citilabs

Dynamic Traffic Assignment and Mesoscopic Simulation with Cube Avenue

Colby M. Brown, PTP, Citilabs

SEFL FSUTMS User Group July 8 Meeting

Traffic Analysis Tools and Specialized Applications

citilabs

Transportation modelling tools

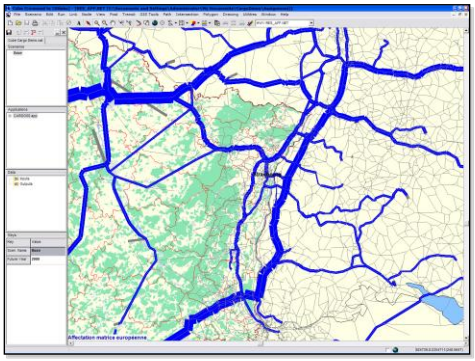
Macroscopic and Static Modelling

Mesoscopic and Dynamic Modelling

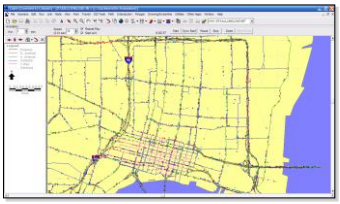
Microscopic and Dynamic Modelling

citilabs

A Multi-Resolution Traffic Modeling Suite



Cube Voyager for regional flows



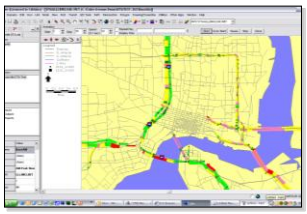
Cube Avenue for dynamic traffic assignment and bottleneck simulation

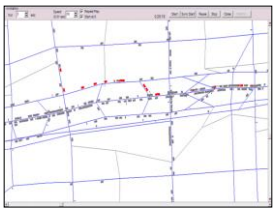
Cube Dynasim for corridor/subarea operations

citilabs

What is Cube Avenue?

- **Avenue is an optional add-on to Cube Voyager that enables *Dynamic Traffic Assignment* with *Mesoscopic Simulation*:**
  - ***Dynamic Traffic Assignment***
    Routes and flow rates change during the model period based upon congestion
  - ***Mesoscopic Simulation***
    Packets of vehicles are simulated as they move through the network
- **Easy to implement – because it uses the same data formats and scripting language as highway assignment in Cube Voyager**
- **Flexible – supports many of the same features e.g. multi-class, generalized cost...**

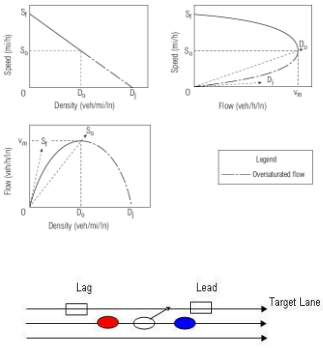




citilabs

Congestion: the key to model scale

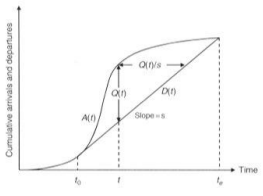
- ***Macroscopic*** **models evaluate congestion using speed-flow curves – based on fundamental traffic engineering relationships**
- ***Microscopic*** **models simulate individual vehicle trajectories on a detailed network and use behavior models (e.g. car following, gap acceptance) to predict second-by-second driver responses to en-route events**

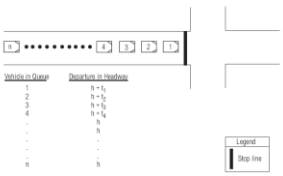


citilabs

Mesoscopic Traffic Modelling

- **Typically** ***simulate*** **movement of trips along their routes at some detailed resolution (packet)**
- **Discretely model traffic** ***queues*** **in network at bottlenecks (e.g. intersections, ramps, tolls)**
- **Traffic** ***stream*** **performance is typically still evaluated using aggregate macroscopic (e.g. speed-flow) relationships**

