CIVIL-557
Decision-Aid Methodologies in Transportation
Applications

Fabian Torres

Transport and Mobility Laboratory TRANSP-OR
École Polytechnique Fédérale de Lausanne EPFL

Outline

1. Crowd-shipping: Case study AmazonFlex

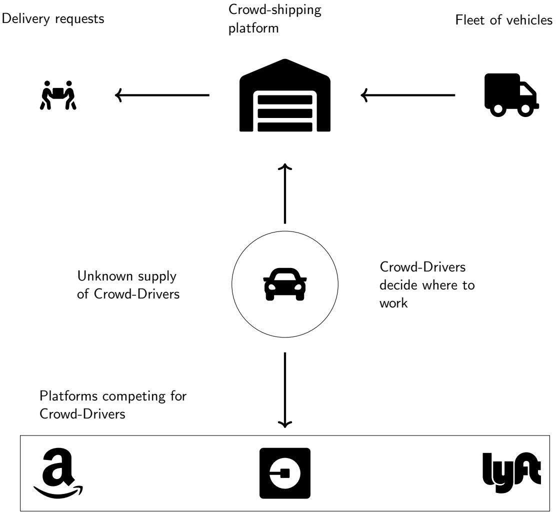
2. Tourist Itinerary planning: Cases study Unki

3. Elevator Dispatching: Case study Schindler

Outline

1. Crowd-shipping: Case study AmazonFlex
2. Tourist Itinerary planning: Cases study Unki
3. Elevator Dispatching: Case study Schindler

Crowd-shipping



Delivery requests

Crowd-shipping platform

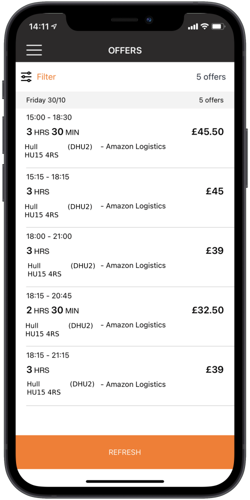
Fleet of vehicles

Unknown supply of Crowd-Drivers

Crowd-Drivers decide where to work

Platforms competing for Crowd-Drivers

AmazonFlex delivery



5 offers

Friday 30/10 5 offers

15:00 - 18:30
3 HRS 30 MIN £45.50
Hull (DHU2) - Amazon Logistics
HU15 4RS

15:15 - 18:15
3 HRS £45
Hull (DHU2) - Amazon Logistics
HU15 4RS

18:00 - 21:00
3 HRS £39
Hull (DHU2) - Amazon Logistics
HU15 4RS

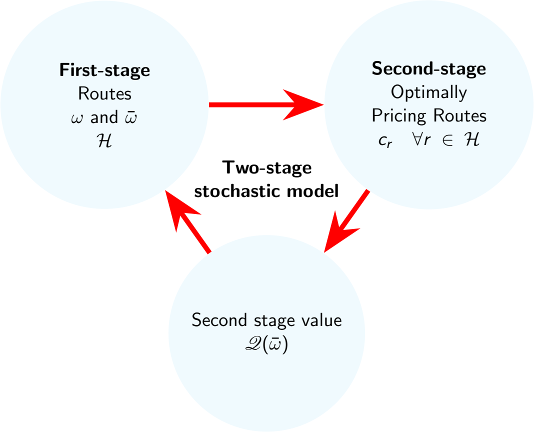
18:15 - 20:45
2 HRS 30 MIN £32.50
Hull (DHU2) - Amazon Logistics
HU15 4RS

18:15 - 21:15
3 HRS £39
Hull (DHU2) - Amazon Logistics
HU15 4RS

Model

$$
\begin{aligned}
\min \quad & \sum_{r \in \Omega \cup \Omega'} c^r \omega^r + \mathscr{Q}(\bar{\omega}) & \\
\text{s.t.} \quad & \sum_{r \in \Omega \cup \Omega'} a_i^r \omega^r = 1 & \forall i \in N \\
& \sum_{r \in \Omega'} \omega_r \leq M & \\
& \omega_r \in \{0, 1\} & \forall r \in \Omega \cup \Omega'
\end{aligned}
$$

Two-stage framework



First-stage
Routes
ω and $\bar{\omega}$
$\mathcal{H}$

Second-stage
Optimally
Pricing Routes
$c_r \quad \forall r \in \mathcal{H}$

Two-stage stochastic model

Second stage value
$\mathscr{Q}(\bar{\omega})$

Demand model: logit model

We have a set of attributes of a route that influence the decision of Crowd-drivers. For example, the total distance, stops, load, and more importantly compensation (i.e., c_r).

