Notes taken during class on day 5 (raw and unpolished)

Check
https://github.com/gboeing/osmnx-examples

1 Basic concepts on networks

Introduce nodes $z \in Z$ (interesctions)
and arcs $a \in A$
an arc $a = xy$ where x is the origin and y is the destination.

A network can be described by a list of arcs

origin	destination	length
29	11	3,300

The basic data of the problem will be c_a = cost of going through a.

1.1 The shortest path problem

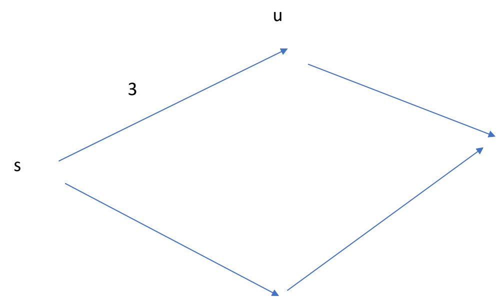
Consider a source $s \in Z$ and a target $t \in Z$
We are looking for the path with lowest cost from s to t.
A path is a sequence z_k of nodes such that $z_0 = s$, $z_K = t$, and $z_k z_{k+1} \in A$.
Thus we are looking for

$$\min_{z_k \text{ path from } s \text{ to } t} \sum_{k=0}^{K-1} c_{z_k z_{k+1}}.$$

1.2 Solution via linear programming

$$\min_{z_k \text{ path from } s \text{ to } t} \sum_{k=0}^{K-1} c_{z_k z_{k+1}}.$$

Introduce $\mu_a \in \{0,1\}$ = dummy variable for “the path goes through arc a”.
What is the condition on μ so that it is associated with a path?
$\sum_{x:xz\in A} \mu_{xz} = \sum_{y:zy\in A} \mu_{zy}$ if $z \neq s,t$
$\sum_{x:xz\in A} \mu_{xz} = \sum_{y:zy\in A} \mu_{zy} - 1$ if $z = s$
$\sum_{x:xz\in A} \mu_{xz} = \sum_{y:zy\in A} \mu_{zy} + 1$ if $z = t$
Introduce $q_z^{s,t} = 0$ if $z \neq s,t$, 1 if $z = t$ and -1 if $z = s$
and reformulate as



$\sum_{x:xz\in A}\mu_{xz}-\sum_{y:zy\in A}\mu_{zy}=q_z^{s,t}$
The problem reformulates as

$$\begin{aligned} \min_{\mu\in\{0,1\}^A} \quad & \sum_{a\in A}\mu_a c_a \\ s.t. \quad & \sum_{x:xz\in A}\mu_{xz}-\sum_{y:zy\in A}\mu_{zy}=q_z^{s,t} \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min_{\mu\geq 0} \quad & \sum_{a\in A}\mu_a c_a \\ s.t. \quad & \sum_{x:xz\in A}\mu_{xz}-\sum_{y:zy\in A}\mu_{zy}=q_z^{s,t} \end{aligned}$$

Example.
Two integral solutions

$$\mu^u=\begin{array}{ll} \text{su} & 1 \\ \text{ut} & 1 \\ \text{sd} & 0 \\ \text{dt} & 0 \end{array} \text{ and } \mu^d=\begin{array}{ll} \text{su} & 0 \\ \text{ut} & 0 \\ \text{sd} & 1 \\ \text{dt} & 1 \end{array}$$

and many real solutions

$$\theta\mu^u+(1-\theta)\,\mu^d=\begin{array}{ll} \text{su} & \theta \\ \text{ut} & \theta \\ \text{sd} & 1-\theta \\ \text{dt} & 1-\theta \end{array}$$

Reexpress

$$\begin{aligned} \min_{\mu\geq 0} \quad & \sum_{a\in A}\mu_a c_a \\ s.t. \quad & \sum_{x:xz\in A}\mu_{xz}-\sum_{y:zy\in A}\mu_{zy}=q_z^{s,t} \end{aligned}$$

in a matrix way ie

$$
\begin{array}{ll}
\min_{\mu\geq 0} & \mu^{\top}c \\
s.t. & \nabla^{\top}\mu = q_z^{s,t} \ \ [p_z]
\end{array}
$$

thus $\left(\nabla^{\top}\right)_{z,a} = \nabla_{a,z} = 1$ if arc a arrives at z, $= -1$ if it leaves z, and 0 otherwise.