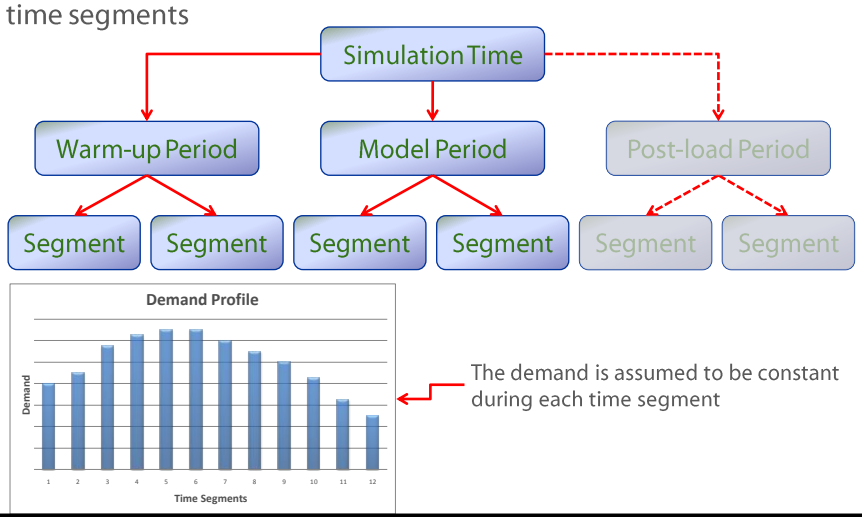




citilabs

Cube Avenue – Dynamic Traffic Assignment (DTA)

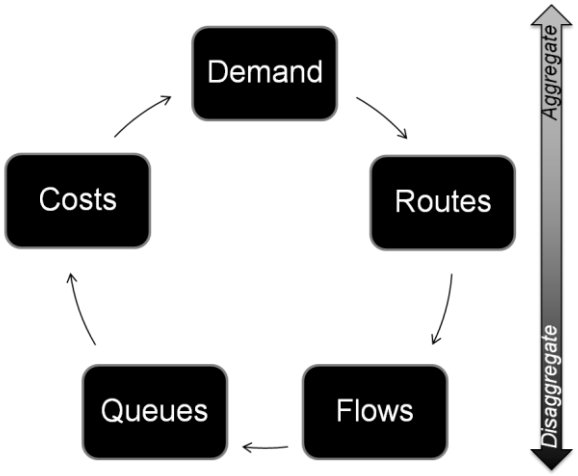
- The model duration is explicitly defined and divided into smaller time segments



The demand is assumed to be constant during each time segment

citilabs

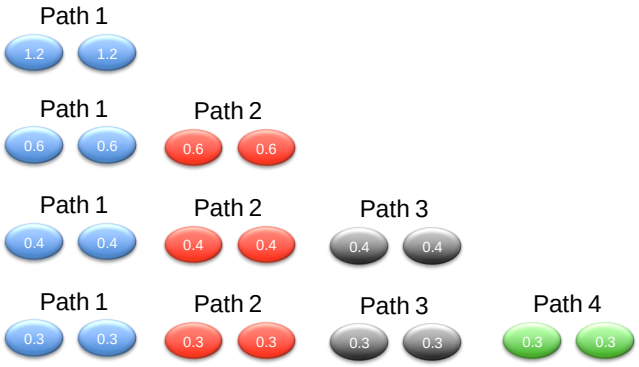
The temporal detail cycle



As Cube Avenue processes the various kinds of information used in dynamic traffic assignment, it moves between different levels of aggregation.

citilabs

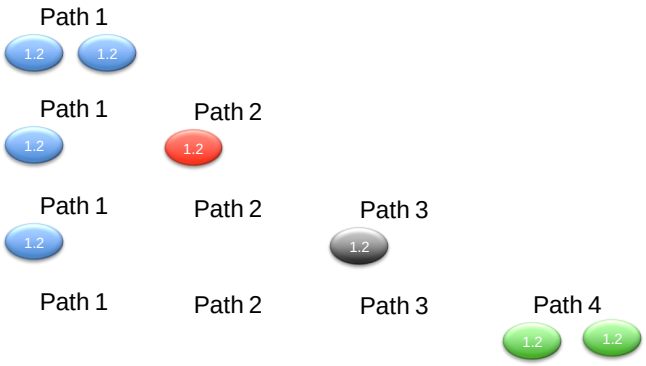
Method of Successive Averages



"Perfect" solution of MSA…
But maybe not practical… each iteration more packets…
Eventually you WILL run out of memory.

