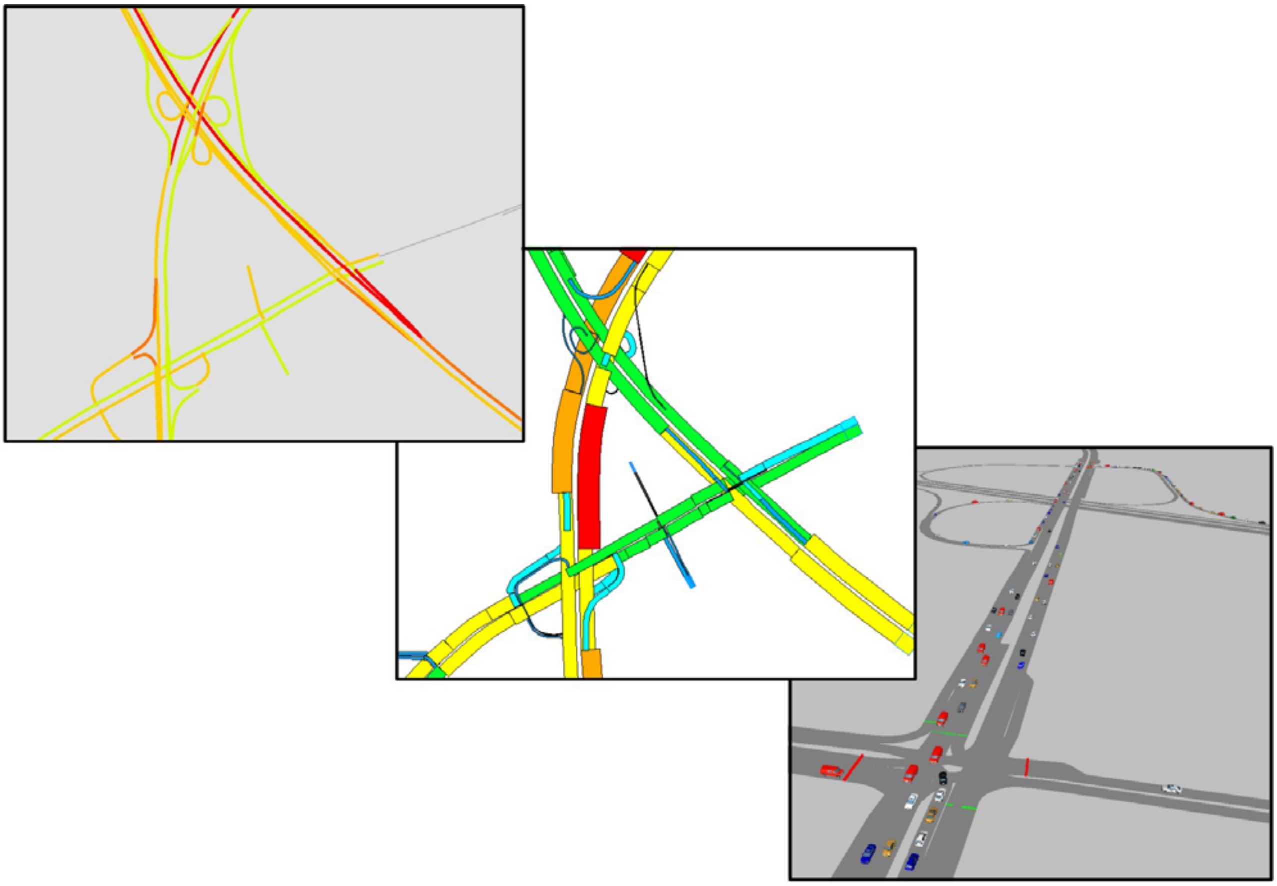
Date: 22-06-2007

Danmarks Tekniske Universitet

Macro-, Meso- and Micro simulation

- A comparison of three DTA softwares



Master thesis written by:

Rune Andreas Harsløf Holst
S041929

Center for Traffic and Transport
Danish Technical University

Abstract:

Micro-, meso- and macro simulation
– A comparison three DTA softwares

Rune Andreas Harsløf Holst,
S041929

Center for trafik og transport, 2007

I opgaven undersøges tre forskellige trafik modellerings redskaber, der tilsammen dækker områderne micro-, meso- og macro simulering.

- VISSIM fra PTV_Vision, som er en micro simulerings software, som benyttes af mange virksomheder til simulering af mindre netværk.
- Dynameq fra INRO, som sigter mod meso simulering af mellemstore trafikale netværk.
- Time Slice Assistant, som er en dynamisk version af den kendte makro simulerings software, Traffic Analyst fra Rapidis.

Som scenarie for afprøvningen valgtes et udsnit af det projektmateriale, som COWI og Vejdirektoratet har udarbejdet i forbindelse med udvidelsen af den eksisterende Motorring 3 fra Jyllingevej i syd til Hillerødmotorvejen i nord. For at inkludere det dynamiske aspekt i afprøvningen, indgik tillige en parallel strækning af O3 samt mellemliggende forbindelsesveje og portzoner.

De trafikale data blev uddraget ved filtrering af OTM-5 datasættet, hvoraf kun OD matricerne fra 3 tidszoner fra 05:00 til 09:00 blev anvendt.

For at teste de tre softwarepakker på lige vilkår, blev det trafikale test-scenarie modelleret så ens som forskelle i de tre programstrukturer tillader, og de tilbageværende programspecifikke forskelle blev ækvivalent omregnet så forsvarligt som muligt.

Foruden den systematiske afprøvning af de tre programmer omfatter opgaven en omfattende sammenligning af konkrete trafikdata som rejsetid, hastighed, flow, og forsinkelse for modellen som helhed og for en række udvalgte målepunkter.

Tillige udtrækkes sammenlignelige data til vurdering af modellens konsistens og af den indbyrdes konsistens mellem de tre programmers resultater.

På dette grundlag *diskuteres* konsistens og validitet af de fremkomne simuleringsdata,

Og der *konkluderes* sammenfattende, at der med Dynameq og især med TSA opnås en dramatisk reduktion i simuleringstid, samtidig med at de grundlæggende data, på basis af de deterministiske algoritmer fra disse to programmer udviser god overensstemmelse med de analytiske data, som er fremkommet ved detaljeret microsimulering i VISSIM.

Målt på en række andre parametre og sekundære programfeatures er der imidlertid store forskelle, som dog i høj grad må tilskrives, at de tre programpakker befinder sig på vidt forskelligt udviklingstrin.

Sammenfattende antydes en ændret rollefordeling mellem disse tre typer simuleringsværktøjer, som tidligere havde hvert sit klare segment, bestemt af netværksstørrelse.

I forhold til et naturligt workflow af analyse, problemfelt, løsningsforslag, modellering og dokumentation, har hvert program sine svagheder og styrkesider, ligesom forskelle i incitament og rolleforholdet indbyrdes mellem de naturlige aktører i dette virkefelt indikerer forskellige præferencer, når det kommer til valg mellem de afprøvede værktøjer.

Preface

This report is the result of a master thesis, by the title of “Micro-, Meso- and Macro simulation – a software comparison” and is made at the Center of Traffic and Transportation, at the Danish Technical University.

The goal of this report is to compare algorithms, performance and usability of three simulation software’s, based on a ‘meso optimal’ model area. Presented in the report is a comparison of:

VISSIM from PTV_Vision, being a micro simulation software optimal for small scale simulation and widely used by many companies.

Dynameq from INRO, mainly focusing on meso simulation of medium sized areas.

Time Slice Assistant, a dynamic version of the macro software, Traffic Analyst from Rapidis.

Thanks to

Internal guidance:	Prof. Otto Anker Nielsen, at CTT, DTU.
INRO:	Prof. Mike Florian, Centre Research on Transport (CRT), University of Montreal
	Senior scientist. Michael Mahut, INRO Montreal
CTT, DTU:	Stephen Hansen
Rapidis:	Bjarke Brun
COWI:	Søren Frost Rasmussen
Københavns amt, Teknisk forvaltning:	Jacob Ravn Jønsson

Software versions

VISSIM 4.16 (11)

Most computers were updated to patch 16 from 11 during the work with this thesis. Except for one computer this worked fine, if hadn't been for the fact that this was the fastest computer available and one serious problem with 4.11 was a memory leak when simulation batch-jobs.

The newest version 4.3 of the VISSIM software – which was not available for the project – might have presented solutions on the problems, discussed in the thesis.

Dynameq 1.2.1

Dynameq has several times been upgraded, both on the request of me, and through the regular process of beta-releases and upgrades.

Some problems has been identified during the thesis..

Delay values of turning-movements does correspond to HCM, but is addressed for wrong movements in the deafault settings.

Some functions still make Dynameq crash (outline color of links when visualizing queues)

TSA

TSA was used as a beta release, throughout the thesis. A larger problem is the lack of documentation.

Table of content

1 Introduction 3

1.1 Objective & Scope 5

1.2 Method 7

1.3 Quality control 10

1.4 Delimitation 11

1.5 Chapters of the report 14

2 Static Traffic Assignment (TA) 15

2.1 The 4 step model 15

2.2 Interrelation of speed, flow and density 16

3 Dynamic Traffic Assignment (DTA) 20

3.1 Route choice models 22

3.1.1 All or nothing (AON) 23

3.1.2 Wardrop's user equilibrium (UE) 24

3.1.3 Stochastic equilibrium (SE) 25

3.1.4 Stochastic user equilibrium (SUE) 25

4 The scenario 26

4.1 OD-matrices 27

4.1.1 Filter 27

4.1.2 Processing filter data 27

4.2 Traffic statistics 28

4.2.1 OD matrix structure in VISSIM 30

4.2.2 OD matrix structure in Dynameq 31

4.2.3 OD matrix structure in Time Slice Assistant 31

4.3 Processing the network 32

4.3.1 Infrastructure & portzones 32

4.4 Signal plans 34

4.4.1 Signal group plans 36

4.4.2 Green wave 36

4.4.3 VAP 38

5 Software specific implementation of DTA 40

5.1 Dynamic Traffic Assignment in VISSIM 40

5.1.1 General cost calculations 40

5.1.2 Route choice 42

5.1.3 Vehicle specifications 44

5.2 Dynamic Traffic Assignment in Dynameq 47

5.2.1 Path choice 48

5.2.2 Assignment 49

5.2.3 Traffic generation 50

5.2.4 Path travel-time and pruning 50

5.2.5 Gap and convergence 51

5.2.6 Traffic flow 52

5.2.7 Merge and intersections 54

5.3 Dynamic Traffic Assignment in Time Slice Assistant [13] 56

5.3.1 The Algorithm 57

5.3.2 Traffic on links 59

5.3.3 Capacity check and queue building 60

5.3.4 FIFO 61

5.3.5 Preferences 62

6 Results 63

6.1 Iterations *65*
6.2 Convergence *65*
6.2.1 Overall convergence 67
6.2.2 Distribution of travel time 70
6.3 Seeds *75*
6.4 Time consumption of convergence *78*
6.5 Totals *79*
6.5.1 Travel-time 80
6.5.2 Distance 81
6.5.3 Delay 81
6.5.4 Speed 81
6.6 Measurements *82*
6.7 BPR *83*
6.7.1 Speed and queue 86
6.8 Delay *91*
6.9 MAPS *93*
6.9.1 Speed 93
6.9.2 Delay 95
7 Discussion 96
7.1 Network Size *96*
7.2 Data preparation costs *97*
7.3 Modeling costs *98*
7.4 Simulation time *99*
7.5 Theoretical consistency *99*
7.6 Details of results *100*
7.7 Output validity *101*
7.8 Post processing features (Access and Excel) *102*
7.9 Presentation *102*
7.10 Post processing costs *103*
8 Conclusion 104
8.1 Initial considerations *104*
8.2 Core performance: Modeling and output validity *105*
8.3 The input process *106*
8.4 The output process *106*
9 List of figures: 108
10 List of tables: 109
11 References 110

1 Introduction

Simulation of traffic has been an expanding area for identifying impacts of future infrastructure projects. Both the demands for small projects like the design of a crossover, intersections etc. and larger infrastructure projects like bridge/tunneling projects, bypass-roads, planning of public transportation, calls for simulations to accurately predict the consequences.

Before computers were used to plan traffic, usually architects and city planners were hired to come up with estimates and possible solutions to a growing demand for infrastructural connectivity. But even if these projects were calculated and forecasted, still surprises from real life impact should be expected, since modeling could only give a rough indication for the future outcome.

With the introduction of computers, not only cad-softwares to shape and construct, but also simple simulation softwares to assess future impacts became an important source of information. Earlier, planners had to rely on intuition, knowledge and experience to ensure that the best possible solution was chosen. Now with the introduction of simulation software, alternatives and related modeling results are easily visualized to politicians and contractors.

The first simulation tool to be introduced was macro-simulation with a static traffic assignment, which gives a rough picture of generalized impacts, modeling only larger infrastructural implementation. Following increasing complexity in urban traffic patterns and time-based variations, results from the static traffic assignment (TA) became inadequate to fully describe changes.

Based on this technology, micro simulation was introduced, to simulate traffic on very restricted networks, with a very high level of details and explanatory results. The dynamic traffic assignment (DTA), divides the simulation period into time-slices, and thus gives a far more detailed picture of variations in the traffic situation.

As the computational power increased the networks of micro-simulation could be made more comprehensive, but even today, iterative simulation over larger areas may take days to complete. And since the costs of modeling and simulating can very well exceed the benefits for the client, the time spent on a project might have little or no value, or the answers might come too late in the process of solving temporary traffic problems.

There is no doubt that for simulating on both local- and vast regional areas, the preferred simulation tools have found their ground. As a rule of thumb the larger the area, the less comprehensive the level of resulting data will be. Thus between micro- and macro-level simulation is gap where a meso-level simulation is needed. One fit for medium-sized infrastructural projects, where the demand for more reliable results call for the implementation of dynamic traffic assignments in a less demanding total workflow, ending up in an acceptable cost and simulation time. With the introduction of meso-simulation, the void between lacking computer power and the need for details has recently been filled. Compared to micro-simulation, the dynamic aspect of the modeling in meso-simulation has recently been added to previously static macro-simulation software too.

The static traffic assignment (TA), works by the principle of letting all the traffic enter the model once every iteration, thus letting the traffic that enters the model experience the same travel-time regardless of departure time. Within each new iteration, a range of variables ensure that a part of the traffic searches for paths in the network, using an increased number of paths for getting from zone A to zone B, thus obtaining equilibrium. An obvious error in this approach lies in the fact, that most drivers will aim to reduce driving time, by choosing an alternate route, if heavy congestion is encountered.

In the dynamic traffic assignment (DTA) the simulation period is split into several periods, where calculated travel-time and cost are updated in the traffic scenario at regular intervals. This type of modeling thereby includes the incentive for some drivers to benefit from alternatives routes.

So when dealing with a near-capacity scenario of congested traffic, the results of the two approaches may be very different. While the TA only produces one set of results for the entire simulation period, the DTA produces a set for each calculation. This enables the user to monitor time-dependent changes, like queues, flow or travel-time thus identifying the impacts of an infrastructural initiative.

Since dynamic traffic phenomena like congestion, delay and poor viability are only too well known in urban traffic systems worldwide, the demand for modeling the dynamic aspect in the full scope of micro-, meso- and macro-systems has of course been a strong incentive for software developers to add these dynamic features to their existing modeling tools.

But even if the principle of time-slicing is generally the same for most DTA tools, the integration of the dynamic aspect might still differ considerably.

1.1 Objective & Scope

The objective of this Master’s thesis is to identify and describe characteristic principles for the dynamic aspect found in three different traffic modeling softwares, designed for the micro-, meso- and macro-field respectively. Furthermore, it will analyze the differences between these, in order to give a comprehensive comparison of their performance, in a chosen real-world scenario of Copenhagen infrastructure.

These three softwares are:

1) **VISSIM** is a micro-simulation tool that enables the user to follow each single car in real-time. The driving pattern of each individual car is generally influenced by the surrounding cars in the model and the rules for modeling this impact can be chosen by the operator.

After running the simulation, reports on issues like delay, queue-length, and travel-time are presented numerically, and thus call for some work before they can be interpreted, visualized or otherwise presented.

2) **Dynameq** is a quite new DTA tool, specifically designed to include the dynamic aspect in medium sized networks.

To reduce modeling complexity no single driver’s behavior is directly modeled, but incorporated into the model as a consideration to travelers’ individual behavior through the effect of overall interacting traffic.

Results are provided as graphic visualization of speed, flow, density, queue-length for each lane and movements in intersections. When visualizing the results, the selection and representation of these phenomena can be modified to meet demands, focuses and preferences. Furthermore most results can be exported as tables for further processing if necessary.

3) **Time-Slice Assistant** is a recently developed extension, adding a dynamic aspect to the static macro simulation toolbox, 'Traffic Analyst', designed for the ArcGIS environment.

So far no user interface has been supplied to handle the post-modeling process. The basic results are presented in tables only, and consequently these require some database processing before interpretation and presentation.

All three softwares use different DTA algorithms to calculate the equilibrium of the models. Thus the performances covering this area are compared.

The comparison of three simulations will focus on total convergence, travel-time and delay, and it is expected to give a general overview of the software performances. On a more detailed level a few OD-pairs will be compared on the basis of number-of-paths, flow-distribution, speeds etc. Finally, user-friendliness, pre- and post operations will be discussed.

1.2 Method

In choosing a scenario as a testing-ground for the three chosen DTA softwares, the dynamic aspect of course came into consideration, and the size and complexity of this is thus be chosen to fit the 'meso gap', previously mentioned between micro- and macro-simulation. Likewise the chosen scenario should allow for a consistent data-preparation and modeling in the three softwares, which are so similar that reasons and explanations for inconsistent results should be sought within the structure of the different modeling softwares.

In order to investigate reasons for software-specific differences in modeling results, the empiric basis of the testing-scenario should also be so consistent and well documented – in terms of topography, OD matrices and other in-data – that simple errors and other external factors could be identified and ideally eliminated.

Also the practical viability – in terms of data-conversion and pre-simulation-operations – called for a well-structured and well-documented urban traffic-scenario to make the modeling results valid and realistic.

In order to meet the requirements of such a medium-sized model, containing different road classes and a network that allows for several OD-paths, COWI and Vejdirektoratet were kind enough to provide a section of the network, used in the lane-expansion-project of Motorway 3 (M3), surrounding Copenhagen.

The empirical data used to carry out the preliminary assignment were the OD-matrices and network structure of the OTM-5 (Øresund Traffic Model).

In order to explicate the dynamic aspect, only the OD-matrices, containing the morning rush hour starting at 05:00 and ending at 09:00 have been selected for this testing scenario. In rough figures this time span includes one two-hour period before peak hours and two separate one-hour periods of max-level congestion.

Furthermore the data of the OTM-5 model has given the opportunity to make distinctions between several vehicle classes with differing preferences and attributes. Also the data-structure closely resembles the form and complexity of the softwares used. Since the basic network segment and the OD-matrices are the same for each of the three softwares, only the softwares add complexity to each model.

Figure 1.1: The scenario

The segment used in this project covers the motorway M3 from Jyllingevej in south to the crossing of Hillerød motorway in the north. Parallel to the

M3, the corresponding part of Ring 3 (O3) was admitted to the model to add some complexity by alternate route choices. Interconnecting the M3 and O3 is Jyllingevej to the south and Frederiksundsvej further north.

Building the three models for *VISSIM, Dynameq* and *Time Slice Assistant (TSA)* has throughout the process been a question of building three separate models within the capabilities of the softwares and at the same time obtaining three models so much alike, that results could be compared and conclusions be drawn. But since the objective has not only been to compare the dynamic modeling of three different softwares on the same scenario, the differences between these softwares must also be considered:

Since the M3 project was originally modeled in VISSIM to contain the highest degree of details, it was evident that the VISSIM model was to be the base-model, and the ArcGIS model was to be changed accordingly, subsequently making up the basis for Dynameq.

Basically the geometrical features of the roads, ‘turning movements’, number of lanes, etc. are made in similar ways in the three models. However, individual restrictions within the three softwares made it crucial to decide on a single approach, in terms of the features of the model. Both signal-control-plans, handling the intersections, and model features like driving-behavior, speed-distributions influencing the single driver, is kept alike in the models.

In the case of signal group plans, VISSIM was reduced somewhat to resemble the accuracy level of Dynameq. Also – in order to equalize the three models – some simplification of the DOGS program, increasing the ‘green times’ as traffic loads increase, had to be undertaken in the dynamic signal programming add-on to VISSIM, VAP.

1.3 Quality control

…and general reflections on the empirical data available for this survey:

1) The topographic data

Before proceeding with the data conversions for the three software-specific models, the basic network data from COWI / Vejdirektoratet were carefully checked against all known factors, using.

Arial photos[1]

Signal control plans[2]

Field observations

2) The OTM-5 data set

Comparing the empiric data from the OTM-5 model to observed reality has not been possible within the boundaries of this project, but it is generally assumed, that this data-set is the one coming closest to observable traffic behavior in the scenario.

The first version of the OTM model was produced in 1994/95. Since then the model has been used for several large-scale infrastructural projects and it has undergone continuous improvements. In the prior versions of the OTM model, the traffic was produced on the basis of traffic surveys dated back to the late 80’ and start 90’.[3]

To generate the most recent version OTM-5.0 from 2006, more than 2000 traffic counts, distributed throughout the OTM model area, were used to adjust the content by the method of ‘multiple path estimation method’ (MPME[4]). At the same time the number of zones was increased from 618 to 835 zones, to improve the resolution of the zone-structure. In combination with the OTM-5, ‘preload’ was introduced as the load that public transportation influences on the regular traffic.

Regarding accuracy, this range of changes has both improved the resolution and accuracy of the OTM-5 significantly compared to previous versions.

1.4 Delimitation

Within the framework of this thesis, a real-world scenario from Copenhagen traffic has been the testing ground, and a lot of attention has been paid to ensure the validity of topographic details in order to choose the best available dataset on contemporary traffic behavior, which has – in previous processes by other parties – been strongly optimized for resolution and accuracy.

Based on this, the process of selection and filtration of in-data for modeling has produced a very consistent data-set on the actual demand for traffic-flow, in the scenario.

In order to calibrate and verify the modeling results – including the dynamic aspect, from these three different software solutions – the natural thing would be to compare tables of measurements and aggregated results from the modeling to real-world traffic behavior.

In this area, several measuring points are established, visual surveillance is in fact undertaken, and dynamic phenomena like speed, traffic flow, congestion and queue lengths could very well be compared to testing results. Only time and costs would have to be considered.

However, it comes beyond the scope of this thesis to provide such documentation in order to validate modeling results. The consistency of modeling results is only tested for:

– Internal consistency for each of the chosen software solutions

– Relative consistency between the three chosen software solutions.

In chapter 8, ‘Results’ and in chapter 9, ‘Discussion’, these issues are addressed, comparing and judging on aggregated and visualized results, on a number of issues like convergence, speed, flow/density, travel time and delay, queue lengths and spill-back.

Only it should be kept in mind, that consistency between modeling results is only a preliminary test. Consistent results could still be a construction of poor validity.

When judging the relative consistency between the three chosen software solutions, roles are not quite evenly distributed between the three. The modeling principles of VISSIM come distinctively closer to the complexity of real-world traffic, since network details, single-driver-behavior and interaction between these can be closely controlled by parameters. Also reporting-facilities allow for a close look into the interrelations between these phenomena.

So in order to judge this relative consistency, VISSIM comes close to an adequate reference, when network-size allows for the full complexity to be included. In order to obtain the best possible reference, network size and level of detail has in fact been taken to the limits of processing power, program licenses and time available, and a lot of circumstances have been controlled as described in chapter 4, 'Method'.

In order to compensate – in this thesis – for the lack of access and opportunity to validate modeling results towards real-world traffic observations, VISSIM has gotten this reference role, which is in fact quite questionable. But while keeping this in mind, it is probably the best available reference within the limits of resources and access.

In the two other softwares, details are processed on a higher level of abstraction, where the construction, balance and calibration of algorithms would be issues of the highest importance. Necessary details and consistency to the phenomena of traffic reality could be lost in abstraction. And even if abstractions were well-considered and formally correct, software programming, user interface or reporting facilities might still be questionable.

For this thesis the prime approach for an evaluation has been the process of testing the three software solutions on even conditions in a well-chosen scenario. In the initial phases this has called for a systematized operational approach for modeling, in order to have similar conditions in the scenario, and potentially, in the next phase, giving opportunity for distinctions and analysis, where differences and inconsistencies might occur.

During this process a number of limitations have become much more influential, than initially expected, basically due to unavailable information and inadequate … or simply non-existing documentation.

This problem is not evenly distributed on a timeline for the three software solutions:

VISSIM has from the beginning been a well-implemented and documented tool, where only a few issues of questionable consequences of values had to be addressed in relation to parameters available on the user interface.

Dynameq. The access to the software was delayed in process, and further to the standard user-manual, access to further information has been available from the developers.

Even if this meso-simulation program has been implemented in a number of infrastructure projects, it is still in a process of further development and testing in a joint venture between.

TSA. The software was first considered a part of this master thesis half way into the writing of it, and information given on the program-structure and documentation for principles of modeling has been very sporadic.

The program-module is presently in an early phase of development and testing and no official program-release has yet been issued. Since basic information on parameters and algorithms has only to some degree become available during the last weeks of the process of analysis, a number of serious inconsistencies have remained unresolved and hard to explain.

1.5 Chapters of the report

- Chapter 2 and 3 describes the static- and dynamic assignment models.
- Chapter 4 describes the scenario including OD matrices and overall network actions.
- Chapter 5 addresses the implementation of dynamic traffic assignments.
- Chapter 6 presents the results and their significance.
- Chapter 7 discusses these results
- Chapter 8 concludes on the usage of each of the three softwares.

2 Static Traffic Assignment (TA)

In this chapter, basic understandings, distinctions and explanations supporting the following chapters are presented.

2.1 The 4 step model

To produce the scenario OD-matrices the static traffic assignment, was used in combination with the existing OTM-5 model. The TA differs basically from the dynamic model by assuming that travel-times are constant for the entire OD period, thus only launching the traffic once. The TA works through a process of 4 steps, with the possibility of loopbacks. The TA results in a stochastic user equilibrium, which optimizes each driver’s utility, within the limits of the stochastic preferences of the operator.

Trip-generation

In the trip-generation, the generation and attraction of each zone is transformed into a comprehensive list of trips from and to each zone.

Trip-Distribution

Based on a previous list, the trips from each zone are distributed between zone-pairs.

Mode-choice

The trips are distributed into modes of transportation. In this model 5 categories are used. (home-work, business, education/other, van and truck) where the first three categories represent car travel.

Car assignment

Through an iterative process, traffic-loads are distributed on routes, minimizing costs.

2.2 Interrelation of speed, flow and density

When relating traffic to the BPR curve (Bureau of Public Roads), one may see the correlation between driving in increasing congestion, and the speed/flow curve.

Mathematically the relationship is given by *Flow = Speed * Density*, but the interaction is more complex than that. Figure 2.1 shows how reducing speed, when density is increased and subsequently influences the Flow.

If considering the curve from V_{free} towards 0 km/h all drivers have experienced the fact that the traffic slows down as the flow and density increases, this instance is true as long as the maximum flow (Q_{max}) condition is not reached. After this point the density keeps increasing while both the flow and speed decreases.

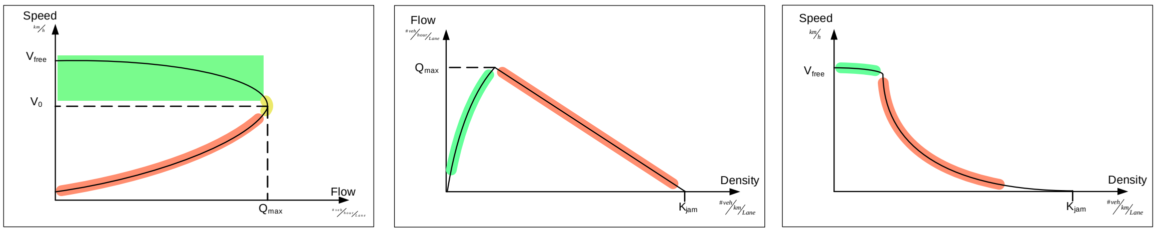


Figure 2.1: speed-flow, Flow-Density, Speed-Density

These three will be presented in full size in Figure 2.2, Figure 2.3 and Figure 2.4.

