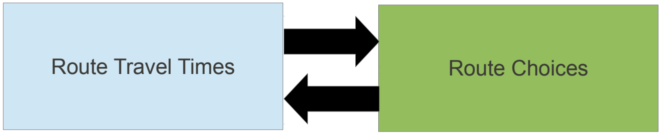
Towards equilibrium: combining network loading and behavior

CE 392D

Recall that the dynamic traffic assignment problem combines a model for traffic flow and congestion with a model for user behavior.



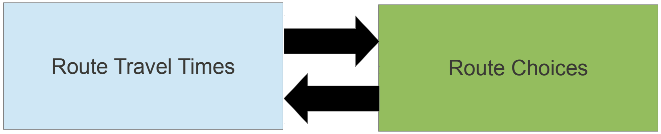
Thus far in the course we have treated each box separately. How do we reconcile them?

We need to do the following things:

1. Represent path choices and departure time choices
2. Calculate travel times from the results of dynamic network loading
3. Calculate turning fractions p from path choices.

REPRESENTING PATH CHOICES

How do we represent the path and departure time choices of individual travelers?



A few possibilities in current use

Discrete vehicles: Each vehicle is an "agent" assigned a departure time and path

Continuous path flows: Let h_t^π be the number of vehicles departing on path π during time interval t

Node turning fractions: Let $\alpha_{hij,s}^t$ be the fraction of vehicles leaving link (h, i) and heading to s that turn onto (i, j) during interval t

Each possibility has pros and cons, depending on the network loading procedure. Furthermore these can all be applied with either discrete or continuous time.

Given how we've introduced dynamic network loading and time-dependent shortest paths, "continuous path flows" are easiest to explain.

The entries h_t^π can be placed into a matrix H where the rows and columns index the paths in the network, and the possible departure times.

A matrix H is *feasible* if it satisfies these conditions:

1. $h_t^\pi >= 0$ for all π and t (non-negativity)
2. $\sum_{\pi \in \Pi_{rs}} h_t^\pi = d_t^{rs}$ for all r, s, and t (no vehicle left behind)

Denote the set of all feasible path flow matrices as $\bar{H}$.

Alternatively, when there is departure time choice, the set of all feasible path flow matrices can be written as $\hat{H}$:

1. $h_t^\pi >= 0$ for all π and t (non-negativity)
2. $\sum_{\pi \in \Pi_{rs}} h_t^\pi = d^{rs}$ for all r, s (no vehicle left behind)

Advantages:

- Using continuous variables fits with fluid-based dynamic network loading (LWR) model
- It is possible to identify which vehicles are on which links at what times (select link analysis)
- Time-dependent shortest path algorithms give a best path π which can be directly identified with a specific row in the H matrix.

Disadvantages:

- The number of paths can grow exponentially in the size of the network. (Solution: generate rows of H only when needed)
- Some extra work is needed to compute p values for dynamic network loading. (Solution: see later slides)

Alternative: discrete vehicles

Advantages:

- Behavior is easy to identify: each vehicle is assigned
- Amenable to agent-based simulations with more complex behavior rules.

Disadvantages:

- Equilibrium can be harder to find (there may be no “pure-strategy” equilibrium, think matching pennies)
- Be careful with rounding, especially with small time steps.

Also, memory requirements now scale with the number of vehicles, rather than network size. This can be either an advantage or disadvantage.

Alternative: turning proportions

Advantages:

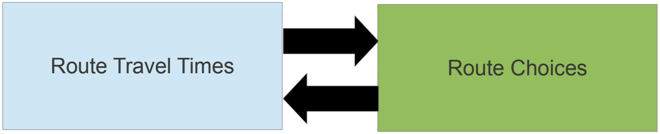
- Number of variables scales linearly with network size, rather than exponentially.
- p values are trivially calculated.

Disadvantages:

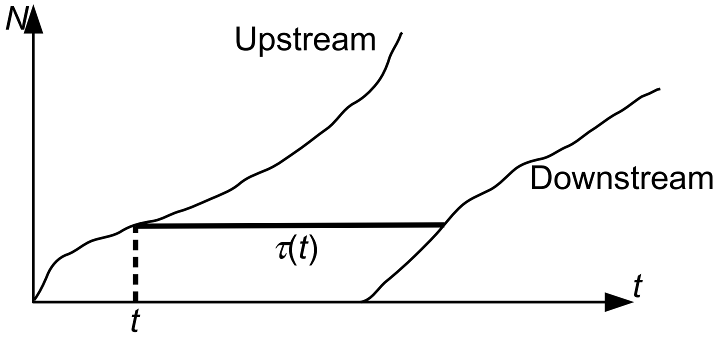
- Harder to identify the paths that specific vehicles take.
- Harder to connect with time-dependent shortest paths.

CALCULATING TRAVEL TIMES

How do we get link/path travel times for time-dependent shortest path?



