Efficient Traffic Assignment for Public Transit Networks
Lars Briem, Sebastian Buck, Holger Ebhart, Nicolai Mallig, Ben Strasser, Peter Vortisch, Dorothea Wagner, and Tobias Zündorf

SEA · July 22nd, 2017
Overview

Introduction:
- Public Transit Network
- Demand
- Problem Statement

Our Algorithm:
- Perceived Arrival Time
- Assignment
- Decision Model

Evaluation:
- Performance
- Result Quality
Public Transit Network

Timetable components:

- A set of stops $\mathcal{S}$ (stops, platforms)
- A set of connections $\mathcal{C}$
- A set of trips $\mathcal{T}$ (minimum change times, walking)
- A set of transfer edges $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ (vehicles)
Public Transit Network

Timetable components:
- A set of stops $S$ (stops, platforms)
- A set of connections $C$
- A set of trips $T$ (minimum change times, walking)
- A set of transfer edges $E \subseteq S \times S$ (vehicles)
Public Transit Network

Timetable components:
- A set of stops $S$ (stops, platforms)
- A set of connections $C$
- A set of trips $T$ (minimum change times, walking)
- A set of transfer edges $E \subseteq S \times S$ (vehicles)

Connection:
- departure stop $s_{dep} \in S$
- arrival stop $s_{arr} \in S$
- trip $t \in T$
- departure time $\tau_{dep} \in \mathbb{R}$
- arrival time $\tau_{arr} \in \mathbb{R}$
Public Transit Network

Timetable components:
- A set of stops $S$ (stops, platforms)
- A set of connections $C$
- A set of trips $T$ (minimum change times, walking)
- A set of transfer edges $E \subseteq S \times S$ (vehicles)

Connection:
- departure stop $s_{\text{dep}} \in S$
- arrival stop $s_{\text{arr}} \in S$
- departure time $\tau_{\text{dep}} \in \mathbb{R}$
- arrival time $\tau_{\text{arr}} \in \mathbb{R}$
Public Transit Network

Timetable components:
- A set of stops $S$ (stops, platforms)
- A set of connections $C$
- A set of trips $T$ (minimum change times, walking)
- A set of transfer edges $E \subseteq S \times S$ (vehicles)

Trip:
- Subsequent connections served by the same vehicle

Trip 1:
Public Transit Network

Timetable components:
- A set of stops $\mathcal{S}$ (stops, platforms)
- A set of connections $\mathcal{C}$
- A set of trips $\mathcal{T}$ (minimum change times, walking)
- A set of transfer edges $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ (vehicles)

Trip:
- Subsequent connections served by the same vehicle

Trip 2:

9:00
9:30
10:00
10:10
10:20
10:40
Public Transit Network

Timetable components:
- A set of stops $\mathcal{S}$ (stops, platforms)
- A set of connections $\mathcal{C}$
- A set of trips $\mathcal{T}$ (minimum change times, walking)
- A set of transfer edges $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ (vehicles)

Transfer graph:
- Describes possible transfers between stops
Demand

Definition:
- Demand is a list of passengers, each with:
  - An origin stop
  - A destination stop
  - A desired departure time

Example:

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Departure Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paddington</td>
<td>King’s Cross</td>
<td>8:00 am</td>
</tr>
<tr>
<td>King’s Cross</td>
<td>Temple</td>
<td>9:00 am</td>
</tr>
<tr>
<td>Paddington</td>
<td>Embankment</td>
<td>9:30 am</td>
</tr>
<tr>
<td>Piccadilly Circus</td>
<td>Westminster</td>
<td>9:30 am</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem Statement: Assignment

Given:
- A public transit network (timetable & transfer graph)
- Demand

Problem:
- Compute the utilization of every vehicle, at every given time
- Assign all passengers to journeys
Problem Statement: Assignment

Given:
- A public transit network (timetable & transfer graph)
- Demand

Problem:
- Compute the utilization of every vehicle, at every given time
- Assign all passengers to journeys

Note:
- A passenger may be assigned proportionally to multiple journeys
- Assigned journeys are not necessarily optimal
Perceived Arrival Time (PAT)

