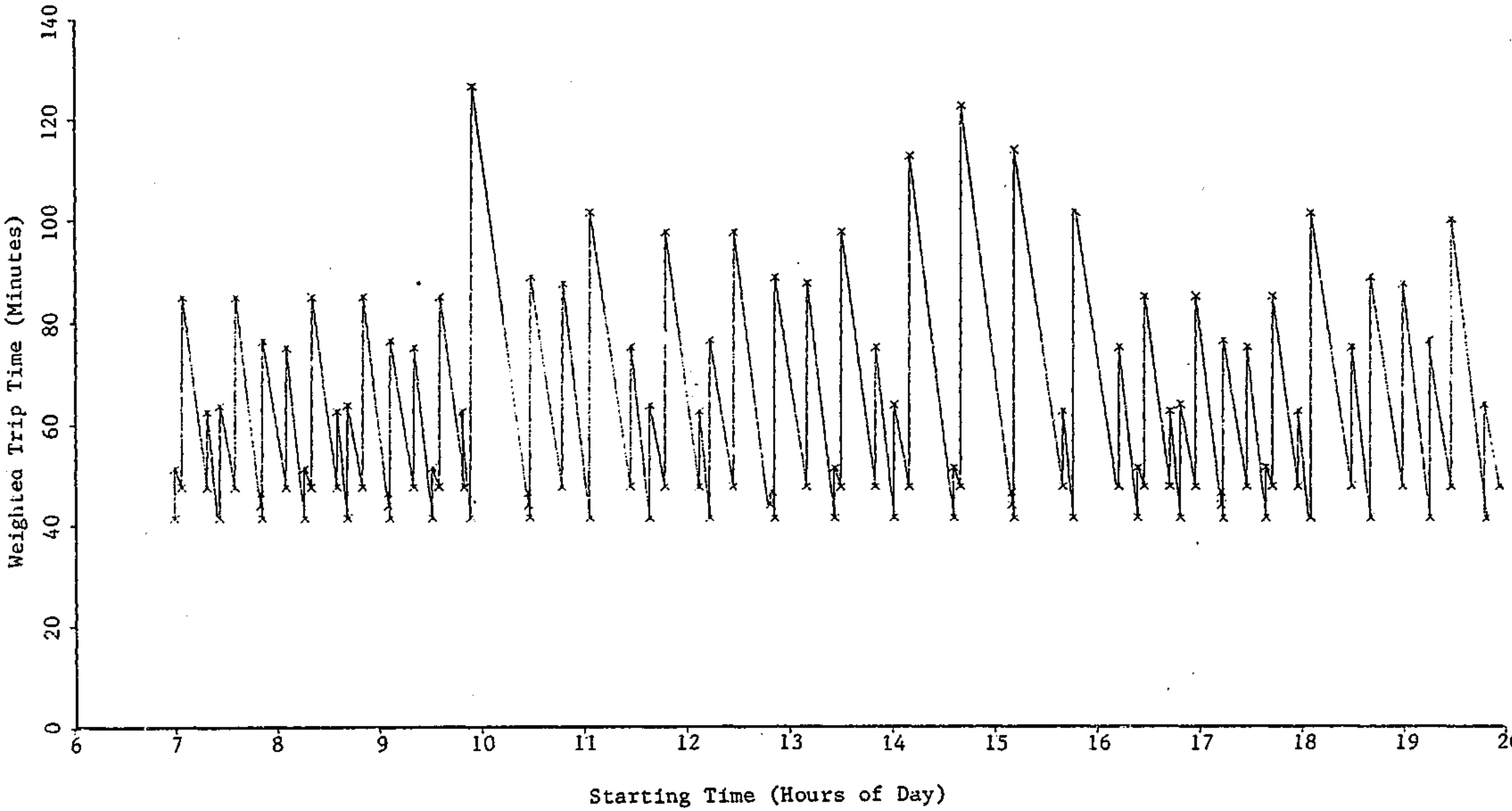


Figure 5.2B Variation of weighted trip time with starting time: zone 4 to zone 2

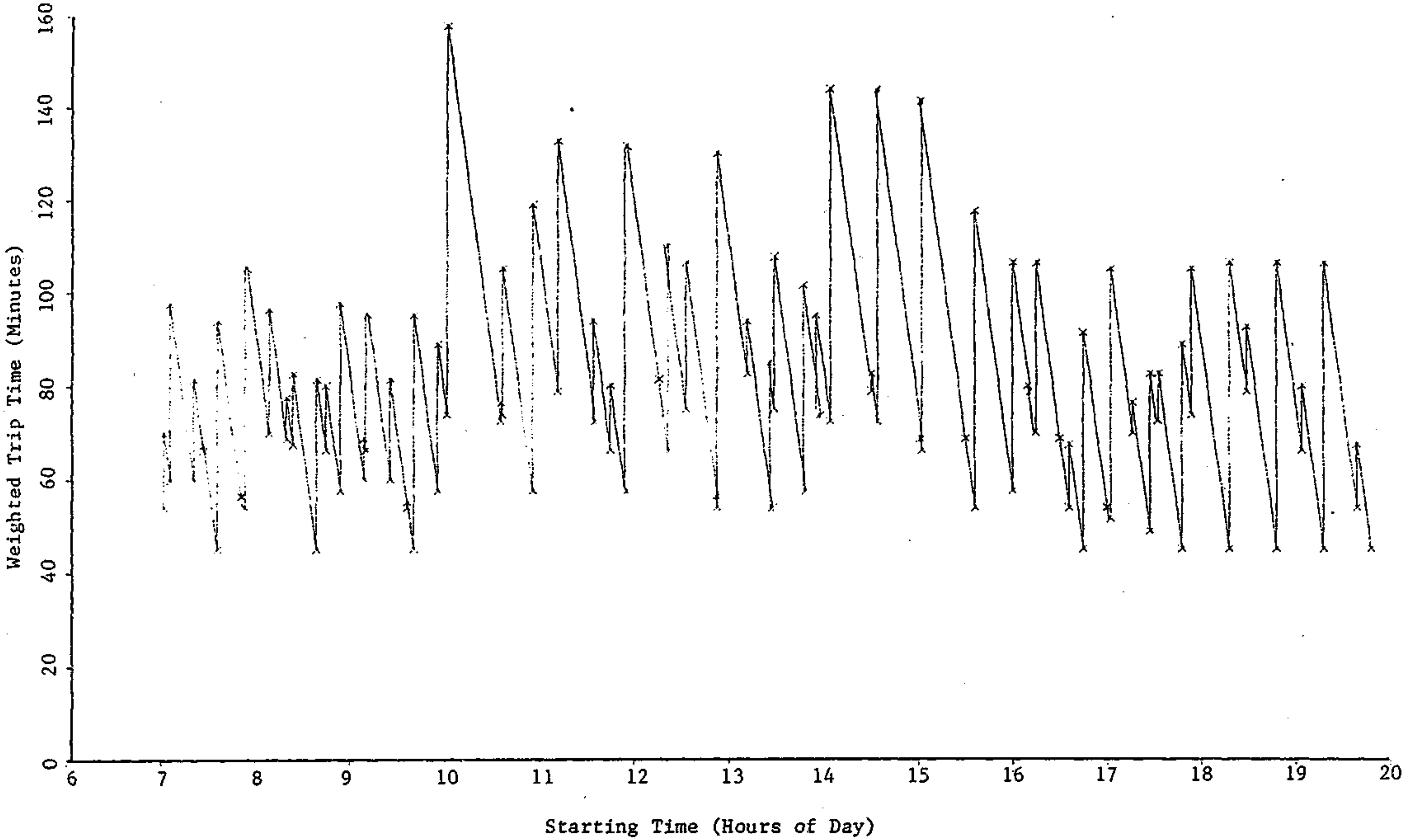


Figure 5.2C Variation of weighted trip time with starting time: zone 4 to zone 3

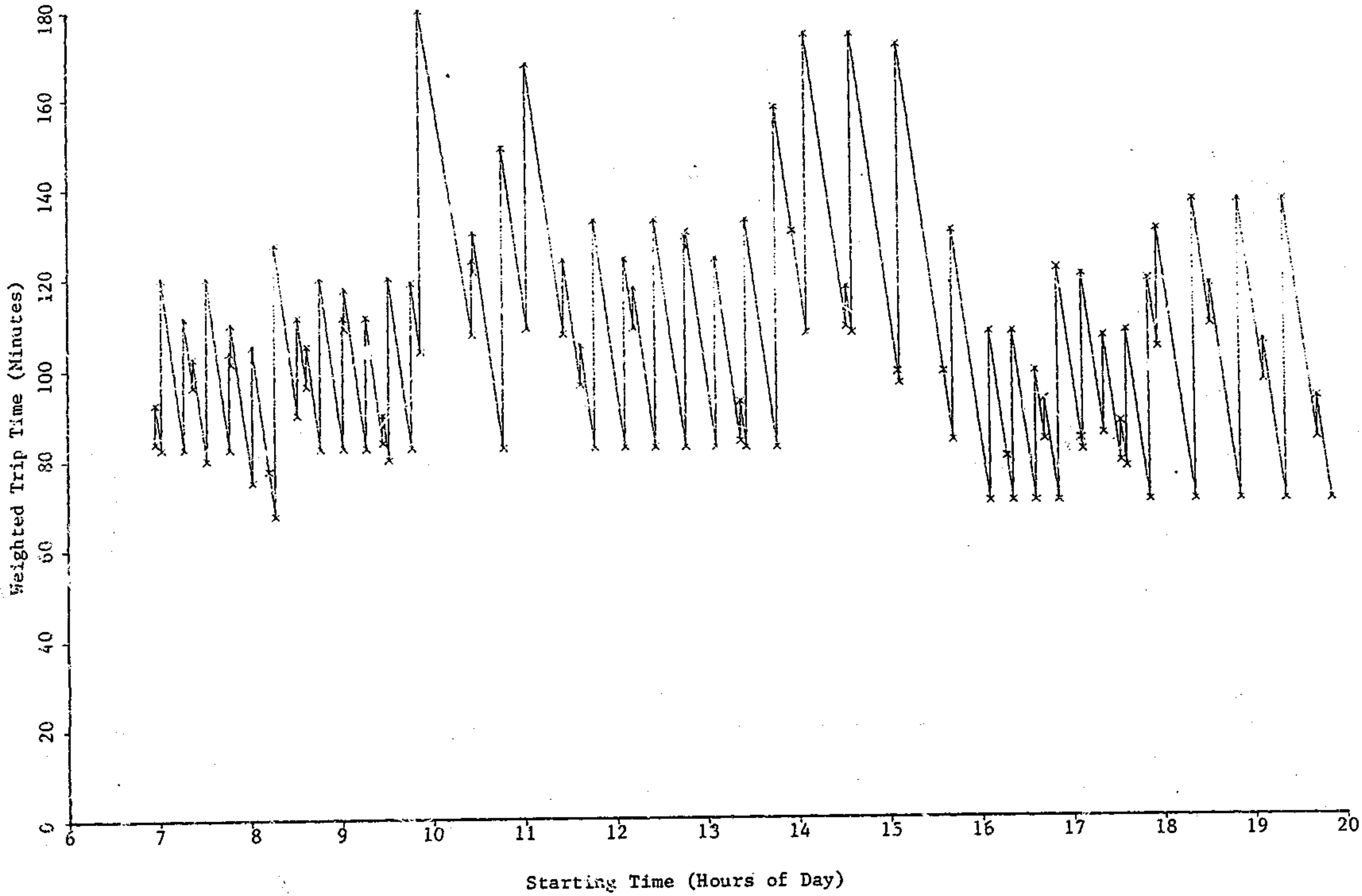
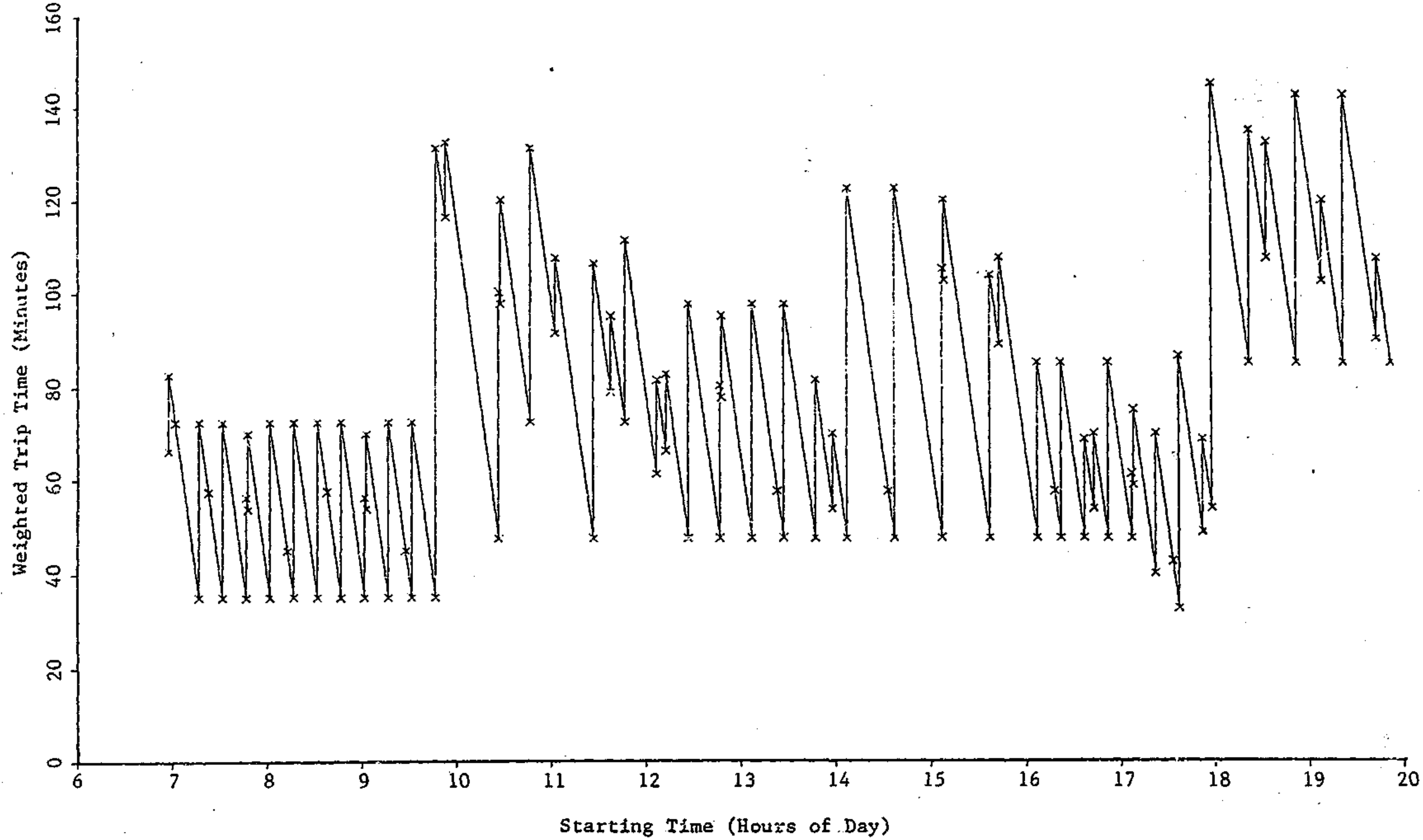


Figure 5.2D Variation of weighted trip time with starting time: zone 4 to zone 5



periods can then be calculated. For example, the 8 to 9 am and daily accessibilities of zone 4 for work trips are shown in Table 5.2.

Sensitivity coefficient	b =0.17	b =0.22
Based on average weighted trip time between 8 to 9 am	-62.5	-61.9
Based on average daily weighted trip time	-70.9	-70.3

Table 5.2 Temporal accessibility of zone 4

The results confirm that with the expression used, the accessibility is sensitive to changes in weighted trip time but is not too sensitive to changes in the value of b.

5.3.4 Application to Evaluation of Transit Schedules

As accessibility has been calculated from weighted trip time based on route schedules, it can be used in the evaluation of alternative route schedules. For example, suppose it is decided to cancel the 8:17 am service of route B3 and the 8:19 service of route A4, then the recalculated average weighted trip times from zone 4 to the other zones are shown in Table 5.3 and the resultant change in accessibility calculated with b=0.17 is shown in Table 5.4.

From zone 4 to zone	Average weighted trip time (minutes) 8 am to 9 am	daily
1	73	69
2	80	86
3	114	112
5	64	78

Table 5.3 Revised trip time from zone 4 to other zones

	new schedules	old schedules
Based on average weighted trip time between 8 to 9 am	-73.4	-62.5
Based on average daily weighted trip time	-71.8	-70.9

Table 5.4 Revised temporal accessibility of zone 4.

The results show that the change in schedules causes a great reduction in morning peak accessibility but only a small reduction in daily accessibility.

5.3.5 Accessibility based on Arrival Time rather than Starting Time

In the above analysis, it has been assumed that there is no waiting time at the destination. This assumption is reasonable for some types of trips, such as work trips in which the working hours are flexible or shopping trips. However, in other instances, such as attendance at schools, or workers with rigid working hours, waiting time at the destination end of the trip should be included as part of the weighted time of the trip.

Using the model, it is also possible to find the weighted time of a trip when the arrival time at the destination is specified. In this case, the waiting time at the origin end is assumed to be zero.

As an example, assume that the workers living in zone 4 work in two shifts, the first half has to arrive at work at an exact time falling randomly within the period 8:45 am to 9:00 am while the other half has to arrive at work at an exact time falling randomly within the period from 2:20 pm to 2:35 pm.

Then in order to measure the accessibility of zone 4 for these workers, the average weighted times of trips to the other zones must be based on these new conditions.

A graph of weighted trip time versus arrival time can be constructed. A path-finding algorithm which can find the optimal path with a specified arrival time is used for generating the necessary data. From the schedules, a set of different arrival times at the destination in which the waiting time is zero is obtained and the minimum weighted trip times are found for this set of arrival times. The data is then plotted as troughs in the graph. From these troughs, vertical lines and lines of slope equal to 2.5 are drawn to complete the graph.

As an example, a graph showing the variation of weighted trip time with arrival time for trips from zone 4 to zone 1 is given in Figure 5.3.

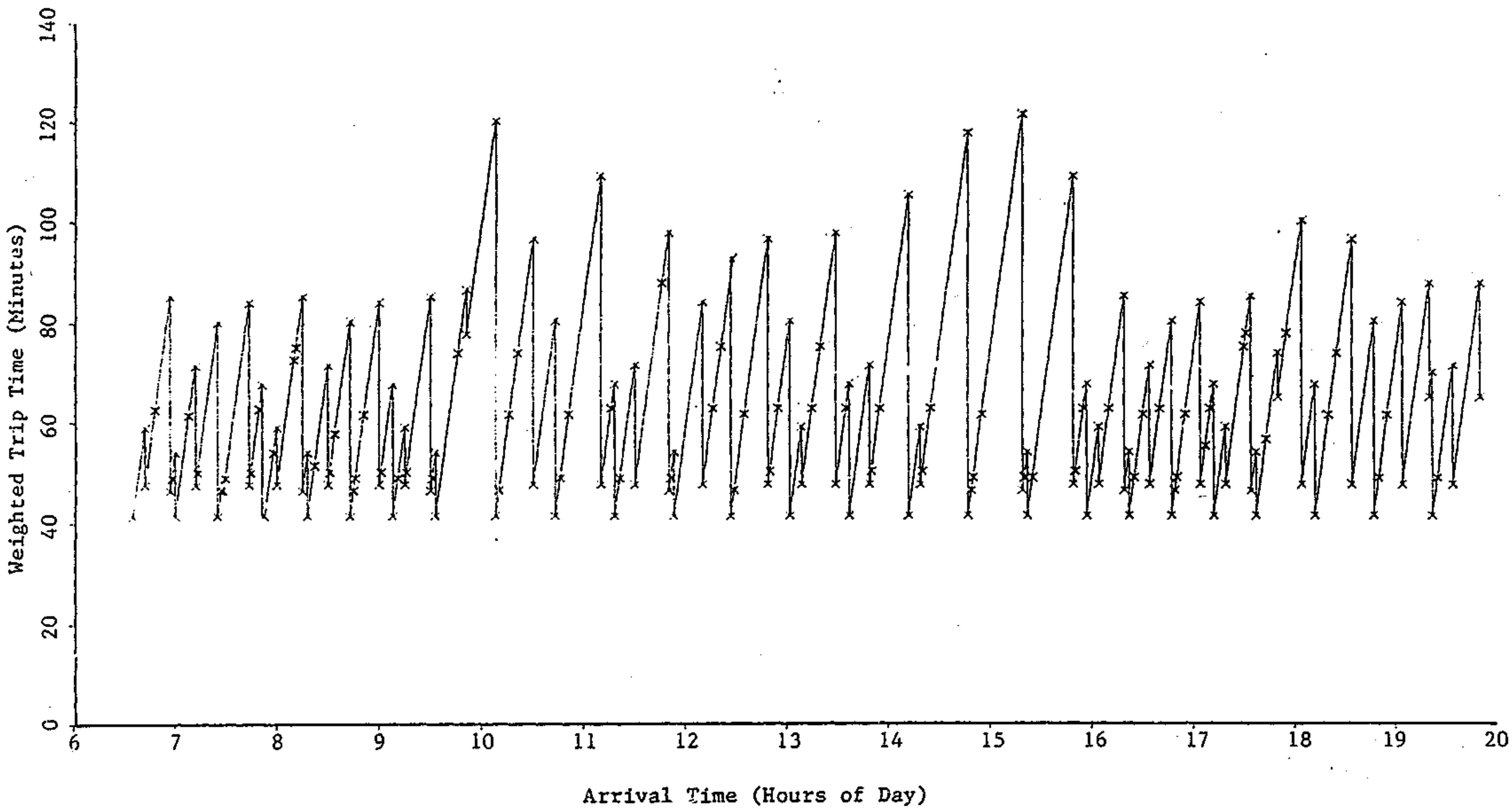
From the graph, the average weighted trip time for any arrival time interval can be determined. Using the results, an arrival time based accessibility can then be calculated.

The arrival time based accessibility measured for the two shifts of workers residing in zone 4 is shown in Table 5.5.

From zone 4 to zone	Average weighted trip time (minutes) 8:45 am to 9:00 am	2:20 pm to 2:35 pm
1	55	71
2	84	78
3	101	113
5	55	75
Accessibility with b =0.17	-57.5	-73.2

Table 5.5 Arrival time based accessibility of zone 4

Figure 5.3 Variation of weighted trip time with arrival time: zone 4 to zone 1



5.3.6 Computer Requirements

It would be more meaningful to discuss the computer requirements of running the model for an actual city rather than for the hypothetical example shown. In the case of Melbourne, for a transit network with approximately 500 nodes and 3000 arcs consisting of rail, tram and government operated bus services, generating the data on the daily variation of weighted trip time from one zone to another would require a core capacity of about 150 K bytes and CPU time of approximately 14 minutes. The total CPU time requirement for the calculation of temporal accessibility for a zone is therefore equal to 14 times the number of zones considered in the analysis.

5.4 Discussion

5.4.1 Introduction

While the proposed accessibility model has provided some improved techniques in the measurement of transit accessibility a number of simplifying assumptions have been adopted. This section will be devoted to a discussion of whether these assumptions are justifiable and if not what further research is needed to improve accessibility modelling.

5.4.2 Determination of Average Weighted Trip Time

Firstly, when calculating temporal accessibility, the average weighted trip time during a specified period is used. This is only correct when passenger starting times are random. However, in many instances, passenger arrivals at a transit stop are not random. Danas, 1980 suggested that there can be two levels of variation in passenger arrival rates:

(a) on a macroscopic level, the variation is caused by the change in travel demand. For example, the passenger demand may increase gradually from a low level to a maximum level and then decrease during a peak period.

(b) on a microscopic level, if route headways are small, arrivals may be unrelated to schedules. However, if headways are large, passenger arrivals would be related to vehicle departure times.

If passenger arrivals are assumed to be random, a supply type accessibility measure can be obtained as shown previously in the case study. It can be regarded as a measure of the ease and opportunity provided for travelling to various destinations and at different times without regard to the demand of travellers for using these services.

As pointed out by Morris et al. (1979):

> "There is a basic dilemma in choosing between "process" indicators (measures of the supply characteristics of the system and/or individuals) and "outcome" indicators (such as actual use and levels of satisfaction)....This basic conflict gives rise to a range of accessibility measures which differ in terms of their behavioural component."

When used as an evaluation index, Morris et al. (1979) preferred the use of "process" indicators rather than "outcome" indicators. The main reason is that when using "outcome" indicators, it is difficult to disentangle the influence of choices and constraints. For example, passengers may start their trip early because the route schedules are known to be variable and not because of preference for travelling at a particular time. Hence, it is not always correct to take an existing travel demand pattern as the preferred travel pattern and use it as a basis for evaluation. Another problem is that when comparing hypothetical schemes the demand pattern is not known.

From the above, it is considered that the assumption of random starting times in accessibility modelling is justified. Without this assumption, the modelling of temporal accessibility would be much more complex. In this case, apart from the need to obtain the arrival pattern of travellers at stations, there is the additional problem of determining an appropriate average travel cost for trips starting within a given time period. As the aim is to measure accessibility based on actual behaviour, a logit type model would be more suitable than a gravity type model. Therefore, under a logit formulation accessibility would be related to the maximum benefits perceived by travellers from an available composite set of choices over travel time, destinations, routes and modes. At present, the development of such a model to include choice over time of travel via a schedule-based transit system is still a largely un-researched area.

5.4.3 Accessibility on a Loaded or Unloaded Network?

In the model the effects of network overloading have not been accounted for. The path-finding algorithm used assumes that every transit vehicle in the network is operating within available capacity. Hence it can be regarded as the accessibility measured on an unloaded network.

The question of whether accessibility should be measured on a loaded or unloaded network is again related to the argument of whether "outcome" or "process" indicators should be used in accessibility measures. As an evaluation criterion, it may be desirable to exclude the effects of demand on travel costs. The reason is that for example, if a transit system is improved, it may attract more passengers. In this case, the increase in accessibility as a result of the improvements can be more accurately reflected by "process" indicators rather than "outcome" indicators.

When accessibility is intended to be used as a determinant of travel choice, then it should be measured on a loaded network. The problems related to the development of a path-finding algorithm that can account for capacity restraint on a scheduled transit network is discussed in Chapter 7.

5.4.4 Idle-Time

In a transit trip, there are three different categories of waiting times:

(a) waiting at the origin station;

(b) waiting during a route interchange and

(c) waiting at the destination.

One characteristic of a transit trip is that the transit user is constrained by transit route schedules and this may force him to start from the origin or arrive at the destination earlier than the desired time. Starkie (1971) termed the excess time waiting at the destination as "idle-time" and suggested that this could be caused by the following factors:

(a) schedule delay, which is caused by the fact that scheduled transit services often do not fit in with the traveller's desired arrival time;

(b) trip time variations and unreliability make it difficult for the traveller to predict accurately the trip time and hence very frequently travellers allow more time for their trip, resulting in early arrival.

Similarly, "idle-time" could also occur at the origin. For example, a traveller may have to wait at the origin station due to schedule delay or the unpunctuality in the departure time of transit vehicles.

As accessibility is a function of both transport service and activities, the services provided by a scheduled transit network in relation to the time of participation in activities must be accurately reflected in the model. The present model can to a certain degree account for this. The case study has already demonstrated how accessibility can be measured with specified trip starting or arrival times. However, although idle-time time at the origin or destination has been included in the total trip time, the idle-time due to schedule delay only has been accounted for. The effect of schedule variations and/or unreliability on trip time is not considered in the present model.

When using accessibility as an evaluation criterion, idle-time due to schedule delay should be considered. For example, when measuring the accessibility for work trips, only those parts of the transport service operating within time periods suitable to commuters are selected in the determination of accessibility. If it is possible to obtain estimates of travellers' allowances for idle-time due to trip time variations and unreliability, then it could also be considered in the evaluation process. Abkowitz (1981) used expected loss parameters for early and late arrival in the development of a logit model for predicting the departure time of work trips by transit users. The model requires considerations of different types of destinations and activities, such as, for example, the flexibility of working hours for workers and trip time variability. More research is be required in order to extend the present accessibility model to include this particular aspect.

Apart from the prediction of the magnitude of idle-time, its perceived cost to travellers must also be considered. In the present model, all waiting times are given the same weighting. The path-finding algorithm used in the model can, however, with some minor modifications, accept different weightings for different categories of waiting times.

Previous studies on the value of travel time have rarely considered idle-time and it is not possible to find any reported research on its perceived value. Compared to the other components of travelling time, idle-time is different in nature in many aspects. For example, it can be either positive or negative (early or late arrival), and its potential utility much more variable (shopping, eating in coffee shops etc.). More research is warranted in this area for application in accessibility modelling.

CHAPTER 6

COMPUTER GENERATED TRAVEL INFORMATION FOR URBAN TRANSIT NETWORKS

CHAPTER 6

COMPUTER GENERATED TRAVEL INFORMATION FOR URBAN TRANSIT NETWORKS

6.1 Introduction

It is generally acknowledged that there is a need for providing adequate travel information to public transport riders. It is believed that adequate guidance on the usage of a public transport system helps travellers to select the optimal route and time for travel and hence make the most efficient use of the system. It is also a means whereby improvements to the transit system can be communicated to users and non-users. Furthermore, knowledge concerning the properties of the transit system adds to the confidence in its usage and this is often an important attribute in the attraction of people to use the system. In many instances, travellers are reluctant to use public transport due to insufficient knowledge concerning transit services. It has been shown, for example, by Brog (1982) that knowledge of the transit system is a constraining influence in people's mode choice decisions on about 25% of occasions. Surveys conducted by Ellson and Tebb (1978,1981) also showed that the distribution of information leaflets on public transport services is effective in attracting more people to use the public transport system. Providing a good passenger information system could therefore be an effective means of reducing private car trips and hence congestion on the roads.

There has recently been much interest among researchers on what is the most appropriate travel information that should be given to travellers. Diewald, Frost and Bamberg (1983) identify six major components of a passenger information system as consisting of maps, timetables, signs, distributed information, people and telephone information systems. The purpose of these components varies from providing either general information on the services available or specific information to enable a passenger to plan a particular trip using the system. The need for route guidance often increases with the complexity of the transit services provided.

The present research is concerned with the requirements of providing route guidance, and the feasibility of using computers to generate the guidance information.

The need for better information about public transport services has been paralleled, in recent years, by significant changes in computer technology. The introduction of powerful and inexpensive minicomputers and microcomputers has made possible many applications which were not previously feasible. One such application is the use of computers to generate travel information for users of the public transport system. A popular concept is the use of stand-alone computers at decentralised sites to provide travellers with guidance information. In this way, travellers can obtain accurate guidance information without the need for personal attendance.

Although several authors have shown that a computer system of this type can be established, there have been several areas of further research recommended by these studies. The aim of the present chapter is to review previous research on the subject and discuss potential areas for improvement. A study is reported on the feasibility of establishing a travel information system for the transit network in Melbourne, Australia. The city of Melbourne is served by a comprehensive multi-modal transit system consisting of train, tram and bus services. The study consisted of the establishment of a computerized database of route timetables; development of computer software for finding time-dependent optimal paths within the multi-modal transit network; and comparison of various methods for presenting travel information to system users. Experience gained from the study on possible improvements to existing travel information systems is then presented.

The scope of this chapter is limited to the generation of information about complete trips through a transit system. It is not concerned with the question of providing information at transit stops about the next scheduled arrival (e.g. the Telerider system in Canada and the U.S.), nor is it concerned with real-time information systems for transit passengers whereby information is updated by means of an automatic vehicle monitoring system (e.g. Gorstein and Tilles, 1981; Forsyth, 1985; James, 1985). Rather it is concerned with systems whereby passengers can obtain route-planning information which has been generated by computer. These systems may be used as part of a telephone information system, or else the traveller can access the computer directly, or else the results are stored in some user-accessible format (such as microfiche).

6.2 Review of Previous Research

One of the first comprehensive studies on the establishment of computerized transit travel information systems was made by the Transport and Road Research Laboratory in England (Ellson and Tebb, 1978; Pickett, 1978, 1980, 1981, 1982). The study was conducted in several stages and included:

(a) an assessment of the costs of establishment and the benefits that can be derived from the system;

(b) the development of computer software to generate travel information; and

(c) a trial system establishment at Wiltshire- a county in England with a population of approximately half a million.

The trial system was developed for an inter-urban transit network consisting of train and coach services between towns. The computer software developed was able to find, from a database of route timetables, the quickest path, or cheapest path, or path involving the least number of route changes. The output from the computer program which consisted of travel alternatives between towns at various times during the day was stored in microfiche and placed in travel information offices and public libraries. Although analysis showed that substantial benefits could be derived, results from the trial showed that, due to a number of reasons, both travel enquiry staff and the public were not too enthusiastic about using the system.

The main reasons given for its lack of acceptance were that the publicity campaign did not induce travellers to try out the system and staff in the libraries and information offices subsequently forgot how the system worked and how to use the microfiche readers. The list of trips included on the microfiche were predominantly multi-modal, but these types of trips were rarely required by the travelling public. In addition, it could be argued that the information was not readily available at the time and place when travellers needed it most, and hence travellers perceived that it was not worth the trouble to obtain the information.

Microfiche storage of transit information was also used, but as part of a telephone information system, by the Washington Metropolitan Area Transit Authority (WMATA) in an early version of their passenger information system (Diewald et.al 1983). They found, however, that frequent changes to routes and schedules made production of up-to-date microfiches time-consuming and expensive. The microfiches were also not suited for use in the generation of complex itineraries. As a result, the WMATA system was converted to an Automated Information Directory Service (AIDS) which utilised a computer to generate travel information (Wilson-Hill, 1982). A similar computer-aided system was also installed in Los Angeles (Wilson-Hill, 1981).

Other computerized travel information systems have been reported in the literature. The Personalized Public Transport Directory described by Williamson and Miller (1981) provides an on-line journey enquiry system in which the user indicates their zone of origin, their required destination and the time at which they wish to begin their journey. The system responds by indicating the quickest route between these points at the time the user wishes to travel. The route selection allows for walking time from a hypothetical zone centroid to the network and makes allowance for buses not running to schedule by assuming that they may run up to three minutes late. Using a relatively simple test database in North Manchester they found that on-line enquiries had a response time of 1 to 4 seconds. For a metropolitan network, they estimated that the response time would more likely to be in the range from 10 to 40 seconds.

In discussions with operators, they found that several improvements to the system would be desirable. Firstly, it would be desirable to output a range of travel options, rather than just the quickest path, from which the user could make a selection. Secondly, the quickest path may contain a large number of interchanges in order to shorten the travel time. It would be desirable to have a route selection criteria which was able to minimise the number of interchanges (and other undesirable parts of the journey such as waiting time). Thirdly, it would be desirable to allow the user to specify the latest arrival time at the destination, rather than the earliest departure time from the origin.

A major difference between these studies lies in the input and output devices adopted. Whereas the TRRL study produced travel information off-line which was then accessed from microfiche, most of the recent systems engage an on-line interactive interface in which the traveller inputs travel desires and the computer outputs optimal paths. The system developed by Williamson and Miller (1981) produces travel information in a narrative form, while the system developed by Hayashi, Itoh and Suzuki (1983) in Japan engages input and output interfaces with graphic display.

The Hamburg automated telephone information system (Diewald et.al., 1983) is fully automated, with no human information agents, and is capable of user-computer dialogue, with the computer speaking by means of synthetic voice generation and the user "speaking" by dialing code numbers on the telephone. An alternative method of using the system is by means of remote terminals located at strategic points within the Hamburg area. The user communicates via a keyboard and the computer prints out a personalised travel itinerary for the user.

An automatic route-planning and mapping system which is being developed in the University of Glasgow (Giffen, 1985) suggests a system with microcomputers situated at user's homes connected to a main frame computer via telephone lines. The main frame computer stores the database and carries out the computations while the microcomputer is the interface for receiving the input data on travel requirements from the user and displaying the output which is in the form of guidance maps. Although the system is being developed for producing road maps to guide motorists, it has however demonstrated some interesting features which are also applicable to the development of transit guidance systems. One of these features is the use of cartography for the display and printing of maps. Hence, in a transit information system, apart from just displaying a network diagram indicating the proposed route, a map of the area covering the route can also be displayed or printed.

A review of previous research has shown that few studies have considered the practicability of establishing a travel information system for the transit network of a large city. It is also considered that more advanced path-finding algorithms can be developed to improve the quality of travel information that can be generated.

6.3 Path-Finding Algorithms

A Dijkstra type algorithm (Dijkstra, 1959) is commonly used in travel information systems for finding the quickest path from origin to destination (Hayashi et.al., 1982). In this type of algorithm, when more than one quickest path exists, only one path is found. The choice among the alternative quickest paths are then quite arbitrary, and is normally based on the ability of a path to reach intermediate nodes as early as possible, but not based on other more appropriate criteria such as minimising walking times, number of route changes required etc. Experience on the use of this type of algorithm for the transit network in Melbourne, for example, showed that quickest paths generated are often unsatisfactory, due to the inability to select a path with the minimum number of route changes when several alternative quickest paths are present.

In some cases, unnecessary route changes are made just in order to arrive at an intermediate stop sooner without causing a reduction in total trip time.

There are definite advantages in developing more advanced path-finding algorithms. Apart from the need to find the correct quickest path, travellers often wish to optimise the generalised cost of a trip rather than trip time. The present research has, however, concentrated it efforts on developing an algorithm that can find an optimal path with minimum weighted trip time. The main reason for excluding fares from the generalised cost function is discussed in Section 3.4.1.

Another useful feature of the path-finding algorithm would be the ability to limit the number of transfers required in a path to a user-specified number. While this feature has not been included in the present algorithm developed, from the description of the path generation process shown in Appendix 3, it is quite easy to amend the algorithm to include this feature.

Apart from finding paths connecting transit stations or zone centroids, there may also be a need to develop an algorithm for finding paths between any two locations as defined by their street map co-ordinates. For trips within a multi-modal transit network, there are often a number of stations within walking distance from the traveller's origin or destination. The optimal path generated should include access links in a door-to-door path rather than just a path between stations or zone centroids. The algorithm can then be used to generate paths to specific named locations, such as shopping centres, the airport, or other places of interest to the general public.

Finally, the algorithm should be efficient so that information processing and generation is fast and accurate. The computer software must also be flexible and able to provide the system user a number of options in the type of information required for input and output.

In connection with the Melbourne study, computer software which can find door-to-door paths with minimum weighted time has been developed. A description of the theory in the development of the network model and path-finding algorithms is given in Chapters 2 and 3 while a user's manual of the software is contained in Volume 2 of this thesis.

6.4 Network Description

The travel information system requires for its operation a computerized database of route timetables, station locations and interchange walk links. For network models used in trip assignment the network database need not be extremely accurate, for example, a transit route can usually be approximately represented by coding the locations of a few major stops along the route and operating schedules at these stops. However, for network models used as travel guidance, the network database must be updated and accurate and especially with regard to the locations of stops along a transit route and the operating schedules. Otherwise, when errors are detected in the guidance information generated, the system will lose credibility and its potential benefit could be much reduced. As discussed in Section 2.5, for a large or complex transit network, data coding can be a time consuming task. It is considered that computer software must be carefully designed to faciliate the coding and checking of network data.

Apart from data coding, the structure of the network model also needs to be different from those used in trip assignment. The reason is that the model would be very inefficient if all transit stops are represented as nodes in the network. The technique which has been developed for modelling detailed networks is described in Section 2.9.

6.5 Service Reliability

A major difficulty of establishing transit information systems is how to deal with system unreliability. When services do not operate exactly according to published timetables, the travel information generated by the system may lose its usefulness and credibility. A number of approaches can be considered for overcoming this problem.

First, route schedules can be updated continuously on-line using information based on current operating conditions. Such a system is only possible either when the transit network is simple, or when there is equipment available to continuously detect and monitor the locations and operation of transit vehicles within the transit network.

