Dette resumé er udgivet i det elektroniske tidsskrift
Artikler fra Trafikdage på Aalborg Universitet
(Proceedings from the Annual Transport Conference at Aalborg University)
ISSN 1603-9696

Dynamic Traffic Assignment in a large-scale transport model (COMPASS)

Thomas Kjær Rasmussen, tkra@dtu.dk, DTU Management
Bjarke Brun, bb@rapidis.com, Rapidis ApS
Christian Overgård Hansen, christian@cohtrafik.dk, COH ApS

Abstract
A large-scale Dynamic Traffic Assignment (DTA) has been developed for a new activity-based traffic model for the Greater Copenhagen Area (Compass) to capture network dynamics that is not possible in static route choice models. The DTA model improves the estimate of peak hour travel times and provides similar or even better link load accuracy than the static route choice model.

1. Background and purpose
A realistic and well-functioning traffic assignment model constitutes an essential part of any Transport Model used for e.g. evaluating the effects of potential infrastructure investments or policies. Historically, models spanning a considerable area has used either a static or pseudo-dynamic approach for conducting traffic assignment. Static traffic assignments load all demand onto the network simultaneously. Congestion is calculated by comparing traffic on the link with capacity in the timespan under consideration e.g., the rush hour. The Danish National Transport Model (LTM) use a pseudo-dynamic approach (Rasmussen & Hansen, 2021). This has a more detailed representation of temporal variations, since typically the time horizon modelled is split in certain intervals (e.g. 10 time periods across a day) and trips start in certain time intervals and traverse the network as time progress. A whole day may thus be calculated with trips ending when they reach the destination. Congestion is calculated for individual time intervals. However, even such an approach is limited in its capability of representing network spatial and temporal dynamics accurately, especially in congested urban settings where traffic demand fluctuates considerably and queues spill-back across links and cause add-on effects.

Dynamic Traffic Assignment (DTA) models address these issues, providing a highly disaggregate temporal and spatial representation of the network dynamics. Moreover, DTA treats trips individually and each trip has an individual start time, and time is modelled at a far more disaggregate temporal resolution. In particular, DTA models model in detail queue lengths and spill-back effects at intersections, lane merging etc. giving a more detailed representation of network dynamics than static route choice models.

Recently, a DTA model based on the ideas developed by Bliemer (2003) has been developed by Rapidis. The DTA model is used in a new activity-based model for the Greater Copenhagen Area (the COMPASS model). Section 2 describes the principles of DTA, while Section 3 gives an in-depth description of the approach taken and algorithms used. Section 4 presents a few examples of output from the DTA COMPASS-model.

2. DTA principles

The modelling of route choice is similar to the approach taken in the static route choice model (Brun et. al, 2020) with the exception that it treats individual trips with an individual departure time, i.e. input trips are specified as individual trips with a specific starting time. The following focuses on the modelling of the flow dynamics which constitutes a major difference compared to other models used in practice in Denmark.

The flow dynamics on each network link is calculated in one minute intervals based on a model where each link is split into a queueing part and a moving part, as illustrated in Figure 1.

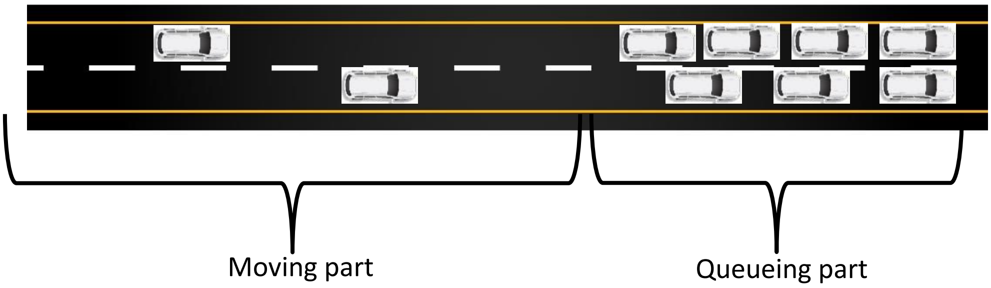

Figure 1: Representation of link

The queue length is determined by the number of vehicles queuing on the link. It is assumed that the head of the queue is always located at the end of the link and that it propagates towards the start of the link according to its length - denote the relative length of the queue as QS (queue share), then the relative length of the moving part is 1-QS. When QS=1, the link is fully occupied by queueing vehicles.