**Purpose:**
- Associated with a connection $c$ and a specific destination $d$
- Measures how useful $c$ is for reaching $d$
- Depends on four parameters:
  - Cost for changing between vehicles $\lambda_{\text{trans}}$
  - Cost factor for waiting $\lambda_{\text{wait}}$
  - Cost factor for walking $\lambda_{\text{walk}}$
  - The maximum delay of a connection $\Delta_{\tau}^{\text{max}}$

**Assumption:**
- Passengers try to optimize their PAT
Perceived Arrival Time (PAT)

**Formal definition:**

\[ \tau_{\text{arr}}^p(c, c', d) := \tau_{\text{trans}}^p(c, c') + \tau_{\text{wait}}^p(c, c') + \tau_{\text{arr}}^p(c', d) \]

\[ \tau_{\text{arr}}^p(c, d \mid \text{walk}) := \begin{cases} \tau_{\text{arr}}(c) & \text{if } v_{\text{arr}}(c) = d \\ \tau_{\text{arr}}(c) + \lambda_{\text{walk}} \cdot \tau_{\text{trans}}(v_{\text{arr}}(c), d) & \text{otherwise} \end{cases} \]

\[ T(c) := \{ c' \in C \mid \text{trip}(c') = \text{trip}(c) \land \tau_{\text{dep}}(c') \geq \tau_{\text{arr}}(c) \} \]

\[ \tau_{\text{arr}}^p(c, d \mid \text{trip}) := \begin{cases} \min \{ \tau_{\text{arr}}^p(c', d) \mid c' \in T(c) \} & \text{if } T(c) \neq \emptyset \\ \infty & \text{otherwise} \end{cases} \]

\[ \tau_{\text{arr}}^p(c, c', d) := \tau_{\text{trans}}^p(c, c') + \tau_{\text{wait}}^p(c, c') + \tau_{\text{arr}}^p(c', d) \]

\[ R(c) := \{ c' \in C \mid \tau_{\text{wait}}(c, c') \geq 0 \} \]

\[ R_{\text{opt}}(c) := \{ c' \in R(c) \mid \forall \bar{c} \in R(c) : \tau_{\text{wait}}(c, \bar{c}) \geq \tau_{\text{wait}}(c, c') \Rightarrow \tau_{\text{arr}}^p(c, \bar{c}, d) \geq \tau_{\text{arr}}^p(c, c', d) \} \]

\[ \langle c_1, \ldots, c_k \rangle \text{ with } \forall i \in [1, k]: c_i \in R_{\text{opt}}(c) \land \forall i \in [2, k]: \tau_{\text{wait}}(c, c_i) \geq \tau_{\text{wait}}(c, c_{i-1}) \]

\[ \tau_{\text{wait}}^c(i) := \begin{cases} \tau_{\text{wait}}(c, c_i) & \text{if } i \in [1, k] \\ -\infty & \text{otherwise} \end{cases} \]

\[ \tau_{\text{arr}}^p(c, d \mid \text{trans}) := \begin{cases} \sum_{i=1}^{k} \left( \frac{P[\tau_{\text{wait}}^c(i-1) < \Delta \tau \leq \tau_{\text{wait}}^c(i)]}{P[\Delta \tau \leq \tau_{\text{wait}}^c(k)]} \cdot \tau_{\text{arr}}^p(c, c_i, d) \right) & \text{if } k > 0 \\ \infty & \text{otherwise} \end{cases} \]
Perceived Arrival Time (PAT)

Example:

- $\lambda_{\text{walk}} = 3$, $\lambda_{\text{wait}} = 2$, $\lambda_{\text{trans}} = 5 \text{ min}$

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_4$</td>
<td></td>
</tr>
<tr>
<td>$c_3$</td>
<td></td>
</tr>
<tr>
<td>$c_2$</td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td></td>
</tr>
</tbody>
</table>
Perceived Arrival Time (PAT)

Example:

- \( \lambda_{\text{walk}} = 3, \ \lambda_{\text{wait}} = 2, \ \lambda_{\text{trans}} = 5 \text{ min} \)

- **Case 1:** Connection \( c \) reaches destination

\[ \Rightarrow \text{PAT} = \text{arrival time } \tau_{\text{arr}}(c) \]
Perceived Arrival Time (PAT)

Example:

- $\lambda_{\text{walk}} = 3$, $\lambda_{\text{wait}} = 2$, $\lambda_{\text{trans}} = 5 \text{ min}$
- **Case 2**: Walk from connection $c$ to destination
  \[ \Rightarrow \text{PAT} = \tau_{\text{arr}}(c) + (\lambda_{\text{walk}} \cdot \tau_{\text{walking}}) \]

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_4$</td>
<td>11:00</td>
</tr>
<tr>
<td>$c_3$</td>
<td>11:10</td>
</tr>
<tr>
<td>$c_2$</td>
<td></td>
</tr>
<tr>
<td>$c_1$</td>
<td></td>
</tr>
</tbody>
</table>
Perceived Arrival Time (PAT)

Example:

- \( \lambda_{walk} = 3, \ \lambda_{wait} = 2, \ \lambda_{trans} = 5 \text{ min} \)
- **Case 3:** Continue with con. \( c' \) of same trip
  \[ \Rightarrow \text{PAT} = \text{PAT} \ c' \]

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_4 )</td>
<td>11:00</td>
</tr>
<tr>
<td>( c_3 )</td>
<td>11:10</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>11:00</td>
</tr>
<tr>
<td>( c_1 )</td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**
- \( c_1: 9:00 - 9:30 \)
- \( c_2: 9:40 - 10:30 \)
- \( c_3: 10:10 - 10:40 \)
- \( c_4: 10:32 - 11:00 \)
- Destination
- 5 min
- 15 min
- 10 min
Perceived Arrival Time (PAT)

Example:
- $\lambda_{\text{walk}} = 3$, $\lambda_{\text{wait}} = 2$, $\lambda_{\text{trans}} = 5$ min
- **Case 4:** Continue with con. $c'$ of different trip

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_4$</td>
<td>11:00</td>
</tr>
<tr>
<td>$c_3$</td>
<td>11:10</td>
</tr>
<tr>
<td>$c_2$</td>
<td>11:00</td>
</tr>
<tr>
<td>$c_1$</td>
<td></td>
</tr>
</tbody>
</table>
Perceived Arrival Time (PAT)

Example:

- \( \lambda_{\text{walk}} = 3, \quad \lambda_{\text{wait}} = 2, \quad \lambda_{\text{trans}} = 5 \text{ min} \)
- **Case 4:** Continue with con. \( c' \) of different trip

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_4 )</td>
<td>11:00</td>
</tr>
<tr>
<td>( c_3 )</td>
<td>11:10</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>11:00</td>
</tr>
<tr>
<td>( c_1 )</td>
<td></td>
</tr>
</tbody>
</table>

\( o_1: 11:00 + 5 + 3 \cdot 5 + 2 \cdot 5 = 11:30 \)

\( o_2: 11:10 + 5 + 3 \cdot 15 + 2 \cdot 25 = 12:50 \)
Perceived Arrival Time (PAT)

Example:
- $\lambda_{\text{walk}} = 3$, $\lambda_{\text{wait}} = 2$, $\lambda_{\text{trans}} = 5 \text{ min}$
- **Case 4:** Continue with some option $o_i$

\[
PAT = \sum_i \left( \text{transfer probability}(o_i) \cdot o_i \right)
\]

<table>
<thead>
<tr>
<th>Connection</th>
<th>PAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_4$</td>
<td>11:00</td>
</tr>
<tr>
<td>$c_3$</td>
<td>11:10</td>
</tr>
<tr>
<td>$c_2$</td>
<td>11:00</td>
</tr>
<tr>
<td>$c_1$</td>
<td></td>
</tr>
</tbody>
</table>

Case 4:
- Continue with some option $o_i$

\[
P[1] \cdot o_1 + (P[2] - P[1]) \cdot o_2
\]

- $o_1: 11:00 + 5 + 3 \cdot 5 + 2 \cdot 5 = 11:30$
- $o_2: 11:10 + 5 + 3 \cdot 15 + 2 \cdot 25 = 12:50$

Diagram:
- $c_1: 9:00 - 9:30$
- $c_2: 9:40 - 10:30$
- $c_3: 10:10 - 10:40$
- $c_4: 10:32 - 11:00$
- Destination
Perceived Arrival Time (PAT)