Second, system reliability can be taken as one of the criteria affecting route choice. Thus, a route is chosen not entirely based on minimising the weighted, average time of a trip but consideration is also given to previous records of service reliability on different routes. An estimate of trip time variability could then be included in the travel information generated. This approach however, would increase substantially the size of the database required and the processing time of the path-finding program. Even if probability concepts were incorporated into the results, however, this may not be readily understood by the general public who are expected to use the system.

Third, the system user can be supplied with additional information which may become relevant when the selected routes do not operate as scheduled. For example, at critical points along the path recommended, such as at initial boarding or interchange stations, additional information can be given on the schedules of succeeding services. This would enable the traveller to replan his journey if the recommended route was affected by unexpected cancellations or variations of transit services.

On comparison, the third approach would seem to be the most suitable solution for most urban transit networks. The main reason is that there is no requirement to code additional data, and there is no radical change to the system software used. The system is also flexible in that the traveller has the option of deciding whether the additional information is to be included in the program output or not.

It needs to be stressed that the need for route guidance does not diminish with system unreliability. The approach which should be adopted is to try to develop information systems which can take into account this factor rather than abandon the concept due to the difficulties that may arise.

6.6 Input/Output Devices and Presentation of Travel Information

Many different types of input/output devices can be used. There are also various ways in which travel desires can be input and the relevant travel information presented to the user. When the path generated is door to door, and the software used is one which optimises a user supplied weighted time function, the most suitable input/output device is probably an on-line interactive keyboard terminal with a video screen. The required information is obtained from the traveller through successive questions appearing on the screen. An example of the type of questions asked and answers expected are shown as follows: (note that the exact wording and format of these questions is subject to refinement to enhance the user-friendliness of the system).

```
*************************************************************

  Enter the Melway map grid of your origin ?          68E5

  Do you wish to specify departure time (Y/N) ?       N

  Then enter latest arrival time (hr,min) ?           9,00

  Enter the map grid of your destination ?            2HA4

  The  default  weightings (any number from 1 to 9,  a  larger
  number    denote less preference) applied to  the  following
  travel   components are:

      in-vehicle travel = 1

      waiting at a station during a route change = 3

      walking = 5

      changing routes = 7

  Do you wish to change the weightings (Y/N) ?        Y

      in-vehicle travel ?                             1

      waiting at a station during a route change ?    2

      walking ?                                       3

      changing routes ?                               7

*************************************************************
```

The basic information which is to be presented to the traveller is a description of details of the optimal path generated, based on the travel desires which are input to the system. The description of transit routes and schedules can either be in narrative or diagrammatic form. The information can first be displayed on a video screen and a printed copy of the information can then be produced if required.

The choice between narrative and diagrammatic output will depend largely on the hardware available for output. Bartram (1980) has shown that users are better able to understand spatial information (such as a recommended transit route) if it is presented in a spatial (diagrammatic) format. For this reason, Giffen (1985) used a four colour printer/plotter to generate maps for use by drivers using a highway route-planning system. However, the generation of maps generally require more time and more expensive output equipment, and for this reason narrative outputs are often preferred.

Examples of path descriptions in narrative and diagrammatic form using the Melbourne rail network are shown in Figures 6.1, 6.2 and 6.3. The basic path descriptions consists of station names and locations, transit routes used and schedules. Optional additional information can be included to enable the traveller to replan the journey when the original path cannot be followed due to unforeseen circumstances. For travellers who are unfamiliar with the transit system, it may be necessary to provide them with other supplementary information, such as street maps which show the locations and names of transit routes and stops.

Basic information	Map code	Station name	Additional information
Origin at	68E5		
3 minute walk			
Depart 8:26 am	68E5	Glenhuntly	Next departure 8:37 am
Frankston Line			
Arrive 8:44 am	2FG5	Flinders Street	Arrive 8:56 am
3 minute walk			
Depart 8:47 am	2FG5	Princes Bridge	Next departure 9:02 am
Epping Line			
Arrive 8:51 am	2GH4	West Richmond	Arrive 9:06 am
5 minute walk			
Destination at	2HA4		

Figure 6.1 Travel information in narrative form

****

Leave your origin (Melway map code 68/E5) at 8.23 am and walk for 3 minutes to Glenhuntly Station (Melway map code 68/E5).

Catch the 8.26 am train on the Frankston line travelling towards Flinders Street.

(If you miss this train, the next one is at 8.37 am).

Alight at Flinders Street (Melway map code 2F/G5) arriving at 8.44 am.

The 8.37 am train will arrive at 8.56 am).

Walk to Princess Bridge Station (Melway map code 2F/G5) arriving at 8.47 am.

Catch the 8.47 am train on the Epping line travelling towards West Richmond.

(If you miss this train, the next one is at 9.02 am).

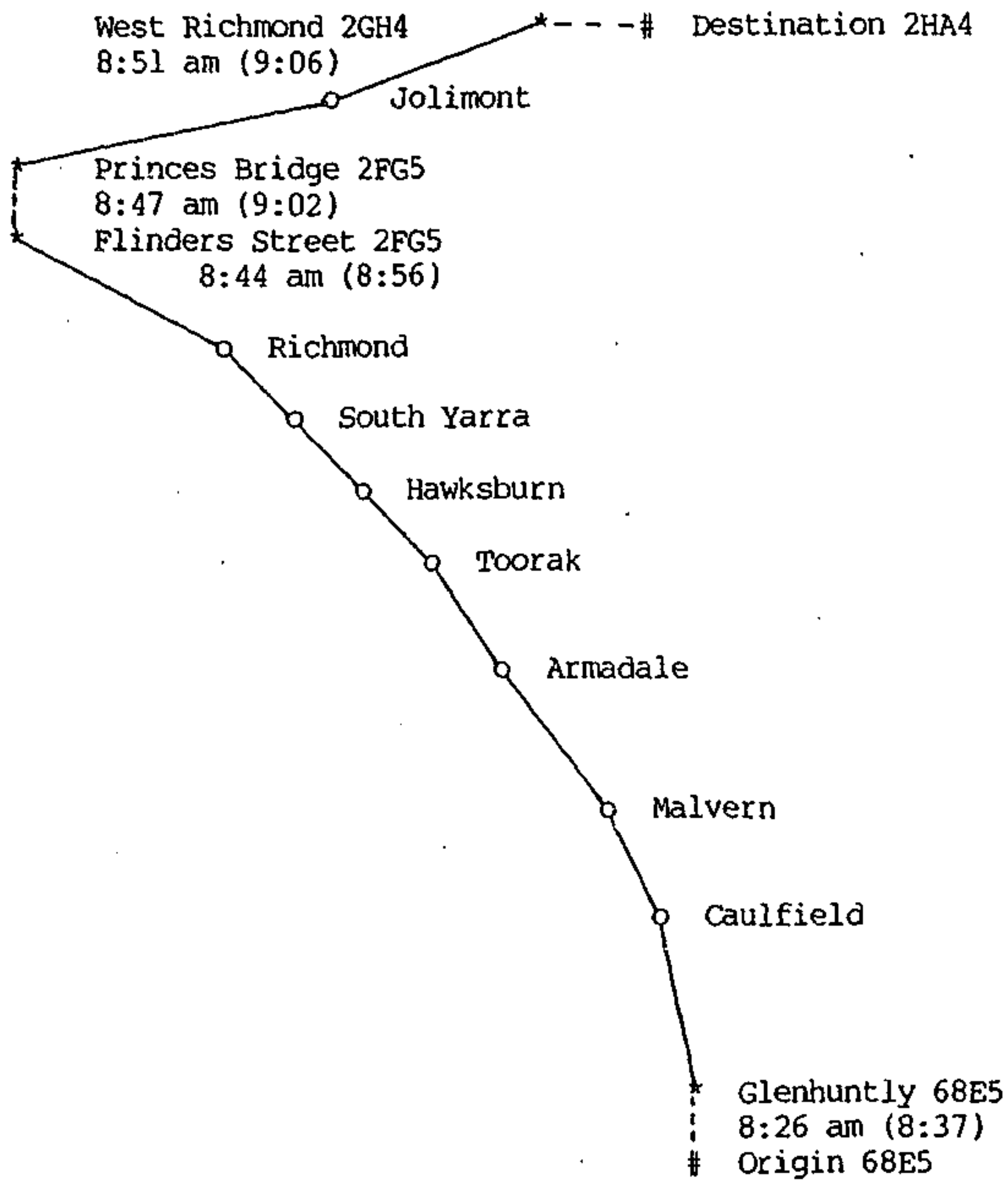
Alight at West Richmond (Melway map code 2G/H4) arriving at 8.52 am).

(The 9.02 am train will arrive at 9.06 am).

Walk to your destination (Melway map code 2H/A4) arriving at 8.56 am.

****

Figure 6.2 Travel information in alternative narrative form



Note: Times shown indicate scheduled vehicle departure or arrival times. Time shown inside brackets indicate next scheduled departure or arrival time.

Legend:
origin or destination specified
* start, end or interchange station
o other station type along the path
—— train
--- walk

Figure 6.3 Travel information in diagrammatic form.

On comparison, information in narrative form is simple and straight-forward to produce, it is also easy to understand. More information can in fact be inserted, such as the fare and the names of intermediate stations. However, this may make the information sheet more complicated. A travel information sheet in diagrammatic form has the advantage of being able to show the relative positions of transit routes and stations. This would be a useful feature for paths involving bus or tram routes where there are a lot of stops with unfamiliar names. The disadvantages are that it is more difficult to produce and the layout is more complex. Although it should be quite simple to plot a diagram showing the locations of transit routes and stations used, the difficulty lies in finding suitable spaces in the diagram for inserting the appropriate route information so that the two types of information do not overlap.

An alternative to the production of written output for the use of the traveller is for the computerized travel information system to be used to augment the information currently provided by telephone-based traveller information services. The computer program could be used by the operator, based on information supplied over the telephone by the traveller, and then the travel information generated by the program relayed verbally to the traveller over the telephone. Using the computer system in this way obviates the need for precise ergonomic design of the input/output interfaces, since the operators would all be relatively skilled in the use of the system.

6.7 Computer Requirements

In the Melbourne study, the software for generating travel information has been written in Fortran 77 for processing in main frame computers. The system is readily portable and has been used in several computers including a VAX 11/780, a Sperry 1100/61 and a Prime 250-II. Software for the graphic display of route information, however has been written in Basic language for processing in an IBM home computer. The programs have not been compiled into a package mainly because the programs for path generation and programs for graphic display of route information were developed in different computer systems. The other reason is that as discussed in Section 6.6 many different types of input/output devices can be used for a travel information system. There are also various ways in which travel desires can be input and the relevant travel information presented to the user. Therefore, it is only possible to compile a package when the specific requirements of a transit authority are known.

In the Melbourne transit network, the database of weekday timetables for all train, tram and MTA bus services requires a storage capacity of 400 k bytes. However, for a particular optimal path computation, it is only necessary to use a portion of the database, i.e. timetables for a selected 2 hour period within the day. Hence, the required capacity of the programs will generally be below 150 k bytes. The processing time required to generate an optimal path would of course depend on a number of factors, such as the length and complexity of the paths considered, and the type of path information detail required. For a network with approximately 500 nodes and 3000 arcs, the CPU time required for generating an optimal path with a user specified weighted time function will normally be less than one minute. Without actually coding private bus services, it is difficult to estimate accurately the effects of including these services on computer requirements. This will probably require a 20 per cent increase in the size of the database, the same percentage increase is estimated in the required program size and CPU time.

6.8 Costs And Benefits

Having outlined the development of the basic components of a computerised transit information system, it is useful to recapitulate on the major costs and benefits associated with such a system, so that system implementation can be considered in the light of these factors.

The main costs of establishing a transit travel information system are:

(a) staffing costs required for the setting up of a database of transit timetables;

(b) costs for development of computer software; and

(c) computer hardware costs.

The benefits that can be derived from the system by the transit operator are:

(a) attract more transit users through improved information services;

(b) supplement information provided by existing information services, such as telephone-based answering of queries concerning transit services, and the printing of route timetables; and

(c) when supplemented with other program software, the system can be used for the planning and evaluation of route schedules.

The benefits that can be derived from the system by transit users are:

(a) more detailed and up-to-date information on transit services; and

(b) accurate guidance on optimal use of transit services with user supplied travel preferences.

6.9 Conclusion

A review of previous research and results from the Melbourne study showed that there are several areas in which transit travel information systems can be improved.

First, there is a need for improved minimum path algorithms. For complex multi-modal transit networks in particular, conventional type quickest path algorithms may not always generate a satisfactory path. There is a need to develop algorithms which can generate optimal paths by minimising weighted trip time, where the weightings applied to each component in the weighted time function are user specified.

Second, path-finding software needs to be more flexible and provide the user with a number of options concerning the information to be input and output. For example, the algorithm should be able to generate paths connecting two points as specified by street map co-ordinates which are not necessarily transit stations and travelers can specify the arrival time of a trip rather than the departure time.

Finally, various types of input/output devices and various ways of presenting travel information can be employed. More research should be directed to this area, especially to the graphic display of route information.

CHAPTER 7

CONCLUSION

CHAPTER 7 CONCLUSION

Results of the present research show that a schedule-based computer network model has various applications in the planning of transit systems. In fact, for transit systems operating to fixed schedules, the network model developed can provide many extra functions which cannot be achieved by a conventional headway-based network model.

The method of coding a timetable database by computer and generating a schedule-based computer network description has been described. Particularly for large transit networks, the coding of a timetable database has been found to be a time consuming task. However, once a database is established, it can be used not only for the establishment of a network model but also used for other tasks of the transit authority, such as timetable planning and updating. The method of network generation is very flexible as the model user is given various options in network generation, such as the selection of a particular time period, routes or number of stops to be included in the network descriptions.

On comparison with other conventional headway-based network models, the new model has been shown to provide more accurate modelling, as well as having a wider scope of application in short-range transit planning. The applications of the model in trip assignment, accessibility measurement nd generation of passenger information have been developed and described in detail in this thesis.

Minimum path algorithms are indispensable elements of a network model. A review of previous minimum path algorithms showed that they are mostly not applicable to schedule-based networks. Furthermore, apart from quickest path algorithms, there is a need to develop other types of algorithms for particular needs of the model. For example, it is often necessary to find an optimal path with minimum weighted trip time where the weightings applied to the various components of the weighted trip time function are user supplied.

Computer software incorporating new algorithms which can perform these functions has been developed. The software has been fully tested with the existing rail, tram and government operated bus networks of the city of Melbourne, Australia.

Having developed a schedule-based computer network description with associated minimum path algorithms, the next stage of model development was on trip assignment.

Whilst a review of previous research has shown that schedule-based network assignment models were rarely used, the present research has, however, shown that more advanced assignment models could in fact be developed from such a model. Various new transit assignment techniques have been investigated and these were incorporated into two assignment models. The main characteristics of these models are that they are dynamic and schedule-based and hence are suitable for the short-range planning of transit systems and especially in the evaluation of transit route schedules. Although, as compared to conventional transit assignment models, more detailed input information on travel demand and longer processing time are required in these new models, however, with the recent advancement in methods for data collection and processing, these factors are now rarely constraining influences on the application of the models.

Capacity restraint procedures have not been incorporated into the assignment models because such procedures can not be developed easily. In a capacity restrained assignment, it is necessary to consider the chronological order in which travellers arrive at stations and board transit vehicles. Often, chronological simulation techniques will have to be applied to account for the formation and dissipation of queues at stations. Even if these techniques can be developed, it may not be justifiable to incorporate them into the present network model. The reason is that the present model is intended for dealing with larger transit systems and the necessary procedure for capacity restraint assignment would almost certainly be too complex and lengthy to be practical. Another reason is that capacity restraint assignment cannot account for the fact that demand changes as a resulting of system overloading. Travellers may shift to other modes or destinations, travel at other times of the day, etc. It may also be argued that capacity restraint is not an important feature in a planning model because it would conceal traveller's actual desires. On the other hand, assignment without capacity restraint would give a more diagnostic measure of the performance of the transit system.

Apart from trip assignment, accessibility measurement has recently become an important criterion in the planning and evaluation of transit systems. However, a literature review showed that previous transit accessibility measures used by planners were mostly simple measures and hence not suitable for the detailed evaluation of transit systems.

The present research has shown that more accurate transit accessibility measures can be obtained from a schedule-based model. For example, a method of measuring temporal transit accessibility has been invented and described. The importance of the new method is that as the measure is closely related to transit route schedules it can therefore be used in the planning of route schedules or the evaluation of alternative route schedules.

The present research has shown how a temporal transit accessibility based on the supply characteristics of transit systems and individuals can be measured. There is however a whole spectrum of different types of accessibility measures that are applicable to different planning tasks. For example, while a supply type accessibility measure is applicable to the evaluation of transit systems, a demand type accessibility measure, which is based on actual travel demand and level of satisfaction that can be derived from the transit system is required in travel demand modelling. There are still many unresolved problems in obtaining these measures. This thesis has only discussed these problems briefly. It has pointed out that many other factors should ideally be considered in these accessibility measures such as the effect of schedules on passenger arrival patterns at the origin station, network overloading and "idle-time", but it is often difficult to incorporate these factors into the measures. There is insufficient scope within the present study to deal with this subject in detail.

The present research has also provided several improvements in the development of computer generated travel information systems.

First, as the study is concerned with travel information systems in which computers are used to assist travellers to find optimal paths within complex transit networks, path-finding algorithms are therefore essential elements of the system. However, a literature review showed that existing minimum path algorithms used in these systems are not always satisfactory. The reason is that most of these algorithms can only be used to find the quickest path but not an optimal path with minimum weighted trip time, where the weightings applied to each component in the weighted time function are user specified. Such improved algorithms have been developed from the present research, and found to be most suitable for application to travel information systems.

Second, the path-finding software which has been developed in the present study is also flexible and able to provide the user with a number of options concerning the information to be input and output. For example, the software is able to generate paths connecting two points as specified by street map coordinates which are not necessarily transit stations. In addition, travellers can specify either the starting time or arrival time of the trip. The option to generate additional information to cater for unforeseen circumstances such as service unreliability is another characteristic of the software developed.

Third, various types of input/output devices and various ways of presented travel information have been investigated. In particular, software for displaying graphic route information on microcomputers has been developed.

Finally, using the existing transit system in the city of Melbourne, Australia as a case study, the research has demonstrated that it is feasible, using the software developed, to provide a computer generated transit travel information system for a complex multi-modal urban transit network in a large city.

APPENDIX 1A

THE AVERAGE DISTANCE BETWEEN TWO CELLS

APPENDIX 1A THE AVERAGE DISTANCE BETWEEN TWO CELLS

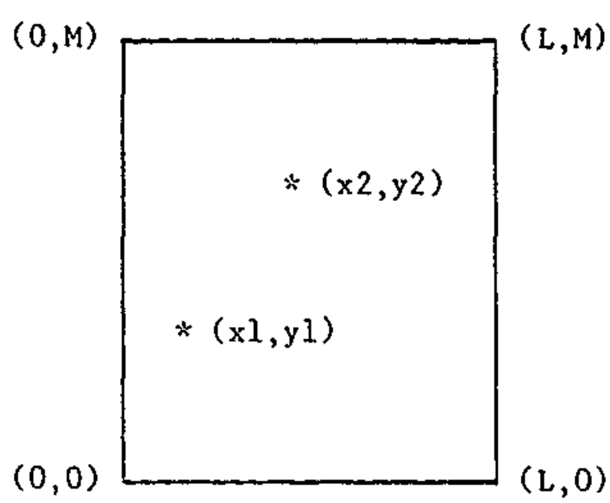
The Melbourne street network follows a gridiron pattern and the directions of the main roads are approximately pointing north-south and east-west. Most pedestrian footpaths are therefore aligned in the same directions. It would therefore be more appropriate to assume that the access distance between two points is not the direct distance but the sum of the x and y distances, where the x and y axes are parallel to the directions of the map grids, which is approximately pointing north-south and east-west.

As the location of a point is specified by the cell in which it is located in, it is necessary to calculate the average x and average y distances between two points when they are randomly located in the same or different cells. Based on the above assumptions, access distances can be calculated under two cases:

(a) when the two points are located in the same cell; and

(b) when the two points are located in different cells.

CASE 1 Average distance between two points located in the same cell.



Given the above rectangular cell measuring L by M units, two points with co-ordinates ($x1,y1$) and ($x2,y2$) are randomly located in the cell.

Firstly suppose that the first point has a fixed x co-ordinate with $x1 = u$ and the second point has a random x co-ordinate with $x2 = x$. The horizontal distance between the two points is then equal to:

$u - x$ (when u is greater than x) or

$x - u$ (when x is greater than u).

The average distance between the two points can be expressed as:

$$\frac{1}{L}\left[\int_0^u (u - x)\, dx + \int_u^L (x - u)dx \right]$$

$$= \frac{1}{L}\left\{ \left[ux - \frac{x^2}{2} \right]_0^u + \left[\frac{x^2}{2} - ux \right]_u^L \right\}$$

$$= \frac{1}{L}\left[u^2 - \frac{u^2}{2} + \frac{L^2}{2} - uL - \frac{u^2}{2} + u^2 \right]$$

$$= \frac{1}{L}\left[\frac{L^2}{2} - uL + u^2 \right]$$

However, the first point is in fact not fixed and hence u can have any random value lying between 0 and L. Therefore, the average distance between the two points is expressed as:

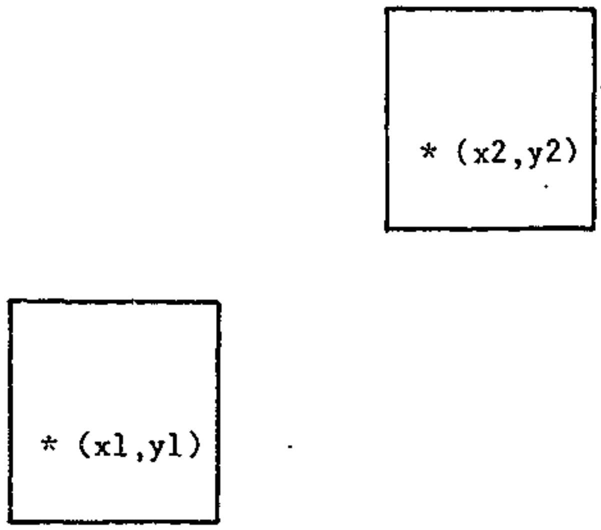
$$\int_0^L \left\{ \frac{1}{L} \left[\frac{L^2}{2} - uL + u^2 \right] \frac{du}{L} \right\}$$

$$= \frac{1}{L^2} \left[\frac{L^2 u}{2} - \frac{u^2 L}{2} + \frac{u^3}{3} \right]_0^L$$

$$= \frac{L}{3}$$

Similarly, it can be shown that the average vertical distance between the two points is equal to one third of the height of the cell.

CASE 2: Average distance between two points located in different cells.



Given two rectangular cells as shown in the above diagram. The horizontal and vertical axes are x and y respectively, and the co-ordinates of two randomly located points, one in each of the cells, are (x_1,y_1) and (x_2,y_2). If there are n randomly located pairs of points, then the average horizontal distance between two points located in different cells is equal to:

$$(\mathrm{Sum}\ (x_2-x_1))/n$$

$$= (\mathrm{Sum}\ x_2)/n - (\mathrm{Sum}\ x_1)/n$$

= horizontal distance between the two cell centroids.

Similarly, the vertical distance between the two cells is the vertical distance between the two cell centroids.

However, if the two cells are aligned vertically, then as shown in case 1, the average horizontal distance between two points is not zero but equal to one third the width of a cell. Similarly if the two cells are aligned horizontally, the average vertical distance is equal to one third the height of a cell.

APPENDIX 1B

THE RECTANGULAR DISTANCE BETWEEN TWO POINTS

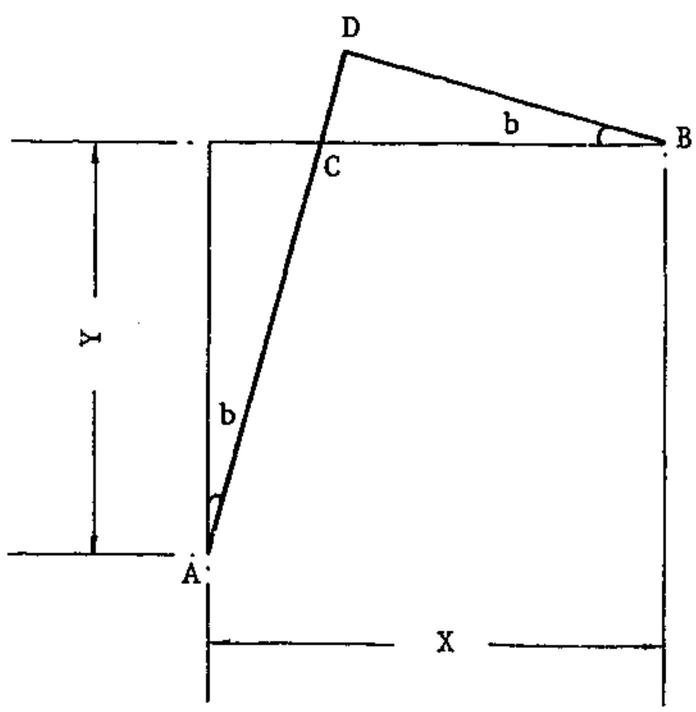
APPENDIX 1B THE RECTANGULAR DISTANCE BETWEEN TWO POINTS

In Appendix 1A it is assumed that walking paths are located along a road network oriented in the north-south and east-west directions, hence the rectangular distance between two points is equal to the sum of the horizontal separation (X) and vertical separation (Y) between the two points. However, if the road network is rotated in the clockwise direction by an angle b, then as shown in the Figure below, the rectangular distance between points A and B is equal to:

$$AC + CD + DB$$

$$= Y/\cos b + (X - Y \tan b)\sin b + (X - Y \tan b)\cos b$$

$$= Y/\cos b + (X - Y \tan b)(\sin b + \cos b)$$



APPENDIX 2

ROUTE SCHEDULES

APPENDIX 2 ROUTE SCHEDULES

ROUTE A1

Stop	11	12	13	14	15
	7: 0	7:10	7:20	7:30	7:50
	7:15	7:25	7:35	7:45	8: 5
	7:30	7:40	7:50	8: 0	8:20
	7:45	7:55	8: 5	8:15	8:35
	8: 0	8:10	8:20	8:30	8:50
	8:12	8:22	8:32	8:42	9: 2
	8:24	8:34	8:44	8:54	9:14
	8:36	8:46	8:56	9: 6	9:26
	8:48	8:58	9: 8	9:18	9:38
	9: 0	9:10	9:20	9:30	9:50
	9:15	9:25	9:35	9:45	10: 5
	9:30	9:40	9:50	10: 0	10:20
	9:45	9:55	10: 5	10:15	10:35
	10: 0	10:10	10:20	10:30	10:50
	10:30	10:40	10:50	11: 0	11:20
	11: 0	11:10	11:20	11:30	11:50
	11:30	11:40	11:50	12: 0	12:20
	12: 0	12:10	12:20	12:30	12:50
	12:20	12:30	12:40	12:50	13:10
Depart	12:40	12:50	13: 0	13:10	13:30
	13: 0	13:10	13:20	13:30	13:50
times	13:20	13:30	13:40	13:50	14:10
	13:40	13:50	14: 0	14:10	14:30
	14: 0	14:10	14:20	14:30	14:50
	14:30	14:40	14:50	15: 0	15:20
	15: 0	15:10	15:20	15:30	15:50
	15:30	15:40	15:50	16: 0	16:20
	16: 0	16:10	16:20	16:30	16:50
	16:15	16:25	16:35	16:45	17: 5
	16:30	16:40	16:50	17: 0	17:20
	16:45	16:55	17: 5	17:15	17:35
	17: 0	17:10	17:20	17:30	17:50
	17:12	17:22	17:32	17:42	18: 2
	17:24	17:34	17:44	17:54	18:14
	17:36	17:46	17:56	18: 6	18:26
	17:48	17:58	18: 8	18:18	18:38
	18: 0	18:10	18:20	18:30	18:50
	18:30	18:40	18:50	19: 0	19:20
	19: 0	19:10	19:20	19:30	19:50
	19:30	19:40	19:50	20: 0	20:20
	20: 0	20:10	20:20	20:30	20:50

ROUTE B1

Stop	15	14	13	12	11
	7: 5	7:25	7:35	7:45	7:55
	7:20	7:40	7:50	8: 0	8:10
	7:35	7:55	8: 5	8:15	8:25
	7:50	8:10	8:20	8:30	8:40
	8: 2	8:22	8:32	8:42	8:52
	8:14	8:34	8:44	8:54	9: 4
	8:26	8:46	8:56	9: 6	9:16
	8:38	8:58	9: 8	9:18	9:28
	8:50	9:10	9:20	9:30	9:40
	9: 5	9:25	9:35	9:45	9:55
	9:20	9:40	9:50	10: 0	10:10
	9:35	9:55	10: 5	10:15	10:25
	9:50	10:10	10:20	10:30	10:40
	10:20	10:40	10:50	11: 0	11:10
	10:50	11:10	11:20	11:30	11:40
	11:20	11:40	11:50	12: 0	12:10
	11:50	12:10	12:20	12:30	12:40
	12:20	12:40	12:50	13: 0	13:10
Depart	12:40	13: 0	13:10	13:20	13:30
	13: 0	13:20	13:30	13:40	13:50
times	13:20	13:40	13:50	14: 0	14:10
	13:40	14: 0	14:10	14:20	14:30
	14: 0	14:20	14:30	14:40	14:50
	14:30	14:50	15: 0	15:10	15:20
	15: 0	15:20	15:30	15:40	15:50
	15:30	15:50	16: 0	16:10	16:20
	16: 0	16:20	16:30	16:40	16:50
	16:15	16:35	16:45	16:55	17: 5
	16:30	16:50	17: 0	17:10	17:20
	16:45	17: 5	17:15	17:25	17:35
	17: 0	17:20	17:30	17:40	17:50
	17:12	17:32	17:42	17:52	18: 2
	17:24	17:44	17:54	18: 4	18:14
	17:36	17:56	18: 6	18:16	18:26
	17:48	18: 8	18:18	18:28	18:38
	18: 0	18:20	18:30	18:40	18:50
	18:30	18:50	19: 0	19:10	19:20
	19: 0	19:20	19:30	19:40	19:50
	19:30	19:50	20: 0	20:10	20:20
	20: 0	20:20	20:30	20:40	20:50

ROUTE A2

Stop	11	12	13	17	19
	7: 5	7:15	7:25	7:35	7:45
	7:25	7:35	7:45	7:55	8: 5
	7:45	7:55	8: 5	8:15	8:25
	8: 5	8:15	8:25	8:35	8:45
	8:25	8:35	8:45	8:55	9: 5
	8:45	8:55	9: 5	9:15	9:25
	9: 5	9:15	9:25	9:35	9:45
	9:25	9:35	9:45	9:55	10: 5
	9:45	9:55	10: 5	10:15	10:25
	10: 5	10:15	10:25	10:35	10:45
	10:35	10:45	10:55	11: 5	11:15
	11: 5	11:15	11:25	11:35	11:45
	11:35	11:45	11:55	12: 5	12:15
	12: 5	12:15	12:25	12:35	12:45
	12:35	12:45	12:55	13: 5	13:15
Depart	13: 5	13:15	13:25	13:35	13:45
	13:35	13:45	13:55	14: 5	14:15
times	14: 5	14:15	14:25	14:35	14:45
	14:35	14:45	14:55	15: 5	15:15
	15: 5	15:15	15:25	15:35	15:45
	15:35	15:45	15:55	16: 5	16:15
	16: 5	16:15	16:25	16:35	16:45
	16:25	16:35	16:45	16:55	17: 5
	16:45	16:55	17: 5	17:15	17:25
	17: 5	17:15	17:25	17:35	17:45
	17:25	17:35	17:45	17:55	18: 5
	17:45	17:55	18: 5	18:15	18:25
	18: 5	18:15	18:25	18:35	18:45
	18:35	18:45	18:55	19: 5	19:15
	19: 5	19:15	19:25	19:35	19:45
	19:35	19:45	19:55	20: 5	20:15
	20: 5	20:15	20:25	20:35	20:45

ROUTE B2

Stop	19	17	13	12	11
	7: 5	7:15	7:25	7:35	7:45
	7:25	7:35	7:45	7:55	8: 5
	7:45	7:55	8: 5	8:15	8:25
	8: 5	8:15	8:25	8:35	8:45
	8:25	8:35	8:45	8:55	9: 5
	8:45	8:55	9: 5	9:15	9:25
	9: 5	9:15	9:25	9:35	9:45
	9:25	9:35	9:45	9:55	10: 5
	9:45	9:55	10: 5	10:15	10:25
	10: 5	10:15	10:25	10:35	10:45
	10:35	10:45	10:55	11: 5	11:15
	11: 5	11:15	11:25	11:35	11:45
	11:35	11:45	11:55	12: 5	12:15
	12: 5	12:15	12:25	12:35	12:45
	12:35	12:45	12:55	13: 5	13:15
Depart	13: 5	13:15	13:25	13:35	13:45
	13:35	13:45	13:55	14: 5	14:15
times	14: 5	14:15	14:25	14:35	14:45
	14:35	14:45	14:55	15: 5	15:15
	15: 5	15:15	15:25	15:35	15:45
	15:35	15:45	15:55	16: 5	16:15
	16: 5	16:15	16:25	16:35	16:45
	16:25	16:35	16:45	16:55	17: 5
	16:45	16:55	17: 5	17:[illegible]	17:25
	17: 5	17:15	17:25	17:35	17:45
	17:25	17:35	17:45	17:55	18: 5
	17:45	17:55	18: 5	18:15	18:25
	18: 5	18:15	18:25	18:35	18:45
	18:35	18:45	18:55	19: 5	19:15
	19: 5	19:15	19:25	19:35	19:45
	19:35	19:45	19:55	20: 5	20:15
	20: 5	20:15	20:25	20:35	20:45

ROUTE A3

Stop	11	12	16	18
	7: 2	7:12	7:22	7:32
	7:27	7:37	7:47	7:57
	7:52	8: 2	8:12	8:22
	8:17	8:27	8:37	8:47
	8:42	8:52	9: 2	9:12
	9: 7	9:17	9:27	9:37
	9:32	9:42	9:52	10: 2
	9:57	10: 7	10:17	10:27
	10:32	10:42	10:52	11: 2
	11: 7	11:17	11:27	11:37
	11:42	11:52	12: 2	12:12
	12:17	12:27	12:37	12:47
Depart	12:52	13: 2	13:12	13:22
	13:27	13:37	13:47	13:57
times	14: 2	14:12	14:22	14:32
	14:37	14:47	14:57	15: 7
	15:12	15:22	15:32	15:42
	15:47	15:57	16: 7	16:17
	[illegible]	16:32	16:42	16:52
	16:47	16:57	17: 7	17:17
	17:12	17:22	17:32	17:42
	17:37	17:47	17:57	18: 7
	18: 2	18:12	18:22	18:32
	18:37	18:47	18:57	19: 7
	19:12	19:22	19:32	19:42
	19:47	19:57	20: 7	20:17

ROUTE B3

Stop	18	16	12	11
	7: 2	7:12	7:22	7:32
	7:27	7:37	7:47	7:57
	7:52	8: 2	8:12	8:22
	8:17	8:27	8:37	8:47
	8:42	[illegible]	9: 2	9:12
	[illegible]	[illegible]	9:27	9:37
	9:32	[illegible]	[illegible]	10: 2
	9:57	[illegible]	10:17	10:27
	10:32	[illegible]	10:52	11: 2
	11: 7	[illegible]	11:27	11:37
	11:42	11:52	12: 2	12:12
	12:17	12:27	12:37	12:47
Depart	12:52	13: 2	13:12	13:22
	13:27	13:37	13:47	13:57
times	14: 2	14:12	14:22	14:32
	14:37	14:47	14:57	15: 7
	15:12	15:22	15:32	15:42
	15:47	15:57	16: 7	16:17
	16:22	16:32	16:42	16:52
	16:47	16:57	17: 7	17:17
	17:12	17:22	17:32	17:42
	17:37	17:47	17:57	18: 7
	18: 2	18:12	18:22	18:32
	18:37	18:47	18:57	19: 7
	19:12	19:22	19:32	19:42
	19:47	19:57	20: 7	20:17

ROUTE A4

Stop	21	20	17	14
	7: 4	7:14	7:24	7:39
	7:19	7:29	7:39	7:54
	7:34	7:44	7:54	8: 9
	7:49	7:59	8: 9	8:24
	8: 4	8:14	8:24	8:39
	8:19	8:29	8:39	8:54
	8:34	8:44	8:54	9: 9
	8:49	8:59	9: 9	9:24
	9: 4	9:14	9:24	9:39
	9:19	9:29	9:39	9:54
	9:34	9:44	9:54	10: 9
	9:49	9:59	10: 9	10:24
	10:29	10:39	10:49	11: 4
	10:49	10:59	11: 9	11:24
	11:29	11:39	11:49	12: 4
	11:49	11:59	12: 9	12:24
	12: 9	12:19	12:29	12:44
	12:29	12:39	12:49	13: 4
Depart	12:49	12:59	13: 9	13:24
	13: 9	13:19	13:29	13:44
times	13:29	13:39	13:49	14: 4
	13:49	13:59	14: 9	14:24
	14: 9	14:19	14:29	14:44
	14:39	14:49	14:59	15:14
	15: 9	15:19	15:29	15:44
	15:39	15:49	15:59	16:14
	16: 9	16:19	16:29	16:44
	16:24	16:34	16:44	16:59
	16:39	16:49	16:59	17:14
	16:54	17: 4	17:14	17:29
	17: 9	17:19	17:29	17:44
	17:24	17:34	17:44	17:59
	17:39	17:49	17:59	18:14
	17:54	18: 4	18:14	18:29
	18:24	18:34	18:44	18:59
	18:54	19: 4	19:14	19:29
	19:24	19:34	19:44	19:59
	19:54	20: 4	20:14	20:29

ROUTE B4

Stop	14	17	20	21
	7: 5	7:20	7:30	7:40
	7:20	7:35	7:45	7:55
	7:35	7:50	8: 0	8:10
	7:50	8: 5	8:15	8:25
	8: 5	8:20	8:30	8:40
	8:20	8:35	8:45	8:55
	8:35	8:50	9: 0	9:10
	8:50	9: 5	9:15	9:25
	9: 5	9:20	9:30	9:40
	9:20	9:35	9:45	9:55
	9:35	9:50	10: 0	10:10
	9:50	10: 5	10:15	10:25
	10:20	10:35	10:45	10:55
	10:50	11: 5	11:15	11:25
	11:20	11:35	11:45	11:55
	11:50	12: 5	12:15	12:25
	12:20	12:35	12:45	12:55
Depart	12:40	12:55	13: 5	13:15
	13: 0	13:15	13:25	13:35
times	13:20	13:35	13:45	13:55
	13:40	13:55	14: 5	14:15
	14: 0	14:15	14:25	14:35
	14:30	14:45	14:55	15: 5
	5: 0	15:15	15:25	15:35
	15:30	15:45	15:55	16: 5
	16: 0	16:15	16:25	16:35
	16:15	16:30	16:40	16:50
	16:30	16:45	16:55	17: 5
	16:45	17: 0	17:10	17:20
	17: 0	17:15	17:25	17:35
	17:12	17:27	17:37	17:47
	17:24	17:39	17:49	17:59
	17:36	17:51	18: 1	18:11
	17:48	18: 3	18:13	18:23
	18: 0	18:15	18:25	18:35
	18:30	18:45	18:55	19: 5
	19: 0	19:15	19:25	19:35
	19:30	19:45	19:55	20: 5
	20: 0	20:15	20:25	20:35

ROUTE A5

Stop	22	20	16	11
	7: 4	7:14	7:24	7:39
	7:19	7:29	7:39	7:54
	7:34	7:44	7:54	8: 9
	7:49	7:59	8: 9	8:24
	8: 4	8:14	8:24	8:39
	8:19	8:29	8:39	8:54
	8:34	8:44	8:54	9: 9
	8:49	8:59	9: 9	9:24
	9: 4	9:14	9:24	9:39
	9:19	9:29	9:39	9:54
	9:34	9:44	9:54	10: 9
	9:49	9:59	10: 9	10:24
	10:29	10:39	10:49	11: 4
	10:49	10:59	11: 9	11:24
	11:29	11:39	11:49	12: 4
	11:49	11:59	12: 9	12:24
	12: 9	12:19	12:29	12:44
	12:29	12:39	12:49	13: 4
Depart	12:49	12:59	13: 9	13:24
	13: 9	13:19	13:29	13:44
times	13:29	13:39	13:49	14: 4
	13:49	13:59	14: 9	14:24
	14: 9	14:19	14:29	14:44
	14:39	14:49	14:59	15:14
	15: 9	15:19	15:29	15:44
	15:39	15:49	15:59	16:14
	16: 9	16:19	16:29	16:44
	16:24	16:34	16:44	16:59
	16:39	16:49	16:59	17:14
	16:54	17: 4	17:14	17:29
	17: 9	17:19	17:29	17:44
	17:24	17:34	17:44	17:59
	17:39	17:49	17:59	18:14
	17:54	18: 4	18:14	18:29
	18:24	18:34	18:44	18:59
	18:54	19: 4	19:14	19:29
	19:24	19:34	19:44	19:59
	19:54	20: 4	20:14	20:29

ROUTE B5

Stop	11	16	20	22
	7: 5	7:20	7:30	7:40
	7:20	7:35	7:45	7:55
	7:35	7:50	8: 0	8:10
	7:50	8: 5	8:15	8:25
	8: 5	8:20	8:30	8:40
	8:20	8:35	8:45	8:55
	8:35	8:50	9: 0	9:10
	8:50	9: 5	9:15	9:25
	9: 5	9:20	9:30	9:40
	9:20	9:35	9:45	9:55
	9:35	9:50	10: 0	10:10
	9:50	10: 5	10:15	10:25
	10:20	10:35	10:45	10:55
	10:50	11: 5	11:15	11:25
	11:20	11:35	11:45	11:55
	11:50	12: 5	12:15	12:25
	12:20	12:35	12:45	12:55
Depart	12:40	12:55	13: 5	13:15
	13: 0	13:15	13:25	13:35
times	13:20	13:35	13:45	13:55
	13:40	13:55	14: 5	14:15
	14: 0	14:15	14:25	14:35
	14:30	14:45	14:55	15: 5
	15: 0	15:15	15:25	15:35
	15:30	15:45	15:55	16: 5
	16: 0	16:15	16:25	16:35
	16:15	16:30	16:40	16:50
	16:30	16:45	16:55	17: 5
	16:45	17: 0	17:10	17:20
	17: 0	17:15	17:25	17:35
	17:12	17:27	17:37	17:47
	17:24	17:39	17:49	17:59
	17:36	17:51	18: 1	18:11
	17:48	18: 3	18:13	18:23
	18: 0	18:15	18:25	18:35
	18:30	18:45	18:55	19: 5
	19: 0	19:15	19:25	19:35
	19:30	19:45	19:55	20: 5
	20: 0	20:15	20:25	20:35

APPENDIX 3

DESCRIPTION OF THE OPTIMAL PATH ALGORITHM

APPENDIX 3 DESCRIPTION OF THE OPTIMAL PATH ALGORITHM

1 Start

2 Input from keyboard the network identification number N9

3 Read the following files

node identification file

network file

reversed network file

4 Input from keyboard the following information:

origin node number	S
destination node number	DD
clock start time from origin	STO
weighting for walk time	WALK
wait time	WAIT
route change penalty	RUMIN

5 Find quickest paths from origin node to all other nodes by quickest path algorithm (see Section 3.3.3 in main text).

6 If destination node cannot be reached print error message and terminate program. Else proceed to next step.

7 Write the clock arrival times at all nodes in array ARTIME(I)

8 Output on the terminal the quickest path from origin to destination and calculate the weighted trip time (WEITIM).

9 Calculate the "weighted finish time" at the destination:

ATF = STO + WEITIM.

