# Economics of Information Networks

**Stephen Turnbull**

Division of Policy and Planning Sciences

Lecture 2b: November 19, 2020

**Abstract**

In Part 2, we look at models of flow in networks, and the Braess Paradox.

**There is no class on November 26, 2020 due to entrance examinations.**

# Network Flow Problems

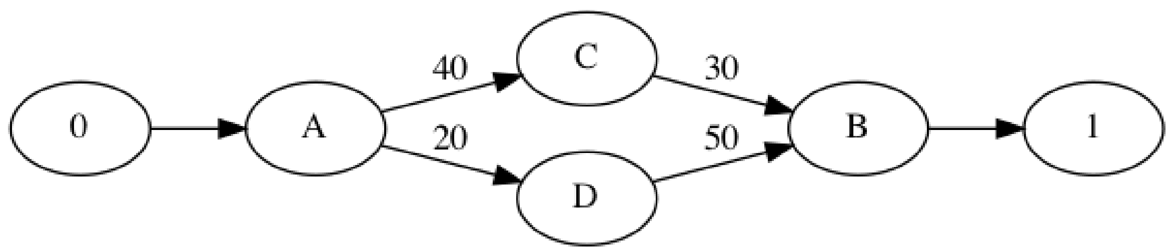
- From the pure network externality of cellphones and the Internet, we turn to a congestion externality on a network.
- Congestion is a familiar concept: too many users of a facility impose time costs (at a bridge or tollgate, for example) or degraded quality (pollution).
- First we look at the no congestion case.

# No Congestion Case

- We use a network model of a *weighted graph*, where each link has a number associated with it. The generic term is "weight," but we will interpret it as *capacity*.
  - Both *directed* and *undirected* graphs are useful. Directed graphs are used in controlled situations of production, such as an oil refinery, and for automobile traffic. Undirected graphs are used in situations such as the effect of tsunamis on river systems.
- Each link has a variable *flow* associated with it. If the graph is undirected, it may be negative. If directed, it must be positive. The capacity is the upper limit on the absolute value of the flow through the link.
- Two kinds of analysis are done:
  - Maximum flow (logistics, transportation and communication networks), where the *demand* is given.
  - Actual flow (transportation planning, flood prevention), where the *supply* is given.

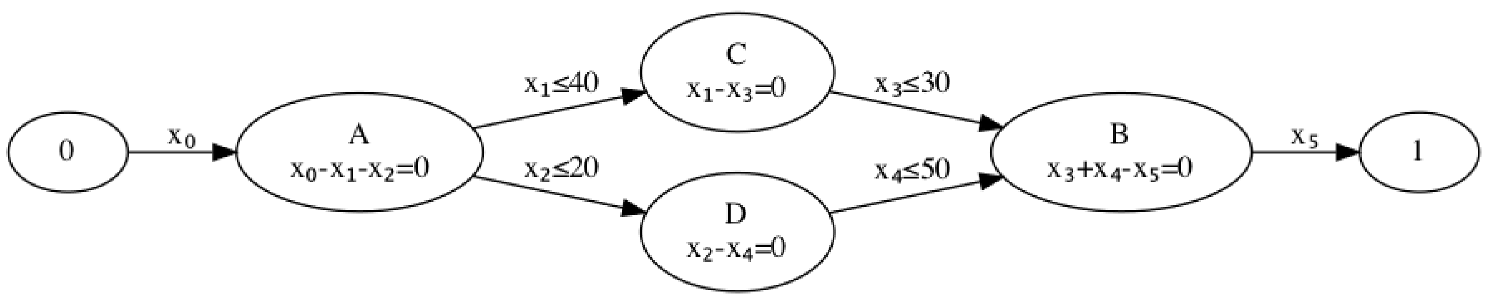
# Network Flow Example

"0" and "1" are the labels for the *source* and the *sink* (In general there could be several of each.) The link weights are capacities. The problem is to maximize flow from 0 to 1. In this simple example it's easy to see that the upper path can handle $\min\{40, 30\}$ and the lower $\min\{20, 45\}$, for a total maximum flow of 50.



# Linear Programming Solution

This optimization can be analyzed as a *linear programming problem* (LP).



# Transform the Network Flow Problem to LP

1. For each node, there is a constraint that the difference between the sum of flows *into* that node and the sum of flows *out of* that node equals zero.

