Spring 2025

# 05 Static Traffic Assignment: Base Model I

## CIVIL-477 Transportation network modeling & analysis

École polytechnique fédérale de Lausanne

# Outline

- Formulations
  - Classic
  - VI

- Properties
  - Existence and uniqueness
  - Price of anarchy

- Solution algorithm
  - Frank-Wolfe
  - Gradient projection

# Review

- Definition of traffic assignment
  - Assign traffic flows on a given transportation network according to certain ***rules*** and satisfying certain ***constraints***

- Routing principles
  - User equilibrium (UE)
    - choose route to min ***own*** travel time, i.e., selfish routing
  - System optimum (SO)
    - choose route to min ***total*** travel time, i.e., selfless routing

- Flow constraints
  - ∑ path flow between each OD pair = OD demand
  - ∑ path flow on each link = link flow

# Review

- Network $\mathcal{G} = (N, A)$
  - Node $i \in N$
  - Origin-destination (OD) $w = (r, s) \in W,\ r, s \in N$
  - Link $a \in A$
    - Link flow $x_a \in \mathbb{R}_+,\quad a \in A$
    - Link cost $t_a \in \mathbb{R}_+,\quad a \in A$
  - Path $k \in P_w$ with $w = (r, s) \in W$
    - Path flow $f_k \in \mathbb{R}_+,\ k \in \bigcup_{w \in W} P_w$
    - Path cost $c_k \in \mathbb{R}_+, k \in \bigcup_{w \in W} P_w$
    - Min path cost $\mu_w \in \mathbb{R}_+, w \in W$
  - Link-path matrix $\Delta = \{\delta_{ap}\}_{a \in A, p \in P}$
  - OD-path matrix $\Lambda = \{\lambda_{wp}\}_{w \in W, p \in P}$

# Review

- Network $\mathcal{G} = (N, A)$
  - Node — $i \in N$
  - Origin-destination (OD) — $w = (r, s) \in W,\ r, s \in N$
  - Link — $a \in A$
    - Link flow — $x_a \in \mathbb{R}_+,\ \ a \in A$; $\mathbf{x} = (x_a)_{\forall a \in A} \in \mathbb{R}_+^{|A|}$
    - Link cost — $t_a \in \mathbb{R}_+,\ \ a \in A$; $\mathbf{t} = (t_a)_{\forall a \in A} \in \mathbb{R}_+^{|A|}$
  - Path — $k \in P_w$ with $w = (r, s) \in W$
    - Path flow — $f_k \in \mathbb{R}_+,\ k \in \cup_{w \in W} P_w$; $\mathbf{f} = (f_k)_{\forall k \in P} \in \mathbb{R}_+^{|P|}$
    - Path cost — $c_k \in \mathbb{R}_+, k \in \cup_{w \in W} P_w$; $\mathbf{c} = (c_k)_{\forall k \in P} \in \mathbb{R}_+^{|P|}$
    - Min path cost — $\mu_w \in \mathbb{R}_+, w \in W$; $\mu = (\mu_w)_{\forall w \in W} \in \mathbb{R}_+^{|W|}$
  - Link-path matrix — $\Delta = \{\delta_{ak}\}_{a \in A, k \in P}$; $\mathbf{x} = \Delta\mathbf{f}, \mathbf{t} = \Delta\mathbf{c}, \mathbf{c} = \Delta^T\mathbf{t}$
  - OD-path matrix — $\Lambda = \{\lambda_{wk}\}_{w \in W, k \in P}$; $\mathbf{q} = \Lambda\mathbf{f}$

# Review



- Traffic equilibrium condition
  - A feasible path flow $f^*$ such that $\forall w \in W, k \in P_w$,

$$f_k^*(c_k(f^*) - \mu_w^*) = 0, \qquad c_k(f^*) \geq \mu_w^*$$

- Equivalent KKT conditions

$$\begin{aligned} (\mathbf{f}^*)^T(\mathbf{c}^* - \Lambda^T\mu^*) &= 0 \\ \mathbf{c}^* - \Lambda^T\mu^* &\geq \mathbf{0} \\ \Lambda\mathbf{f}^* &= \mathbf{q} \\ \mathbf{f}^* &\geq \mathbf{0} \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \mathbf{c}^* - \Lambda^T\mu^* &= \lambda^* \\ \mathbf{f}^* &\geq \mathbf{0} \\ \Lambda\mathbf{f}^* &= \mathbf{q} \\ \lambda^* &\geq 0 \\ (\lambda^*)^T\mathbf{f}^* &= 0 \end{aligned}$$

$$\Leftrightarrow 0 = \nabla_{\mathbf{f}}\mathcal{L}(\mathbf{f}^*, \lambda^*, \mu^*) = \mathbf{c}^* - \lambda^* - \Lambda^T\mu^*$$

- ***Q: What is the corresponding optimization problem?***

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \ Z(\mathbf{f})$$
$$s.t. \ \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Lagrangian $\mathcal{L}(\mathbf{f},\lambda,\mu) = Z(\mathbf{f}) - \lambda^T\mathbf{f} - \mu^T(\Lambda\mathbf{f} - \mathbf{q})$

- KKT conditions

$$\nabla_{\mathbf{f}}\mathcal{L}(\mathbf{f},\lambda,\mu) = \nabla_{\mathbf{f}}Z(\mathbf{f}) - \lambda - \Lambda^T\mu = 0$$
$$\mathbf{f}^* \geq \mathbf{0}$$
$$\Lambda\mathbf{f}^* = \mathbf{q}$$
$$\lambda^* \geq 0$$
$$(\lambda^*)^T\mathbf{f}^* = 0$$

- ***Q: What is $Z(\mathbf{f})$ such that $\nabla_{\mathbf{f}}Z(\mathbf{f}) = \mathbf{c}$?***

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \; Z(\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Beckmann function

$$Z(\mathbf{f}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

where $$x_a = \sum_{k \in P} \delta_{ak} f_k$$

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \; Z(\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Beckmann function

$$Z(\mathbf{f}) = \sum_{a} \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

$$\frac{\partial Z(\mathbf{f})}{\partial f_k} = \sum_{a\in A} \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial f_k} = \sum_{a\in A} \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial x_a}\frac{\partial x_a}{\partial f_k} = \sum_{a\in A} t_a(x_a)\frac{\partial \sum_k \delta_{ak} f_k}{\partial f_k} = \sum_{a\in A} t_a(x_a)\,\delta_{ak} = c_k$$

  - A compact form $\nabla_{\mathbf{f}} Z(\mathbf{f}) = \mathbf{c}$

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \; Z(\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Beckmann function

$$Z(\mathbf{f}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

  - Only depend on link flows $\mathbf{x}$

$$z(\mathbf{x}) = z(\Delta\mathbf{f}) = Z(\mathbf{f})$$

  - ***Q: What are*** $\nabla_{\mathbf{x}} z(\mathbf{x})$ ***and*** $\nabla_{\mathbf{f}} z(\mathbf{x})$ ***?***