citilabs

Packet Allocation Methodology

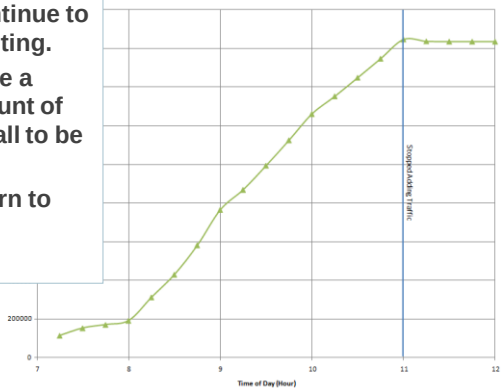


With each iteration, each packet has a MSA probability (1/Iter)
of being "Selected" for the fancy new path.
Globally the proper amount of trips are moved to the new path…
never runs out of memory or slows down.

citilabs

Loading in Congested Networks

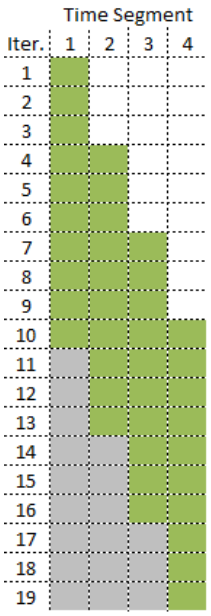
- In extremely congested networks, an "All or Nothing" first iteration can cause gridlock.
- Later time segments have no hope of 'learning' how to navigate the network when earlier time segments continue to see drastic changes to their routing.
- When later iterations finally have a stable initial condition, the amount of traffic being adjusted is too small to be meaningful.
- A time segment must wait its turn to begin being assigned.



citilabs

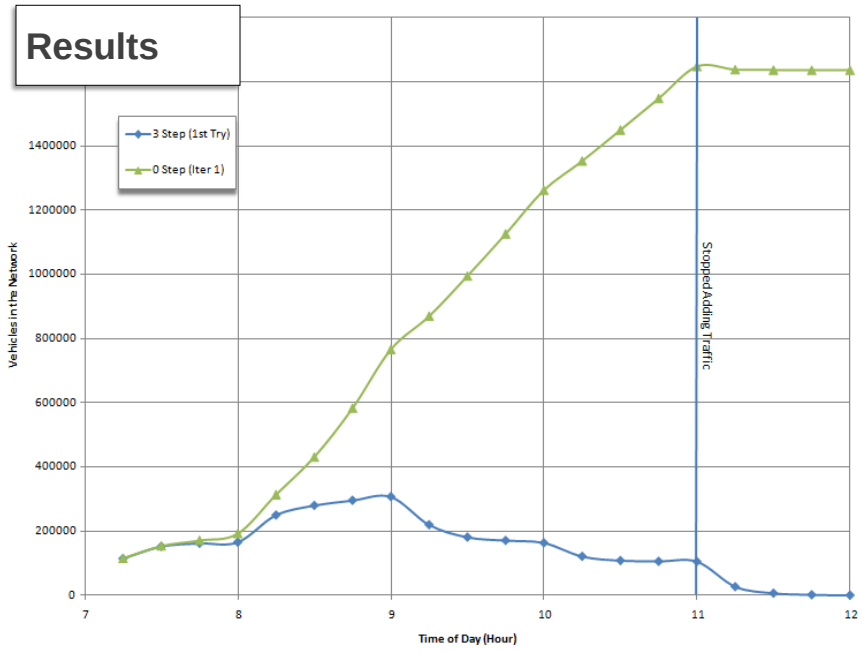
Incremental Time Segment Loading

- The user defines a number of iterations that each time segment gets to itself before the next time segment may start.
- Each time segment respects its own convergence and maximum number of iterations (maxiters)



