5.0 Alternative Numerical Solution Techniques

5.1 Introduction

One of the most important data required to obtain O-D trip table are link counts. Often link counts are obtained predominantly through loop detectors and pneumatic tubes. These counting devices are not completely reliable. They do have errors which leads to inconsistencies in link flows. This also occurs when link counts are obtained on different occasions. Hence one of the first steps, before the link volumes can be used to obtain the most likely trip matrix, is to make sure that the link counts satisfy link flow continuity.

To obtain an equivalent link count, which is as close as possible to the observed one and simultaneously satisfying link flow continuity, we try to minimize the error between the observed and estimated link flow, while maintaining flow continuity. After this preprocessing is complete one can go ahead and obtain the most likely trip matrix based on maximum likelihood techniques. There are two approaches to take into account the inconsistency in link counts and obtain the most likely trip matrix. These two approaches are discussed below. Further, some software packages which can be used to implement these approaches is are discussed.

5.2 Double Step Approach

In this approach, we break down the problem into two steps. In the first step we try to obtain link volumes which give link flow continuity, while deviating the least from the observed link counts. In the second step, the equivalent link volumes are utilized to obtain the most likely O-D matrix. Specifically, in the first step we minimize the error as shown in Equation 5.1.

Minimize: $Z(T_{ij}) = \sum_a \left(V_a - V_a' \right)^2 w_a \qquad \forall\, a$ (5.1)

Where:

V_a Actual observed link volume

V_a' Volumes that are closest to V_a and satisfy flow continuity

w_a Weighing factor, it could be *1.0 or* $\frac{1}{\sqrt{V_a}}$ *or* $\frac{1}{V_a}$

Such that

$$V_a' = \sum_{ij} T_{ij}\, p_{ij}^a \qquad \forall\, a \tag{5.1a}$$

Here we are solving for V_a'. To solve for V_a', subject to conditions in Equation 5.1a, the constraints in Equation 5.1a are multiplied by Lagrangian Multipliers. They are then included in the objective function, and the partial derivatives are set to 0. Solving this set of equations gives us V_a'. Once the equivalent link volumes are obtained, we can then proceed to obtain the most likely trip matrix based on the maximum entropy technique, as shown in Equation 5.2.

Maximize: $Z(T_{ij}, t_{ij}) = \frac{T!}{\prod_{ij}(T_{ij}!)} \prod_{ij} \left(\frac{t_{ij}}{\sum_{ij} t_{ij}} \right)^{T_{ij}}$ (5.2)

Such that

$$V_a' = \sum_{ij} T_{ij}\, p_{ij}^a \qquad \forall\, a \tag{5.2a}$$

As the objective function in Equation 5.2 is always positive, the natural logarithm of the Equation 5.2 could alternatively maximized as shown in Equation 5.3, subject to the constraint in Equation 5.2a. There are couple of advantages in doing so. Firstly, the probability is often an extremely low value, depending on the size of the network, and link flow on it. The larger the network and the link flow, the smaller the value. Secondly it is difficult to obtain the derivatives of factorials. Since, automation of the complete

process is also needed, the range of values which likelihood can take might fall below the smallest positive real number which a computer can store. To avoid this, we take the natural logarithm of the Equation 5.2 and maximize it, doing so still does not change the solution outcome.

$$\text{Maximize: } Z(T_{ij}, t_{ij}) = Ln\left(\frac{T!}{\prod_{ij}(T_{ij}!)} \prod_{ij} \left(\frac{t_{ij}}{\sum_{ij} t_{ij}} \right)^{T_{ij}} \right) \tag{5.3}$$

Expanding Equation 5.3 using Stirling's approximation we get Equation 5.4, subject to the constraints in Equation 5.2a.

$$\text{Maximize: } TLn\left(\frac{T}{t}\right) - T - \sum_{ij}\left(T_{ij} Ln\left(\frac{T_{ij}}{t_{ij}}\right) - T_{ij} \right) \tag{5.4}$$