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \; Z(\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Beckmann function

$$z(\mathbf{x}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

$$\frac{\partial z(\mathbf{x})}{\partial x_a} = \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial x_a} = t_a(x_a) \qquad \Leftrightarrow \qquad \nabla_{\mathbf{x}} z(\mathbf{x}) = \mathbf{t}$$

$$\frac{\partial z(\mathbf{x})}{\partial f_k} = \sum_{a \in A} \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial x_a} \frac{\partial x_a}{\partial f_k} = \sum_{a \in A} t_a(x_a)\,\delta_{ak} = c_k \qquad \Leftrightarrow \qquad \nabla_{\mathbf{f}} z(\mathbf{x}) = \mathbf{c}$$

***Q: Another way to express this?***

# Classic formulation

- Path-based formulation

$$\min_{\mathbf{f}} \; Z(\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

- Beckmann function

$$z(\mathbf{x}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

$$\frac{\partial z(\mathbf{x})}{\partial x_a} = \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial x_a} = t_a(x_a) \qquad \Leftrightarrow \quad \nabla_{\mathbf{x}} z(\mathbf{x}) = \mathbf{t}$$

$$\frac{\partial z(\mathbf{x})}{\partial f_k} = \sum_{a \in A} \frac{\partial \int_0^{x_a} t_a(u)\,\mathrm{d}u}{\partial x_a} \frac{\partial x_a}{\partial f_k} = t_a(x_a)\delta_{ak} \qquad \Leftrightarrow \quad \nabla_{\mathbf{f}} z(\mathbf{x}) = \nabla_{\mathbf{f}} z(\Delta\mathbf{f}) = \Delta^T \nabla_{\mathbf{x}} z(\mathbf{x}) = \Delta^T \mathbf{t} = \mathbf{c}$$

# Classic formulation

- Path-based formulation

$$
\begin{aligned}
\min_{\mathbf{f}} \quad & z(\mathbf{x}) \\
s.t. \quad & \Lambda\mathbf{f} = \mathbf{q} \\
& \Delta\mathbf{f} = \mathbf{x} \\
& \mathbf{f} \geq \mathbf{0}
\end{aligned}
$$

- Compact and straightforward
- Useful for analyzing the equilibrium properties
- Feasible for solving equilibrium in small networks but not large networks
  - Number of paths increases exponentially with the network size

# Classic formulation

- Link-based formulation

$$\min_{\mathbf{x}} \quad z(\mathbf{x}) = \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(u)\,\mathrm{d}u$$

$$s.t. \quad \sum_{j\in N_i^+} x_{ij}^r - \sum_{j\in N_i^-} x_{ji}^r = q_i^r = \begin{cases} \sum_s q_{rs} & i = r \\ -q_{rs} & i = s \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in N, r \in R$$

$$\sum_{r\in R} x_{ij}^r = x_{ij}, \qquad \forall (i,j) \in A$$

$$x_{ij}^r \geq 0, \qquad \forall (i,j) \in A$$

where $N$

- $N, R$: set of nodes/origins
- $N_i^-, N_i^+$: set of upstream/downstream nodes of node $i$
- $A$: set of links

# Classic formulation

- Link-based formulation

$$\min_{\mathbf{x}} \quad z(\mathbf{x}) = \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(u)\,\mathrm{d}u$$

$$s.t. \quad \sum_{j\in N_i^+} x_{ij}^r - \sum_{j\in N_i^-} x_{ji}^r = q_i^r = \begin{cases} \sum_s q_{rs} & i = r \\ -q_{rs} & i = s \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in N, r \in R$$

$$\sum_{r\in R} x_{ij}^r = x_{ij}\,, \qquad \forall (i,j) \in A$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A$$

- Also known as Beckmann's formulation
- ***Q: Does the formulation look familiar?***

# Classic formulation

- Link-based formulation

$$\min_{\mathbf{x}} \quad z(\mathbf{x}) = \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(u)\,\mathrm{d}u$$

$$s.t. \quad \sum_{j\in N_i^+} x_{ij}^r - \sum_{j\in N_i^-} x_{ji}^r = q_i^r = \begin{cases} \sum_s q_{rs} & i = r \\ -q_{rs} & i = s \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in N, r \in R$$

$$\sum_{r\in R} x_{ij}^r = x_{ij}, \qquad \forall (i,j) \in A$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A$$

- Also known as Beckmann's formulation
- Decomposed by origin and based on node-wise flow conservation
- ***Q: How many decision variables are there?***

# Classic formulation

- Link-based formulation

$$\min_{\mathbf{x}} \quad z(\mathbf{x}) = \sum_{(i,j)\in A} \int_0^{x_{ij}} t_{ij}(u)\,\mathrm{d}u$$

$$s.t. \quad \sum_{j\in N_i^+} x_{ij}^r - \sum_{j\in N_i^-} x_{ji}^r = q_i^r = \begin{cases} \sum_s q_{rs} & i = r \\ -q_{rs} & i = s \\ 0 & \text{otherwise} \end{cases}, \qquad \forall i \in N, r \in R$$

$$\sum_{r\in R} x_{ij}^r = x_{ij}, \qquad \forall (i,j) \in A$$

$$x_{ij} \geq 0, \qquad \forall (i,j) \in A$$

- Also known as Beckmann's formulation
- Decomposed by origin and based on node-wise flow conservation
- Fewer variables than the path-based formulation $\mathcal{O}(|N|^3)$ vs $\mathcal{O}\left(2^{|N|^2}\right)$

# VI formulation

- Equivalent VI problem
  - Find $\mathbf{f}^* \in \Omega_\mathbf{f} = \{\mathbf{f} | \Lambda\mathbf{f} = \mathbf{q}, \mathbf{f} \geq 0\}$ such that

$$\langle \mathbf{c}(\mathbf{f}^*), \mathbf{f} - \mathbf{f}^* \rangle \geq 0, \qquad \forall \mathbf{f} \in \Omega_\mathbf{f}$$

  - Recall that $\mathbf{c}(\mathbf{f}^*) = \Delta^T \mathbf{t}^* = \Delta^T \mathbf{t}(\mathbf{x}^*)$, then

$$\langle \mathbf{c}(\mathbf{f}^*), \mathbf{f} - \mathbf{f}^* \rangle = \langle \Delta^T \mathbf{t}(\mathbf{x}^*), \mathbf{f} - \mathbf{f}^* \rangle = \langle \mathbf{t}(\mathbf{x}^*), \Delta(\mathbf{f} - \mathbf{f}^*) \rangle = \langle \mathbf{t}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle$$

- Link-based VI problem
  - Find $\mathbf{x}^* \in \Omega_\mathbf{x} = \{\mathbf{x} | \Delta\mathbf{f} = \mathbf{x}, \Lambda\mathbf{f} = \mathbf{q}, \mathbf{f} \geq 0\}$ such that

$$\langle \mathbf{t}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \qquad \forall \mathbf{x} \in \Omega_\mathbf{x}$$

