CE 272
Traffic Network Equilibrium

Lecture 20

Path Flows and Gradient Projection

Previously on Traffic Network Equilibrium...

Equilibrium solutions can be computed in terms of the link flows or the path flows.

Knowledge of either of them lets us compute link travel times using the delay functions.

The travel time on a path is simply the sum of the travel times on the links belonging to the path.

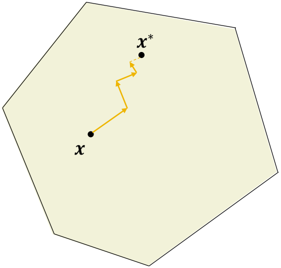
Lecture Outline

Drawbacks of Link-based Methods

Drawbacks of Link-based Methods

Introduction

Link-based methods are attractive because they require minimal storage.



However, they are prone to zig-zagging and some other issues.

Drawbacks of Link-based Methods

Steps Size Selection

The flows between all OD pairs in FW and MSA are updated using the same step size.

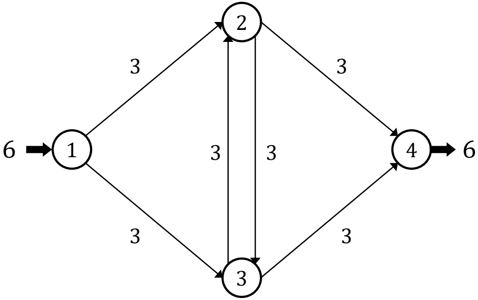
This slows convergence since the flows for some OD pairs may be closer to equilibrium than others.

Can we solve this issue by having different step sizes for different OD pairs?

Drawbacks of Link-based Methods

Cyclic Flows

Consider the following link flow solution. A feasible path flow decomposition is to have 3 travelers each on paths 1-2-3-4 and 1-3-2-4. (For e.g., this could occur in the second iteration of MSA.)



Can we have travelers on both (2,3) and (3,2) at equilibrium? Both MSA and FW will leave some residual cyclic flows. What about U-turns?

Lecture Outline

Gradient Projection

Gradient Projection
Introduction

Gradient projection is a path-based methods in which the decision variables are the path flows $\mathbf{y}$.

However, since the number of paths are exponential, we will work with a subset of paths $\hat{P}_{rs}$.

At each iteration, new paths will be added to this set if they are shortest, flows will be shifted from longer to shorter ones, and old paths will be removed if they are no longer used.

Gradient Projection

Introduction

We will first study the mathematical framework for this problem. Later, an alternate simplified version will be discussed.

For notational ease, imagine a network with a single OD pair. Extending it to the general case is trivial.

Gradient Projection

Modified Formulation

Consider the Beckmann formulation in terms of the path flows

$$\min \sum_{(i,j)\in A} \int_0^{\sum_{p\in P} \delta_{ij}^p y_p} t_{ij}(\omega)\, d\omega$$

$$\text{s.t.} \sum_{p\in P} y_p = d$$

$$y_p \geq 0 \;\forall\; p \in P$$

Gradient Projection

Modified Formulation

One option to solve the Beckmann formulation is to

1. Start with a feasible solution and take a step in the direction of the negative gradient.
2. If we reach an infeasible point, project it back to the feasible region.

Step 2 of this approach is not easy. However, if we did not have the supply demand constraints, finding the projection is a cakewalk.

Simply set all negative y_ps to zeros. How do we get rid of the supply-demand constraints? If we know the demand and the flows all the paths except one, we can get the flow on the excluded path.

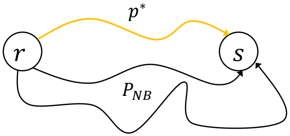
Gradient Projection

Modified Formulation

Suppose for some path flow solution **y**, let p^* be the path with the shortest travel time, which we call *basic* path. The remaining paths are called *non-basic* paths. Then,

$$y_{p^*} = d - \sum_{p \in P_{NB}} y_p$$

where P_{NB} is the set of all non-basic paths. Substituting this in the Beckmann formulation, we get a transformed objective $\hat{f}$



$$\min \sum_{(i,j) \in A} \int_0^{\delta_{ij}^{p^*}(d - \sum_{p \in P_{NB}} y_p) + \sum_{p \in P_{NB}} \delta_{ij}^{p} y_p} t_{ij}(\omega)\, d\omega$$

$$\text{s.t. } y_p \geq 0 \ \forall \ p \in P_{NB}$$

This modified formulation has one less variable and does not require explicit supply-demand constraints. (Why?)

$$\min \sum_{(i,j)\in A} \int_0^{\delta_{ij}^{p^*}\left(d-\sum_{p\in P_{NB}} y_p\right)+\sum_{p\in P_{NB}} \delta_{ij}^{p} y_p} t_{ij}(\omega)\, d\omega$$

$$\text{s.t. } y_p \geq 0 \ \forall\ p \in P_{NB}$$

Non-basic paths are expensive than p^*. Hence, their flow decreases but we have a constraint to ensure that they are always ≥ 0.