$$
\begin{aligned}
\mathcal{U}_r &= \sum_{i \in \mathcal{A}} \beta_i x_i^r + \beta_c c_r + \varepsilon \quad \forall r \in \mathcal{H} \\
\mathcal{U}_0 &= 0
\end{aligned}
$$

Second stage problem: Notation

- Let $\mathcal{H}$ be the set of routes created in the first stage for crowd drivers.
- $\aleph$ is the penalty if the route is rejected.
- V_r is the deterministic part of the utility function, i.e., without the error term for route "r".
- In the logit model the probability of acceptance is equal to $\frac{e^{V_r}}{e^{V_r}+1}$

Second stage problem

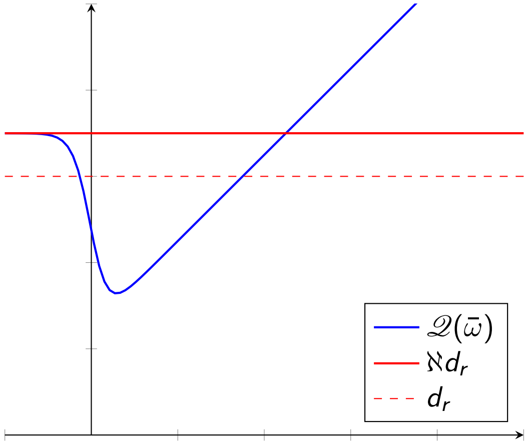
Route Assignment and Pricing (RAP)

$$c_r = \frac{V_r - x_r}{\beta_c}$$

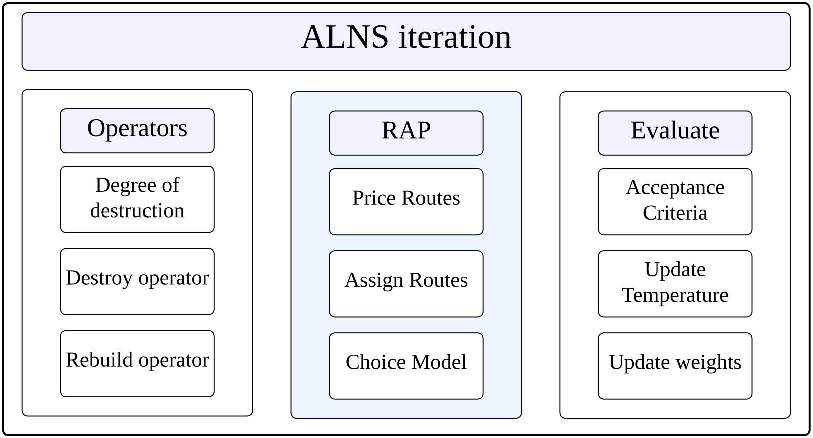
$$\mathscr{Q}(\bar{\omega}) = \min \sum_{r \in \mathcal{H}} \bar{\omega}_r \frac{e^{V_r}}{e^{V_r}+1} \cdot \frac{V_r - x_r}{\beta_c} + \sum_{r \in \mathcal{H}} \bar{\omega}_r \left(1 - \frac{e^{V_r}}{e^{V_r}+1}\right) \cdot \aleph d_r$$

Second stage problem

Route Assignment and Pricing (RAP)



ALNS


ALNS iteration
Operators
Degree of destruction
Destroy operator
Rebuild operator
RAP
Price Routes
Assign Routes
Choice Model
Evaluate
Acceptance Criteria
Update Temperature
Update weights

ALNS

```
Incumbent sol = Initial solution;
Sol = Incumbent sol;
Set k;
Set Temperature;
Set Cooling rate;
for $it = 1$ to $k$ do
    $Sol^*$ = Sol;
    Select removal operator;
    RAP;
    Select insertion operator;
    RAP;
    if $Sol^*$ value < Incumbent sol value then
        Incumbent sol = $Sol^*$;
    end
    if Accept($Sol^*$, Sol) then
        Sol = $Sol^*$;
    end
    Cool Temperature;
    if $Temperature < \epsilon$ then
        Stop;
    end
end
return Incumbent sol;
```