There is a long list of suggestions and calculations of how one determines the mathematical formulas for these speed-flow curves. Some is determined purely by empirical data, while others are based on the interaction between average headway as a function of the speed, which can be formulated to a relationship between speed and density.[10]

For traffic assignments in USA, the "link performance equation" also known as the BPR model (equation 2.1) describing the connection between speed and flow, is used. In 1973 the values of α = 0,15 and β = 4 was presented by the Federal Highway Association.[5] Later these values were revised (α = 0,2 and β = 10) to approximate observations made by the Metropolitan Transportation commission.[6]

$$V = \frac{V_f}{1 + \alpha \cdot \left(Q/C \right)^{\beta}} \qquad \textbf{2.1}$$ [7]

Instead of using the speed-flow curve the TA-model uses the equation 2.2 to derive the travel-time given by the Flow. In addition the factor differs by the introduction of the γ variable that takes into consideration the influence of the cars in the opposite direction.

$$t = \begin{cases} t_{free} \cdot \left(1 + \alpha \cdot \left(\dfrac{Q + \gamma Q_{Opposite\,direction}}{C} \right)^{\beta} \right) & ,\left(Q + \gamma Q_{Opp.dir} \right) < C \\ t_{Queue} & ,\left(Q + \gamma Q_{Opp.dir} \right) \geq C \end{cases} \qquad \textbf{2.2}$$ [8]

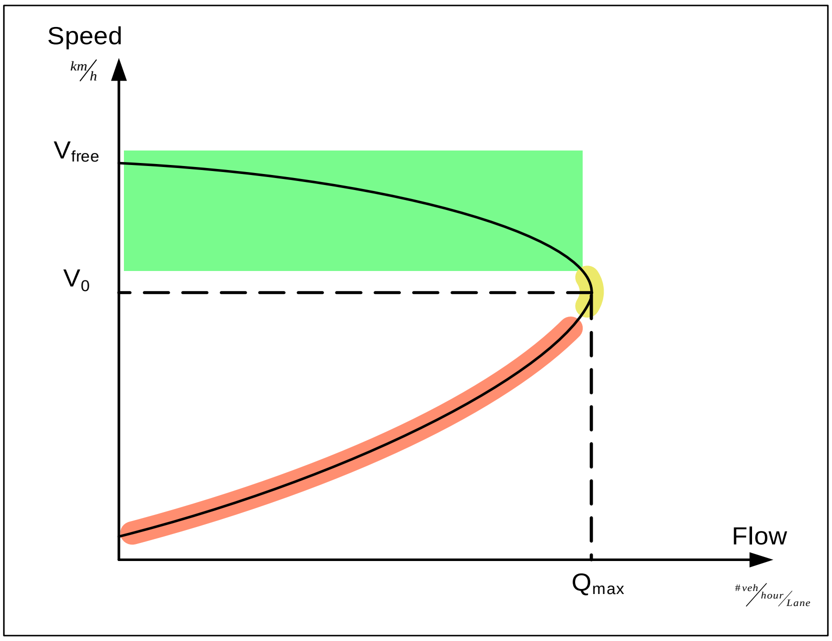


Figure 2.2: Speed / Flow curve: Free flow in green area, maximum flow in yellow area, congestion in the red area.

According to observed traffic, the speed-flow curve is partitioned into three states.

Free flow (green) is given by the upper part of the curve. This part of the curve starts at the V_{free}, usually representing the speed limit of the road.

Max flow (yellow) is represented the optimal flow-conditions in the area between free flow and congestion. The maximum possible flow-rate is expressed in the veh/hr/lane that a specific link can carry for a specific vehicle type.[9]

Congestion (red) is represented by the lower part of the curve, where the interactions between the cars begin to affect the speed and the flow. It is within the lower part of the curve that queues are formed.[10] At the bottom, flow is close to zero, while the density of the road is close to its maximum level.

In Figure 2.3 this is the outher most point at K_{jam}.

K_{jam} is the maximum # vehicles pr. km pr lane from each class.[9]

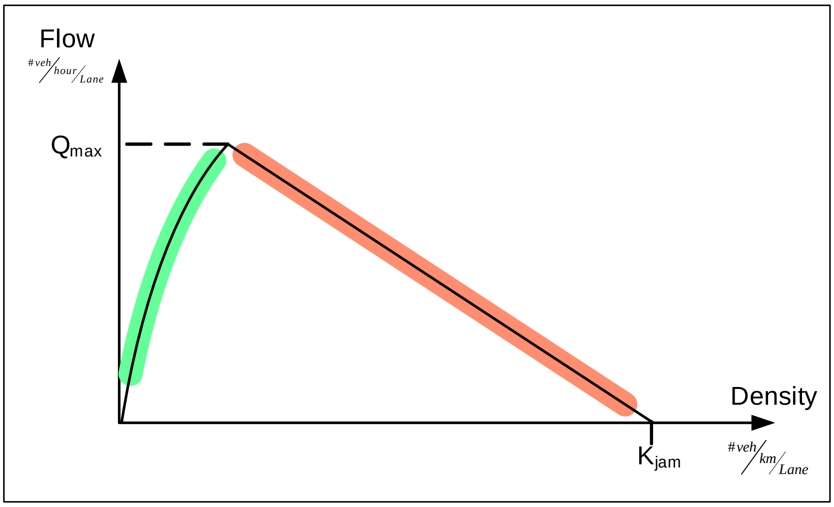


Figure 2.3: Flow / Density diagram

In the flow/density diagram see Figure 2.3 the level of congestion is given by density of veh/km/lane, increasing toward the level of traffic jam (K_{jam}), where the speed is zero. On the vertical axis the flow increases to a max flow (Q_{max}), which is also the breaking point between non-congested and congested conditions on the graph.

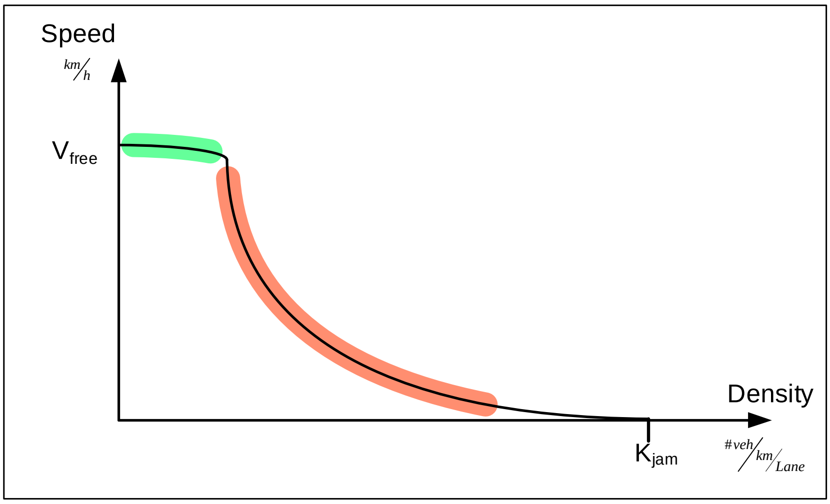


Figure 2.4: Speed / Density diagram

The speed-density diagram, see Figure 2.4, shows the relation between speed as a function of density. The free speed is maintained for non-congested conditions, while the speed is reduced non-linear. When jam density (K_{jam}) is reached, speed is, according to the graph, 0 km/h. Most DTA models are programmed to follow this set of curves to a lower speed. At this point the model changes to queue conditions moving along with a fixed speed. For TSA this value is 0,3*V_{free}.

3 Dynamic Traffic Assignment (DTA)

In the *static macro simulation*, the 4-step traffic assignment model is used. It includes trip generation, trip distribution, mode-choice and route assignment. Zone generation and attraction, on a disaggregated level is turned into constant flows following routes, at an aggregated link level. Because traffic flowing from origin to destination is assumed to be constant over a period of time, the model lacks the time-dependent dynamics of real traffic.[11]

In the *dynamic model*, the time is sliced into assignment- and calculation-intervals. This way, the variation in departure time and flow, influences the model over time, because a single trip occurs across several time-slices, during which traffic demand and network conditions can vary and congestion may occur.[12] Vehicles are proportionally assigned to a set of paths, depending on the travel-times of this interval. Travel-times are in the beginning only dependent on the free speed, but as traffic enters the model, the flow influences the travel-times. At this point it is the average travel-times for the vehicles leaving a link-piece that sums up the path travel-times. In order to replicate the drivers, learning about the network, and optimizing their travel-time, iterations move a fraction of the traffic from slower paths to the fastest, converging the model toward user equilibrium.

As the softwares are primarily used for different sized networks, they automatically fall into two different categories of dynamic modeling.

- Analytic: that mainly solves simulation on smaller networks.[13]
- Deterministic: Simulation-based models handles medium and large sized networks, but at the same time demands large amounts of computational power. With the introduction of MSA, research shows that deterministic dynamic route-choice-models can simulate relatively large networks.[13]

Basically the softwares use User Equilibrium as route choice model, where each driver’s utility is optimized. The use of stochastic take into account the fact that not all drivers have full knowledge of the network, and thus the stochastic element is introduced for both congested, uncongested conditions and as an overall value for the model.

	VISSIM	Dynameq	TSA
Basic model	Analytic	Deterministic	Derterministic
Route choice model	UE	UE	UE
Stochastics	Yes	No	Yes
Probability function		Logit	Probit
Other	No overlapping praths		

Table 3.1: Times for turning movements

In the following section, route-choice models will be dealt with in depth.

The difference between the Logit and Probit probability functions are that the Logit uses the constant coefficient and error term for all individuals, thou altering across alternatives, while the Probit function uses a normal distribution.[14]

3.1 Route choice models[15,16]

In describing the different route choice models, different factors are included in each of them. The overall conception is that all travelers search to optimize their travel-time (cost), and thus drive the model closer to equilibrium. Two basic approaches can be chosen: 1) Models that do not take cost-flow, (congestion), into consideration, and thus are easy to calculate and 2) models where the principle of cost-flow relationship is an important factor, describing how the flow on the roadway in relation to the capacities, influences the travel-time. See equation 3.1

$$t = t_f \left[1 + \alpha \left(\frac{V_a}{c} \right)^{\beta} \right] \tag{3.1}$$

Where

T = Congested travel-time

t_f = Free-flow travel-time

v_a = Link volume

c = Link capacity

α,β = Model parameters

With the introduction of stochastic, a variation between each driver's utility function and thus their knowledge of the road network, adds to a more realistic modelling.

Based on stochastic and congestion, four groups of models are formed.

	No-stochastic	Stochastic
No-congestion	AON	SE
congestion	Wardrop (UE)	SUE

Table 3.2: An overview of route choice models

3.1.1 All or nothing (AON)

“The traveler minimizes his or her deterministic travel-time”

In the “all or nothing” all travelers choose the shortest path based on speed and length, not taking into consideration any kind of delays due to traffic conditions. see equation 3.2. Further more all travelers have the same preferences and all have a complete knowledge about the network. In general this method will be used in the first iteration when no previous traffic has entered the model.

$$T_{ijar} = \max_{R}\left[U = -\sum_{a\in R} c_a\right] \qquad \textbf{All or nothing} \qquad \textbf{3.2}$$

Where
T_{ijar} = *Travel-time* between origin i and destination j, on link a on path r.
C_a = Cost of link a

3.1.2 Wardrop's user equilibrium (UE)

In 1952, Wardrop formulated 2 principles of equilibrium.

System equilibrium:

"Under equilibrium conditions traffic should be arranged in congested networks in such a way that the average (or total) travel cost is minimized"

User equilibrium:

"Under equilibrium conditions traffic arrange itself in congested networks in such a way that no individual trip maker can improve his/her path cost by switching routes"

Contrary to user equilibrium, in system equilibrium, the model searches to minimize the travel-time for the entire model. In order to obtain this, a fraction of travelers has to accept a non-optimal travel-time. The results of the system- and user-equilibrium usually differs, but by introducing factors like road pricing, preferences of every single traveler can change enough, to give closely related results.

See equation 3.3

$$T_{ijar} = \max_R \left[U = -\sum_{a \in R} c_a \right] \qquad \textbf{User Equilibrum} \qquad \textbf{3.3}$$

$$where \quad c_a = f(t, t_f, v, c, \alpha, \beta)$$

3.1.3 Stochastic equilibrium (SE)

In the stochastic equilibrium model, a certain stochastic variation is introduced to the utility or their knowledge of the network. [15]

“Every single traveler minimizes his or her perceived (stochastic) travel resistance”

$$T_{ijar} = \max_R \left[U = -\sum_{a \in R} \left(t_{a(0)} + \varepsilon_a \right) \right]$$

Stochastic equilibrium **3.4**

With the Logit function all routes are considered independent, and thus traffic is allocated equally on each path.[15]

3.1.4 Stochastic user equilibrium (SUE)

Using the method of successive averages in a combination with a Probit route function, SUE combines both the stochastic difference in utility functions and values for congestion in the network.

“Equilibrium is reached, when no travelers can reduce their deterministic travel resistance, by solemnly changing path”

$$T_{ijar} = \max_R \left[U = -\sum_{a \in R} \left(t_a + \varepsilon \right) \right]$$

Stochastic user equilibrium **3.5**

$$where \quad t_a = f\left(T_a, t_{a(0)} \right)$$

4 The scenario

The network used for this project is a small fraction, taken from the network of the entire OTM model, covering the greater Copenhagen region. The area (see Figure 1.1) was mainly chosen because of the access to VISSIM data covering the area. To ensure a correct traffic flow in the project network, a static traffic assignment was carried out in ArcGIS. Filters were introduced on the links leading in and out of the network, in order to isolate the traffic flowing in the network. In a series of query operations the traffic between filter zones were collected in a new OD-matrix ready to be converted into the proper format used in each of the three softwares. See Figure 4.1.

In the following section the process is described in depth.

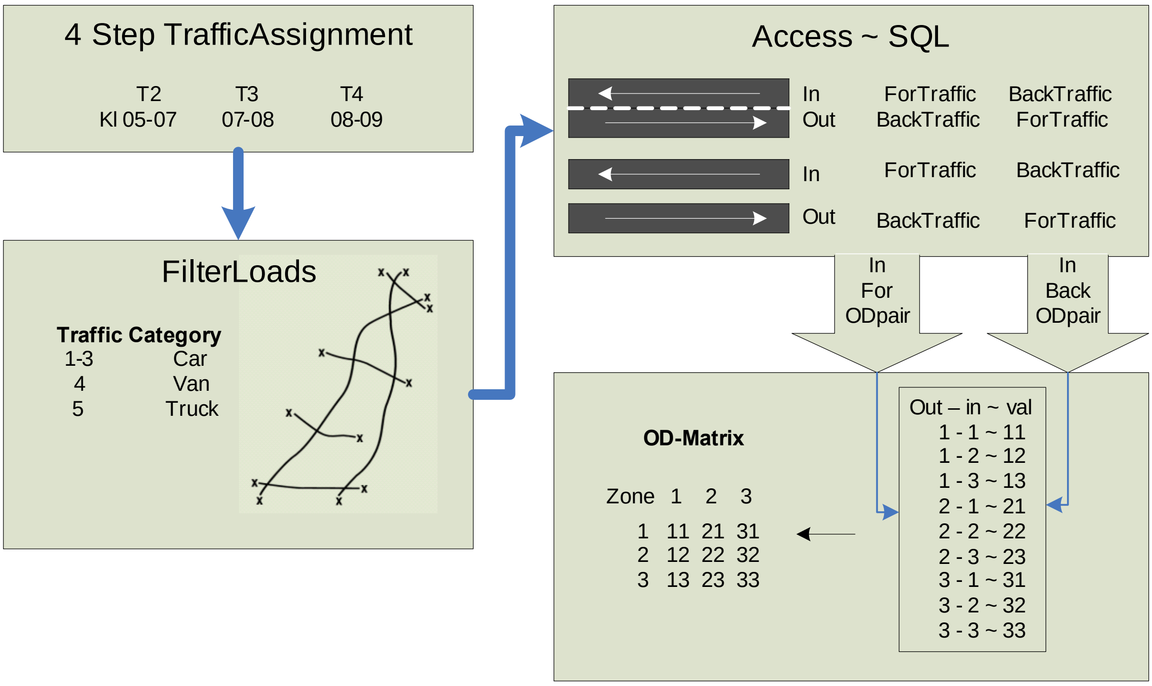


Figure 4.1: OD- matrix work process

4.1 OD-matrices

To represent the variation of traffic demand throughout the simulation period, three OD matrices were used.

1. kl: 05:00 to 07:00 (T2)
2. kl: 07:00 to 08:00 (T3)
3. kl: 08:00 to 09:00 (T4)

To isolate the traffic using the project network a filter was made to capture the Traffic entering the model on one link and exiting on all other links. By doing this for all links entering the model, no vehicles could enter the model without being registered.

4.1.1 Filter

In order to identify the relevant entries in the entire OTM data set, the filter uses the *filterID* to separate the different entries to the model. In order to ease the process of quality control and troubleshooting errors, *filterID* and *linkID* was given the same values. To separate the vehicle classes, the *filterID* was appended the *categoryID*, for van and truck. For cars, only the *linkID* were used.

ObjectID	FilterID	LinkID	FromZoneID	ToZoneID	CatID	(Category)
1	140545	140545	-1	-1	1	Home-work
2	140545	140545	-1	-1	2	Work-work (Business)
3	140545	140545	-1	-1	3	Other
4	4140545	140545	-1	-1	4	Van
5	5140545	140545	-1	-1	5	Truck

Table 4.1: Example of filter definition

4.1.2 Processing filter data

In processing the filter data it was important to identify whether the in- / outgoing traffic was flowing in the forward or backwards direction according to the digitized direction. This is important because the traffic flowing on each measured link is divided into a forward and a backward motion, which can be entering or exiting the model.

In order to capture the outgoing traffic, for each of the filter-links, two sets of queries were made to capture either backward- or forward flowing traffic, later these were merged to form a single basic OD-matrix for each time period and vehicle type.

Each of the three softwares had different syntax for their OD-matrix, thus Excel was used to transform the basic OD matrices into the required software-specific syntax.

4.2 Traffic statistics

For the simulation in the test-scenario, three time-frames, T2, T3 and T4, covering the periods (05:00-07:00, 07:00-08:00, 08:00-09:00), have been chosen from the OD-matrices of the OTM-5 data set. The three vehicle classes cover 5 classes of purpose. Three of these are for *car* while *van* and *truck* represents their own classes. The classes are established to individualize the costs and thus the utility of the trips made, for each purpose. Car purposes are divided into *home-work*, *business* and *education/other*. As the cost of making a business trip is larger than trips with other purposes, preferences will differ when choosing the route. While running the TA model to generate OD-scenario data, this separation was maintained.

After running the traffic assignment in ArcGIS, the filter showed, that a total of approx. 72000 veh. were entering the model during the 4 hours. During the first two hours, 05:00 to 07:00 only approximately 1/7 of the traffic was assigned, and the remaining load was almost evenly distributed on the following two hours.

Veh.	05 - 07	07 - 08	08 - 09	Total (class)
Car	7244	24499	25804	57546
Van	2636	4125	4045	10805
Truck	1234	1101	1278	3612
Total (time)	11113	29724	31126	71964

%	05 - 07	07 - 08	08 - 09
Car	65	82	83
Van	24	14	13
Truck	11	4	4

Table 4.2: Vehicles by time and class.

In Figure 4.2 and Table 4.2, the distribution between classes and time can be observed. Trucks are more or less constant, while vans double and cars triple in the hours after 07:00. In the scenario this gives a strong increase in commuting traffic right around 07:00, while in reality; a more gradual increase is a fact. In Figure 4.3 the increase can be observed following the blue line.

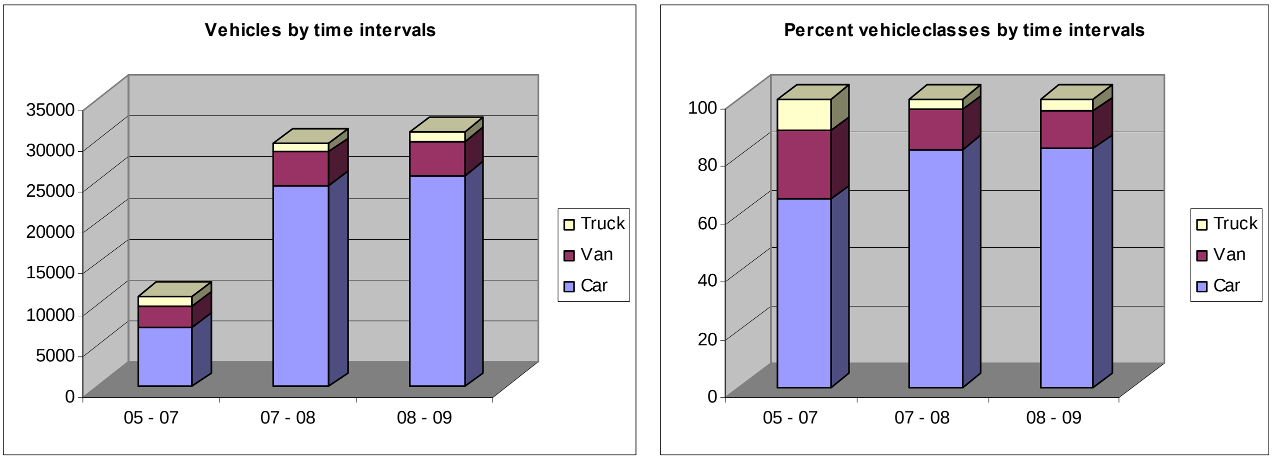


Figure 4.2: Vehicles visualized by time and class.

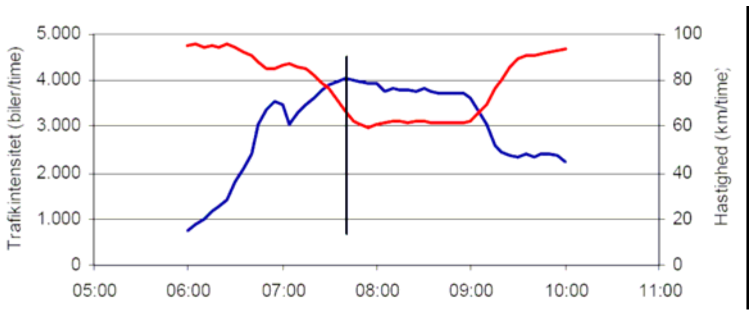


Figure 4.3: Traffic intensity (blue: cars/hour) between kl: 6:00 and kl: 10:00[17]

As an option the constant traffic of T2 could be simulated closer to reality; introducing much less traffic between 05:00 to 06:00 than in the hour between 06:00 to 07:00. See Figure 4.3. Also a finer division of this period would open for a closer imitation of the increasing rush hour traffic. But since this is a comparison between models and no real world data exists to form a basis against reality, the basic distribution of three time intervals is kept the same for the three softwares, and thus the basis for comparison is present.

4.2.1 OD matrix structure in VISSIM

VISSIM uses an OD-matrix for each vehicle class in each time-period. The OD-matrices are set up in Excel and converted to ASCII format. In the header, metadata for the matrix is noted.

- Time interval: Duration of OD-matrix
- Scale factor: Factor to scale the traffic in the OD-matrices
- Number of zones: Field for identifying the port zones

Following this information the actual OD-matrix is given. Each value represents the exact number of vehicles in that timeperiod. To ease the conversion, the temporary matrices were sorted, according to ‘Out’ then ‘In’. see Figure 4.4

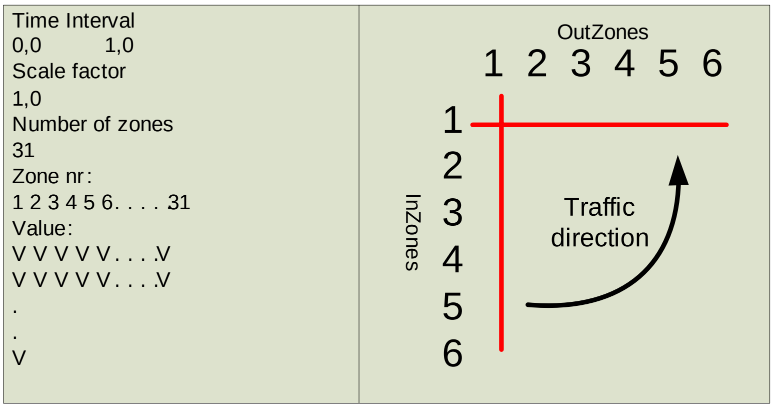


Figure 4.4: VISSIM OD-matrix structure

4.2.2 OD matrix structure in Dynameq

Dynameq refers to three matrix files for each vehicle class. But instead of importing three OD-matrices pr. vehicle class, all three are written in the same file, only split into slices representing the time intervals. In the Dynameq OD-matrix file, the metadata holds information on vehicle class, and total time interval. Following, the end-time-slice, OD data continues with values of veh/h.

In Dynameq the formatting is sorting on ‘In’ then ‘Out’.

```
Vehicle_class
Car
Time Interval
500
900
Slice 700
X Y val
X Y val
Slice 800
X Y val
X Y val
Slice 900
X Y val
X Y val
```

Figure 4.5: Dynameq OD-matrix structure.

4.2.3 OD matrix structure in Time Slice Assistant

As the TSA toolbox is developed based on the TA toolbox, the OD data is presented the same way. Thee different tables representing the time periods, contains *From-* and *Tozone*, *categoryID* for identifying each vehicle type. See Figure 4.6

```
[FromZone],[ToZone],[CategoryID],[value]
```

Figure 4.6: TSA OD - matrix structure.

4.3 Processing the network

In the process of developing the three models, it was important to make the models as equal as possible, while keeping the level of details on a balance between running time and complexity. Since real-time simulation in VISSIM is very demanding on computer power and memory, making the iterative process of dynamic assignment an extremely slow procedure, preliminary stress tests were carried out to determine whether the model had some room for expansion. The simulation time indicated a surplus for the following to be implemented:

- Dynamic traffic light control for intersections with more than one program
- 3 vehicle classes: car, van and truck.
- Expanding the infrastructure in areas, where traffic would naturally use secondary roads to avoid congested areas.

4.3.1 Infrastructure & portzones

Prior to identifying the port zones of the model, a revision of the existing OTM connectors was carried out. In general most of the connectors were already placed strategically correct, but in order to obtain data in the filter, some connectors had to be moved or established. In the areas between M3, O3 and Frederikssundsvej the zone structure and connector attachments left the area with no connection to the active network. One solution could have been to divide the existing zone, but this would call for a lot of work. See Figure 4.7. In most cases the connectors were attached to nodes making the use of the filter impossible. In these cases one small piece of link were made active and the connector was reattaches at the end of this new link.

