Basic network loading: Combining node and link models

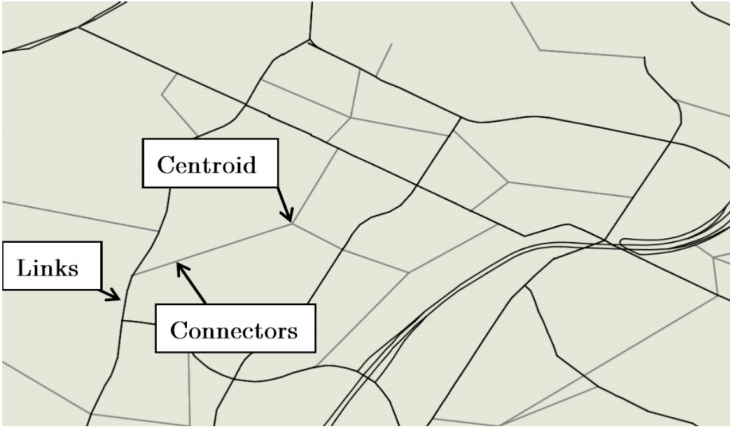
CE 392D

OUTLINE

1. Centroid connectors
2. Combining node and link models
3. Example

CENTROID CONNECTORS

We've seen node and link models that work with traffic which is already on the network... but how do vehicles enter and exit the network in the first place?



It is common to aggregate small neighborhoods into "centroids" and "centroid connectors," omitting the details of the exact origins and destinations.

This is for two reasons:

Computational: Most of the streets in a city are actually small and uncongested. The ones we care most about occupy a fairly small fraction of lane miles. We should spend most of our effort modeling the most important links.

Data: It is very hard to predict the exact start and end addresses of trips. We can do a better job predicting the aggregate flows between neighborhoods.

As an aside, centroid connector placement affects results more than is commonly known. Research is underway about how best to place centroid connectors.

In our modeling framework, we will treat centroids with distinct *node* models. Centroid connectors can be modeled with any type of link model, with suitably chosen parameters.

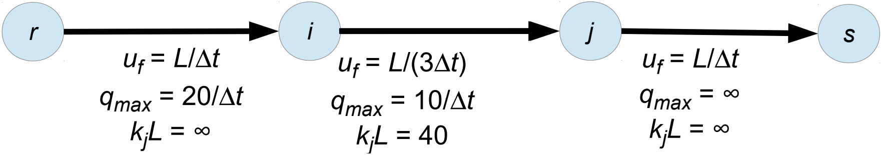
- Origin centroids: if D_r is the total demand leaving at node r during this time step, and p_{ri} is the fraction starting their trips on centroid connector (r, i), increase $N^{\uparrow}_{ri}$ by $p_{ri} D_r$ for the next time step. (Load all trips on connectors.)
- Destination centroids: if S_{is} is the sending flow from a centroid connector (i, s), increase $N^{\downarrow}_{is}$ by S_{is} for the next time step. (All the sending flow leaves.)
- Centroid connectors: infinite jam density for connectors from origins; infinite capacity for connectors to destinations. (Artificial links should not be bottlenecks.)

COMBINING NODE AND LINK MODELS

Algorithm

1. Initialize all count variables $N_{ij}^{\uparrow}(0)$ and $N_{ij}^{\downarrow}(0)$ to zero, set $t \leftarrow 0$.
2. Use link models to compute S_{ij} and R_{ij} for all links.
3. Use node models to compute transition flows y_{ijk} for all non-centroid nodes j.
4. Update cumulative counts based on y_{ijk} (set upstream/downstream counts for $t+1$)
5. Load new trips at origins (set upstream counts for origin centroid connectors)
6. Terminate trips at destinations (set downstream counts for destination connectors)
7. Increase t by 1. Stop if t equals the time horizon $\bar{T}$, otherwise return to step 2.

Example



Links use spatial queue model. Nodes *i* and *j* follow the "links in series" model, except that node *j* will force $y_{ijk} = 0$ for time steps 5, 6, 7, 8, and 9 (red light).

Example

t	d_{rs}	Link (r,i) R_{ri}	$N^{\uparrow}_{ri}$	$N^{\downarrow}_{ri}$	S_{ri}	Node i y_{rij}	Link (i,j) R_{ij}	$N^{\uparrow}_{ij}$	$N^{\downarrow}_{ij}$	S_{ij}	Node j y_{ijs}	Link (j,s) R_{js}	$N^{\uparrow}_{js}$	$N^{\downarrow}_{js}$	S_{js}
0	15	∞	0	0	0	0	20	0	0	0	0	∞	0	0	0
1	15	∞	15	0	15	15	20	0	0	0	0	∞	0	0	0
2	15	∞	30	15	15	15	20	15	0	0	0	∞	0	0	0
3	15	∞	45	30	15	10	10	30	0	10	10	∞	0	0	0
4	15	∞	60	40	20	10	10	40	10	10	10	∞	10	0	10
5	15	∞	75	50	20	10	10	50	20	10	0	∞	20	10	10
6	10	∞	90	60	20	0	0	60	20	10	0	∞	20	20	0
7	10	∞	100	60	20	0	0	60	20	10	0	∞	20	20	0
8	10	∞	110	60	20	0	0	60	20	10	0	∞	20	20	0
9	10	∞	120	60	20	0	0	60	20	10	0	∞	20	20	0
10	10	∞	130	60	20	0	0	60	20	10	10	∞	20	20	0
11	10	∞	140	60	20	10	10	60	30	10	10	∞	30	20	10
12	10	∞	150	70	20	10	10	70	40	10	10	∞	40	30	10
13	0	∞	160	80	20	10	10	80	50	10	10	∞	50	40	10
14	0	∞	160	90	20	10	10	90	60	10	10	∞	60	50	10
15	0	∞	160	100	20	10	10	100	70	10	10	∞	70	60	10
16	0	∞	160	110	20	10	10	110	80	10	10	∞	80	70	10
17	0	∞	160	120	20	10	10	120	90	10	10	∞	90	80	10
18	0	∞	160	130	20	10	10	130	100	10	10	∞	100	90	10
19	0	∞	160	140	20	10	10	140	110	10	10	∞	110	100	10
20	0	∞	160	150	10	10	10	150	120	10	10	∞	120	110	10
21	0	∞	160	160	0	0	10	160	130	10	10	∞	130	120	10
22	0	∞	160	160	0	0	20	160	140	10	10	∞	140	130	10
23	0	∞	160	160	0	0	20	160	150	10	10	∞	150	140	10
24	0	∞	160	160	0	0	20	160	160	0	0	∞	160	150	10
25	0	∞	160	160	0	0	20	160	160	0	0	∞	160	160	0