Outline

1 Crowd-shipping: Case study AmazonFlex

2 Tourist Itinerary planning: Cases study Unki

3 Elevator Dispatching: Case study Schindler

Tourist itinerary planning

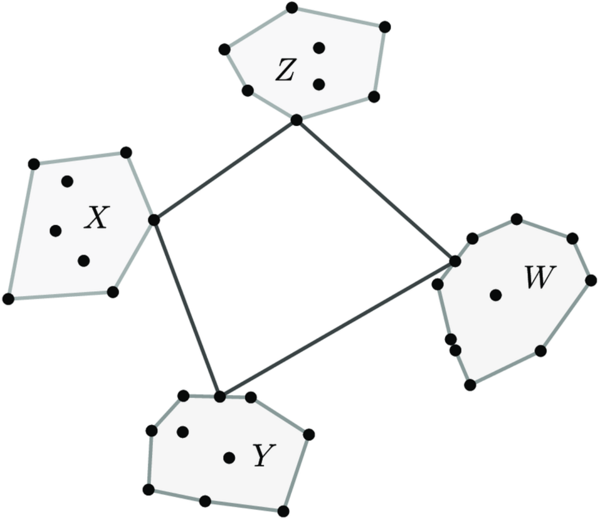
Unki

Unki is an EPFL startup. The company wants to create itineraries for tourists that visit a new city.

- Based on the preferences of customers, a utility function is created for each possible activity.
- Tourist schedules are Hotel – Activity – Eat – Activity – Eat – Hotel.
- Realistic constraints that consider the cost and time to travel from one activity to another are necessary. For example, going from the hotel to an activity that is too far away takes a lot of time.

Problem

The problem is an elementary shortest path problem with resource constraints (ESPPRC).



Problem: Notation

- $U_j = \begin{cases} U_i + U_{a_j}^{participation} + U_{a_j}^{starttime} + U_{a_i}^{duration}, & \text{if } a_j \neq a_i \\ U_i, & \text{otherwise} \end{cases}$
- $t_j = \begin{cases} t_i + t_{(a_i,a_j)}, & \text{if } a_j \neq a_i \\ t_i + 1, & \text{otherwise} \end{cases}$
- $d_j = \begin{cases} 0, & \text{if } a_j \neq a_i \\ d_i + 1, & \text{otherwise} \end{cases}$
- $c_j = c_i + c_{(a_i,a_j)}$
- $b_j = b_i + b_{(a_i,a_j)}$
- $\mathscr{U}_j = \begin{cases} \mathscr{U}_i \cup a_i, & \text{if } a_j \neq a_i \\ \mathscr{U}_i, & \text{otherwise} \end{cases}$

Dominance rules:

1. $a_1 = a_2$
2. $t_1 = t_2$
3. $d_1 = d_2$
4. $U_1 - c_1 > U_2 - c_2$
5. $b_1 \leq b_2$
6. $\mathscr{U}_1 \subseteq \mathscr{U}_2$

Labeling algorithm

Algorithm 1: Basic DP

```
Initiate time, $t \leftarrow 0$;
Initiate the list of labels, Label[1] $= \{\mathcal{L}_0 = (Hotel, 0, 1, 1, 0, 0, \{\})\}$;
while $t \neq \mathcal{T}$ do
    $t \leftarrow t + 1$;
    for $\ell_1$ in Label[t] do
        Extend label to all feasible extensions;
        $\mathcal{L}^* \leftarrow$ feasible extension;
        $t^* \leftarrow$ REF($t$);
        for $\ell_2$ in Label[$t^*$] do
            Apply dominance rules;
            if $\ell_2$ is dominated then
                delete $\ell_2$;
                continue;
            if $\mathcal{L}^*$ is dominated then
                delete $\mathcal{L}^*$;
                break;
        Label[$t^*$] $\cup \mathcal{L}^*$
return Best label at time $\mathcal{T}$
```

Outline

1. Crowd-shipping: Case study AmazonFlex

2. Tourist Itinerary planning: Cases study Unki

3. Elevator Dispatching: Case study Schindler

Elevator Dispatching Problem with Destination Control

Elevators are vertical transportation systems.

- Historically, elevators had a conductor inside that would control the elevator as passengers got on and off at each floor.
- In most modern elevators, the controls remain inside the elevator. Passengers enter the elevator and then choose the destination floor.
- In Destination Control, passengers choose the destination floor outside the elevator when making the call.