After completing network loading, we need to find each link's travel time at each time interval: $\tau_{ij}(t)$



Network loading provides us with cumulative entries and exits to each link $N^{\uparrow}_{ij}(t)$ and $N^{\downarrow}_{ij}(t)$, we can use these to find travel times.

The main idea: invert the cumulative counts to obtain the entry and exit times for a particular vehicle: convert $N^{\uparrow}(t)$ and $N^{\downarrow}(t)$ to $T^{\uparrow}(n)$ and $T^{\downarrow}(n)$.

Then for a particular vehicle n, its travel time is $T^{\downarrow}(n) - T^{\uparrow}(n)$

So $\tau_{ij}(t) = T^{\downarrow}_{ij}(N^{\uparrow}_{ij}(t)) - T^{\uparrow}_{ij}(N^{\uparrow}_{ij}(t)) = T^{\downarrow}_{ij}(N^{\uparrow}_{ij}(t)) - t$

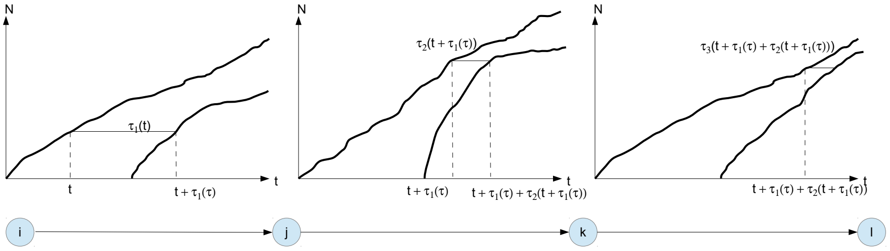
A few difficulties:

- If we are discretizing time, the values of $N^{\uparrow}(t)$ and $N^{\downarrow}(t')$ may not align. (Solution: interpolate)
- If vehicles are not entering a link at a particular time, $N^{\uparrow}(t)$ is constant and the inverse function is not well-defined. (Same with exiting vehicles.)

Introduce the tie-breaking rule $T^{\uparrow}(n) = \arg\min_t \left\{ t : N_{ij}^{\downarrow}(t) = n \right\}$ and the same for $T^{\downarrow}$. Then ensure τ_{ij} is at least equal to free-flow time:

$$\tau_{ij}(t) = \max\left\{ T_{ij}^{\downarrow}(N_{ij}^{\uparrow}(t)) - t, u_f L \right\}$$

The travel time along a path can be calculated by "stitching together" travel times on links.

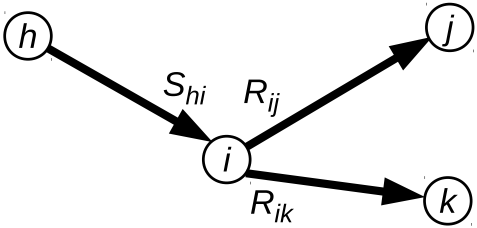


N
t
$\tau_1(t)$
t
$t + \tau_1(\tau)$
N
t
$\tau_2(t + \tau_1(\tau))$
$t + \tau_1(\tau)$
$t + \tau_1(\tau) + \tau_2(t + \tau_1(\tau))$
N
t
$\tau_3(t + \tau_1(\tau) + \tau_2(t + \tau_1(\tau)))$
$t + \tau_1(\tau) + \tau_2(t + \tau_1(\tau))$
i
j
k
l

The formula is messier than the idea, just keep this figure in mind.

CALCULATING TURNING PROPORTIONS

Route choice in networks takes place at diverges and general intersections



Thus far, we have used p_{ij} values to describe this behavior.

However, we need something more sophisicated to tie this to route choice. Drivers are leaving from different origins, and destinations, and it is very difficult to pre-specify p_{ij} values from the actual route choices.

Let Π_{rs} be the set of paths connecting origin r to destination s

We can define cumulative counts for *paths*, too. Let $N^\pi(x,t)$ be the number of vehicles *on path* π who have passed location x by time t

Then, $N(x,t) = \sum_{r,s} \sum_{\pi \in \Pi_{rs}} N^\pi(x,t)$.

Route choice can be completely specified by $N^\pi(0, t)$, where $x = 0$ refers to the origin node in the path. Why?

If the rate of demand leaving r for destination s at time t is $d_{rs}(t)$, then we need $\sum_{\pi \in \Pi_{rs}} N^\pi(0, t) = \int_0^t d_{rs}(t')\, dt'$ for all t.

(Sums instead of integrals in a discrete flow model.)

We want to do two things with these N^π values:

1. Track the flow of vehicles on each path over time
2. Determine the p_{ij} values "on the fly" *based on the route choices* $N^\pi(0, t)$

Furthermore, all of this must be done in the context of one of the flow models we've developed previously.