The dual

$$
\begin{array}{ll}
\max_{p_z} & \sum p_z q_z^{s,t} \\
s.t. & (\nabla p)_a \leq c_a \ \ [\mu_a \geq 0]
\end{array}
$$

we have $(\nabla p)_{xy} = p_y - p_x$.

The constraint expresses $p_y - p_x \leq c_{xy}$

By complementary slackness $\mu_{xy} > 0 \implies p_y - p_x = c_{xy}$

1.3 Solution via dynamic programming

Bellman-Ford algorithm (dynamic programming).

For $k \in \{0, 1, ...\}$, let us introduce p_z^k=minimal cost of going from s to z in k steps.

$$
\begin{aligned}
p_z^0 &= 0 \text{ if } z = s \\
&= +\infty \text{ if } z \neq s
\end{aligned}
$$

$$
p_z^{k+1} = \min\left\{p_z^k, \min_{x:xz\in A}\left\{p_x^k + c_{xz}\right\}\right\}
$$

This algorithm has the form

$$
p^{k+1} = T\left(p^k\right)
$$

where

$$
T_z\left(p\right) = \min\left\{p_z, \min_{x:xz\in A}\left\{p_x + c_{xz}\right\}\right\}
$$

Properties of T:

- Order preserving: if $p \leq p'$, then $T\left(p\right) \leq T\left(p'\right)$
- p is a fixed point of T if and only if

$$
p_z = \min\left\{p_z, \min_{x:xz\in A}\left\{p_x + c_{xz}\right\}\right\}
$$

that is iff

$$p_z \leq \min_{x:xz\in A} \{p_x + c_{xz}\}$$

that is

$$\begin{aligned} p_z - p_x &\leq c_{xz} \forall x : xz \in A \\ p_0 &= 0 \end{aligned}$$

further $p_z - p_x = c_{xz}$ if xz is part of the optimal path.

ASSIGNMENT: Implement the algorithm and recover the lentgh of the path between NYU Paris and Sciences Po. Caution: take the 'walk' network, and criterion=length.

2 Nonlinear shortest path

Now the problem is to arrive at destination t as soon as possible, leaving from s at time 0.

The duration of the travel though a is assumed to depend on the departure time.

p_x is now interpreted as a TIME.

More specifically, if $a = xy$ and if passenger wants to arrive at node y at time p_y, then she needs to leave at time p_x where

$$p_x = C_{xy}(p_y)$$

and we have $p_x = C_{xy}(p_y) < p_y$. Before, we had simply

$$C_{xy}(p_y) = p_y - \frac{l_{xy}}{v_{xy}} = p_y - c_{xy}.$$

Problem. The "equilibrium flow problem" G+Samuelson+Vernet (2020).

Look for a path from s to t. $\mu_a \geq 0$ and schedule p_z such that

(1) conservation of mass holds

$$\nabla^\top \mu = q^{st}$$

(2) rationality holds: for any arc xy, one has

$$p_x \geq C_{xy}(p_y)$$

[Generalization of $p_y - p_x \leq c_{xy}$ ie. $p_y - c_{xy} \leq p_x$]

Intepretation: if we use direct arc xy, then we need to leave at $C_{xy}(p_y)$ to arrive at y at time p_y; if there is a shortcut though an intermediate arc, then we can afford to leave later]

(3) if $\mu_{xy} > 0$ then $p_x = C_{xy}(p_y)$

[In that case, there is no shortcut, we need to go through arc xy, and we need to leave at time $p_x = C_{xy}(p_y)$]

2.1 Reformulation of classical problems as EQF problem

2.1.1 Matching with TU

$$\begin{aligned}
\max_{\mu \geq 0} \quad & \sum_{xy} \mu_{xy} \Phi_{xy} \\
& \sum_{y} \mu_{xy} = n_x \\
& \sum_{x} \mu_{xy} = m_y
\end{aligned}$$

$$\begin{aligned}
\min_{u,v} \quad & \sum_{x} n_x u_x + \sum m_y v_y \\
s.t. \quad & u_x + v_y \geq \Phi_{xy}
\end{aligned}$$

Complementary slackness: μ, u, v is a solution iff
(1) μ is primal feasible

$$\begin{aligned}
-\sum_{y} \mu_{xy} = -n_x = q_x \\
\sum_{x} \mu_{xy} = m_y = q_y
\end{aligned}$$

which reformulates as $\nabla^{\top} \mu = q$ where $q = (-n, m)$ and ∇ is defined on the network.