2. For each *internal* link $i$, add two constraints on link flow $x_i$: $x_i \leq w_i$ and $x_i \geq 0$ (directed graph) or $x_i \geq -w_i$ (undirected graph).

3. The object is to Maximize the sum of flows into the sink, 1.

# Fundamental Theorems

Assuming the graph of the network is connected,

1. Because the number of equality constraints is not greater than the number of links (flow variables), and the variable limits are non-empty, the constraint set is always non-empty.

2. A variant of the simplex method solves the problem efficiently.

# The Braess Paradox

- We now consider a *game* on a network which leads to the paradoxical result that adding capacity to a network can cause it to have *lower* flow in equilibrium.
- This requires congestion in the links.

# Congestion

- Congestion is a familiar concept: too many users of a facility impose time costs (at a bridge or tollgate, for example) or degraded quality (pollution).
  - We typically model this as a cost to each user based on the total number of users:

    $$U_i(n) = v_i - c_i n$$

    which is very similar to the model leading to Metcalfe's law, except that the coefficient on $n$ is negative.
  - In some cases (*e.g.*, pollution) it's useful to consider the users' quantity choices as well:

    $$U_i(q) = v_i(q_i) - c_i \sum_{j=0}^{n} q_j.$$

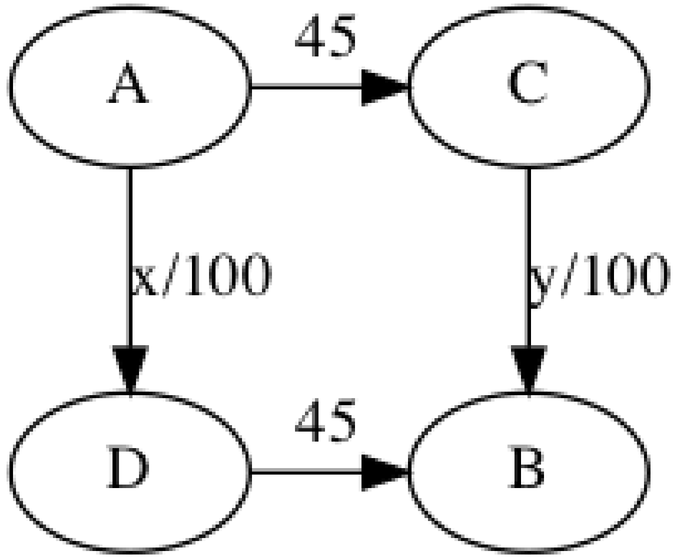
# Latency

- In the network flow problems we solved by linear programming, the capacities are given, and the "liquid" automatically flows from the source(s) into the network and out to the sink(s) at rates up to the maximum.
- We did not consider *how long* it takes to get from a source to a sink. This parameter is called *latency*.
- In a production context, we often trade latency for capacity. By waiting a while for output, we may be able to increase *throughput*.
- This will increase the rate of output once the process gets started, and this is the foundation of the economies of scale due to "continuous process production" technologies.

# The Braess Example

- In *consumption* (*e.g.*, streaming videos or commuting to school or work) we care about latency. This drives Braess's example.
- There is road network used by 4000 drivers to get from $A$ to $B$.
- There are two routes, one via $C$ and one via $D$. $A \to C$ and $D \to B$ are high-capacity highways that take 45 minutes regardless of traffic.
- $C \to B$ and $A \to D$ are bridges of small capacity such that the time to cross increases with the number of cars trying to cross the bridge. We denote the number of cars crossing $A$ to $D$ and continuing to $B$ (path ADB) as $x$, and the number starting at $A$, then crossing $C$ to arrive at $B$ (path ACB) as $y$. The bridges have the same capacity, and the time to cross is the same: $x/100$ for ADB, and $y/100$ for ACB.
  - We'll see later why we name the paths rather than the links. It's a notational convenience.

# Braess Example Network




Here's a diagram of the network.

# Nash Equilibrium

- We want each driver to choose a route (path) that minimizes her travel time.
- Then we have one constraint: $x + y = 4000$, and two travel time formulæ: $\text{time(ADB)} = 45 + x/100$ and $\text{time(ACB)} = y/100 + 45$.
  - Suppose $x > y$. Then $\text{time(ADB)} > \text{time(ACB)}$, and some of the drivers will want to change route from ADB to ACB. The opposite is true if $y > x$.
    * This is true for $x = 4000$ or $y = 4000$, no “corner” equilibria.