Queues build due to insufficient receiving capacity of the downstream link. For example as the result from lane merging, an oversaturated intersection, or queue spill-back. The main concept is to calculate the change in queue in small time steps, comparing demand in passenger car unit (PCU) outflow of each link to the current inflow capacity of the next link. If demand exceeds inflow capacity, queue is accumulated, and if it is lower than inflow capacity, any existing queue can be reduced. The inflow capacity of the next link is determined by its nominal capacity, the current queue length and flow.

The travel speed in the queueing part is assumed proportional to the free-flow speed (see Section 3.3.1), while the travel speed in the moving part is calculated from speed-flow relations using the BPR-formula. Consequently, the total travel time of traversing the link can be compiled from link length, the QS, the speed in the queueing part and the speed in the moving part.

DTA uses the same intersection modelling as the static route choice model of Compass. The only difference is, that the turn congestion delays are not used directly. Instead the calculated turn capacities are used as input for the queue length (QS) calculation described above. The turn capacities are modelled using average traffic flow per time interval, typically 1 hour.

An elaborate description of the various model components can be found in Rapidis (2021).

3. Algorithm

3.1 General algorithm

In general, the DTA model identifies a stochastic user equilibrium. This is done through an iterative algorithm that iterates between demand (route choice of individual trips as response to network travel times/costs) and supply (network travel times as response to route choice of individual trips). Instead of using traffic loads per time interval from the previous iteration to calculate network edge travel time, DTA also calculates queuing per link per time increment, and include the queue in the travel time calculations. As mentioned above, the calculation of the queues is based on tracking the queue length changes by comparing desired traffic (PCU) outflow of each link to the current inflow capacity of the receiving link.

The general DTA algorithm is structured as follows:

-For each iteration *i*:

I) Based on traffic loads and queues from previous iteration (Network Loading): Calculate new network edge travel times for each time increment

II) For each trip *t*:

-Search for new relevant paths to consider for trip *t*

-Update path set and distribution of flow between these for trip *t,* including threshold-based removal of worst path from path set

III) Network Loading: Based on trips, paths, path distributions and travel times, load traffic onto paths and update queue for network edges

- If final iteration reached:

I) For each trip *t* calculate Level-of-Service (LOS)

Updating the queues for the network edges as part of the network loading is computationally burdensome. Thus, in the initial iterations only 'simple' network loading is performed ignoring the effects of queues. It saves time and generates an initial set of routes to avoid flip/flop behavior in first iterations. Network loading and calculation of travel times i.e., steps III) and I), is outlined below.

3.2 Network Loading

In the network loading, queue lengths are updated per *time increment* (tix) corresponding to one minute. Moreover, for each edge the following is tracked in each tix:

- Inflow – the traffic entering the edge in the tix
- Outflow – the traffic exiting the edge in the tix
- PassFlow – the traffic which completely traverses the edge in a given time tix
- Queue – the current queue
- TravelTime – the current edge travel time.

The network loading algorithm updates traffic flows, edge queue sizes and travel times, based on the path sets and path distribution for the input trips. It starts from an empty network in the first time step (tix, τ), and then successively steps through each time increment, loading vehicles onto the network according to their departure time and track their progress as time increments, recording the corresponding network edge attributes for each time increment. More specifically, the algorithm works as follows:

1. Clear existing traffic, flow, queues
2. Load traffic in time increments to calculate queue sizes:
   For each time increment τ, starting from the first
   a. For each trip *t*
      i. For each path *p* of *t*; if not at end of path
         1. Calculate progress (current edge *e* and measure) along edges of *p* at time of τ, using speed from previous iteration combined from flow speed and queue speed based on queue for τ-1 for edge *e*
         2. Record PCU inflow for path *p* for τ for each edge *e* of *p* if entering edge in tix τ
         3. Record PCU outflow for path *p* for τ for each edge *e* of *p* if leaving edge in tix τ