# VI formulation

- Link-based VI problem
  - Find $\mathbf{x}^* \in \Omega_{\mathbf{x}} = \{\mathbf{x} | \Delta\mathbf{f} = \mathbf{x}, \Lambda\mathbf{f} = \mathbf{q}, \mathbf{f} \geq 0\}$ such that

$$\langle \mathbf{t}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \qquad \forall \mathbf{x} \in \Omega_{\mathbf{x}}$$

  - More compact and connected to general equilibrium in game theory
  - Often used in the case when equivalent optimization problem does not exist
  - Useful for analyzing the equilibrium properties
  - Lead to a class of solution methods
    - generalize Frank-Wolfe, Newton-type method, and gradient projection

# Questions?

# Properties of traffic equilibrium



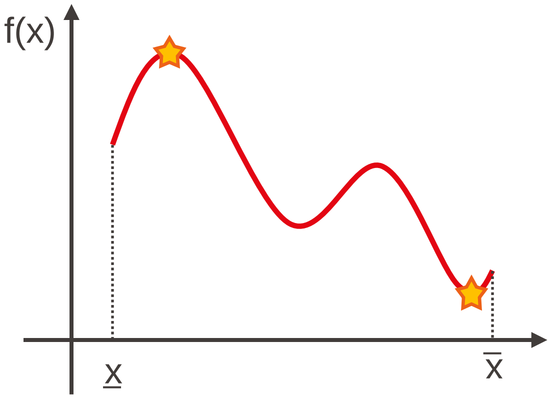
- Existence
  - With and without equivalent optimization problem

- Uniqueness
  - Link vs path flows

- Price of anarchy

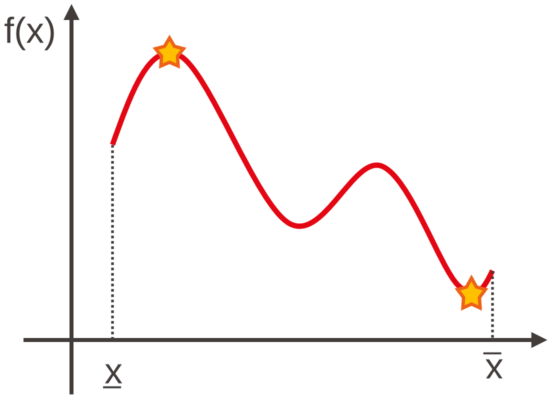
# Existence of equilibrium

- Extreme value theorem
  - Any ***continuous*** function over a ***compact*** feasible set has ***global*** minimum and maximum
    - **compact** = **close** + **bounded**
    - **close**: contain all limit points
    - **bounded**: norm is less than some finite value



# Existence of equilibrium

- Extreme value theorem
  - Any ***continuous*** function over a ***compact*** feasible set has ***global*** minimum and maximum
    - **compact** = **close** + **bounded**
    - **close**: contain all limit points
    - **bounded**: norm is less than some finite value




- Path-based formulation

$$\min_{\mathbf{f}} \; z(\Delta\mathbf{f})$$
$$s.t. \;\; \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

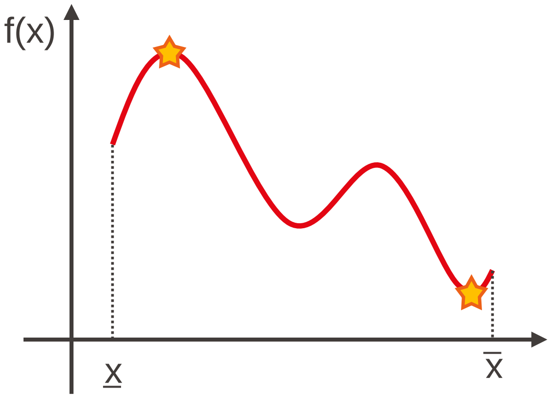
***Q: Does this problem satisfy the conditions?***

- ☐ Continuous objective function
- ☐ Compact feasible set

Note: $z(\mathbf{x})$ denotes Beckmann function

# Existence of equilibrium

- Extreme value theorem
  - Any ***continuous*** function over a ***compact*** feasible set has ***global*** minimum and maximum
    - **compact** = **close** + **bounded**
    - **close**: contain all limit points
    - **bounded**: norm is less than some finite value




- Path-based formulation

$$\min_{\mathbf{f}} \; z(\Delta \mathbf{f})$$

$$s.t. \;\; \Lambda \mathbf{f} = \mathbf{q}$$

$$\mathbf{f} \geq \mathbf{0}$$

***Q: Does this problem satisfy the conditions?***

- ☑ Continuous objective function
  - sum of continous functions (integrals)
- ☑ Compact feasible set
  - path flow $f_k \in [0, q_w], \forall k \in P_w, w \in W$

Note: $z(\mathbf{x})$ denotes Beckmann function

# Existence of equilibrium



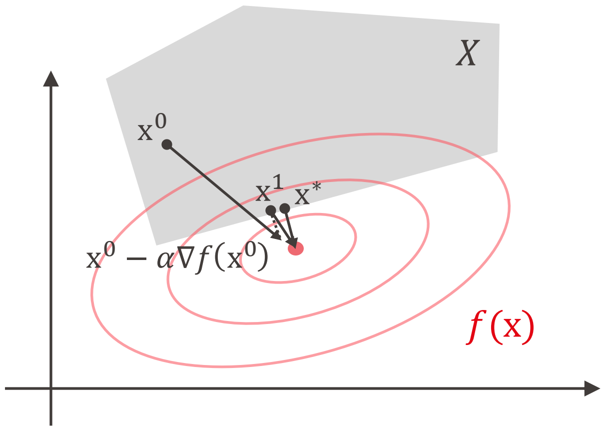
- Reformulation of VI as fixed-point problem
  - VI problem
    - Find $x^* \in X$ such that $\langle F(x^*), x - x^* \rangle \geq 0, \ \forall x \in X$
  - Fixed-point
    - Find $x^*$ such that $x^* = G(x^*)$ for some mapping $G: X \to X$

  - If $X$ is close and convex, then the equivalent fixed-point is

$$x^* = G(x^*) = \Pi_X(x^* - \alpha F(x^*))$$

where

- $\Pi_X$: projection operator
- $\alpha > 0$: any step size



# Existence of equilibrium

- Reformulation of VI as fixed-point problem
  - VI formulation
    - Find $\mathbf{x}^* \in \Omega_{\mathbf{x}} = \{\mathbf{x} | \Delta\mathbf{f} = \mathbf{x}, \Lambda\mathbf{f} = \mathbf{q}, \mathbf{f} \geq 0\}$ such that

$$\langle \mathbf{t}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \qquad \forall \mathbf{x} \in \Omega_{\mathbf{x}}$$