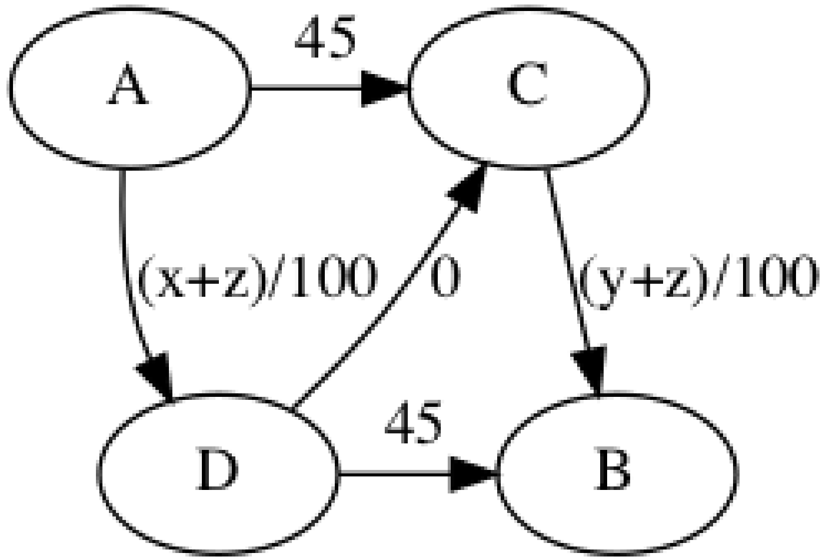
    So the equilibrium share is $x = y$.
  - Then $4000 = x + y = 2x$, so $x = y = 2000$.
  - This makes sense for society, as it has the smallest maximum travel time over all commuters. In fact everybody takes the same amount of time: $45 + 2000/100 = 65$ minutes.

# The Paradox

- Now suppose that a new bridge of very high capacity is built between D and C. It takes 0 time to cross the bridge. (Relaxing this unrealistic assumption will be a homework problem.)
  - There are three routes: A to D to B (taken by $x$ drivers), A to C to B (taken by $y$ drivers), and A to D to C to B crossing all three bridges (taken by $z$ drivers).

# The Paradox Network

Here's a diagram of the network inducing the paradox.



# Equilibrium in the Paradox

- The "everybody goes to work" constraint is $x + y + z = 4000$.
  - time(ADB) $= (x + z)/100 + 45$, time(ACB) $= (y + z)/100 + 45$, and time(ADCB) $= (x + z)/100 + (y + z)/100$.
  - Suppose $x > 0$. Then time(ADCB) $<$ time(ADB), and some of those drivers will want to switch from ADB to ADCB. This is true for *any* $x > 0$, so the only equilibrium $x$ is $x = 0$. Similarly, equilibrium $y$ is $y = 0$.
  - time(ADB) $=$ time(ACB) $= 85$ minutes, time(ADCB) $= 80$ minutes!

# Homework #8

**Due: December 3, 2020 at 11:00.** Submit to `turnbull@sk.tsukuba.ac.jp` with `Subject: Homework #8 0AL0200`.

Consider the following variants on the Braess model.

1. Consider the original network flow problem (without the new bridge), but with asymmetric capacities for the bridges: the time to cross $C \to B$ is longer: $y/80$. Is there an equilibrium? If so, what is it? If not, why not?

2. Generalize the formula for equilibrium in part 1 to the case where the time to cross $A \to D$ is $\alpha x$ and to cross $C \to B$ is $\beta y$. Ignore boundary conditions ($\alpha$ and $\beta$ are "too different" for the formula to work).

3. Discuss the boundary conditions for the formula in part 2.

4. In the three-bridge network, formulate the relevant equations when, instead of labelling paths with traffic levels, we label links with traffic levels:

$A \to D$: $x$

$C \to B$: $y$

$D \to C$: $z$ Compare this notation with the notation used in class. Which

$A \to C$: $u$

$D \to B$: $v$

is easier to solve? Which provides more information, or are they the same?

You can save yourself a lot of annoyance by choosing appropriate notation! You're also likely to make your readers (including advisors and examiners!) happier.

5. Realistically, bridges take time to cross. Suppose in the three-bridge problem it takes one minute to cross the bridge $D \to C$. How does the equilibrium change?

6. Is there a cost (in minutes) to crossing $D \to C$ large enough so that no driver will use that bridge in equilibrium? If so, what is the smallest such time? What are the equilibria for slightly smaller times?