# CE 272
# Traffic Network Equilibrium

Lecture 15

## Bicriterion Shortest Paths

# Previously on Traffic Network Equilibrium...

The function $\hat{t}_{ij}(x) = t_{ij}(x) + x\, t'_{ij}(x)$ is called the marginal cost function. It consists of

- The original delay function $t_{ij}(x)$
- The externality caused by an additional traveler $xt'_{ij}(x)$

Externalities are costs/benefits incurred due to one's actions by all the other agents in the system.

In the context of traffic, when an additional traveler takes link $(i, j)$ he or she increases the travel time by $t'_{ij}(x)$. This imposes a negative externality of $xt'_{ij}(x)$ on all users on $(i, j)$.

**Proposition (System Optimal)**

*At an SO state, all used routes have equal and minimal marginal costs.*

# Previously on Traffic Network Equilibrium...

Thus, by setting tolls that equal the congestion externalities imposed by a traveler one can achieve a SO solution! In other words, solve the SO problem and set a toll of $x_{ij}^{SO} t'_{ij}(x_{ij}^{SO})$ on each link.

When a network has tolls, we will assume that travelers minimize **generalized cost of travel** $= \gamma$ (travel time) $+$ toll.

$\gamma$ is the value of time (VoT) measured in ₹/min. For now, assume that all travelers have the same VoT of 1.

# Lecture Outline

1. Bicriterion Shortest Paths
2. Example
3. All-or-Nothing Assignment

# Lecture Outline

## Bicriterion Shortest Paths

# Bicriterion Shortest Paths

## Introduction

In the models studied so far, the value of time (VoT) for all travelers was assumed to be the same.

However, in reality, VoT varies across population and hence for the same set of link travel times and link tolls, different individuals might find different paths "shortest".

# Bicriterion Shortest Paths

## Introduction

We will begin by studying the problem of equilibrium flows when:

- The VoT parameter for an OD pair is a random variable with a known distribution.
- Fixed values of tolls are collected on different links and this information is known.

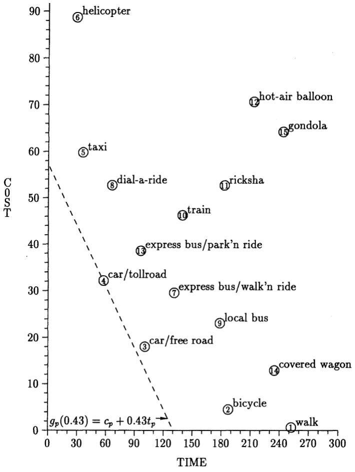
In the next set of lectures, we will also explore system optimality and the right set of tolls for this problem.

The distribution of the VoT parameter is usually assumed to follow income distribution or can be estimated from mixed-logit models or may be estimated from data on path choices.

To understand the impact of variability in VoT, let us look at a related example. Consider the process of selecting a travel mode.

Each mode takes a certain amount of time and cost as shown by the points in the adjacent figure.

Among the 15 options, just as with paths, we will pick a mode which minimizes the generalized cost of travel $\gamma t_p + c_p$, where $c_p$ and $t_p$ are the cost and time of mode $p$ and $\gamma$ is the VoT.



Imagine a traveler with a VoT of 0.43 ₹/min. He/she will choose a mode that minimizes $0.43t_p + c_p$.

Construct lines with slope -0.43 and the line closest to the origin that intercepts the modes will be optimal. Hence, car/tollroad will be chosen.

What if the VoT is 0.44? 0.45? There exists a range of VoT values for which choosing car/tollroad is optimal.

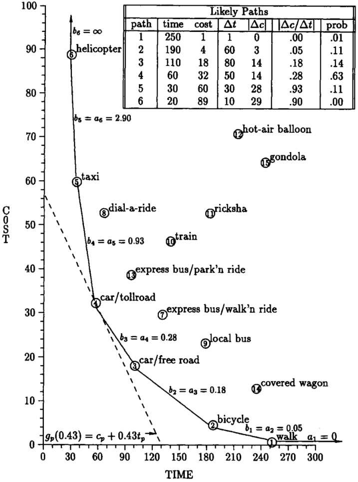
Will someone choose hot-air balloon or dial-a-ride in this example?

# Bicriterion Shortest Paths

## Efficiency Frontier

For different values of $\gamma$, only the modes 1-6 maybe optimal. The boundary joining these modes is called the **efficiency frontier**. We will refer to points where the slopes change as **extreme points**.