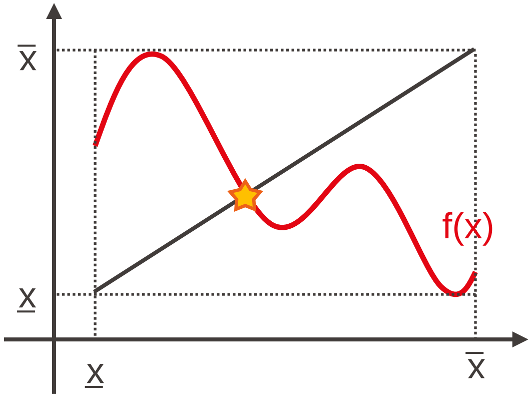
  - Fixed-point formulation
    - Find $\mathbf{x}^*$ such that

$$\mathbf{x}^* = \Pi_{\Omega_{\mathbf{x}}}(\mathbf{x}^* - \alpha \mathbf{t}(\mathbf{x}^*))$$

# Existence of equilibrium

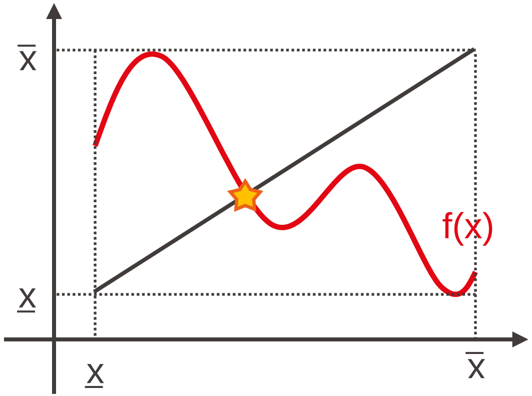


- Brouwer's fixed-point theorem
  - Any ***continuous*** function from a ***nonempty convex compact*** set ***to itself*** has a fixed point



# Existence of equilibrium

- Brouwer's fixed-point theorem
  - Any ***continuous*** function from a ***nonempty convex compact*** set ***to itself*** has a fixed point




- Fixed-point formulation
  - Find $\mathbf{x}^*$ such that $\mathbf{x}^* = \Pi_{\Omega_{\mathbf{x}}}(\mathbf{x}^* - \alpha\mathbf{t}(\mathbf{x}^*))$

- ***Q: Does this problem satisfy the conditions?***

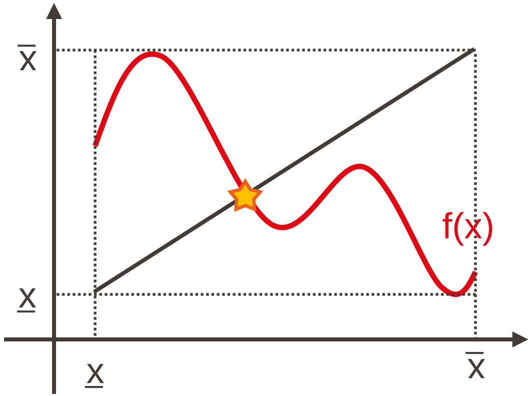
☐ Continuous operator

☐ Nonempty convex compact feasible set

# Existence of equilibrium



- Brouwer's fixed-point theorem
  - Any ***continuous*** function from a ***nonempty convex compact*** set ***to itself*** has a fixed point




- Fixed-point formulation
  - Find $\mathbf{x}^*$ such that $\mathbf{x}^* = \Pi_{\Omega_\mathbf{x}}(\mathbf{x}^* - \alpha\mathbf{t}(\mathbf{x}^*))$

- ***Q: Does this problem satisfy the conditions?***

☑ Continuous operator
- due to continuous $\mathbf{t}$ and convex $\Omega_\mathbf{x}$

☑ Nonempty convex compact feasible set
- physical meaning of $\Omega_\mathbf{x}$

# Questions?

# Uniqueness of equilibrium



- Solution uniqueness of a constrained nonlinear optimization

$$\min_{x} \quad f(x)$$
$$s.t. \quad x \leq X$$

- If the feasible set $X$ is ***convex*** and ***compact***
- If objective function $f(x)$ is ***strictly convex***

- ***Q: How to check if a twice differentiable function is convex, and strictly convex?***

# Uniqueness of equilibrium

- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Path-based formulation

$$\min_{\mathbf{f}} \quad z(\Delta\mathbf{f})$$
$$s.t. \quad \Lambda\mathbf{f} = \mathbf{q}$$
$$\mathbf{f} \geq \mathbf{0}$$

***Q: Does this problem satisfy the conditions?***

- ? ☐ Strictly convex objective function
- ✓ ☐ Convex and compact feasible set

Note: $z(\mathbf{x})$ denotes Beckmann function

# Uniqueness of equilibrium



- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Convexity of Beckmann function

$$z(\Delta\mathbf{f}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

- If link cost function $t_a$ is differentiable, then $z(\Delta\mathbf{f})$ is twice differentiable

$$\frac{\partial z(\Delta\mathbf{f})}{\partial f_k} = \sum_{a\in A} t_a(x_a)\,\delta_{ak} \Rightarrow \frac{\partial^2 z(\Delta\mathbf{f})}{\partial f_k^2} = \sum_{a\in A} t'_a(x_a)\,\delta_{ak}^2, \qquad \frac{\partial^2 Z(\mathbf{f})}{\partial f_k \partial f_{k'}} = \sum_{a\in A} t'_a(x_a)\,\delta_{ak}\delta_{ak'}$$

$$\nabla_{\mathbf{f}} z(\Delta\mathbf{f}) = \Delta^T \nabla_{\mathbf{x}} z(\mathbf{x}) \qquad \Rightarrow \nabla_{\mathbf{f}}^2 z(\Delta\mathbf{f}) = \nabla_{\mathbf{f}}\big(\nabla_{\mathbf{x}} z(\mathbf{x})\big)\Delta = \Delta^T \nabla_{\mathbf{x}}^2 z(\mathbf{x})\Delta$$

# Uniqueness of equilibrium



- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Convexity of Beckmann function

$$z(\Delta\mathbf{f}) = \sum_a \int_0^{x_a} t_a(u)\, \mathrm{d}u$$

- If link cost function $t_a$ is strictly increasing, then $t_a' > 0$
  - $\nabla_{\mathbf{x}}^2 z(\mathbf{x})$ is a diagonal matrix with all positive diagonal elements
  - $\nabla_{\mathbf{x}}^2 z(\mathbf{x}) \succ 0$ is positive definite

- ***Q: Does it imply $z(\Delta\mathrm{f})$ is strictly convex?***

# Uniqueness of equilibrium

- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Convexity of Beckmann function

$$z(\Delta\mathbf{f}) = \sum_a \int_0^{x_a} t_a(u)\,\mathrm{d}u$$

- If link cost function $t_a$ is differentiable and strictly increasing, then
  - $z(\mathbf{x})$ is strictly convex with $\mathbf{x}$ because $H_{\mathbf{x}} = \nabla_{\mathbf{x}}^2 z(\mathbf{x}) \succ 0$
  - $z(\Delta\mathbf{f})$ is only convex function with $\mathbf{f}$ because $H_{\mathbf{f}} = \nabla_{\mathbf{f}}^2 z(\Delta\mathbf{f}) = \Delta^T \nabla_{\mathbf{x}}^2 z(\mathbf{x}) \Delta \succcurlyeq 0$