Such that

$$V_a' = \sum_{ij} T_{ij} p_{ij}^a \qquad \forall\, a \tag{5.4a}$$

Where:

$$T = \sum_{ij} T_{ij} \tag{5.4b}$$

$$t = \sum_{ij} t_{ij} \tag{5.4c}$$

In order to solve the above problem we take Equation 5.2a in to objective function 5.4, by introducing the Lagrangian Multiplier as shown in Equation 5.5.

$$TLn\left(\frac{T}{t}\right) - T - \sum_{ij}\left(T_{ij} Ln\left(\frac{T_{ij}}{t_{ij}}\right) - T_{ij} \right) + \sum_a \lambda_a \cdot \left(V_a' - \sum_{ij} T_{ij} p_{ij}^a \right) \tag{5.5}$$

To get the most likely trip matrix we need to partially differentiate the above equation with respect to T_{ij} and λ_a and set the resulting derivatives to zero and solve for T_{ij} and λ_{ij}. If a network has 'm' O-D pairs and there are 'n' links on the network, we have a total of 'm+n' equations. The 'n' equations obtained by differentiating with respect to λ_a are linear, where as the other 'm' equations that are obtained by differentiating with respect to T_{ij} are non-linear in nature.

5.3 Single Step Approach

In this approach we try to obtain link volumes which give link flow continuity. The first step is to obtain a set of link flows, which give us flow continuity at nodes, and which are as close as possible to the observed link flows. In Equation 5.6 we minimize the error subject to constraints in Equation 5.6a.

$$\text{Minimize: } Z(T_{ij}) = \sum_a \left(V_a - V_a' \right)^2 \qquad \forall\, a \tag{5.6}$$

Where:

V_a Actual observed link volume

V_a' Volumes that are closest to V_a and satisfy flow continuity

Such that

$$V_a' = \sum_{ij} T_{ij}\, p_{ij}^a \qquad \forall\, a \tag{5.6a}$$

Using the constraint in Equation 5.6a, and substituting back in Equation 5.6, we get Equation 5.7.

$$\text{Minimize: } Z(T_{ij}) = \sum_a \left(V_a - \sum_{ij} T_{ij}\, p_{ij}^a \right)^2 \qquad \forall\, a \tag{5.7}$$

In order to solve the above unconstrained problem we partially differentiate the equation with respect to T_{ij} and set it to zero we get a system of equations as shown in Equation 5.8. Solving the set of linear equations will give us the equivalent link flow.

$$0 = 2\left(\sum_a \left(V_a \cdot p_{ij}^a \right) - \left(\sum_a p_{ij}^a \left(\sum_{ij} T_{ij} p_{ij}^a \right) \right) \right) \qquad \forall\, i, j \tag{5.8}$$

In order to obtain the most likely matrix, we maximize Equation 5.9 subject to conditions in Equation 5.8. Thus we are enforcing or making sure that flow continuity is satisfied irrespective of whether the observed link flow satisfies the flow continuity. If flow continuity is not satisfied Equations 5.8 will ensure that an equivalent link flow (V_a') is obtained which satisfies flow continuity.

$$\text{Maximize: } Z\left(T_{ij}, t_{ij}\right) = \frac{T!}{\prod_{ij}\left(T_{ij}!\right)} \prod_{ij} \left(\frac{t_{ij}}{\sum_{ij} t_{ij}} \right)^{T_{ij}} \tag{5.9}$$

As Equation 5.9 denotes likelihood, it is often an extremely low value. The objective function in Equation 5.9 is changed to the natural logarithm of Equation 5.9, as shown in Equation 5.10. The reasons for doing was already explained in the double step approach.

$$\text{Maximize: } Z\left(T_{ij}, t_{ij}\right) = Ln\left(\frac{T!}{\prod_{ij}\left(T_{ij}!\right)} \prod_{ij} \left(\frac{t_{ij}}{\sum_{ij} t_{ij}} \right)^{T_{ij}} \right) \tag{5.10}$$

Using Stirling's approximation for $Ln(x!)$ the objective function reduces to Equation 5.11.