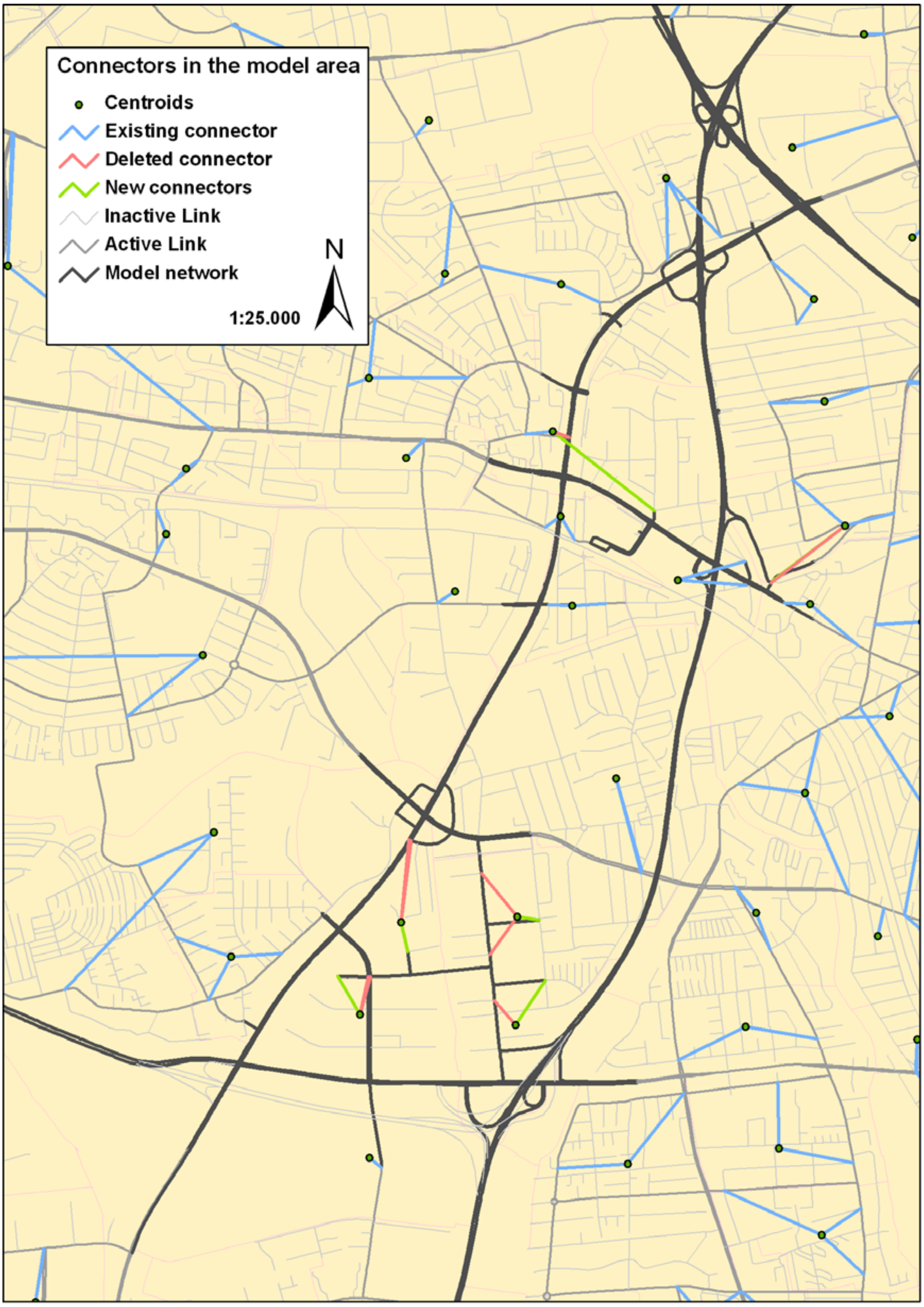


Figure 4.7: Connectors before and after

In the southern part of the model, Islevdalvej called for the movement of three connectors in order to identify the traffic streams. The solution was to activate pieces of Glerupvej and Krondalvej, and reattach the connector. The same was the case for Erhvervsvej and Produktionsvej.

4.4 Signal plans

Since the model covers several municipalities, several traffic light plans and cycle lengths, are used in the model. In general all intersections with fixed cycles are kept this way. Along O3, 8 of 11 intersections are programmed dynamically to change the cycles, according to traffic loads. 5 of these are controlled by the DOGS program. The remaining 3 dynamic intersections would in reality be trigged by sensors surrounding the intersection.

Each time the traffic increases beyond a certain level, DOGS increases the cycle time with increasing traffic. With a 10 sec. split, the program can increase the cycle time with a ratio of 8 sec. for the direction of O3 and 2 sec. for the secondary direction. In theory the cycle time can be increased from 80 sec. to 140 sec., but readings have shown that the system has only reached 120 sec. once. Thus 80 sec. and 120 sec. have been chosen as the implemented cycle times. Furthermore Figure 4.8 shows that the expansion roughly starts around 7:30 in the morning and continues for the remainder of the simulation period.[18]

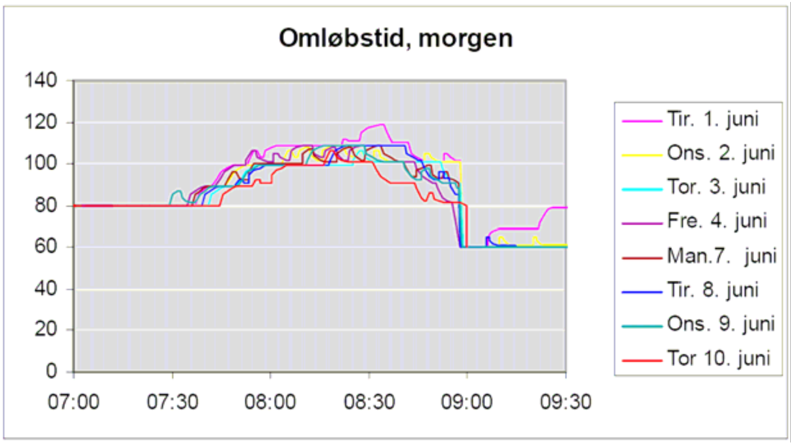


Figure 4.8: DOGS expanding cycles during the morning rushhour.

When implementing signal plans, at an early stage it was clear, that VISSIM was capable of a much more dynamic traffic light control than Dynameq.

VISSIM can implement VAP programming to adapt the light intervals to the current traffic situation. This dynamic way of adapting gives a big advantage when planning the infrastructure, because rising problems are dealt with at an early stage. The lack of dynamic signal/VAP programming in Dynameq, calls for a compromise. Dynameq uses signal plans that divide the timeframe. Each plan is global for the model, and since each intersection has individual phases assigned, the intersections in Dynameq are semi dynamic. The global Signal plan cannot respond immediately to the growing traffic, but it is possible to adapt a new signal plan for each new plan-period.

Because of this difference, the question arose, which approach would indicate the best results?

1. To give the two models individual light responding plans.
2. To simplify the light plan of VISSIM to match Dynameq.
3. To make a readout of the final light plan induced by the traffic in VISSIM, and copy this to the model of Dynameq.

#1) Traffic will not be handled the same way, thus the result could not be compared without taking this into consideration.

#2) The simulation of the traffic would not be completely realistic. Queues might form faster, bringing traffic to a standstill. Thus, changing the final pattern from what would be expected and causing unrealistic route choices, which could be difficult to explain.

#3) The most correct handling of the problem, would be to let the dynamic intersections follow the convergence of the scenario and then match the VISSIM phases to the static phases of Dynameq. In this case Dynameq would have to make up a new plan every 2-5 min. for four hours of modeling time. With 20 intersections, each with 2-3 phases, it would take a huge amount of time and increase the possibility of making errors.

Considering the fact that the results of the route choice models have to be compared and thus minimizing all exterior influence is of the essence, the second approach gives the least variance between the models, and is chosen for implementation.

4.4.1 Signal group plans

Applying two different signal plans In Dynameq was done without encountering any larger problems, but doing exactly the same in VISSIM turned out to go beyond the capabilities of VAP programming. To obtain the same result, the number of cycles was counted and the number being the key of altering the program.

4.4.2 Green wave

To secure a flow through the model, green waves were implemented, but since no data have been available, schemes were made to ensure a near optimal solution. Because of alternating distances between the intersections in combination with the periods of green, it is not always possible to prioritize both directions.

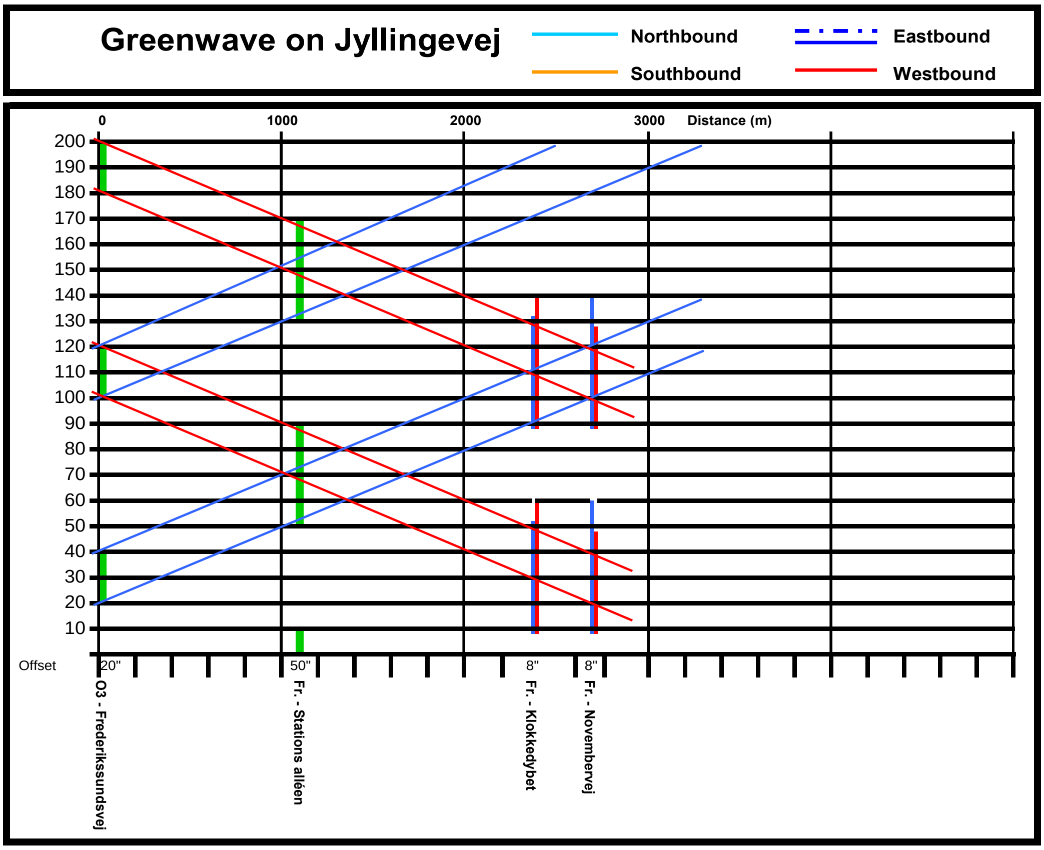


Figure 4.9: Green wave on Jyllingevej

Figure 4.9 shows the waves on O3 and Jyllingevej. On Jyllingevej the eastbound waves is prioritized because it for large parts consists of waves, while the westbound traffic is more or less constant. This part of the model is only influenced by the alteration of the intersection between O3 and Jyllingevej that adds 10 sec in the direction of Jyllingevej.

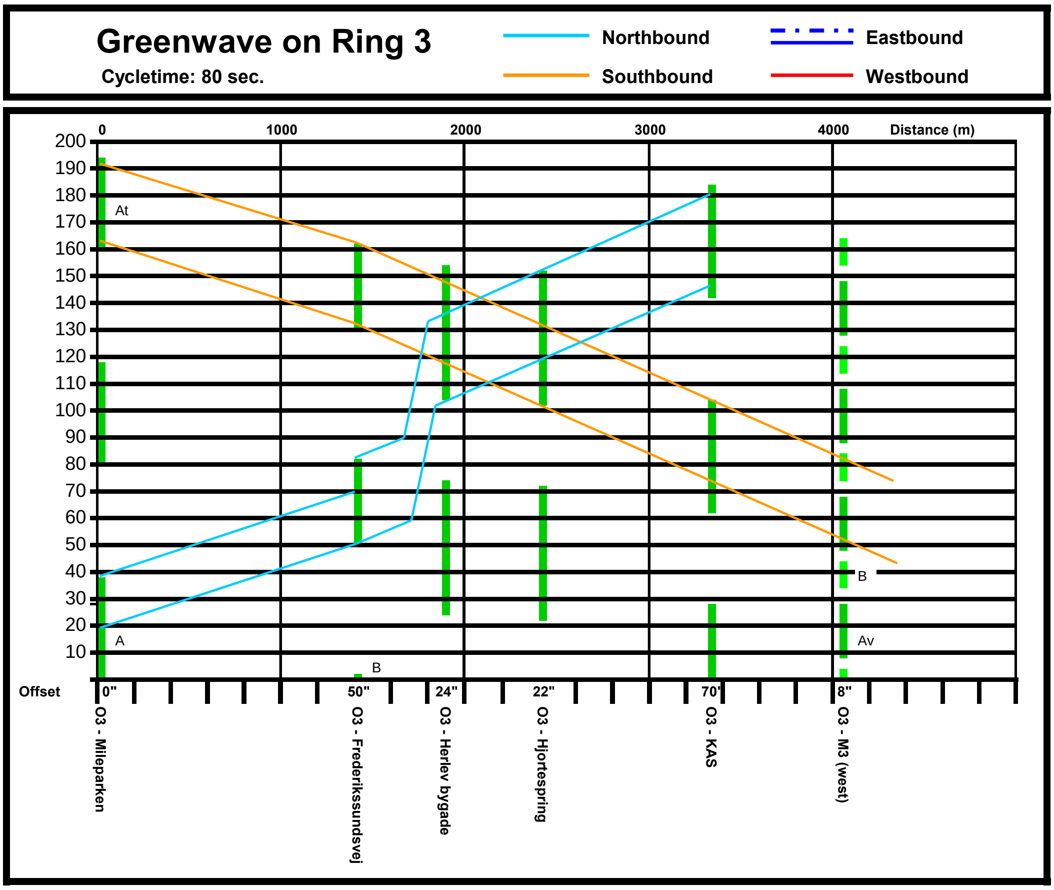


Figure 4.10: Green wave on Ring 3, at 80 sec. cycle.

On O3 the prioritized direction is mainly southbound. This priority causes delay in the northbound flow between Frederiksundsvej and Herlev Bygade, where only limited parts of the flow will have direct passage.

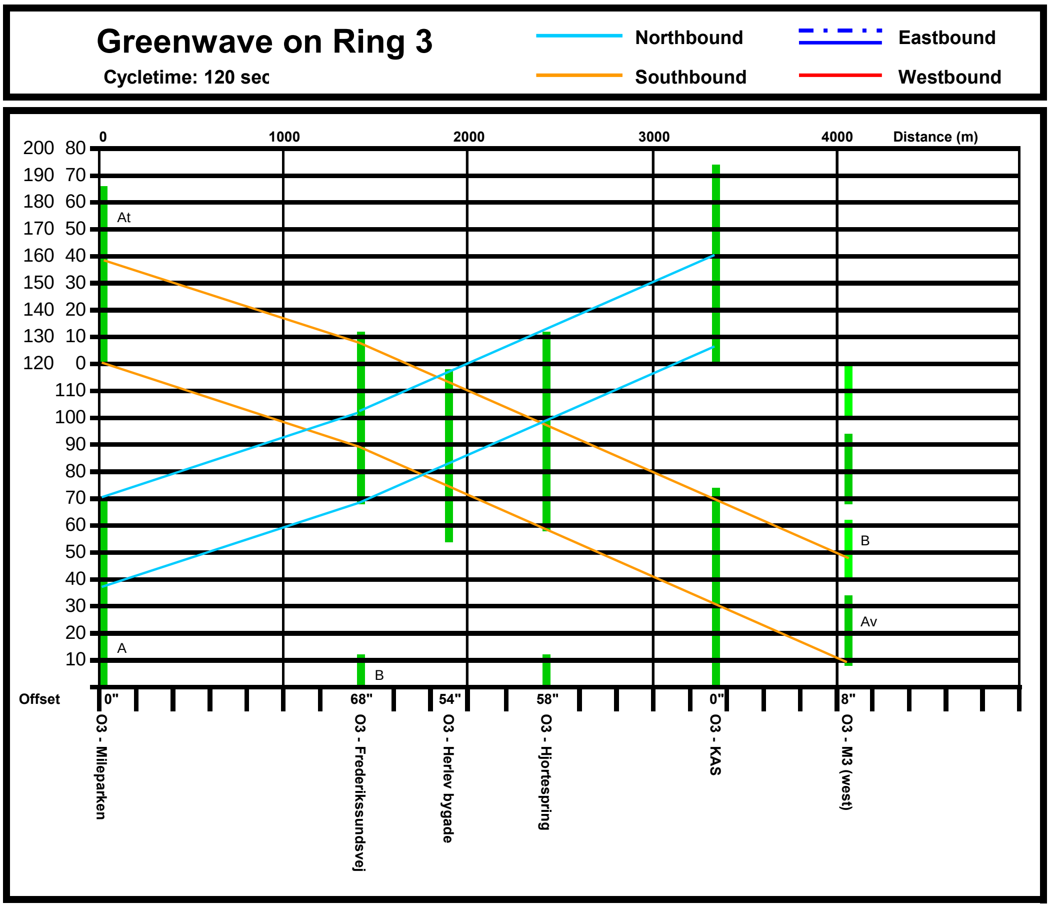


Figure 4.11: Green wave on Ring 3, at 120 sec. cycle.

The disruption of the flow diminishes when the extended cycle is introduced, allowing a larger flow in both north- and southbound directions.

4.4.3 VAP

In order to simulate the changes in cycle time, VISSIM uses the VAP programming. In general VAP is used, when sensors in the road are used to invoke some sort of change in the cycle time, due to a specific traffic situation. In this case, a program resembling the DOGS could have been programmed, but the difference between Dynameq and VISSIM would have influenced the results. Thus a solution replicating the capabilities of Dynameq was needed. Dynameq applies control plans that divide the timeframe at a specific time during the simulation. Resembling that model, would call for a VAP method that tested a variable against the simulation clock. Unfortunately this method is not part of the programming library. The solution, see Figure 4.12 was to count the cycles [Count = Count+1], compare them to a variable [count = 112] equal to the time of the first control plan divided by the first cycle time. At this time the cycle should change to a larger cycle time.

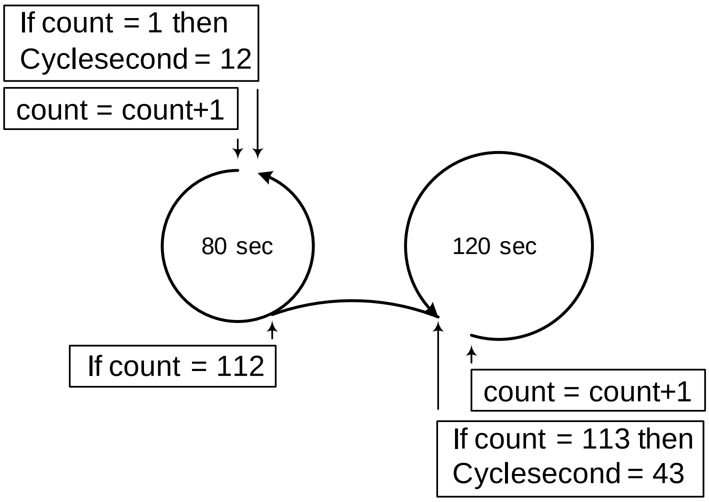


Figure 4.12: Schematics of VAP program.

A second problem was to make the offsets fit the green-wave diagrams. In each of the first cycles, the time of the secondary green-time was prolonged [Cycle-second = xx]. Potentially having a second cycle-second after the 80'th second, meant that the program's cycle-time – which is fixed, regardless of the programmed cycle-time – had to have the least common multiple of 240 sec, in order not to drift during the simulation.

5 Software specific implementation of DTA

5.1 Dynamic Traffic Assignment in VISSIM

The VISSIM road network model is very detailed in order to allow an exact reproduction of the traffic flow in high resolution of time and space. Travel-times, as a result of the micro simulation, are calculated for each edge for each evaluation interval. The dynamic traffic assignment refers to a more abstract model consisting of edges and nodes, arranged in a graph. This way the DTA can work much more efficiently, only taking into consideration the general cost, when allocating traffic to the set of paths chosen.

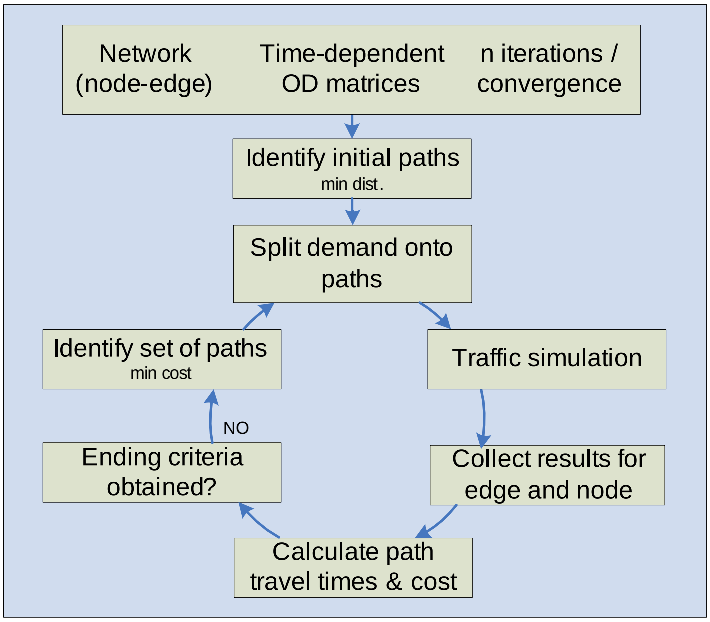


Figure 5.1: Principle of dynamic traffic assignment for VISSIM

5.1.1 General cost calculations

The General cost calculations for each edge in the graph is used as a basis for route search and -choice. General cost is made of travel-time being a traffic dependent factor, travel distance being path dependent and other costs like tolls are added to this.

$$GC = \alpha \cdot travel\,time + \beta \cdot travel\,dist. + \gamma \cdot financial\,costs + \sum other$$ **General Cost** **5.1**

Where α, β, γ is user defined.

In order to calculate the route search and -choice, the travel-time (see equation 5.2) that make up an important part of the general cost, is based on the results in the evaluation interval in the previous iteration. To determine the travel-time used in general cost, VISSIM makes use of both measured travel-time and a smoothed expected travel-time from the previous iteration. See Figure 5.2.

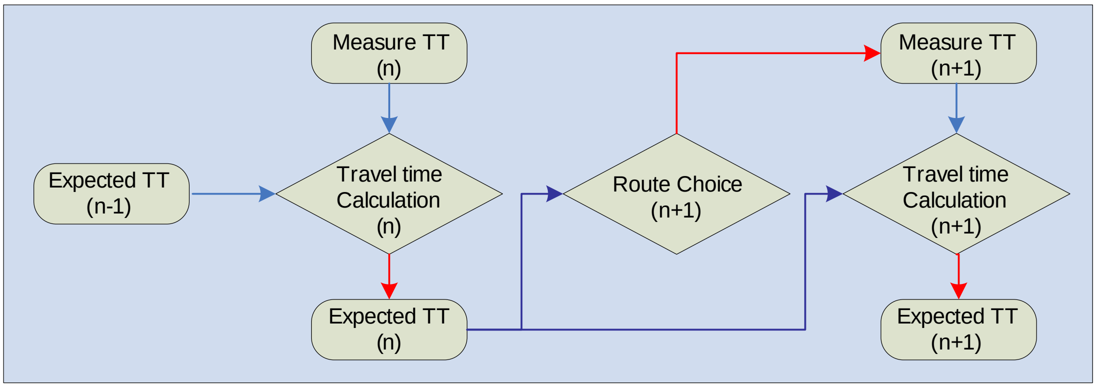


Figure 5.2: Travel-time calculation

$T_a^{n,k} = (1-\alpha)\cdot T_a^{n-1,k} + \alpha \cdot TO_a^{n,k}$	**Travel-time calculation**	**5.2**

Where

$T_a^{n,k}$ = Expected travel-time

$TO_a^{n,k}$ = measured travel-time

K = Evaluation interval

a = edge

α = smooth factor

n = Iteration

VISSIM uses a smoothening factor (α = 0,5), to calculate the expected travel, using 50% of the previous expected travel-time. This influence is inherited through iterations. The inheritance of the values diminishes with 50% for each previous iteration, thus calculating expected travel-time for n, the influence would be T(n-1) 50%, T(n-2) 25% …and so on.

5.1.2 Route choice

For a given set of discrete alternatives the probabilities for the alternatives to be chosen must be determined, due to comparable values. In order to prioritize more attractive (less cost) paths, general cost is converted to utility for each path, consisting of edges and nodes. See equation 5.3.

$$U_j = \frac{1}{C_j}$$ **Utility function 5.3**

Where
U_j = the utility of route j
C_j = the general cost of route j

To calculate the probability distribution on each single path the basic logit function can be used (see equation 5.4). Combined with the utility function, the Logit function leaves a model with the drawback, that it can't distinguish the importance of the 5 min between 5 to 10 min of travel-time from that of 105 to 110 min. This holds against the human perception of shortest/fastest route, where the gain in speed/time is experienced relative to the total.

$$p(R_j) = \frac{e^{\mu U_j}}{\sum_i e^{\mu U_i}}$$ **Logit function 5.4**

If this difference is to be prioritized, a logarithmic transformed Kirchhoff distribution formula in combination with Logit function can be used.

The Equations 5.5 show the Kirchhoff functions, while Table 5.1, shows the difference of the results.

k is a sensitivity factor that determines how much influence the difference between utility is.

$$p(R_j) = \frac{e^{-k \cdot \log U_j}}{\sum_i e^{-k \cdot \log U_i}}$$ **Kirchhoff Logit function** **5.5**

Where

p(Rj) = probability of route j to be chosen.
C_j = General cost of route i
U_j = Utility of route j.
μ,k = sensitivity of the model.

	A		B	
Cj	5	15	105	115
Uj	0,2	0,067	0,0095	0,0087
μ	2,5	2,5	2,5	2,5
	1,65	1,18	1,02	1,02
	Sum of A		**Sum of B**	
p(Rj)	58,3	41,7	50,1	49,9
dif	16,5		0,1	

	A		B	
Cj	5	15	105	115
Uj	0,2	0,0667	0,0095	0,0087
k	-2,5	-2,5	-2,5	-2,5
	0,1742	0,0529	0,0064	0,0058
	Sum of A		**Sum of B**	
p(Rj)	76,7	23,3	52,5	47,5
dif	53,4		4,9	

Table 5.1; values of Logit (left) and Kirchhoff's logit functions (right).

The factorial values given for 5 to15 min. and 105 to 115 min. can be found at the bottom of the upper tables. At the bottom row in the lower tables, the probability of choosing between 5 min. or 15 min. compared to 105 min. or 115 min. is illustrated as using the Kirchhoff function presents a more realistic weight.

5.1.3 Vehicle specifications

Specific for VISSIM is the possibility of adjusting settings for vehicle specifications, speed distributions and driving behavior. With each of these subjects, and other options, it is possible to realistically simulate traffic.

For accuracy it was decided that the models should contain 4 vehicle-classes (car, van, truck and bus) each with special specifications.

The busses are directed by a separate schedule, while the rest are operated by OD-matrices.

	Car		**Van**		**Truck**		**Bus**		
	max 0 km/h	min 250 km/h	max 0 km/h	min 250 km/h	max 0 km/h	min 250 km/h	max 0 km/h	min 170 km/h	
Acc. Max	3,5	0	3,5	0	2,5	0	2,7	0	m/s2
Acc. Desired	3,3	0	3	0	2,2	0	2,5	0	m/s2
							0 km/h	210 km/h	
Dec. Max	-7,5	-5,1	-7,5	-5,1	-7,5	-5,1	-6	-0,1	m/s2
Dec. Desired	-2,8	-2,8	-2,8	-2,8	-1,3	-1,3	-0,9	-0,9	m/s2

Table 5.2: Speeding decisions for vehicles in VISSIM

5.1.3.1 Speed decisions

In VISSIM it is possible to ad levels of free speed, representing the natural distribution between individual drivers. Also differences in driving pattern influence by regulations can be visualized.