As VoT increases, travel times of the optimal mode decreases but the cost of travel increases.




| Likely Paths | | | | | | |
|---|---|---|---|---|---|---|
| path | time | cost | $\Delta t$ | $\lvert\Delta c\rvert$ | $\lvert\Delta c/\Delta t\rvert$ | prob |
| 1 | 250 | 1 | 1 | 0 | .00 | .01 |
| 2 | 190 | 4 | 60 | 3 | .05 | .11 |
| 3 | 110 | 18 | 80 | 14 | .18 | .14 |
| 4 | 60 | 32 | 50 | 14 | .28 | .63 |
| 5 | 30 | 60 | 30 | 28 | .93 | .11 |
| 6 | 20 | 89 | 10 | 29 | .90 | .00 |

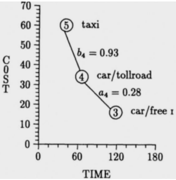
# Bicriterion Shortest Paths

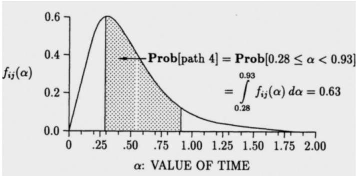
## Path Choice Probabilities

Let us now try and address the question of how many travelers will choose a certain path. For illustrative purposes, say there are a 100 travelers who are deciding which mode to take.

Clearly, modes not on the efficiency frontier will not be chosen. To determine the demand for modes 1 to 6, we use the density function of the VoT.

For example, the probability of selecting car/tollroad is the probability which which $\gamma$ lies between the negative of the slope connecting modes 4 and 5 and the negative of the slope of the line connecting modes 4 and 3.



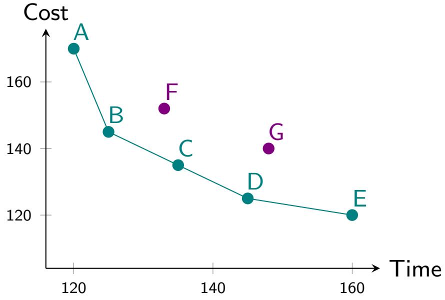




Thus, 100(0.63) travelers will choose the car/tollroad option.

# Bicriterion Shortest Paths

## Likely Paths

Define likely paths (modes) as paths (modes) that have non-zero probability of being selected. All remaining paths are unlikely.




In the above figure, identify

- Points on the efficiency frontier: A, B, C, D, E
- Extreme points: A, B, D, E
- Likely paths: A, B, D, E (if $\gamma$ is continuous); Potentially A, B, C, D, E (if $\gamma$ is discrete)

# Bicriterion Shortest Paths

## Estimating VoT PDF

For bicriterion shortest path and equilibrium models, we assume that the VoT PDF is given.

One way of estimating it in practice is to collect data on how many users take different routes between an OD pair in a network.

This can be used to estimate the empirical probability of selecting the likely paths. Since the travel times and tolls on these paths can be measured, we also know the slopes of the line segments connecting points on the efficiency frontier.

A curve fitting technique can be used to match the probability of selecting paths on the efficiency frontier.

# Bicriterion Shortest Paths

## Algorithm

For a given $\gamma$, the path that minimizes the generalized cost of travel can be obtained using standard shortest path algorithms.

To enumerate all the likley paths on the efficiency frontier, we can construct an algorithm that first estimates the slope of the line connecting the fastest path and cheapest path.

We then seek a new optimal path that minimizes the generalized cost with VoT equal to negative of the estimated slope.

If the new path has better generalized cost, more paths can be discovered on either sides of the newly found path using recursion.

# Bicriterion Shortest Paths

## Algorithm

In the algorithm that follows, Label Correcting takes arguments – a graph G, origin node $r$, and the VoT $\gamma$, which is used to convert link weights to link generalized costs (i.e., $\gamma t_{ij} + c_{ij}$).

Assume that we break ties in favor of lower cost paths.

```
LIKELY PATHS(G, r, p)
  k ← 1
  p̂ ← LABEL CORRECTING(G, r, ∞)
  p_1 ← LABEL CORRECTING(G, r, 0)
  MID PATHS (G, r, p̂, p, k)
```