Example:

- \( \lambda_{walk} = 3 \), \( \lambda_{wait} = 2 \), \( \lambda_{trans} = 5 \) min
- **Case 4:** Continue with some option \( o_i \)
  \[
  \Rightarrow \text{PAT} = \sum_i \left( \text{transfer probability}(o_i) \cdot o_i \right)
  \]

0.5 \cdot 11:30 + 0.5 \cdot 12:50 = 12:10

\( o_1: 11:00 + 5 + 3 \cdot 5 + 2 \cdot 2.5 = 11:30 \)

\( c_4 \): 11:00
\( c_3 \): 11:10
\( c_2 \): 11:00
\( c_1 \): 12:10

\( c_1 \): 9:00 – 9:30
\( c_2 \): 9:40 – 10:30
\( c_3 \): 10:10 – 10:40
\( c_4 \): 10:32 – 11:00

5 min P = 50%

15 min P = 100%

10 min

Destination

P = 50%

P = 100%
Our Algorithm

Concept:
- Simulate passengers movement through the network
- Decide per connection $c$, which passengers use $c$
- Passengers with same destination meet
  $\Rightarrow$ Have to make the same decisions
  $\Rightarrow$ Algorithm can benefit from synergy effects
Our Algorithm

Concept:
- Simulate passengers movement through the network
- Decide per connection $c$, which passengers use $c$
- Passengers with same destination meet
  ⇒ Have to make the same decisions
  ⇒ Algorithm can benefit from synergy effects

Overview:
- Sort passengers by destination
- Compute assignment for each destination in 3 steps:
  - Compute PATs for every connection
  - Simulate passenger movement based on PATs
  - Remove unwanted cycles from journeys (optional)
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

1. Generate passengers from demand

Time: 9:00
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

2. Decide which passengers enter the connection

Time: 9:00

- $c_1$: 9:00 – 9:30
  - PAT: 11:30
- $c_2$: 9:35 – 10:30
  - PAT: 11:30
- $c_3$: 9:40 – 10:30
  - PAT: 11:00
- $c_4$: 10:10 – 10:40
  - PAT: 11:10
- $c_5$: 10:32 – 11:00
  - PAT: 11:00

Destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not
- Decide which passengers leave the trip

Time: 9:00

Connections:
- \( c_1 \): 9:00 – 9:30
- \( c_2 \): 9:35 – 10:30
- \( c_3 \): 9:40 – 10:30
- \( c_4 \): 10:10 – 10:40
- \( c_5 \): 10:32 – 11:00

PAT:
- 9:00
- 9:30
- 11:00
- 11:30
- 11:00
- 11:10
- 11:00
- 11:00

Destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

4. Move disembarking passengers to their next stop

Time: 9:00

c1: 9:00 – 9:30
PAT: 11:30

c2: 9:35 – 10:30
PAT: 11:30

c3: 9:40 – 10:30
PAT: 11:00

c4: 10:10 – 10:40
PAT: 11:10

PAT: 11:00

PAT: 11:00

PAT: 11:00

destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

1. Generate passengers from demand
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

2. Decide which passengers enter the connection

Time: 9:35

- Connection c1: 9:00 – 9:30, PAT: 11:30
- Connection c2: 9:35 – 10:30, PAT: 11:30
- Connection c3: 9:40 – 10:30, PAT: 11:00
- Connection c4: 10:10 – 10:40, PAT: 11:10
- Connection c5: 10:32 – 11:00, PAT: 11:00

Destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

3. Decide which passengers leave the trip

Time: 9:35
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

4. Move disembarking passengers to their next stop

Time: 9:35
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not
  1. Generate passengers from demand

Time: 9:40

Diagram:
- Connection c1: 9:00 – 9:30, PAT: 11:30
- Connection c2: 9:35 – 10:30, PAT: 11:30
- Connection c3: 9:40 – 10:30, PAT: 11:00
- Connection c4: 10:10 – 10:40, PAT: 11:10
- Connection c5: 10:32 – 11:00, PAT: 11:00

Destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

2. Decide which passengers enter the connection

---

**Time:** 9:40

c_1: 9:00 – 9:30
PAT: 11:30

c_2: 9:35 – 10:30
PAT: 11:30

PAT: 11:00

PAT: 11:10

PAT: 11:00

destination
Assignment Computation – Example

- Process connections in ascending order by departure time
- Decide whether passengers use a connection or not

3. ...