# Uniqueness of equilibrium

- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \; f(\mathrm{x})$$
$$s.t. \;\; \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Uniqueness of traffic equilibrium
  - Suppose link cost function $t_a$ is differentiable and strictly increasing

$$\min_{\mathbf{x}} \; z(\mathbf{x})$$
$$s.t. \;\; \mathbf{x} \in \Omega_{\mathbf{x}}$$

***Q: Does this problem satisfy the conditions?***

- ☑ Strictly convex objective function
- ☑ Convex and compact feasible set
  - $\mathbf{x} = \Delta\mathbf{f}$ linear combination of convex set is also convex

# Uniqueness of equilibrium

- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

- Uniqueness of traffic equilibrium
  - Suppose link cost function $t_a$ is differentiable and strictly increasing

$$\min_{\mathbf{f}} \quad z(\Delta\mathbf{f})$$
$$s.t. \quad \mathbf{f} \in \Omega_{\mathbf{f}}$$

***Q: Does this problem satisfy the conditions?***

- ☐ ✗ Strictly convex objective function
- ☐ ✓ Convex and compact feasible set

# Uniqueness of equilibrium

- Solution uniqueness of a constrained nonlinear optimization

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$

- If objective function $f(\mathrm{x})$ is ***strictly convex***
- If the feasible set $X$ is ***convex*** and ***compact***

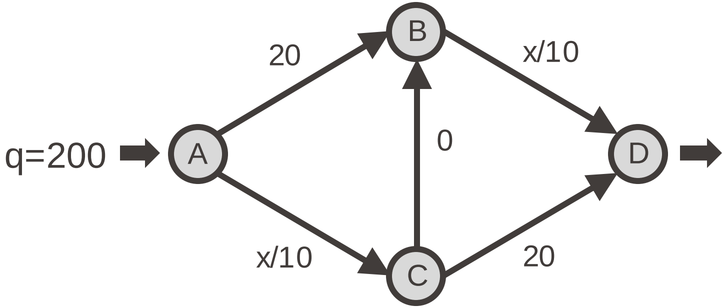
- Uniqueness of traffic equilibrium
  - Suppose link cost function $t_a$ is differentiable and strictly increasing
  - There exists unique UE link flows $\mathbf{x}^*$
  - Any path flows $\mathbf{f}^*$ that satisfies $\mathbf{x}^* = \Delta\mathbf{f}^*$ is UE path flows

# Questions?

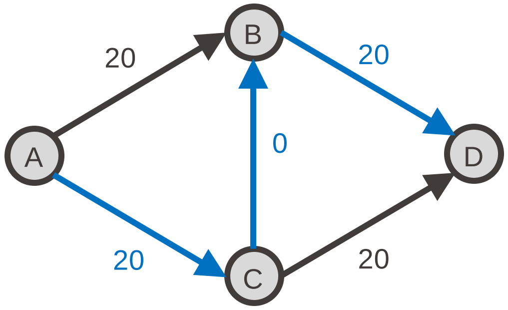
# Price of anarchy (PoA)

- Example of Braess paradox

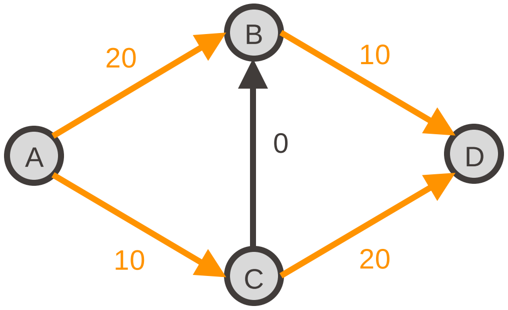
$$PoA = \frac{TT_{\mathrm{UE}}}{TT_{\mathrm{SO}}} = \frac{4}{3}$$




$TT_{\mathrm{UE}} = 40 * 200$



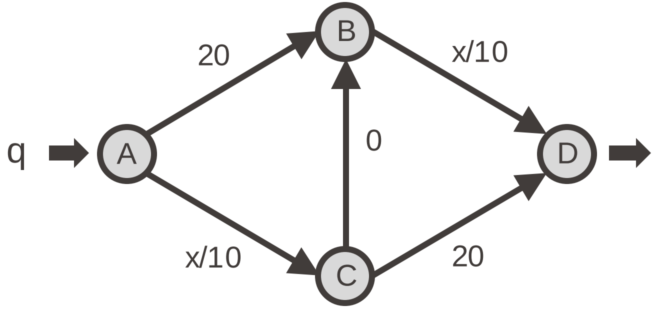

$TT_{\mathrm{SO}} = 30 * 200$




- ***Q: What are main factors of PoA?***

# Price of anarchy (PoA)

- Main factors
  - Demand
    - q=200, PoA = 4/3
    - q=500, PoA = 1
  - Network topology
    - Remove link C→B
  - Link cost function
    - Linear function t(x) = x/10, PoA = 4/3
    - Quadratic function t(x) = $x^2$ + x/10, PoA = 1



# Price of anarchy (PoA)

- Upper bound of PoA
  - Seminal work by Prof. Tim Roughgarden

**How Bad Is Selfish Routing?**

TIM ROUGHGARDEN AND ÉVA TARDOS

*Cornell University, Ithaca, New York*

Abstract. We consider the problem of routing traffic to optimize the performance of a congested network. We are given a network, a rate of traffic between each pair of nodes, and a latency function for each edge specifying the time needed to traverse the edge given its congestion; the objective is to route traffic such that the sum of all travel times—the total latency—is minimized.

In many settings, it may be expensive or impossible to regulate network traffic so as to implement an optimal assignment of routes. In the absence of regulation by some central authority, we assume that each network user routes its traffic on the minimum-latency path available to it, given the network congestion caused by the other users. In general such a “selfishly motivated” assignment of traffic to paths will not minimize the total latency; hence, this lack of regulation carries the cost of decreased network performance.

In this article, we quantify the degradation in network performance due to unregulated traffic. We prove that if the latency of each edge is a linear function of its congestion, then the total latency of the routes chosen by selfish network users is at most 4/3 times the minimum possible total latency (subject to the condition that all traffic must be routed). We also consider the more general setting in which edge latency functions are assumed only to be continuous and nondecreasing in the edge congestion. Here, the total latency of the routes chosen by unregulated selfish network users may be arbitrarily larger than the minimum possible total latency; however, we prove that it is no more than the total latency incurred by optimally routing *twice* as much traffic.