10 Execute the quickest path algorithm with a reversed network (see Section 2.11 in main text) and find the quickest paths from all nodes to the destination node with a specified "weighted finish time" = ATF.

11 If a path cannot be found from origin to destination print error message and terminate program. Else proceed to next step.

12 Write the clock departure time for all nodes in array DPTIME(I).

13 Determine node accessibility. (See Section 3.4.3 in main text).

14 For any link, if either its A-NODE and/or B-NODE is inaccessible, remove it from the network.

Comment: Generate paths by fanning out from the origin node. Paths generated are stored in the following set of records:

GN = record set number for the A-NODE of the link

II = record set number for the B-NODE of the link

MT(GN,1) = A-NODE node number

MT(GN,2) = arrive time at the A-NODE

MT(GN,3) = weighted time of trip from origin to A-NODE

B$(GN) = route used for arrival at A-NODE

PTR(GN) = record pointer

PTL(GN) = record label

15 The path generation process start from the origin node by filling the first record set:

GN = 1

II = 1

MT(GN,1) = S

MT(GN,2) = STO

MT(GN,3) = -RUMIN (see note 1)

Comment: The path generation process continues by iterating from steps 16 to 29. Each iteration finds all feasible extensions of a path by one link length. Each feasible extension is recorded by filling a new set of records.

16 Label the record set: PTL(GN) = 1

17 If MT(GN,1) = DD go to 29. Else proceed to next step.

18 Check whether the A-NODE has been reached previously by searching previous record sets. If none can be found go to 20. Else proceed to next step.

19 If the arrival time at the A-NODE is found to be equal to and the weighted trip time is found to be greater than that of a previous record set, go to 29. (See Section 3.4.5 (a) in main text.). Else go to 18 and continue the search.

20 Try to extend the path by finding a link starting from the A-NODE. If no more link can be found go to 29. Else proceed to next step.

21 Calculate the arrival time ATN and weighted trip time WTN at the B-NODE Z.

22 If ATN is larger than DPTIME(Z), go to 28. (See Section 3.4.5 (b) in main text.) Else proceed to next step.

23 Calculate the quickest possible trip time from node Z to node DD (see note 2): QAA = ARTIME(DD) - ARTIME(Z)

24 Calculate the maximum trip time available for travelling from node Z to node DD: QBB = ATF - DPTIME(Z)

25 If (ATF - STO - WTN) is less than QAA or QBB go to 28. (See note 3 and Section 3.4.5 (c) in main text.)

26 If path is not efficient go to 28. (See Section 3.4.4 in main text.)

27 Store the path extension in a new record set:

II = II + 1

MT(II,1) = B-NODE number

MT(II,2) = arrival time

MT(II,3) = weighted trip time

B$(II) = route used to arrive at B-NODE

PTR(II) = GN

28 Is there another link starting from the same A-NODE? If yes go to 21. Else proceed to next step.

29 Search for another unlabelled record set, if one can be found, put GN = record number and go to 16. Else proceed to next step.

30 Now that all record sets are labelled, proceed to backtrack the optimal path: search all record sets to find those with MT(I,1) = DD, from these record sets select the one with the smallest MT(I,3). Backtrack the optimal path from this record.

31 End

Note 1.

The initial weighted trip time is set to be -RUMIN so that no route change penalty is recorded for the first transit route used.

Note 2.

If node Z is on the quickest path from S to DD, then QAA is equal to the quickest trip time from Z to DD. Else QAA is less than the quickest trip time from Z to DD.

Note 3.

The total weighted time available for the trip is (ATF - STO). When reaching node Z, an amount of weighted time equal to WTN has been used up, the remaining weighted time for the whole trip is (ATF - STO - WTN), if this time is less than QAA or QBB stop the path.

APPENDIX 4

GLOSSARY

APPENDIX 4 GLOSSARY

Algorithm	A systematic procedure for solving a mathematical problem.
Arc	An oneway link.
Basic network	A network which is generated directly from the timetable database.
Detailed network	A basic network with more nodes added at the regions surrounding the origin and destination so as to model more accurately the access and egress points of a trip using the network.
Earliest arrival time	The earliest arrival time at a node when starting from the origin with a specified starting time.
Essential node	A node representing a transit stop which is either (a) the first stop or last stop of a transit route, (b) the first stop or last stop of a path under consideration, or (c) a potential transfer point to other routes.
Headway	The time interval between successive arrivals of vehicles at a stop or point in a transportation system.
Inaccessible node	A node which has no chance of being included optimal path.
Latest departure time	The latest departure time at a node so as to arrive at the destination not later than the weighted finish time.

Link	A line drawn on a graph used to represent part of a transportation system, such as a section of a road or a section of a bus route or railway route.
Major stop	A transit stop represented as a node in a basic network.
Minor stop	A transit stop not represented as a node in a basic network.
Network	Graphical representation of a transport system by means of links and nodes.
Node	One end of a link.
Non-essential node	A node which is not an essential node.
Optimal path	A path with minimum weighted trip time.
Path	An ordered sequence of links.
Profile	The space-time trajectory of a vehicle when it is moving along a path.
Quickest path	A path from origin to destination with minimum trip time.
Reversed network	A network with reversed timetables and directions of arcs reversed.
Reversed timetable	The time schedules of a reversed timetable is equal to 24 hours minus the time schedules in the original timetable.
Station	A tram stop, bus stop or railway station.
Timetable	A table showing the scheduled departure time or arrival time of transit vehicles at various stops along a route.

Transit route	A designated path for the operation of a transit vehicle.
Tree	Part of a network representing the paths from one origin node to many other destination nodes.
Sketch network	A network in which all nodes represent essential stops within a transit system.
Weighted finish time	Starting time of trip plus weighted trip time.

REFERENCES

REFERENCES

Abkowitz, M.D. (1981). "An Analysis of the Commuter Departure Time Decision." Transportation 10: 283-297.

Bartram, D.J. (1980). "Comprehending Spatial Information: The Relative Efficiency of Different Mehods of Representing Information about Bus Routes." J. of Applied Psychology, 65:103-110.

Beckmann, M., McGuire, C.B. and Winsten, C.B. (1956). "Studies in the Economics of Transportation." Yale University Press, New Haven, Connecticut.

Bellman, R.E. (1958). "On a Routing Problem." Quart. Appl. Math. 16:87-90.

Bellman, R.E. and Kalaba, R. (1960). "On kth Best Policies." J. SIAM 8:582-588.

Ben-Akiva, M. and Lerman, S. (1978). "Disaggregate Travel and Mobility Choice Models and Measures of Accessibility." Proc. of Third International Conference on Behavioural Travel Modelling, Tarnunda, South Australia. Croom Helm, London.

Bonsall, P. (1976). "Tree-Building with Complex Cost Structures - A New Algorithm for Incorporation into Transport Demand Models." Transportation 5:309-329

Bowyer, D.P. (1983). Case Study of the Bus Service Design System (BUSDES). Australian Road Res. Board Internal Report AIR 351-2.

Brog, W. (1982). "The Situational Approach - An Alternative Model Concept - Theorectical Foundations and Practical Applications.". Forum Papers, 7th Australian Transport Res. Forum, 2:547-592.

Bruzelius, N. (1979). "The Value of Travel Time." Croom Helm Ltd., London.

Burns, L.D. (1979). Transportation, Temporal and Spatial Components of Accessibility. Mass.: Lexington Books, D.C. Health.

Burrell, J.E. (1968). "Multiple Route Assignment and Its Application to Capacity Restraint." Proc. of 4th International Symposium on the Theory of Traffic Flow, Karlsruhe, Germany.

------ (1976). "Multipath Route Assignment: A Comparison of Two Methods." Traffic Equilibrium Methods, Lecture Notes in Economics and Mathematical Systems, Volume 118, Springer-Verlag, New York.

Chua, T.A. (1984). "The Planning of Urban Bus Routes and Frequencies: a Survey." Transportation 12: 147-172.

Clercq, F. le (1972). "A Public Transport Assignment Method." Traff. Engng. Control 13:91-96.

Cooke, K.L. and Halsey, E. (1966). "The Shortest Route through a Network with Time-Dependent Internodal Transit Times." J. Math. Anal and Appl. 14:493-498.

Dafermos, S.C. (1971) "An Extended Traffic Assignment Model with Applications to Two-way Traffic." Transpn. Sci. 5(4):366-389.

Daganzo, C.F. and Sheffi, Y. (1977). "On Stochastic Models of Traffic Assignment." Transpn. Sci. 11(3):253-274.

Dallal, E.A. (1980). "Public Transport Accessibility Measurement." Traff. Engng. Control 21: 494-495.

Dalvi, M.Q. and Daly, A.J. (1976). "The Valuation of Travelling Time: Theory and Estimation." Report T72, Local Government Operational Res. Unit, Reading, England.

Dalvi, M.Q. and Martin K.M. (1976). "The Measurement of Accessibility: some Preliminary Results." Transportation 5(1): 17-42.

Daly, A.J. and Zachary S. (1978). "Improved Multiple Choice Models." in Henser, D.A. and Delvi, M.Q. (Eds.) Determinants of Travel Choice, Saxon House, Sussex.

Danas, A. (1980) "Arrivals of Passengers and Buses at Two London Bus-Stops." Traff. Engng. Control 21: 472-475.

Dantzig, G.B. (1960). "On the Shortest Route through a Network." Management Sci. 6:187-190.

DeSerpa, A.C. (1971). "A Theory of the Economics of Time." Economic Journal 81:828-846.

Dial, R.B. (1967). "Transit Pathfinder Algorithm." Highway Res. Record No. 205: 67-85.

------- (1971). "A Probabilistic Multipath Assignment Model which Obviates Path Enumeration." Transpn. Res. 5(2): 83-111

Diewald, W.J., Frost, W.H. and Bamberg, W. (1983). Assessment of Transit Passenger Information Systems. UMTA Report NO. IT-06-0248-83-1, U.S. Dept. of Transpn.

Dijkstra, E.W. (1959). "A Note on Two Problems in Connection with Graphs." Numerische Mathematik 1:269-271.

Dobson, R. (1975). "Towards the Analysis of Attitudinal and Behavioural Responses to Transportation System Characteristics." Transportation 4: 267-290.

Dreyfus, S.E. (1968). "An Appraisal of Some Shortest Path Algorithms." Operations Res., 17(3):395-412.

Ellson, P.B. and Tebb, R.G.P. (1981). "Costs and Benefits of a Bus-Service Information Leaflet." Transport and Road Res. Lab. Report LR 825, Crowthorne, England.

------- (1981). "Benefits, Costs and Effects of Providing Additional Information about Urban Public Transport Services." Transport and Road Res. Lab. Report LR 991, Crowthorne, England.

Floyd, R.W. (1962). "Algorithm 97, Shortest Path." Comm. ACM 5:345.

Forsyth, E. (1985). "Real-Time Information Systems for Public Transport Passengers." 19th Universities' Transport Studies Group Annal Conference, University of Birmingham.

Goodwin, P.B. (1974). "Generalised Time and the Problem of Equity in Transport Studies." Transportation 3:1-24.

Gorstein, M. and Tilles, R. (1981). "Automating the Delivery of Ground Transportation Information." Transpn. Res. Board Annual Meeting, Washington DC, U.S.A.

Gray,A. (1978). "The Generalised Cost Dilemma." Transportation 7: 261-280.

Greater London Council. (1966). London Traffic Survey.

Griffen, M. (1985). "An Automatic Route-Planning and Mapping System."

Hall, R.M. (1983). "Traveller Route Choice: Travel Time Implications of Improved Information and Adaptive Decisions." Transpn. Res. 17A(3): 201-214.

Hans G. von Cube (1958). "Assignment of Passengers to Transit Systems." Traff. Engng. 28.

Hansen, W.G. (1959). "How Accessibility Shapes Land Use." J. of American Inst. Planners 25: 73-76.

Hayashi,I., Itoh,K., Suzuki,S. (1982). "Man-Machine Interactive Guidance for Urban Railway Networks." Computing and Graphics 7:59-72.

Hewing, R.B. and Hoffman, M.L.H. (1969). "An Explanation of the Transitnet Algorithm." Res. Memorandum No 198, G.L.C. Department of Highways and Transportation, England.

Hoffman W. and Ravley R. (1959). "A Method for the Solution of the Nth Best Path Problem." J. ACM 6:506-514.

Horowitz, A.J. (1981). "Subjective Value of Time in Bus Transit Travel." Transportation 10:149-164.

Huber, M.J., Boutwell H.B., and Witheford D.K. (1968). "Comparative Analysis of Traffic Assignment Techniques with Actual Highway Use." National Cooperative Highway Res. Program Report 58, Highway Res. Board.

Ingram, D.R. (1971). "The Concept of Accessibility: a Search for an Operation Form." Regional Studies 5: 101-107.

James, N. (1985). "Bridging the Information Gap." 19th Universities' Transport Studies Group Annal Conference, University of Birmingham.

Kirby, R.F., Green, M.A., and Olsson, M.L. (1979). An Assessment of Short-Range Transit Planning in Selected U.S. Cities. The Urban Institute. Washington, D.C.

Lai, F.K. and Van Vliet D. (1979). "Trip Assignment Techniques Current in the U.K." Traff. Engng. Control 20 (7): 348-351.

Lamb, G.M. (1970). "Introduction to Transportation Planning: 5.Assignment and Restraint." Traff. Engng. Control 11 (5):32-37.

Last, A. and Leak, S.E. (1976). "Transept: A Bus Model." Traff. Engng. Control 17 (1): 14-17

Leake, G.R. and Huzayyin, A.S. (1980). "Importance of Accessibility Measures in Trip Production Models." Transpn. Planning and Technology 6: 9-20.

LeBlanc, L., Morlok, E.K. and Pierskalla, W.P.(1975). "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem." Transpn. Res. 9:309-318.

Lenntorp, B. (1976). "Paths in Space-Time Environments: a Time-Geographic Study of Movement Possibilities of Individuals." Lund Studies in Geography B44. (Gleerup, Lund.)

Leonard, D.R., Tough, J.B. and Baguley, P.C. (1978). "CONTRAM: A Traffic Assignment Model for Predicting Flows and Queues During Peak Periods." Transport and Road Res. Lab. LR 841 Crowthorne, England.

Martin, B.V., Memmott, F.W. and Bone, A.J. (1963) Principles and Techniques of Predicting Future Demand for Urban Area Transportation. MIT and Mass. Dept. of Public Works.

Martin, B.V. and Manheim, M.L. (1965) " A Research Program for Comparison of Traffic Assignment Techniques." Highway Res. Rec. 88, 69-84.

McIntosh, P.T. and Quarmby, D.A. (1970). "Generalised Cost and the Estimation of Movement Costs and Benefits in Transportation Planning." Department of Environment MAU Note 179, England.

Moore, E.F. (1957). "The Shortest Path Through a Maze." Proc. of International Symposium on the Theory of Switching. pp. 22-25. Harvard University.

Morris,J.M., Dumple, P.L., Wigan, M.R. (1979). "Accessibility Indicators for Transport Planning." Transp. Res. 13A: 91-109.

Moskowitz, K. (1956). "California Method of Assigning Diverted Traffic to Proposed Freeways." Highway Res. Board Bull. 130, 1-26.

Newell, G.F. (1980). Traffic Flow on Transportation Networks. MIT Press.

Nguyen, S. (1974). "An Algorithm for the Traffic Assignment Problem." Transpn. Sci. 8(3):203-216.

Pickett, M.W. (1978). "Some Benefits of an Integrated Public Transport Travel Information System." Transport and Road Res. Lab. Report LR 830, Crowthorne, England.

------- (1980). "The Generation of Integrated Public Transport Travel Information by Computer." Transport and Road Res. Lab. Report SR 630, Crowthorne, England.

------- (1981). "The Production, Dissemination and Costs of an Integrated Public Transport Travel Information System." Transport and Road Res. Lab. Report SR 657, Crowthorne, England.

------- (1982). "Trials of Computer-Generated Public Transport Travel Information in Wiltshire." Transport and Road Res. Lab. Report LR 1036, Crowthorne, England.

Pike, D.H., Fuller, P.I., White, M.T. (1976). "Telford Transportation Study, 1. An Aid to the Preliminary Evaluation of Long-Term Public Transport Options." Traff. Engng. Control 17: 52-54.

Pirie, G.H. (1979). "Measuring Accessibility: a Review and Proposal." Environment and Planning. 11A: 299-312.

Potts and Oliver (1972). "Flows on Transportation Networks." Mathematics in Science and Engng. 90.

Rand Corporation, The (1955). A Million Random Digits with 100,000 Normal Deviates. The Free Press, Glencoe, Illinois.

Rapp, M.H. and Gehner, G.D. (1976). "Transfer Optimisation to Minimise Delays: Stage II of an Interactive Graphic system for Operational Transit Planning." Transpn. Res. Record 619: 27-33.

Richardson, A.J. (1980). "Advanced Demand Modelling for Advanced Transport Systems." J. of Advanced Transpn. 14 (2): 113-131.

Schmidt, J.W. and Knight, R. (1980) "The Status of Computer-Aided Scheduling in North America." in Wren, A. (Eds.) Computer Scheduling of Public Transport. Elsevier North-Holland Inc. New York.

Schneider, M. (1956). "Traffic Assignment by Mechanical Methods." Highway Res. Board Bull. 130.

Smock, R. (1962). "An Iteraive Approach to Capacity Restraint on Arterial Networks." Highway Res. Board Bull. 347, 60-66.

Starkie, D.N.M. (1971) "Model Split and the Value of Time." J. of Transport Economics and Policy 5: 216-220.

Thomson, J.M. (1977). Great Cities an Their Traffic. Victor Gollancz Ltd. London.

Tresidder, J.O., Meyers D.A., Burrell J.E. and Powell T.J. (1968). "The London Transportation Study: Methods and Techniques." Proc. Inst. Civ. Eng. 39: 433-464.

Van Vliet, D. (1976). "Road Assignment- I: Principles and Parameters of Model Formulation." Transpn. Res. 10:145-149.

------- (1982). "Saturn- A Modern Assignment Model." Traff. Engng. Control 23(12): 578-581.

Van Vliet, D. and Dow, P.D.C. (1979) "Capacity-restrained Road Assignment." Traff. Engng. Control 20 (6) 296-299.

Von Cube, H.G. (1958) "Assignment of Passengers to Transit Systems." Traff. Engng. 28:21.

Wachs, M. and Kumagai, T.G. (1973). "Physical Accessibility as a Social Indicator." Socio-Econ. Plan. Sci. 7: 437-456.

Wardrop, J.G. (1952) "Some Theorectical Aspects of Road Traffic Research." Proc. Inst. Civ. Eng., 1, Part II:325-378.

Wilbur Smith and Associates. (1964). Baltimore Metropolitan Area Transportation Study.

Williams, H.C.W.L. (1977). "On the Formation of Travel Demand Models and Economic Evaluation Measures of User Benefit." Environment and Planning 9A:285-344.

Williamson, R.H. and Miller, A.J. "A Personalised Public Transport Directory." Wootton Jeffreys & Partners, unpublished paper.

Wilson-Hill Associates, Inc. (1981). The Computerized Customer Information System at the Southern California Rapid Transit District, Draft Evaluation Report, U.S. Dept. of Transpn., Transportation Systems Centre, Cambridge, Massachusetts.

------- (1982). The Automated Information Directory System (AIDS) at the Washington Metropolitan Area Transit Authority, Draft Evaluation Report, U.S. Dept. of Transpn., Transpn Systems Centre, Cambridge, Massachusetts.

Wren, A.(1980). "General Review of the Use of Computers in Scheduling Buses and Their Crews." in Wren, A. (Eds.) Computer Scheduling of Public Transport. Elsevier North-Holland Inc. New York.

Yagar, S. (1971). "Dynamic Traffic Assignment by Individual Path Minimization and Queueing." Transpn. Res. 171-196.

Yago, G. (1984). The Decline of Transit- Urban Transportation in German and U.S. Cities 1900-1970. Cambridge University Press. Cambridge

Young, W. and Richardson, A.J. (1981). "Regional Freight Mode Choice: An Application of the Elimination-by-Aspects Model." Report to Australian Railway Res. and Development Organisation, Melbourne, Australia.

Young, W., Richardson, A.J., Ogden, K.W. and Rattray, A.L. (1982). "Rail-road Freight Mode Choice: The Application of an Elimination-by-aspects Model." Transportation Research Record 838:38-44.

A SCHEDULE-BASED TRANSIT NETWORK MODEL

Volume 2 - Program Manual

by

C. O. Tong, B.Sc., M.Sc.

A thesis presented in fulfilment of the requirements for the

the degree of Doctor of Philosophy

Monash University

Department of Civil Engineering

1986

CONTENTS

LIST OF FIGURES

LIST OF TABLES

PREFACE

This manual provides a user's guide to a package of programs developed for transit network modelling.

The main characteristics of the network model is that route timetables are included in the computer network description. The contents of the manual are summarized as follows:

SECTION 1. "TIMETABLE CODING", describes the coding of timetables.

SECTION 2. "CODING LOCATIONS OF NODES" describes the coding of map co-ordinates of nodes in the network.

SECTION 3. "CODING OF MINOR STOPS", describes the coding required for building a detail network which includes all transit stops.

SECTION 4. "NETWORK BUILDING", describes the building of various types of transit networks using the data base coded in Sections 1 to 3.

SECTION 5. "SOLVING MINIMUM PATH PROBLEMS", describes how various types of minimum path problems can be solved using the networks built in Section 4.

SECTION 6. "ZONAL NETWORK MODELLING", describes the building of networks with connections to traffic zones and the construction of zonal trip time matrices.

SECTION 7. "ASSIGNMENT MODELLING", describes two assignment sub-models for assigning trips to transit networks.

SECTION 8. "ACCESSIBILITY MODELLING", describes programs for calculating start time based and arrival time based accessibility for trips between zones.

APPENDIX A. "A SAMPLE SESSION", shows an example of how to use the programs in the package.

APPENDIX B. "OPERATION SEQUENCE OF PROGRAMS", shows in a tabular form the operation sequence of all the programs in the package.

The notational conventions used in this manual are as follows:

(a) Program names are included in square brackets. (For example, [Tblcode].)
(b) File names are in uppercase letters. (For example, TIMETBL1.)

For ease of understanding, it is suggested that the various Sections in this manual should be read in conjuction with the sample session contained in Appendix A.

This manual does not describe a specific package of programs for generating travel information for a particular transit system. The reason is that as discussed in Volume 1, Section 6.6 of this thesis, many different types of input/output devices can be used for a travel information system. There are also various ways in which travel desires can be input and the relevant travel information presented to the user. However, all the basic programs for generating travel information are described in Section 5 of this manual. If the required input/output format of the travel information system is specified, then a package can be constructed by selecting the appropriate programs and modifying them to suit the required input/output format.

SECTION 8. "ACCESSIBILITY MODELLING", describes programs for calculating start time based and arrival time based accessibility for trips between zones.

APPENDIX A. "A SAMPLE SESSION", shows an example of how to use the programs in the package.

APPENDIX B. "OPERATION SEQUENCE OF PROGRAMS", shows in a tabular form the operation sequence of all the programs in the package.

The notational conventions used in this manual are as follows:

(a) Program names are included in square brackets. (For example, [Tblcode].)
(b) File names are in uppercase letters. (For example, TIMETBL1.)

For ease of understanding, it is suggested that the various Sections in this manual should be read in conjuction with the sample session contained in Appendix A.

SECTION 1

TIMETABLE CODING

SECTION 1 TIMETABLE CODING

1.1 Introduction

In this section, you will be taught how to code up a set of transit route timetables into a computer file. The purpose is to establish a database of public transport information. This database can then be used as input to other programs included in this package for various purposes, such as finding minimum paths in transit networks.

1.2 Timetables

Obtain a set of timetables covering all transit routes and schedules which you wish to include into you database. These can include rail, bus, tram or any other types of transit services provided in your study area. Examples of a rail timetable and a tram timetable is shown in Figures 1.1 and 1.2 respectively.

For convenience in the management of your database, it is preferable to code different groups of transit routes into separate files. For example, rail, tram and bus routes should be coded into separate files.

1.3 Preparing a Network Diagram

A network diagram consists of lines (links) and points (nodes) representing the routes and stops respectively of a transit system. Example of a rail network diagram is shown in Figure 1.3. If a suitable network diagram is provided by the transit operator, then it is not necessary to draw one. It is necessary however, to assign numbers to the nodes in the

GLEN WAVERLEY LINE

FROM MELBOURNE – Monday to Friday (Current from 1/11/82)

PARLIAMENT Dep.	..	..	..	..	..	..	..	..	6 51
MUSEUM	..	..	..	..	..	..	..	..	6 53
FLAGSTAFF	..	..	..	..	..	..	..	..	•
SPENCER STREET	..	..	..	..	..	..	..	..	6 56
FLINDERS STREET	4 47	5 22	5 44	6 12	6 31	6 39	6 48	6 55	7 05
RICHMOND	4 50	5 25	5 48	6 15	6 34	6 42	6 51	6 58	7 08
EAST RICHMOND	4 52	5 27	5 50	6 17	6 36	6 44	6 53	7 01	7 10
BURNLEY	4 54	5 29	5 52	6 19	6 38	6 45	6 55	7 03	7 12
HEYINGTON	4 57	5 32	5 55	6 22	6 41	6 49	6 58	7 06	7 15
KOOYONG	4 59	5 34	5 57	6 24	6 43	6 51	7 00	7 08	7 17
TOORONGA	5 01	5 36	5 59	6 26	6 45	6 53	7 02	7 10	7 19
GARDINER	5 03	5 38	6 01	6 28	6 47	6 55	7 04	7 12	7 21
GLEN IRIS	5 05	5 40	6 03	6 30	6 49	6 57	7 06	7 14	7 23
DARLING	5 07	5 42	6 05	6 32	6 51	6 59	7 08	7 16	7 25
EAST MALVERN	5 09	5 44	6 07	6 34	6 53	7 01	7 10	7 18	7 27
HOLMESGLEN	5 12	5 47	6 10	6 37	6 56	7 04	7 13	7 21	7 30
JORDANVILLE	5 15	5 50	6 13	6 40	6 59	7 07	7 16	7 24	7 33
MOUNT WAVERLEY	5 17	5 52	6 15	6 42	7 01	7 09	7 18	7 26	7 35
SYNDAL	5 20	5 55	6 18	6 45	7 04	7 12	7 21	7 29	7 38
GLEN WAVERLEY Arr.	5 23	5 58	6 21	6 48	7 07	7 15	7 24	7 32	7 41

PARLIAMENT Dep.	7 08	..	7 43	7 56	8 07	8 23	8 35	9 00	..
MUSEUM	7 10	..	7 45	7 58	8 09	8 25	8 37	9 02	..
FLAGSTAFF	•	..	•	•	•	•	•	•	..
SPENCER STREET	7 13	..	7 48	8 01	8 12	8 28	8 40	9 05	..
FLINDERS STREET	7 17	7 30	7 53	8 07	8 18	8 32	8 45	9 10	9 25
RICHMOND	7 20	7 33	7 56	8 10	8 21	8 35	8 48	9 13	9 28
EAST RICHMOND	7 22	•	7 58	8 12	8 23	8 37	8 50	9 15	9 30
BURNLEY	7 24	7 36	8 00	8 14	8 25	8 39	8 52	9 17	9 32
HEYINGTON	7 27	7 39	8 03	8 17	8 28	8 42	8 55	9 20	9 35
KOOYONG	7 29	7 41	8 05	8 19	8 30	8 44	8 57	9 22	9 37
TOORONGA	7 31	7 43	8 07	8 21	8 32	8 46	8 59	9 24	9 39
GARDINER	7 33	7 45	8 09	8 23	8 34	8 48	9 01	9 26	9 41
GLEN IRIS	7 35	7 47	8 11	8 25	8 36	8 50	9 03	9 28	9 43
DARLING	7 37	7 49	8 13	8 27	8 38	8 52	9 05	9 30	9 45
EAST MALVERN	7 39	7 51	8 15	8 29	8 40	8 54	9 07	9 32	9 47
HOLMESGLEN	7 42	7 54	8 18	8 32	8 43	8 57	9 10	9 35	9 50
JORDANVILLE	7 45	7 57	8 21	8 35	8 46	9 00	9 13	9 38	9 53
MOUNT WAVERLEY	7 47	7 59	8 23	8 37	8 48	9 02	9 15	9 40	9 55
SYNDAL	7 50	8 02	8 26	8 40	8 51	9 05	9 18	9 43	9 58
GLEN WAVERLEY Arr.	7 53	8 05	8 29	8 43	8 54	9 08	9 21	9 46	10 01

PARLIAMENT Dep.	..	9 44	10 00	10 15	10 30	..	10 59	11 14	11 29
MUSEUM	..	9 46	10 02	10 17	10 32	..	11 01	11 16	11 31
FLAGSTAFF	..	•	•	•	•	..	•	•	•
SPENCER STREET	..	9 49	10 05	10 20	10 35	..	11 04	11 19	11 34
FLINDERS STREET	9 40	9 55	10 10	10 25	10 40	10 55	11 10	11 25	11 40
RICHMOND	9 43	9 58	10 13	10 28	10 43	10 58	11 13	11 28	11 43
EAST RICHMOND	9 45	10 00	10 15	10 30	10 45	11 00	11 15	11 30	11 45
BURNLEY	9 47	10 02	10 17	10 32	10 47	11 02	11 17	11 32	11 47
HEYINGTON	9 50	10 05	10 20	10 35	10 50	11 05	11 20	11 35	11 50
KOOYONG	9 52	10 07	10 22	10 37	10 52	11 07	11 22	11 37	11 52
TOORONGA	9 54	10 09	10 24	10 39	10 54	11 09	11 24	11 39	11 54
GARDINER	9 56	10 11	10 26	10 41	10 56	11 11	11 26	11 41	11 56
GLEN IRIS	9 58	10 13	10 28	10 43	10 58	11 13	11 28	11 43	11 58
DARLING	10 00	10 15	10 30	10 45	11 00	11 15	11 30	11 45	12 00
EAST MALVERN	10 02	10 17	10 32	10 47	11 02	11 17	11 32	11 47	12 02
HOLMESGLEN	10 05	10 20	10 35	10 50	11 05	11 20	11 35	11 50	12 05
JORDANVILLE	10 08	10 23	10 38	10 53	11 08	11 23	11 38	11 53	12 08
MOUNT WAVERLEY	10 10	10 25	10 40	10 55	11 10	11 25	11 40	11 55	12 10
SYNDAL	10 13	10 28	10 43	10 58	11 13	11 28	11 43	11 58	12 13
GLEN WAVERLEY Arr.	10 16	10 31	10 46	11 01	11 16	11 31	11 46	12 01	12 16

FLINDERS STREET Dep.	11 47	12 02	12 17	12 32	12 47	1 02	1 17	1 32	1 47
SPENCER STREET	11 50	12 05	12 20	12 35	12 50	1 05	1 20	1 35	1 50
FLAGSTAFF	•	•	•	•	•	•	•	•	•
MUSEUM	11 53	12 08	12 23	12 38	12 53	1 08	1 23	1 38	1 53
PARLIAMENT	11 55	12 10	12 25	12 40	12 55	1 10	1 25	1 40	1 55
RICHMOND	11 58	12 13	12 28	12 43	12 58	1 13	1 28	1 43	1 58
EAST RICHMOND	12 00	12 15	12 30	12 45	1 00	1 15	1 30	1 45	2 00
BURNLEY	12 02	12 17	12 32	12 47	1 02	1 17	1 32	1 47	2 02
HEYINGTON	12 05	12 20	12 35	12 50	1 05	1 20	1 35	1 50	2 05
KOOYONG	12 07	12 22	12 37	12 52	1 07	1 22	1 37	1 52	2 07
TOORONGA	12 09	12 24	12 39	12 54	1 09	1 24	1 39	1 54	2 09
GARDINER	12 11	12 26	[illegible]	12 56	1 11	1 26	1 41	1 56	2 11
GLEN IRIS	12 13	12 28	12 43	[illegible]	1 13	1 28	1 43	1 58	2 13
DARLING	12 15	12 30	12 45	1 00	1 15	1 30	1 45	2 00	2 15
EAST MALVERN	12 17	12 32	12 47	1 02	1 17	1 32	1 47	2 02	2 17
HOLMESGLEN	12 20	12 35	12 50	1 05	1 20	1 35	1 50	2 05	2 20
JORDANVILLE	12 23	12 38	12 53	1 08	1 23	1 38	1 53	2 08	2 23
MOUNT WAVERLEY	12 25	12 40	12 55	1 10	1 25	1 40	1 55	2 10	2 25
SYNDAL	12 28	12 43	12 58	1 13	1 28	1 43	1 58	2 13	2 28
GLEN WAVERLEY Arr.	12 31	12 46	1 01	1 16	1 31	1 46	2 01	2 16	2 31

FLINDERS STREET Dep.	2 02	2 17	2 32	2 46	3 01	3 13	3 25	3 45	3 54
SPENCER STREET	2 05	2 20	2 35	2 49	3 04	3 16	3 28	3 48	3 57
FLAGSTAFF	•	•	•	•	•	•	•	•	•
MUSEUM	2 08	2 23	2 38	2 52	3 07	3 19	3 31	3 51	4 00
PARLIAMENT	2 10	2 25	2 40	2 54	3 09	3 21	3 33	3 53	4 02
RICHMOND	2 13	2 28	2 43	2 57	3 12	3 24	3 36	3 56	4 05
EAST RICHMOND	2 15	2 30	2 45	2 59	3 14	3 26	3 38	3 58	4 07
BURNLEY	2 17	2 32	2 47	3 01	3 16	3 28	3 40	4 00	4 09
HEYINGTON	2 20	2 35	2 50	3 04	3 19	3 31	3 43	4 03	4 12
KOOYONG	2 22	2 37	2 52	3 06	3 21	3 33	3 45	4 05	4 14
TOORONGA	2 24	2 39	2 54	3 08	3 23	3 35	3 47	4 07	4 16
GARDINER	2 26	2 41	2 56	3 10	3 25	3 37	3 49	4 09	4 18
GLEN IRIS	2 28	2 43	2 58	3 12	3 27	3 39	3 51	4 11	4 20
DARLING	2 30	2 45	3 00	3 14	3 29	3 41	3 53	4 13	4 22
EAST MALVERN	2 32	2 47	3 02	3 16	3 31	3 43	3 55	4 15	4 24
HOLMESGLEN	2 35	2 50	3 05	3 19	3 34	3 46	3 58	4 18	4 27
JORDANVILLE	2 38	2 53	3 08	3 22	3 37	3 49	4 01	4 21	4 30
MOUNT WAVERLEY	2 40	2 55	3 10	3 24	3 39	3 51	4 03	4 23	4 32
SYNDAL	2 43	2 58	3 13	3 27	3 42	3 54	4 06	4 26	4 35
GLEN WAVERLEY Arr.	2 46	3 01	3 16	3 30	3 45	3 57	4 09	4 29	4 38

FLINDERS STREET Dep.	4 09	4 19	4 31	4 48	4 57	4 56	5 06	5 05	5 16
SPENCER STREET	4 12	4 22	4 34	..	..	4 59	..	5 08	..
FLAGSTAFF	•	•	•	..	..	•	..	•	..
MUSEUM	4 15	4 25	4 37	..	..	5 02	..	5 11	..
PARLIAMENT	4 17	4 27	4 39	..	..	5 04	..	5 13	..
RICHMOND	4 20	4 30	4 42	4 51	5 00	5 07	5 09	5 16	5 19
EAST RICHMOND	4 22	4 32	4 44	4 53	5 02	•	5 11	•	5 21
BURNLEY	4 24	4 34	4 46	4 55	5 04	•	5 13	•	5 23
HEYINGTON	4 27	4 37	4 49	4 58	5 07	•	5 16	•	5 26
KOOYONG	4 29	4 39	4 51	5 00	5 09	•	5 18	•	5 28
TOORONGA	4 31	4 41	4 53	5 02	5 11	•	5 20	•	5 30
GARDINER	4 33	4 43	4 55	5 04	5 13	•	5 22	•	5 32
GLEN IRIS	4 35	4 45	4 57	5 06	5 15	•	5 24	•	5 34
DARLING	4 37	4 47	4 59	5 08	5 17	5 20	5 26	5 29	5 36
EAST MALVERN	4 39	4 49	5 01	5 10	5 19	5 22	5 28	5 31	5 38
HOLMESGLEN	4 42	4 52	5 04	5 13	5 22	5 25	5 31	5 33	5 41
JORDANVILLE	4 45	4 55	5 07	5 16	5 25	5 28	5 34	5 37	5 44
MOUNT WAVERLEY	4 47	4 57	5 09	5 18	5 27	5 30	5 36	5 39	5 46
SYNDAL	4 50	5 00	5 12	5 21	5 30	5 33	5 39	5 42	5 49
GLEN WAVERLEY Arr.	4 53	5 03	5 15	5 24	5 33	5 36	5 42	5 45	5 52