         Until end of tix τ or end of path *p*
   b. For each node/intersection *n*
      i. Update queue sizes on edges to *n*
3. Update edge travel times for all edges for all time increments using new queue sizes
4. Load traffic in time intervals to calculate speeds from the speed-flow curve:
   For each trip *t*
   a. For each path *p* update traffic load in time intervals along all edges of p

3.3 Travel time and queue size calculation

3.3.1 Travel time on edges

Each edge is as explained in Section 2 split in two parts – a queueing part and a 'moving' part. Given the queue size *Q* in PCUs for a time increment τ, the queue share QS of the edge is calculated as:

$$QS(\tau) = \frac{Q(\tau) * PCUQueueLength}{EdgeLanes * EdgeLength}$$

Where *EdgeLanes* and *EdgeLength* refer to the number of lanes and length of the edge. For simplicity and computation efficiency, the vehicle density (*PCUQueueLength*) is approximated by a fixed length.

The queue share can exceed 1 which may be counter-intuitively. But it is useful in order to ensure convergence. Also, it can be argued that queue shares greater than 1 can be viewed as a measure of the queue density, at least to some degree.

The travel time per distance in the queuing part TT_{Queue} is assumed proportional to the per distance free-flow travel time, scaled by a *link-type* specific parameter.

The travel time on the moving part of the link, Tt_{Flow}, is calculated based on the length of the moving part and the speed-flow-curve speed calculated using the average traffic load in the time interval that the current time increment falls in. Based on the above, the travel time per edge for a given time increment is calculated as

$$Tt = Tt_{Queue} * QS_n + Tt_{Flow} * (1 - QS_n)$$

In Figure 2, the combination of a flow-dependent speed on the moving part and detailed modelling of queue is illustrated.

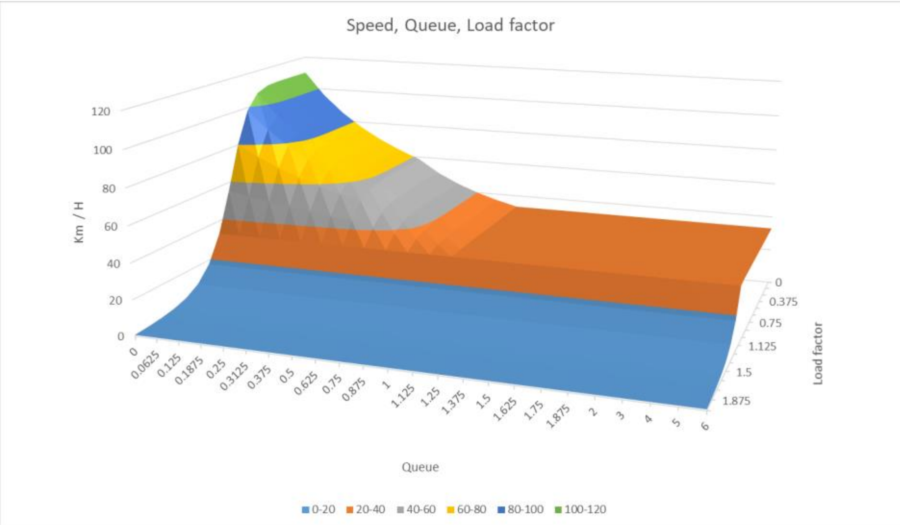

Figure 2: Speed as function of load factor and queue

In the case (corner) with 0 load and 0 queue, the speed is the free-flow speed (in this example 110 km/h). Increasing the flow (with 0 queue) will follow a classic speed-flow curve. Increasing the queue share, a larger share of the link will be queued and the speed decreases accordingly.

To calculate the total edge travel time of a trip/path entering an edge at a time which falls in time increment τ (and the leaves in time increment τ +n), the partial travel times in each time increment is summed, starting with time increment τ until the end of the edge is reached.