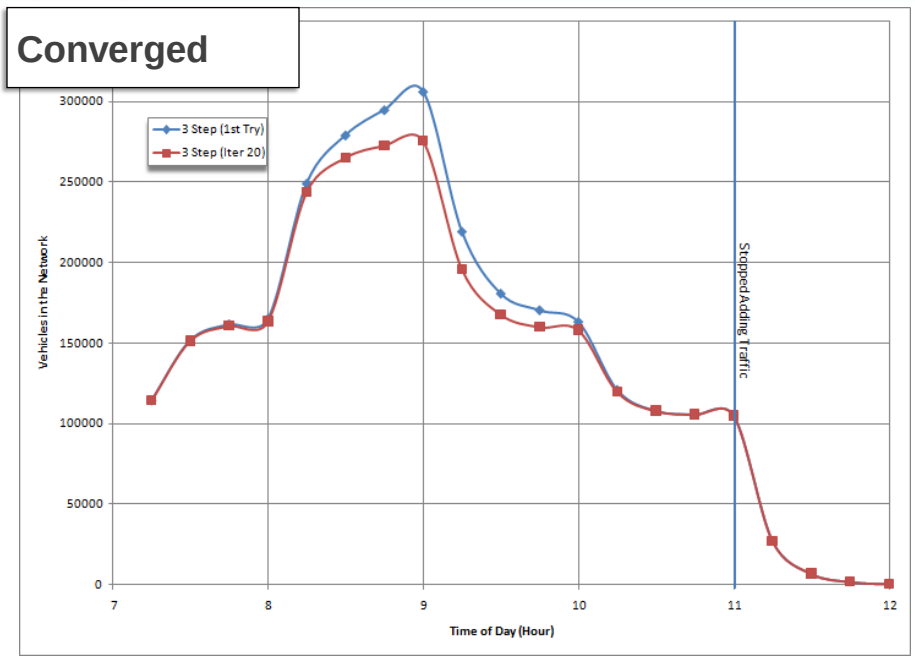
citilabs

Results



citilabs

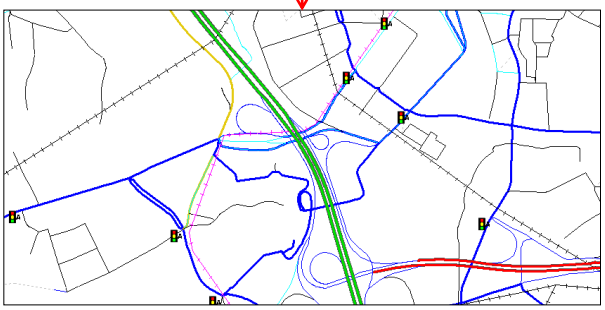
Converged



citilabs

Cube Avenue – Input Data

- Time-varying O-D travel demand (flow per time segment)
- Link properties (Capacity, T0, Storage, Junction Modelling, ...)



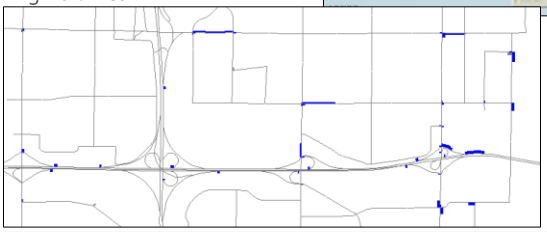
Time Segment N

	D_1	D_i	D_z
O_1	$T_{1,1}$	$T_{1,i}$	$T_{1,z}$
O_i	$T_{i,1}$	$T_{i,i}$	$T_{i,z}$
O_z	$T_{z,1}$	$T_{z,i}$	$T_{z,z}$

citilabs

Cube Avenue – Output Data

- Dynamic bandwidth:
  - Flows
  - Queues
  - Vol/Cap ratio
  - ...
- Dynamic data at junctions:
  - LOS and delays
  - Queues
  - Turning volumes
  - ...