FLINDERS STREET Dep.	5 14	5 23	5 34	5 42	6 00	6 19	6 39	6 59	7 29
SPENCER STREET	5 17	5 26	5 37	5 45	6 03	6 22	6 42	7 02	7 32
FLAGSTAFF	•	•	•	•	•	•	•	•	•
MUSEUM	5 20	5 29	5 40	5 48	6 06	6 25	6 45	7 05	7 35
PARLIAMENT	5 22	5 31	5 42	5 50	6 08	6 27	6 47	7 07	7 37
RICHMOND	5 25	5 34	5 45	5 53	6 11	6 30	6 50	7 10	7 40
EAST RICHMOND	•	5 36	5 47	5 55	6 13	6 32	6 52	7 12	7 42
BURNLEY	5 28	5 38	5 49	5 57	6 15	6 34	6 54	7 14	7 44
HEYINGTON	5 31	5 41	5 52	6 00	6 18	6 37	6 57	7 17	7 47
KOOYONG	5 33	5 43	5 54	6 02	6 20	6 39	6 59	7 19	7 49
TOORONGA	5 35	5 45	5 56	6 04	6 22	6 41	7 01	7 21	7 51
GARDINER	5 37	5 47	5 58	6 06	6 24	6 43	7 03	7 23	7 53
GLEN IRIS	5 39	5 49	6 00	6 08	6 26	6 45	7 05	7 25	7 55
DARLING	5 41	5 51	6 02	6 10	6 28	6 47	7 07	7 27	7 57
EAST MALVERN	5 43	5 53	6 04	6 12	6 30	6 49	7 09	7 29	7 59
HOLMESGLEN	5 46	5 56	6 07	6 15	6 33	6 52	7 12	7 32	8 02
JORDANVILLE	5 49	5 59	6 10	6 18	6 36	6 55	7 15	7 35	8 05
MOUNT WAVERLEY	5 51	6 01	6 12	6 20	6 38	6 57	7 17	7 37	8 07
SYNDAL	5 54	6 04	6 15	6 23	6 41	7 00	7 20	7 40	8 10
GLEN WAVERLEY Arr.	5 57	6 07	6 18	6 26	6 44	7 03	7 23	7 43	8 13

FLINDERS STREET Dep.	7 59	8 29	8 59	9 29	9 59	10 29	10 59	11 29	12 00
SPENCER STREET	8 02	8 32	9 02	9 32	10 02	10 32	11 02	11 32	12 03
FLAGSTAFF	•	•	•	•	•	•	•	•	•
MUSEUM	8 05	8 35	9 05	9 35	10 05	10 35	11 05	11 35	12 06
PARLIAMENT	8 07	8 37	9 07	9 37	10 07	10 37	11 07	11 37	12 08
RICHMOND	8 10	8 40	9 10	9 40	10 10	10 40	11 10	11 40	12 11
EAST RICHMOND	8 12	8 42	9 12	9 42	10 12	10 42	11 12	11 42	12 13
BURNLEY	8 14	8 44	9 14	9 44	10 14	10 44	11 14	11 44	12 16
HEYINGTON	8 17	8 47	9 17	9 47	10 17	10 47	11 17	11 47	12 18
KOOYONG	8 19	8 49	9 19	9 49	10 19	10 49	11 19	11 49	12 20
TOORONGA	8 21	8 51	9 21	9 51	10 21	10 51	11 21	11 51	12 22
GARDINER	8 23	8 53	9 23	9 53	10 23	10 53	11 23	11 53	12 24
GLEN IRIS	8 25	8 55	9 25	9 55	10 25	10 55	11 25	11 55	12 26
DARLING	8 27	8 57	9 27	9 57	10 27	10 57	11 27	11 57	12 28
EAST MALVERN	8 29	8 59	9 29	9 59	10 29	10 59	11 29	11 59	12 30
HOLMESGLEN	8 32	9 02	9 32	10 02	10 32	11 02	11 32	12 02	12 33
JORDANVILLE	8 35	9 05	9 35	10 05	10 35	11 05	11 35	12 05	12 36
MOUNT WAVERLEY	8 37	9 07	9 37	10 07	10 37	11 07	11 37	12 07	12 38
SYNDAL	8 40	9 10	9 40	10 10	10 40	11 10	11 40	12 10	12 41
GLEN WAVERLEY Arr.	8 43	9 13	9 43	10 13	10 43	11 13	11 43	12 13	12 44

Figure 1.1 A typical railway timetable

Figure 1.2 A typical tram timetable

WEEKDAYS

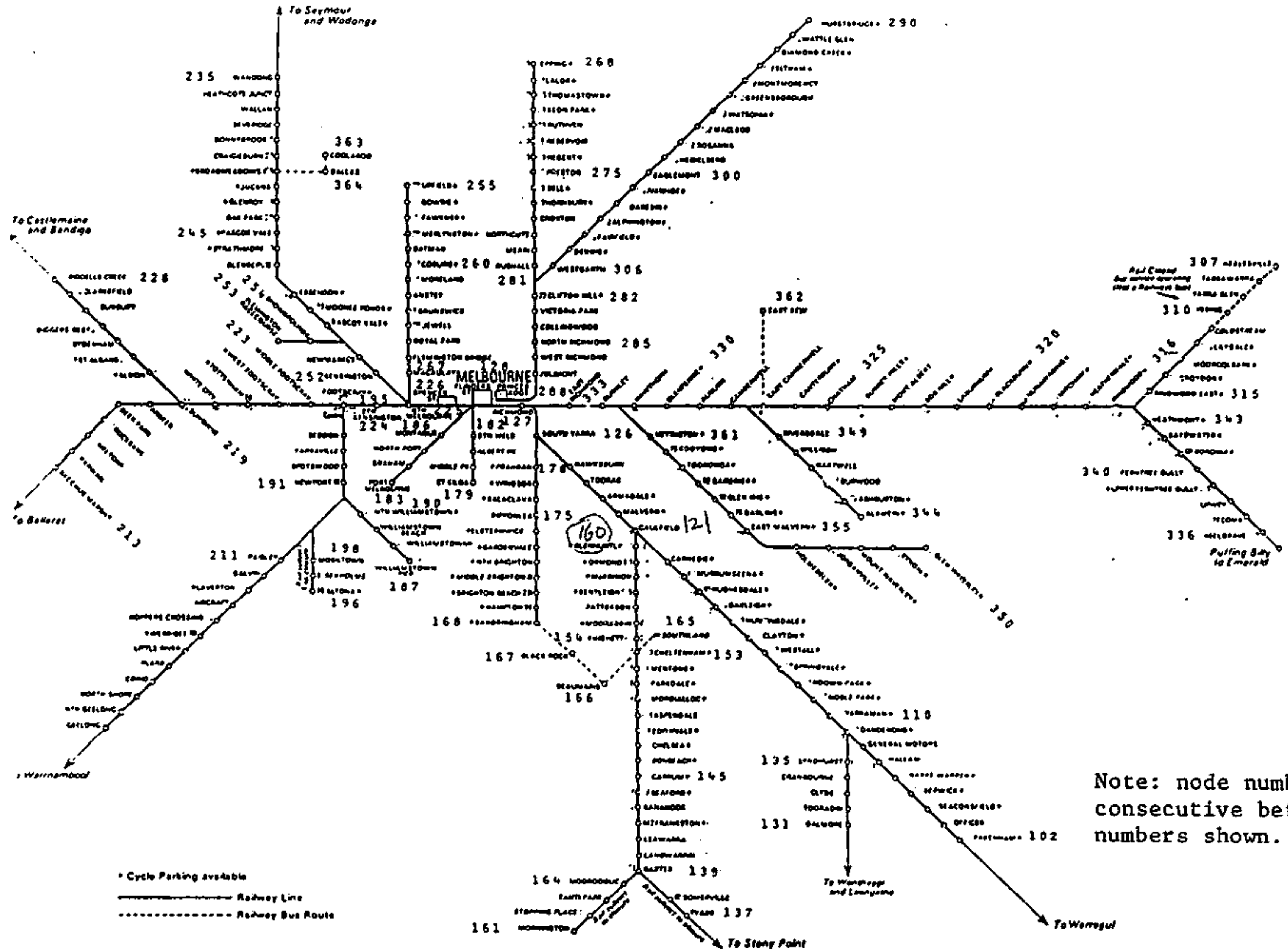
WATTLE PARK – CITY

DEP ELGAR ROAD	DEP WATT-LE PARK	DEP RIVE-RSDALE	DEP CAMB DEP-OT	DEP GLEN-FERRIE ROAD	DEP BUR-NLEY	DEP CHURCH ST.	DEP PUNT ROAD	ARRIVE PRINCES BRIDGE
			5.22	5.27	5.33	5.37	5.41	5.45
5.42	5.46	5.51	5.59	6.04	6.10	6.14	6.18	6.22
5.57	6.01	6.06	6.14	6.19	6.25	6.29	6.33	6.37
6.12	6.16	6.21	6.29	6.34	6.40	6.44	6.48	6.52
6.25	6.29	6.34	6.42	6.47	6.53	6.57	7.01	7.05
6.35	6.39	6.44	6.52	6.57	7.03	7.07	7.11	7.15
6.45	6.49	6.54	7.02	7.07	7.13	7.17	7.21	7.25
6.51	6.55	7.00	7.08	7.13	7.19	7.23	7.27	7.31
6.57	7.01	7.06	7.14	7.19	7.25	7.29	7.33	7.37
7.03	7.07	7.12	7.20	7.25	7.31	7.35	7.39	7.43
7.08	7.12	7.17	7.25	7.30	7.36	7.40	7.44	7.48
7.14	7.18	7.23	7.31	7.36	7.42	7.46	7.50	7.54
7.19	7.23	7.28	7.36	7.41	7.47	7.51	7.55	7.59
7.24	7.28	7.33	7.41	7.46	7.52	7.56	8.00	8.04
7.29	7.33	7.38	7.46	7.51	7.57	8.01	8.05	8.09
7.34	7.38	7.43	7.51	7.56	8.02	8.06	8.10	8.14
7.39	7.43	7.48	7.56	8.01	8.07	8.11	8.15	8.19
7.43	7.47	7.52	8.00	8.05	8.11	8.15	8.19	8.23
7.47	7.51	7.56	8.04	8.09	8.15	8.19	8.23	8.27
7.51	7.55	8.00	8.08	8.13	8.19	8.23	8.27	8.31
7.55	7.59	8.04	8.12	8.17	8.23	8.27	8.31	8.35
8.00	8.04	8.09	8.17	8.22	8.28	8.32	8.36	8.40
8.06	8.10	8.15	8.23	8.28	8.34	8.38	8.42	8.46
8.14	8.18	8.23	8.31	8.36	8.42	8.46	8.50	8.54
8.22	8.26	8.31	8.39	8.44	8.50	8.54	8.58	9.02
8.32	8.36	8.41	8.49	8.54	9.00	9.04	9.08	9.12
8.42	8.46	8.51	8.59	9.04	9.10	9.14	9.18	9.22
8.54	8.58	9.03	9.11	9.16	9.22	9.26	9.30	9.34
9.06	9.10	9.15	9.23	9.28	9.34	9.38	9.42	9.46
9.18	9.22	9.27	9.35	9.40	9.46	9.50	9.54	9.58
9.30	9.34	9.39	9.47	9.52	9.58	10.02	10.06	10.10
9.42	9.46	9.51	9.59	10.04	10.10	10.14	10.18	10.22
9.54	9.58	10.03	10.11	10.16	10.22	10.26	10.30	10.34
10.06	10.10	10.15	10.23	10.28	10.34	10.38	10.42	10.46
10.18	10.22	10.27	10.35	10.40	10.46	10.50	10.54	10.58
10.30	10.34	10.39	10.47	10.52	10.58	11.02	11.06	11.10
10.42	10.46	10.51	10.59	11.04	11.10	11.14	11.18	11.22
10.54	10.58	11.03	11.11	11.16	11.22	11.26	11.30	11.34
11.06	11.10	11.15	11.23	11.28	11.34	11.38	11.42	11.46
11.18	11.22	11.27	11.35	11.40	11.46	11.50	11.54	11.58
11.30	11.34	11.39	11.47	11.52	11.58	12.02	12.06	12.10
11.42	11.46	11.51	11.59	12.04	12.10	12.14	12.18	12.22
11.54	11.58	12.03	12.11	12.16	12.22	12.26	12.30	12.34
12.06	12.10	12.15	12.23	12.28	12.34	12.38	12.42	12.46
12.18	12.22	12.27	12.35	12.40	12.46	12.50	12.54	12.58
12.30	12.34	12.39	12.47	12.52	12.58	1.02	1.06	1.10
12.42	12.46	12.51	12.59	1.04	1.10	1.14	1.18	1.22
12.54	12.58	1.03	1.11	1.16	1.22	1.26	1.30	1.34
1.06	1.10	1.15	1.23	1.28	1.34	1.38	1.42	1.46
1.18	1.22	1.27	1.35	1.40	1.46	1.50	1.54	1.58
1.30	1.34	1.39	1.47	1.52	1.58	2.02	2.06	2.10
1.42	1.46	1.51	1.59	2.04	2.10	2.14	2.18	2.22
1.54	1.58	2.03	2.11	2.16	2.22	2.26	2.30	2.34
2.06	2.10	2.15	2.23	2.28	2.34	2.38	2.42	2.46
2.18	2.22	2.27	2.35	2.40	2.46	2.50	2.54	2.58
2.26	2.30	2.35	2.43	2.48	2.54	2.58	3.02	3.06
2.36	2.40	2.45	2.53	2.58	3.04	3.08	3.12	3.16

DEP ELGAR ROAD	DEP WATT-LE PARK	DEP RIVE-RSDALE	DEP CAMB DEP-OT	DEP GLEN-FERRIE ROAD	DEP BUR-NLEY	DEP CHURCH ST.	DEP PUNT ROAD	ARRIVE PRINCES BRIDGE
2.45	2.49	2.54	3.02	3.07	3.13	3.17	3.21	3.25
			3.08	3.13	3.19	3.23	3.27	3.31
2.56	3.00	3.05	3.13	3.18	3.24	3.28	3.32	3.36
			3.20	3.25	3.31	3.35	3.39	3.43
3.07	3.11	3.16	3.24	3.29	3.35	3.39	3.43	3.47
3.17	3.21	3.26	3.34	3.39	3.45	3.49	3.53	3.57
3.25	3.29	3.34	3.42	3.47	3.53	3.57	4.01	4.05
		3.39	3.47	3.52	3.58	4.02	4.06	4.10
3.36	3.40	3.45	3.53	3.58	4.04	4.08	4.12	4.16
			3.58	4.03	4.09	4.13	4.17	4.21
3.46	3.50	3.55	4.03	4.08	4.14	4.18	4.22	4.26
			4.08	4.13	4.19	4.23	4.27	4.31
3.56	4.00	4.05	4.13	4.18	4.24	4.28	4.32	4.36
			4.18	4.23	4.29	4.33	4.37	4.41
4.05	4.09	4.14	4.22	4.27	4.33	4.37	4.41	4.45
4.13	4.17	4.22	4.30	4.35	4.41	4.45	4.49	4.53
4.20	4.24	4.29	4.37	4.42	4.48	4.52	4.56	5.00
	4.28	4.33	4.41	4.46	4.52	4.56	5.00	5.04
4.28	4.32	4.37	4.45	4.50	4.56	5.00	5.04	5.08
			4.49	4.54	5.00	5.04	5.08	5.12
4.37	4.41	4.46	4.54	4.59	5.05	5.09	5.13	5.17
	4.47	4.52	5.00	5.05	5.11	5.15	5.19	5.23
4.47	4.51	4.56	5.04	5.09	5.15	5.19	5.23	5.27
	4.58	5.03	5.11	5.16	5.22	5.26	5.30	5.34
4.57	5.01	5.06	5.14	5.19	5.25	5.29	5.33	5.37
	5.08	5.13	5.21	5.26	5.32	5.36	5.40	5.44
5.07	5.11	5.16	5.24	5.29	5.35	5.39	5.43	5.47
	5.18	5.23	5.31	5.36	5.42	5.46	5.50	5.54
5.17	5.21	5.26	5.33	5.38	5.44	5.47	5.51	5.55
5.30	5.34	5.39	5.46	5.51	5.57	6.00	6.04	6.08
5.40	5.44	5.49	5.56	6.01	6.07	6.10	6.14	6.18
5.52	5.56	6.01	6.08	6.13	6.19	6.22	6.26	6.30
5.57	6.01	6.06	6.13					
6.07	6.11	6.16	6.23	6.28	6.34	6.37	6.41	6.45
6.22	6.26	6.31	6.38	6.43	6.49	6.52	6.56	7.00
6.29	6.33	6.38	6.45					
6.37	6.41	6.46	6.53	6.58	7.04	7.07	7.11	7.15
6.51	6.55	7.00	7.07	7.12	7.18	7.21	7.25	7.29
7.05	7.09	7.14	7.21	7.26	7.32	7.35	7.39	7.43
7.25	7.29	7.34	7.41	7.46	7.52	7.55	7.59	8.03
7.45	7.48	7.53	8.00	8.05	8.11	8.14	8.17	8.20
8.05	8.08	8.13	8.20	8.25	8.31	8.34	8.37	8.40
8.11	8.14	8.19	8.26					
8.25	8.28	8.33	8.40	8.45	8.51	8.54	8.57	9.00
8.45	8.48	8.53	9.00	9.05	9.11	9.14	9.17	9.20
9.05	9.08	9.13	9.20	9.25	9.31	9.34	9.37	9.40
9.25	9.28	9.33	9.40	9.45	9.51	9.54	9.57	10.00
9.45	9.48	9.53	10.00	10.05	10.11	10.14	10.17	10.20
10.05	10.08	10.13	10.20	10.25	10.31	10.34	10.37	10.40
10.25	10.28	10.33	10.40	10.45	10.51	10.54	10.57	11.00
10.45	10.48	10.53	11.00	11.05	11.11	11.14	11.17	11.20
11.05	11.08	11.13	11.20	11.25	11.31	11.34	11.37	11.40
11.28	11.31	11.36	11.43	11.48	11.54	11.57	12.00	12.03
11.39	11.42	11.47	11.54					
11.59	12.02	12.07	12.14					
12.19	12.22	12.27	12.34					
12.41	12.44	12.49	12.56					

DETAILED TIME POINTS (As per Headings)

ELGAR ROAD	–	Elgar and Riversdale Roads
WATTLE PARK	–	Warrigel and Riversdale Roads
RIVERSDALE	–	Wattle Valley & Riversdale Roads
CAMBERWELL DEPOT	–	Depot Entrance – Just west of Camberwell Junction.
GLENFERRIE ROAD	–	Glenferrie and Riversdale Roads
BURNLEY	–	Park Grove and Swan Street
CHURCH STREET	–	Church and Swan Streets.
PUNT ROAD	–	Punt Road and Swan Street
PRINCES BRIDGE	–	Batman Avenue and Princes Bridge



Note: node numbers are consecutive between numbers shown.

101 → 366

Figure 1.3 Node numbers for rail stations in Melbourne rail network

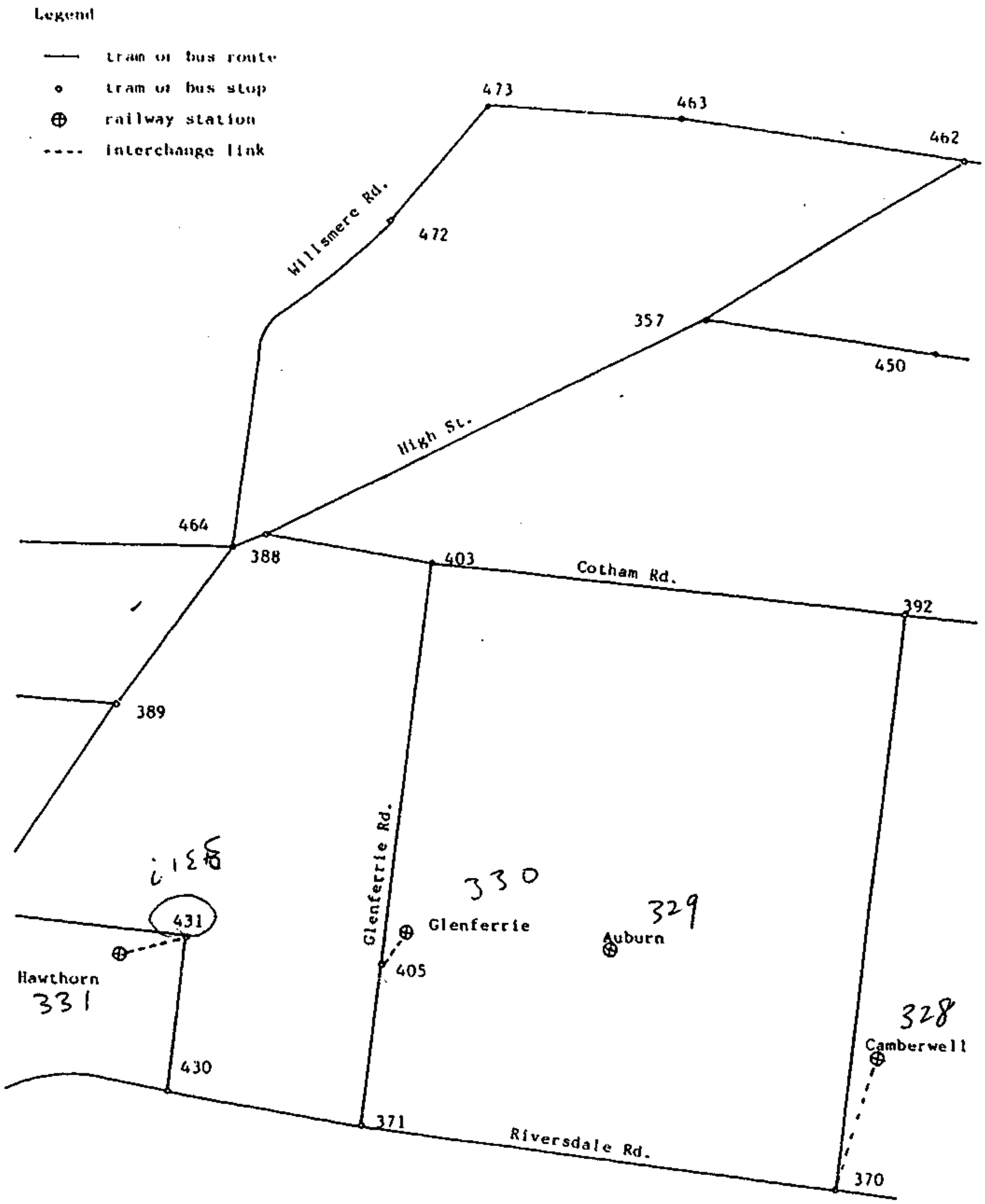


Figure 1.4 Part of the tram and bus network of Melbourne

network. For the rail network shown in Figure 1.3, every stop is assumed to be a node and is assigned a unique node number. If you do not intend to built a network with connections to traffic zones, then the smallest node number can be 1 otherwise it must be larger than the largest zone number used. Node numbers should not be larger than 9900.

For a tram or bus network, there are usually too many stops and it is not practicable to make every stop a node. In this case, only major stops and/or timing points referred to in the route timetables are treated as nodes in the network. If a suitable network diagram is not available, it is necessary to draw one. The best way of doing it is to overlay tracing paper on a base plan covering the study area and draw on it the alignment of the various bus and tram routes as indicated in the timetables. On each route, major stops are indicated as nodes in the diagram and every new node thus produced is assigned a new unique node number. If different routes use the same stop, then only one node and one node number is used for that stop.

Figure 1.4 shows part of a bus and tram network diagram drawn by a method as explained in the previous paragraph. The locations of rail stations are also indicated in the same diagram so that suitable walk links connecting different stops/stations for interchange between routes can be drawn.

As numbers are assigned to nodes in the network, then for future reference, these are marked on the route timetables adjacent to the stops along the route which they represent. In the case of bus or tram routes, if there are too few timing points shown on the timetables, then some other stops along the route can be converted to nodes and added to the network.

These additional node numbers are referenced at the appropriate places in the timetable shown in Figure 1.5.

1.4 Identifying Various Routes on a Timetable.

In the database, the definition of a route is different from those normally used and shown in the timetables. It is specified by the following:

(a) a sequence of nodes representing timing points, and

(b) the difference in starting time between successive nodes along the route (this will be referred to as the route profile).

Using the above definition of a route, the time schedules of a timetable are inspected to check whether they all stop at the same timing points and operate at the same profile. Very often it is found that the time schedules have to be separated into different routes. Figure 1.6 shows an example of how a rail timetable is separated into different routes.

In some transit services terminating with a loop, such as the Melbourne Underground Rail Loop in the Glen Waverley Line timetable shown in Figure 1.6, the end of the route going into the city and the beginning of the route coming out of the city are not well defined. In fact, in this case, the looping service is shown on the timetables of both the inward and outward services. With the coding procedure described in Section 1.6 this will give rise to duplication of services in the data file. If the duplication is avoided by setting some arbitary point as the beginning of one route and the end of the other, this would however wrongly imply the need for a route transfer at this point. The same problem also arises in some through-running services such as the Bulleen-Garden City bus route. While there appears to be no convenient way to avoid the duplication, it does not seem to cause any error in the modelling.

Figure 1.5 Assigning node numbers to a tram route

WEEKDAYS

WATTLE PARK — CITY

429 (between nodes 369 and 370) 430 (between nodes 371 and 372)

node numbers

367 DEP ELGAR ROAD	368 DEP WATTLE PARK	369 DEP RIVERSDALE	370 DEP CAMB DEPOT	371 DEP GLENFERRIE ROAD	372 DEP BURNLEY	373 DEP CHURCH ST.	374 DEP PUNT ROAD	375 ARRIVE PRINCES BRIDGE
			5.22	5.27	5.33	5.37	5.41	5.45
5.42	5.46	5.51	5.59	6.04	6.10	6.14	6.18	6.22
5.57	6.01	6.06	6.14	6.19	6.25	6.29	6.33	6.37
6.12	6.16	6.21	6.29	6.34	6.40	6.44	6.48	6.52
6.25	6.29	6.34	6.42	6.47	6.53	6.57	7.01	7.05
6.35	6.39	6.44	6.52	6.57	7.03	7.07	7.11	7.15
6.45	6.49	6.54	7.02	7.07	7.13	7.17	7.21	7.25
6.51	6.55	7.00	7.08	7.13	7.19	7.23	7.27	7.31
6.57	7.01	7.06	7.14	7.19	7.25	7.29	7.33	7.37
7.03	7.07	7.12	7.20	7.25	7.31	7.35	7.39	7.43
7.08	7.12	7.17	7.25	7.30	7.36	7.40	7.44	7.48
7.14	7.18	7.23	7.31	7.36	7.42	7.46	7.50	7.54
7.19	7.23	7.28	7.36	7.41	7.47	7.51	7.55	7.59
7.24	7.28	7.33	7.41	7.46	7.52	7.56	8.00	8.04
7.29	7.33	7.38	7.46	7.51	7.57	8.01	8.05	8.09
7.34	7.38	7.43	7.51	7.56	8.02	8.06	8.10	8.14
7.39	7.43	7.48	7.56	8.01	8.07	8.11	8.15	8.19
7.43	7.47	7.52	8.00	8.05	8.11	8.15	8.19	8.23
7.47	7.51	7.56	8.04	8.09	8.15	8.19	8.23	8.27
7.51	7.55	8.00	8.08	8.13	8.19	8.23	8.27	8.31
7.55	7.59	8.04	8.12	8.17	8.23	8.27	8.31	8.35
8.00	8.04	8.09	8.17	8.22	8.28	8.32	8.36	8.40
8.06	8.10	8.15	8.23	8.28	8.34	8.38	8.42	8.46
8.14	8.18	8.23	8.31	8.36	8.42	8.46	8.50	8.54
8.22	8.26	8.31	8.39	8.44	8.50	8.54	8.58	9.02
8.32	8.36	8.41	8.49	8.54	9.00	9.04	9.08	9.12
8.42	8.46	8.51	8.59	9.04	9.10	9.14	9.18	9.22
8.54	8.58	9.03	9.11	9.16	9.22	9.26	9.30	9.34
9.06	9.10	9.15	9.23	9.28	9.34	9.38	9.42	9.46
9.18	9.22	9.27	9.35	9.40	9.46	9.50	9.54	9.58
9.30	9.34	9.39	9.47	9.52	9.58	10.02	10.06	10.10
9.42	9.46	9.51	9.59	10.04	10.10	10.14	10.18	10.22
9.54	9.58	10.03	10.11	10.16	10.22	10.26	10.30	10.34
10.06	10.10	10.15	10.23	10.28	10.34	10.38	10.42	10.46
10.18	10.22	10.27	10.35	10.40	10.46	10.50	10.54	10.58
10.30	10.34	10.39	10.47	10.52	10.58	11.02	11.06	11.10
10.42	10.46	10.51	10.59	11.04	11.10	11.14	11.18	11.22
10.54	10.58	11.03	11.11	11.16	11.22	11.26	11.30	11.34
11.06	11.10	11.15	11.23	11.28	11.34	11.38	11.42	11.46
11.18	11.22	11.27	11.35	11.40	11.46	11.50	11.54	11.58
11.30	11.34	11.39	11.47	11.52	11.58	12.02	12.06	12.10
11.42	11.46	11.51	11.59	12.04	12.10	12.14	12.18	12.22
11.54	11.58	12.03	12.11	12.16	12.22	12.26	12.30	12.34
12.06	12.10	12.15	12.23	12.28	12.34	12.38	12.42	12.46
12.18	12.22	12.27	12.35	12.40	12.46	12.50	12.54	12.58
12.30	12.34	12.39	12.47	12.52	12.58	1.02	1.06	1.10
12.42	12.46	12.51	12.59	1.04	1.10	1.14	1.18	1.22
12.54	12.58	1.03	1.11	1.16	1.22	1.26	1.30	1.34
1.06	1.10	1.15	1.23	1.28	1.34	1.38	1.42	1.46
1.18	1.22	1.27	1.35	1.40	1.46	1.50	1.54	1.58
1.30	1.34	1.39	1.47	1.52	1.58	2.02	2.06	2.10
1.42	1.46	1.51	1.59	2.04	2.10	2.14	2.18	2.22
1.54	1.58	2.03	2.11	2.16	2.22	2.26	2.30	2.34
2.06	2.10	2.15	2.23	2.28	2.34	2.38	2.42	2.46
2.18	2.22	2.27	2.35	2.40	2.46	2.50	2.54	2.58
2.26	2.30	2.35	2.43	2.48	2.54	2.58	3.02	3.06
2.36	2.40	2.45	2.53	2.58	3.04	3.08	3.12	3.16

DEP ELGAR ROAD	DEP WATTLE PARK	DEP RIVERSDALE	DEP CAMB DEPOT	DEP GLENFERRIE ROAD	DEP BURNLEY	DEP CHURCH ST.	DEP PUNT ROAD	ARRIVE PRINCES BRIDGE
2.45	2.49	2.54	3.02	3.07	3.13	3.17	3.21	3.25
			3.08	3.13	3.19	3.23	3.27	3.31
2.56	3.00	3.05	3.13	3.18	3.24	3.28	3.32	3.36
			3.20	3.25	3.31	3.35	3.39	3.43
3.07	3.11	3.16	3.24	3.29	3.35	3.39	3.43	3.47
3.17	3.21	3.26	3.34	3.39	3.45	3.49	3.53	3.57
3.25	3.29	3.34	3.42	3.47	3.53	3.57	4.01	4.05
		3.39	3.47	3.52	3.58	4.02	4.06	4.10
3.36	3.40	3.45	3.53	3.58	4.04	4.08	4.12	4.16
			3.58	4.03	4.09	4.13	4.17	4.21
3.46	3.50	3.55	4.03	4.08	4.14	4.18	4.22	4.26
			4.08	4.13	4.19	4.23	4.27	4.31
3.56	4.00	4.05	4.13	4.18	4.24	4.28	4.32	4.36
			4.18	4.23	4.29	4.33	4.37	4.41
4.05	4.09	4.14	4.22	4.27	4.33	4.37	4.41	4.45
4.13	4.17	4.22	4.30	4.35	4.41	4.45	4.49	4.53
4.20	4.24	4.29	4.37	4.42	4.48	4.52	4.56	5.00
	4.28	4.33	4.41	4.46	4.52	4.56	5.00	5.04
4.28	4.32	4.37	4.45	4.50	4.56	5.00	5.04	5.08
			4.49	4.54	5.00	5.04	5.08	5.12
4.37	4.41	4.46	4.54	4.59	5.05	5.09	5.13	5.17
	4.47	4.52	5.00	5.05	5.11	5.15	5.19	5.23
4.47	4.51	4.56	5.04	5.09	5.15	5.19	5.23	5.27
	4.58	5.03	5.11	5.16	5.22	5.26	5.30	5.34
4.57	5.01	5.06	5.14	5.19	5.25	5.29	5.33	5.37
	5.08	5.13	5.21	5.26	5.32	5.36	5.40	5.44
5.07	5.11	5.16	5.24	5.29	5.35	5.39	5.43	5.47
	5.18	5.23	5.31	5.36	5.42	5.46	5.50	5.54
5.17	5.21	5.26	5.33	5.38	5.44	5.47	5.51	5.55
5.30	5.34	5.39	5.46	5.51	5.57	6.00	6.04	6.08
5.40	5.44	5.49	5.56	6.01	6.07	6.10	6.14	6.18
5.52	5.56	6.01	6.08	6.13	6.19	6.22	6.26	6.30
5.57	6.01	6.06	6.13					
6.07	6.11	6.16	6.23	6.28	6.34	6.37	6.41	6.45
6.22	6.26	6.31	6.38	6.43	6.49	6.52	6.56	7.00
6.29	6.33	6.38	6.45					
6.37	6.41	6.46	6.53	6.58	7.04	7.07	7.11	7.15
6.51	6.55	7.00	7.07	7.12	7.18	7.21	7.25	7.29
7.05	7.09	7.14	7.21	7.26	7.32	7.35	7.39	7.43
7.25	7.29	7.34	7.41	7.46	7.52	7.55	7.59	8.03
7.45	7.48	7.53	8.00	8.05	8.11	8.14	8.17	8.20
8.05	8.08	8.13	8.20	8.25	8.31	8.34	8.37	8.40
8.11	8.14	8.19	8.24					
8.25	8.28	8.33	8.40	8.45	8.51	8.54	8.57	9.00
8.45	8.48	8.53	9.00	9.05	9.11	9.14	9.17	9.20
9.05	9.08	9.13	9.20	9.25	9.31	9.34	9.37	9.40
9.25	9.28	9.33	9.40	9.45	9.51	9.54	9.57	10.00
9.45	9.48	9.53	10.00	10.05	10.11	10.14	10.17	10.20
10.05	10.08	10.13	10.20	10.25	10.31	10.34	10.37	10.40
10.25	10.28	10.33	10.40	10.45	10.51	10.54	10.57	11.00
10.45	10.48	10.53	11.00	11.05	11.11	11.14	11.17	11.20
11.05	11.08	11.13	11.20	11.25	11.31	11.34	11.37	11.40
11.28	11.31	11.36	11.43	11.48	11.54	11.57	12.00	12.03
11.39	11.42	11.47	11.54					
11.59	12.02	12.07	12.14					
12.19	12.22	12.27	12.34					
12.41	12.44	12.49	12.56					

DETAILED TIME POINTS (As per Headings)

ELGAR ROAD	— Elgar and Riversdale Roads
WATTLE PARK	— Warrigal and Riversdale Roads
RIVERSDALE	— Wattle Valley & Riversdale Roads
CAMBERWELL DEPOT	— Depot Entrance — Just west of Camberwell Junction.
GLENFERRIE ROAD	— Glenferrie and Riversdale Roads
BURNLEY	— Park Grove and Swan Street
CHURCH STREET	— Church and Swan Streets
PUNT ROAD	— Punt Road and Swan Street
PRINCES BRIDGE	— Batman Avenue and Princes Bridge

GLEN WAVERLEY LINE

FROM MELBOURNE – Monday to Friday (Current from 1/11/82)

		1	1	1	1	1	1	1	1	2	3	4	5	6	6	3	5	5	1	1	5	6	6	6	1	5	5	5
PARLIAMENT	Dep.	..	..	..	..	..	..	..	..	6 51	7 08	..	7 43	7 56	8 07	8 23	8 35	9 00	..	..	9 44	10 00	10 15	10 30	..	10 59	11 14	11 29
MUSEUM		..	..	..	..	..	..	..	..	6 53	7 10	..	7 45	7 58	8 09	8 25	8 37	9 02	..	..	9 46	10 02	10 17	10 32	..	11 01	11 16	11 31
FLAGSTAFF		..	..	..	..	..	..	..	..	•	•	..	•	•	•	•	•	•	..	..	•	•	•	•	..	•	•	•
SPENCER STREET		..	..	..	..	..	..	..	..	6 56	7 13	..	7 48	8 01	8 12	8 28	8 40	9 05	..	..	9 49	10 05	10 20	10 35	..	11 04	11 19	11 34
FLINDERS STREET		4 47	5 22	5 44	6 12	6 31	6 39	[illegible]	[illegible]	7 05	7 17	7 30	7 53	8 07	8 18	8 32	8 45	9 10	9 25	9 40	9 55	10 10	10 25	10 40	10 55	11 10	11 25	11 40
RICHMOND		4 50	5 25	5 48	6 15	6 34	6 42	6 51	6 59	7 08	7 20	7 33	7 56	8 10	8 21	8 35	8 48	9 13	9 28	9 43	9 58	10 13	10 28	10 43	10 58	11 13	11 28	11 43
EAST RICHMOND		4 52	5 27	5 50	6 17	6 36	6 44	6 53	7 01	7 10	7 22		7 58	8 12	8 23	8 37	8 50	9 15	9 30	9 45	10 00	10 15	10 30	10 45	11 00	11 15	11 30	11 45
BURNLEY		4 54	5 29	5 52	6 19	6 38	6 46	6 55	7 03	7 12	7 24	7 36	8 00	8 14	8 25	8 39	8 52	9 17	9 32	9 47	10 02	10 17	10 32	10 47	11 02	11 17	11 32	11 47
HEYINGTON		4 57	5 32	5 55	6 22	6 41	6 49	6 58	7 06	7 15	7 27	7 39	8 03	8 17	8 28	8 42	8 55	9 20	9 35	9 50	10 05	10 20	10 35	10 50	11 05	11 20	11 35	11 50
KOOYONG		4 59	5 34	5 57	6 24	6 43	6 51	7 00	7 08	7 17	7 29	7 41	8 05	8 19	8 30	8 44	8 57	9 22	9 37	9 52	10 07	10 22	10 37	10 52	11 07	11 22	11 37	11 52
TOORONGA		5 01	5 36	5 59	6 26	6 45	6 53	7 02	7 10	7 19	7 31	7 43	8 07	8 21	8 32	8 46	8 59	9 24	9 39	9 54	10 09	10 24	10 39	10 54	11 09	11 24	11 39	11 54
GARDINER		5 03	5 38	6 01	6 28	6 47	6 55	7 04	7 12	7 21	7 33	7 45	8 09	8 23	8 34	8 48	9 01	9 26	9 41	9 56	10 11	10 26	10 41	10 56	11 11	11 26	11 41	11 56
GLEN IRIS		5 05	5 40	6 03	6 30	6 49	6 57	7 06	7 14	7 23	7 35	7 47	8 11	8 25	8 36	8 50	9 03	9 28	9 43	9 58	10 13	10 28	10 43	10 58	11 13	11 28	11 43	11 58
DARLING		5 07	5 42	6 05	6 32	6 51	6 59	7 08	7 16	7 25	7 37	7 49	8 13	8 27	8 38	8 52	9 05	9 30	9 45	10 00	10 15	10 30	10 45	11 00	11 15	11 30	11 45	12 00
EAST MALVERN		5 09	5 44	6 07	6 34	6 53	7 01	7 10	7 18	7 27	7 39	7 51	8 15	8 29	8 40	8 54	9 07	9 32	9 47	10 02	10 17	10 32	10 47	11 02	11 17	11 32	11 47	12 02
HOLMESGLEN		5 12	5 47	6 10	6 37	6 56	7 04	7 13	7 21	7 30	7 42	7 54	8 18	8 32	8 43	8 57	9 10	9 35	9 50	10 05	10 20	10 35	10 50	11 05	11 20	11 35	11 50	12 05
JORDANVILLE		5 15	5 50	6 13	6 40	6 59	7 07	7 16	7 24	7 33	7 45	7 57	8 21	8 35	8 46	9 00	9 13	9 38	9 53	10 08	10 23	10 38	10 53	11 08	11 23	11 38	11 53	12 08
MOUNT WAVERLEY		5 17	5 52	6 15	6 42	7 01	7 09	7 18	7 26	7 35	7 47	7 59	8 23	8 37	8 48	9 02	9 15	9 40	9 55	10 10	10 25	10 40	10 55	11 10	11 25	11 40	11 55	12 10
SYNDAL		5 20	5 55	6 18	6 45	7 04	7 12	7 21	7 29	7 38	7 50	8 02	8 26	8 40	8 51	9 05	9 18	9 43	9 58	10 13	10 28	10 43	10 58	11 13	11 28	11 43	11 58	12 13
GLEN WAVERLEY	Arr.	5 23	5 58	6 21	6 48	7 07	7 15	7 24	7 32	7 41	7 53	8 05	8 29	8 43	8 54	9 08	9 21	9 46	10 01	10 16	10 31	10 46	11 01	11 16	11 31	11 46	12 01	12 16