For the heavy vehicles, this distribution is quite simple, since the tachographs installed in all busses and trucks ensures that no heavy vehicles exceed the speed limit set for this group of vehicles. Figure 5.3.(right)

Even in regulated areas the speed distribution for cars and vans is more complex, since regulations don't ensure uniform driving patterns. These variations are represented in the distributions seen in Figure 5.3.(left)

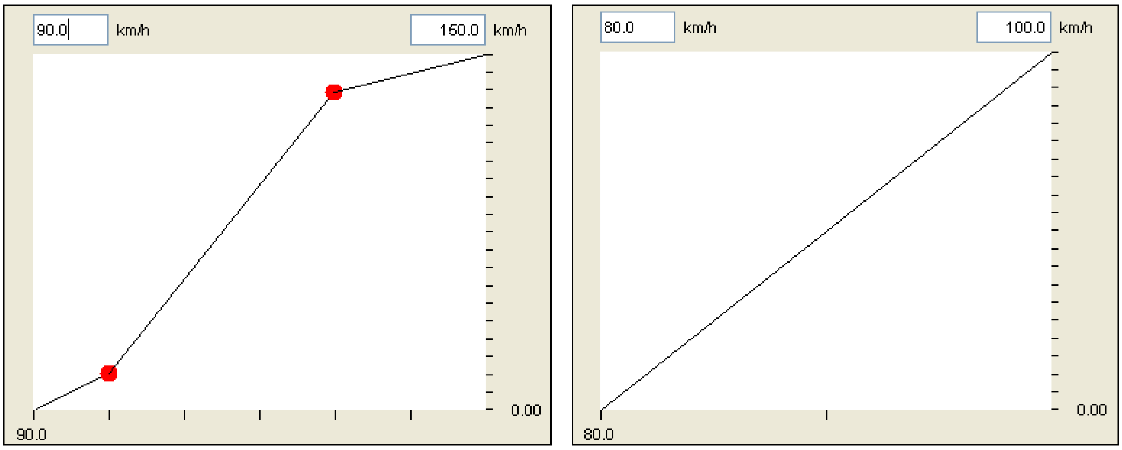


Figure 5.3: Speed distributions motorway, for (left) car/van, (right) truck/bus

The distribution on the left is assumed for the remaining speed distributions, to be seen in Table 5.3. So far no material has been found to support these distributions

Distribution ID	min.	10%	90%	max.	
309 (car + van)	90	100	130	150	km/h
209 (truck + bus)	80			100	km/h
80	75	80	90	100	km/h
70	65	70	80	90	km/h
60	55	60	70	80	km/h
50	45	50	60	70	km/h

Table 5.3: Speed distributions.

5.1.3.2 Driving behavior

Two different driving behavior models are used in the network. Motorized "inter urban" (right side rule) is used at the motorways, while urban (free lane selection) is used elsewhere including ramps.

Driving behavior	Urban		Inter urban	
Following				
Traffic follow model	Wiedemann 74		Wiedemann 99	
>Standstill distrance	2,00 m		1,50	
>Desired safety add.	2,00			
>Desired safety mult.	3,00			
>Headway			0,9 sec	
>Follow var.			8,0 m	
>Threshold --> 'follow'			-8,00	
>'Follow' lower			-3,50	
>'Follow' upper			3,50	
>Speed dep. Oscillation			11,44 m	
>Acceleration oscillation			0,25	m/s^2
>Acceleration 'standstill'			3,50	m/s^2
>Acceleration '80 km/h'			1,50	m/s^2
Observed objects	5		5	
Look ahead	0-300 m		0-300 m	
Lane change				
Lane behavior	Free lane selection		Right side rule	
	Own	Trailing	Own	Trailing
max. Deceleration	-4,00	-3,00	-4,00	-3,00
-1,0 m/s2 per distance	100 m	100 m	200 m	200 m
Accepted deceleration	-1,00	-1,00	-1,00	-0,50
Diffusion time	60 sec		60 sec	
min headway	0,5 m		0,5 m	
min hw. 'to slow lane change'	--		11 sec	
Lateral				
Desired pos. At free flow Overtaking not enabled	middle		middle	
Amber signal				
Amber check	Continuous		Continuous	

Table 5.4: Driving behavior parameters

A lot of different changes could have been made to differentiate the modeling of driving behavior, but in order to test for relevance and changes in the outcome, the only value changed from the default settings are *observed objects* from 3 to 5. In earlier cases the model has proven to run more smoothly, as more objects are being observed, though too many objects slow down the model considerably.

5.2 Dynamic Traffic Assignment in Dynameq

Prior to each vehicle assignment, link travel-times are calculated, and thus travel-times for each path. Based on the path's travel-time, a percentage of drivers is assigned to each path. The iterations are divided into assignment intervals, which represent departure time, in which the percentage of drivers is constant on paths. The paths and percentages change between assignment intervals. In reality this represents a preferred set of paths if a driver leaves home at 07:15, and another set if he leaves at 07:55. Due to alterations in the traffic situation, like a build-up of queues, alternative routes become more attractive.

In Figure 5.4 the process of DTA in Dynameq is represented graphically.

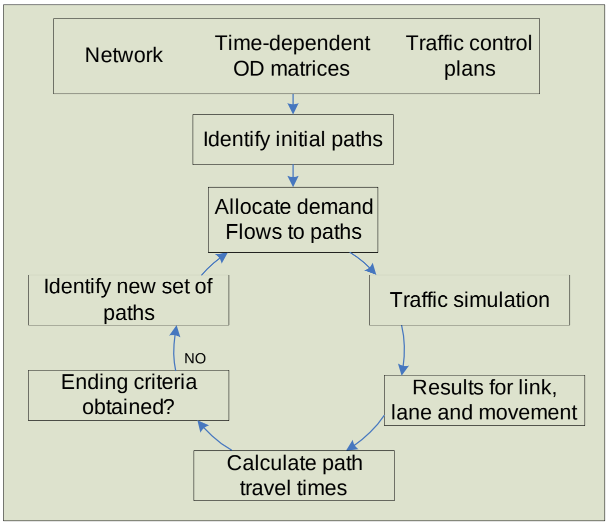


Figure 5.4: DTA model in Dynameq

5.2.1 Path choice

The driver's path choices for a specific assignment interval are defined by the path travel-time in the previous iteration connecting an O-D pair.

n = 1. (all or nothing).

The model assumes that traffic flows at free speed on all links. The shortest path is identified and all the traffic is assigned.

n ≤ 10 (path generation).

In the DTA, the number of path's to be identified (Y) is given by the user. The traffic is equally assigned on all paths. Thus in the 5. iteration 1/5 of the O-D traffic is assigned to each of the 5 shortest paths.

n > 10 (convergence stage).

By each following iteration (n) the number of vehicles is adjusted to balance out travel-times. This is done by adding a fraction of the number of vehicles to the shortest path for a specific assignment interval (a), and removing this amount in a proportional way from all the other paths.[19]

By each following iteration (n), the number of vehicles is adjusted to balance out travel-times. This is done by adding a fraction $\left(\frac{1-n}{n}\right)$ of the number of vehicles to the shortest path for a specific O-D pair (i) and assignment interval (a), and removing this amount in a proportional way from all the other paths.[19]

5.2.2 Assignment

Each simulation period is divided into assignment intervals (AS). Each assignment interval defines a departure time, in which the models values are constant. In this way, alterations happen between intervals.

$$h_k^{a,n} = \begin{cases} h_k^{a,n-1}\left(\frac{1-n}{n}\right) + \frac{g_i^a}{n} & if \;\; s_k^{a,n} = u_i^{a,n} \\ h_k^{a,n-1}\left(\frac{1-n}{n}\right) & otherwise \end{cases}$$ **Traffic allocation** **5.6**

Where:

$h_k^{a,n}$ = Demand flow for path k, assignment interval a, iteration

$s_k^{a,n}$ = Travel-time for path k, assignment interval a, iteration n

$u_i^{a,n}$ = Shortest travel-time for O-D pair i, assignment interval a iteration n

g_i^a = Demand flow for O-D pair i, assignment interval a

As the equality $s_k^{a,n} = u_i^{a,n}$ identifies the shortest path of a specific O-D pair, equation 5.6 reduces flow on all paths with $1-n/n$ and adds the same fraction g_i^a/n of flow to the shortest path. If all paths within one assignment interval have the same travel-time, a state of Dynamic User Equilibrium has been obtained.

5.2.3 Traffic generation

In general the traffic generator can vary on two factors: the inter-departure time and the generated number of vehicles. In Dynameq three levels can be chosen:

	Inter-departure time	Generated veh.
Poisson	Random	Random
Conditional	Random	Constant
Constant	Constant	Constant

Table 5.5: Levels of traffic distribution

The random inter-departure time (time between vehicles) follows a negative potential distribution, while the randomly generated vehicles follow a Poisson distribution. The constant inter departure time is given by the matrix duration divided by generated vehicles, which is in turn given by multiplying flow-rate with the matrix duration.

For constant generated vehicles the total amount of veh. can vary as a result of the rounding process used to produce integer values for the model. Using the random model the number of veh. may sometimes be higher because the variance of the Poisson distribution is equal to the mean.

5.2.4 Path travel-time and pruning

Travel-times for a specific path are made up from the travel-times on its links, in any given calculation interval. After each traffic simulation, travel-times are calculated for each link. The average travel-time for vehicles leaving the link through a specific movement, make out the travel-time.

During a traffic simulation, the travel-times are constant within each calculation interval of example: 5 min. The assigned traffic moving through the model is a part of several calculation intervals, and thus influences the traffic situation and travel-time – perhaps causing congestion, buildups or even spill-backs to other links.

Pruning can be used for reducing the number of paths. At the time of introduction, it assigns zero flow to paths that were the shortest path only when they were first discovered. If the path at a later time is recognized as the shortest path, it is included once again.

5.2.5 Gap and convergence

While the flow on each, generated path is adjusted with the iteration, the DTA is said to be converging towards equilibrium. In reality, as each driver optimizes his or her path from origin to destination, the average travel-time for all users comes closer to the fastest travel-time registered for each assignment interval.

In equation 5.7 the relative gap is given by the difference between the average and fastest travel-time divided by the fastest time.

$$RGap^{a,n} = \frac{\sum_{i \in I} \sum_{k \in K_i^a} h_k^{a,n} \cdot s_k^{a,n} - \sum_{i \in I} g_i^a \cdot u_i^{a,n}}{\sum_{i \in I} g_i^a \cdot u_i^{a,n}} \qquad \textbf{Relative Gap} \quad \textbf{5.7}$$

Where

I = is the set of all OD-pairs

K_i^a = the set of paths for OD-pair i and assignment interval a

5.2.6 Traffic flow

To describe how vehicles of a specific class move on the roadway, equation 5.8 is used. Specifically it describes the position of the vehicles on the roadway, at time *t*.

$$x_f(t) = MIN\left[\left(x_f(t-R) + V_{free} \bullet R\right), \left(x_l(t-R) - L\right)\right] \qquad \textbf{Car-follow model} \quad \textbf{5.8}$$

Where

$x_f(t)$	= Position of following car at time t
$x_l(t)$	= Position of the leading car at time t
V_{free}	= Free speed on the roadway
L	= Effective length of following car
R	= Response time of following car

The 'car follow' model used in Dynameq, describes how vehicles move on the roadway according to 'steady state' properties. 'Steady state' properties can be observed when traffic is moving at a constant speed, which also results in constant flow and density. The relationship is given by *Flow* = *Speed x Density*, and can be expressed in three ways.

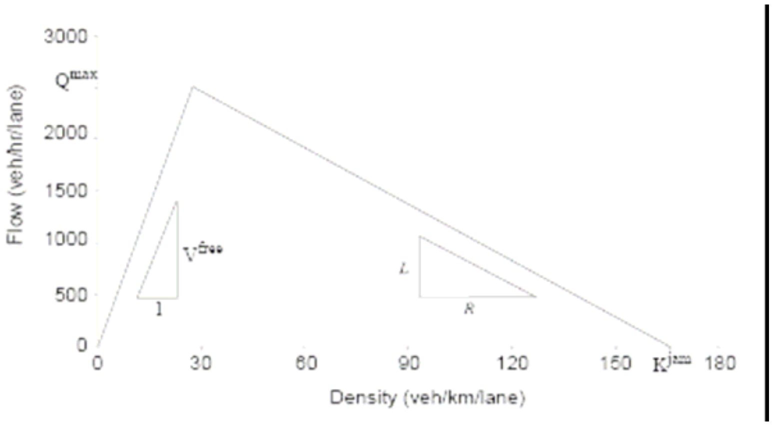


Figure 5.5: Flow / Density diagram

In the flow/density diagram see Figure 5.5 the level of congestion is given by density of veh/hr/lane, increasing to the level of jam ($K_{jam)}$, when speed is zero. On the vertical axis the flow increases to a max flow (Q_{max}), which is also the breaking point between non-congested and congested conditions on the graph.

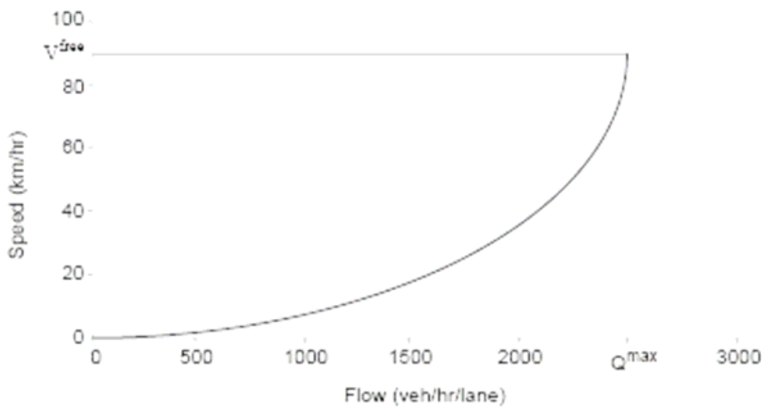


Figure 5.6: Flow / Speed diagram

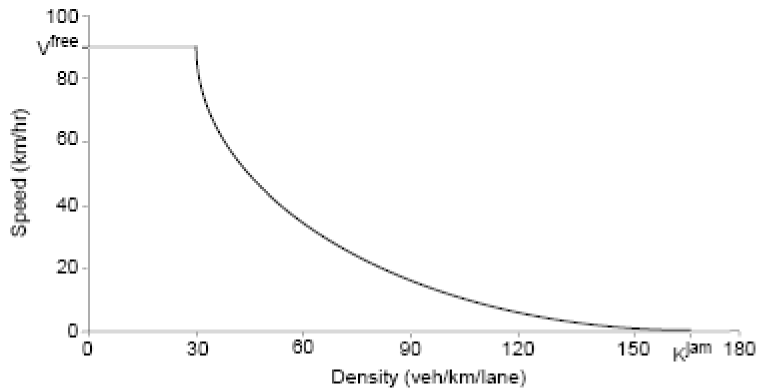


Figure 5.7: Speed / Density diagram

The diagrams in Figure 5.5, Figure 5.6, Figure 5.7 both show speed as a function of either flow or density. In both diagrams the free speed is maintained for non-congested conditions, while the speed is reduced non-linear from that point on – till the jam-density is reached and the speed is zero.

The extreme values of these diagrams, is given by:

$$Q_{\max}\left(Max\,Flow\right) = \frac{1}{\left(\frac{R}{3600}\right) + \frac{\left(\frac{L}{1000}\right)}{V_{free}}} \qquad \textbf{5.9}$$

Q_{max} is given for a specific vehicle class and the free speed of the roadway.

$$K_{jam}\left(Jam\,Density\right) = \frac{1000}{L} \qquad \textbf{5.10}$$

K_{jam} is the number of vehicles from one class that fit one lane, one km, when standing still.

$$V_{wave}\left(Wave\,Speed\right) = \frac{K_{jam}}{\left(\frac{3600}{R}\right)} \qquad \mathbf{5.11}$$

V_{wave} is the wave speed given by the time it takes from the first to the last vehicle to move divided by the distance between them. From this density, the condition of the lane is perceived as queuing.

5.2.7 Merge and intersections

Three turning movement templates have been used in this model. Signalized, wherever an intersection have been signalized. Merge was mainly used where ramps intersect with other ramps or motorways. Two-Way Stop Cross (TWSC) handles traffic in all non-signalized intersections.

Dynameq uses Available gap and Relative wait in combination with Critical gap and Critical wait, to determine the probability of precedence. Demand time (Dt) holds the arrival time of the vehicle at the movement, while Supply time (St) is the earliest time the vehicle can execute its movement. St can be updated if another vehicle should be chosen to execute its movement first.

$$\begin{aligned} Gap(t) &= \left(St_h - R_h\right) - \left(St_l - R_l\right) \\ Wait(t) &= \left(Dt_h - R_h\right) - \left(Dt_l - R_l\right) \end{aligned} \qquad \text{Available gap \& Relative wait} \qquad 5.12$$

Where:
l = low priority
h = high priority

Each of the respective times are then compared to the respective critical values found in Table 5.6. When the wait is compared to the critical wait time, the precedence is determined, and thus the probability of completing the movement. See Figure 5.8

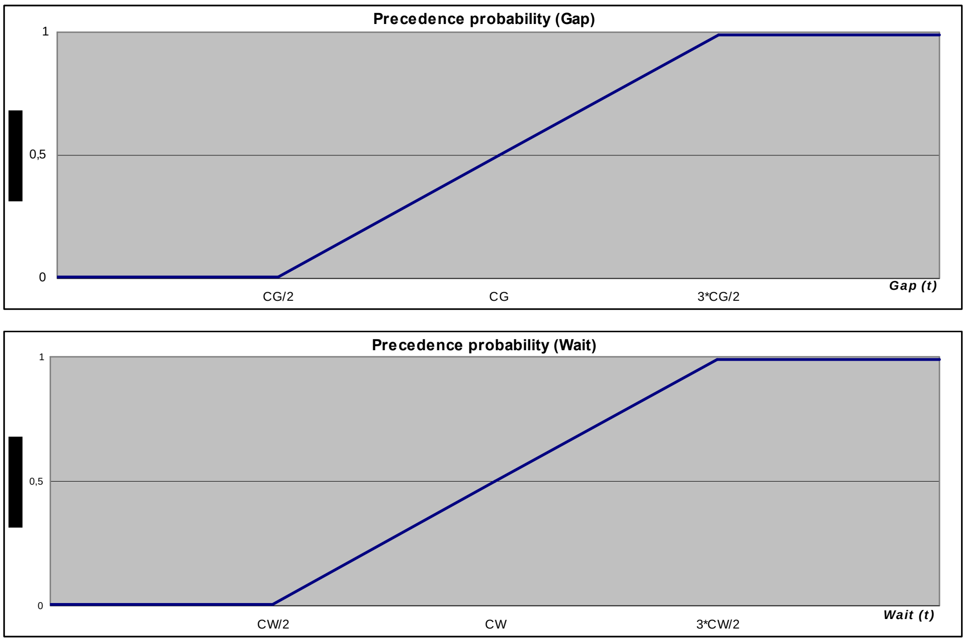


Figure 5.8: Precedence probability for critical gap and -wait[19]

Two Way Stop Cross

	Critical gap	Follow up time
Left turn from major	4,10	2,20
Through from minor	6,50	4,00
Right turn from minor	6,20	3,30
Left turn from minor	7,10	3,50
Critical Wait	30,00	

Signalized intersection

	Critical gap	Follow up time	Capacity
U-turn (not used)	6,50	4,00	900
Left turn		2,50	1440
Right turn		2,50	1440
Thorugh		1,80	2000
TurnOnRed (not allowed)	6,20	4,00	900
Left turn conflicting through	4,50		
Left turn conflicting right	4,50		
Critical Wait	30,00		

Signalized intersection

Critical gap	2,00
Follow-up time	0,00
Critical wait	2,00

Table 5.6: Times for turning movements

5.3 Dynamic Traffic Assignment in Time Slice Assistant [13]

…developed for the Traffic Analyst toolbox, integrated in the ArcGIS environment

The TSA differs from the other softwares, by offering the possibility of link-independent calculation-intervals for updating the traffic loads. The paths are identified by a shortest-path algorithm. Following this, the traffic is assigned and travel-times, based on traffic loads, are updated. Each link is partitioned into independent calculation intervals. During each interval a temporary traffic load variable is updated before new traffic enters the link.

Increasing the number of time-slices for a number of chosen links will increase the dynamic aspect of modeling for certain areas of interest, at least potentially, without increasing the calculation-load for the total network simulation.

In large networks, this feature might allow for a more precise simulation of heavily congested traffic on a limited number of links. This concept might be a cost/efficient approach for a local 'dynamic zoom' into existing large networks, especially where resources have already been spent on modeling, in the Traffic Analyst toolbox. Question is, to which degree the interactive effect from neighboring zones can be neglected or reduced without compromising valid modeling.

One approach to reduce this uncertainty would be the inclusion of a number of neighboring links, chosen from past experiences with congestion problems. In complex networks a pre-run on the network would make it easy to identify neighboring links', infliction on the chosen "area of interest". Now, knowing the level of max congestion, one could accordingly increase the number of calculation intervals.

5.3.1 The Algorithm

The algorithm consists of two loops. An outer loop taking care of the MSA update and an inner loop performing incremental launch of traffic into the model.[13] In the search for the shortest path in the graph, the shortest-path-algorithm seeks out information of the updated and recalculated travel-times in the relevant time interval. Following the shortest path, the traffic launch is made, and all volumes and travel-times are updated. The first in first out (FIFO) makes sure the order of entering and leaving traffic is maintained, while the capacity check ensures that congestion causes queues and spill backs.

The Algorithm consists of:

A. Initiation
   ~ All variables are preset to fixed values.
B. Assignment update
   ~ Coefficients are generated.
   ~ Shortest path tree is calculated. ”incremental”.
   ~ Traffic volumes are updated. “incremental” (eq. 3.2),(eq. 5.14)
   ~ Travel-times are updated (eq. 5.15)
   ~ First in first out (FIFO), Capacity, Delay
C. Final update (links with no traffic)
   ~ Traffic volumes are updated (eq. 5.16)
   ~ Travel-times are updated (eq. 5.15)
   ~ First in first out (FIFO), Capacity, Delay

$$T_{akx(temp)} = T_{akx(temp)} + T_{ijkx} \tag{5.13}$$

$$T_{akx(n)} \coloneqq (1-\alpha)\cdot T_{akx(temp)} + \alpha \cdot T_{akx(temp)} \tag{5.14}$$

$$t_{akx(n)} = f\left(t_{(0)ax}, T_{ax(n)}\right) \qquad \forall k \tag{5.15}$$

$$T_{akx(n)} \coloneqq (1-\alpha)\cdot T_{akx(temp)} \tag{5.16}$$

Where:

T	= Traffic-load
t	= Travel-time
a	= Link
k	= Vehicle type
x	= Time-interval
α	= 1/n

Equation 3.2: Instead of a global update of all edges at one time, the TSA updates each edge with a temporary load as the incremental traffic is launched. This way the launch time and the independent link update functions without conflicting.

Equation 5.14: Is the MSA-updated traffic load. The temporary traffic load will be updated for each route entering the link. After the last route is added, all edges that are used in a route have been updated at least once, leaving those without traffic. This reduces the complexity of the algorithm.

Equation 5.15: Is the calculated travel-time on each link *a*, for vehicle type *k* and in time interval *x*. This value is a function of the congestion on the link and initial BPR.

Equation 5.16: After the temporary updates of traffic loads, some links might not have any traffic assigned. Still traffic loads and travel-times have to be updated accordingly.

5.3.2 Traffic on links

The OD-traffic enters the first link in the middle the OD's launch interval. As the traffic is influenced by congestion and buildups, and the time of leaving the link equals the entering time of the following link, unless buildups prevent it.

The traffic situation on a specific link can be divided into 4 different categories.

Free flow:

The situation in calculation interval (x) is uncongested and flow < Q_{max}. and the traffic condition is situated on the upper part of the Speed/flow curve.

Congestion:

Increasing traffic load causes the travel-time to increase. This is a result of the increasing density, and thus decreasing speed. As the flow pass the point of Q_{max} on the speed /flow curve, a state of queue-conditions sets in.

As congesting influences the flow/density curve largely id given by the K_{jam} value, the calculation of the queue speed also depends upon K_{jam}. The value of queue-flow would under normal circumstances be estimated from measurements. Here the parameter is estimated to $\frac{2}{3}Q_{\max}$ giving equation 5.17 for the queue-speed.[13]

$$v = \frac{Q}{D} = \frac{2 \cdot Q_{\max}}{2 \cdot \frac{Q_{\max}}{v_{free}} \cdot (1+\alpha) \ + \ K_{jam}} \qquad \mathbf{5.17}$$

Queue:

If queuing conditions is present traffic is allowed onto the link in a flow rate defined by the density.13See equation 5.18.

$$Q_{(in)_{\max}} = D - Q_{(out)} \qquad \mathbf{5.18}$$

Spill back:

Spill back emerges on the previous links of links with queue, as long as the traffic waiting to enter, exceed $Q_{(in)}$. This queue is considered a part of the prior link, and has to wait at least until the next time-period to enter the link.[13]

To control these different situations, the algorithm makes use of capacity checks and a routine to ensure the traffic entering the link leaves the link in the same order “first in first out” (FIFO). During the update both capacity and delay are tested and updated.

$$\begin{array}{ll} & t_{(x),e,out} \\ t_{(x),e,in} \nearrow & t_{(x),s,out} \\ t_{(x),s,in} \nearrow & \end{array} \tag{5.19}$$

Where:

x	time interval
s,e	start, end
in,out	incoming and outgoing traffic

5.3.3 Capacity check and queue building

Following the updates of traffic loads and travel-times, the capacity of the links has to be tested. In order to check whether the capacity limit of the link is reached, a test is performed. See equation 5.20. In general the test performed compares the length of the link with the line of cars on the link including the gap between them. If the line of cars is longer than the link, the link is closed for usage until the queue has been diminished.

$$\frac{\sum_k l_k \cdot T_{akx} \cdot min\left\{1, \frac{t_{akx}}{\left(t_{akx2in} - t_{akx1in} + t_{akx2out} - t_{akx1out}\right)/2}\right\}}{\#lanes} \leq l_a \tag{5.20}$$

Queue speed q is given as a fraction of free flow *time*.

5.3.4 FIFO

First in first out is a test that insures that the traffic arriving first at a link leaves before the next coming traffic. Normal this is not a problem, but if the traffic streams are given different travel-times, so later arrivals are getting a shorter travel-time, the exit-time can be corrupted.

Two cases have to be tested:

$$t_{(x),s,out} < t_{(x\text{-}1),e,out} \qquad \Rightarrow \quad t_{(x),s,out} = t_{(x\text{-}1),e,out} \tag{5.21}$$

$$t_{(x),e,out} < t_{(x),s,out} \qquad \Rightarrow \quad t_{(x),e,out} = t_{(x),s,out} \tag{5.22}$$

$$t_{(x)} = \frac{\left(t_{(x),e,out} + t_{(x),s,out}\right)}{2} - \frac{\left(t_{(x),e,in} + t_{(x),s,in}\right)}{2} \tag{5.23}$$

This process continues in the following time intervals to make sure all the following traffic is updated.