Categories and Subject Descriptors: F.0 [**Theory of Computation**]: General

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Braess’s Paradox, Nash equilibria, network flow, selfish routing

Tim Roughgarden

FOLLOW

Columbia University

Verified email at cs.columbia.edu - Homepage

Algorithms Game Theory Networks

| TITLE | CITED BY | YEAR |
|---|---|---|
| Algorithmic game theory<br>T Roughgarden<br>Communications of the ACM 53 (7), 78-86 | 5666 | 2010 |
| How bad is selfish routing?<br>T Roughgarden, É Tardos<br>Journal of the ACM (JACM) 49 (2), 236-259 | 2606 | 2002 |
| Selfish routing and the price of anarchy<br>T Roughgarden<br>MIT press | 1277 | 2005 |
| The price of stability for network design with fair cost allocation<br>E Anshelevich, A Dasgupta, J Kleinberg, É Tardos, T Wexler, ...<br>SIAM Journal on Computing 38 (4), 1602-1623 | 1201 | 2008 |
| The price of anarchy is independent of the network topology<br>T Roughgarden<br>Proceedings of the thiry-fourth annual ACM symposium on Theory of computing ... | 732 | 2002 |

# Price of anarchy (PoA)

- Upper bound of PoA
  - For linear, non-negative, and non-decreasing link cost functions, the PoA is at most 4/3
  - Proof. based on VI

At UE, we have

$$\langle \mathbf{t}(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^* \rangle \geq 0, \qquad \forall \mathbf{x} \in \Omega_{\mathbf{x}}$$

Then, for any feasible link flow $\mathbf{x} \in \Omega_{\mathbf{x}}$, we have

$$TT_{UE} \leq \mathbf{t}(\mathbf{x}^*)^T \mathbf{x} = \mathbf{t}(\mathbf{x}^*)^T \mathbf{x} - \mathbf{t}(\mathbf{x})^T \mathbf{x} + \mathbf{t}(\mathbf{x})^T \mathbf{x}$$

$$= \left(\mathbf{t}(\mathbf{x}^*) - \mathbf{t}(\mathbf{x})\right)^T \mathbf{x} + \mathbf{t}(\mathbf{x})^T \mathbf{x}$$

With linear cost $t_a(x_a) = \beta_{0,a} + \beta_{1,a} x_a$ with $\beta_{0,a}, \beta_{1,a} \geq 0$, the first term is further expanded as

$$\left(\mathbf{t}(\mathbf{x}^*) - \mathbf{t}(\mathbf{x})\right)^T \mathbf{x} = \sum_a x_a(\beta_{0,a} + \beta_{1,a} x_a^*) - \sum_a x_a\left(\beta_{0,a} + \beta_{1,a} x_a\right) = \sum_a \beta_{1,a}(x_a x_a^* - x_a^2)$$

$$\leq \sum_a \beta_{1,a} \frac{1}{4} (x_a^*)^2 \leq \frac{1}{4} \sum_a x_a^* \left(\beta_{0,a} + \beta_{1,a} x_a^*\right) = \frac{1}{4} TT_{UE}$$

Combining the above results, we conclude SO flows $\mathbf{x}^{SO} \in \Omega_{\mathbf{x}}$

$$TT_{UE} \leq \frac{1}{4} TT_{UE} + \mathbf{t}\left(\mathbf{x}^{SO}\right)^T \mathbf{x}^{SO} \Rightarrow TT_{SO} = \mathbf{t}\left(\mathbf{x}^{SO}\right)^T \mathbf{x}^{SO} \geq \frac{3}{4} TT_{UE} \;\Rightarrow\; PoA = \frac{TT_{UE}}{TT_{SO}} \leq \frac{4}{3}$$

# Questions?

# Solution algorithms

- Link-based
  - Method of successive average (MSA)
  - ***Frank-Wolfe***
- Path-based
  - ***Gradient projection***
  - Maximum entropy (next lecture)
- Bush-based
  - Algorithm B (beyond this course)

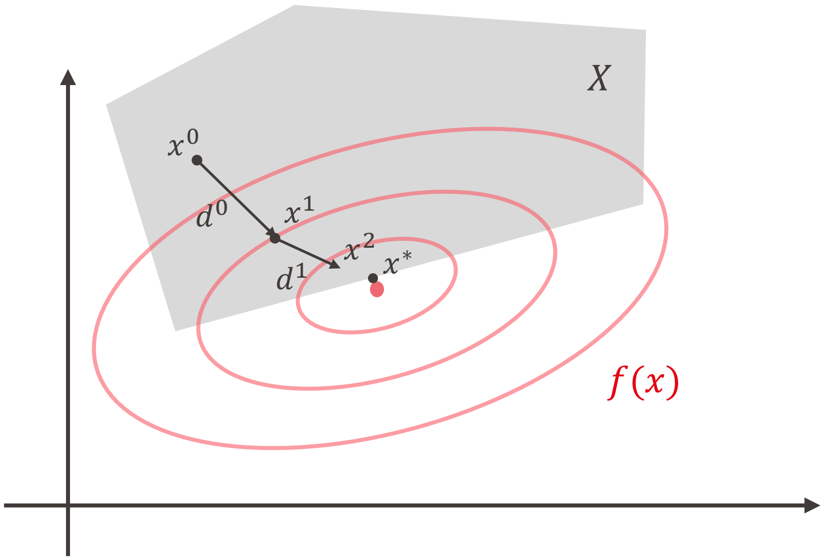
# Frank-Wolfe algorithm

- Typical idea of solving a constrained optimization problem
  - At each iteration $k$, find a good descent direction $d^k$ to update

$$x^{k+1} \leftarrow x^k + \alpha d^k$$

with some step size $\alpha$

$$\min_{\mathrm{x}} \quad f(\mathrm{x})$$
$$s.t. \quad \mathrm{x} \leq X$$




- Key questions
  - How to determine direction $d^k$?
  - How to set step $\alpha$?

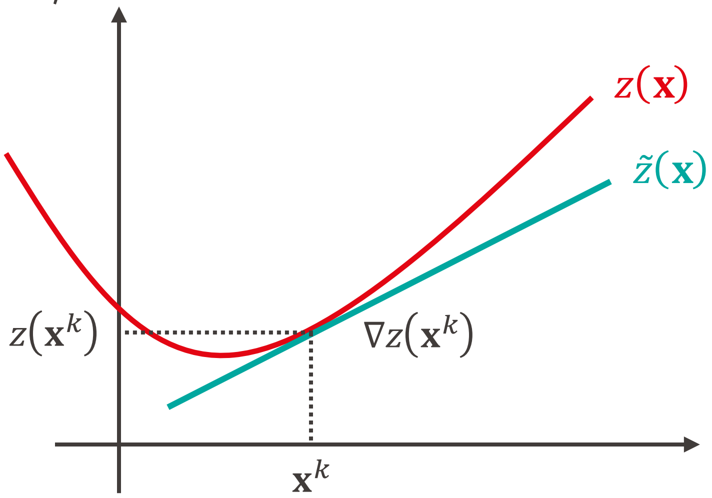
# Frank-Wolfe algorithm

- Main idea of FW algorithm
  - How to determine direction $d^k$?
    - construct an auxiliary problem that is easier to solve, and use its optimal solution $y^k$ to set $d^k = y^k - x^k$
  - How to set step $\alpha$?
    - search the optimal step size $\alpha$ that leads to the min objective