3.3.3 Calculation of queue size

The general assumption is that queues are caused by changes in network capacity, either changes in nominal capacity due to network topology (e.g., a change in the number of lanes from one link to the next); or varying changes such as intersection turn capacities which vary over time according to traffic load, or link capacity changes which we model as varying according to queue.
The queue sizes are updated successively starting at tix=0, calculating queue for entire network in the tix, and then moving on to the next time increment. A starting queue size of 0 for tix=0 is used for all edges. Queues are recorded on network links and calculated for network nodes. The approach varies across types of nodes (simple nodes or nodes in intersections with turn edges), but the basic principle of the queue calculations is to compare the desired node/intersection flow on the incoming edges (measured in PCUs) to the capacity of the outgoing edges and/or turns. If flow exceeds capacity of outgoing edges, queue is accumulated on ingoing edges, and if flow is lower than capacity, any existing queue can be reduced. More details on the computation of the queues can be found in Rapidis (2021).

4. Examples

The DTA model produces numerous output, both disaggregate and aggregate. Below are a few examples of outputs of the model, more details can be found in Hansen (2021).

Figure 3: Estimated travel time by car from Hareskov to Toldbodgade on a working day in 2017

shows travel time in direction from Hareskov to Toldbodgade. The x-axis shows the time-of-day in minutes and travel time is shown in minutes on the y-axis. There is as expected long travel times in the morning peak which corresponds very well with Google-data. The travel time in the afternoon peak is much lower because commuting is primarily in direction towards the city. This is also in line with observations from Google.

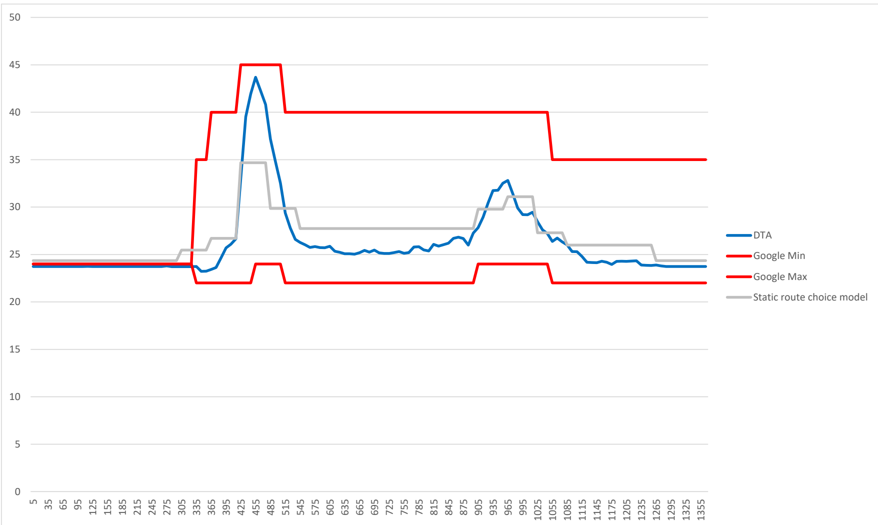

Figure 3: Estimated travel time by car from Hareskov to Toldbodgade on a working day in 2017

Figure 4: Ratio between DTA-estimated speed and free flow speed in the morning peak from 7 am to 8 am on roads in the M3-corridor shows the average ratio between estimated speed and free flow speed for the morning peak (7-8 am) in the M3-corridor. Red segments indicate heavy congestion and very low speed. In particular, low speeds are estimated on Hillerød Motorway from Farum to M3 and on M3 between Buddinge and Frederikssunds Motorway. Speeds on the most congested motorway links are on average reduced to 30 to 60 km/h. It reflects well the observed traffic situation in the morning peak.

There are few red spots on local roads. Some of them are probably caused by large delays in prioritized intersections because heavy traffic on the main road makes it difficult to enter it from the local road. Others may be caused by large connector loads onto the local road segment or simply inaccurate input data.

Figure 5 shows the expected queueing leading up to bottlenecks on Hillerød motorways and M3. The model also estimates queueing on Ring 4 between Hillerød and Frederikssund motorway caused by intersections along Ring 4 which reduce capacity.