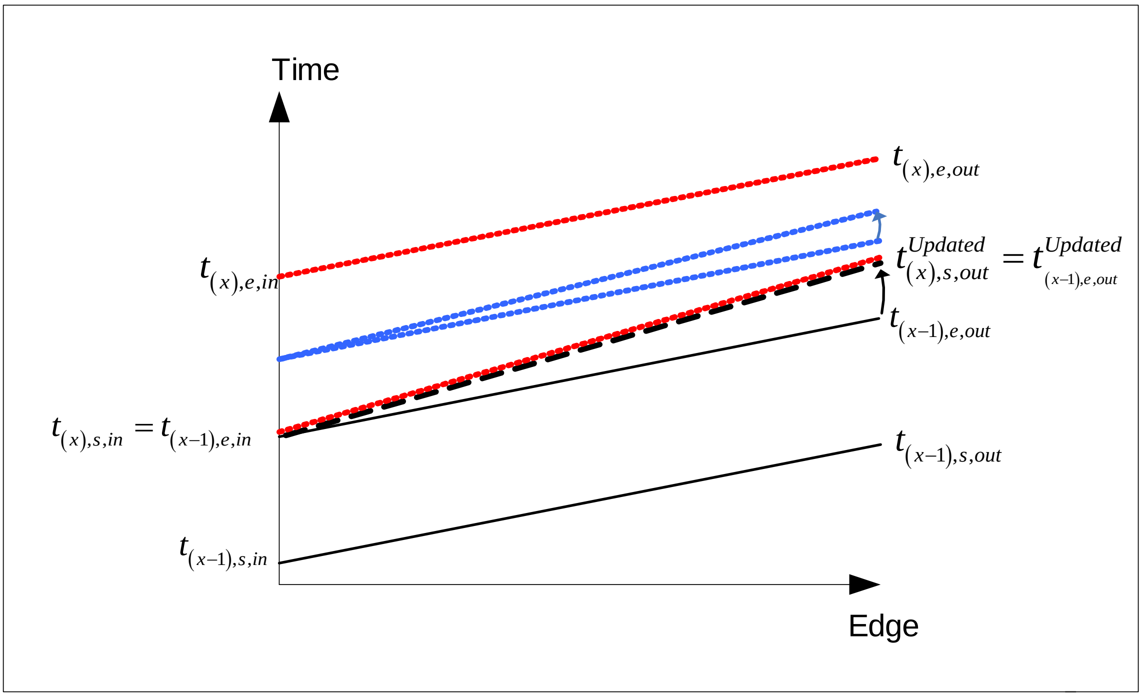


Figure 5.9: FIFO test of equation 5.22 shown.[13]

Imagine you enter a congested stretch of road. Prior to your arrival, a large amount of traffic has entered the stretch. Now you have to slow down, using longer time to pass. This situation, so well known to drivers, is processed this way:

As a high traffic load in (*x-1*), turns this interval to “congested” the travel-time of the link increases (black arrow). This reduces the capacity of interval (*x*), and thus increases the travel-time (blue lines) of (*x*). If the congestion in (*x*-1) is so heavy that the travel-time increases beyond the time interval of (*x*), (black arrow rises past "$t_{(x),e,out}$"), this time interval is now completely blocked and all incoming traffic is turned into a buildup.

5.3.5 Preferences

TSA being a GIS traffic model, preferences like capacity, speed and distance is connected to the model at link level. No specific turning movements are connected to the nodes, leaving the mode less exact compared to both VISSIM and Dynameq.

In the launch window preferences for the simulations are given. See table Table 5.7.

	T2 (05:00-07:00)	T3,T4 (07:00-09:00)
Launch partitions	12	6
Extra time pr. iteration	0 min.	0 min.
Time-incremental load iterations	5	5
K_{jam}	134 veh/km	151 veh/km
Capacity reduction factor	0,66	0,66

Table 5.7: Times for turning movements

Seeds in TSA are handled by the software choosing a new random seed for each new simulation. As oppose to the other two softwares, TSA adds an inner loop consisting of incremental load iterations, adding (see table) 1/5 of the traffic load in each iteration. TSA handles queue conditions by assuming that the flow of the lower BPR curve is set to $\frac{2}{3} \cdot Q_{\max}$ when the density $D < Dm$, calculated from the K_{jam}.

6 Results

Usually the process of modeling an existing problem in traffic will serve the two purposes: 1) calibrating the model to produce consistent simulation results, compared to observed data and 2) identifying typical patterns of interrelated problems.

In the next process these interrelations will be the analytical basis for alternative modifications of infrastructure, and now a series of simulations will provides the documentation for expected improvements from a calculated investment.

For clarity's sake, complex reports of numeric data will have to be summarized and visualized for the process of analysis, discussion and decisions.

The process of generating visualized results can be divided into 4 steps, depending on the software:

- Choosing data-collection-points on strategic locations within the model, in order to cover specific problems.
- Extracting relevant tables, to have relevant information on the area of interest.
- Processing of extracted data in order to visualize/present relevant data.
- Making maps, graphs or tables, visualizing the problem areas and if modeled, the improved changes as a result of infrastructural improvements.

All three softwares have different approaches to presenting whatever output is chosen or automatically generated. Where the user of VISSIM needs to make a lot of settings, both Dynameq and TSA present the user with a finite set of tables.

As VISSIM offers the possibility of generating *.avi movies, all other result are generally presented in graphs or tables. TSA only generates 4 database tables covering 3 levels of aggregation distributed on two dimensions, feature and time (Path-Total, Link/Connector-Total, Link-

TimeSlices),.Dynameq presents a combination of visualization and tables. Within Dynameq is found a well-organized tool for presenting results, capable of visualizing several results at the same time, on different aggregation levels. At the same time these results can be exported to tables and visualized in ArcGIS, by exporting the model to *.Shp.

6.1 Iterations

All simulations were carried out with 100 iterations, except for TSA that was extended to 500 in order to observe convergence more closely.

Due to time restrictions only a limited number of iterations are visualized in the results, since data like travel-time for the intermediate iterations gradually converge. Since changes are more considerable during the first iterations, number “(1), 2, 5, 10, 20, 50, 100” are chosen. Due to the fact that 1. iteration resembles “all or nothing”, 2. iteration is generally used to generate curves of convergence. For a few outputs also iteration 3 and 4 were included to enhance the resolution of inconclusive data.

6.2 Convergence

Convergence expresses how the model reduces the total travel-time during a series of iterations. This is done by balancing the traffic-flow, using several paths between origin and destination. As one flow at a specific time influences the traffic of the next coming time-period, the iteration balances the flow in such a way that travel-time is minimized. This looped process minimizes not only travel-time in a specific time-slice, but the overall travel-time for the entire model.

The process of extracting convergence data has generally been a process of isolating travel-time from result tables and summarizing a number of paths or linkID in order to make a common level of representation. These data are organized by iterations, seeds and time, ending up in 3 dimensions. Unfortunately each of the softwares exports data a little different, resulting in three different query setups.

Dynameq and TSA both have more tables each representing a seed - only the rows and columns are switched. VISSIM have the number of iterations in separate tables, though only the values {2,5,10,20,50,100,(500)} have been chosen. These differences called for slightly different methods of deriving the final graphs as can be seen in Figure 6.1, Figure 6.2 and Figure 6.3.

Initially the softwares produce a range of tables from which a vehicle travel-time is calculated. This part of the process is carried out in Access, while the secondary calculations were handled easier in excel. Thick arrows represent a calculation of some sort, while thin arrows are collected values for graphs represented by ovals.

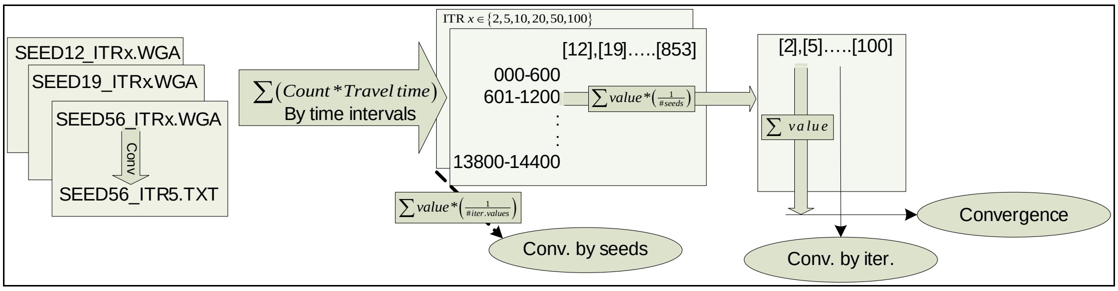


Figure 6.1: Flow process of making VISSIM convergence data.

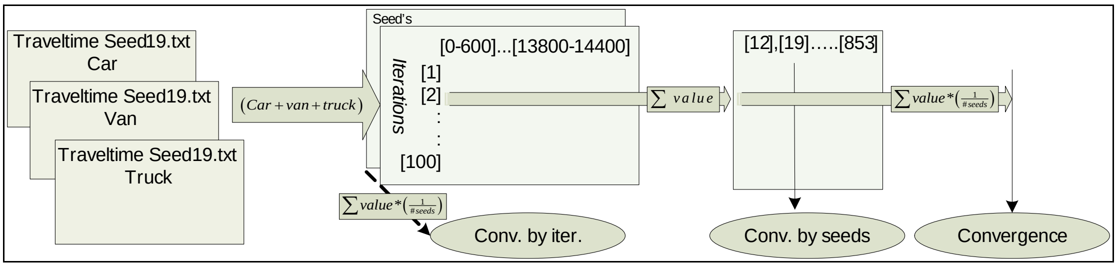


Figure 6.2: Flow process of making Dynameq convergence data.

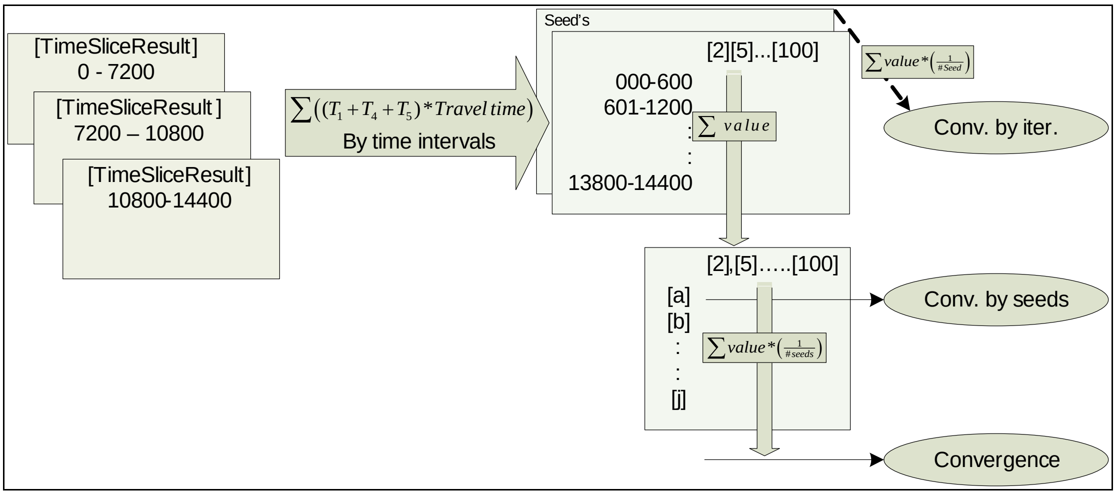


Figure 6.3: Flow process of making TSA convergence data.

6.2.1 Overall convergence

Software specific convergence looks at the softwares ability of converging toward a settled level of travel time. Usually the rate of convergence is measured as a *relative gap*. Instead of looking at the relative gap, the value of the travel-time is compared and examined.

During the 100 iterations VISSIM converges to a total travel-time of ~5500 hours. On Figure 6.4 the convergence of VISSIM is shown. VISSIM uses the first 10 iterations to reduce the travel-time with approximately 1000 hours, while the following 90 iterations are used for making relative minute changes, while the model converges.

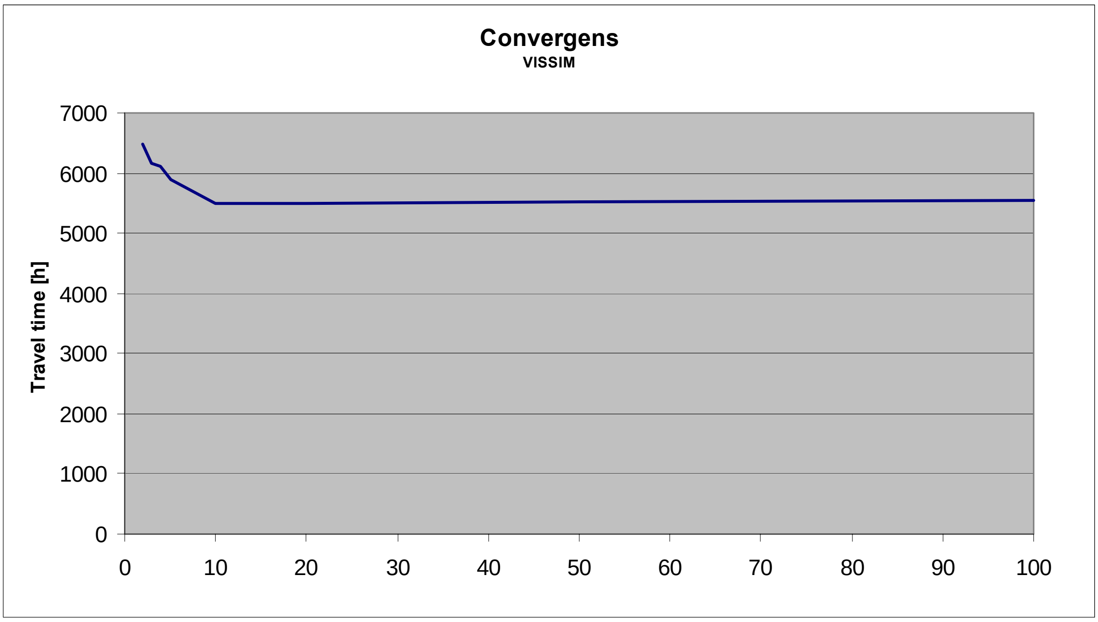


Figure 6.4: Convergence of VISSIM

In comparison with VISSIM, Dynameq uses 50 iteration to reach a reasonable level of converges at around 2800 hours of travel time. During the 50 iterations the model reduces with 2300 hours, from 5100 hours at the 2. iteration.

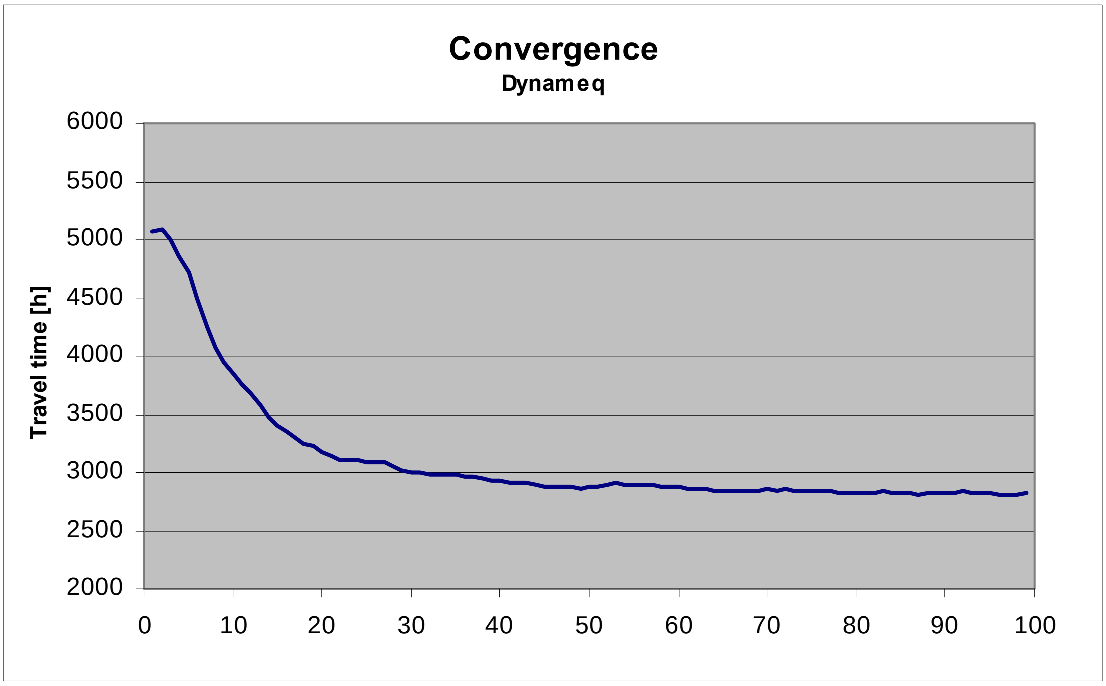


Figure 6.5: Convergence of Dynameq

Examining the convergence curve the relative low sum of travel time, raises the question if the model possibly could be calibrated with too low values of either:

- Capacity, allowing free speed for a larger interval of flow, thus minimizing travel-time
- K_{jam}, an unrealistic high value of K_{jam}-density would allow for a screwed lower BPR curve, altering queue conditions.
- #veh, if for some reason a lower amount of vehicles were to enter the model this could be the result.

As mentioned earlier, the TSA uses a model with inner loops, doing incremental loading, while an outer loop calculates MSA. This should help the model to converge relatively fast, as the inner loops times MSA iterations make up the total number of calculated iterations. In Figure 6.6 the curve, only given by MSA iterations, shows an inverted curve, which starts at a minimum converging upward towards 5000 hours.

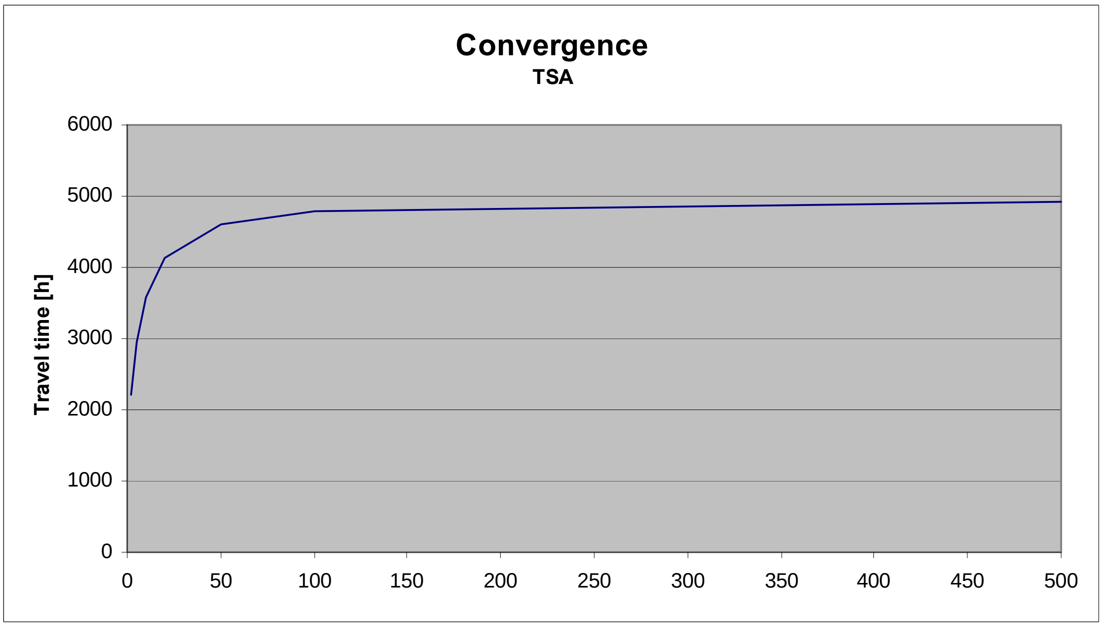


Figure 6.6: Convergence of TSA

6.2.2 Distribution of travel time

In the following section the aim is to visualize when the convergence actually takes place. According to the basic understanding of convergence, new paths are identified or the flow is rearranged to obtain a maximized overall flow. Thus if the very little traffic fills the model no considerable congestion influences travel time. So in order to actually cause congestion on links, the traffic load must be of a size where the traffic density causes shifts to the lower part of the BPR curve.

On the graph from VISSIM a clear division round 07:00 shows the beginning of the morning rush hour. Prior to this point only a slight increase in the 2. iteration can be traced. The following increase of flow causes travel-time to increase. In the later hour a small increase in the flow makes the model increase. As the traffic load increases in the later hour, a small increase can be observed.

In accordance with the convergence curve in Figure 6.7 it is clear to see how the convergence is almost complete after 10 iterations.

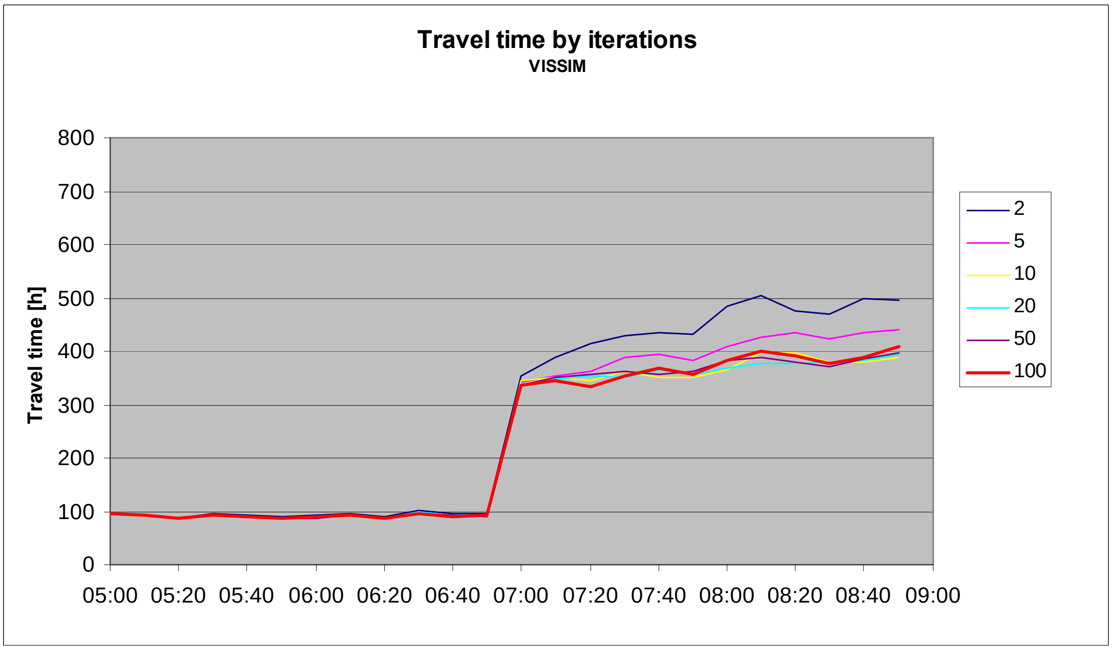


Figure 6.7: Convergence by iterations, VISSIM

When investigating the graph of Dynameq an obvious difference from the VISSIM graph appears. Where VISSIM in the early iterations keeps on inclining, Dynameq peaks and then decreases. Besides the fact that Dynameq has an overall lower travel time, the converged graphs look alike. This could indicate that early iterations hits the maximum level of capacity already in the first peak hour, but it does not explain the lack of congestion in the second peak hour where flow have increased.

In the process of reallocating traffic to alternative paths, Dynameq optimizes very strongly from iteration 1 to 50 the travel-time coming very close to the ideal process of converges. At the same time Dynameq converges by reallocating early traffic resulting in a delay of the peak load.

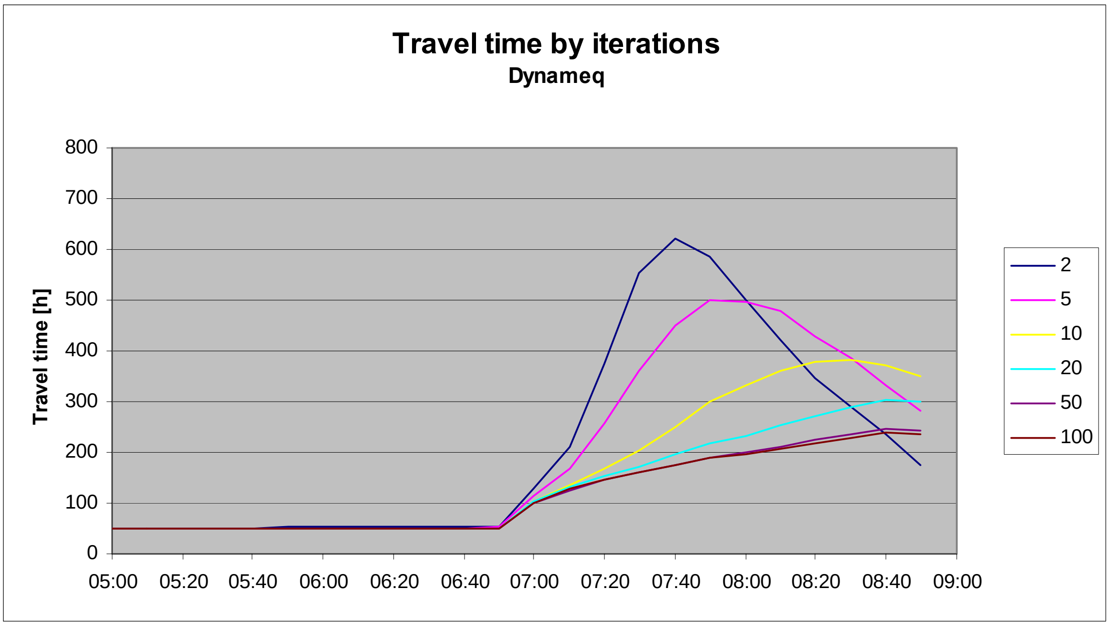


Figure 6.8: Convergence by iterations, Dynameq

Reacting very strongly to the first sign of congestion in the early iterations, Dynameq's process of optimization reduces this early congestion, by reallocating early traffic accepting increased congestion in later time slices.

The graph for TSA follows the convergence of TSA perfectly.

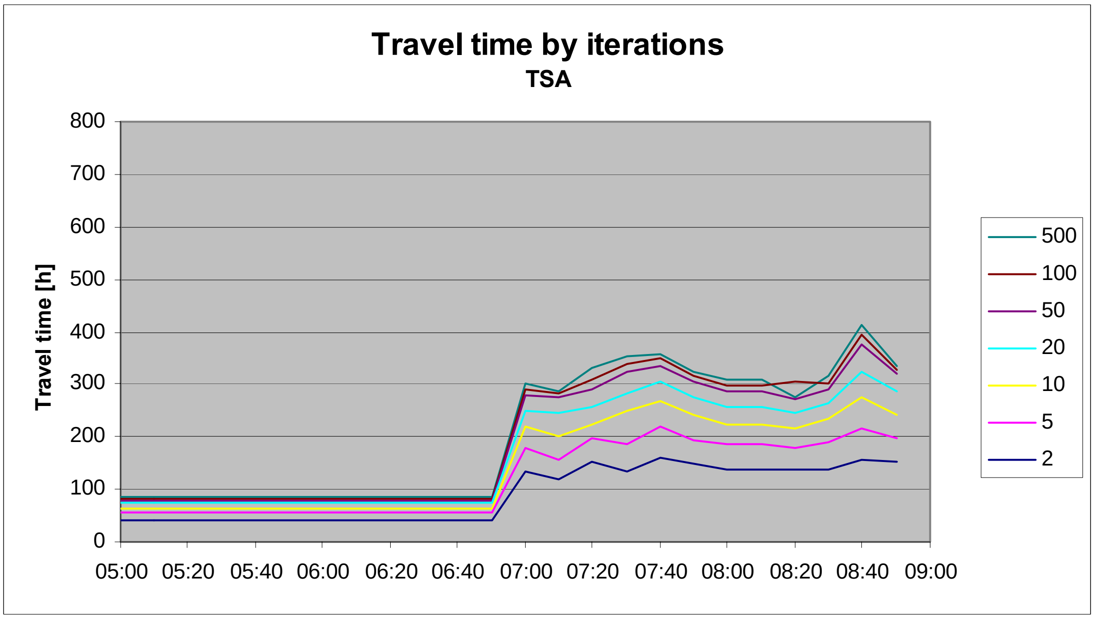


Figure 6.9: Convergence by iterations, TSA

The fact that travel-time increases, though it is under no or very little influence of congestion seems strange but can be explained by incremental iterations

The graph in figure Figure 6.10 is the product of early mistakes. Trying to locate this error, the first value to be altered was the time-incremental load iteration (TILI) set to 1 indicating that only one inner loop was carried out. This all-or-nothing iteration was thought to initiate queue conditions, which the model would not be able to end. In order to change this, the TILI value was increased to 5, this should diminish this effect, because of the incremental loading.