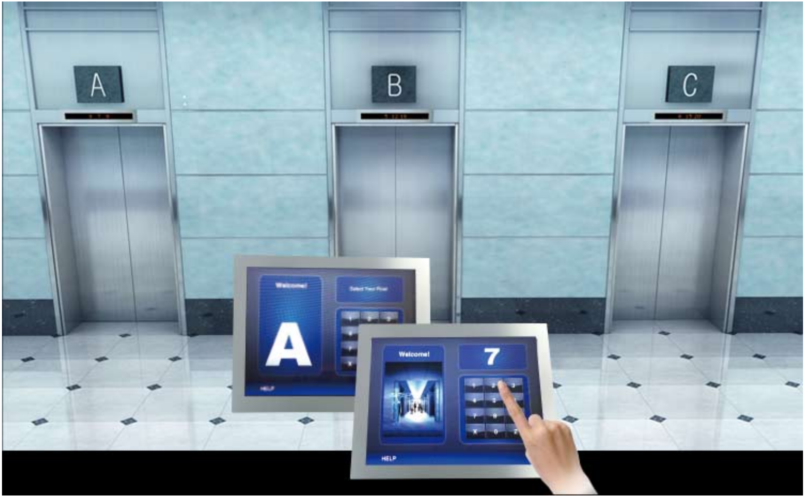
Destination Control

Schindler

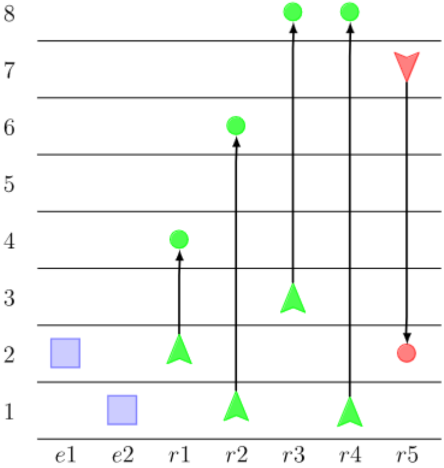
Schindler wants to evaluate the performance of their heuristic with an exact method.

We can assume that perfect information is known about future passenger arrivals.

Destination Control



Passenger Calls



KS logic

Elevators must follow a set of rules:

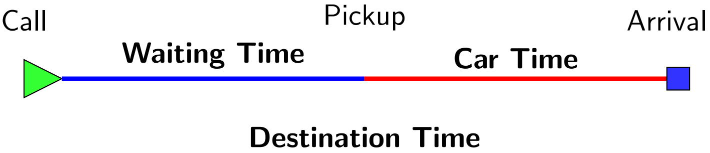
1. Passengers should never travel in the direction opposite to their destination.
2. Passengers that travel in different directions cannot be in the same elevator at the same time.
3. Passengers should not pass their destination floor without getting out.
4. The elevators can only change direction when empty

Static EDPDC

In the static problem, assume we have perfect information about passenger arrivals. **We know the future.**

Objective 1: minimize destination time

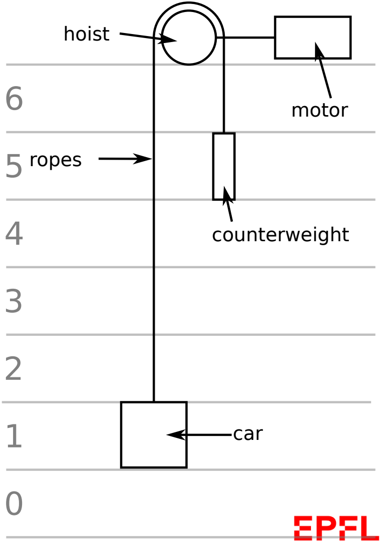
Objective: Minimize average destination time.



Objective 2: minimize energy

$$\bar{\xi}_{ij}(q) = \left(q[kg] - \frac{Q}{2}[kg]\right)\left(j - i\right) \times$$
$$distance[m] \times g[m/s^2]$$

$$\xi_{ij}(q) = \begin{cases} \frac{1}{0.75} \times \bar{\xi}_{ij}(q) & \text{if } \bar{\xi}_{ij}(q) > 0 \\ 0.75 \times \bar{\xi}_{ij}(q) & \text{if } \bar{\xi}_{ij}(q) \leq 0. \end{cases}$$



Calculating time

- **Doors open:** 3 seconds;
- **Doors close:** 4 seconds;
- **Elevator moves:** "$t_{i,j}$" time to travel from floor i to floor j
- **Waiting Time:** begins when a passenger makes a call and ends when the elevator arrives, before opening the doors.
- **Car Time:** begins when elevator arrives, before opening the doors and ends when the elevator arrives at the destination, before opening the doors.
- **Destination Time:** is the sum of waiting time and car time.
- **Passengers enter:** It takes 1 second for a passenger to enter the elevator once the doors have opened;
- **Passengers exit:** It takes 1 second for a passenger to exit the elevator once the doors have opened.