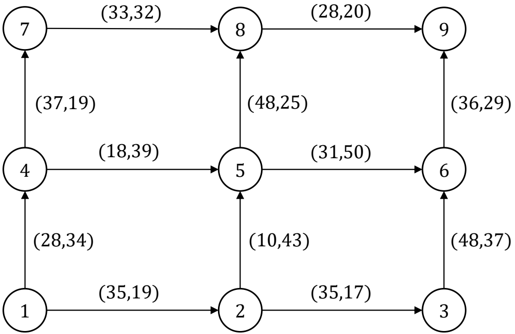
```
MID PATHS(G, r, p̂, p, k)
  γ ← −(c_p̂ − c_{p_k})/(t_p̂ − t_{p_k})
  p̃ ← LABEL CORRECTING(G, r, γ)
  if g_p̃ < γ t_p̂ + c_p̂ then
      MID PATHS (G, r, p̃, p, k)
      MID PATHS (G, r, p̂, p, k)
  else
      k ← k + 1
  end if
  p_k ← p̂
```

# Example

# Example

## 9 Node Network

Consider the following 9 node network with link travel times and costs $(t_{ij}, c_{ij})$. Assume that the origin and destination nodes are 1 and 9 respectively.



# Example

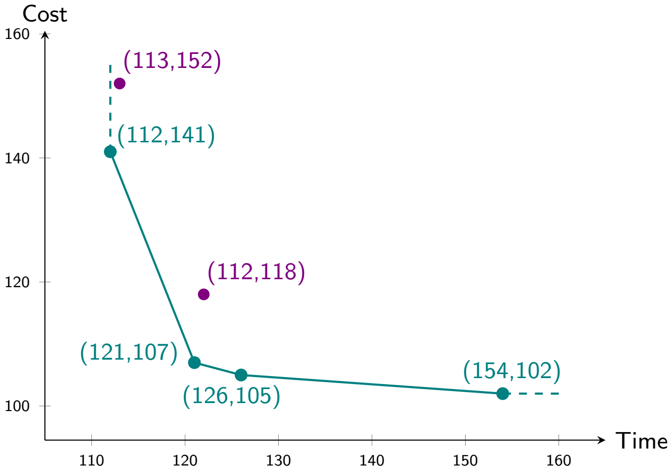
## 9 Node Network

There are 6 paths from node 1 to 9. The following table lists the path travel time and costs.

| Path | Time | Cost |
|---|---|---|
| 1-2-3-6-9 | 154 | 102 |
| 1-2-5-6-9 | 112 | 141 |
| 1-2-5-8-9 | 121 | 107 |
| 1-4-5-6-9 | 113 | 152 |
| 1-4-5-8-9 | 122 | 118 |
| 1-4-7-8-9 | 126 | 105 |

# Example

## Paths in the 9 Node Network



# Example

## Inception Time

Each mid paths function calls two additional mid paths sub routines. To keep track of the recursive steps, denote an iteration using the first or second mid paths call and also keep track of the mid paths function from where it was called.

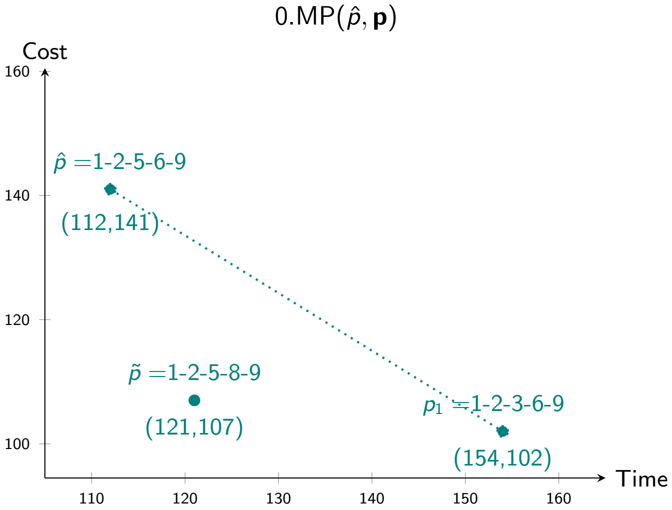
For example, let 0.MP be the first mid paths function call from the function likely paths.

We write 0.1.MP and 0.2.MP to denote the first and second calls of mid paths functions.

Thus, 0.1.2.1.MP would denote the first mid paths function called from the second instance of the mid paths function that was called from the first instance of the mid paths function called from likely paths!!