Later, by coincidence the value of *numbers of minutes worth of extra time slices* were decreased from default 30 min. to zero. This produced the graph above.

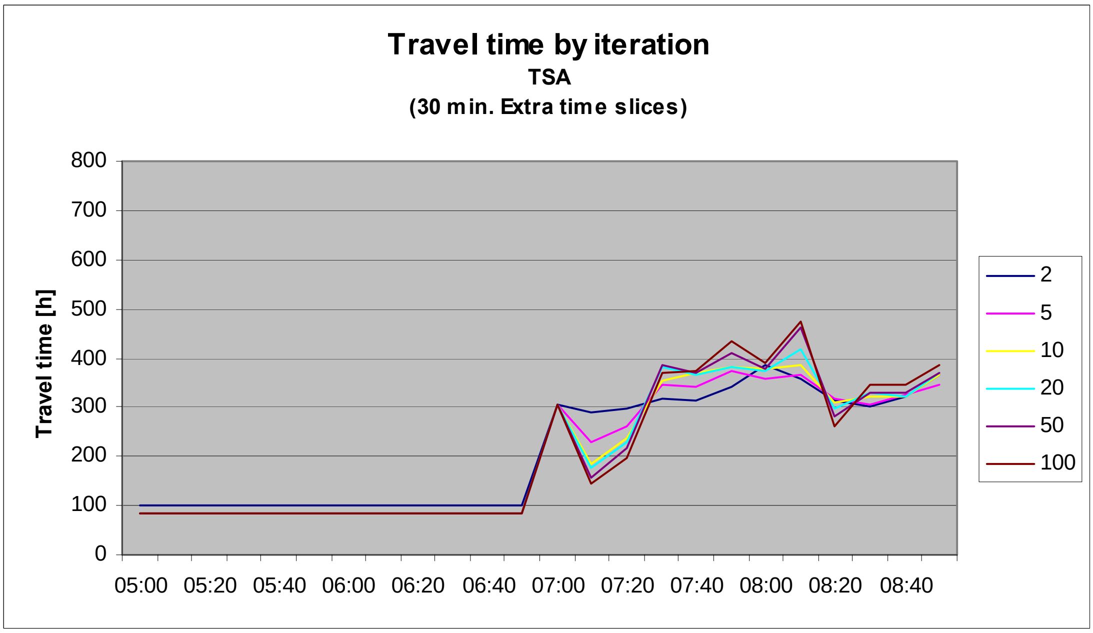


Figure 6.10: Convergence by iterations, TSA (30 min. extra timeslices)

To compare the combined effects of 30 min. worth of extra time slices and the difference between using 1 and 5 iterations, Figure 6.11 shows how removing 30 min. worth of extra time slices reduces the alterations heavily, while increasing from 1 to 5 inner loops smoothes the peaks in both cases.

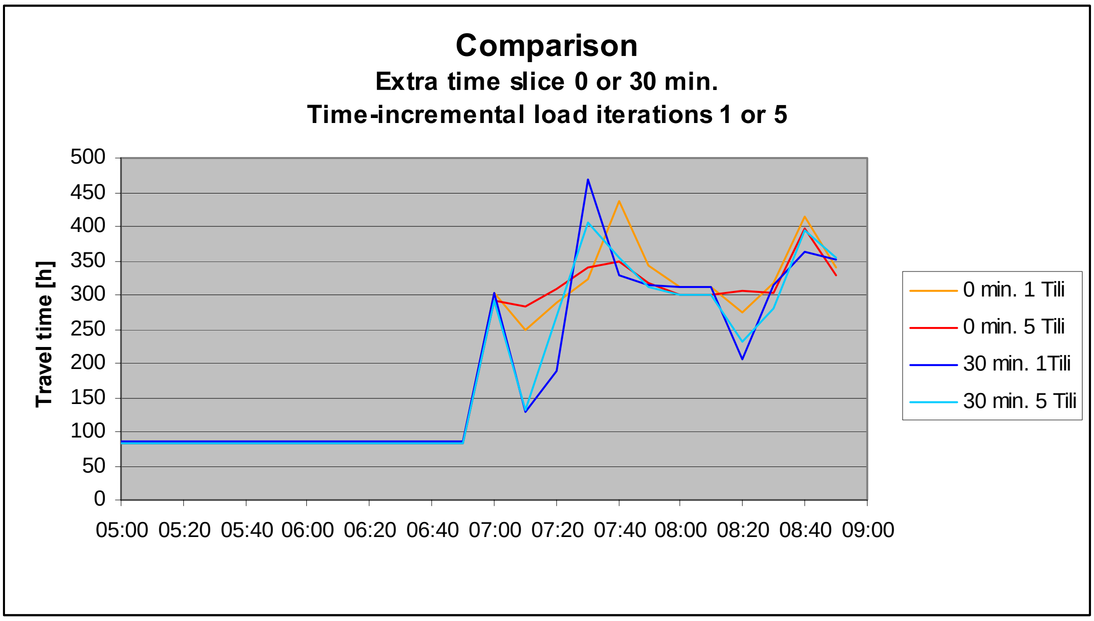


Figure 6.11: A comparison of different inner loop iterations and extra time worth of time slices at 100 MSA iterations.

The combination of 0 min. of extra time slices and 5 inner loop iterations was used to produce the results of the TSA.

A visual comparison of the 100. iteration, from each of the softwares shows a consistency between VISSIM and TSA. Only the overall value of Dynameq is clearly lower. It is particularly interesting to note that without congestion Dynameq is still considerably lower compared to the other two softwares. As mentioned earlier, this might be based on lower traffic loads, higher capacities or screwed k_{jam} value disrupting the queue conditions.

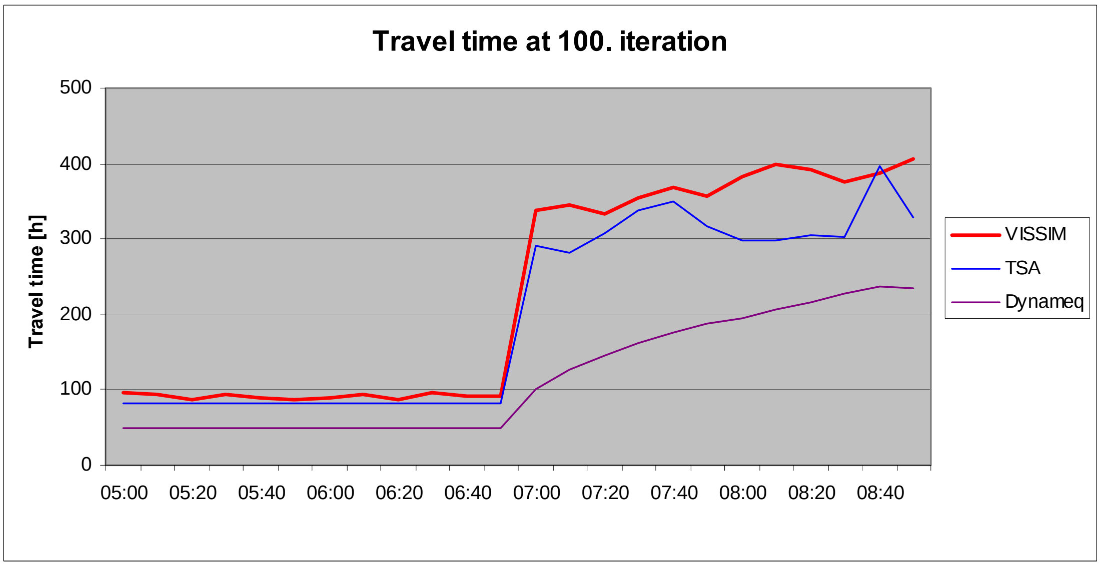


Figure 6.12: Travel-time at the 100. Iterations, all three softwares compared

6.3 Seeds

For each program a range of random seeds were chosen in order to represent the stochastic. By calculating an average value on these results a base is formed on which the assumptions and conclusions can be drawn.

The seed alters the traffic volume pr. launch in order to test the way in which the model handles different assignment patterns. This in turn alters the model-results, and introduces variance.

Observing the graph of VISSIM Figure 6.13 the variance of the results is relatively small. The seeds produce most variance in the first 20 iterations. Around the 50th iteration, the least variance is observed, while an increase occurs around the 100th iteration.

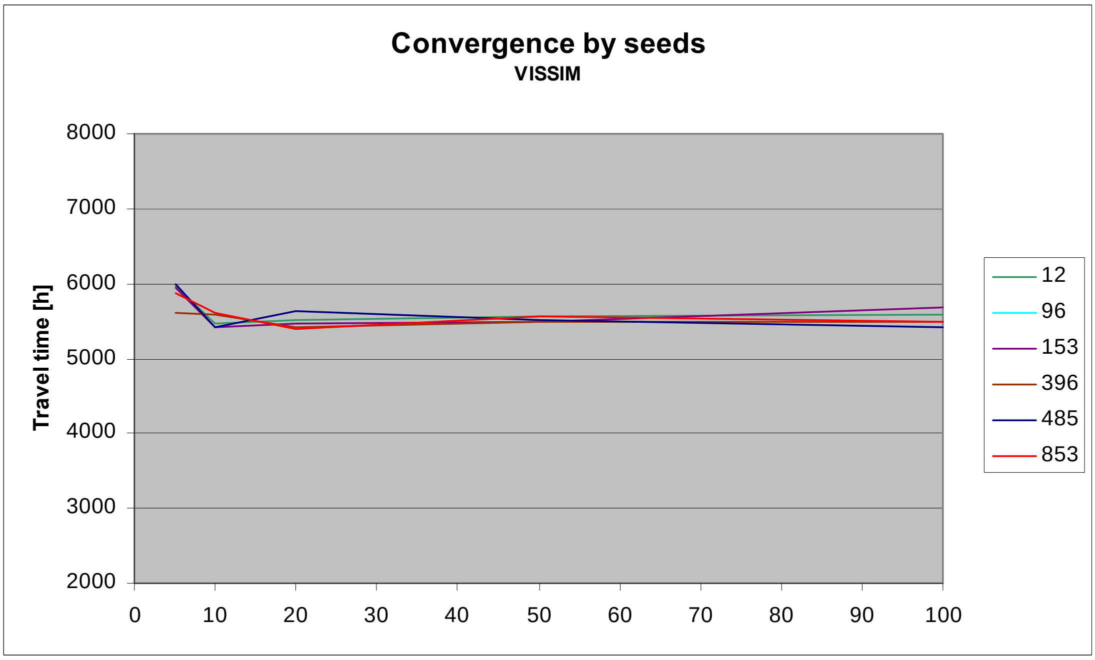


Figure 6.13: Convergence by seeds, VISSIM

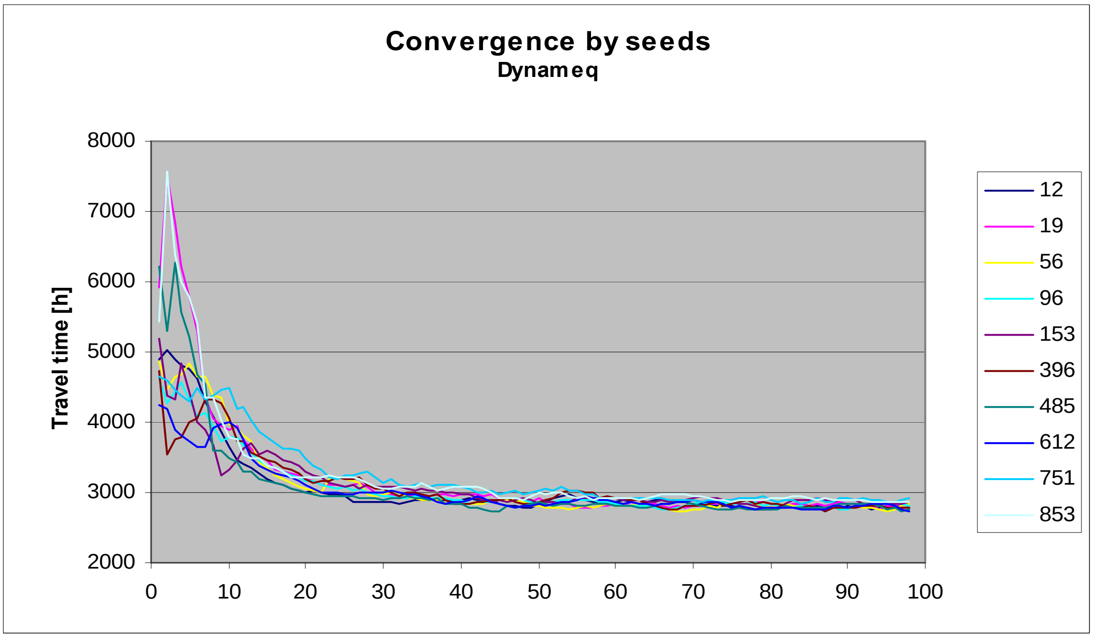


Figure 6.14: Convergence by seeds, Dynameq

As the Dynameq graph converges, large fluctuations can be observed in the first 20 iterations, while the decrease continues towards the 50th iteration. From this point on the variance stabilizes itself. This trend can also be observed in the *convergence by iterations* in Figure 6.14where the reduction also follows this pattern of iterations.

The convergence graph of TSA only consists of two simulations, due to a restricted timeframe. But already here a remarkably low value of variance is noticeable.

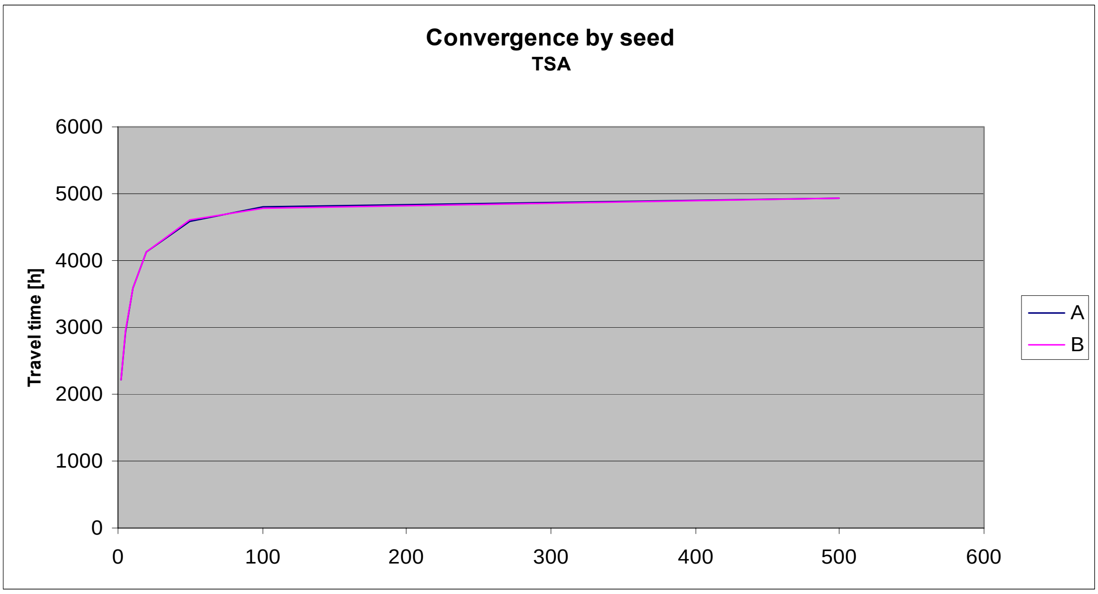


Figure 6.15: Convergence by seeds, TSA

Comparing the standard deviation of the three softwares, the values indicate that the seeds and thus the variation in traffic load of Dynameq, causes the software to fluctuate before the model obtains convergence. VISSIM causes some fluctuations, while TSA, as observed above, has next to no variance. Though introducing several simulations might increase this value.

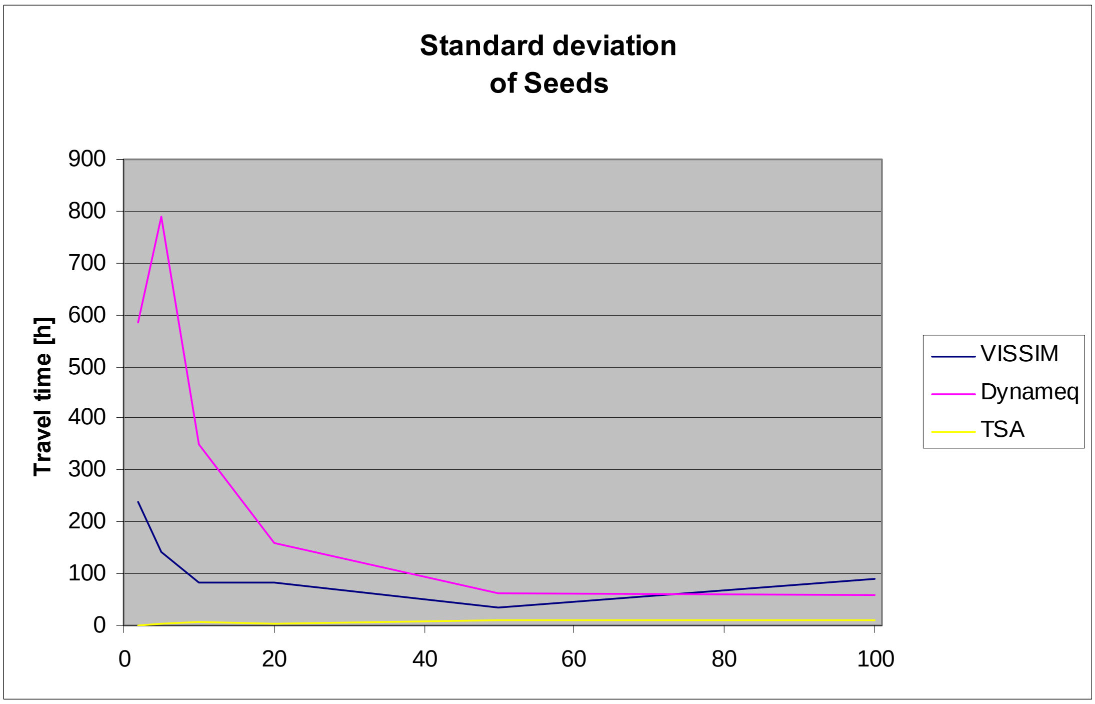


Figure 6.16: Standard deviation of convergence by seeds.

6.4 Time consumption of convergence

In the previous section, the softwares' ability to converge through a number of iterations was compared.

In the following the focus will be on the time it actually take one of the softwares to reach a fair level of convergence. Comparing the time consumption between the three programs, VISSIM and Dynameq was based on the time spend on one iteration and then multiplied with the number of iterations. This could be done because simulation was carried out as a continuous process from 1. to 100. iteration. The time consumption of TSA was actually measured directly, because the algorithm starts over each time a new result for a specific number of iterations is produced. TSA was timed with 5 inner loops, and zero min. of extra time slices.

Depending of the machine, VISSIM uses between 20 and 30 minutes pr. iteration. Processing 4 hours of traffic, and roughly 72000 vehicles. In this comparison 20 min. was used because this was the time spend on the same machine as simulations with Dynameq and TSA. The time for Dynameq was set to 1 min. pr. iteration. In Figure 6.17 the time consumption of VISSIM can be viewed, compared to both Dynameq and TSA.

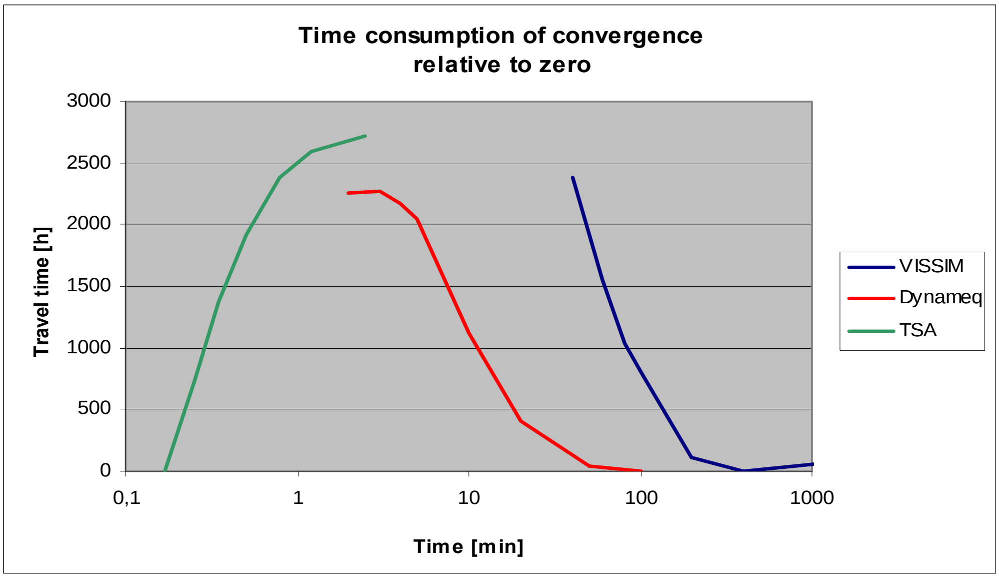


Figure 6.17: Actual time consumption for convergence.

6.5 Totals

In comparing the three softwares' overall performance, the totals of the simulations are compared. After extracting the overall summation for Travel-time, Distance traveled, Delay and Average Speed, Most of the values seem to be within what could be expected, while a few values seem to deviate from previous results.

Time-demand for extracting data from the different programs is very different, though:

VISSIM presents the user with a summation, containing only one total for each class (travel time, Delay, Distance, speed and vehicle count).

Dynameq presents a summation over time slices, leaving the user with a bit more material to work with, but still handling the data takes only minutes.

TSA presents results in the 4 previously described tables. For calculating these few values, a series of queries had to be made, in order to get a basis for summation.

Totals	VISSIM	Dynameq	Time Slice Assistant
Travel Time	5080,9 hours	6372,8 hours	4789,9 hours
Distance	296999 km	307871 km	317994 km
Delay	1703 hours	2862,1 hours	2060,8 hours
Speed (AVG)	59 km/h	60,6 km/h	55,6 km/h

Table 6.1: Total values of running simulations.

6.5.1 Travel-time

In Table 6.1, the values differ with 1583 hours, with Dynameq having the highest value of 6372,8 hours followed by VISSIM presenting a value of 5080,9 h. and TSA presenting a value of 4789,9 h. Previously in the assignment the low value of Dynameq was commented setting up theories of :

- Capacity, allowing free speed for a larger interval of flow, thus minimizing travel-time
- K_{jam}, an unrealistic high value of K_{jam}-density would allow for a screwed lower BPR curve, altering queue conditions.
- #veh, if for some reason a lower amount of vehicles were to enter the model this could be the result.

Though one or all of these theories could explain the low convergence value of 2800 h., and were thoroughly examined for error, the summation of *network_results* point to the fact, that a wrong summation of values has been made.

The lower value of TSA, might be explained by lack of delay in nodes. A reasonable assumption would be that almost any vehicle entering an intersection would experience some seconds of delay passing through.

6.5.2 Distance

In all three models the total km. driven is around 310.000 km. Ranking the values, VISSIM seems to have found the most optimal paths, followed by Dynameq and TSA with a maximum value of 317994 km.

6.5.3 Delay

Comparing the values of delay VISSIM clearly presents the lowest value of 1703 h. only making up 33% of the total travel-time, while the delay of both Dynameq and TSA makes up 43-44% of the total travel-time. One explanation of this division might be the difference between algorithms. Where the congestion is widely controlled by the interaction of the vehicles in VISSIM, it's the aggregated flows in combination with capacity that triggers the congestion conditions in both Dynameq and TSA.

6.5.4 Speed

One has to take into account both the number of cars driving on the road and their speed, in order to get the correct result, in terms of average speed.

As TSA reads a slightly lower average speed-value than the other softwares, while paradoxically driving more km in a shorter time, this points to an incongruity. The average relationship between speed, distance and time thereby differs between TSA, Dynameq and VISSIM. VISSIM seem to have found a balance where the driven kilometers actually correspond to the hours spend, based on the speed recorded, while Dynameq seems to swing the opposite way of TSA's incongruous readings.

6.6 Measurements

In VISSIM the possibility of introducing collection points gives the operator the opportunity to collect measurements at a specific location. Furthermore the collection point can be set to measure several factors. In Dynameq the export of results is basically given on a link, movement, node or path basis. TSA presents the user with 4 tables covering path, connectors and street segments.

Figure 6.18: Map of data collection points,

The placement of the collection points is supposed to get a representable set of measurements. See Figure 6.18. Data collected at M3, in the northbound lanes (blue), should represent free flowing motorway conditions. At the ramp leading from M3 to O3 at Dynamovej (Orange) the collection point should represent congested conditions on the motorway.

At the intersection of Dynamovej (Green) a queue counter was established to represent queuing within an identified problem area with large buildups. At the intersection of O3 and Frederiksundsvej, a queue collection (purple) was established to represent queues at a large intersection.

The colors of the rings can be traced to the outline of the following graphs.

Collection points could have been placed at several other places around the model, but because of time restrictions in the final phase of this thesis, representations was limited.

6.7 BPR

In VISSIM a collection point placed on the M3 motorway close to the exit at O3 gathering values of Speed, Queue delay time, occupancy rate and #vehicles with intervals of 15 sec. These values were used to form speed/flow, Flow/Density and Speed/density curves.

The Speed flow curve from the motorway does not present a clearly defined upper and lower curve, but does resemble the shape. The intervals along the flow-axis is a result of integer values of counted cars passing the collection point with in the 15 sec. An estimation of a Q_{max} value would in this case be very difficult to perform, but a qualified guess would place it right around 1600 to 1700 veh/h, maybe as high as 1800 veh/h.

The lower part of the curve reduces fairly fast towards a queue speed of 20 km/h, from the point of Q_{max}.

The lack of accuracy is for once a result of a restricted dataset, but also if other values describing the traffic had been implemented, like headway or occupancy, results could have been less aggregated.

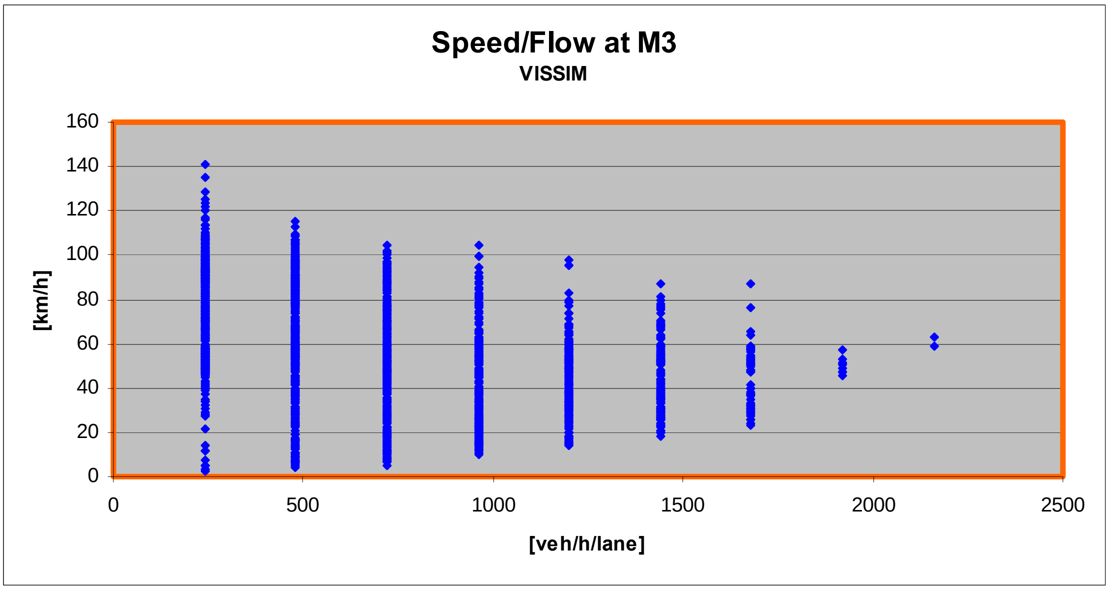


Figure 6.19: Measure of Speed/Flow on Motorway, VISSIM

Visualizing flow and occupancy a clear defined edge on the free flow part of the curve shows how the flow increases with an increasing density. A rough estimation of Dm around 15 to 18 veh/km, marks the change from Free flow to congested conditions, reducing flow as density increases toward K_{jam}, marking the upper limit of cars to fit one km of road (in one lane).