When $y_p \downarrow$ for $p \in P_{NB}$, $(d - \sum_{p\in P_{NB}} y_p) \uparrow$. Again, because of the non-negative constraints, the flow on the basic path can never exceed d.

Gradient Projection

Descent Direction

Suppose f denotes the original Beckmann function and $\hat{f}$ represents the modified objective.

$$f = \sum_{(i,j)\in A} \int_0^{\sum_{p\in P} \delta_{ij}^p y_p} t_{ij}(\omega)\, d\omega$$

$$\hat{f} = \sum_{(i,j)\in A} \int_0^{\delta_{ij}^{p^*}\left(d - \sum_{p\in P_{NB}} y_p\right) + \sum_{p\in P_{NB}} \delta_{ij}^p y_p} t_{ij}(\omega)\, d\omega$$

Recall that

$$\frac{\partial f}{\partial y_p} = \sum_{(i,j)\in A} \delta_{ij}^p t_{ij}(x_{ij}) = \tau_p$$

We will first show that for all $p \in P_{NB}$,

$$\frac{\partial \hat{f}}{\partial y_p} = \frac{\partial f}{\partial y_p} - \frac{\partial f}{\partial y_{p^*}} = \tau_p - \tau_{p^*}$$

Gradient Projection

Descent Direction

$$\hat{f} = \sum_{(i,j)\in A} \int_0^{\delta_{ij}^{p^*}\left(d - \sum_{p\in P_{NB}} y_p\right) + \sum_{p\in P_{NB}} \delta_{ij}^p y_p} t_{ij}(\omega)\, d\omega$$

$$\begin{aligned}
\frac{\partial \hat{f}}{\partial y_p} &= \sum_{(i,j)\in A} \frac{\partial}{\partial x_{ij}} \int_0^{\delta_{ij}^{p^*}\left(d - \sum_{p\in P_{NB}} y_p\right) + \sum_{p\in P_{NB}} \delta_{ij}^p y_p} t_{ij}(\omega)\, d\omega \frac{\partial x_{ij}}{\partial y_p} \\
&= \sum_{(i,j)\in A} t_{ij}(x_{ij}) \frac{\partial}{\partial y_p} \left(\delta_{ij}^{p^*} \left(d - \sum_{p\in P_{NB}} y_p \right) + \sum_{p\in P_{NB}} \delta_{ij}^p y_p \right) \\
&= \sum_{(i,j)\in A} t_{ij}(x_{ij}) \Big(-\delta_{ij}^{p^*} + \delta_{ij}^p \Big) \\
&= \tau_p - \tau_{p^*}
\end{aligned}$$

Caution: $\partial x_{ij} / \partial y_p$ is not just δ_{ij}^p because changing the flow on path p also affects the flow on the basic path p^*

To compute the step size, a quasi-Newton method is used in which the inverse of the diagonal elements of the Hessian are used to update the y values.

In other words, the updates to flows on non-basic paths are made as

$$y_p^{k+1} = y_p^k - \left(\frac{\partial^2 \hat{f}}{\partial y_p^2}\right)^{-1}\Bigg|_{y_p = y_p^k} (\tau_p^k - \tau_{p^*}^k)$$

However, this may result in negative flows, in which case we take its projection on the feasible region. That is,

$$y_p^{k+1} = \left[y_p^k - \left(\frac{\partial^2 \hat{f}}{\partial y_p^2}\right)^{-1}\Bigg|_{y_p = y_p^k} (\tau_p^k - \tau_{p^*}^k)\right]^+$$

Gradient Projection

Step Size

For a path $p \in P_{NB}$,

$$
\begin{aligned}
\frac{\partial^2 \hat{f}}{\partial y_p^2} &= \frac{\partial}{\partial y_p}(\tau_p - \tau_{p^*}) \\
&= \frac{\partial}{\partial y_p} \sum_{(i,j)\in A} \left(\delta_{ij}^p - \delta_{ij}^{p^*}\right) t_{ij}(x_{ij}) \\
&= \sum_{(i,j)\in A} \left(\delta_{ij}^p - \delta_{ij}^{p^*}\right) t'_{ij}(x_{ij}) \frac{\partial x_{ij}}{\partial y_p} \\
&= \sum_{(i,j)\in A} \left(\delta_{ij}^p - \delta_{ij}^{p^*}\right) t'_{ij}(x_{ij}) \frac{\partial}{\partial y_p} \left(\delta_{ij}^{p^*} \left(d - \sum_{p\in P_{NB}} y_p \right) + \sum_{p\in P_{NB}} \delta_{ij}^p y_p \right) \\
&= \sum_{(i,j)\in A} \left(\delta_{ij}^p - \delta_{ij}^{p^*}\right)^2 t'_{ij}(x_{ij})
\end{aligned}
$$

Gradient Projection

Step Size

Let $\hat{A}$ represent the set of links that are contained either in path p or p^* but not both. Then,

$$\begin{aligned}\frac{\partial^2 \hat{f}}{\partial y_p^2} &= \sum_{(i,j)\in A} \left(\delta_{ij}^{p} - \delta_{ij}^{p^*}\right)^2 t'_{ij}(x_{ij}) \\ &= \sum_{(i,j)\in \hat{A}} t'_{ij}(x_{ij})\end{aligned}$$

Gradient Projection

Alternate Derivation

We can derive similar expressions for the GP algorithm in a simpler, but relatively less formal way.