Maximize: $$TLn\left(\frac{T}{t}\right) - T - \sum_{ij}\left(T_{ij}Ln\left(\frac{T_{ij}}{t_{ij}}\right) - T_{ij}\right) \qquad \forall\, i,\, j \tag{5.11}$$

Where:

$$T = \sum_{ij} T_{ij} \tag{5.11a}$$

$$t = \sum_{ij} t_{ij} \tag{5.11b}$$

Now we need to maximize the Equation 5.11 subject to the Equation 5.8. Since Equations 5.8 are equality constraints, we can take them in the objective function using Lagrangian multipliers, as show in Equation 5.12.

$$L \equiv TLn\left(\frac{T}{t}\right) - T - \sum_{ij}\left(T_{ij}Ln\left(\frac{T_{ij}}{t_{ij}}\right) - T_{ij}\right) + \sum_{ij}\left(\lambda_{ij} \cdot 2\left(\sum_{a}\left(V_a \cdot p_{ij}^{a}\right) - \left(\sum_{a} p_{ij}^{a}\left(\sum_{ij} T_{ij}\, p_{ij}^{a}\right)\right)\right)\right) \qquad \forall\, i, j \tag{5.12}$$

To solve the problem we now need to partially differentiate the above Equation with respect to T_{ij} and λ_{ij}, and solve the resulting set of non-linear equations. Differentiating with respect to T_{ij} will lead to non-linear equations, whereas differentiating with respect to λ_{ij} will lead to linear equations. If there are a set of 'n' O-D pairs, then the number of equations to be solved will be '2n', 'n' of them being linear, and the other 'n' equations being non-linear. Solving these equations gives us the most likely O-D trip matrix, taking into consideration the fact that observed link counts may or may not always satisfy flow continuity.

5.4 Solver Considerations and Attributes

There are many software packages which are available to solve systems of linear as well as non-linear equations. These packages could be used to solve the equations which are obtained after partial differentiation of Equations 5.5 and 5.12. There are limitations

when we utilize these packages and solve real life networks. There are computer considerations which need to be taken care of due to the properties of the hardware of the computer. Then there are mathematical considerations which need to be addressed so that a solution is obtained, some of which are discussed below.

5.4.1 Computer Considerations

I. Computational Time

Computational time typically is directly tied to the number of number of variables one is trying to solve for. For example, a network having a 2000 links and 200 O-D pairs generates about 2200 equations with 2200 variables if we solve it by the double step approach, or 400 equations with 400 variables if we solve it by single step approach. The larger the network, the greater the number of variables and equations, and hence the longer the computational time. The computational time is also affected by the algorithm used to obtain the solution. An efficient one will be able to get to the solution faster than an inefficient one. Some packages also have an upper limit on the number of decision variables, which severely constraints the usability of the packages when dealing with larger networks.

II. Memory

Memory requirements are one of the most important attributes of any solver. Here we are talking about the Random Access Memory (RAM) of the system on which the solvers occupy in order to execute a process. In fact the computational time is correlated to the RAM. When we have a large RAM the system is able to do the computations quickly, but it slows down dramatically if the computations increase. This is due to the fact that computations are done in chunks, which reduces its efficiency, due to memory limitation. Hence it is imperative that we choose the right type of computer with the required memory and the appropriate package to solve the problem. For want of increased speed

or lesser computational time often there is restriction on number of variables or sizes of networks.

5.4.2 Mathematical Considerations

Most of the algorithms used to solve non-linear equations involve iterative methods to get to the final solution. Here, the use of a suitable starting solution plays an important part. Depending on the algorithm used, as well the starting solution, one may or may not converge to a solution. If the initial starting solution is poor we could end up with solutions being complex numbers which do not have physical meaning in real world. Initial starting solutions play an important part especially when we have multiple solutions.

5.5 Alternative Solvers and Packages

Three software's were utilized to solve these equations. Some limitations and advantages are discussed below.