$Z = X \cup Y$, $A = X \times Y$. [bipartite network]
(2) (u, v) is dual feasible

$$u_x + v_y \geq \Phi_{xy}$$

(3) Complementary slackness:

$$\mu_{xy} > 0 \implies u_x + v_y = \Phi_{xy}$$

Change signs: $p_x = u_x$ and $p_y = -v_y$ and we have

$$\begin{aligned}
p_y - p_x \quad & \leq \quad -\Phi_{xy} \\
p_x \quad & \geq \quad \Phi_{xy} + p_y =: C_{xy}(p_y)
\end{aligned}$$

2.1.2 Matching with ITU

$u_x + v_y \geq \Phi_{xy}$ was replaced by $D_{xy}(u_x, v_y) \geq 0$ [distance to the feasible utility set is nonnegative]. Define $p_x = u_x$ and $p_y = -v_y$, and we define $C_{xy}(p_y)$ by the implicit equation

$$D_{xy}(C_{xy}(p_y), -p_y) = 0$$

2.1.3 Hedonic models (supply and demand)

x=supply side $U_{xz}(p_z)$ = producer's utility of choosing z if price is p_z (increasing in p_z)

y=demand side $V_{zy}(p_z)$ =consumer's surplus of choosing z if price is p_z (increasing in p_z)

There are n_x producers of type x and m_y consumers of type y. Both n and m are exogenous.

Let l_z be the quantity of z produced (and consumed) at equilibrium. l_z (and p_z) is endogenous.

Consider u_x=the indirect utility of producer x and v_y=indirect surplus of consumer y.

$u_x \geq U_{xz}(p_z)$ for all x and z

$v_z \geq V_{zy}(p_z)$ for all y and z

Consider a network with nodes $X \cup Z \cup Y$ and $A = (X \times Z) \cup (Z \times Y)$.

$p_x = u_x$

$p_z = p_z$

$p_y = -v_y$

Consider $C_{xz}(.) = U_{xz}(.)$ and $C_{zy}(p_y) = V_{zy}^{-1}(-p_y)$

[indeed, $-p_y \geq V_{zy}(p_z)$ iff $V_{zy}^{-1}(-p_y) \leq p_z$]

$q_x = -n_x$, $q_y = m_y$ $q_z = 0$

2.1.4 Dynamic programming problems – already covered

Scheduling problem.

2.2 Regularized EQF problem and gross substitutes

2.2.1 The regularized problem

$$\begin{cases} \nabla^\top \mu = q \\ \mu_{xy} = \exp\left(\frac{C_{xy}(p_y) - p_x}{T}\right) \forall xy \in A \end{cases}$$

Plug in the form for μ into the first equation

$$\sum_{x:xz\in A} \exp\left(\frac{C_{xz}(p_z) - p_x}{T}\right) - \sum_{y:zy\in A} \exp\left(\frac{C_{zy}(p_y) - p_z}{T}\right) = q_z$$

which is of the form $e(p) = q$, where

$$e_z(p) = \sum_{x:xz\in A} \exp\left(\frac{C_{xz}(p_z) - p_x}{T}\right) - \sum_{y:zy\in A} \exp\left(\frac{C_{zy}(p_y) - p_z}{T}\right)$$

satisfies Gross Substitutes.

2.2.2 The gravity model in trade

Particular case: $C_{xz}(p_z) = \Phi_{xy} + p_z$ and a bipartite network, $T = 1$ then setting $A_x = e^{-p_x}$ and $B_y = e^{p_y}$, then

$$\begin{aligned} \sum_y e^{\Phi_{xy}} A_x B_y &= n_x \\ \sum_x e^{\Phi_{xy}} A_x B_y &= m_y \end{aligned}$$

which is the gravity model of international trade.

2.3 Computation of the scheduling problem using dynamic programming

Left as an exercise.