		7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	1	1	8	1	8	1
FLINDERS STREET	Dep.	11 47	12 02	12 17	12 32	12 47	1 02	1 17	1 32	1 47	2 02	2 17	2 32	2 46	3 01	3 13	3 25	3 45	3 54	4 09	4 19	4 31	4 48	4 57	4 56	5 06	5 05	5 16
SPENCER STREET		11 50	12 05	12 20	12 35	12 50	1 05	1 20	1 35	1 50	2 05	2 20	2 35	2 49	3 04	3 16	3 28	3 48	3 57	4 12	4 22	4 34	..	..	4 59	..	5 08	..
FLAGSTAFF		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	..	..	•	..	•	..
MUSEUM		11 53	12 08	12 23	12 38	12 53	1 08	1 23	1 38	1 53	2 08	2 23	2 38	2 52	3 07	3 19	3 31	3 51	4 00	4 15	4 25	4 37	..	..	5 02	..	5 11	..
PARLIAMENT		11 55	12 10	12 25	12 40	12 55	1 10	1 25	1 40	1 55	2 10	2 25	2 40	2 54	3 09	3 21	3 33	3 53	4 02	4 17	4 27	4 39	..	..	5 04	..	5 13	..
RICHMOND		11 58	12 13	12 28	12 43	12 58	1 13	1 28	1 43	1 58	2 13	2 28	2 43	2 57	3 12	3 24	3 36	3 56	4 05	4 20	4 30	4 42	4 51	5 00	5 07	5 09	5 16	5 19
EAST RICHMOND		12 00	12 15	12 30	12 45	1 00	1 15	1 30	1 45	2 00	2 15	2 30	2 45	2 59	3 14	3 26	3 38	3 58	4 07	4 22	4 32	4 44	4 53	5 02	•	5 11	•	5 21
BURNLEY		12 02	12 17	12 32	12 47	1 02	1 17	1 32	1 47	2 02	2 17	2 32	2 47	3 01	3 16	3 28	3 40	4 00	4 09	4 24	4 34	4 46	4 55	5 04	•	5 13	•	5 23
HEYINGTON		12 05	12 20	12 35	12 50	1 05	1 20	1 35	1 50	2 05	2 20	2 35	2 50	3 04	3 19	3 31	3 43	4 03	4 12	4 27	4 37	4 49	4 58	5 07	•	5 16	•	5 26
KOOYONG		12 07	12 22	12 37	12 52	1 07	1 22	1 37	1 52	2 07	2 22	2 37	2 52	3 06	3 21	3 33	3 45	4 05	4 14	4 29	4 39	4 51	5 00	5 09	•	5 18	•	5 28
TOORONGA		12 09	12 24	12 39	12 54	1 09	1 24	1 39	1 54	2 09	2 24	2 39	2 54	3 08	3 23	3 35	3 47	4 07	4 16	4 31	4 41	4 53	5 02	5 11	•	5 20	•	5 30
GARDINER		12 11	12 26	12 41	12 56	1 11	1 26	1 41	1 56	2 11	2 26	2 41	2 56	3 10	3 25	3 37	3 49	4 09	4 18	4 33	4 43	4 55	5 04	5 13	•	5 22	•	5 32
GLEN IRIS		12 13	12 28	12 43	12 58	1 13	1 28	1 43	1 58	2 13	2 28	2 43	2 58	3 12	3 27	3 39	3 51	4 11	4 20	4 35	4 45	4 57	5 06	5 15	•	5 24	•	5 34
DARLING		12 15	12 30	12 45	1 00	1 15	1 30	1 45	2 00	2 15	2 30	2 45	3 00	3 14	3 29	3 41	3 53	4 13	4 22	4 37	4 47	4 59	5 08	5 17	5 20	5 26	5 29	5 36
EAST MALVERN		12 17	12 32	12 47	1 02	1 17	1 32	1 47	2 02	2 17	2 32	2 47	3 02	3 16	3 31	3 43	3 55	4 15	4 24	4 39	4 49	5 01	5 10	5 19	5 22	5 28	5 31	5 38
HOLMESGLEN		12 20	12 35	12 50	1 05	1 20	1 35	1 50	2 05	2 20	2 35	2 50	3 05	3 19	3 34	3 46	3 58	4 18	4 27	4 42	4 52	5 04	5 13	5 22	5 25	5 31	5 34	5 41
JORDANVILLE		12 23	12 38	12 53	1 08	1 23	1 38	1 53	2 08	2 23	2 38	2 53	3 08	3 22	3 37	3 49	4 01	4 21	4 30	4 45	4 55	5 07	5 16	5 25	5 28	5 34	5 37	5 44
MOUNT WAVERLEY		12 25	12 40	12 55	1 10	1 25	1 40	1 55	2 10	2 25	2 40	2 55	3 10	3 24	3 39	3 51	4 03	4 23	4 32	4 47	4 57	5 09	5 18	5 27	5 30	5 36	5 39	5 46
SYNDAL		12 28	12 43	12 58	1 13	1 28	1 43	1 58	2 13	2 28	2 43	2 58	3 13	3 27	3 42	3 54	4 06	4 26	4 35	4 50	5 00	5 12	5 21	5 30	5 33	5 39	5 42	5 49
GLEN WAVERLEY	Arr.	12 31	12 46	1 01	1 16	1 31	1 46	2 01	2 16	2 31	2 46	3 01	3 16	3 30	3 45	3 57	4 09	4 29	4 38	4 53	5 03	5 15	5 24	5 33	5 36	5 42	5 45	5 52

		9	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
FLINDERS STREET	Dep.	5 14	5 23	5 34	5 42	6 00	6 19	6 39	6 59	7 29	7 59	8 29	8 59	9 29	9 59	10 29	10 59	11 29	12 00
SPENCER STREET		5 17	5 26	5 37	5 45	6 03	6 22	6 42	7 02	7 32	8 02	8 32	9 02	9 32	10 02	10 32	11 02	11 32	12 03
FLAGSTAFF		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
MUSEUM		5 20	5 29	5 40	5 48	6 06	6 25	6 45	7 05	7 35	8 05	8 35	9 05	9 35	10 05	10 35	11 05	11 35	12 06
PARLIAMENT		5 22	5 31	5 42	5 50	6 08	6 27	6 47	7 07	7 37	8 07	8 37	9 07	9 37	10 07	10 37	11 07	11 37	12 08
RICHMOND		5 25	5 34	5 45	5 53	6 11	6 30	6 50	7 10	7 40	8 10	8 40	9 10	9 40	10 10	10 40	11 10	11 40	12 11
EAST RICHMOND		•	5 36	5 47	5 55	6 13	6 32	6 52	7 12	7 42	8 12	8 42	9 12	9 42	10 12	10 42	11 12	11 42	12 13
BURNLEY		5 28	5 38	5 49	5 57	6 15	6 34	6 54	7 14	7 44	8 14	8 44	9 14	9 44	10 14	10 44	11 14	11 44	12 15
HEYINGTON		5 31	5 41	5 52	6 00	6 18	6 37	6 57	7 17	7 47	8 17	8 47	9 17	9 47	10 17	10 47	11 17	11 47	12 18
KOOYONG		5 33	5 43	5 54	6 02	6 20	6 39	6 59	7 19	7 49	8 19	8 49	9 19	9 49	10 19	10 49	11 19	11 49	12 20
TOORONGA		5 35	5 45	5 56	6 04	6 22	6 41	7 01	7 21	7 51	8 21	8 51	9 21	9 51	10 21	10 51	11 21	11 51	12 22
GARDINER		5 37	5 47	5 58	6 06	6 24	6 43	7 03	7 23	7 53	8 23	8 53	9 23	9 53	10 23	10 53	11 23	11 53	12 24
GLEN IRIS		5 39	5 49	6 00	6 08	6 26	6 45	7 05	7 25	7 55	8 25	8 55	9 25	9 55	10 25	10 55	11 25	11 55	12 26
DARLING		5 41	5 51	6 02	6 10	6 28	6 47	7 07	7 27	7 57	8 27	8 57	9 27	9 57	10 27	10 57	11 27	11 57	12 28
EAST MALVERN		5 43	5 53	6 04	6 12	6 30	6 49	7 09	7 29	7 59	8 29	8 59	9 29	9 59	10 29	10 59	11 29	11 59	12 30
HOLMESGLEN		5 46	5 56	6 07	6 15	6 33	6 52	7 12	7 32	8 02	8 32	9 02	9 32	10 02	10 32	11 02	11 32	12 02	12 33
JORDANVILLE		5 49	5 59	6 10	6 18	6 36	6 55	7 15	7 35	8 05	8 35	9 05	9 35	10 05	10 35	11 05	11 35	12 05	12 36
MOUNT WAVERLEY		5 51	6 01	6 12	6 20	6 38	6 57	7 17	7 37	8 07	8 37	9 07	9 37	10 07	10 37	11 07	11 37	12 07	12 38
SYNDAL		5 54	6 04	6 15	6 23	6 41	7 00	7 20	7 40	8 10	8 40	9 10	9 40	10 10	10 40	11 10	11 40	12 10	12 41
GLEN WAVERLEY	Arr.	5 57	6 07	6 18	6 26	6 44	7 03	7 23	7 43	8 13	8 43	9 13	9 43	10 13	10 43	11 13	11 43	12 13	12 44

Figure 1.6 A railway timetable with 9 different route profiles

1.5 Assigning Routes With Names

When the various routes have been identified, each route is assigned with a route name. A route name can be up to three digits long with each digit being either a character or number.

It is preferable to adopt a system of naming routes so that routes of different modes can be identified from the name. For example rail route names can be specified to start with certain alphabets, while tram or bus routes with others.

1.6 Running Program [Tblcode]

This program can be used for coding timetable data interactively. The program is written so that as you are running the program, questions will appear in your terminal asking you to enter various types of data.

When you are coding a route by running [Tblcode], you will be prompted to enter the following information:

(a) the route name,

(b) the node numbers representing various major stops along the route,

(c) the route profile, and

(d) the list of starting times of each run. (All clock times used within the context of this manual refer to the 24-hour clock time system).

When you are coding walk links, you will be prompted to enter the following information:

(a) the route number which is called "WAL",

(b) for each walk link, the start node number, end node number and walking time in minutes of the walk link.

All walk links are assumed to be two-way links, therefore it is only necessary to code one direction of the link.

A sample of the type of questions appearing on your terminal and the data which you should enter is shown in Appendix A.

It is not necessary to code all you timetable data during one interactive session. You may stop after you have finished coding any route. The data which you have coded during each interactive session is stored in a temporary file called FOR003.

After running [Tblcode], you should proceed to check the data which you have entered and edit any errors detected. The procedure of doing this is shown in the flow chart in Figure 1.7.

1.7 Running Program [Tblcheck]

After you have finished one coding session and before starting another, you should first check the data which you have coded by running [Tblcheck].

This program performs the following systematic checks on the data:

(a) the departure times in a route profile must be in ascending order,
(b) starting times of different runs must not be earlier than 4.00 am, and
(c) the array of starting times of different runs must be in ascending order.

The routes will be checked sequentially in the order in which they are entered. When no error is detected in a route, the route number and the number of nodes in the route will appear on the terminal, otherwise additional error messages will be shown.

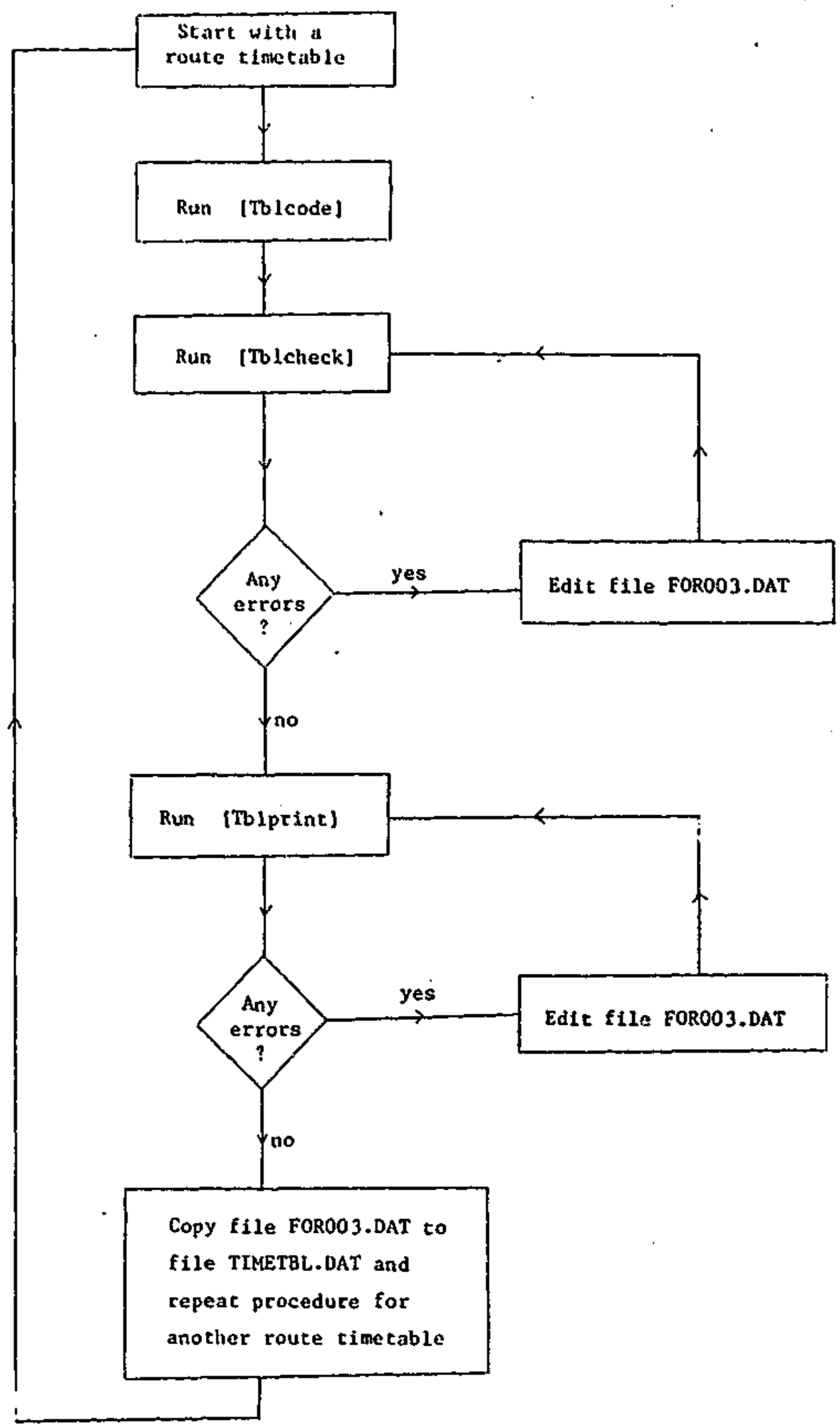


Figure 1.7 Operation sequence of timetable coding programs

When errors are detected, they should be corrected by editing the file FOR003.

1.8 Running Program [Tblprint]

This program regenerates the timetables and stores it on a file called FOR005. The file can then be printed for further checking.

1.9 Transfering Coded Data to a Permanent File

Your coded data is now still in a temporary file called FOR003. After you have checked the data and before running [Tblcode] again, you should transfer the data to a permanent file.

The permanent files for storing timetable data are called TIMETBLN9. The last two digits of the file name N9 is a network identification code.

For example, if you use a file called TIMETBL3 to store all rail timetables, then after coding some rail timetables from running [Tblcode], you transfer the data contained in FOR003 to TIMETBL3. The command used for this operation may vary with different computer systems, however, a typical command would be as follows:

```
@COPY TIMETBL3+FOR003  TIMETBL3
```

SECTION 2

CODING LOCATION OF NODES

SECTION 2 CODING LOCATION OF NODES

2.1 Introduction

Section 1 describes the assignment of major stops along a transit route as nodes and the construction of network diagrams such as those shown in Figures 1.3 and 1.4. However, the exact location of nodes have still not been coded. The coding of node locations is optional, there are some programs in this package which require this information and others which don't. If you wish to start coding node locations now, this section explains how this can be done. If not, you can proceed to read Section 4.

The method developed for coding is based on the convention adopted by the set of street maps for Melbourne, Australia which is published by Melway Publishing Pty. Ltd. (1982 edition).

2.2 Running Program [Melway]

The location of a node is first coded as its Melway map code. A typical Melway map is shown in Figure 2.1. The map code of any location within the map consists of the following three parts:

(a) page code,
(b) x-code (a character code), and
(c) y-code (a numeral code).

For the map shown in Figure 2.1, the map code is its page code, the x-code is a single character ranging from A to K (with character I omitted) and the y-code is a number ranging

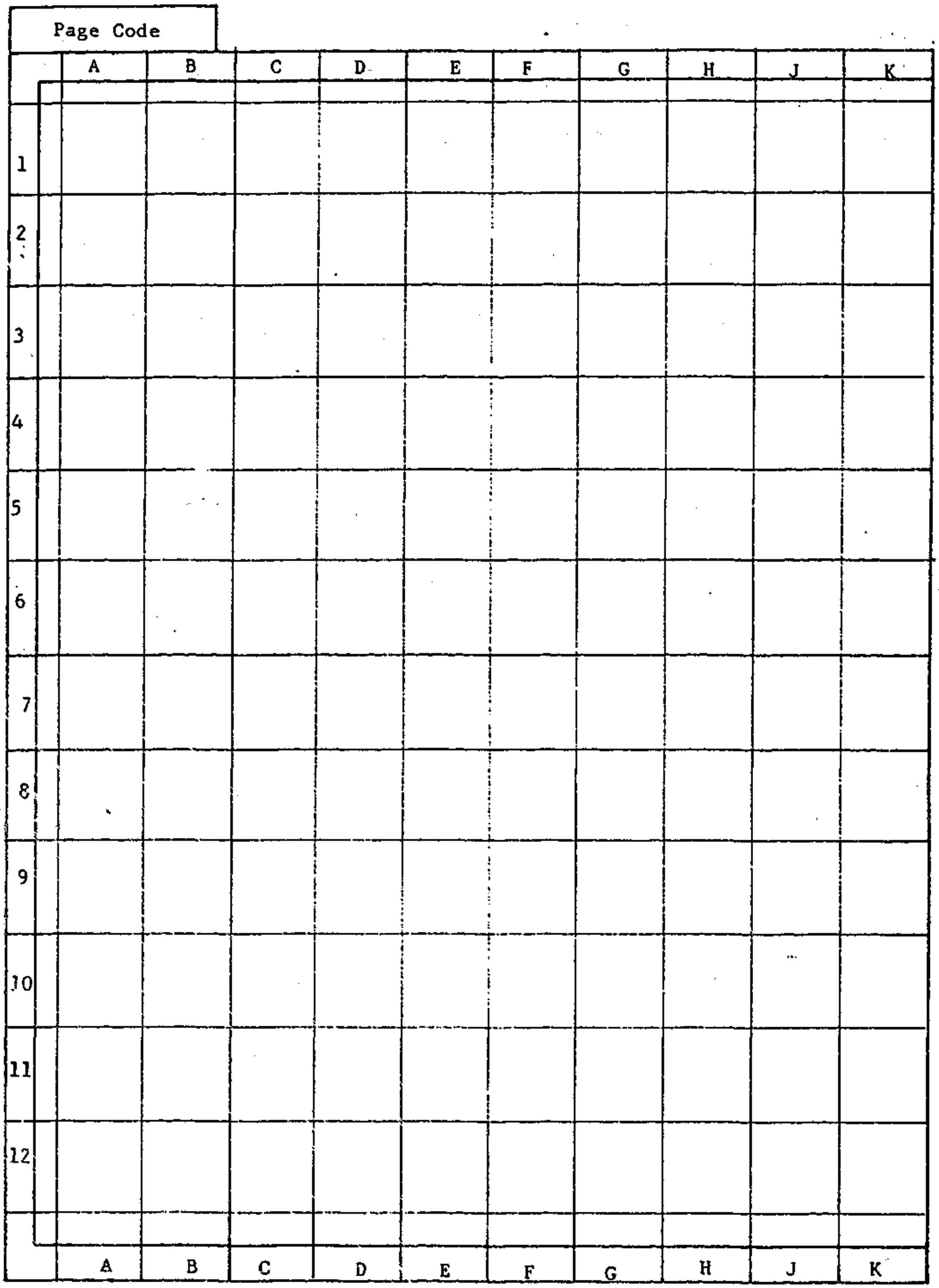


Figure 2.1 System of map grids used in a street map of Melbourne (Published by Melway Pty. Ltd.)

from 1 to 12. Thus, Melway map code 45D12 represent a point which falls within a cell located on map page code 45, column D and row 12.

For convenience in various computations, these map codes are converted to a set of x and y co-ordinates which are not Australian Map Grid Co-ordinates but quite arbitrarily defined as shown in Figure 2.2. This Figure shows the location of maps within the assigned x and y axes. There are two types of maps, the typical type is to scale 1:20000 with numeral page codes. However, there are also 12 maps of scale 1:10000 which are assigned with codes ranging from 2A to 2M.

The x and y co-ordinate of a cell within any map is given by the co-ordinates of the top left hand corner of that cell.

The location of maps within the xy-plane is coded by running [Melway]. When running this program, you will be prompted to enter the following information:

(a) is there an existing MELCODE file (Y/N)?
(b) page code,
(c) page type (1 for 1:20000 scale maps and 2 for 1:10000 scale maps),
(d) x co-ordinate of the top left hand corner of the map,
(e) y co-ordinate of the top left hand corner of the map.

The information coded is stored in a file called MELCODE. The coding of information can be continued in several runs of [Melway] and previously coded data changed by re-coding.

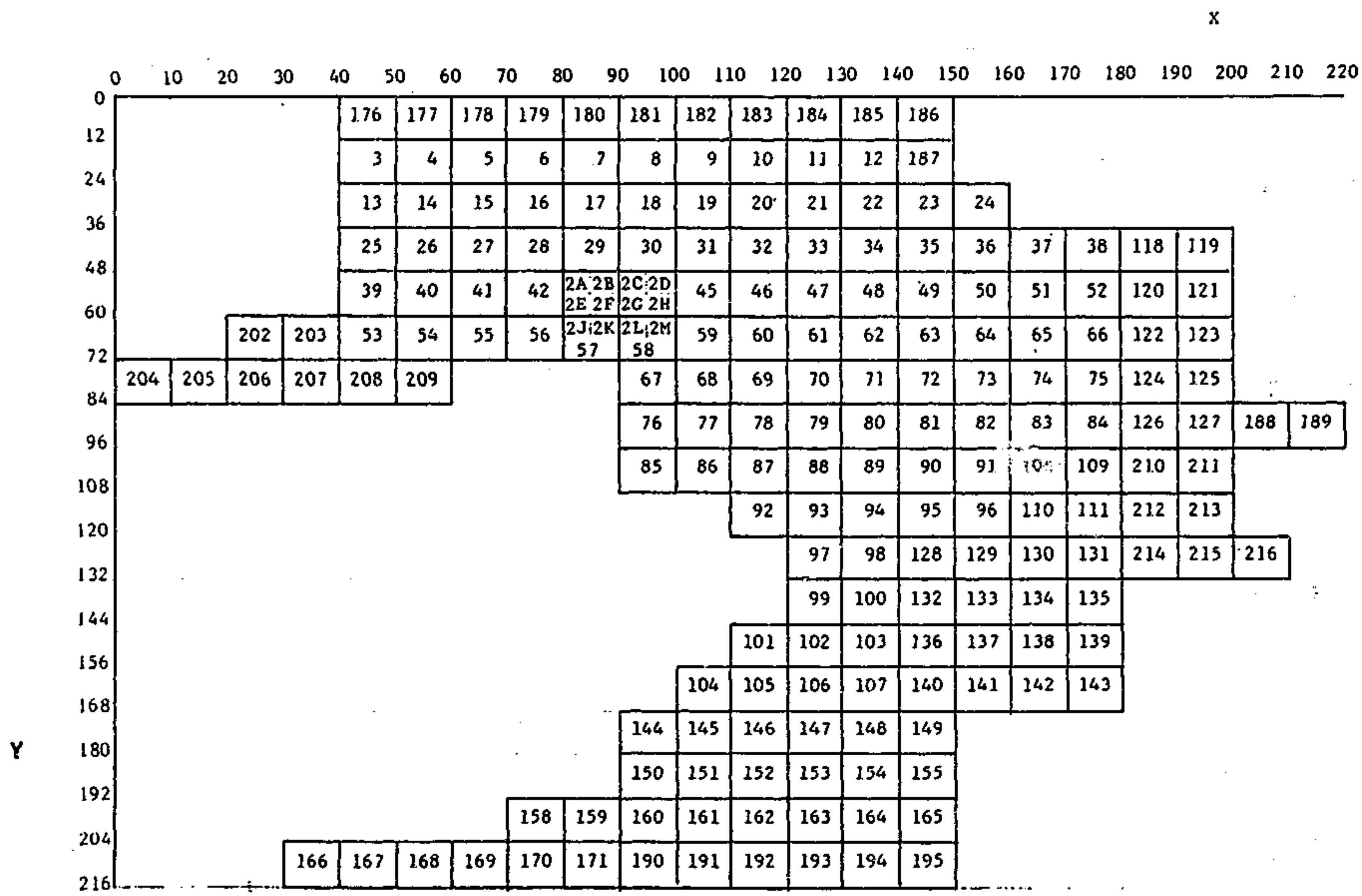


Figure 2.2 Location of street maps of Melbourne in hypothetical XY plane (according to Melway Publishing Pty. Ltd. 1982 edition)

2.3 Running Program [Nodcode]

The location of nodes is coded by running [Nodcode]. When running this program, you will be prompted to enter the following information:

(a) largest node number used in network,

(b) is there an existing MAPCODE file (Y/N),

then for each node to be coded, enter

(c) node number,

(d) page code,

(e) x code,

(f) y code, and

(g) name of station or stop represented by node (optional).

If the locations of some nodes have already been coded during previous sessions, then you enter Y in (b) and for each node to be coded, the program will then check whether this node has already been coded or not. If the node has already been coded, a warning message will appear in the terminal together with the previously coded Melway code. You then have the option of either using the previous code or inputting a new code.

The coded information is stored in a file called MAPCODE. Every time after you run [Nodcode], the file MAPCODE is updated.

2.4 Running Program [Grid]

[Grid] is used to convert the Melway map codes of nodes to xy co-ordinates. However, not all nodes that have previously been coded are converted by each run of [Grid]. Before running [Grid], it is necessary to generate a network file by running [Selink] (see Section 4.3), then only nodes which are used in this network are converted. The output file of [Grid] is XYCODEN9.

SECTION 3

CODING OF MINOR STOPS

SECTION 3 CODING OF MINOR STOPS

3.1 Introduction

In transit networks, there are often many stops in the transit routes and it is not practical to include every stop in the network as nodes. Using major stops as nodes can provide an approximate network model and the network generated is called a basic network. However, programs have been developed in this package which will enable you to include minor stops and hence build a detailed network.

The method used is to treat minor stops as extra nodes and insert them into the basic network in regions of the network where they are required. It is not practicable or necessary to include all minor stops of a network in any particular problem. For example, if it is required to find a minimum path between a specified origin-destination pair, then those minor stops which are considered as potential access or egress points from the origin or to the destination will be inserted into the basic network to form a detailed network. This will not increase a great deal the number of nodes in the network but will enable more accurate modelling.

If you decide to use this option of network generation, then your data base must be further expanded by the coding work which will be described in this section.

3.2 Running Program [Stpcod]

When coding minor stops by this program, you will need to have a network diagram and a map showing the location of these stops. The coding process is iterative and in each iteration, you specify one link and code the minor stops located within this link. When running this program , you will be prompted to enter the following information:

(a) is there an existing STOP file (Y/N)?
(b) enter A-node of link specified,
(c) enter B-node of link specified,

If the answer to (a) is (Y), then the program will check whether this link has been coded previously, if not, or if the answer to (a) is (N), then you will be prompted to enter:

(d) number of stops within the specified link,
(e) the distance of each stop from the A-node expressed as a percentage of the link distance,
(f) number of routes using this set of stops, and
(g) the name of each route.

If the answer to (a) is (Y) and previous record(s) of the same link is found, then the terminal will show previously recorded information, and you will have the options of either changing previous records or adding new records to the file.

If different routes on a link do not use the same set of stops, then different stop and route records have to be coded for these links. Stop records are uni-directional because they record the stops used by routes moving from the A-node to the B-node of the link and not in the reverse direction.

The output from the program consists of two files: STOP which contains stop records and ROUTE which contains route records.

SECTION 4

NETWORK BUILDING

SECTION 4 NETWORK BUILDING

4.1 Introduction

The three previous sections are concerned with coding various informations of the transit system to establish a database. This section will describe various types of information about programs for building different types of network descriptions from this database.

Some of these programs will only require the coding work described in Section 1, while others will require the coding work described in Sections 2 and/or 3 as well.

4.2 Building a Basic Network

A basic network is one which consists of links connecting nodes which have been coded as major stops. It can be built by running [Selink]. This program only requires the timetable information which can be coded as described in Section 1.

When running [Selink], you will be prompted to enter the following information:

(a) network identification, N9
(b) start time of selected period,
(c) finish time of selected period,
(d) smallest node number in network, and
(e) largest node number in network.

As the program generates a transit network with schedules, it is necessary in (b) and (c) to specify the selected period for inclusion of schedules.

For computation efficiency, the program uses a set of node numbers which is different from the coded node numbers. It is therefore necessary to input the information in (d) and (e).

The input file to the program is TIMETBLN9. The output files are LINKTN9 (a network file) and NODEIDN9 (a node number identification file).

4.3 Building a Basic Network with Legs

A basic network as described in Section 4.2 contains line connections between nodes which are major stops within the transit system. This network can be used to generate paths between these nodes but not between other points. When it is required to generate paths in which the origin and/or destination are not nodes, a network with legs should be built.

A leg is defined as an access link between a point and a node of the basic network. A basic network with legs can be built by running [Nlegl].

When running [Nlegl], you will be prompted to enter the following information:

(a) network identification, N9
(b) Melway map code of origin,
(c) access mode from origin,
(d) Melway map code of destination,
(e) access mode to destination, and
(f) option to generate detailed network (Y/N)

if answer to (f) is (N) then you will be prompted to enter:

(g) new network identification, M9

else, the user will be asked to proceed to run [Nleg2].

In (c) and (e) the choice of access mode can be walk, cycle or car at travelling speeds of 5.5, 10 and 30 km/hr respectively.

In (f), if only basic network is required, the input is (N), and (g) specifies the identification for the new network. The method of building a detail network with legs will be described in the next section.

The input files to [Nlegl] are NODEIDN9, LINKTN9, MELCODE and XYCODEN9. The first two files are output from [Selink] while the last two files are output from programs described in Section 2.

The output files from [Nlegl] are LEGl, NODEIDM9, LINKTM9, and XYCODEM9.

Compared with Network N9, Network M9 contains two additional nodes, the origin node and the destination node. These two nodes are assigned numbers following from the largest node number used in Network N9. For example, if the largest node number used in Network N9 is 350, then the origin node number will be 351 and the destination node will be 352.

4.4 Building a Detail Network with Legs

A detailed network is defined as a basic network with some minor stops added. These minor stops are those which are located close to either the origin or destination and hence are potential access/egress points of the network.

The procedure is first to run [Nleg1] as described in Section 4.3 and then to run [Nleg2] and [Linkad].

When running [Nleg2] or [Linkad], you will be prompted to enter the following information:

(a) old network identification, N9 and
(b) new network identification, D9.

The input files to [Nleg2] are NODEIDN9, XYCODEN9, LEG1 and STOP. and the output files are LEG2, NODEIDD9 and XYCODED9.

The input files to [Linkad] are NODEIDD9, LINKTN9, STOP, ROUTE and the output file is LINKTD9.

The two files STOP and ROUTE are obtained from [Stpcod] as described in Section 3.

Additional nodes formed in Network D9 are assigned with numbers following the number assigned to the destination node. For example, if the largest node number in the basic network is 350, then origin node will be 351, the destination node 352 and any other additional nodes formed will have numbers starting from 353.

4.5 Network File Sorting

The previous sub-sections have shown how different types of network files can be generated. These files all contain link records written with the same format. The A-node and B-node of a link is stored from the 2nd to 8th position of a line record. Before these files can be used as input to programs for minimum paths analysis (these programs will be described in Section 5), it is necessary to sort the A-nodes and B-nodes of link records in ascending order.

No program has been written for network sorting, however, most computer systems should already have this facility installed.

Network files which are not sorted have names with a letter T placed before the network identification code. Thus LINKTN9 or LINKTM9 are unsorted files. After sorting, the network file should be given a name with a letter S instead of T placed before the network identification code. Thus, LINKSN9 or LINKSM9 are sorted network files.

4.6 Building a Network with Reversed Timetables

For some programs used in minimum path analysis, it is necessary to generate a network with reversed timetables. This can be produced by [Rvselink].

When running [Rvselink], you will be prompted to enter the following information:

(a) network identification, N9

The input network file is LINKSN9 and the output network file is RVLINKTN9.

SECTION 5

SOLVING MINIMUM PATH PROBLEMS

SECTION 5 SOLVING MINIMUM PATH PROBLEMS

5.1 Introduction

This section describes some programs developed for finding minimum paths in transit networks.

Minimum path problems can be classified into different types, for example, the criteria for determining the minimum path can be either total trip time or weighted trip time. In the first case, the minimum path is the quickest path. In the second case, the minimum path is the optimal path which minimizes a weighted trip time function consisting of walking time, waiting time, in-vehicle time and a time penalty added for the requirement of a transit user to change routes. The weightings applied to the various components of the weighted time function are supplied by the program user.

Furthermore, a minimum path can be based either on a given earliest starting time or latest arrival time.

Another classification of minimum path problems is the number of origins and destinations which are included in these problems.

The following is a list of the types of problems and programs that can be used for solving them.

Problem type	Programs
1. Quickest path from one origin to one or all destinations, with a given starting time at the origin.	[Dij],[Pripth]
2. Quickest path from one or all origins to one destination, with a given arrival time at the destination.	[Rvdij],[Pripth]
3. Optimal path from one origin to one destination, with a given starting time at the origin.	[Spath]
4. Optimal path from one origin to one destination, with a given arrival time at the destination.	[Apath]

5.2 Solving Type 1 Problems

This type of problems can be solved by running [Dij]. When running this program, you will be prompted to enter the following information:

(a) network identification N9,
(b) origin node number,
(c) trip start time, and
(d) destination node number (optional).

If a destination number is specified in (d), then the program will find the quickest path from the origin to this destination. However, if a number -1 is entered in (d), then the program will find paths from the origin to all other nodes in the network.

Different types of networks which are generated by methods described in Section 4 can be used as input to [Dij]. For example, a basic network or detail network, network with legs or without legs can all be used as input to this program.

5.3 Solving Type 2 Problems

This type of problems can be solved by running [Rvdij]. When running this program you will be prompted to enter the following information:

(a) network identification N9,
(b) destination node number,
(c) arrival time at the destination, and
(d) origin node number (optional).

If an origin number is specified in (d), then the program will find minimum paths from this origin to the destination. However, if a number -1 is entered in (d), then the program will find paths from all nodes in the network to the destination.

5.4 Solving Type 3 Problems

This type of problems can be solved by running program [Spath]

When running [Spath], you will be prompted to enter the following information:

(a) network identification N9,
(b) origin node number,
(c) start time from origin,
(d) destination node number, and
(e) weightings for weighted time function W1,W2,W3.
(f) print out minimum path tree for inspection (Y/N) ?

The following weighted time function is assumed in the program:

Weighted Trip time = (in-vehicle time) + W1 (walking time) + W2 (waiting time) +W3 (number of route changes)

W1 and W2 can be integer or real numbers while W3 is an integer value denoting the time penalty in minutes added to each route change.

[Spath] finds an optimal path from the specified origin to destination with a given starting time and this path is stored in file PATH.

5.5 Solving Type 4 Problems

This type of problems can be solved by program [Apath]

When running [Apath], you will be prompted to enter the following information:

(a) network identification N9,
(b) destination node number,
(c) latest arrival time at destination,
(d) origin node number,
(e) weightings for weighted time function W1,W2,W3.
(f) print out minimum path tree for inspection (Y/N) ?

The weighted time function used is the same as those used in the previous section.

[Apath] finds an optimal path from the specified origin to destination with a given latest arrival time and this path is stored in file PATH.

5.6 Printing Out Minimum Paths

In problems type 1 and 2, where there can be more than one origin or destination specified, a tree table rather than individual paths are output from the program. This tree table is stored in file FOR004.

Individual paths can be printed out for inspection by running [Pripth]. When running this program, you will be prompted to enter the following information:

(a) origin and destination node numbers.

SECTION 6

ZONAL NETWORK MODELLING

SECTION 6 ZONAL NETWORK MODELLING

6.1 Introduction

Traffic zones are often used in network models developed for transportation studies. Programs described in this section will enable you to build networks with connections to traffic zones. Using these networks, zonal trip time matrices based on the quickest travel time between zone centroids can be constructed. The weighted trip time between zones can can also be obtained but these types of problems are described in Section 8.

6.2 Building a Network with Connections to Traffic Zones.

First run [Zone], this program is used for coding the access connections between zone centroids and nodes in the network. When running this program, you will be prompted to enter the following information:

(a) network identification N9,

and for each access link in turn enter:

(b) connecting zone number (end with 0),
(c) connecting node number and link time in minutes.

In (b), zone numbers should start from 1, be consecutive and not larger than 100. All access links are assumed to be two way.

The data coded in [Zone] is output to a file called ZONETMN9.

Next copy the timetable data which you wish to include into the network to a file called TIMETBLN9 and then run [Selinkz] to build the network. When running this program, you will be prompted to enter the following information:

(a) network identification N9,
(b) start time of selected period,
(c) finish time of selected period,
(d) largest zone number in network,
(e) smallest node number in network, and
(f) largest node number in network.

This program is similar to [Selink] described in Section 4.2 except that there is one more input file to the program which is ZONETMN9. The network generated is a sketch network, which means that only essential nodes are selected as nodes in the network.

6.3 Building a Zonal Trip Time Matrix

A zonal trip time matrix based on the quickest trip time from origin zone to destination zone can be obtained by running [Dijz]. When running this program, you will be prompted to enter the following information:

(a) network identification number, N9
(b) do you wish to back-track quickest paths (Y/N) ?,and
(c) start time from the origin zone.