citilabs

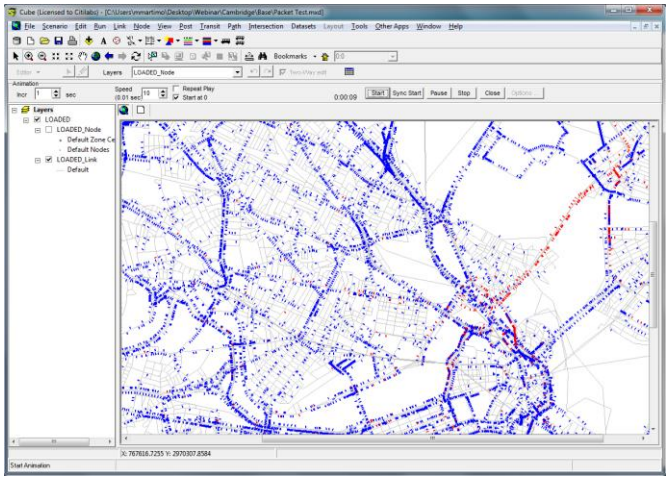
Cube Avenue – Output Data

- Dynamic bandwidth:
  - Flows
  - Queues
  - Vol/Cap ratio
  - ...
- Dynamic data at junctions:
  - LOS and delays
  - Queues
  - Turning volumes
  - ...
- Packet animation

citilabs

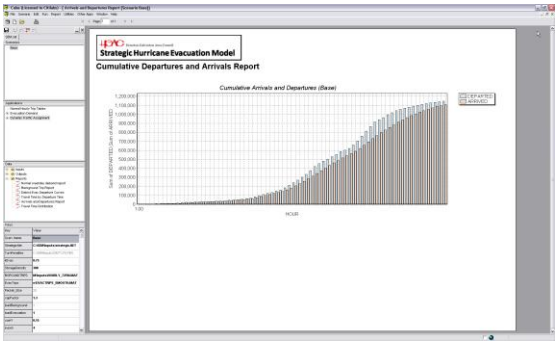
Packet Log Animations

- **Working closely with ESRI, we have now been able to add detailed animations to the GIS window.**



citilabs

Reporting Simulated Arrivals and Departures



- **Cumulative departures and arrivals by hour**
- **Gap between curves indicates travel time > 1 hour (queueing)**
- **Slope of arrivals curve implies network throughput**
- **Graph shows only evacuation trips**

citilabs

Processing the Packet Log Data

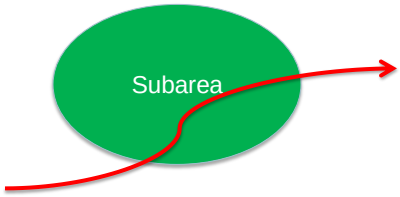
By applying record processing techniques to packet log output data, you can implement many advanced analyses with Avenue:

- **Build origin-destination table from log file**
- **Select node/link analysis**
  - Select link/node trip table: build table of trips using some node at some particular time
  - Check whether packets used a particular link/node and build a link table from the list of nodes used by these packets (2 passes of the file)
- **Extract average queue for specific packets (departure minus arrival)**
- **Temporal disaggregation (e.g. build 15-minute matrices from peak hour simulation output based upon recorded departures)**
- **Peak spreading**
  - Build packet table from log file, flag packets that failed to arrive at their destination
  - Shift packets to new departure time segment based upon logit or other decision rule
  - Re-build hourly trip matrix from packet table
- **Other applications: ITS/VMS, parking, sub-area extraction, ME, etc…**

citilabs

Hybrid Static/Dynamic Traffic Assignment

- **Example: sub-area analysis**
  - Identify trips whose paths traverse subarea and extract from table
  - Assign remainder statically, subarea trips dynamically
- **Addresses boundary problem**
  - Static subarea extraction process – Can't determine when the trip entered subarea
  - Dynamic subarea extraction process – Hybrid assignment
  - Realistic flow rates at the subarea boundary
  - Estimate of congestion from static and dynamic loads