Time: 9:40
Assignment Computation – Decision Graph

- Generate passengers from demand
  - Reached destination?
    - Yes
      - Demand satisfied
    - No
      - Transfer to another stop?
        - Yes
          - Board next vehicle?
            - Yes
              - Sitting in vehicle
            - No
              - Waiting at stop
        - No
          - Arriving at stop
            - Yes
              - Exit vehicle at next stop
            - No
              - Waiting at stop
Decision Model

Purpose:
- Determines which connections a passenger takes
- Depends on the passenger’s delay tolerance $\lambda_{\Delta \text{max}}$
Decision Model

Purpose:
- Determines which connections a passenger takes
- Depends on the passenger’s delay tolerance $\lambda_{\Delta_{\text{max}}}$

Definition:
- Given the options $o_1, \ldots, o_k$ and their PATs
- Assign a gain $g(i)$ to every option:
  $$g(i) := \max \left( 0, \min_{j \neq i} (\text{PAT}(o_j)) - \text{PAT}(o_i) + \lambda_{\Delta_{\text{max}}} \right)$$
- The probability $P[i]$ that a passenger chooses option $i$ is:
  $$P[i] := \frac{g(i)}{\sum_{j=1}^{k} g(j)}$$
Cycles

Cycle definition:

- Visiting a stop more than once
- Assigning cycles might be undesirable
- Journey with cycle can have minimum PAT
- High waiting cost leads to cycles
Cycles

Cycle definition:

- Visiting a stop more than once
- Assigning cycles might be undesirable
- Journey with cycle can have minimum PAT
- High waiting cost leads to cycles
Cycles

Cycle definition:

- Visiting a stop more than once
- Assigning cycles might be undesirable
- Journey with cycle can have minimum PAT
- High waiting cost leads to cycles
Evaluation

Instance:

- Greater region of Stuttgart
- Reaching as far as Frankfurt, Basel or Munich
- Comprises the traffic of one day

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of vertices</td>
<td>15,115</td>
</tr>
<tr>
<td>Number of stops</td>
<td>13,941</td>
</tr>
<tr>
<td>Number of edges</td>
<td>33,890</td>
</tr>
<tr>
<td>Number of edges w/o loops</td>
<td>18,775</td>
</tr>
<tr>
<td>Number of connections</td>
<td>780,042</td>
</tr>
<tr>
<td>Number of trips</td>
<td>47,844</td>
</tr>
<tr>
<td>Number of passenger</td>
<td>1,249,910</td>
</tr>
</tbody>
</table>
Evaluation – Running Time

Used parameters:

- Walking cost factor $\lambda_{\text{walk}} = 2$
- Waiting cost factor $\lambda_{\text{wait}} = 0.5$
- Transfer cost $\lambda_{\text{trans}} = 5 \text{ min}$
- Delay tolerance $\lambda_{\Delta_{\text{max}}} = 5 \text{ min}$
- Max delay $\Delta_{\tau_{\text{max}}} = 1 \text{ min}$

Running time comparison:

- VISUM running time $\approx 30 \text{ min}$ (using 8 threads)
- Our algorithm: (passenger multiplier = 10)

<table>
<thead>
<tr>
<th>Number of threads</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running time [sec]</td>
<td>108.92</td>
<td>65.57</td>
<td>38.41</td>
</tr>
</tbody>
</table>
Evaluation – Running Time

Used parameters:

- Walking cost factor $\lambda_{\text{walk}} = 2$
- Waiting cost factor $\lambda_{\text{wait}} = 0.5$
- Transfer cost $\lambda_{\text{trans}} = 5 \text{ min}$
- Delay tolerance $\lambda_{\Delta_{\text{max}}} = 5 \text{ min}$
- Max delay $\Delta_{\tau_{\text{max}}} = 1 \text{ min}$

\{ No measurable influence on the running time, Influence the running time \}

Running time comparison:

- VISUM running time $\approx 30 \text{ min}$ (using 8 threads)
- Our algorithm: (passenger multiplier = 10)