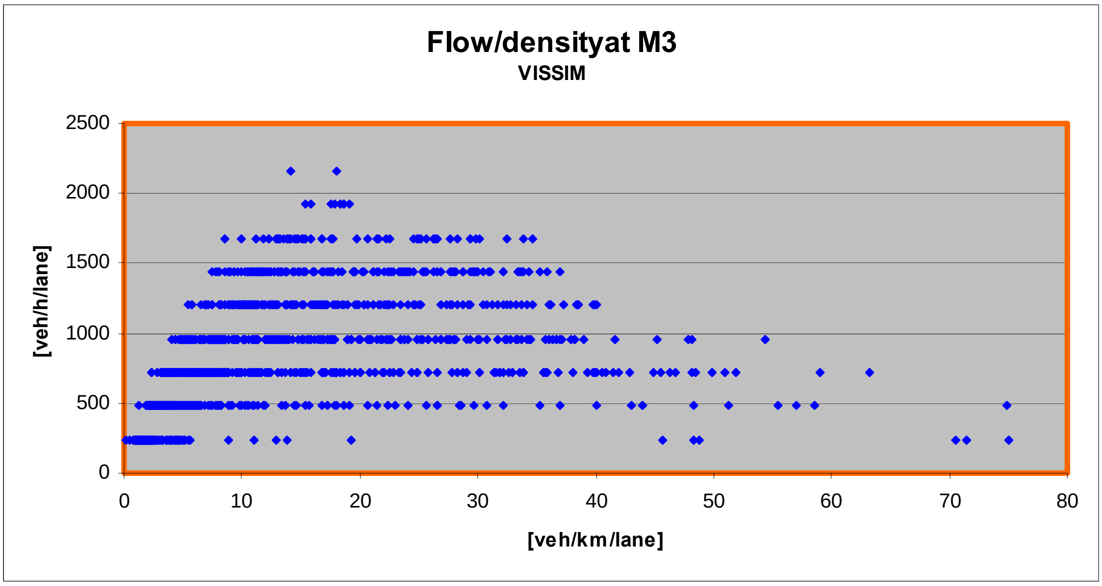


Figure 6.20: Measure of Flow/Density on Motorway, VISSIM

In theory a constant level of free speed should be observed for increasing density, till the Dm value is reached. Observing the Speed/density curve below this value seems to be non existing, reducing speed almost as soon as density starts increasing.

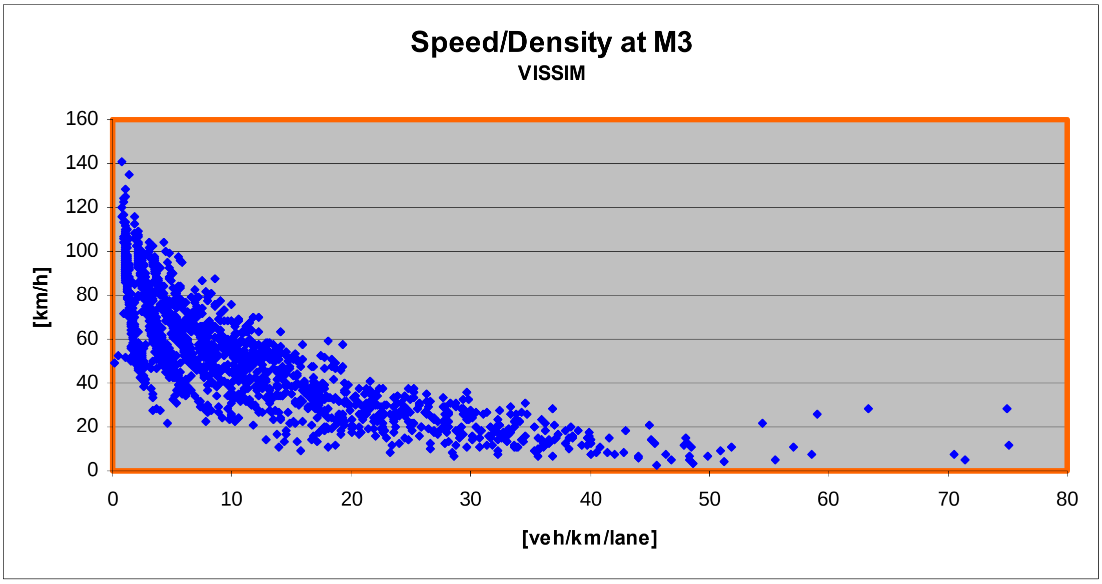


Figure 6.21: Measure of Speed/Density on Motorway, VISSIM

6.7.1 Speed and queue

Examining the relation between reduced speed and the forming of queues, the following graphs clearly show how the increase in traffic loads during the rush hour causes buildups where the traffic from O3 and those vehicles exiting the motorway merges. See Figure 6.22, Figure 6.23.

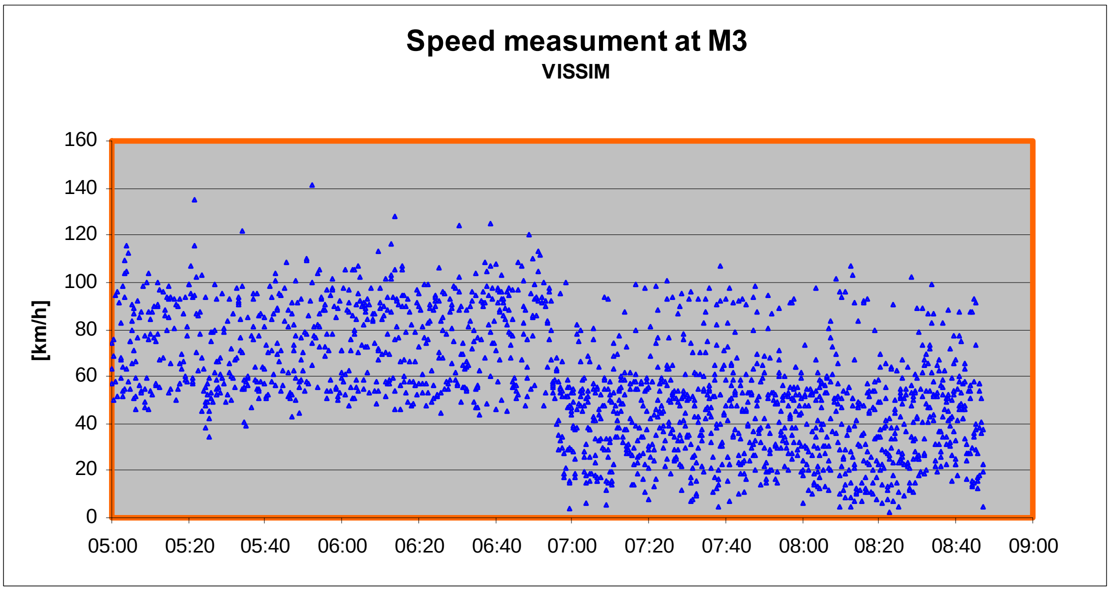


Figure 6.22: Measure of speed on Motorway, VISSIM

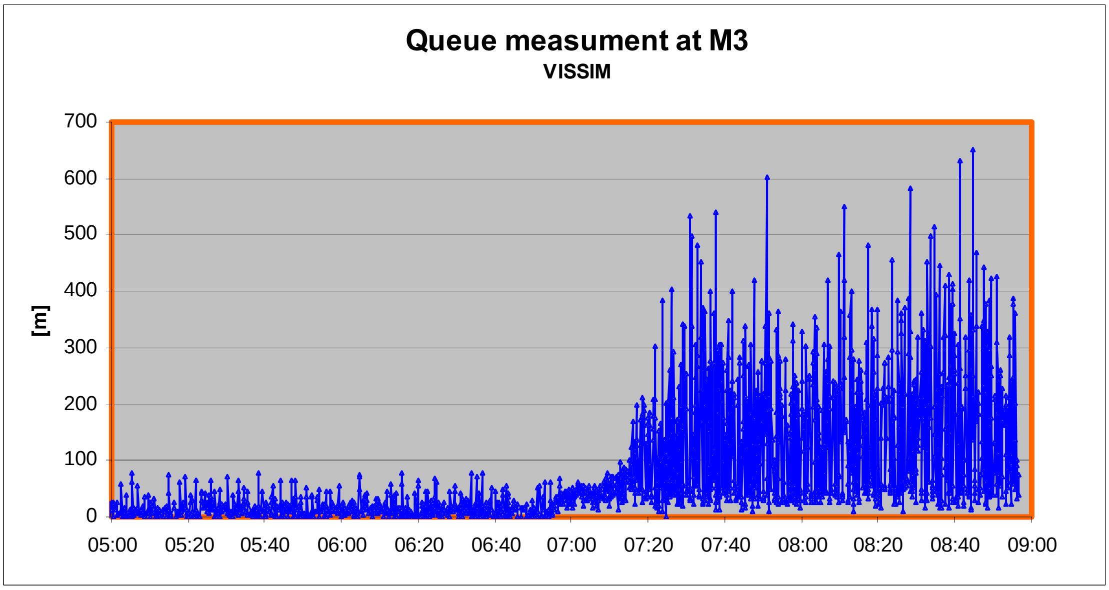


Figure 6.23: Measure of queue length on Motorway, VISSIM

In VISSIM one route chosen by vehicles traveling at O3 bound to leave the model by M3, is by the Hillerød motorway(HM) using the very capacity restricted left turn at O3/HM. As parts of this turning lane was made as one of a 3 lane link piece, vehicles positioned them selves on the outside of those vehicles already in the turning lane, reducing the capacity of O3 at this point. This bottleneck in combination with a signal plan for Dynamovej making a separate phase for left turning traffic (opposite direction, crossing the overloaded northbound lanes) resulted in fairy long queues forming on both O3 and M3.

Both Figure 6.24 and Figure 6.26 are made with the use of Queue counters in VISSIM. The queue counter for Figure 6.24 was placed on O3 collecting the queue caused by the left turn.

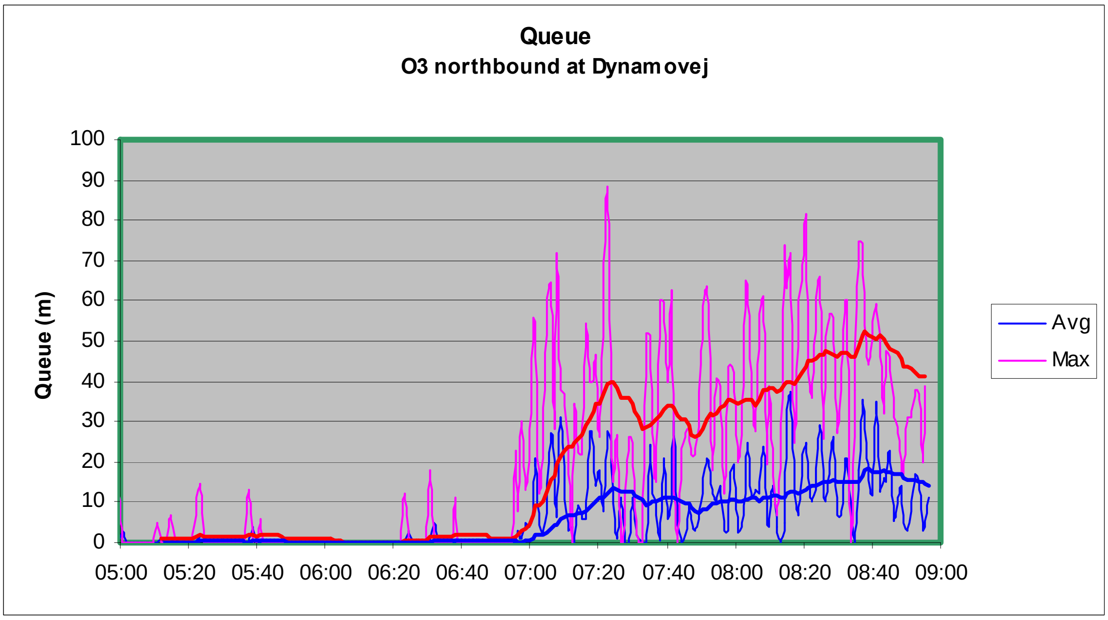


Figure 6.24: Queue measurement of northbound traffic at Dynamovej (Using queue-filter)

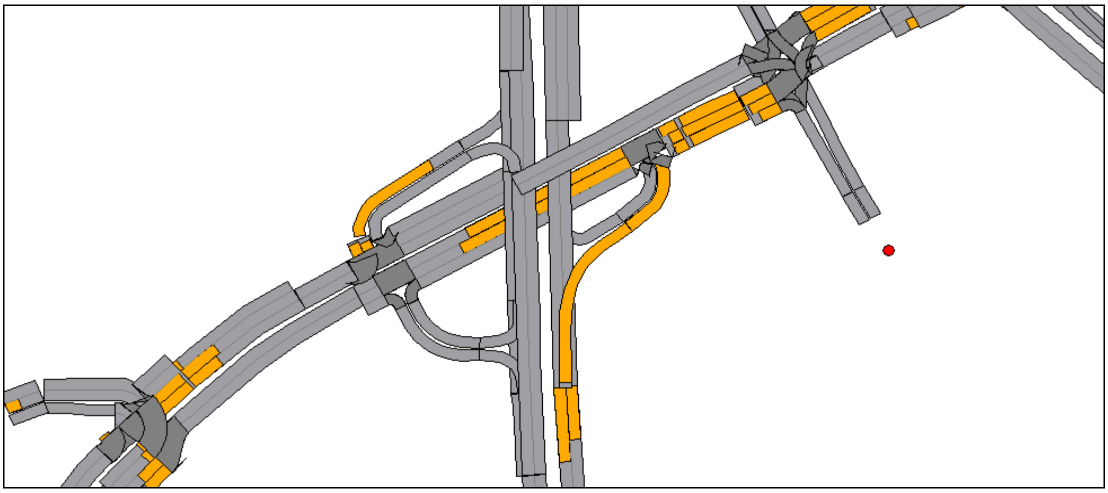


Figure 6.25: Example of queues visualized in Dynameq

In Dynameq it is possible to visualize the several results at the same time, identifying changes by an interactive time scale. In Figure 6.25 the same queues as measured in Figure 6.24, is displayed by Dynameq.

The intersection of O3 and Frederiksundsvej, being on of the more complex structured intersections in the model, interactions between the cycle-change at 07:30 and the change in queue length can be observed. In the half hour between 07:00 and 07:30 queues form relative fast. Due to the synchronization of green-wave almost all queues are dissolved shortly, only to form after a few minutes. See Figure 6.26.

.

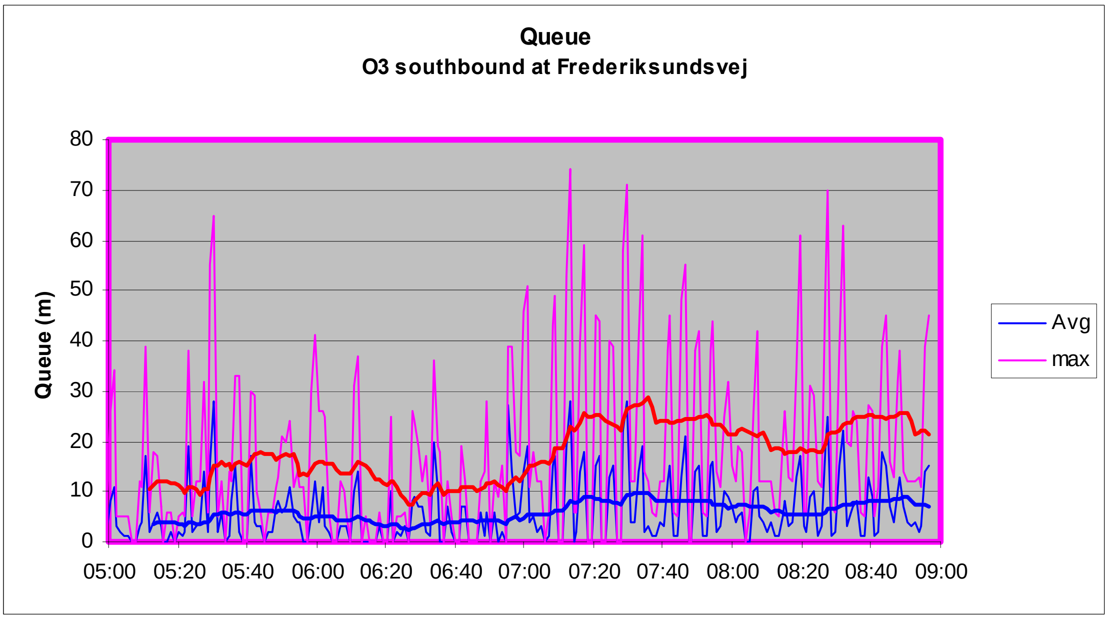


Figure 6.26: Queue measurement of southbound traffic at the intersection (O3 and Frederiksundsvej)

All three softwares allow for speed measurements, on some level of aggregation. At the southern collection point on M3, measurements were carried out to visualize the speed reduction caused from an increase in density, not yet forming serious congestion.

On the speed reduction graph of VISSIM (see Figure 6.27) a slight reduction in the speeds occurs around 07:00, as the rush hour sets in. Visualized on the graph is trend lines indicating a reduction of approximately 8 km/h.

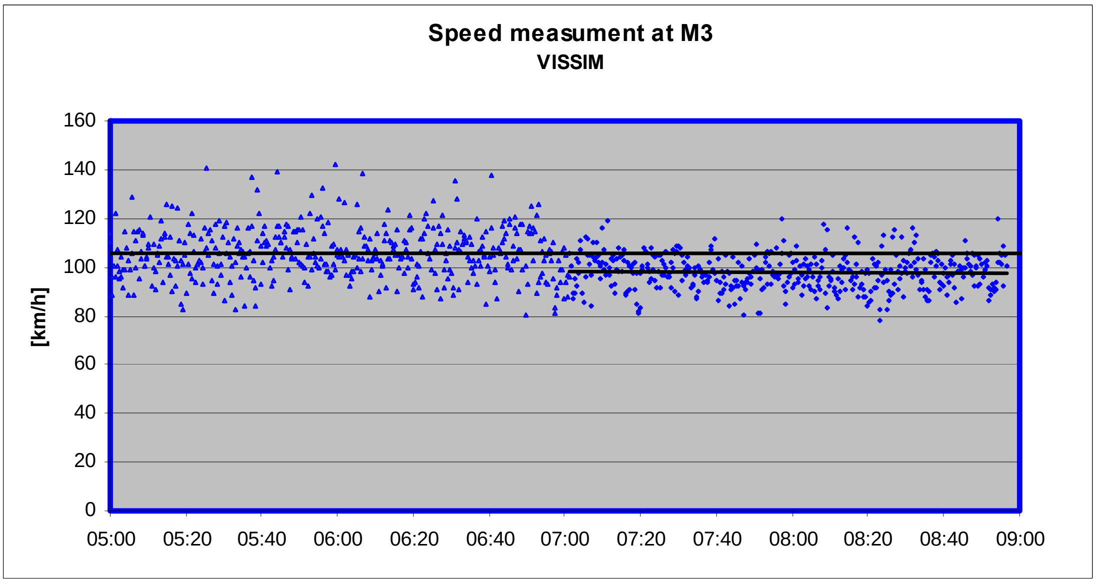


Figure 6.27: Measure of speed on Motorway, VISSIM

Observing the speed reduction modeled by Dynameq in Figure 6.28, only one value of speed pr time slice is given. A steady reduction of approximately 35 km/h can be traced from 07:30 toward 08:00, possibly caused by the increasing traffic load of the morning rush hour.

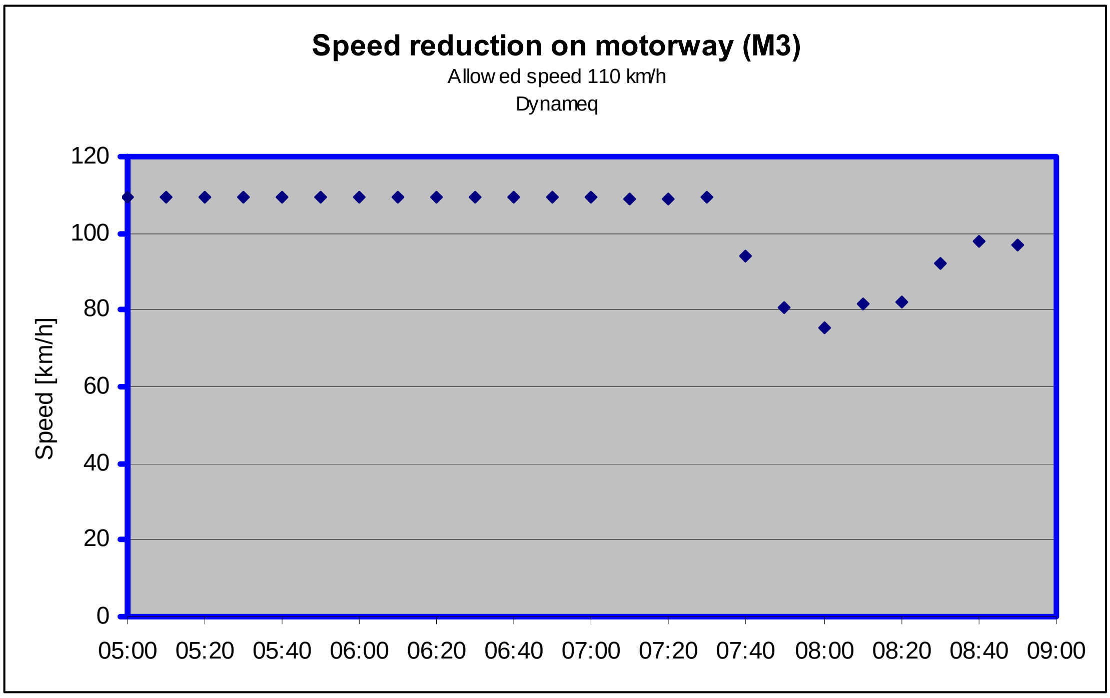


Figure 6.28: Measure of speed on Motorway, Dynameq

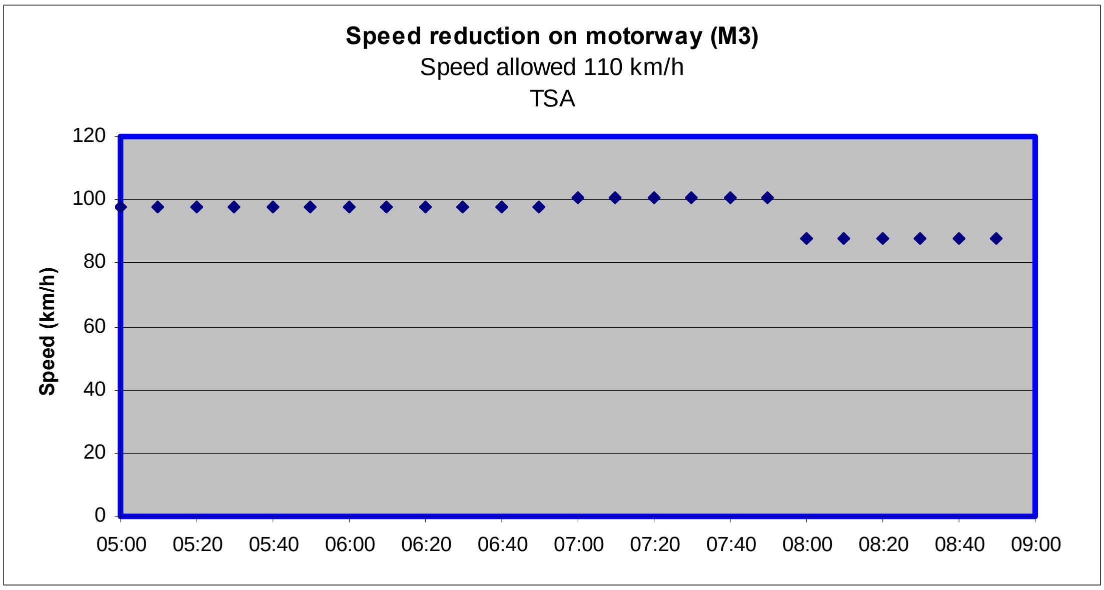


Figure 6.29: Measure of speed on Motorway, TSA

In TSA the average speed, for each OD time interval is given in the *TSAStreetSegmentResult* table. A more detailed visualization could be given by linking the *Link* table and *TSATimeSliceResults* table, dividing the *Shape_Length* with the *Travel-time*.

6.8 Delay

Delay as a function of observed speed is calculated by equation 6.1 where an accepted level of speed is given by $(0,8 \cdot \textit{Allowed speed})$[17].

$$Delay(h) = \sum_{links}\left((Travel\,time) - (Accepted\,travel\,time)\right) \cdot veh \qquad 6.1$$

In the visualized delay of Dynameq, levels indicate an unrealistic low values. See Figure 6.30 Parts of this deviation from the expected results could be explained by examining the BPR curve. Oppose to TSA, in the upper part of the speed/flow curve, no reduction in the free speed is assigned. As speed in Dynameq is a question of free or congested conditions a reduction of 20 % as an acceptable level, delay will only be recorded as driving conditions enter congested.