5.5.1 Microsoft Excel Solver

Microsoft Excel Solver is an add-in which is marketed along with the standard package Microsoft Excel. One can solve both maximization and minimization problems using it. To solve linear or non-linear equations the setup is changed to solve them. Although the Excel Solver in version 8.0 is not suited to solve non-linear equations the up coming versions are expected to solve non-linear equations. Microsoft Excel Solver uses the Generalized Reduced Gradient (GRG2) nonlinear optimization code developed by Leon Lasdon, University of Texas at Austin, and Allan Waren, Cleveland State University (Microsoft 1997). The initial starting solution plays an important role when dealing with

non-linear equations. If a starting solution is poor the solver simply stalls midway. Further the GRG2 can handle only 200 variables, and hence cannot be used to solve practical networks. The variables are coded as cells whose values change in the spread sheet and the constraints are setup in a similar fashion. One could control the number of iterations and the precision to which the constraints need to be satisfied. Since it is easy to setup the problem for small networks, it has good potential to be used as a tool by students. Microsoft Excel version 8.0's accompanying Solver version was used for solving the various formulations discussed earlier in this thesis.

5.5.2 MATLAB

The name MATLAB stands for Matrix Laboratory. MATLAB is a technical computing environment for high-performance numeric computation and visualization, produced by The Math Works Inc (MATLAB 1998). It includes a number of subject specific toolboxes as well as a dynamic system simulation package, SIMULINK. Generally problems are coded in as M-files which are executed in a workspace. A separate toolbox for optimization is available which contains subroutines which solve non-linear equations. MATLAB is a powerful tool as it can solve simultaneous, transcendental, non-linear equations. It can take a large number of decision variables and is very fast.

MATLAB utilizes the Gauss-Newton algorithm as a default in solving non-linear equations. If the Gauss-Newton algorithm gives poor convergence MATLAB automatically switches to Levenberg-Marquardt (LM) algorithm. The LM algorithm is a modification of Gauss-Newton algorithm, which is much more robust and takes less computational time. MATLAB version 5.2 was used for solving the various formulations. Large networks could be solved using MATLAB, before doing so the problem must be coded in properly as an M-file to be executed. For large networks automating this process would be essential.

5.5.3 QUEENS-OD

QUEENS-OD is a model for estimating origin-destination traffic demands based on observed link traffic flows, developed originally by Dr. Van Aerde (QUEENS-OD 1998) at Queen's University in Canada. It not only takes into account the observed link traffic flows but also observed link turning movement counts, link travel times and, potentially, additional information on drivers' route choices to estimate the O-D trip table. The model is capable of estimating both static and dynamic Origin-Destination (O-D) traffic demands. QUEENS-OD has been created to act as a support model tool for the INTEGRATION traffic network simulation model which was also developed by Dr. Van Aerde. The two models, therefore, share the same data file structure and file format. However, it is not necessary to have, or use, the INTEGRATION model in order to use the QUEENS-OD model to estimate origin-destination traffic demands.

The model implements a customized code to solve the single step formulation. Sample results in subsequent chapters of this thesis indicate that the solution obtained through QUEENS-OD matches solution for full formulation from MATLAB and MS-EXCEL Solver. This model is very versatile, and large networks could be represented and the O-D trip table could be obtained, depending upon the model size. QUEENS-OD release 2.0 (QUEENS-OD 1998) was used for the purpose of obtaining O-D matrices based on link counts.

5.6 Summary

In this Chapter two approaches were described first to obtain the most likely O-D trip table. The single step approach is more elegant as we need not deal with the problem of inconsistencies in link flow separately. It is built into the final equations, where as in the double step approach the problem of inconsistencies of flow is solved initially and then based on maximum likelihood the most probable O-D matrix is obtained. Another advantage of the single step approach is that it requires a relatively smaller number of

equations compared to the double step approach. This is a major advantage from a computational point of view.

The utility of some software packages which could be used to implement these approaches was also discussed. In the following Chapter various formulations to obtain the O-D trip table is compared, for two different networks. These formulations include the single step approach, which is implemented by QUEENS-OD.