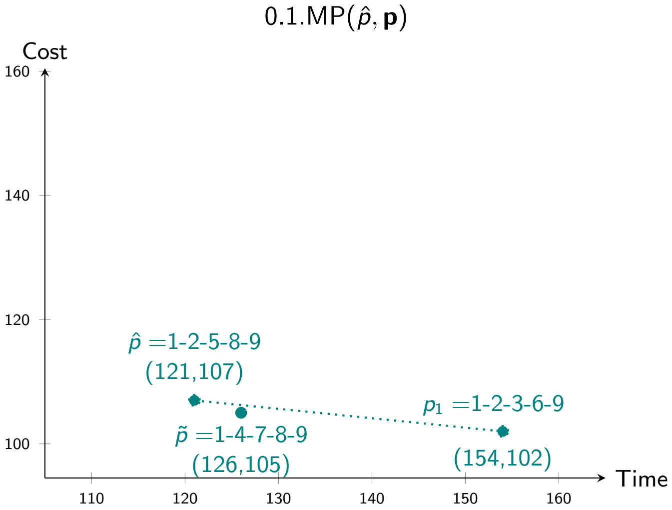
# Example

## Iteration 1



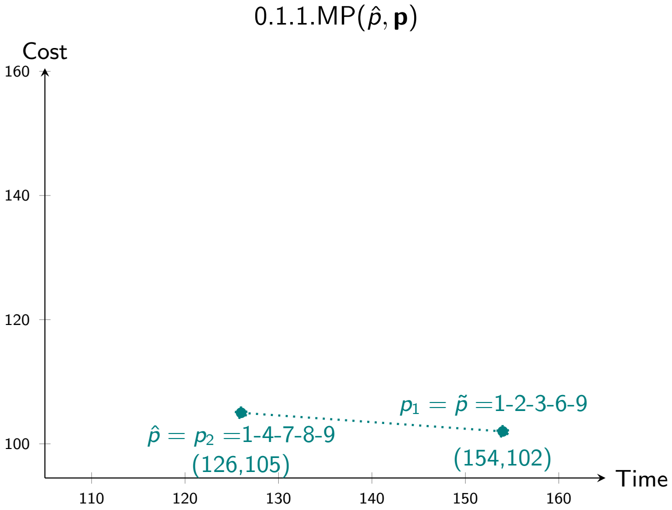
# Example

## Iteration 2



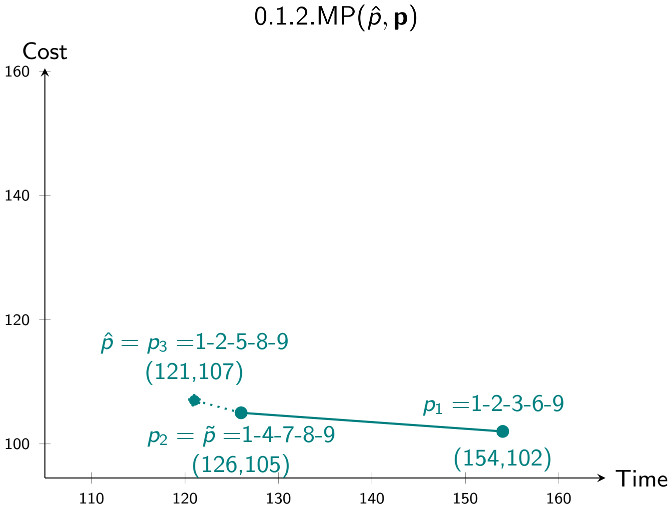
# Example

## Iteration 3



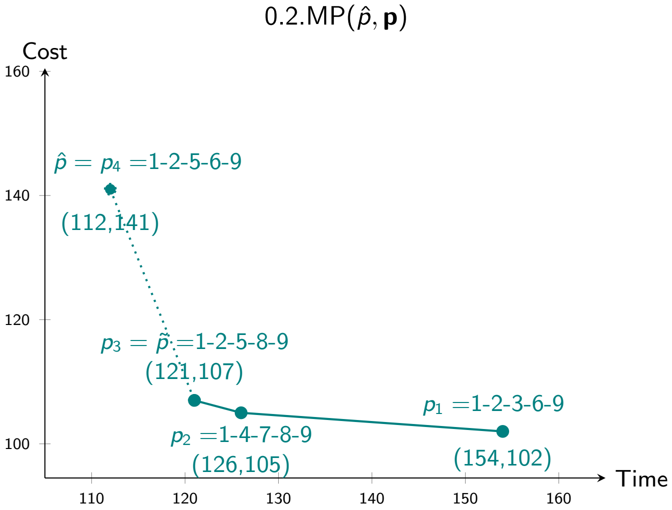
# Example

## Iteration 4



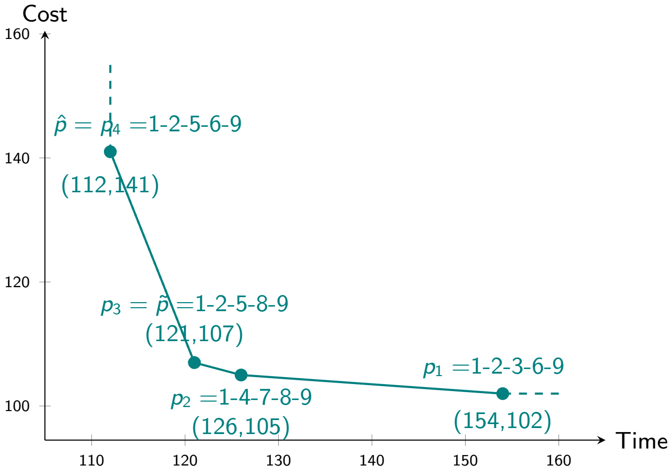
# Example

## Iteration 5



# Example

Iteration 6

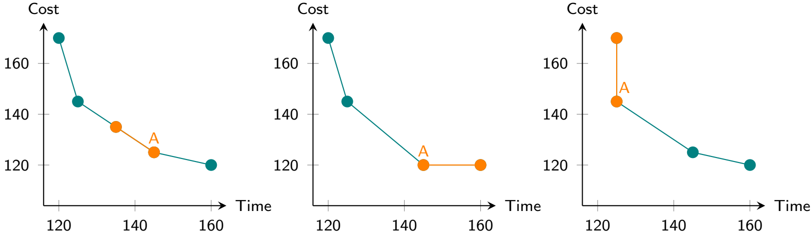


# Bicriterion Shortest Paths

## Tie Resolution

Recall that the likely paths algorithm only finds the extreme points on the efficiency frontier.

For instance, in the following scenarios, both orange paths are optimal for some $\gamma$s but we'd like to select only path A.




Such tie breaking rules can be easily incorporated in the label correcting algorithm by minor modifications to the if condition that checks the optimality principle.

# Lecture Outline

## All-or-Nothing Assignment

# All-or-Nothing Assignment

## Introduction

For a graph with fixed link travel times and tolls, we would like to load travelers on the shortest generalized cost paths to derive an all-or-nothing flow solution.

The all-or-nothing flows will be a target direction just as in regular traffic assignment.

# All-or-Nothing Assignment

## Notation

We will use the following terms in describing the all-or-nothing assignment:

| Notation | Description |
|---|---|
| $x_{ij}^r(\gamma)$ | Origin-based flow on link $(i,j)$ for trips with VoT $\gamma$ |
| $x_{ij}(\gamma)$ | Flow of trips with VoT $\gamma$ on link $(i,j) = \sum_{r \in Z} x_{ij}^r(\gamma)$ |
| $x_{ij}$ | Total flow on link $(i,j)$ |