The output from the program is QUKTIMEN9 which stores the zonal quickest trip time matrix. If the answer to (b) is Y, then a second file called PATHN9 will be output which stores quickest paths between zone pairs.

SECTION 7

ASSIGNMENT MODELLING

SECTION 7 ASSIGNMENT MODELLING

7.1 Introduction

This section describes two assignment sub-models. The first is called a multi-interval assignment sub-model. It can be used to assign a number of zonal flow matrices of successive time intervals onto a zonal transit network. The second is called a dynamic assignment sub-model. It can be used to assign a number of station to station flow matrices of successive time intervals onto a basic transit network.

7.2 Multi-Interval Assignment

A zonal transit network is required. The network file can be built by the method as described in Section 6.

The next step is to code an input file containing a set of multi-internal zonal flow matrices. An example of a set of multi-interval flow matrices that can be used as input is shown in Table A.7 of Appendix A. The file containing this data is called FLOWMATFI. The last two digits of the file name FI is an identification code for the particular set of flow matrices used. The file contains the following records:

(a) number of multi-interval flow matrices in the file, N

(b) start time of first time interval (minutes); start time of successive time intervals (minutes); and end time of last time interval (minutes),

then for the next Z records:

(c) the flow contained in each row of the flow matrix.

where Z is the number of zones in the network and each row of the flow matrix correspond to one zone in the network arranged in ascending order of zone numbers.

The model is described as multi-interval because it repeats the assignment procedure for several sequential time intervals. The duration of each time interval can be variable and specified in FLOWMATFI. There are two stochastic processes in the model, the first is the selection of a trip starting time within each time interval and the second is the selection of coefficients for the weighted time function. The first requires a set of ordinary random numbers while the second requires a set of random numbers which are normally distributed. The required data is obtained by running [Rangen]. When running this program, you will be prompted to enter the following information:

(a) an arbitrary 4 digit number used for random number generation,

(b) the mean and standard deviation of the weighting coefficient for walking time,

(c) the mean and standard deviation of the weighting coefficient for waiting time,

(d) the mean and standard deviation of the weighting coefficient for route changes.

[Rangen] outputs the required information in two files: RAND and WEIFUN.

Next run the main assignment program called [Intass]. When running this program, you will be prompted to enter the following information:

(a) network identification number N9,

(b) flow matrix identification number FI,

(c) iteration number NI, and

(d) minimum flow for optimal path assignment.

The assignment path used is normally the optimal path from origin zone to destination zone based on a user supplied weighted time function. However, if the flow to be assigned to this path is less than a value enter in (d) above, then a quickest path is used for assignment. The reason is that it is much quicker to find the quickest path rather than the optimal path and this procedure would increase the efficiency of the program without much reduction in accuracy.

The assignment results from [Intass] show the passenger flow along each link of the network during different time intervals. This information is output to a file called LINKFLOWNI. The other output file from [Intass] is PATHNI which contains all the assignment paths generated in the program.

For higher accuracy, [Rangen] and [Intass] can be run for a number of times, each run is then given an identification code of NI which is input fr m (c) when running [Intass]. The output files of this program are then appended with this identification code.

The assignment results obtained from several iterations can be averaged by running [Average]. When running this program, you will be prompted to enter the following information:

(a) flow matrix identification number, FI,

(b) name of link flow file containing assignment results of any one iteration (end with STOP).

(c) name assigned to output link flow file containing average assignment results.

The assignment results output from [Intass] is in link flow format, this can be converted to timetable format by running

[Intload]. When running this program, you will be prompted to enter the following information:

(a) network identification N9,

(b) flow matrix identification number, and

(b) name of input link flow file.

The results are output to a file called ILOAD.

7.3 Dynamic Assignment

This model assigns station to station flows onto a basic transit network. The method of building a basic transit network file has been described in Section 4.2.

The input flow matrix used is also multi-interval and similar to that used in the previous model. The differences are that node numbers are used instead of zone numbers and it is not necessary to include all nodes in the network into the flow matrix. The name of the file is also called FLOWMATFI. The format adopted is shown in Section A.15 of Appendix A.

An assignment is made by running [Rangen] and [Vehass]. The procedure for running [Rangen] has already been described in the previous section. When running [Vehass], you will be prompted to enter the following information:

(a) flow matrix identification number FI,

(b) network identification number N9,

(c) start time of assignment period,

(d) end time of assignment period,

(e) minimum accumulated flow for assignment.

The model adopts a dynamic assignment procedure. This means that when considering an origin and destination station pair and starting from the beginning of the assignment period, successive optimal paths are generated as time progresses until the end of the assignment period. The flow accumulated at the origin station between the departure time of the nth path and (n+1) path is then assigned to the (n+1)th path. However, in order to control the efficiency of the program an assignment is not made when the flow accumulated during this time is less than that specified in (c) above.

The assignment results from [Vehass] is output to a file called VEHFLOW. The data in this file is vehicle-based and in link format. This means that for each link in the network, the number of passengers assigned to each vehicle passing through the link during the assignment period is known. The other output file is NODEPATH which stores all assignment paths generated.

The assignment results can be converted to timetable format by running [Vehload]. When running this program, you will be prompted to enter the following information:

(a) network identification number N9.

The output file is TBLOAD.

SECTION 8

ACCESSIBILITY MODELLING

SECTION 8 ACCESSIBILITY MODELLING

8.1 Introduction

This section describes programs for generating data for the calculation of transit accessibility. Based on the calculation of the temporal variation of trip time between zones, various types of accessibility measures can be obtained. The programs can be used to calculate trip time based on the following different criteria:

(a) quickest path,

(b) optimal path with minimum weighted time

(c) starting time based, and

(d) arrival time based.

8.2 Start Time Based Accessibility

In start time based accessibility, it is necessary to calculate the trip times of travelling from one zone to other zones when various trip starting times from the origin zone are specified.

Firstly, it is required to obtain the set of starting times at the zone centroid in which the initial waiting time for boarding a transit vehicle is zero. This is obtained by running [Gens]. When running this program, you will be prompted to enter the following information:

(a) network identification number, and

(b) origin zone number.

The list of starting times are output to file STIM.

Secondly, run program [Access] to obtain the trip times when travelling from the origin zone to all other destination zones at the various starting times generated. Both the quickest time and minimum weighted time will be calcuated. When running [Access], you will be prompted to enter the following information:

(a) network identification number,

(b) weightings for walk, wait and route change, and

(c) origin zone number

The program outputs quickest trip times to file QTIMS and optimal trip times to file OTIMS. The optimal paths generated are output to file PATHS.

The trip times obtained from the above program correspond to those with optimal starting times, hence these data would represent the troughs of a trip time verses starting time graph such the one shown in Figure A.6 of Appendix A. However, in order to plot a complete graph, it is necessary to plot the crests of the graph as well. The complete set of data required for plotting quickest trip time or weighted trip time verses starting time can be obtained by running [Plots], when running this program, you will be prompted to enter the following information:

(a) origin and destination zone,

(b) quickest or weighted time (Q/W)

The data is output to file called DSETS.

8.3 Arrival Time Based Accessibility

In arrival time based accessibility, it is required to calculate the trip times of travelling from one zone to another when various latest arrival times are specified. The first step is to calculate for a particular destination zone various arrival times in which the waiting time at the destination is zero. This is obtained by running [Gena]. When running [Gena], you will be prompted to enter the following information:

(a) network identification number, and

(b) destination zone number.

The list of arrival times are output to file ATIM. Next run [Accessa] to obtain the trip times when travelling from various origin zones to the specified destination zone with the list of arrival times generated. Both the quickest and weighted trip times will be calculated. When running [Accessa], you will be prompted to enter the following information:

(a) network identification number,

(b) weightings for walk, wait and route change, and

(c) destination zone number.

The program outputs quickest trip times to file QTIMA, minimum weighted trip times to file OTIMA and optimal paths to file PATHA.

The data for plotting variations of quickest trip times and optimal trip times with arrival time can be obtained by running [Plota]. When running this program, you will be prompted to enter the following information:

(a) origin zone and destination zone,

(b) quickest or weighted trip time (Q/W).

The data is output to file DSETA.

APPENDIX A

SAMPLE SESSION

APPENDIX A: A SAMPLE SESSION

A.1 Introduction

A simple example will be used to illustrate the applications of programs included in this package.

The transit network used is shown in Figure A.1 and timetables of routes operating in the network are shown in Table A.1.

A.2 Timetable Coding

Timetables can be coded by running [Tblcode], an example of using this program to code the timetable of route A3 is shown below:

```
RUN TBLCODE
 ENTER ROUTE NUMBER (END WITH 999) ? A3
 ENTER NODE NUMBER     1 ?(END WITH 0)  11
 ENTER NODE NUMBER     2 ?(END WITH 0)  12
 ENTER NODE NUMBER     3 ?(END WITH 0)  16
 ENTER NODE NUMBER     4 ?(END WITH 0)  18
 ENTER NODE NUMBER     5 ?(END WITH 0)  0
 NODE NUMBER IS    11 ENTER TIME SCHEDULE (HR,MIN)? 7,2
 NODE NUMBER IS    12 ENTER TIME SCHEDULE (HR,MIN)? 7,12
 NODE NUMBER IS    16 ENTER TIME SCHEDULE (HR,MIN)? 7,2
 NODE NUMBER IS    18 ENTER TIME SCHEDULE (HR,MIN)? 7,32
 ENTER START TIME OF RUN NUMBER    1(END WITH -1,0) 7,2
 ENTER START TIME OF RUN NUMBER    2(END WITH -1,0) 7,27
 ENTER START TIME OF RUN NUMBER    3(END WITH -1,0) 7,52
 ENTER START TIME OF RUN NUMBER    4(END WITH -1,0) 8,17
 ENTER START TIME OF RUN NUMBER    5(END WITH -1,0) 4,42
```

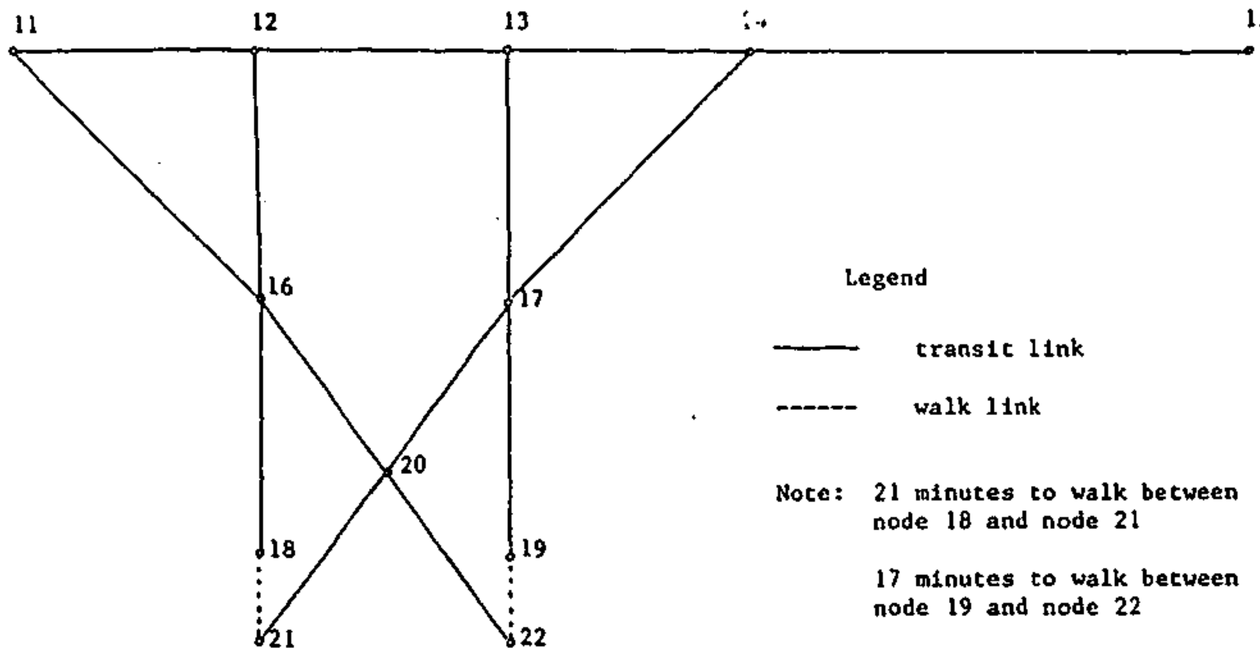


Figure A.1 Transit network 2

Table A-1 Route timetables

Route A1

Stop	11	12	13	14	15
	7:00	7:10	7:20	7:30	7:50
	7:15	7:25	7:35	7:45	8:05
	7:30	7:40	7:50	8:00	8:20
	7:45	7:55	8:05	8:15	8:35
Depart	8:00	8:10	8:20	8:30	8:50
	8:12	8:22	8:32	8:42	9:02
times	8:24	8:34	8:44	8:54	9:14
	8:36	8:46	8:56	9:06	9:26
	8:48	8:58	9:08	9:18	9:38
	9:00	9:10	9:20	9:30	9:50
	9:15	9:25	9:35	9:45	10:05
	9:30	9:40	9:50	10:00	10:20
	9:45	9:55	10:05	10:15	10:35

Route B1

Stop	15	14	13	12	11
	7:05	7:25	7:35	7:45	7:55
	7:20	7:40	7:50	8:00	8:10
	7:35	7:55	8:05	8:15	8:25
Depart	7:50	8:10	8:20	8:30	8:40
	8:02	8:22	8:32	8:42	8:52
times	8:14	8:34	8:44	8:54	9:04
	8:26	8:46	8:56	9:06	9:16
	8:38	8:58	9:08	9:18	9:28
	8:50	9:10	9:20	9:30	9:40
	9:05	9:25	9:35	9:45	9:55
	9:20	9:40	9:50	10:00	10:10
	9:35	9:55	10:05	10:15	10:25
	9:50	10:10	10:20	10:30	10:40

Route A2

Stop	11	12	13	17	19
	7:05	7:15	7:25	7:35	7:45
	7:25	7:35	7:45	7:55	8:05
	7:45	7:55	8:05	8:15	8:25
Depart	8:05	8:15	8:25	8:35	8:45
	8:25	8:35	8:45	8:55	9:05
times	8:45	8:55	9:05	9:15	9:25
	9:05	9:15	9:25	9:35	9:45
	9:25	9:35	9:45	9:55	10:05
	9:45	9:55	10:05	10:15	10:25

Route B2

Stop	19	17	13	12	11
	7:05	7:15	7:25	7:35	7:45
	7:25	7:35	7:45	7:55	8:05
	7:45	7:55	8:05	8:15	8:25
Depart	8:05	8:15	8:25	8:35	8:45
	8:25	8:35	8:45	8:55	9:05
times	8:45	8:55	9:05	9:15	9:25
	9:05	9:15	9:25	9:35	9:45
	9:25	9:35	9:45	9:55	10:05
	9:45	9:55	10:05	10:15	10:25

Route A3

Stop	11	12	16	18
	7:02	7:12	7:22	7:32
	7:27	7:37	7:47	7:57
Depart	7:52	8:02	8:12	8:22
	8:17	8:27	8:37	8:47
times	8:42	8:52	9:02	9:12
	9:07	9:17	9:27	9:37
	9:32	9:42	9:52	10:02
	9:57	10:07	10:17	10:27

Route B3

Stop	18	16	12	11
	7:02	7:12	7:22	7:32
	7:27	7:37	7:47	7:57
Depart	7:52	8:02	8:12	8:22
	8:17	8:27	8:37	8:47
times	8:42	8:52	9:02	9:12
	9:07	9:17	9:27	9:37
	9:32	9:42	9:52	10:02
	9:57	10:07	10:17	10:27

Route A4

Stop	21	20	17	14
	7:04	7:14	7:24	7:39
	7:19	7:29	7:39	7:54
	7:34	7:44	7:54	8:09
	7:49	7:59	8:09	8:24
Depart	8:04	8:14	8:24	8:39
	8:19	8:29	8:39	8:54
times	8:34	8:44	8:54	9:09
	8:49	8:59	9:09	9:24
	9:04	9:14	9:24	9:39
	9:19	9:29	9:39	9:54
	9:34	9:44	9:54	10:09
	9:49	9:59	10:09	10:24

Route B4

Stop	14	17	20	21
	7:05	7:20	7:30	7:40
	7:20	7:35	7:45	7:55
Depart	7:35	7:50	8:00	8:10
	7:50	8:05	8:15	8:25
times	8:05	8:20	8:30	8:40
	8:20	8:35	8:45	8:55
	8:35	8:50	9:00	9:10
	8:50	9:05	9:15	9:25
	9:05	9:20	9:30	9:40
	9:20	9:35	9:45	9:55
	9:35	9:50	10:00	10:10

Table A.1 Route timetables (cont.)

Route A5

Stop	22	20	16	11
	7:04	7:14	7:24	7:39
	7:19	7:29	7:39	7:54
	7:34	7:44	7:54	8:09
	7:49	7:59	8:09	8:24
Depart	8:04	8:14	8:24	8:39
	8:19	8:29	8:39	8:54
times	8:34	8:44	8:54	9:09
	8:49	8:59	9:09	9:24
	9:04	9:14	9:24	9:39
	9:19	9:29	9:39	9:54
	9:34	9:44	9:54	10:09
	9:49	9:59	10:09	10:24

Route B5

Stop	11	16	20	22
	7:05	7:20	7:30	7:40
	7:20	7:35	7:45	7:55
Depart	7:35	7:50	8:00	8:10
	7:50	8:05	8:15	8:25
times	8:05	8:20	8:30	8:40
	8:20	8:35	8:45	8:55
	8:35	8:50	9:00	9:10
	8:50	9:05	9:15	9:25
	9:05	9:20	9:30	9:40
	9:20	9:35	9:45	9:55
	9:35	9:50	10:00	10:10

```
 ENTER START TIME OF RUN NUMBER    6(END WITH -1,0) 9,7
 ENTER START TIME OF RUN NUMBER    7(END WITH -1,0) 9,32
 ENTER START TIME OF RUN NUMBER    8(END WITH -1,0) 9,57
 ENTER START TIME OF RUN NUMBER    9(END WITH -1,0) -1,0
 ENTER ROUTE NUMBER (END WITH 999) ? WAL
 ENTER WALK LINK (ANODE,BNODE,LTIME) END WITH -1,1,1?18,21,21
 ENTER WALK LINK (ANODE,BNODE,LTIME) END WITH -1,1,1?19,22,17
 ENTER WALK LINK (ANODE,BNODE,LTIME) END WITH -1,1,1
 ENTER ROUTE NUMBER (END WITH 999) ? 999
PROGRAM TERMINATED
```

An explanation of how the timetable data should be entered in the above program is given in Section 1.6 of this manual.

You probably have noticed that there are two errors in the data entry, now you can run [Tblcheck] to check the data which has been coded:

```
RUN TBLCHECK
 ERROR IN PROFILE OF ROUTE A3  BETWEEN STOP NO.    2
 AND STOP NO.    3
 ROUTE NUMBER A3  NUMBER OF NODES =    4
                  NUMBER OF RUNS =    8
 ERROR IN NO.    4  START TIME OF ROUTE A3
PROGRAM TERMINATED
```

The data coded from [Tblcode] is contained in file FOR003. Your next step is to edit this file to correct the errors which has been detected. The required corrections are illustrated below.

FOR003

A3

11		
12		
16		
18		
0		
7	2	
7	12	
7	2	22
7	32	
7	2	
7	27	
7	52	
8	17	
4 8	42	
9	7	
9	32	
9	57	
-1	0	

WAL

18	21	21
19	22	17
-1	1	1

After correcting the errors in FOR003, You can then run [Tblprint] to reconstruct the timetable from coded data for further checking. The output file is FOR005:

FOR005

ROUTE NUMBER A3

11	12	16	18
7 2	7 12	7 22	7 32
7 27	7 37	7 47	7 57
7 52	8 2	8 12	8 22
8 17	8 27	8 37	8 47
8 42	8 52	9 2	9 12
9 7	9 17	9 27	9 37
9 32	9 42	9 52	10 2
9 57	10 7	10 17	10 27

WALK LINKS

FROM NODE	TO NODE	TIME
18	21	21
19	22	17

When you are satisfied that the data coded is correct, the data coded in FOR003 is transfered to a permanent file. First, you have to give the network an identification code. Suppose the network code is 2, then the permanent timetable data file should be called TIMETBL2, and the command for transfering data from one file to another normally is:

```
COPY TIMETBL2+FOR003 , TIMETBL2
```

This procedure should be repeated until all timetables operating in the network are coded.

A.3 Building a Basic Network

After finishing coding all timetables you may proceed to build a basic transit network using the procedure described below:

```
RUN SELINK
 ENTER NETWORK IDENTIFICATION NUMBER?  2
 ENTER START TIME OF SELECTED PERIOD (HR,MIN)? 7,0
 ENTER FINISH TIME OF SELECTED PERIOD (HR,MIN)? 10,0
 ENTER SMALLEST NODE NUMBER? 11
 ENTER LARGEST NODE NUMBER? 22
PROGRAM TERMINATED
```

The above program generates a basic transit network containing transit schedules operating within the time period from 7 am to 10 am. The program outputs a node identification file called NODEID2 and a link record file called LINKT2. The file LINKT2 should then be sorted and written to a new file called LINKS2. A usual command for doing this is:

```
SORT/KEY=(POSITION:2,SIZE:8) LINKT2, LINKS2
```

The contents of LINKT2 is shown in Table A.2.

A.4 Solving Type 1 Minimum Path Problems in a Basic Network

As explained in Section 5.1, a type 1 problem finds the quickest paths from one origin to all destinations with a given starting time. The procedure is shown below:

```
RUN DIJ
 ENTER NETWORK IDENTIFICATION NUMBER? 2
 ENTER ORIGIN NODE NUMBER? 21
 ENTER START TIME (HR,MIN)? 7,15
 ENTER DESTINATION NUMBER (-1) IF NOT REQUIRED? -1
PROGRAM TERMINATED
```

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

1	2	10	13	420	435	450	465	480	492	504	516	528	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0A1
2	3	10	13	430	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0A1
3	4	10	12	440	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0A1
4	5	20	11	450	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0A1
5	4	20	13	425	440	455	470	482	494	506	518	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0B1
4	3	10	12	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0	0B1
3	2	10	11	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B1
2	1	10	10	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B1
1	2	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A2
2	3	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A2
3	7	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A2
7	9	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A2
9	7	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B2
7	3	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B2
3	2	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B2
2	1	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B2
1	2	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A3
2	6	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A3
6	8	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A3
8	5	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B3
6	2	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B3
2	1	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B3
11	10	10	12	424	439	454	469	484	499	514	529	544	559	574	589	0	0	0	0	0	0	0	0	0	0	0	0	0A4
10	7	10	12	434	449	464	479	494	509	524	539	554	569	584	599	0	0	0	0	0	0	0	0	0	0	0	0	0A4
7	4	15	11	444	459	474	489	504	519	534	549	564	579	594	0	0	0	0	0	0	0	0	0	0	0	0	0	0A4
4	7	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B4
7	10	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B4
10	11	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B4
12	10	10	9	424	439	454	469	484	499	514	529	544	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A5
10	6	10	9	434	449	464	479	494	509	524	539	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A5
6	1	15	9	444	459	474	489	504	519	534	549	564	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A5
1	6	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B5
6	10	15	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B5
10	12	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B5
9	12	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
12	9	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
8	11	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
11	9	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL

Table A·2 LINKT2

The above procedure finds quickest paths from origin node 21 to all other nodes in the network at a starting time of 7:15 am. If a destination node number is specified when running the program, then only the quickest path from the origin to this destination will be found. If a destination node number of -1 is specified, then quickest paths to all destinations will be found. The program outputs a quickest path tree in the file below called FOR004:

```
 12
2
     11        16         7 54    A5
     12        11         8 10    A1
     13        17         8  5    B2
     14        17         7 54    A4
     15        14         8 20    A1
     16        20         7 39    A5
     17        20         7 39    A4
     18        21         7 36    WAL
     19        22         7 57    WAL
     20        21         7 29    A4
     21         0         7 15
     22        20         7 40    B5
```

The quickest path from node 21 to any other node in the network can be back-tracked and printed out by the following procedure:

```
RUN PRIPTH
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 21,11
      FROM          TO             ARRIVE       ROUTE
      NODE          NODE           TIME         USED
                    21              7 15
       21           20              7 29        A4
       20           16              7 39        A5
       16           11              7 54        A5
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 21,14
      FROM          TO             ARRIVE       ROUTE
      NODE          NODE           TIME         USED
                    21              7 15
       21           20              7 29        A4
       20           17              7 39        A4
       17           14              7 54        A4
       14           15              8 20        A1
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 21,13
      FROM          TO             ARRIVE       ROUTE
      NODE          NODE           TIME         USED
                    21              7 15
       21           20              7 29        A4
       20           17              7 39        A4
       17           13              8  5        B2
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 21,12
      FROM          TO             ARRIVE       ROUTE
      NODE          NODE           TIME         USED
                    21              7 15
       21           20              7 29        A4
       20           16              7 39        A5
       16           11              7 54        A5
       11           12              8 10        A1
```

```
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 21,22
     FROM          TO                ARRIVE      ROUTE
     NODE          NODE              TIME        USED
                   21                 7 15
      21           20                 7 29        A4
      20           22                 7 40        B5
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 0,0
PROGRAM TERMINATED
```

A.5 Solving Type 2 Minimum Path Problems in a Basic Network

In a type 2 problem, it is required to find the quickest paths from all nodes in a network to one destination. The procedure is explained as follows:

```
RUN RVSELINK
 ENTER NETWORK IDENTIFICATION NUMBER? 2
PROGRAM TERMINATED
```

The above program generates a temporary network file with reversed timetables called RVLINKT2 which is shown in Table A.3. This file should then be sorted and the records written to a new file called RVLINKS2. A typical sorting command is as follows:

```
SORT/KEY=(POSITION:2, SIZE:8) RVLINKT2, RVLINKS2

RUN RVDIJ
 ENTER NETWORK IDENTIFICATION NUMBER? 2
 ENTER DESTINATION NODE NUMBER? 11
 ENTER ARRIVAL TIME (HR,MIN)? 8,20
 ENTER ORIGIN NODE NUMBER (-1) IF NOT REQUIRED? -1
PROGRAM TERMINATED
```

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
   2   1  10  13245024352420240523902378236623542342233023152300 2285  0  0  0  0  0  0  0  0  0  0  0  OA1
   2   1  10   824482423239823732348232322982273  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA3
   2   1  10   92445242524052385236523452325230522 85  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB5
   6   1  15  1124402425241023952380236523502335232023052290  0  0  0  0  0  0  0  0  0  0  0  0  0  OA2
   1   2  10   72428240323782353232823052278  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB3
   1   2  10  1024052390237523602348233623242312230022 85  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB1
   1   2  10   8241523952375235523352315229522 75  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB2
   3   2  10   924352415239523752355233523152295 2275  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA2
   3   2  10  132440242524102395238023682356234423322320230522902275  0  0  0  0  0  0  0  0  0  0  0  OA1
   6   2  10   72438241323882363233823132288  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA3
   2   3  10   82425240523852365234523252305 2285  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB2
   2   3  10  11241524002385237023582346233423222310229 52280  0  0  0  0  0  0  0  0  0  0  0  0  0  OB1
   4   3  10  1224302415240023852370235823462334232223102295 2280  0  0  0  0  0  0  0  0  0  0  0  0  OA1
   7   3  10   82425240523852365234523252305 2285  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA2
   3   4  10  122425241023952380236823562344233223202305229 02275  0  0  0  0  0  0  0  0  0  0  0  0  OB1
   5   4  20  1124102395238023652350233823262314230222902275  0  0  0  0  0  0  0  0  0  0  0  0  0  OA1
   7   4  15  1124402425241023952380236523502335232023052290  0  0  0  0  0  0  0  0  0  0  0  0  0  OB4
   4   5  20  1324352420240523902378236623542342233023152300228 52270  0  0  0  0  0  0  0  0  0  0  0  OB1
   1   6  15   9242124062391237623612346233 123162301  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA5
   2   6  10   72438241323882363233623132288  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB3
   8   6  10   72428240323782353232823032278  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA3
  10   6  10  1224302415240023852370235523402325231022952280  0  0  0  0  0  0  0  0  0  0  0  0  0  OB5
   3   7  10   924352415239523752355233523152295 2275  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB2
   4   7  15  1124212406239123762361234623312316230122862271  0  0  0  0  0  0  0  0  0  0  0  0  0  OA4
   9   7  10   8241523952375235523352315229522 75  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA2
  10   7  10  1124302415240023852370235523402325231022952280  0  0  0  0  0  0  0  0  0  0  0  0  0  OB4
   6   8  10   824482423239823732348232322982273  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB3
  11   9  21   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OWAL
   7   9  10   9244524252405238523652345232523052285  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB2
  12   9  17   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OWAL
   6  10  10   924362421240623912376236123462331 2316  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA5
   7  10  10  12243624212406239123762361234623312316230122862271  0  0  0  0  0  0  0  0  0  0  0  0  OA4
  11  10  10  10242024052390237523602345233023152300 2285  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB4
  12  10  10  10242024052390237523602345233023152300 2285  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OB5
   8  11  21   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OWAL
  10  11  10  1224462431241624012386237123562341232623112296 2281  0  0  0  0  0  0  0  0  0  0  0  0  OA4
   9  12  17   1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OWAL
  10  12  10   92446243124162401238623712356234 12326  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  OA5
```

Table A·3   RVLINKT2

The above procedure finds quickest paths from all nodes in the network to destination node 11 with a latest arrival time of 8:20 am. The quickest path tree is written to file FOR004:

```
FOR004
  12
 2
      11          0          8 20
      12         11          8  0      B1
      13         12          7 50      B1
      14         13          7 40      B1
      15         14          7 20      B1
      16         11          7 54      A5
      17         13          7 35      B2
      18         16          7 27      B3
      19         17          7 25      B2
      20         16          7 44      A5
      21         20          7 34      A4
      22         20          7 34      A5
```

The quickest path from all other nodes to node 11 can be traced and printed out by running [Pripth]

```
RUN PRIPTH
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 11,21
     FROM        TO          ARRIVE    ROUTE
     NODE        NODE        TIME      USED
                 11          8 20
      11         16          7 54      A5
      16         20          7 44      A5
      20         21          7 34      A4
```

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 11,18

FROM NODE	TO NODE	ARRIVE TIME	ROUTE USED
	11	8 20	
11	16	7 54	A5
16	18	7 27	B3

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 11,15

FROM NODE	TO NODE	ARRIVE TIME	ROUTE USED
	11	8 20	
11	12	8 0	B1
12	13	7 50	B1
13	14	7 40	B1
14	15	7 20	B1

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 11,22

FROM NODE	TO NODE	ARRIVE TIME	ROUTE USED
	11	8 20	
11	16	7 54	A5
16	20	7 44	A5
20	22	7 34	A5

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 11,19

FROM NODE	TO NODE	ARRIVE TIME	ROUTE USED
	11	8 20	
11	12	8 0	B1
12	13	7 50	B1
13	17	7 35	B2
17	19	7 25	B2

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0) 0,0

PROGRAM TERMINATED

When running the above program for a type 2 problem, it should be bourne in mind that the quickest path is traced in the reverse direction, hence, in the input to [Pripth] at the terminal, the origin node and destination nodes are reversed.

A.6 Solving Type 3 Minimum Path Problems in a Basic Network

Apart from finding quickest paths, you can also find optimal paths with a minimum weighted trip time. The procedure for finding an optimal path from node 21 to node,12 is shown as follows:

```
RUN SPATH
 ENTER NETWORK IDENTIFICATION NUMBER? 2
 ENTER ORIGIN NODE NUMBER? 21
 ENTER START TIME (HR,MIN)? 7,15
 ENTER DESTINATION NODE NUMBER? 12
 ENTER WEIGHTINGS FOR WALK, WAIT, ROUTE CHANGE? 1,1,10
 DO YOU WISH TO PRINT TREE (YES=1,NO=2) 2
PROGRAM TERMINATED
```

When running [Spath], ypu have the option of printing the tree which shows the various routes searched by the program. This procedure is not necessary unless the program is not operating properly and you wish to check the path tracking procedure.

The optimal path is output onto a file called PATH. This file is listed in the following:

PATH

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED
21	7:15	7:15	WAL
18	7:36	7:52	B3
16	8: 2	8: 2	B3
12	8:12		

WEIGHTINGS FOR WALK TIME=1.0, WAIT TIME=1.0,
ROUTE CHANGE= 10 MIN
IN-VEH TIME= 20, WALK TIME= 21, WAIT TIME= 16,
ROUTE CHANGES= 0
JOURNEY TIME= 57 MIN.   WEIGHTED JOURNEY TIME= 57 MIN.

The weightings for walk, wait and route change has been choosen to give preference to walking or waiting rather than changing routes. The reason for choosing this set of weightings is to illustrate that the optimal path can be different from the quickest path.

A.7 Solving Type 4 Minimum Path Problems in a Basic Network

Alternatively, it may be required to find the optimal path from one origin to one destination with a specified latest arrival time. The procedure for finding an optimal path from node 21 to node 11 with a latest arrival time of 8:22 am is shown below:

```
RUN APATH
 ENTER NETWORK IDENTIFICATION NUMBER? 2
 ENTER DESTINATION NODE NUMBER? 11
 ENTER ARRIVAL TIME (HR,MIN)? 8,22
 ENTER ORIGIN NODE NUMBER? 21
```

```
ENTER WEIGHTINGS FOR WALK, WAIT, ROUTE CHANGE? 1,2,5
DO YOU WISH TO PRINT TREE (YES=1,NO=2) 2
PROGRAM TERMINATED
```

The optimal path is output onto a file called PATH. This file is listed in the following:

```
PATH
 NODE  ARRIVE  DEPART  ROUTE
       TIME    TIME    USED
 ----------------------------
  21            7:31   WAL
  18    7:52    7:52   B3
  16    8: 2    8: 2   B3
  12    8:12    8:12   B3
  11    8:22    8:22
 WEIGHTINGS FOR WALK TIME=1.0, WAIT TIME=2.0,
 ROUTE CHANGE= 5 MIN
 IN-VEH TIME= 30, WALK TIME= 21, WAIT TIME= 0,
 ROUTE CHANGES= 0
 JOURNEY TIME= 51 MIN.     WEIGHTED JOURNEY TIME= 51 MIN.
```

A.8 Building a Basic Network with Legs

Building a basic network with legs allow you to solve minimum path problems in which the origins and destinations are not nodes in the network but points defined by map grids. In order to do this, the locations of all nodes in the network have to be coded. Figure A.2 shows the locations of nodes in Network 2 in the form of map grids which are based on a street map of Melbourne pulished by Melway Publishing Pty. Ltd.

Before coding the map grids of nodes, it is necessary to define the location of maps with respect to an assigned x and y axes. This is shown in Figure A.2 and the procedure for coding this information is shown below:

```
RUN MELWAY
 IS THERE AN EXISTING MELCODE FILE (Y/N)? N
 DO YOU WSIH TO CODE A NEW PAGE (Y/N)? Y
 ENTER PAGE CODE? 2F
 ENTER PAGE TYPE
  TYPE=1 FOR 1:20000 SCALE MAPS
  TYPE=2 FOR 1:10000 SCALE MAPS? 2
 ENTER X CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 85
 ENTER Y CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 54
 DO YOU WISH TO CODE A NEW PAGE (Y/N)? Y
 ENTER PAGE CODE? 2G
 ENTER PAGE TYPE
  TYPE=1 FOR 1:20000 SCALE MAPS
  TYPE=2 FOR 1:10000 SCALE MAPS? 2
 ENTER X CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 90
 ENTER Y CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 54
 DO YOU WISH TO CODE A NEW PAGE (Y/N)? N
PROGRAM TERMINATED
```

The coding is continued in another session with [Melway]:

```
RUN MELWAY
 IS THERE AN EXISTING MELCODE FILE (Y/N)? Y
 EXISTING MELCODE FILE
 PAGE PAGE    X     Y
 CODE TYPE
 2F       2   85    54
 2G       2   90    54
```

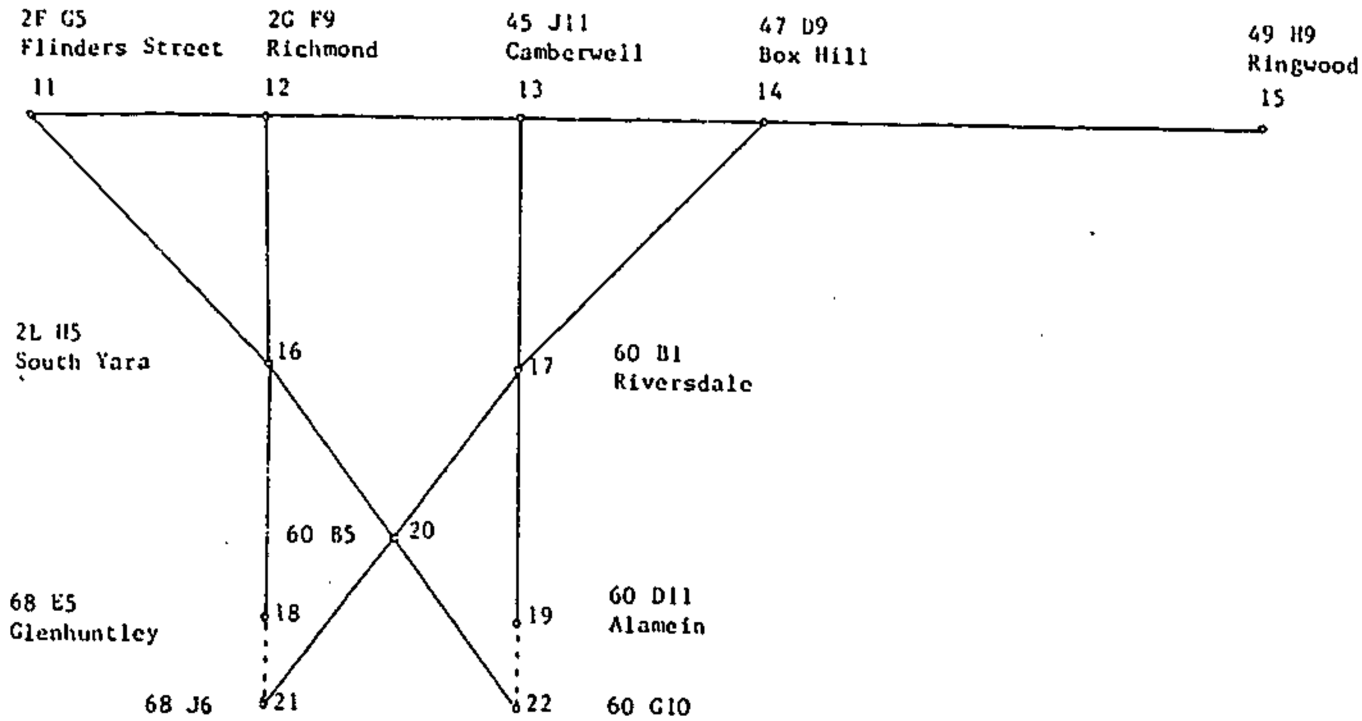


Figure A.2 Map co-ordinates of nodes

```
NEW RECORDS CAN NOW BE ADDED
OR OLD RECORDS AMENDED BY RE-CODING
DO YOU WISH TO CODE A NEW PAGE (Y/N)? Y
ENTER PAGE CODE? 2L
ENTER PAGE TYPE
 TYPE=1 FOR 1:20000 SCALE MAPS
 TYPE=2 FOR 1:10000 SCALE MAPS? 2
ENTER X CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 90
ENTER Y CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 64
DO YOU WISH TO CODE A NEW PAGE (Y/N)? Y
ENTER PAGE CODE? 47
ENTER PAGE TYPE
 TYPE=1 FOR 1:20000 SCALE MAPS
 TYPE=2 FOR 1:10000 SCALE MAPS? 1
ENTER X CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 120
ENTER Y CO-ORDINATE OF TOP LEFT HAND CORNER OF MAP? 48
DO YOU WISH TO CODE A NEW PAGE (Y/N)? N
PROGRAM TERMINATED
```

This procedure should be repeated until all map pages referenced in Figure A.2 have been coded. On completion of coding the contents of MELCODE should be as follows:

MELCODE

	8		
2F	2	85	54
2G	2	90	54
2L	2	90	60
47	1	120	48
45	1	100	48
49	1	130	48
60	1	110	60
68	1	100	72

	10
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
J	8
K	9

The procedure for coding the location of nodes is as shown below:

```
RUN NODCODE
 ENTER LARGEST NODE NUMBER USED IN YOUR NETWORK? 22
 IS THERE AN EXISTING MAPCODE FILE (Y/N)? N
 START CODING NOW
 ENTER NODE NUMBER? 11
 ENTER MELWAY PAGE CODE? 2F
 ENTER CHARACTER CODE? G
 ENTER NUMERAL CODE? 5
 ENTER STATION OR STOP NAME? FLINDER'S STREET
 DO YOU WISH TO CODE ANOTHER NODE (Y/N)? Y
 START CODING NOW
 ENTER NODE NUMBER? 12
 ENTER MELWAY PAGE CODE? 2G
 ENTER CHARACTER CODE? F
 ENTER NUMERAL CODE? 9
 ENTER STATION OR STOP NAME? RICHMOND
 DO YOU WISH TO CODE ANOTHER NODE (Y/N)? Y
 START CODING NOW
```

```
 ENTER NODE NUMBER? 13
 ENTER MELWAY PAGE CODE? 45
 ENTER CHARACTER CODE? J
 ENTER NUMERAL CODE? 10
 ENTER STATION OR STOP NAME? CAMBERWELL
 DO YOU WISH TO CODE ANOTHER NODE (Y/N)? N
PROGRAM TERMINATED
```

You can code any number of nodes during a run of [Nodcode], and the data is recorded in file MAPCODE. Errors in the data can be corrected by re-coding. For example, the following shows how you can correct errors and continue the coding in another run of [Nodcode]:

```
RUN NODCODE
 ENTER LARGEST NODE NUMBER USED IN YOUR NETWORK? 22
 IS THERE AN EXISTING MAPCODE FILE (Y/N)? Y
 START CODING NOW
 ENTER NODE NUMBER? 13
 THIS NODE WAS PREVIOUSLY CODED
 45  J 10 NAME   CAMBERWELL
 DO YOU WISH TO CHANGE THE CODE (Y/N)? Y
 ENTER MELWAY PAGE CODE? 45
 ENTER CHARACTER CODE? J
 ENTER NUMERAL CODE? 11
 ENTER STATION OR STOP NAME? CAMBERWELL
 DO YOU WISH TO CODE ANOTHER NODE (Y/N)? Y
 START CODING NOW
 ENTER NODE NUMBER? 14
 ENTER MELWAY PAGE CODE? 47
 ENTER CHARACTER CODE? D
 ENTER NUMERAL CODE? 9
 ENTER STATION OR STOP NAME? BOX HILL
```

```
DO YOU WISH TO CODE ANOTHER NODE (Y/N)? N
PROGRAM TERMINATED
```

When all nodes in the network have been coded, then file MAPCODE should contain the following records:

```
MAPCODE
   11 2F  G  5 FLINDER'S STREET
   12 2G  F  9 RICHMOND
   13 45  J 11 CAMBERWELL
   14 47  D  9 BOX HILL
   15 49  H  9 RINGWOOD
   16 2L  H  5 SOUTH YARA
   17 60  B  1 RIVERSDALE
   18 68  E  5 GLENHUNTLY
   19 60  D 11 ALAMEIN
   20 60  B  5
   21 68  J  6
   22 60  G 10
```

The next step is to convert the map grids of nodes coded in MAPCODE to xy co-ordinates using the relationship defined in MELCODE. The procedure is as follows:

```
RUN GRID
 ENTER NETWORK IDENTIFICATION NUMBER? 2
PROGRAM TERMINATED
```

The converted xy co-ordinates of nodes are written to file XYCODE2. The contents of this file is shown below:

XYCODE2

11	88	56
12	92	58
13	108	58
14	123	56
15	147	56
16	93	62
17	111	60
18	104	76
19	113	70
20	111	64
21	108	77
22	116	69

At this stage the coding work is completed and you can proceed to specify the locations of the origin and destination and generate a network with legs connecting to these two points.