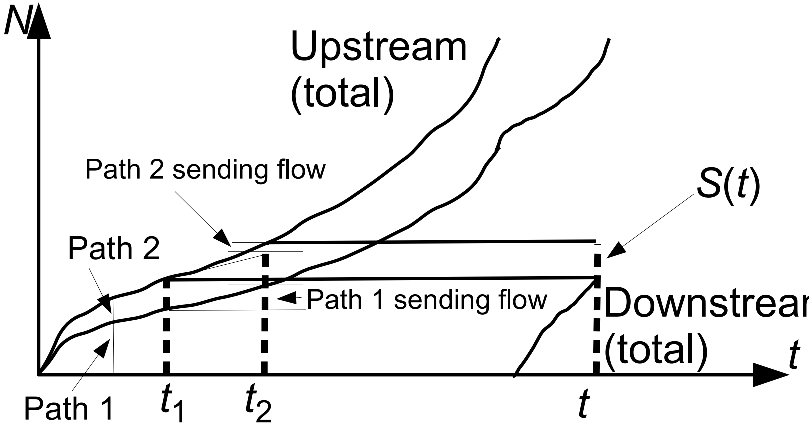
For a discrete model, $S(t)$ is the sending flow during time interval t. We want to know how much each path contributes to this sending flow: $S^\pi(t)$.

Let t_1 and t_2 to be the times at which the *first* and *last* vehicles in this sending flow entered the link.

That is, $N^\uparrow(t_1) = N^\downarrow(t)$ and $N^\uparrow(t_2) = N^\downarrow(t + S(t))$.

Interpolation is often necessary.

We now calculate $S^\pi(t) = N^\uparrow_\pi(t_2) - N^\uparrow_\pi(t_1)$.


N
Upstream (total)
Path 2 sending flow
$S(t)$
Path 2
Path 1 sending flow
Downstream (total)
t
Path 1
t_1
t_2
t

The p values are now calculated based on $S^{\pi}(t)$: we can see what proportion of the vehicles in the sending flow want to move in each direction.

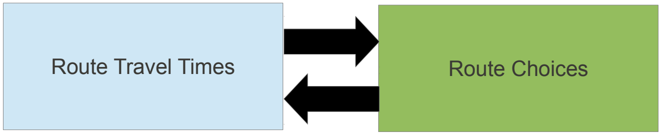
Once we calculate the transition flows $y_{ij}(t)$, there are two approaches to updating $N_{\pi}^{\uparrow}$ and $N_{\pi}^{\downarrow}$:

Method I: Recalculate t_1 and t_2 values based on the actual moving flow, not sending flow; $N_{\pi}^{\downarrow}(t+1) = N_{\pi}^{\downarrow}(t) + y_{ij}^{\pi}(t)$ for upstream link, $N_{\pi}^{\uparrow}(t+1) = N_{p}^{\uparrow}i(t)$ for downstream link.

Method II: Assume that the flows from each path are evenly distributed within the sending flow, $N_{\pi}^{\downarrow}(t+1) = N_{\pi}^{\downarrow}(t) + (y_{ij}(t)/p_{ij}S_{ij}(t))S_{ij}^{\pi}(t)$.

Method II is common in practice; if the time step is small the error introduced by this assumption is manageable.

DYNAMIC USER EQUILIBRIUM



We seek a *dynamic user equilibrium* solution which is mutually consistent: drivers choose paths and departure times such that they have no desire to change after those choices are realized.

This connects with the game theory introduced at the start of the semester.

		Bob	
		Cactus Cafe	Desert Drafthouse
Alice	Cactus Cafe	$(-1, -1)$	$(\mathbf{1}, \mathbf{1})$
	Desert Drafthouse	$(\mathbf{1}, \mathbf{1})$	$(-1, -1)$

Mathematically, we can model this as a fixed point problem or as a variational inequality.

We can write the matrix of travel times T as

$$T = \mathcal{N}(H)$$

where $\mathcal{N}$ represents the dynamic network loading of the path flow matrix H.

Given travel times T, we can express the allowable path flows as the **set** $\mathcal{B}(T)$.

(Examples: $\mathcal{B}$ places travelers only on paths with minimum travel time; on paths and departure times minimizing generalized cost; on paths within ϵ of the minimum travel tiem)

Then, the path flow matrix H solves the dynamic traffic assignment problem if

$$H \in \mathcal{B}(\mathcal{N}(H))$$

The most common behavior rule is that departure times are fixed, but travelers choose routes to minimize travel time ("dynamic user equilibrium"). Then we can express a solution to the dynamic traffic assignment problem as a matrix $H^* \in \bar{H}$ satisfying the variational inequality

$$\mathcal{N}(H^*) \cdot (H^* - H) \leq 0$$

for all other matrices $H \in \bar{H}$

We can do something similar with departure time choice as well.

In static assignment, we can use the fixed point and variational inequality formulations to prove existence of solutions, uniqueness, etc.

However, these results relied on continuity arguments that do not always hold in dynamic traffic assignment. Later in this course we will see some strange behavior that can occur as a result.