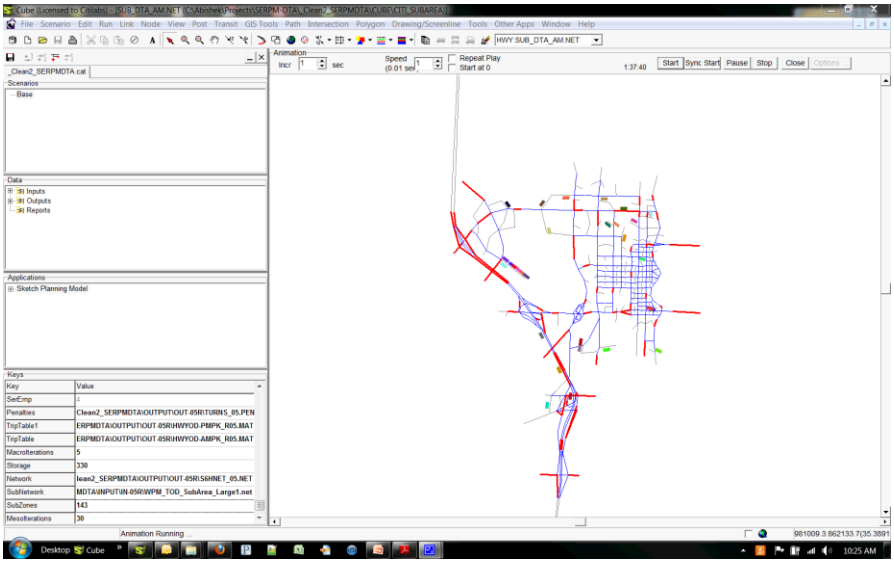
citilabs

SERPM Highway Network

Subarea Highway Network

citilabs

Animation in Cube Avenue



citilabs

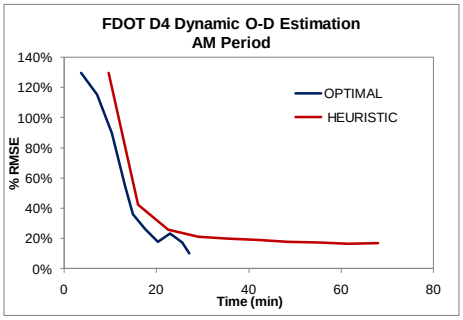
Dynamic Origin-Destination Matrix Estimation

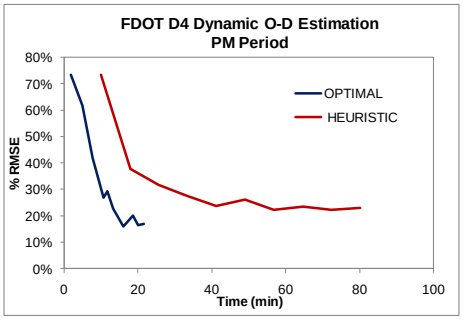
- **(Adjust) Subarea OD pattern ← Regional OD pattern**
  - Before passing through subarea simulation model
- **Cube Analyst: Limitations when working with Cube Avenue**
  - Cube Avenue builds true time-dependent shortest paths
  - Cube Analyst assumes route choice probabilities that are constant over time
  - Simulation enforces capacity constraints – Not necessarily balanced
- **Development of Dynamic OD Estimation process**
  - Mathematic Optimization method
  - Heuristic method