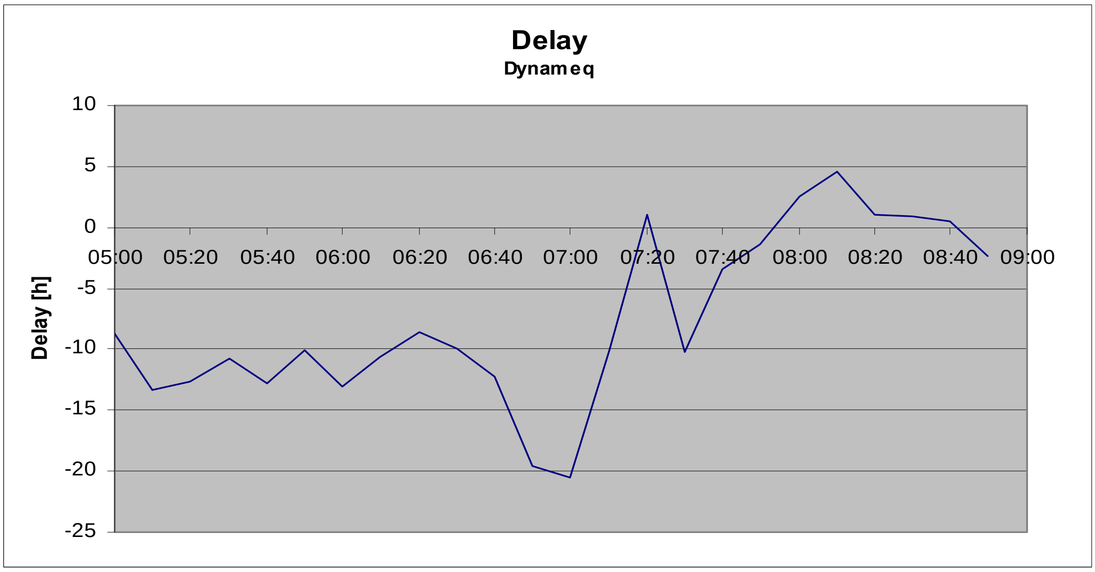


Figure 6.30: Delay of the model, Dynameq

The levels of TSA represent a fare more realistic value of delay following the patters of used travel time. Still the summarized value of 315 hours seems low. See Figure 6.31

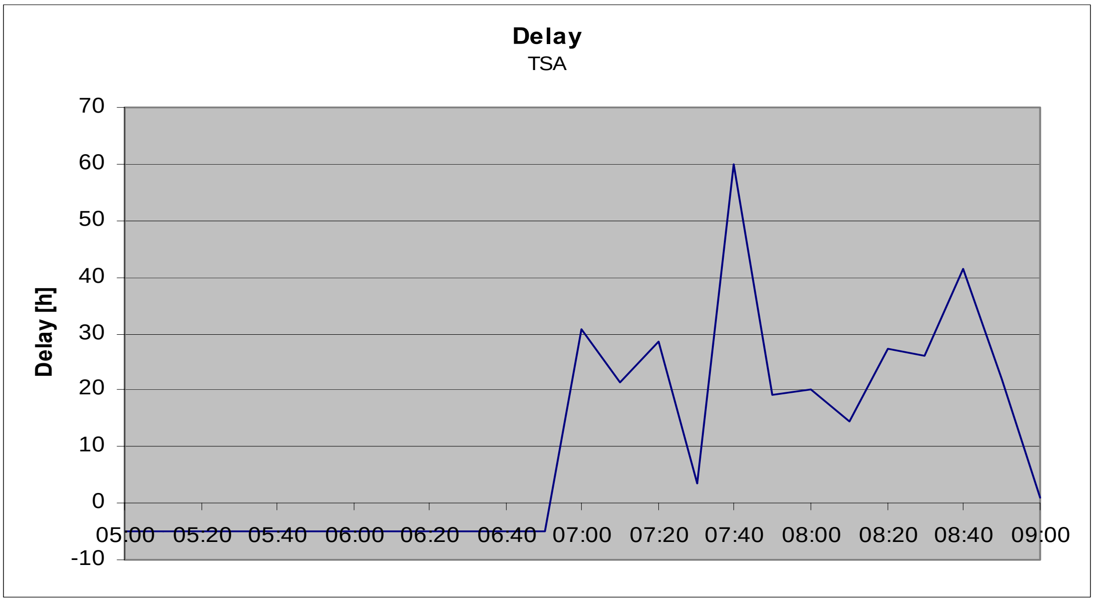


Figure 6.31: Delay of the model, TSA

6.9 MAPS

Maps poses as a potential powerful way of visualizing both fixed values like speed, delay, congestion, traffic loads or to show differences between senarios, visualizing changes difficult to observe as the differences only appearing on several maps. In order to visualize data in maps, is the connection between features and result-data. For TSA this connection is obvious since the calculations is performed on features in a geo-database. In Dynameq the connection is made as the Dynameq network can be exported to the shape file format, supported by ArcGIS. For VISSIM theoretically this could be done if the network of Dynameq and VISSIM was completely identical. In this thesis maps will only be presented from TSA and Dynameq.

In general the dynamic changes that characterize the DTA, is not particular “map friendly” because of the changes presented over time. In the following section the average values of the original three time periods make up the base for maps.

6.9.1 Speed

Presented in Figure 6.32 and Figure 6.33 are the difference between the average of measured speeds. Two sets of maps we made showing the difference between adjacent OD-time zones T2, T3 and T3, T4.

As Figure 6.32 shows the difference between early hours and rush-hour, this is where the most considerable changes appear. Comparing the two maps a clear difference appears here too, as the changes are much less in Figure 6.33.

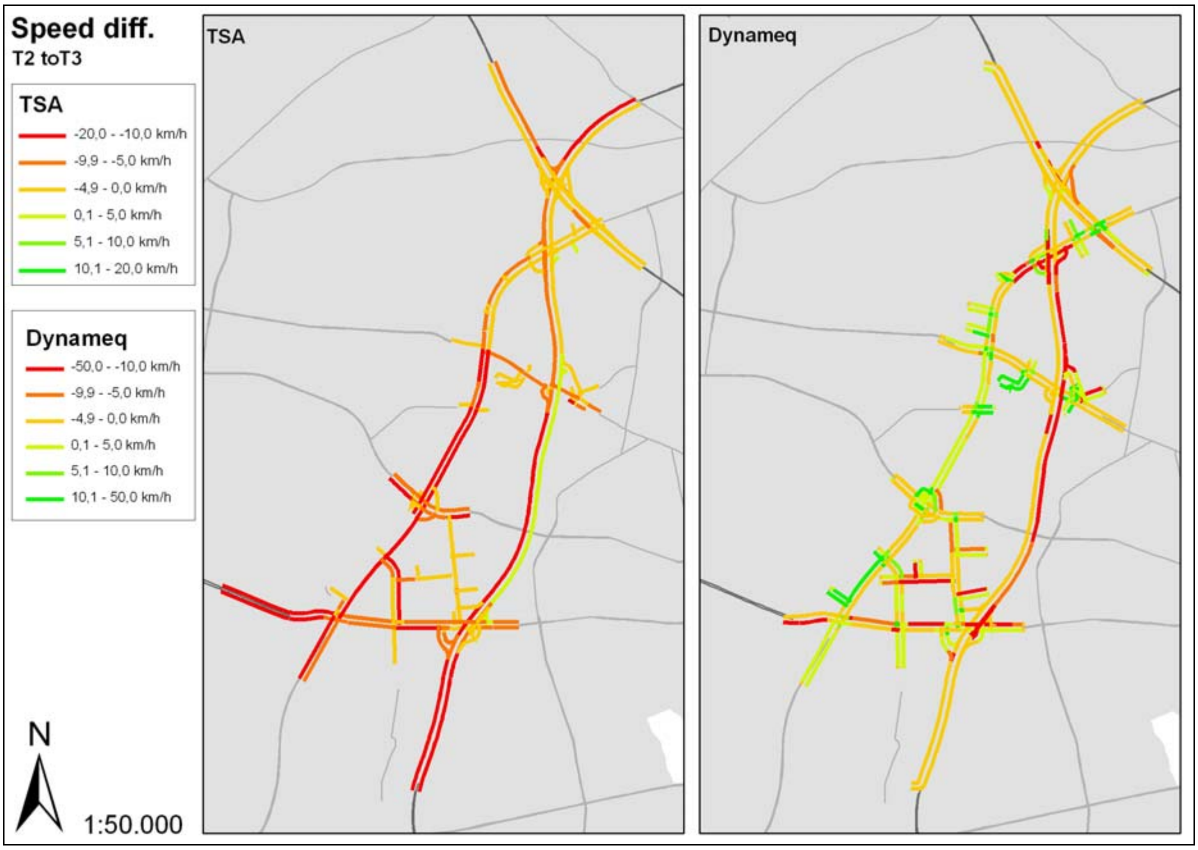


Figure 6.32: Speed difference: Difference between T2 and T3, left TSA and right Dynameq.

Figure 6.33: Speed difference: Difference between T3 and T4, left TSA and right Dynameq.

6.9.2 Delay

The Delay maps visualize an average of the delay experienced on each link in the scenario. Observing the maps from TSA and Dynameq a clear difference is noticeable, as TSA shows considerable higher values. This consistent with the delay graphs in Figure 6.30and Figure 6.31.

Compared to the visual observations of VISSIM a few clear inconsistencies are obvious. Two few areas of congestion were considerable during the simulations. On Frederiksunds motorway toward O3 and on O3, northbound, forming at the left turn at the Hillerød motorway. On neither of the two models these areas are shown as congested, which can be explained with a different choice of paths. In TSA a few of the links leading out of the model are displayed with a slight delay. Except for at capacity constraint this can only be explained based on the algorithm.

Figure 6.34: Speed difference: Difference between T2 and T3, left TSA and right Dynameq.

7 Discussion

This discussion has been formed as scoreboard, where green values represent benefit and red values, costs. Areas without color represent general discussions of general understandings.

7.1 Network Size

VISSIM	DYNAMEQ	TSA
Micro	Meso	Macro

There is little doubt that each of the three softwares are programmed for their respective, optimized network size. Depending on the algorithms making up the structure, an increasing modeling area will demand larger computational power and memory to simulate. As the focus move from micro-simulation toward meso- and macro-simulation, we encounter a change from analytic to deterministic algorithms. Capability of simulating traffic on larger networks, thereby results in resolution of the results diminishing. This reverse function, compared to the demand for computer power, categorizes the different levels of simulation.

As VISSIM handles real-time micro-simulation of traffic with a high resolution of output data, it naturally finds it place as the preferred tool of advisory-companies, since a substantial part of the assignments handled, fall within micro-simulation territory.

Dynameq covers the meso-level of simulation. Presenting more aggregated results than VISSIM, but giving the user the possibility of visualizing data on a timescale, Dynameq enters the arena as a tool better suited for those models reaching beyond the limit of VISSIM. As Dynameq is a fairly new product for meso-simulation, however, only a few companies have so far realized the potential of covering this size of the area.

Traditionally the static TA software has been the preferred tool for handling macro-simulation of large infrastructural changes. But there is little doubt that the TSA software offers a dramatically increased modeling performance, and when thoroughly tested for a higher degree of reliable results, it will be a strong tool for the modeling of dynamic, time-dependent traffic phenomena. This assessment is based on the assumption, that better documentation and a well-designed user interface is needed.

7.2 Data preparation costs

VISSIM	DYNAMEQ	TSA
✖	✖✖	✖✖✖

If a model should be built from scratch, making a micro-sized model would always be the preferred choice for a network, stretching for many kilometers. But assuming the assignment was to simulate infrastructure within the greater Copenhagen area, the perspective would surely change, since data are already available for TSA and thereby – on a lower level – for Dynameq.

Comparing the three softwares, given the present state of the OTM model, there is little doubt that the TSA model would be the easiest one to start up, but this is only because the model is established and continuously updated and improved. Working from scratch with TSA is thereby both slow and time-consuming.

For VISSIM goes, that a rather large number of traffic counts and observations must be carried out, in order to attain a suitable basis – even for a small model.

OD-matrices for Dynameq would have to be made the same way as the larger OD-matrices for TSA and TA. This would result in a huge amount of work, and continuous updates in the future. When TA or TSA data are available, these OD-matrices can be filtered to form a basis for Dynameq, but this still requires a fair amount of quality control.

7.3 Modeling costs

VISSIM	DYNAMEQ	TSA
✖✖✖	✖✖	✖

As VISSIM is meant for relatively small networks, and TSA for large networks, this comparison is made relative to the size of the model. So comparing, the fact of the price scaling with the size, is ignored.

Building a VISSIM network is not necessarily a particularly time-consuming process, but even in a small network, it takes a lot of extra time to prepare the basic model for simulation. A long range of decisions has to be taken on issues like distribution, rules, links, intersections etc., and features have to be set for classes of vehicles. Although these values can be transferred as a template, fixing these values for the model at hand is still time-consuming. Also, it should be mentioned that this work was done in VISSIM 4.16, whereas the later versions have a new user-interface for implementation.

In Dynameq two approaches can be chosen: 1) Constructing the model, rather like in VISSIM, entering the values by hand, or 2) using the export and import features for the shape file format supported by ArcGIS. In this process a lot of features can be added by database operations. Furthermore, if a larger model like the OTM-model exists, sections of this can be exported, by implicating only minor modifications.

In TSA, the model-structure is very simple. All attributes are found in tables of the same geo-database, making the updating of both features and metadata easy.

7.4 Simulation time

VISSIM	DYNAMEQ	TSA
✖✖✖	✖	✖

As figure 6.17 clearly shows, there is a big difference between the simulation time the three systems need, measured on a 100-iterations process. In rough figures, giving TSA the reference value = 1, the simulation-time is factor 100 for Dynameq and factor 1000 for VISSIM, and even if CPU-time is not the most precious resource, time is still a factor in project-management.

7.5 Theoretical consistency

When comparing the theoretical consistency, VISSIM, with its analytic approach, presents the operator with a wide range of possible adjustments, in order to resemble a real-life driving behavior. The algorithm simulates the interaction between vehicles, accessing range, speed, path-choice, gaps and a whole range of other features, characterizing driving behavior. In VISSIM the time-slices are substituted by a real-time simulation of traffic.

Contrary to this, the algorithms of both Dynameq and TSA are based on deterministic principles, looking at an aggregated flow, balancing flow while searching for the shortest path to include in the solution.

In Dynameq, each iteration estimates congestion, calculates costs for the shortest paths, assigns traffic, updates loads and identifies a new set of paths on which to balance flow.

TSA uses an inner and an outer loop to perform incremental iterations leveling out initial effects of the “all-or-nothing” iteration. Thus using less iterations to obtain convergence. In models of similar size, as the one simulated in this thesis, number of iteration for convergence means little. Still, usability must be addressed, since by example, tests on the entire OTM network shows, that calculation time will exceed 40 hours.

Although remarkable differences in convergence-levels for the tree softwares have been documented and discussed in this thesis, still a comparison between the overall speed, travel time and delay for the entire model (table 6.1), indicates a fair consistency.

But still there are differences: VISSIM and TSA will only need some 10–15 iterations to convergence to an acceptable level, while Dynameq will need some 30-40 iterations to reach the same level.

When examining the span of results from iterations by seeds, Dynameq is a bit fluctuous for a start, but after the same 30-40 iterations the span conforms to the other programs.

7.6 Details of results

VISSIM	DYNAMEQ	TSA
✖✖✖✖	✖✖	✖

After placing different types of collection-points in VISSIM, a large range of individual settings can be set for the features measured. This interaction gives VISSIM a wide range of adaptability, and an exceptional ability to expose problem areas. In addition to this, a limited number of pre-defined tables can be generated by choice of the operator.

In Dynameq only a general time interval between results can be set. But as values are only updated once every calculation interval, these results are dependent of the number of calculation chosen for each launch. For a range of results, additional tables, summarized by time-slices, are generated.

TSA produces only 4 tables. For connectors only, total vehicles are presented. On a zone-level a generalized cost is calculated, while link data are only presented as a total summation, or based on time-slices.

7.7 Output validity

VISSIM	DYNAMEQ	TSA
✖✖✖✖	✖✖✖	✖✖

As VISSIM simulates in real-time, giving the operator the opportunity to perform a visual quality control of the interactions in the model, usually it is possible to identify and correct irrational behavior and thus catch many errors and mistakes early in the process of simulation.

As Dynameq present results interactively, your attention will often be called to inconsistent values during visualization. In most cases this will be your entry for controlling further the detailed results in tables generated.

An example on this was the remarkable inconsistency in Dynameq's convergence level (fig. 6.14), which called for a comprehensive check-up.

As described in the 'Totals' section; this check-up included all default settings, operator values and all calculated values. No base data differed seriously from the values used in the other softwares, but when comparing the *network_results* from a single 100-iteration, it was quite remarkable that Vehicle Hour Travel Time (VHT) was reported (in good consistency with the other two programs) at a total of 6372,8 hours, while summarizing the total travel time for the three vehicle classes converged on a level of only 2800 hours. This inconsistency points to the fact that Dynameq generates the convergence data, either by summarizing wrong data or on a faulty basis. In all probability, an error in the progression of software updates has caused an internal inconsistency in the program software at hand.

From TSA, only processed results can give an indication of faulty values, and thus very little quality control can be performed. But since the model is less complicated, fewer possibilities for making errors present themselves.

7.8 Post processing features (Access and Excel)

VISSIM	DYNAMEQ	TSA
✖✖✖	✖✖✖	✖

As VISSIM exports a large range of text tables, quite a lot of post processing goes with the presentation of meaningful data, covering the problem. Possibly with experience, it will get easier to pick out spots of interest within the model, but as a novice, one tends to set up more collection-points in order to have a selection of representative data from which to choose.

Post-processing data from Dynameq is fairly easy, as the tables are either average values of time-slices, or countings, ready to be imported into Access or Excel.

In order to ease the post-process of TSA tables, the structure has yet to be optimized significantly, in order to minimize redundant work. One idea could be to copy the feature that handles multiple tables in each simulation, in VISSIM and Dynameq.

7.9 Presentation

VISSIM	DYNAMEQ	TSA
✖✖	✖✖✖✖	✖

Presentation must be compared to the area-size of simulation. Micro-simulation will focus on details, meso-simulation will report a fairly detailed image of a larger area, while macro-simulation results are presented as generalized values of the modeling network.

The main form of presentation from VISSIM is generally graphs, showing the alteration over time. But these results are supplemented by the option of generating a fairly advanced AVI movie, incorporating zooms, transformations and other features. This feature is a strong tool for presenting results for outsiders like politicians and officials, who are not familiar with the terminology of simulation.

As already mentioned, Dynameq features an interactive visualization tool. But since this tool is integrated in the software packet, this lacks the all-round purpose of communicating results without the operator present.

Compared to the unlimited distribution of VISSIM *.avi movies, Dynameq fails the demand for off-line presentation.

This could have been compensated with a map-generating feature for reporting results on a map basis, where successive changes in traffic patterns could have been visualized and presented, but this has to some extent been substituted by the ability of model-export to ArcGIS.

In TSA, visualization features are non-existing, the closest TSA comes to a visual presentation is the ability to join results to at map in ArcGIS.

7.10 Post processing costs

VISSIM	DYNAMEQ	TSA
✖✖	✖	✖✖✖

Post-processing in VISSIM is a time-consuming process, including the setup of collection points, the import of data, the selection process and finally some aggregation and / or visualization. All this easily ends up becoming a fairly long and time-consuming process.

Though this is probably implemented in newer versions of VISSIM, direct exports to the office package would be appreciated.

Processing the data from Dynameq is fairly easy, compared to the comprehensive range of data available. In general, the tables can either be directly joined with the features in ArcGIS – or this will call for a very limited range of summations.

8 Conclusion

	VISSIM	DYNAMEQ	TSA
Network size	Micro	Meso	Macro
Data prep.costs	✖	✖✖	✖✖✖
Modeling costs	✖✖✖	✖✖	✖
Simulation time	✖✖✖	✖	✖
Theoretical consistency	algorithms to be discussed		
Details of results	✖✖✖✖	✖✖	✖
Output validity	✖✖✖✖	✖✖✖	✖✖
Post processing features	✖✖✖	✖✖✖	✖
Presentation	✖✖	✖✖✖✖	✖
Post processing costs	✖✖	✖	✖✖✖

8.1 Initial considerations

Comparing VISSIM with the other two softwares, it's generally assumed, that the VISSIM modeling results and reported data sets have the highest validity of the three softwares. Also – in this thesis – the VISSIM results are taken as a valid reference for lack of empirical data for verification.

This validity in detailed reports is probably the reason why this solution is generally chosen by engineering companies in advisory and construction. For many of these, VISSIM is the de facto tool for traffic analysis and planning, even if the total costs for these benefits are high.

One important issue here is responsibility. VISSIM simulation, taking a lot of circumstances into consideration, including a lot of features when modeling and reporting to the client, makes it safe to stand by the recommendations given. Even if the initial efforts and costs to establish this detailed basis are high.

Another issue is transparency. The open car-following logic a single vehicle will meet in contemporary traffic – including all the dynamic scenarios – is shown in VISSIM. This makes it easy to both analyze possible reasons in interaction between traffic and structures, to formulate hypothesizes and to generate modeling data on detailed alternative solutions.

Till now the analytic approach has been totally dominant in the market up to the very limit of micro-modeling capacity. And when crossing this limit, a solution to overcome this would be that of sectionizing too large networks into a number of manageable entities … later in the process aggregating the results from these entities into one.

8.2 Core performance: Modeling and output validity

For a conclusion – considering the performance and validity of Dynameq and TSA – the position of the analytic VISSIM approach now seems questionable.

When focusing closely on *the core modeling performance* and *the output validity* – except for a singular explainable extreme – the deterministic principles of Dynameq and TSA provide valid overall results on speed, flow and delay for a fraction of the costs usually spent on meticulously detailed analytic micro-simulations in VISSIM. Other quality measurements, mentioned in 'Results' and 'Discussion', prove a fair consistency, when comparing these softwares relative to each other.

In fact, the simulation speed of Dynameq, reducing this by factor 100 is rather impressive, and for TSA a factor 1000 reduction is a dramatic performance improvement, compared to VISSIM micro simulation. Even if still an unfinished product, this indicates a large potential for dynamic modeling in TSA, compared to TA.

But when adding the other elements of the full workflow for a typical analysis-, modeling- and documentation project, this simple conclusion falls a bit apart, since the other program features mentioned fail to match the core performance.

8.3 The input process

When focusing – for a start – on *the input process*, the huge data preparation cost for a TSA macro simulation, will make the incentive for single project contractors to implement this new tool modest. But when it comes to administrative bodies in society, responsible for geographically comprehensive networks on the long continuous timeline, the initial efforts will not be single project related. The demand for a deep and updated knowledge of the network will call for this work to be done anyway, and when sharing these comprehensive data with project partners of contractors, these initial cost will be reduced to a matter of data exchange formats.

8.4 The output process

When focusing – next – on *the output process*, the obvious barriers for these new deterministic tools are those of reporting, post processing and presentation features. On these features, obviously TSA provides only the most basic information when it comes to network performance, and Dynameq is still in need for improvements, mentioned in the discussion.

Size of networks

In the introduction, it was assumed, that Dynameq – as a typical meso-simulation tool – would simply fill in a ‘meso gap’ between micro- and macro-simulation. The assumption was that the dynamic aspect, previously the privilege of micro simulation programs like VISSIM, would now be an affordable feature, even to networks, reaching from the borderline of micro simulation a bit into the area, previously reserved for macro simulation.

Now, digging a bit deeper, size is no longer the only determinator for a simple distribution of roles, when choosing modeling tool. Neither is the dynamic aspect of modeling, since this can be chosen for any areas of interest:

Local increase of dynamic resolution.

Within Dynameq a special feature allows the user to focus on a particular area of interest leaving the rest of the model, as a “support structure”.

In TSA, a similar feature allows for an increasing number of time-slices on link level. This concept might be a cost/efficient approach for a local ‘dynamic zoom’ into existing large networks, especially where resources have already been spent on modeling, in the Traffic Analyst toolbox.

9 List of figures:

Figure 1.1: The scenario 8
Figure 2.1: speed-flow, Flow-Density, Speed-Density 16
Figure 2.2: Speed / Flow curve: Free flow in green area, maximum flow in yellow area, congestion in the red area. 18
Figure 2.3: Flow / Density diagram 19
Figure 2.4: Speed / Density diagram 19
Figure 4.1: OD- matrix work process 26
Figure 4.2: Vehicles visualized by time and class. 29
Figure 4.3: Traffic intensity (blue: cars/hour) between kl: 6:00 and kl: 10:00 29
Figure 4.4: VISSIM OD-matrix structure 30
Figure 4.5: Dynameq OD-matrix structure. 31
Figure 4.6: TSA OD - matrix structure. 31
Figure 4.7: Connectors before and after 33
Figure 4.8: DOGS expanding cycles during the morning rushhour. 34
Figure 4.9: Green wave on Jyllingevej 36
Figure 4.10: Green wave on Ring 3, at 80 sec. cycle. 37
Figure 4.11: Green wave on Ring 3, at 120 sec. cycle. 38
Figure 4.12: Schematics of VAP program. 39
Figure 5.1: Principle of dynamic traffic assignment for VISSIM 40
Figure 5.2: Travel-time calculation 41
Figure 5.3: Speed distributions motorway, for (left) car/van, (right) truck/bus 45
Figure 5.4: DTA model in Dynameq 47
Figure 5.5: Flow / Density diagram 52
Figure 5.6: Flow / Speed diagram 53
Figure 5.7: Speed / Density diagram 53
Figure 5.8: Precedence probability for critical gap and -wait[xix] 55
Figure 5.9: FIFO test of equation 5.22 shown.[xiii] 61
Figure 6.1: Flow process of making VISSIM convergence data. 66
Figure 6.2: Flow process of making Dynameq convergence data. 66
Figure 6.3: Flow process of making TSA convergence data. 66
Figure 6.4: Convergence of VISSIM 67
Figure 6.5: Convergence of Dynameq 68
Figure 6.6: Convergence of TSA 69
Figure 6.7: Convergence by iterations, VISSIM 70
Figure 6.8: Convergence by iterations, Dynameq 71
Figure 6.9: Convergence by iterations, TSA 72
Figure 6.10: Convergence by iterations, TSA (30 min. extra timeslices) 73
Figure 6.11: A comparison of different inner loop iterations and extra time worth of time slices at 100 MSA iterations. 73
Figure 6.12: Travel-time at the 100. Iterations, all three softwares compared 74
Figure 6.13: Convergence by seeds, VISSIM 75
Figure 6.14: Convergence by seeds, Dynameq 76
Figure 6.15: Convergence by seeds, TSA 76
Figure 6.16: Standard deviation of convergence by seeds. 77
Figure 6.17: Actual time consumption for convergence. 78
Figure 6.18: Map of data collection points, 82
Figure 6.19: Measure of Speed/Flow on Motorway, VISSIM 84
Figure 6.20: Measure of Flow/Density on Motorway, VISSIM 84
Figure 6.21: Measure of Speed/Density on Motorway, VISSIM 85

Figure 6.22: Measure of speed on Motorway, VISSIM 86
Figure 6.23: Measure of queue length on Motorway, VISSIM 86
Figure 6.24: Queue measurement of northbound traffic at Dynamovej (Using queue-filter) 87
Figure 6.25: Example of queues visualized in Dynameq 87
Figure 6.26: Queue measurement of southbound traffic at the intersection (O3 and Frederiksundsvej) 88
Figure 6.27: Measure of speed on Motorway, VISSIM 89
Figure 6.28: Measure of speed on Motorway, Dynameq 90
Figure 6.29: Measure of speed on Motorway, TSA 90
Figure 6.30: Delay of the model, Dynameq 91
Figure 6.31: Delay of the model, TSA 92
Figure 6.32: Speed difference: Difference between T2 and T3, left TSA and right Dynameq. 94
Figure 6.33: Speed difference: Difference between T3 and T4, left TSA and right Dynameq. 94
Figure 6.34: Speed difference: Difference between T2 and T3, left TSA and right Dynameq. 95

10 List of tables:

Table 3.1: Times for turning movements....................21
Table 3.2: An overview of route choice models....................22
Table 4.1: Example of filter definition....................27
Table 4.2: Vehicles by time and class....................28
Table 5.1; values of Logit (left) and Kirchhoff’s logit functions (right)....................43
Table 5.2: Speeding decisions for vehicles in VISSIM....................44
Table 5.3: Speed distributions....................45
Table 5.4: Driving behavior parameters....................46
Table 5.5: Levels of traffic distribution....................50
Table 5.6: Times for turning movements....................55
Table 5.7: Times for turning movements....................62
Table 6.1: Total values of running simulations....................79

11 References

[1] Arial photos from: Ejendomsfakta.netborger.dk

[2] Teknisk forvaltning, Københavns amt; intersection plans.

[3] DTF, 2005: ”Opdatering af turmatricer og OTM 4.0”

[4] Ortuzar, Hensher og Jara-Diaz, 1998, paper:
”Travel Behavior Research: Updating the State of Play”

[5] FHWA, 1974; “Traffic Assignment”

[6] MTC, 1994

[7] TRB et Al., 1999; “Proceedings of the Seventh TRB Conference on the Application of Transportation Planning Methods”

[8] Rapidis, 1995; ”Traffic Analyst Help”

[9] INRO, 2006; ”Dynameq user’s manual”

[10] Hans Martin Kaae Johansen, 2006; ”Trafikafvikling på motorveje”

[11] www.vsp.tu-berlin.de/archive/sim-archive/papers/agdb/html/agdb.html
Section: DTA

[12] lbxml.ust.hk/th_imgo/b805492.pdf

[13] Nielsen, 2007; “Additional notes for Time Slice Assistance”

[14] Ortúzar, Willumsen, 1990; ”Modelling Transport, 3. edition”

[15] Nielsen,2005; ”Vejforbindelse mellem Lyngbyvej og Gittervej
–Et eksempel på braess paradox”

[16] Material from course 13130 traffic models.

[17] Trafikministeriet, 2004: ”Projekt trængsel”

[18] Vejdirektoratet, 2004; ”Evaluering af DOGS”

[19] INRO; ”Dynameq user manual, Appendix A”