Instead of using the modified Beckmann function and new decision variables, simply assume that at each iteration, we identify a basic path p^* and a set of non-basic paths P_{NB}.

Let us just shift flows from all non-basic paths to basic paths to equalize their travel times.

Gradient Projection

Alternate Derivation

Suppose we shift Δy units of flow from a path $p \in P_{NB}$ to p^*.

Let $\tau_p(\Delta y)$ and $\tau_{p^*}(\Delta y)$ be the travel time on the path p and p^* **after shifting** Δy units of flow.

Define $g(\Delta y) = \tau_p(\Delta y) - \tau_{p^*}(\Delta y)$ as the difference in the travel times. The goal is to find Δy such that g is zero.

We can use an iteration of Newton-Raphson method* to find the zeros of a function with $\Delta y = 0$ as the initial solution.

$^* x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

We typically just perform one iteration and avoid finding the zero since flow shifts from other paths and OD pairs will disturb the travel times on these paths.

Hence, the amount of flow that has to be shifted is given by

$$-\frac{g(0)}{g'(0)} = -\frac{\tau_p - \tau_{p^*}}{g'(0)}$$

What is $g'(0)$? If a link does not belong to path p or p^* or if belongs to both paths, shifting flows does not impact the travel times on the link.

Gradient Projection

Alternate Derivation

$$g'(\Delta y) = \tau'_p(\Delta y) - \tau'_{p^*}(\Delta y)$$

- **Suppose a link (i, j) belongs to p and not p^***

  Recall that Δy is shifted from p to p^*. Increasing it will decrease the flow on path p and subsequently the travel time will reduce by $t'_{ij}(x_{ij})$. Hence, $g'(0)$ will contain $-t'_{ij}(x_{ij})$.

- **Suppose a link (i, j) belongs to p^* and not p**

  Increasing Δy will increase flow on p^* and increase its travel time by $t'_{ij}(x_{ij})$. But since $\tau_{p^*}(\Delta y)$ has a negative sign, $g'(0)$ will contain $-t'_{ij}(x_{ij})$.

$$\therefore g'(0) = -\sum_{(i,j)\in\hat{A}} t'_{ij}(x_{ij})$$

Gradient Projection

Alternate Derivation

From the above discussion,

$$-\frac{g(0)}{g'(0)} = \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j)\in\hat{A}} t'_{ij}(x_{ij})}$$

However, this flow shift may result in negative y_p. Hence, perform a projection step by setting

$$\Delta y = \min\left\{y_p, \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j)\in\hat{A}} t'_{ij}(x_{ij})}\right\}$$

GP(G)

Initialize $\hat{P}_{rs} \leftarrow \emptyset \, \forall \, (r, s) \in Z^2$
while Relative Gap $> 10^{-4}$ **do**
  for $r \in Z$ **do**
    DIJKSTRA (G, r)
    for $s \in Z$ **do**
      Add the shortest path p^* to $\hat{P}_{rs}$ if isn't already present
      if $\hat{P}_{rs}$ contains a single path **then**
        Set its flow to d_{rs}
      else for each non-basic path p

$$y_p \leftarrow y_p - \min \left\{ y_p, \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j) \in \hat{A}} t'_{ij}(x_{ij})} \right\}$$

$$y_{p^*} \leftarrow y_{p^*} + \min \left\{ y_p, \frac{\tau_p - \tau_{p^*}}{\sum_{(i,j) \in \hat{A}} t'_{ij}(x_{ij})} \right\}$$

      end if
    end for
    Update link flows and travel times
  end for
  Remove paths from $\hat{P}_{rs}$ that are no longer used
  Relative Gap $\leftarrow TSTT / SPTT - 1$, $k \leftarrow k + 1$
end while

How does the GP algorithm overcome the disadvantages of MSA and FW?

- Same step size for all OD pairs
- Erasing cyclic flows

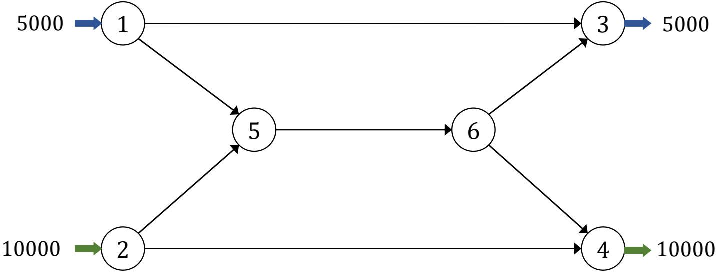
Lecture Outline

Example

Example

Network with 2 OD Pairs

Find the UE flows using GP in the following network where the delay function on each link is $10 + x/100$



Your Moment of Zen

Blame it on Frank-Wolfe?