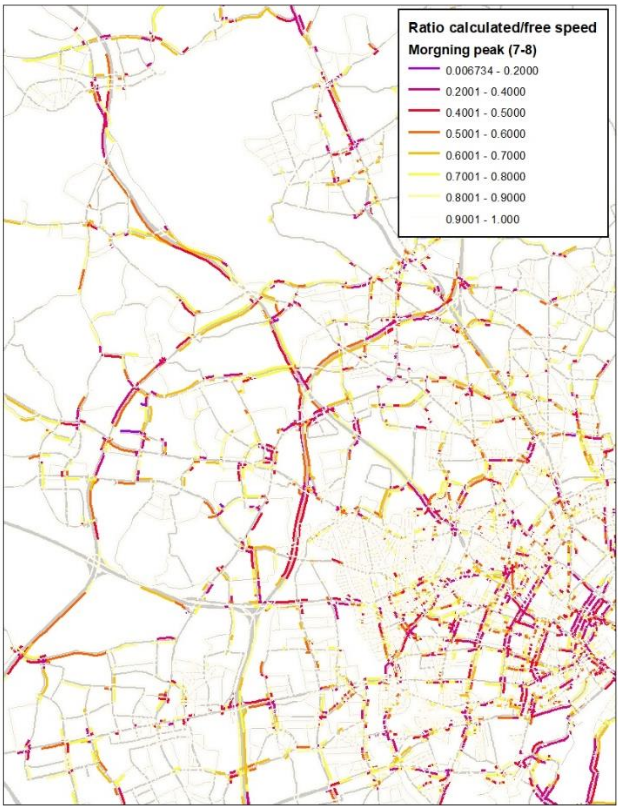

Figure 4: Ratio between DTA-estimated speed and free flow speed in the morning peak from 7 am to 8 am on roads in the M3-corridor

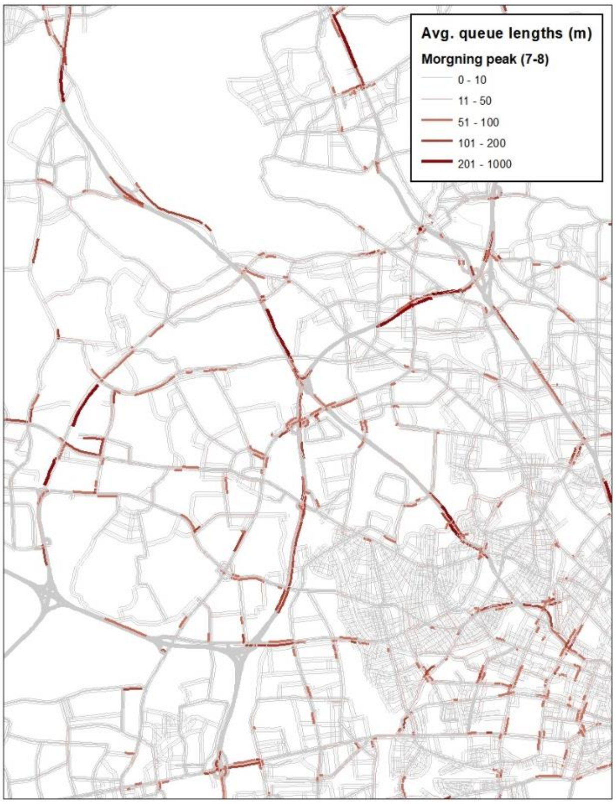

Figure 5: Queue lengths (meter) in the morning peak from 7 am to 8 am on roads in the M3-corridor

5. References

Bliemer, MCJ. (2003). *INDY: New analytical multiclass dynamic traffic assignment model*. 66-68. Abstract from NGSIM workshop Sitges, Spain.

Brun, B., Rasmussen, T.K. and Overgård, C. (2020). Compass. Static Road Route Choice Model (Probe 6.4). Version B, 25 July 2020

Hansen, C.O. (2021). *Final validation of the DTA-model (Probe 6.5B and 15B)*. Doc.no. 51024-024, 17 February, 2021.

Rapidis (2021). Traffic Analyst Manual revision 7.1.

Rasmussen, T.K. and Hansen, C.O. (2021). *Road traffic assignment in LTM 2.3*. DTU, 21. January 2021