```
RUN NLEG1
 ENTER OLD NETWORK IDENTIFICATION? 2
 THE PROGRAM WILL SEARCH LEGS FIRST FROM ORIGIN
 THEN FROM DESTINATION
 FIRST ENTER ORIGIN MELWAY CODE
 ENTER PAGE CODE? 68
 ENTER CHARACTER CODE? G
 ENTER NUMERAL CODE? 3
 CONVERTED CO-ORDINATES = 106  74
 CHOOSE THE FOLLOWING ACCESS MODE:
 ENTER (W) FOR WALK, (B) FOR BICYCLE AND (C) FOR CAR? W
 NODE NUMBER FOUND  18  ACCESS TIME  17 MIN
 NODE NUMBER FOUND  21  ACCESS TIME  21 MIN
 NOW ENTER DESTINATION MELWAY CODE
```

```
ENTER PAGE CODE? 2F
ENTER CHARACTER CODE? H
ENTER NUMERAL CODE? 2
CONVERTED CO-ORDINATES =  88    54
CHOOSE THE FOLLOWING ACCESS MODE:
ENTER (W) FOR WALK, (B) FOR BICYCLE AND (C) FOR CAR? W
NODE NUMBER FOUND  11  ACCESS TIME  10 MIN
NODE NUMBER FOUND  12  ACCESS TIME  34 MIN
DO YOU WISH TO INCLUDE MINOR STOPS (Y/N)? N
ENTER NEW NETWORK IDENTIFICATION NUMBER? 21
NODE NUMBER ASSIGNED TO ORIGIN IS   23
NODE NUMBER ASSIGNED TO DESTINATION IS 24
PROGRAM TERMINATED
```

The above program generates a new network with legs connecting the assigned origin and destination. The network has an identification name of 21 and has two additional nodes compared with Network 2.

The program outputs a new node identity file NODEID21, a node location file XYCODE21 and a temporary link file LINKT21. The link file should be sorted in the usual way to produce a permanent link file LINKS21.

Network 21 is shown in Figure A.3 and LINKT21 in Table A.4.

A.9 Solving Minimum Path Problems in a Basic Network with Legs

Having built a basic network with legs, all three types of minimum path problems can be solved in the same way as described in Sections A.4, A.5 and A.6.

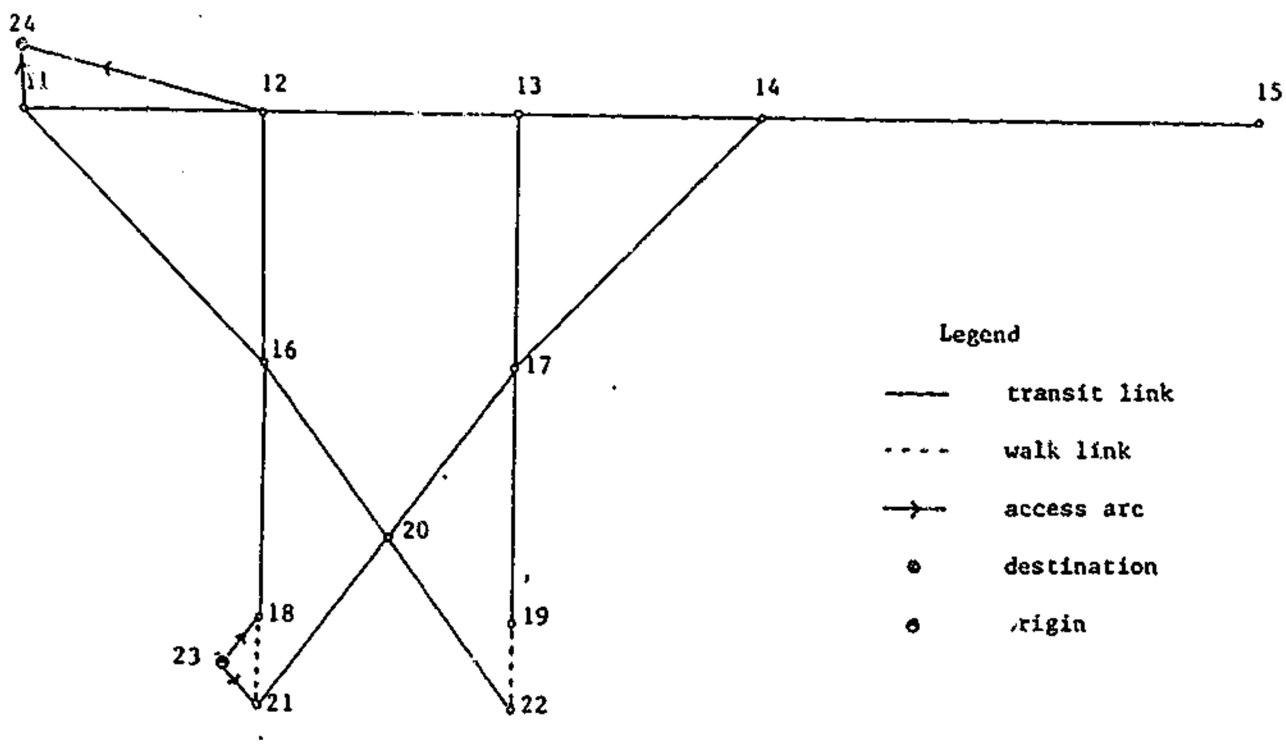


Figure A.3 Transit network 21

12345678901234567890123456789012345678901234567890123456789012345678901234567890

1	2	10	13	420	435	450	465	480	492	504	516	528	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	A 1
2	3	10	13	430	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0	A 1
3	4	10	12	440	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	A 1
4	5	20	11	450	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 1
5	4	20	13	425	440	455	470	482	494	506	518	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	B 1
4	3	10	12	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0	0	B 1
3	2	10	11	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 1
2	1	10	10	465	480	49[illegible]	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 1
1	2	10	9	425	445	46[illegible]	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 2
2	3	10	9	435	455	[illegible]	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 2
3	7	10	8	445	465	[illegible]	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 2
7	9	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 2
9	7	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 2
7	3	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 2
3	2	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 2
2	1	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 2
1	2	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 3
2	6	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 3
6	8	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 3
8	6	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 3
6	2	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 3
2	1	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 3
11	12	10	12	424	439	454	469	484	499	514	529	544	559	574	589	0	0	0	0	0	0	0	0	0	0	0	0	0	A 4
10	7	10	12	434	449	464	479	494	509	524	539	554	569	584	599	0	0	0	0	0	0	0	0	0	0	0	0	0	A 4
7	4	15	11	444	459	474	489	504	519	534	549	564	579	594	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 4
4	7	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 4
7	10	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 4
10	11	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 4
12	10	10	9	424	439	454	469	484	499	514	529	544	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 5
10	6	10	9	434	449	464	479	494	509	524	539	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 5
6	1	15	9	444	459	474	489	504	519	534	549	564	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	A 5
1	4	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 5
6	10	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 5
10	12	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	B 5
9	12	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
12	9	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
8	11	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
11	8	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
13	8	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
13	11	22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
1	14	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL
2	14	35	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	W AL

Table A·4 LINKT21

For example, a type 1 problem can be solved as follows:

```
RUN DIJ
 ENTER NETWORK IDENTIFICATION NUMBER? 21
 ENTER ORIGIN NODE NUMBER? 23
 ENTER START TIME (HR,MIN)? 7,0
 ENTER DESTINATION NODE NUMBER (-1) IF NOT REQUIRED? 24
PROGRAM TERMINATED

RUN PRIPTH
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 23,24
     FROM          TO             ARRIVE      ROUTE
     NODE          NODE           TIME        USED
                    23             7  0
      23            18             7 17        WAL
      18            16             7 37        B3
      16            11             7 54        A5
      11            24             8  4        WAL
 ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0) 0,0
PROGRAM TERMINATED
```

You may wish to continue and try to solve a type 2, 3 or 4 problem. The solution of a type 3 problem is given below:

```
PATH
  NODE  ARRIVE  DEPART  ROUTE
        TIME    TIME    USED
  ----------------------------
   23    7: 0    7: 0    WAL
   18    7:17    7:27    B3
   16    7:37    7:37    B3
   12    7:47    7:47    B3
   11    7:57    7:57    WAL
   24    8: 7
  WEIGHTINGS FOR WALK TIME=2.0, WAIT TIME=2.0,
  ROUTE CHANGE=  5 MIN
  IN-VEH TIME= 30, WALK TIME= 27, WAIT TIME= 10,
  ROUTE CHANGES= 0
  JOURNEY TIME= 67 MIN.     WEIGHTED JOURNEY TIME=104 MIN.
```

A.10 Building a Detailed Network with Legs

A detailed network contains those stops in transit routes which have been omitted in a basic network. For example, Figure A.4 shows the stops in several routes which have been omitted in the basic Network 2. In order to build a detailed network, the first step is to code the locations of these omitted stops by the following procedure:

```
RUN STPCOD
 IS THERE AN EXISTING STOP.DAT FILE (Y/N)? Y
 ENTER A-NODE? 11
 ENTER B-NODE? 16
 ENTER NUMBER OF STOPS? 3
 ENTER PERCENT LINK DISTANCE OF 1 STOP? 25
 ENTER PERCENT LINK DISTANCE OF 2 STOP? 50
 ENTER PERCENT LINK DISTANCE OF 3 STOP? 75
```

```
DO YOU WISH TO CODE A NEW ROUTE LIST (Y/N)? Y
NUMBER OF ROUTES IN LIST? 1
ENTER ROUTE NUMBER OF  1  ROUTE? B5
NUMBER ASSIGNED TO THIS ROUTE LIST IS   1
DO YOU WISH TO CODE ANOTHER LINK (Y/N)? Y
ENTER A-NODE? 16
ENTER B-NODE? 20
ENTER NUMBER OF STOPS? 1
ENTER PERCENT LINK DISTANCE OF 1 STOP? 50
DO YOU WISH TO CODE A NEW ROUTE LIST (Y/N)? N
IF NOT ENTER ROUTE LIST NUMBER USED? 1
DO YOU WISH TO CODE ANOTHER LINK (Y/N)? Y
ENTER A-NODE? 21
ENTER B-NODE? 20
ENTER NUMBER OF STOPS? 2
ENTER PERCENT LINK DISTANCE OF 1 STOP? 30
ENTER PERCENT LINK DISTANCE OF 2 STOP? 60
DO YOU WISH TO CODE A NEW ROUTE LIST (Y/N)? Y
NUMBER OF ROUTES IN LIST? 1
ENTER ROUTE NUMBER OF   1  ROUTE?  A4
NUMBER ASSIGNED TO THIS ROUTE IS  2
DO YOU WISH TO CODE ANOTHER LINK (Y/N)? N
PROGRAM TERMINATED
```

The coding can be continued or previously coded data corrected in another session with [Stpcod]:

```
RUN STPCOD
IS THERE AN EXISTING STOP.DAT FILE (Y/N)? Y
ENTER A-NODE? 11
ENTER B-NODE? 16
THIS RECORDED WAS PREVIOUSLY CODED
A-NODE=  11 B-NODE=  16
```

```
 3 STOPS WERE CODED WITH THE FOLLOWING PERCENT LINK DISTANCE
 25  50  75
ROUTES USING THIS LINK ARE CODED AS
B5
NUMBER ASSIGNED OT THIS ROUTE LIST IS  1
DO YOU WISH TO CHANGE THIS LINK CODE (Y/N)? Y
CHANGE STOP LOCATIONS (Y/N)? Y
ENTER NUMBER OF STOPS? 3
ENTER PERCENT LINK DISTANCE OF 1 STOP? 25
ENTER PERCENT LINK DISTANCE OF 2 STOP? 50
ENTER PERCENT LINK DISTANCE OF 3 STOP? 80
CHANGE LIST OF ROUTE NUMBERS (Y/N)? N
DO YOU WISH TO CODE ANOTHER LINK (Y/N)? Y
ENTER A-NODE? 16
ENTER B-NODE? 11
ENTER NUMBER OF STOPS? 2
ENTER PERCENT LINK DISTANCE OF 1 STOP? 50
ENTER PERCENT LINK DISTANCE OF 2 STOP? 90
DO YOU WISH TO CODE A NEW ROUTE LIST (Y/N)? Y
NUMBER OF ROUTES USED IN LIST (Y/N)? 1
ENTER ROUTE NUMBER OF  1  ROUTE? A5
DO YOU WISH TO CODE ANOTHER LINK (Y/N)? N
PROGRAM TERMINATED
```

All minor stops shown in Figure A.5 have now been coded, the program outputs two files which is shown below:

```
STOP
 4
 11  16  1  3  25  50  80  0  0  0  0  0  0  0
 16  20  1  1  50   0   0  0  0  0  0  0  0  0
 16  11  2  2  50  90   0  0  0  0  0  0  0  0
 21  20  3  2  30  60   0  0  0  0  0  0  0  0
```

```
ROUTE
   3
   1B5
   1A5
   1A4
```

The next step is:

```
RUN NLEG1
 ENTER OLD NETWORK IDENTIFICATION? 2
 THE PROGRAM WILL SEARCH LEGS FIRST FROM ORIGIN
 THEN FROM DESTINATION
 FIRST ENTER ORIGIN MELWAY CODE
 ENTER PAGE CODE? 68
 ENTER CHARACTER CODE? G
 ENTER NUMERAL CODE? 3
 CONVERTED CO-ORDINATES = 106   74
 CHOOSE THE FOLLOWING ACCESS MODE:
 ENTER (W) FOR WALK, (B) FOR BICYCLE AND (C) FOR CAR? W
 NODE NUMBER FOUND  18  ACCESS TIME  17 MIN
 NODE NUMBER FOUND  21  ACCESS TIME  21 MIN
 NOW ENTER DESTINATION MELWAY CODE
 ENTER PAGE CODE? 2F
 ENTER CHARACTER CODE? K
 ENTER NUMERAL CODE? 10
 CONVERTED CO-ORDINATES =  89    58
 CHOOSE THE FOLLOWING ACCESS MODE:
 ENTER (W) FOR WALK, (B) FOR BICYCLE AND (C) FOR CAR? W
 NODE NUMBER FOUND  11  ACCESS TIME  13 MIN
 NODE NUMBER FOUND  12  ACCESS TIME  14 MIN
 NODE NUMBER FOUND  16  ACCESS TIME  35 MIN
 DO YOU WISH TO INCLUDE MINOR STOPS (Y/N)? Y
 NODE NUMBER ASSIGNED TO ORIGIN IS   23
```

```
 NODE NUMBER ASSIGNED TO DESTINATION IS 24
 RUN PROGRAM NLEG2
PROGRAM TERMINATED
```

The above program finds legs from the origin and destination to basic nodes in the network. It is necessary, however, to run the following program to find legs to minor stops in the network as well.

```
RUN NLEG2
 ENTER OLD NETWORK IDENTIFICATION NUMBER? 2
 ENTER NEW NETWORK IDENTIFICATION NUMBER? 22
 NUMBER  1  STOP FOUND BETWEEN NODE  21 AND NODE  20
 ACCESS TIME =  17 MIN.  NODE NUMBER ASSIGNED = 25
 NUMBER  1  STOP FOUND BETWEEN NODE  11 AND NODE  16
 ACCESS TIME =   3 MIN.  NODE NUMBER ASSIGNED = 26
 NUMBER  2  STOP FOUND BETWEEN NODE  16 AND NODE  11
 ACCESS TIME =   6 MIN.  NODE NUMBER ASSIGNED = 27
PROGRAM TERMINATED
```

The above program finds legs to minor stops in the network and outputs the information to three files: LEG2 stores the legs found and access times; NODEID22 is the new node identity file and XYCODE22 is the new node location file.

You then run [Linkad] to generate a network file for the detail network with legs:

```
RUN LINKAD
 ENTER OLD NETWROK IDENTIFICATION NUMBER? 2
 ENTER NEW NETWROK IDENTIFICATION NUMBER? 22
PROGRAM TERMINATED
```

The above program generates a temporary network file LINKT22 which should then be sorted to produce a permanent file LINKS22.

Network 22 is shown in Figure A.4 and the contents of LINKT22 is shown in Table A.5.

A.11 Solving Minimum Path Problems in a Detailed Network

Having built a detailed network, all three types of minimum path problems can be solved in the same way as described in Sections A.4, A.5 and A.6.

For example, a type 1 problem can be solved as follows:

```
RUN DIJ ,
 ENTER NETWORK IDENTIFICATION NUMBER? 22
 ENTER ORIGIN NUMBER? 23
 ENTER START TIME (HR,MIN)? 7,0
 ENTER DESTINATION NODE NUMBER (-1) IF NOT REQUIRED? -1
PROGRAM TERMINATED
```

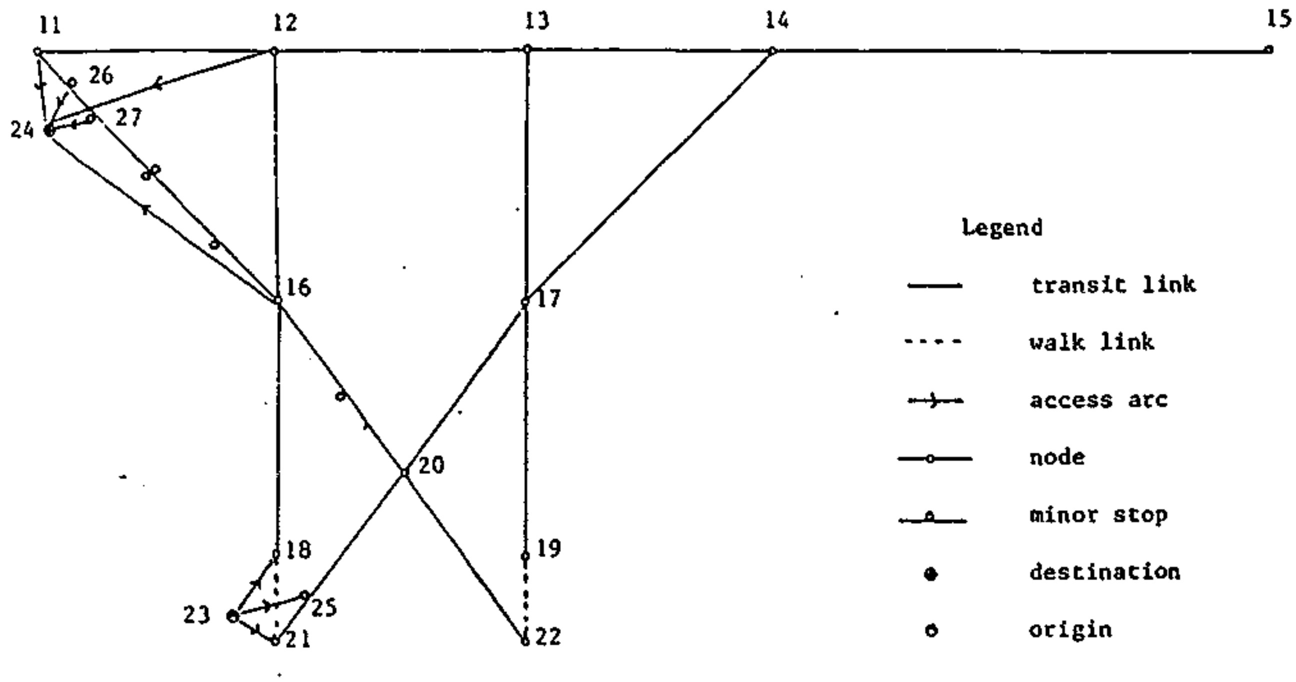


Figure A.4 Transit network 22

1234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890

1	2	10	13	420	435	450	465	480	492	504	516	528	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0A 1
2	3	10	13	430	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0A 1
3	4	10	12	440	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0A 1
4	5	20	11	450	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 1
5	4	20	13	425	440	455	470	482	494	506	518	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0B 1
4	3	10	12	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
3	2	10	11	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
2	1	10	10	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
1	2	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
2	3	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
3	7	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
7	9	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
9	7	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
7	3	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
3	2	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
2	1	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
1	2	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
2	6	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
6	8	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
8	3	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
3	2	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
2	1	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
11	10	10	12	424	439	454	469	484	499	514	529	544	559	574	589	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
10	7	10	12	434	449	464	479	494	509	524	539	554	569	584	599	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
7	4	15	11	444	459	474	489	504	519	534	549	564	579	594	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
4	7	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
7	10	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
10	11	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
12	11	10	9	424	439	454	469	484	499	514	529	544	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
10	6	10	9	434	449	464	479	494	509	524	539	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
9	1	15	9	444	459	474	489	504	519	534	549	564	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
1	6	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
6	12	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
10	12	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
9	12	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
12	9	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
8	11	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
11	8	21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
13	9	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
13	11	22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
13	15	17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
15	10	7	12	427	442	457	472	487	502	517	532	547	562	577	592	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
1	14	13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
16	14	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
1	15	4	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
17	14	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
6	17	2	9	444	459	474	489	504	519	534	549	564	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
2	14	14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL
6	14	35	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0W AL

Table A·5 LINKT22

FOR004

17

22

11	16	7 54	A5
12	16	7 47	B3
13	17	8 5	B2
14	17	7 54	A4
15	14	8 20	A1
16	18	7 37	B3
17	20	7 39	A4
18	23	7 17	WAL
19	22	7 57	WAL
20	25	7 29	A4
21	23	7 22	WAL
22	20	7 40	B5
23	0	7 0	
24	27	7 47	WAL
25	23	7 17	WAL
26	11	8 9	B5
27	16	7 41	A5

RUN PRIPTH

ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 23,24

FROM NODE	TO NODE	ARRIVE TIME	ROUTE USED
	23	7 0	
23	18	7 17	WAL
18	16	7 37	B3
16	27	7 41	A5
27	24	7 47	WAL

```
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 23,15
      FROM          TO             ARRIVE      ROUTE
      NODE          NODE           TIME        USED
                      23             7  0
       23             25             7 17        WAL
       25             20             7 29        A4
       20             17             7 39        A4
       17             14             7 54        A4
       14             15             8 20        A1
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)? 23,13
      FROM          TO             ARRIVE      ROUTE
      NODE          NODE           TIME        USED
                      23             7  0
       23             25             7 17        WAL
       25             20             7 29        A4
       20             17             7 39        A4
       17             13             8  5        B2
ENTER ORIGIN AND DESTINATION NODE(END WITH 0,0)?0,0
PROGRAM TERMINATED
```

You may wish to continue and try to solve a type 2, 3 or 4 problem. The solution of a type 3 problem is given below:

```
NODE  ARRIVE  DEPART  ROUTE
      TIME    TIME    USED
----------------------------
 23    7: 0    7: 0   WAL
 25    7:17    7:22   A4
 20    7:29    7:29   A5
 16    7:39    7:39   A5
 27    7:41    7:41   WAL
 24    7:47
```

WEIGHTINGS FOR WALK TIME=2.0, WAIT TIME=2.0,
ROUTE CHANGE= 5 MIN
IN-VEH TIME= 19, WALK TIME= 23, WAIT TIME= 5,
ROUTE CHANGES= 1
JOURNEY TIME= 47 MIN.   WEIGHTED JOURNEY TIME= 80 MIN.

A different path as shown below may be obtained if a different set of time weightings is used.

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED
23	7: 0	7: 0	WAL
18	7:17	7:27	B3
16	7:37	7:39	A5
27	7:41	7:41	WAL
24	7:47		

WEIGHTINGS FOR WALK TIME=1.0, WAIT TIME=1.0,
ROUTE CHANGE= 10 MIN
IN-VEH TIME= 12, WALK TIME= 23, WAIT TIME= 12,
ROUTE CHANGES= 1
JOURNEY TIME= 47 MIN.   WEIGHTED JOURNEY TIME= 57 MIN.

A.12 Building a Network with Traffic Zones

A network with connections to traffic zones in shown in Figure A.5. This network is given an identification number of 1 and can be built by the following procedure:

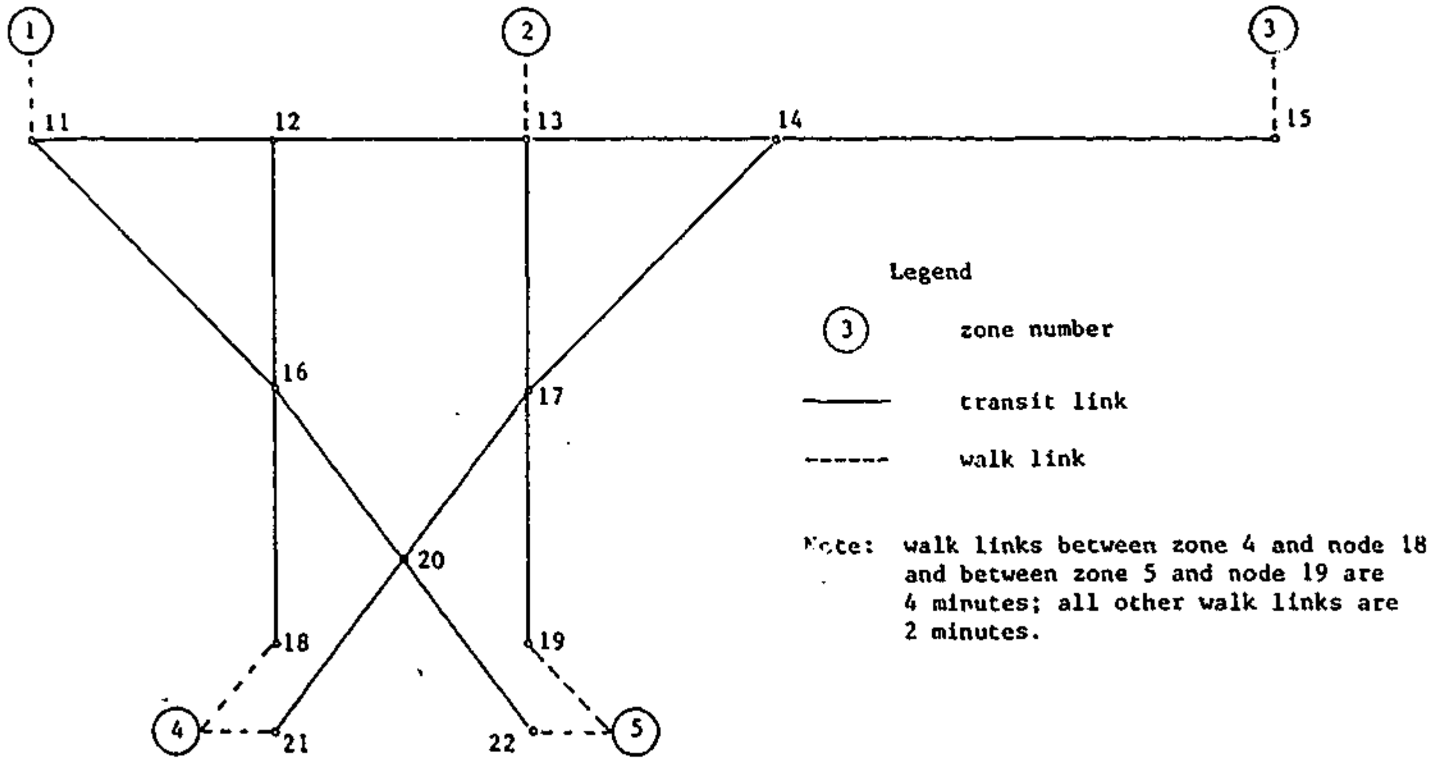


Figure A.5 Zonal transit network 1

```
RUN ZONE
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER ORIGIN ZONE, END WITH 0? 1
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 11,2
 ENTER ORIGIN ZONE, END WITH 0? 2
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 13,2
 ENTER ORIGIN ZONE, END WITH 0? 3
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 15,2
 ENTER ORIGIN ZONE, END WITH 0? 4
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 18,4
 ENTER ORIGIN ZONE, END WITH 0? 4
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 21,2
 ENTER ORIGIN ZONE, END WITH 0? 5
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 19,4
 ENTER ORIGIN ZONE, END WITH 0? 5
 ENTER ACCESS NODE AND LINK TIME IN MINUTES? 22,2
 ENTER ORIGIN ZONE, END WITH 0? 0
PROGRAM TERMINATED
```

The data coded is output to a file called ZONETM1:

ZONETM1.

1	11	2
2	13	2
3	15	2
4	18	4
4	21	2
5	19	4
5	22	2

Copy the timetable data in TIMETBL2 to a new file called TIMETBL1 and proceed as follows:

```
RUN SELINKZ
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER START TIME OF SELECTED PERIOD (HR,MIN)? 7,0
 ENTER FINISH TIME OF SELECTED PERIOD (HR,MIN)? 10,0
 ENTER LARGEST ZONE NUMBER? 5
 ENTER SMALLEST NODE NUMBER? 11
 ENTER LARGEST NODE NUMBER? 22
PROGRAM TERMINATED
```

The above procedure generates a temporary network file called LINKT1 which is shown in Table A.6. This file should then be sorted to produce a permanent file LINKS1.

A.13 Generating a Zonal Trip Time Matrix

The procedure for generating a zonal trip time matrix is described below:

```
RUN DIJZ
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 DO YOU WISH TO BACK-TRACK QUICKEST PATHS(Y/N)? Y
 ENTER START TIME (HR,MIN)? 7,15
PROGRAM TERMINATED
```

The above program generates a zonal trip time matrix based on the quickest trip time between zones called QUKTIME1. You also have the option to back-track quickest paths and the output is written to PATH1.

12345678901234567890123456789012345678901234567890123456789012345678901234567890

1	6	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
6	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
2	8	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
8	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
3	10	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
10	3	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
4	13	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
13	4	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
4	16	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
16	4	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
5	14	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
14	5	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
5	17	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
17	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0WAL
6	7	10	13	420	435	450	465	480	492	504	516	528	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0A 1
7	8	10	13	430	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0A 1
8	9	10	12	440	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0A 1
9	10	20	11	450	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 1
10	9	20	13	425	440	455	470	482	494	506	518	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0B 1
9	8	10	12	445	460	475	490	502	514	526	538	550	565	580	595	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
8	7	10	11	455	470	485	500	512	524	536	548	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
7	6	10	10	465	480	495	510	522	534	546	558	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 1
6	7	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
7	8	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
8	12	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
12	14	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 2
14	12	10	9	425	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
12	8	10	9	435	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
8	7	10	8	445	465	485	505	525	545	565	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
7	6	10	8	455	475	495	515	535	555	575	595	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 2
6	7	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
7	11	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
11	13	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 3
13	11	10	8	422	447	472	497	522	547	572	597	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
11	7	10	7	432	457	482	507	532	557	582	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
7	6	10	7	442	467	492	517	542	567	592	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 3
16	15	10	12	424	439	454	469	484	499	514	529	544	559	574	589	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
15	12	10	12	434	449	464	479	494	509	524	539	554	569	584	599	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
12	9	15	11	444	459	474	489	504	519	534	549	564	579	594	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 4
9	12	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
12	15	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
15	16	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 4
17	15	10	9	424	439	454	469	484	499	514	529	544	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
15	11	10	9	434	449	464	479	494	509	524	539	554	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
11	6	15	9	444	459	474	489	504	519	534	549	564	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0A 5
6	11	15	11	425	440	455	470	485	500	515	530	545	560	575	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
11	15	10	11	440	455	470	485	500	515	530	545	560	575	590	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5
15	17	10	10	450	465	480	495	510	525	540	555	570	585	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0B 5

Table A·6 LINKT1

QUKTIME1.

0	32	67	42	42
32	0	37	42	34
57	37	0	71	72
41	52	67	0	27
41	32	67	27	0

PATH1 is a large file and therefore only the first portion of it is shown below:

PATH1. (first portion)

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART
1	7:15	7:15	WAL
11	7:17	7:25	A2
12	7:35	7:35	A2
13	7:45	7:45	WAL
2	7:47		

IN-VEH TIME= 20, WALK TIME= 4, WAIT TIME= 8,
ROUTE CHANGES= 0 JOURNEY TIME= 32 MIN.

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART
1	7:15	7:15	WAL
11	7:17	7:25	A2
12	7:35	7:35	A2
13	7:45	7:50	A1
14	8: 0	8: 0	A1
15	8:20	8:20	WAL
3	8:22		

IN-VEH TIME= 50, WALK TIME= 4, WAIT TIME= 13,
ROUTE CHANGES= 1 JOURNEY TIME= 67 MIN.

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART
1	7:15	7:15	WAL
11	7:17	7:20	B5
16	7:35	7:35	B5
20	7:45	7:45	B4
21	7:55	7:55	WAL
4	7:57		

IN-VEH TIME= 35, WALK TIME= 4, WAIT TIME= 3,
ROUTE CHANGES= 1 JOURNEY TIME= 42 MIN.

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART
1	7:15	7:15	WAL
11	7:17	7:20	B5
16	7:35	7:35	B5
20	7:45	7:45	B5
22	7:55	7:55	WAL
5	7:57		

IN-VEH TIME= 35, WALK TIME= 4, WAIT TIME= 3,
ROUTE CHANGES= 0 JOURNEY TIME= 42 MIN.

A.14 Multi-Interval Assignment

Transit Network 1 shown in Figure A.5 will be used to illustrate the assignment procedure. The input zonal flow matrices for 6 consecutive time intervals of 20 minutes each are shown in Table A.7. This information is coded into a file called FLOWMAT1 as shown below:

```
FLOWMAT1  (part of file)
     6
   420   440   460   480   500   520   540
     0    60    80    80    60
    60     0    60    60    60
    80    80     0    80    80
```

(ètc. each record correspond to one row of matrices shown in Table A.7)

The next step is to run [Rangen] to generate the various random parameters:

```
RUN RANGEN
 ENTER ANY FOUR DIGIT NUMBER? 1234
 ENTER MEAN AND STANDARD DEVIATION OF WEIGHTING
 COEFFICIENT FOR WALKING TIME? 2
 ENTER MEAN AND STANDARD DEVIATION OF WEIGHTING
 COEFFICIENT FOR WAITING TIME? 1.5
 ENTER MEAN AND STANDARD DEVIATION OF WEIGHTING
 COEFFICIENT FOR ROUTE CHANGE? 3
PROGRAM TERMINATED
```

Now you can proceed to run the main assignment program:

Table A.7 Multi-interval zone to zone trip matrices

For trips starting within period 7:00 am to 7:19 am

	1	2	3	4	5
1	-	60	80	80	60
2	60	-	60	60	60
3	80	80	-	80	80
4	80	60	80	-	60
5	60	60	80	60	-

For trips starting within period 7:20 am to 7:39 am

	1	2	3	4	5
1	-	80	100	100	80
2	80	-	80	80	80
3	100	100	-	100	100
4	100	80	100	-	80
5	80	80	100	80	-

For trips starting within period 7:40 am to 7:59 am

	1	2	3	4	5
1	-	100	120	120	100
2	100	-	100	100	100
3	120	120	-	120	120
4	120	100	120	-	100
5	100	100	120	100	-

For trips starting within period 8:00 am to 8:19 am

	1	2	3	4	5
1	-	120	100	100	120
2	120	-	120	120	120
3	100	100	-	100	100
4	100	120	100	-	120
5	120	120	100	120	-

For trips starting within period 8:20 am to 8:39 am

	1	2	3	4	5
1	-	80	100	100	80
2	80	-	80	80	80
3	100	100	-	100	100
4	100	80	100	-	80
5	80	80	100	80	-

For trips starting within period 8:40 am to 8:59 am

	1	2	3	4	5
1	-	60	80	80	60
2	60	-	60	60	60
3	80	80	-	80	80
4	80	60	80	-	60
5	60	60	80	60	-