# Frank-Wolfe algorithm

- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective

$$\min_{\mathbf{x}} \ \tilde{z}(\mathbf{x}) = z(\mathbf{x}^k) + \langle \nabla z(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k \rangle$$
$$s.t. \quad \mathbf{x} \in \Omega_{\mathbf{x}}$$




which is equivalent to

$$\min_{\mathbf{x}} \ \langle \mathbf{t}^k, \mathbf{x} - \mathbf{x}^k \rangle$$
$$s.t. \quad \mathbf{x} \in \Omega_{\mathbf{x}}$$

# Frank-Wolfe algorithm

- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective

$$\min_{\mathbf{x}} \ \langle \mathbf{t}^k, \mathbf{x} - \mathbf{x}^k \rangle \quad \Rightarrow \quad \min_{\mathbf{f}} \ \langle \mathbf{c}^k, \mathbf{f} \rangle$$

$$s.t. \ \ \mathbf{x} \in \Omega_{\mathbf{x}} \qquad \qquad s.t. \ \ \mathbf{f} \in \Omega_{\mathbf{f}}$$

  - ***Q: What is the physical meaning of this problem?***

# Frank-Wolfe algorithm

- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective
      - **All-or-nothing assignment:** assign all demand to shortest paths to get $\mathbf{y}^k$
  - Step 2: Find optimal step size $\alpha$
    - Solve line search problem

$$\min_{\alpha} \quad \tilde{z}(\alpha) = z\left(\mathbf{x}^k + \alpha \mathbf{d}^k\right)$$

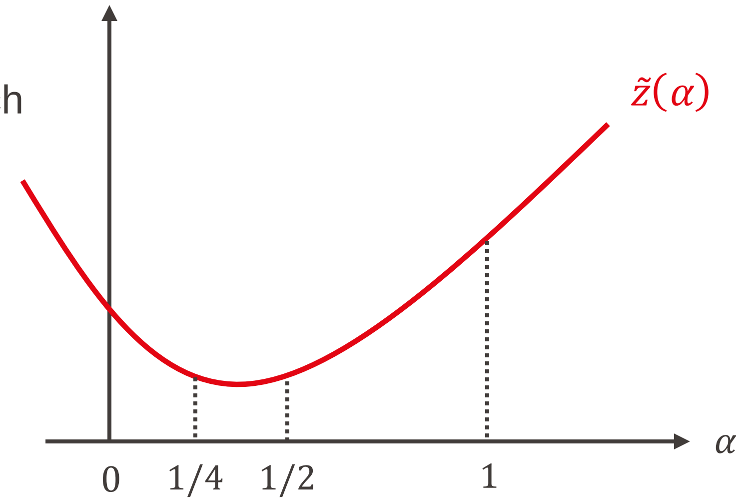
$$s.t. \quad 0 \leq \alpha \leq 1$$

  - ***Q: How to efficiently solve this problem?***

# Frank-Wolfe algorithm



- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective
      - **All-or-nothing assignment:** assign all demand to shortest paths to get $\mathbf{y}^k$
  - Step 2: Find optimal step size $\alpha$
    - Solve line search problem
      - **Bisection search:** reduce half of search space per iteration
    - Terminate when both conditions hold
      - reach max iteration or search space
      - $\nabla_\alpha \tilde{z}(\alpha) = \langle \nabla_{\mathbf{x}} \tilde{z}(\alpha), \mathbf{d}^k \rangle = \langle \mathbf{t}(\alpha), \mathbf{d}^k \rangle \leq 0$



# Frank-Wolfe algorithm

- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective
      - **All-or-nothing assignment:** assign all demand to shortest paths to get $\mathbf{y}^k$
  - Step 2: Find optimal step size $\alpha$
    - Solve line search problem
      - **Bisection search:** reduce half of search space per iteration
  - Step 3: Update link flow $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\mathbf{d}^k$
  - Step 4: Check convergence
    - Compute gap $g = \langle \mathbf{t}^k, -\mathbf{d}^k \rangle$
    - If $g \leq \varepsilon$ for some gap threshold $\varepsilon$, terminate and return $\mathbf{x}^* = \mathbf{x}^k$

# Quick note on MSA

- At iteration $k$,
  - Step 1: Find descent direction $\mathbf{d}^k = \mathbf{y}^k - \mathbf{x}^k$
    - Solve auxiliary problem with linear approximated objective
      - **All-or-nothing assignment:** assign all demand to shortest paths to get $\mathbf{y}^k$
  - Step 2: Simple step size $\alpha = 1/k$
  - Step 3: Update link flow $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha \mathbf{d}^k$
    - Exponential average $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha\left(\mathbf{y}^k - \mathbf{x}^k\right) = (1-\alpha)\mathbf{x}^k + \alpha \mathbf{y}^k$
  - Step 4: Check convergence
    - Compute gap $g = \langle \mathbf{t}^k, -\mathbf{d}^k \rangle$
    - If $g \leq \varepsilon$ for some gap threshold $\varepsilon$, terminate and return $\mathbf{x}^* = \mathbf{x}^k$

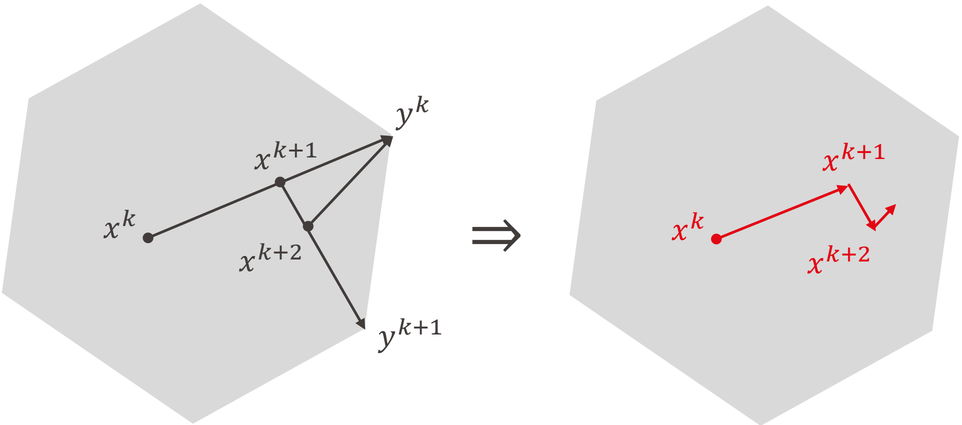
# Questions?