| $f_{rs}(\gamma)$ | PDF of VoT of demand between $(r,s)$ |
| $d_{rs}(\gamma)$ | Demand of travelers with VoT $\gamma$ between $(r,s) = d_{rs} f_{rs}(\gamma)$ |

# All-or-Nothing Assignment

## Definition

Define the set $\mathcal{X}$ as the set of all feasible flow vectors where the components of each vector are the $x_{ij}^r(\gamma)$ values.

At a given flow solution $\mathbf{x}$, we know $x_{ij}^r(\gamma)$, which can be used to find $x_{ij}$ and hence the travel times and tolls. Call them $t_{ij}$ and $c_{ij}$.

Keeping the tolls and travel times fixed, we would like minimize the total generalized trip costs by assigning flows to its particular minimum generalized cost path.

Thus the all-or-nothing assignment or the min-path traffic assignment (MPA) is defined as

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\mathcal{X}} \int_0^{\infty} \sum_{(i,j)\in A} \Big(\gamma t_{ij} + c_{ij}\Big) x_{ij}(\gamma) d\gamma$$

# All-or-Nothing Assignment

## Algorithm

Estimating all-or-nothing flows for the bicriterion setting is similar to what we saw in the regular TAP. However, we may have multiple "optimal" paths between a given OD pair depending on the $\gamma$ values.

1. At a current solution $\mathbf{x}$, estimate the travel times and tolls and find all likely paths.
2. Find the probability of selecting each of the likely paths using the PDF of $\gamma$.
3. Multiply the path choice probabilities with the demand $d_{rs}$ and load them on their respective shortest paths.
4. Repeat for all OD pairs and aggregate link flows.

# Your Moment of Zen