```
RUN INTASS
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER FLOW MATRIX IDENTIFICATION NUMBER? 1
 ENTER ITERATION IDENTIFICATION NUMBER? 1
 ENTER MINIMUM FLOW FOR OPTIMAL PATH ASSIGNMENT? 5
PROGRAM TERMINATED
```

The assignment paths generated by the above program are output to a file called ZONEPATH. Part of the contents of this file is shown in Table A.8.

The assignment results are output to a file called LINKFLOW1. As shown in Table A.9 this file is in link format and shows the flow assigned to each of the 6 time intervals along each link.

As the effect of time lag has not been accounted for, the link flows for the earlier time intervals may be under estimated, since paths which start before 7 am but end after 7 am has not been included in the assignment. It is possible to account for this time lag effect by including earlier time intervals in the assignment.

The procedure can be iterated by running [Rangen] and [Intass] again to provide another set of results. For example, you may repeat the same procedure three more times and obtain the results in the following three files: LINKFLOW2, LINKFLOW3, and LINKFLOW4. These 3 files are not shown because the assignment process is stochasic, and the results which you obtain would be different.

Next, run [Average] to obtain an average result of the above four iterations:

Table A.8 ZONEPATH

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:15	A1	9
12	7:25	7:25	A1	17
13	7:35	7:35	WALK	19
2	7:37			

WEIGHTED TRIP TIME IS

20 + 1.5 X 4 + 1.9 X 0 + 4 X 0
= 26.1 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:15	A1	9
12	7:25	7:25	A1	17
13	7:35	7:35	A1	22
14	7:45	7:45	A1	25
15	8: 5	8: 5	WALK	27
3	8: 7			

WEIGHTED TRIP TIME IS

50 + 2.8 X 4 + 1.9 X 0 + 4 X 0
= 61.4 MIN

FLOW ASSIGNED TO LAST PATH = 80

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:20	B5	12
16	7:35	7:35	B5	32
20	7:45	7:45	B4	43
21	7:55	7:55	WALK	45
4	7:57			

WEIGHTED TRIP TIME IS

35 + 2.4 X 4 + 1.5 X 5 + 4 X 1
= 56.1 MIN

FLOW ASSIGNED TO LAST PATH = 80

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
1	7:13	7:13	WALK	1
11	7:15	7:20	B5	12
16	7:35	7:35	B5	32
20	7:45	7:45	B5	44
22	7:55	7:55	WALK	47
5	7:57			

WEIGHTED TRIP TIME IS

35 + 2.3 X 4 + 1.1 X 5 + 2 X 0
= 49.8 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
2	7:19	7:19	WALK	2
13	7:21	7:25	B2	20
12	7:35	7:35	B2	15
11	7:45	7:45	WALK	8
1	7:47			

WEIGHTED TRIP TIME IS

20 + 1.0 X 4 + 1.5 X 4 + 4 X 0
= 30.1 MIN

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
2	7:19	7:19	WALK	2
13	7:21	7:35	A1	22
14	7:45	7:45	A1	25
15	8: 5	8: 5	WALK	27
3	8: 7			

WEIGHTED TRIP TIME IS

30 + 3.4 X 4 + 1.8 X 14 + 3 X 0
= 69.0 MIN

FLOW ASSIGNED TO LAST PATH = 60

NOTE:

WEIGHTED TRIP TIME = IN-VEHICLE TIME + W_1 (WALKING TIME)
+ W_2 (WAITING TIME) + W_3 (NUMBER OF ROUTE CHANGES)

```
RUN AVERAGE
 ENTER FLOW MATRIX IDENTIFICATION NUMBER? 1
 ENTER NAME OF INPUT LINKFLOW FILE(END WITH STOP)?LINKFLOW1
 ENTER NAME OF INPUT LINKFLOW FILE(END WITH STOP)?LINKFLOW2
 ENTER NAME OF INPUT LINKFLOW FILE(END WITH STOP)?LINKFLOW3
 ENTER NAME OF INPUT LINKFLOW FILE(END WITH STOP)?LINKFLOW4
 ENTER NAME OF INPUT LINKFLOW FILE(END WITH STOP)?STOP
 ENTER NAME ASSIGNED TO OUTPUT FILE? AV1234
PROGRAM TERMINATED
```

File AV1234 is shown in Table A.10. The results can be converted to timetable format by running [Intload]:

```
RUN INTLOAD
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER FLOW MATRIX IDENTIFICATION NUMBER? 1
 ENTER NAME OF INPUT LINKFLOW FILE? AV123
PROGRAM TERMINATED
```

The program outputs a file called ILOAD which is shown in Table A.11. In this file, the results are said to be in timetable format because it shows the passenger loadings of transit vehicles at different times corresponding to departure times at stations as scheduled in timetables.

As the assignment period is from 7 am to 9 am, all transit routes scheduled after 9 am have not been assigned any flow and this explains why the lower portions of the timetabled flows are zero.

Table A.9 LINKFLOW1

A-NODE	B-NODE	ROUTE NO.	FLOW IN INTERVAL NOS. 1	2	3	4	5	6
1	11	WALK	280	360	440	440	360	280
2	13	WALK	240	320	400	480	320	240
3	15	WALK	320	400	480	400	400	320
4	18	WALK	0	0	0	0	0	0
4	21	WALK	280	360	270	610	360	280
5	19	WALK	112	0	100	460	340	140
5	22	WALK	146	340	320	0	0	120
11	1	WALK	0	0	200	340	420	420
11	12	A1	140	0	140	320	180	80
11	12	A3	0	0	100	0	0	80
11	12	A2	0	0	120	120	0	120
11	16	B5	0	140	0	440	180	0
12	11	B3	0	0	0	0	0	0
12	11	B1	0	0	0	230	240	0
12	11	B2	0	60	80	50	80	200
12	13	A2	0	0	120	120	0	120
12	13	A1	0	140	140	220	280	80
12	16	A3	0	0	0	260	0	80
13	2	WALK	0	60	140	400	440	580
13	12	B2	0	60	120	50	80	200
13	12	B1	0	0	200	150	240	260
13	14	A1	0	140	132	400	420	160
13	17	A2	0	120	80	430	520	0
14	13	B1	0	0	650	360	0	520
14	15	A1	0	60	80	638	240	820
14	17	B4	0	0	80	50	120	400
15	3	WALK	0	0	60	80	638	240
15	14	B1	0	720	480	0	800	320
16	11	A5	0	140	180	0	320	100
16	12	B3	0	0	0	0	0	0
16	18	A3	0	0	0	260	0	0
16	20	B5	0	140	0	0	440	180
17	13	B2	0	112	360	100	500	220
17	14	A4	0	166	0	0	660	100
17	19	A2	0	60	80	280	320	0
17	20	B4	0	60	0	80	720	0
18	4	WALK	0	0	0	0	260	0
18	16	B3	0	0	0	0	0	0
19	5	WALK	0	0	60	80	280	320
19	17	B2	0	112	0	100	800	140
20	16	A5	0	140	180	220	100	240
20	17	A4	0	166	360	340	220	320
20	21	B4	0	60	220	100	420	520
20	22	B5	0	60	140	180	340	80
21	4	WALK	0	0	280	0	100	940
21	20	A4	280	360	0	880	360	280
22	5	WALK	0	0	200	0	180	420
22	20	A5	146	340	0	320	0	120

Table A.10 AV1234

A-NODE	B-NODE	ROUTE NO.	FLOW IN INTERVAL NOS. 1	2	3	4	5	6
1	11	WALK	280	360	440	440	360	280
2	13	WALK	240	320	400	480	320	240
3	15	WALK	320	400	480	400	400	320
4	18	WALK	0	50	72	110	0	40
4	21	WALK	280	225	410	372	360	240
5	19	WALK	123	145	130	190	200	75
5	22	WALK	136	155	330	255	155	185
11	1	WALK	0	0	260	215	460	535
11	12	A1	90	65	105	300	280	20
11	12	A3	0	45	85	25	0	20
11	12	A2	15	15	50	30	10	30
11	16	B5	35	195	25	190	320	0
12	11	B3	0	0	0	110	50	0
12	11	B1	0	0	60	172	235	75
12	11	B2	0	60	65	57	60	50
12	13	A2	15	15	50	42	10	30
12	13	A1	0	90	170	220	270	170
12	16	A3	0	45	0	190	25	40
13	2	WALK	0	65	155	315	495	565
13	12	B2	0	60	35	57	60	50
13	12	B1	0	60	145	112	305	250
13	14	A1	0	200	173	300	400	210
13	17	A2	0	120	125	280	505	75
14	13	B1	0	60	480	375	165	410
14	15	A1	0	60	100	569	380	605
14	17	B4	0	20	110	50	140	300
15	3	WALK	0	0	60	100	569	380
15	14	B1	80	540	460	220	700	240
16	11	A5	0	140	90	45	275	140
16	12	B3	0	0	0	122	110	0
16	18	A3	0	20	25	190	50	0
16	20	B5	0	120	110	25	305	180
17	13	B2	0	123	290	207	240	285
17	14	A4	0	156	0	140	405	75
17	19	A2	0	60	125	180	295	75
17	20	B4	0	60	20	110	495	55
18	4	WALK	0	20	25	0	190	50
18	16	B3	0	0	122	110	0	0
19	5	WALK	0	0	60	125	180	295
19	17	B2	0	123	145	130	290	135
20	16	A5	0	140	135	135	140	235
20	17	A4	0	156	362	170	235	350
20	21	B4	0	60	160	270	325	410
20	22	B5	0	75	85	275	225	170
21	4	WALK	0	0	220	135	135	735
21	20	A4	280	180	277	550	360	140
22	5	WALK	0	0	160	150	125	395
22	20	A5	136	135	140	465	125	175

Table A.11 ILOAD

ROUTE NUMBER A1

NODE	11	12	13	14	15
	45	0	100	60	
	45	90	100	100	
	65	85	173	284	
	105	85	300	284	
	150	220	200	380	
	150	135	200	302	
	140	135	105	302	
	140	85	105	0	
	20	85	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B1

NODE	15	14	13	12	11
	80	60	60	60	
	270	240	145	86	
	270	240	112	86	
	460	375	152	235	
	110	82	152	37	
	110	82	125	37	
	350	205	125	0	
	350	205	0	0	
	240	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A2

NODE	11	12	13	17	19
	15	15	120	60	
	15	15	125	125	
	50	50	280	180	
	30	42	505	295	
	10	10	75	75	
	30	30	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B2

NODE	19	17	13	12	11
	0	0	60	60	
	123	123	85	65	
	145	290	57	57	
	130	207	60	60	
	290	240	50	50	
	135	285	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A3

NODE	11	12	16	18
	0	0	20	
	45	45	25	
	85	190	190	
	25	25	50	
	20	40	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B3

NODE	18	16	12	11
	0	0	0	
	0	0	0	
	122	122	110	
	110	110	50	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A4

NODE	21	20	17	14
	140	0	78	
	140	156	78	
	180	181	0	
	277	181	140	
	275	170	202	
	275	235	202	
	360	175	75	
	140	175	0	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B4

NODE	14	17	20	21
	0	30	60	
	10	30	160	
	10	20	135	
	110	110	135	
	50	247	325	
	70	247	410	
	70	55	0	
	300	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A5

NODE	22	20	16	11
	68	0	70	
	68	140	70	
	135	67	90	
	140	67	45	
	232	135	137	
	232	140	137	
	125	117	140	
	175	117	0	
	0	0	0	

ROUTE NUMBER B5

NODE	11	16	20	22
	35	60	75	
	97	60	85	
	97	110	137	
	25	25	137	
	190	152	225	
	160	152	170	
	160	180	0	
	0	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

A.15 Dynamic Assignment

Transit Network 2 shown in Figure A.1 will be used in this assignment sub-model. The input station to station flow matrices are coded to a file called FLOWMAT2 which is shown below:

```
FLOWMAT2  (part of file)
    6
  420   440   460   480   500   520   540
    5
   11    13    15    21    22
    0    60    80    80    60
   60     0    60    60    60
   80    80     0    80    80
```

(etc. each record correspond to one row of matrices shown in Table A.7)

Next run [Rangen] to generate the required random parameters. The procedure is exactly the same as that shown in the previous section.

Now you can run the main assignment program:

```
RUN VEHASS
 ENTER FLOW MATRIX IDENTIFICATION NUMBER? 2
 ENTER NETWORK IDENTIFICATION NUMBER? 2
 ENTER START TIME OF ASSIGNMENT PERIOD (HR,MIN)? 7,0
 ENEER END TIME OF ASSIGNMENT PERIOD (HR,MIN)? 9,0
 ENTER MINIMUM FLOW FOR OPTIMAL PATH ASSIGNMENT? 5
PROGRAM TERMINATED
```

The assignment paths generated by the above program are output to a file called NODEPATH. A portion of this file is shown in Table A.12.

The assignment results obtained is output to a file called VEHFLOW as shown in Table A.13. The results in this file is arranged in a vehicle-based link format. This means that in a network file, a link record contains the scheduled departure times of various transit vehicles departing from the A-NODE of the link. VEHFLOW shows for each link the flow assigned to various transit vehicle at the various scheduled departure times.

To facilitate result evaluation, VEHFLOW can be converted to timetable format by running [Vehload]:

```
RUN VEHLOAD
 ENTER NETWORK IDENTIFICATION NUMBER? 2
PROGRAM TERMINATED
```

The result obtained from the above is output to file TBLOAD which is shown in Table A.14.

As the assignment ends at 9 am, vehicles with departing times after 9 am will have zero flow.

Table A.12 NODEPATH

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 2	7: 2	A3	2
12	7:12	7:12	A3	10
16	7:22	7:22	A3	21
18	7:32	7:32	WALK	28
21	7:34			

WEIGHTED TRIP TIME IS

30 + 2.6 X 2 + .9 X 0 + 4 X 0
= 35.1 MINUTES

FLOW ASSIGNED TO LAST PATH = 8

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 4	7: 5	B5	4
16	7:20	7:22	A3	21
18	7:32	7:32	WALK	28
21	7:34			

WEIGHTED TRIP TIME IS

25 + 2.2 X 2 + 1.2 X 3 + 3 X 1
= 36.0 MINUTES

FLOW ASSIGNED TO LAST PATH = 12

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7: 7	7:20	B5	4
16	7:35	7:35	B5	22
20	7:45	7:45	B4	33
21	7:55			

WEIGHTED TRIP TIME IS

35 + 1.7 X 0 + 1.4 X 13 + 2 X 1
= 55.2 MINUTES

FLOW ASSIGNED TO LAST PATH = 60

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7:21	7:27	A3	2
12	7:37	7:37	A3	10
16	7:47	7:47	A3	21
18	7:57	7:57	WALK	28
21	7:59			

WEIGHTED TRIP TIME IS

30 + 2.0 X 2 + 2.4 X 6 + 2 X 0
= 48.3 MINUTES

FLOW ASSIGNED TO LAST PATH = 35

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
11	7:28	7:35	B5	4
16	7:50	7:50	B5	22
20	8: 0	8: 0	B4	33
21	8:10			

WEIGHTED TRIP TIME IS

35 + 3.0 X 0 + .9 X 7 + 4 X 1
= 45.0 MINUTES

FLOW ASSIGNED TO LAST PATH = 40

NOTE:

WEIGHTED TRIP TIME

= IN-VEHICLE TIME + W_1 (WALKING TIME)

+ W_2 (WAITING TIME)

+ W_3 (NUMBER OF ROUTE CHANGES)

Table A.13 VEHFLOW

A-NODE	B-NODE	ROUTE NO.	VEHICLE-BASED LINK LOADINGS												
11	12	A1	0	90	90	112	165	102	124	108	73	0	0	0	0
11	12	A3	8	35	107	60	59	0	0	0					
11	12	A2	15	55	32	30	20	31	0	0	0				
11	16	B5	27	105	80	70	143	105	115	56	0	0	0		
12	11	B3	0	24	90	62	0	0	0						
12	11	B1	60	85	115	175	98	104	0	0	0	0			
12	11	B2	80	60	64	20	6	0	0	0					
12	13	A2	15	55	32	35	20	31	0	0	0				
12	13	A1	0	90	90	112	165	104	124	162	73	0	0	0	0
12	16	A3	8	35	192	60	87	0	0						
13	12	B2	80	75	64	20	6	0	0	0					
13	12	B1	60	155	115	175	126	104	96	0	0	0	0		
13	14	A1	60	168	159	337	180	110	104	96	0	0	0	0	
13	17	A2	160	140	260	220	272	0	0	0					
14	13	B1	60	156	270	170	210	180	150	120	0	0	0	0	
14	15	A1	60	152	279	337	300	280	274	0	0	0	0		
14	17	B4	0	0	20	130	75	170	130	90	0	0	0		
15	14	B1	80	240	300	340	280	240	240	240	200	0	0	0	0
16	11	A5	28	108	89	45	159	115	140	0	0				
16	12	B3	0	24	97	116	48	0	0						
16	18	A3	20	35	182	75	0	0	0						
16	20	B5	15	105	80	70	153	90	115	0	0	0	0		
17	13	B2	14	102	347	172	190	233	0	0	0				
17	14	A4	30	165	0	120	170	170	122	0	0	0	0		
17	19	A2	80	125	180	195	225	0	0	0					
17	20	B4	6	95	20	145	155	201	130	0	0	0	0		
18	16	B3	0	28	97	121	57	0	0	0					
18	21	WALK	413												
19	17	B2	20	105	132	73	136	135	0	0	0				
19	22	WALK	970												
20	16	A5	28	104	89	45	159	110	131	63	0				
20	17	A4	42	165	215	219	250	150	220	137	0	0	0	0	
20	21	B4	57	199	129	164	318	257	0	0	0	0			
20	22	B5	72	104	109	209	194	152	0	0	0	0			
21	18	WALK	343												
21	20	A4	56	210	237	222	317	209	274	175	0	0	0	0	
22	19	WALK	631												

Table A.14 TBLOAD

ROUTE NUMBER A1

NODE	11	12	13	14	15
	0	0	60	60	
	90	90	168	152	
	90	90	159	279	
	112	112	337	337	
	165	165	180	300	
	102	104	110	280	
	124	124	104	274	
	108	162	96	0	
	73	73	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B1

NODE	15	14	13	12	11
	80	60	60	60	
	240	156	155	85	
	300	270	115	115	
	340	170	175	175	
	280	210	126	98	
	240	180	104	104	
	240	150	96	0	
	240	120	0	0	
	200	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A2

NODE	11	12	13	17	19
	15	15	160	80	
	55	55	140	125	
	32	32	260	180	
	30	35	220	195	
	20	20	272	225	
	31	31	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER B2

NODE	19	17	13	12	11
	20	14	80	80	
	105	102	75	60	
	132	347	64	64	
	73	172	20	20	
	136	190	6	6	
	135	233	0	0	
	0	0	0	0	
	0	0	0	0	
	0	0	0	0	

ROUTE NUMBER A3

NODE	11	12	16	18
	8	8	20	
	35	35	35	
	107	192	182	
	60	60	75	
	59	87	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER B3

NODE	18	16	12	11
	0	0	0	
	28	24	24	
	97	97	90	
	121	116	62	
	57	48	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A4

NODE	21	20	17	14
	56	42	30	
	210	165	165	
	237	215	0	
	222	219	120	
	317	250	170	
	209	150	170	
	274	220	122	
	175	137	0	
	0	0	0	
	0	0	0	
	0		0	
	0	0	0	

ROUTE NUMBER B4

NODE	14	17	20	21
	0	6	57	
	0	95	199	
	20	20	129	
	130	145	164	
	75	155	318	
	170	201	257	
	130	130	0	
	90	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

ROUTE NUMBER A5

NODE	22	20	16	11
	32	28	28	
	149	104	108	
	170	89	89	
	180	45	45	
	250	159	159	
	255	110	115	
	195	131	140	
	127	63	0	
	0	0	0	

ROUTE NUMBER B5

NODE	11	16	20	22
	27	15	72	
	105	105	104	
	80	80	109	
	70	70	209	
	143	153	194	
	105	90	152	
	115	115	0	
	56	0	0	
	0	0	0	
	0	0	0	
	0	0	0	

A.16 Start Time Based Accessibility

Transit network 1 shown in Figure A.5 will be used and it is required to calculate the temporal variation of trips from zone 4 to other zones at different starting times. The first task is to obtain the set of optimal starting times from zone 4 and this can be obtained by running [Gens]:

```
RUN GENS
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER ORIGIN ZONE? 4
PROGRAM TERMINATED
```

The program reads the input data from network file LINKS1. and outputs the set of optimal starting times to file STIM.

```
STIM.
418  422  437  443  452  467  468  482  493  497  512  518
527  542  543  557  568  572  587
```

The next step is to run [Access]:

```
RUN ACCESS
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER WEIGHTINGS FOR WALK, WAIT, ROUTE CHANGE? 1.9,2.5,5
 ENTER ORIGIN ZONE NUMBER? 4
PROGRAM TERMINATED
```

The program outputs quickest trip times to file QTIMS., each file record represents the quickest trip times when travelling from the origin zone to each of the 5 zones in the network with a specified earliest starting time:

QTIMS

36	39	69	0	44
39	45	65	0	40
39	50	65	0	25
33	44	74	0	34
39	35	65	0	25
37	40	65	0	25
36	39	76	0	39
39	45	62	0	25
36	45	63	0	29
39	50	59	0	25
39	35	68	0	25
33	44	74	0	34
39	40	65	0	25
37	45	65	0	25
36	44	79	0	39
39	50	65	0	25
36	39	69	0	29
39	35	65	0	25
39	40	65	0	25

Similarly, minimum weighted trip times are output to file OTIMS.:

41.4	53.9	83.9	.0	66.4
47.6	60.1	82.6	.0	72.6
47.6	60.1	82.6	.0	35.1
41.4	66.4	96.4	.0	57.6
47.6	45.1	80.1	.0	35.1
43.9	56.4	82.6	.0	35.1
41.4	53.9	101.4	.0	53.9
47.6	70.1	75.1	.0	35.1
41.4	68.9	77.6	.0	45.1
47.6	67.6	67.6	.0	35.1

47.6	45.1	90.1	.0	35.1
41.4	66.4	96.4	.0	57.6
47.6	57.6	82.6	.0	35.1
43.9	60.1	82.6	.0	35.1
41.4	66.4	108.9	.0	53.9
47.6	60.1	82.6	.0	35.1
41.4	53.9	83.9	.0	45.1
47.6	45.1	80.1	.0	35.1
47.6	57.6	82.6	.0	35.1

Optimal paths generated are output to file PATHS:

PATHS. (Part of file)

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	6:58	6:58	WAL	4
18	7: 2	7: 2	B3	38
16	7:12	7:12	B3	30
12	7:22	7:22	B3	13
11	7:32	7:32	WAL	8
1	7:34			

WEIGHTED TRIP TIME IS

30 + 1.9 X 6 + 2.5 X 0 + 5 X 0 = 41.4 MIN

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	6:58	6:58	WAL	4
18	7: 2	7: 2	B3	38
16	7:12	7:12	B3	30
12	7:22	7:25	A1	17
13	7:35	7:35	WAL	19
2	7:37			

WEIGHTED TRIP TIME IS

30 + 1.9 X 6 + 2.5 X 3 + 5 X 1 = 53.9 MIN

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	6:58	6:58	WAL	4
18	7: 2	7: 2	B3	38
16	7:12	7:12	B3	30
12	7:22	7:25	A1	17
13	7:35	7:35	A1	22
14	7:45	7:45	A1	25
15	8: 5	8: 5	WAL	27
3	8: 7			

WEIGHTED TRIP TIME IS

60 + 1.9 X 6 + 2.5 X 3 + 5 X 1 = 83.9 MIN

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	6:58	6:58	WAL	4
18	7: 2	7: 2	B3	38
16	7:12	7:20	B5	32
20	7:30	7:30	B5	44
22	7:40	7:40	WAL	47
5	7:42			

WEIGHTED TRIP TIME IS

30 + 1.9 X 6 + 2.5 X 8 + 5 X 1 = 66.4 MIN

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	7: 2	7: 2	WAL	5
21	7: 4	7: 4	A4	46
20	7:14	7:14	A5	41
16	7:24	7:24	A5	29
11	7:39	7:39	WAL	8
1	7:41			

WEIGHTED TRIP TIME IS

35 + 1.9 X 4 + 2.5 X 0 + 5 X 1 = 47.6 MIN

NODE	ARRIVE TIME	DEPART TIME	ROUTE USED TO DEPART	LINK NUMBER
4	7: 2	7: 2	WAL	5
21	7: 4	7: 4	A4	46
20	7:14	7:14	A4	42
17	7:24	7:24	A4	34
14	7:39	7:40	B1	24
13	7:50	7:50	WAL	19
2	7:52			

WEIGHTED TRIP TIME IS

45 + 1.9 X 4 + 2.5 X 1 + 5 X 1 = 60.1 MIN

Data for plotting the variation of trip time with starting time can be obtained by running [Plots], suppose it is required to plot the variation of weighted trip time with starting time for trips from zone 4 to zone 5, the procedure is shown below:

```
RUN PLOTS
 ENTER ORIGIN AND DESTINATION ZONE? 4,5
 QUICKEST TRIP TIME OR OPTIMAL TRIP TIME (Q/O)? O
PROGRAM TERMINATED
```

The data generated is output to file DSETS:

Starting time (hour)	Weighted trip time (min.)	Starting time (hour)	Weighted trip time (min.)	Starting time (hour)	Weighted trip time (min)
6.967	66.4	8.033	72.6	9.050	70.1
6.967	82.6	8.217	45.1	9.283	35.1
7.033	72.6	8.217	45.1	9.283	72.6
7.033	72.6	8.283	35.1	9.467	45.1
7.283	35.1	8.283	72.6	9.467	45.1
7.283	72.6	8.533	35.1	9.533	35.1
7.383	57.6	8.533	72.6	9.533	72.6
7.383	57.6	8.633	57.6	9.783	35.1
7.533	35.1	8.633	57.6		
7.533	72.6	8.783	35.1		
7.783	35.1	8.783	72.6		
7.783	56.4	9.033	35.1		
7.800	53.9	9.033	56.4		
7.800	70.1	9.050	53.9		
8.033	35.1				

With the above data set, a graph showing the variation of trip time with starting time such as that shown in Figure A.6 can be plotted.

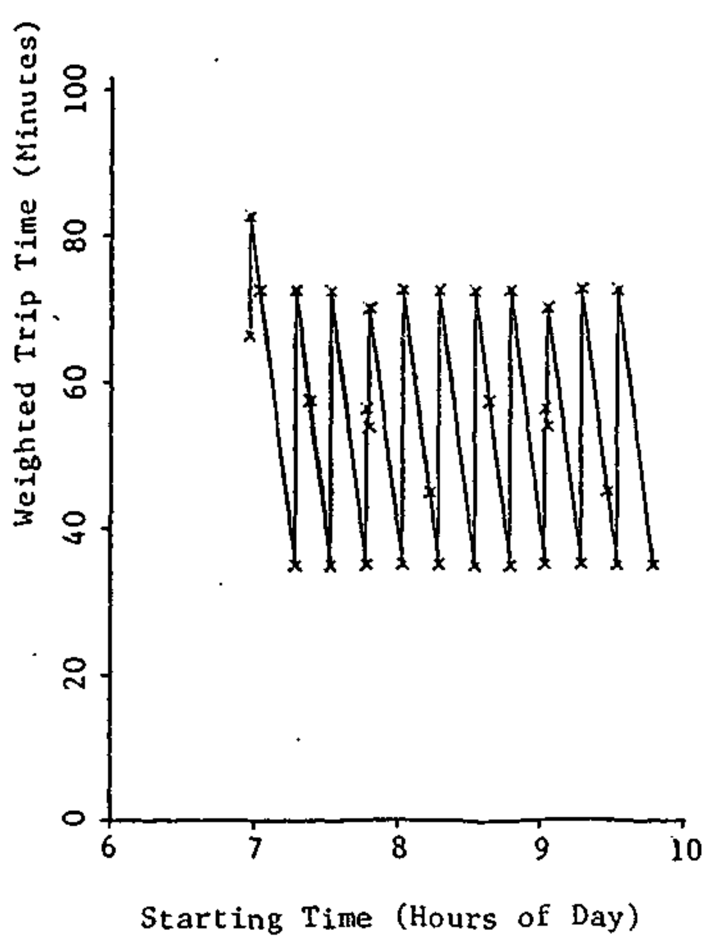


Figure A.6 Variation of weighted trip time with starting time: zone 4 to zone 5

A.17 Arrival Time Based Accessibility

Using the same input information as in Section A.16, suppose it is required to find the temporal variation of trip time with arrival time for trips from zone 4 to zone 5, then the procedure is shown below:

```
RUN GENA
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER DESTINATION ZONE? 5
PROGRAM TERMINATED
```

The set of optimal arrival times at zone 5 is output to file ATIM.:

```
462  469  477  489  492  507  509  522  529  537  549  552
567  569  582  589  597  609  612  629
```

```
RUN ACCESSA
 ENTER NETWORK IDENTIFICATION NUMBER? 1
 ENTER WEIGHTINGS FOR WALK, WAIT, ROUTE CHANGE? 1.9,2.5,5
 ENTER DESTINATION ZONE NUMBERS? 5
PROGRAM TERMINATED
```

The program outputs quickest trip times to file QTIMA, minimum weighted trip times to file OTIMA and optimal paths to file PATHA. In QTIMA, each record contains the quickest trip time for travelling from each of the five zones in the network to the destination zone at a specified latest arrival time. (A value of 9999 means that the destination zone cannot be reached at the earliest arrival time indicated.)

QTIMA

39	9999	9999	25	0
46	26	9999	32	0
39	34	9999	25	0
46	26	66	37	0
39	29	69	25	0
39	44	69	25	0
41	26	56	27	0
39	39	69	25	0
46	26	61	32	0
39	34	69	25	0
46	26	57	37	0
39	29	60	25	0
39	44	63	25	0
41	26	65	27	0
39	39	66	25	0
46	26	61	32	0
39	34	69	25	0
46	26	66	37	0
39	29	69	25	0
39	44	69	40	0

Similarly, minimum weighted trip times are output to file OTIMA:

42.6	999.9	999.9	35.1	.0
51.4	31.4	999.9	52.6	.0
42.6	47.6	999.9	35.1	.0
51.4	31.4	91.4	48.9	.0
42.6	38.9	97.6	35.1	.0
42.6	72.6	97.6	35.1	.0
47.6	31.4	66.4	40.1	.0
42.6	60.1	95.1	35.1	.0
51.4	31.4	78.9	52.6	.0
42.6	47.6	95.1	35.1	.0
51.4	31.4	68.9	48.9	.0
42.6	38.9	75.1	35.1	.0
42.6	72.6	82.6	35.1	.0
47.6	31.4	88.9	40.1	.0
42.6	60.1	90.1	35.1	.0
51.4	31.4	78.9	52.6	.0
42.6	47.6	95.1	35.1	.0
51.4	31.4	91.4	48.9	.0
42.6	38.9	97.6	35.1	.0
42.6	72.6	97.6	72.6	.0

Optimal paths generated are output to file PATHA:

PATHA. (Part of file)

```
NODE  ARRIVE  DEPART  ROUTE USED  LINK
      TIME    TIME    TO ARRIVE   NUMBER
--------------------------------------------
  4           7:17
 21    7:19   7:19    WAL           45
 20    7:29   7:30    A4            43
 22    7:40   7:40    B5            48
  5    7:42   7:42    WAL            7
WEIGHTED TRIP TIME IS
 20 + 1.9 X   4 + 2.5 X   1 +  5 X  1 =  35.1 MIN
NODE  ARRIVE  DEPART  ROUTE USED  LINK
      TIME    TIME    TO ARRIVE   NUMBER
--------------------------------------------
  4           7:17
 21    7:19   7:19    WAL           45
 20    7:29   7:30    A4            43
 22    7:40   7:47    B5            48
  5    7:49   7:49    WAL            7
WEIGHTED TRIP TIME IS
 20 + 1.9 X   4 + 2.5 X   8 +  5 X  1 =  52.6 MIN
NODE  ARRIVE  DEPART  ROUTE USED  LINK
      TIME    TIME    TO ARRIVE   NUMBER
--------------------------------------------
  4           7:32
 21    7:34   7:34    WAL           45
 20    7:44   7:45    A4            43
 22    7:55   7:55    B5            48
  5    7:57   7:57    WAL            7
WEIGHTED TRIP TIME IS
 20 + 1.9 X   4 + 2.5 X   1 +  5 X  1 =  35.1 MIN
```

The data for plotting variation of quickest or optimal trip times with arrival time can be obtained by running [Plota]:

```
RUN PLOTA
 ENTER ORIGIN AND DESTINATION ZONE NUMBER? 4,5
 QUICKEST OR OPTIMAL TRIP TIME (Q/O)? O
PROGRAM TERMINATED
```

Data for plotting the variation of weighted trip time with arrival time for trips from zone 4 to 5 is output to DSETA:

Arrival time (hour)	Weighted trip time (min.)	Arrival time (hour)	Weighted trip time (min.)
7.700	35.1	9.150	48.9
7.817	52.6	9.200	56.4
7.817	52.6	9.200	35.1
7.950	72.6	9.450	72.6
7.950	35.1	9.450	35.1
8.150	65.1	9.483	40.1
8.150	48.9	9.483	40.1
8.200	56.4	9.700	72.6
8.200	35.1	9.700	35.1
8.450	72.6	9.817	52.6
8.450	35.1	9.817	52.6
8.483	40.1	9.950	72.6
8.483	40.1	9.950	35.1
8.700	72.6	10.150	65.1
8.700	35.1	10.150	48.9
8.817	52.6	10.200	56.4
8.817	52.6	10.200	35.1
8.950	72.6	10.450	72.6
8.950	35.1	10.450	72.6
9.150	65.1	10.483	77.6
		10.483	61.4

Using the above data set, a graph showing the variation of weighted trip time with arrival time such as those shown in Figure A.7 can be plotted.

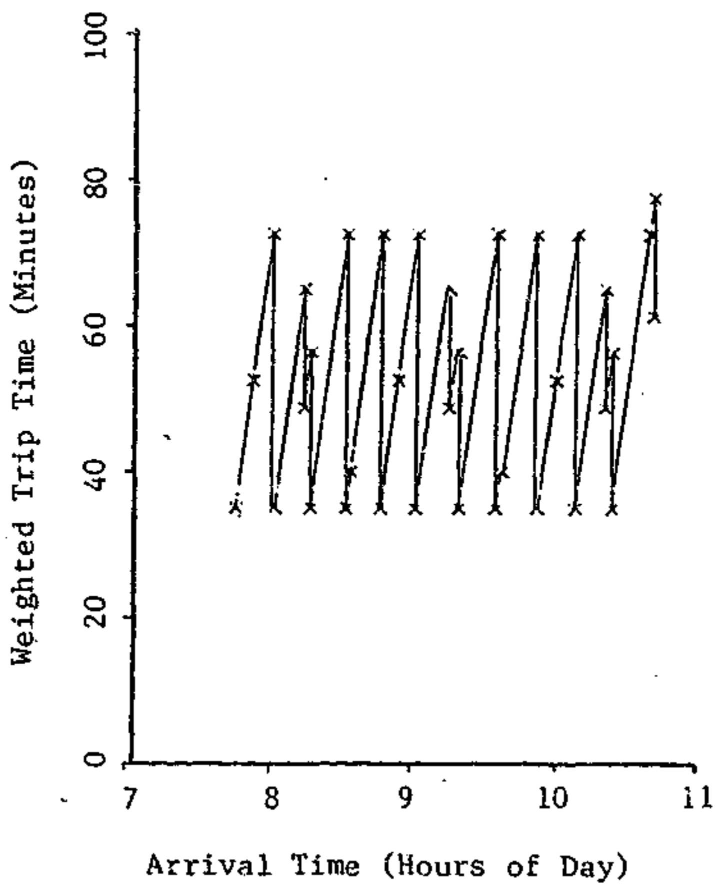
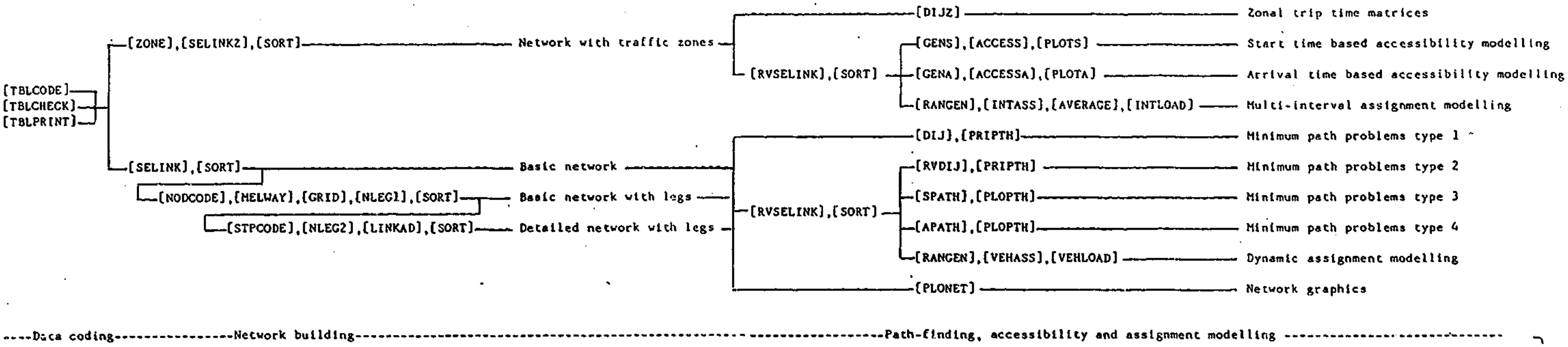


Figure A.7 Variation of weighted trip time with arrival time: zone 4 to zone 5

APPENDIX B

OPERATION SEQUENCE OF PROGRAMS

APPENDIX B OPERATION SEQUENCE OF PROGRAMS



APPENDIX C

PROGRAM FILE INVENTORY

APPENDIX C PROGRAM FILE INVENTORY

PROGRAM	INPUT FILES	OUTPUT FILES
ACCESS	STIM LINKS//N9 NODEID//N9 RVLINKS//N9	QTIM OTIM PATH
ACCESSA	ATIM LINKS//N9 NODEID//N9 RVLINKS//N9	QTIMA OTIMA PATHA
APATH	LINKS//N9 NODEID//N9 RVLINKS//N9	PATH
AVERAGE	LINKFLOW//NI	AVFLOW
DIJ	LINKS//N9 NODEID//N9	ARTIME FOR004
DIJZ	LINKS//N9 NODEID//N9	PATH//N9 QUKTIME//N9
GENA	LINKS//N9	ATIM
GENS	LINKS//N9	STIM
GRID	MELCODE NODEID//N9 MAPCODE	XYCODE//N9
INTASS	FLOWMAT//FI WEIFUN NODEID//N9 LINKS//N9 RVLINKS//N9	LINKFLOW//NI PATH//NI
INTLOAD	LINKS//N9 NODEID//N9 AVFLOW TIMETBL//N9	ILOAD
LINKAD	NODEID//M9 LINKT//N9 STOP ROUTE LEG2	LINKT//M9
MELWAY	MELCODE	MELCODE

PROGRAM	INPUT FILES	OUTPUT FILES
NLEG1	MELCODE XYCODE//N9 NODEID//N9 LINKS//N9	LEG1 LINKT//D9 XYCODE//D9
NLEG2	NODEID//N9 XYCODE//N9 LEG1 STOP	NODEID//D9 XYCODE//D9 LEG2
NODCODE	NODCOD	MAPCODE
PLOTA	QTIM OTIM	DSETS
PLOTS	QTIMA OTIMA	DSETA
PRIPTH	FOR004	
RANGEN		RAND WEIFUN
RVDIJ	RVLINKS//N9 NODEID//N9	FOR004 DPTIME
RVSELINK	LINKS//N9	RVLINKT//N9
SELINK	TIMETBL//N9	LINKT//N9 NODEID//N9
SELINKZ	TIMETBL//N9	LINKT//N9 NODEID//N9
SPATH	LINKS//N9 NODEID//N9 RVLINKS//N9	PATH
STPCODE	ROUTE STOP	ROUTE STOP
TBLCHECK	FOR003	
TBLCODE		FOR003
TBLPRINT	FOR003	FOR005
VEHASS	LINKS//N9 NODEID//N9 RVLINKS//N9 FLOWMAT//FI	NODEPATH VEHFLOW

PROGRAM	INPUT FILES	OUTPUT FILES
VEHLOAD	VEHFLOW TIMETBL//N9	TBLOAD
ZONE		ZONETM//N9