# Gradient projection

- Issue of FW algorithm
  - All-or-nothing assignment solve a linear program
    - Optimal solution is always at the corner

$$\min_{\mathbf{f}} \quad \langle \mathbf{c}^k, \mathbf{f} \rangle$$

$$s.t. \quad \mathbf{f} \in \Omega_{\mathbf{f}}$$




- ***Lead to to zig-zagging behaviors and thus slow convergence***

# Gradient projection

- Fixed-point formulation for path-based VI
  - Find $\mathbf{f}^* \in \Omega_{\mathbf{f}}$ such that $\langle \mathbf{c}(\mathbf{f}^*), \mathbf{f} - \mathbf{f}^* \rangle \geq 0,\ \forall \mathbf{f} \in \Omega_{\boldsymbol{f}}$
  - Find $\mathbf{f}^*$ such that $\mathbf{f}^* = \Pi_{\Omega_{\mathbf{f}}}(\mathbf{f}^* - \alpha \mathbf{c}(\mathbf{f}^*))$

- A simple projection algorithm
  - At each iteration $n$,
    - Step 1: Find descent direction $\mathbf{d}^n = -\mathbf{c}(\mathbf{f}^n) = -\nabla Z(\mathbf{f}^n)$
    - Step 2: Compute candidate path flow $\mathbf{y}^n = \mathbf{f}^n + \alpha \mathbf{d}^n$
    - Step 3: Project back to feasible set $\mathbf{f}^{n+1} = \arg\min_{\mathbf{f} \in \Omega_{\mathbf{f}}} \left|\left|\mathbf{f} - \mathbf{y}^n\right|\right|^2$
    - Step 4: Convergence check
      - Terminate if $\mathbf{f}^{n+1} = \mathbf{f}^n$ and return $\mathbf{f}^* = \mathbf{f}^n$
  - ***Q: Which step is the most challenging?***

# Gradient projection

- Main idea of GP algorithm
  - Make the projection more efficient by modifying the subproblem for each OD

$$\min_{\mathbf{f}} \; Z(\mathbf{f}) = \sum_a \int_0^{\sum_k \delta_{ak} f_k} t_a(u)\,\mathrm{d}u$$
$$s.t. \;\; \Lambda_w \mathbf{f} = q_w$$
$$\mathbf{f} \geq \mathbf{0}$$

  - ***Q: How to represent the constraints by $f_k$?***

# Gradient projection



- Main idea of GP algorithm
  - Make the projection more efficient by modifying the subproblem for each OD

$$\min_{\mathbf{f}} \; Z(\mathbf{f}) = \sum_a \int_0^{\sum_k \delta_{ak} f_k} t_a(u)\, \mathrm{d}u$$

$$s.t. \; \sum_k f_k = q$$

$$f_k \geq 0$$

- OD index $w$ is dropped for simplicity

- ***Q: How to further simplify the constraints?***

# Gradient projection

- Main idea of GP algorithm
  - Make the projection more efficient by modifying the subproblem for each OD

$$\min_{\mathbf{f}} \; Z(\mathbf{f}) = \sum_a \int_0^{\sum_k \delta_{ak} f_k} t_a(u)\, du$$

$$s.t. \; \sum_k f_k = q$$

$$f_k \geq 0$$

  - Given current path flow $\mathbf{f}$, define
    - Basic path $k^*$: the shortest path
    - Non-basic path set $P_{NB}$

$$f_{k^*} = q - \sum_{k \in P_{NB}} f_k$$

# Gradient projection

- Main idea of GP algorithm
  - Make the projection more efficient by modifying the subproblem for each OD
    - Basic path $k^*$: the shortest path
    - Non-basic path set $P_{NB}$

$$\min_{\mathbf{f}} \ \hat{Z}(\mathbf{f}) = \sum_a \int_0^{\delta_{ak^*}\left(q - \sum_{k\in P_{NB}} f_k\right) + \sum_{k\in P_{NB}} \delta_{ak} f_k} t_a(u)\,\mathrm{d}u$$

$$s.t.\ f_k \geq 0$$

  - ***Q: How to implement the projection algorithm?***

# Gradient projection

- At iteration $n$,
  - Step 1: Find descent direction $\mathbf{d}^n = -\nabla\hat{Z}(\mathbf{f}^n)$

$$\frac{\partial \hat{Z}(\mathbf{f})}{\partial f_k} = \sum_{a\in A} t_a(x_a)\left[\frac{\partial \delta_{ak^*}\left(q - \sum_{k'\in P_{NB}} f_{k'}\right)}{\partial f_k} + \frac{\partial \sum_{k'\in P_{NB}} \delta_{ak'} f_{k'}}{\partial f_k}\right]$$

$$= \sum_{a\in A} t_a(x_a)\left(\delta_{ak} - \delta_{ak^*}\right) = c_k - c_{k^*}$$

where $c_k = \sum_{a\in A} t_a(x_a)\,\delta_{ak}$ is the current travel time of path $k$

# Gradient projection

- At iteration $n$,
  - Step 1: Find descent direction $\mathbf{d}^n = -\nabla\hat{Z}(\mathbf{f}^n) = c_{k^*}^n \mathbf{1} - \boldsymbol{c}^n$
  - Step 2: Compute candidate path flow $\mathbf{y}^n = \mathbf{f}^n + \alpha\mathbf{d}^n$
    - Set step size $\alpha$ based on quasi-Newton method

$$\frac{\partial^2 \hat{Z}(\mathbf{f})}{\partial f_k^2} = \frac{\partial}{\partial f_k}(c_k - c_{k^*}) = \frac{\partial}{\partial f_k}\left[\sum_{a\in A} t_a(x_a)(\delta_{ak} - \delta_{ak^*})\right] = \sum_{a\in A} t'_a(x_a)(\delta_{ak} - \delta_{ak^*})^2$$

$$\alpha_k = \left(\frac{\partial^2 \hat{Z}(\mathbf{f}^n)}{\partial f_k^2}\right)^{-1} \Rightarrow \quad y_k^n = f_k^n - \alpha_k(c_k^n - c_{k^*}^n)$$

# Gradient projection

- At iteration $n$,
  - Step 1: Find descent direction $\mathbf{d}^n = -\nabla\hat{Z}(\mathbf{f}^n) = c_{k^*}^n\mathbf{1} - \boldsymbol{c}^n$
  - Step 2: Compute candidate path flow $\mathbf{y}^n = \mathbf{f}^n + \alpha\mathbf{d}^n$
    - Set step size $\alpha$ based on quasi-Newton method
  - Step 3: Project back to feasible set $\mathbf{f}^{n+1} = [\mathbf{y}^n]_+$
    - Since demand constraint $\Lambda\mathbf{f} = q$ has been taken into account, we only need to project the possible negative flow

# Gradient projection

- At iteration $n$,
  - Step 1: Find descent direction $\mathbf{d}^n = -\nabla\hat{Z}(\mathbf{f}^n) = c_{k^*}^n\mathbf{1} - \boldsymbol{c}^n$
  - Step 2: Compute candidate path flow $\mathbf{y}^n = \mathbf{f}^n + \alpha\mathbf{d}^n$
    - Set step size $\alpha$ based on quasi-Newton method
  - Step 3: Project back to feasible set $\mathbf{f}^{n+1} = [\mathbf{y}^n]_+ = \max\{\mathbf{0}, \mathbf{y}^n\}$
  - Step 4: Convergence check
    - Terminate if $\mathbf{f}^{n+1} = \mathbf{f}^n$ and return $\mathbf{f}^* = \mathbf{f}^n$

# Questions?