Input

- **External Id** I id of each passenger;
- **Set of origins** $= P$;
- **Set of destinations** $= D$;
- **Time of calls** $= a_i \quad \forall i \in P$;
- **Set of Elevators** $= E$;
- **Set of Floors** $= F$;
- **Elevator Capacity** $= Q_e$.

Formulation

$$
\begin{aligned}
\min \sum_{r \in \Omega} c_r \lambda_r & & \\
\text{s.t.} \quad \sum_{r \in \Omega} a_i^r \lambda^r = 1 & & \forall i \in P \\
\sum_{r \in \Omega} \lambda_r \leq |E| & & \\
\lambda_r \in \{0, 1\} & & \forall r \in \Omega
\end{aligned}
$$

Part 1: Pricing problem

The reduced cost of a variable is the following:

$$\bar{c}_r = (1-\omega)d_r + \sum_{i \in \bar{P}} \alpha_i a_i^r + \omega \times \xi - \mu \quad \forall r \in \Omega$$

- α is the dual variable from the constraint of each passengers in the master problem;
- μ is the dual variables for the constraint of the total number of elevators;
- c is the total cost, e.g., average destination time;
- ξ the energy used in the route;
- ω is a weight in the objective that multiplies the energy and the destination cost.

Labeling

Let $\mathcal{L} = (f, \ell, t, c, \delta, \bar{D}, \bar{P})$ be a label that represents a partial path (or state) of an elevator.

- f current floor;
- ℓ current load, i.e., the total number of passengers currently in the elevator;
- t the current time;
- c the cost;
- δ the direction of travel, i.e., up or down;
- D the set of deliveries;
- P the set of all passengers picked up.

Label extensions

We start with a label $\mathcal{L}_0$ and make an extension to all possible directions, i.e., pick up locations that do not violate constraints.

- **Resource extension functions (REF)** are used to keep track of resources as they pass through an arc.
- **Feasibility checks** make sure every extension is feasible, i.e., passengers do not travel in the wrong direction.
- **Dominance rules** eliminate non-promising labels

Dominance Destination time

Theorem 1:

Label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if the following conditions are met:

1. $f_1 = f_2$
2. $\delta_1 = \delta_2$
3. $t_1 \leq t_2$
4. $\ell_1 \leq \ell_2$
5. $c_1 + \ell_1 * (t_2 - t_1) \leq c_2$
6. $D_1 \subseteq D_2$
7. $P_1 \subseteq P_2$

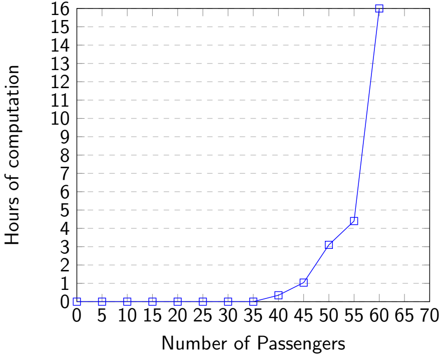
Dominance DT and Energy ξ

Theorem 2:

Label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if the following conditions are met:

1. $f_1 = f_2$
2. $\delta_1 = \delta_2$
3. $t_1 \leq t_2$
4. $\begin{cases} \ell_1 \leq \ell_2 & \text{if } \delta_1 = \text{up } \uparrow \\ \ell_1 = \ell_2 & \text{if } \delta_1 = \text{ down } \downarrow \end{cases}$
5. $c_1 + (1 - \omega)\ell_1 * (t_2 - t_1) \leq c_2.$
6. $\begin{cases} D_1 \subseteq D_2 & \text{if } \delta_1 = \text{up } \uparrow \\ D_1 \equiv D_2 & \text{if } \delta_1 = \text{ down } \downarrow \end{cases}$
7. $P_1 \subseteq P_2$

Complexity



n	Av.DT	T[sec]
5	21.20	0.15
10	34.25	0.21
15	35.75	0.43
20	38.30	0.79
25	38.96	6.63
30	39.93	5.66
35	42.23	18.99
40	42.50	1266.17
45	42.75	3759.45
50	42.50	11157.50
55	42.44	15758.81
60	42.21	58239.26