citilabs

Convergence Results of O-D Estimation

- RMSE (%) by run time (min) for case without Cube Analyst run

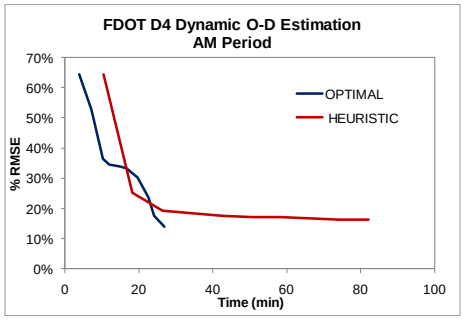


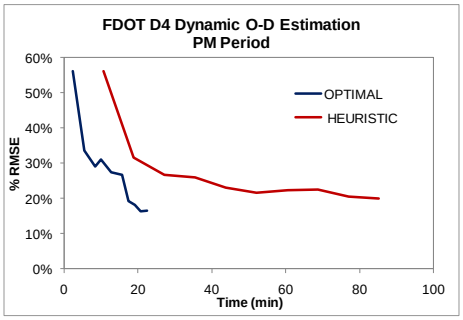


citilabs

Convergence Results of O-D Estimation

- RMSE (%) by run time (min) for case with Cube Analyst run to adjust initial O-D trips





citilabs

Defining Project Scope for Avenue

- **Statement of purpose and need: why do we need DTA?**
  - More realistic highway traffic assignment
  - Animations & visualization of results
  - Simulation of traffic queueing
  - Special events / variable demand & supply
- **What products do we need or want the model to produce in order to satisfy this purpose?**
  - Turning movement flows
  - Queues at intersections
  - Dynamic maps
  - Link speeds by 15-minute interval

citilabs

Typical Cube Avenue Model Scales

- **Bottleneck or project simulation model**
  - Typical network scale: corridor / sub-area
  - Model period = AM or PM peak; Time segments = 5-15 minute intervals
  - Goal: model queuing at specific known bottlenecks
- **Average weekday time-of-day assignment**
  - Typical network scales: metro, county or city-wide
  - Model period = 24 hours; Time segments = 15-60 minute intervals
  - Goal: improve upon standard daily weekday assignment
- **Special event or evacuation model**
  - Typical network scales: statewide, metro, or county
  - Model period = 3-72 hours; Time segments = 30-60 minute intervals
  - Goal: model a specific disaster or system shock

citilabs

Data Needs and Checks for Cube Avenue

- **Origin-Destination Trip Tables by Time Segment**
  - Check to make sure that model period matches network capacity
  - Check (or calibrate) to ensure that the temporal and spatial distribution of demand is consistent with observed patterns (traffic counts)
- **Study area roadway network**
  - Make sure distances are accurate throughout (ideally GIS-based)
  - Check coded directional number of lanes & facility type throughout
- **Intersection definitions and turn penalties**
  - While helpful, it is *not* mandatory to include junction modeling; however check junction coding thoroughly if you are using them
  - Check hidden illogical turning movements that should be banned

citilabs

Capacity Assumptions

- **When calculating network capacity and flow/speed characteristics, we need to replace our planning assumptions with methods that are traditionally reserved for a more operational level of analysis; for example:**
  - Planning models commonly include capacity scaling factors in an attempt to show peak-hour congestion patterns within a peak-period or daily assignment. These should never be used in a DTA model; instead, multiply the hourly capacity by the number of hours in the model period to obtain the effective model period capacity.
  - Always think of CAPACITY and STORAGE as extreme or upper bound values, never use average conditions. For example, STORAGE only matters once a link is full of vehicles, so calculate based upon jam density (not moving+queued vehicles).

citilabs

Congestion Functions

- **Review the TC link congestion functions; consider changes**
  - Reasonableness checks; compare to literature
  - Compare to observed data (if available)
- **It may be helpful to differentiate functions by link class or use Akcelik equations.**
- **However, don't over-rely on the queuing model to tell the routing algorithm about congestion.**
- **For links with strange geometry or weaving and merging issues, it may be worthwhile to develop special curves via LOOKUP tables based on observed speed-flow data**

citilabs

Calibration & Validation Tips

- **Keep the project scope & goals in mind!**
- **Always check at least two of the three fundamental dimensions of traffic:**
  ***speed* (time),**
  ***flow* (volume), and**
  ***density* (queue)**
- **Use visualization tools to help e.g. bandwidth animation of multiple variables changing over time and space**
- **Never vary more than one variable or input at once**
- **Keep the script simple (though the capabilities are there)**

citilabs

Helping You to Create a Better Future