<table>
<thead>
<tr>
<th>Number of threads</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running time [sec]</td>
<td>108.92</td>
<td>65.57</td>
<td>38.41</td>
</tr>
</tbody>
</table>
Evaluation – Running Time

- Total
- Assignment
- Cycle Elimination
- PAT
- Setup

Time [sec]

Passenger Multiplier
Evaluation – Assignment Quality

- Both assignments look similar
- VISUM produces a slightly lower travel time
- Our algorithm produces a slightly lower number of trips

<table>
<thead>
<tr>
<th>Quantity</th>
<th>VISUM</th>
<th>Our Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>mean</td>
</tr>
<tr>
<td>Total travel time [min]</td>
<td>2.98</td>
<td>46.885</td>
</tr>
<tr>
<td>Time spent in vehicle [min]</td>
<td>0.02</td>
<td>21.059</td>
</tr>
<tr>
<td>Time spent walking [min]</td>
<td>2.00</td>
<td>22.394</td>
</tr>
<tr>
<td>Time spent waiting [min]</td>
<td>0.00</td>
<td>3.432</td>
</tr>
<tr>
<td>Trips per passenger</td>
<td>1.00</td>
<td>1.771</td>
</tr>
<tr>
<td>Connections per passenger</td>
<td>1.00</td>
<td>9.396</td>
</tr>
<tr>
<td>Passengers per connection</td>
<td>0.00</td>
<td>12.740</td>
</tr>
</tbody>
</table>
Evaluation – Assignment Quality

- Both assignments look similar
- VISUM produces a slightly lower travel time
- Our algorithm produces a slightly lower number of trips

<table>
<thead>
<tr>
<th>Quantity</th>
<th>VISUM</th>
<th>Our Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>mean</td>
</tr>
<tr>
<td>Total travel time [min]</td>
<td>2.98</td>
<td>46.885</td>
</tr>
<tr>
<td>Time spent in vehicle [min]</td>
<td>0.02</td>
<td>21.059</td>
</tr>
<tr>
<td>Time spent walking [min]</td>
<td>2.00</td>
<td>22.394</td>
</tr>
<tr>
<td>Time spent waiting [min]</td>
<td>0.00</td>
<td>3.432</td>
</tr>
<tr>
<td>Trips per passenger</td>
<td>1.00</td>
<td>1.771</td>
</tr>
<tr>
<td>Connections per passenger</td>
<td>1.00</td>
<td>9.396</td>
</tr>
<tr>
<td>Passengers per connection</td>
<td>0.00</td>
<td>12.740</td>
</tr>
</tbody>
</table>
## Evaluation – Assignment Quality

- Both assignments look similar
- VISUM produces a slightly lower travel time
- Our algorithm produces a slightly lower number of trips

<table>
<thead>
<tr>
<th>Quantity</th>
<th>VISUM</th>
<th>Our Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>min</td>
<td>mean</td>
</tr>
<tr>
<td>Total travel time [min]</td>
<td>2.98</td>
<td>46.885</td>
</tr>
<tr>
<td>Time spent in vehicle [min]</td>
<td>0.02</td>
<td>21.059</td>
</tr>
<tr>
<td>Time spent walking [min]</td>
<td>2.00</td>
<td>22.394</td>
</tr>
<tr>
<td>Time spent waiting [min]</td>
<td>0.00</td>
<td>3.432</td>
</tr>
<tr>
<td>Trips per passenger</td>
<td>1.00</td>
<td>1.771</td>
</tr>
<tr>
<td>Connections per passenger</td>
<td>1.00</td>
<td>9.396</td>
</tr>
<tr>
<td>Passengers per connection</td>
<td>0.00</td>
<td>12.740</td>
</tr>
</tbody>
</table>
Evaluation – Assignment Quality

- Both assignments look similar
- VISUM produces a slightly lower travel time
- Our algorithm produces a slightly lower number of trips

<table>
<thead>
<tr>
<th>Quantity</th>
<th>VISUM</th>
<th>Our Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time [min]</td>
<td>2.98 46.885 429.00</td>
<td>2.98 47.199 429.00</td>
</tr>
<tr>
<td>Time spent in vehicle [min]</td>
<td>0.02 21.059 380.00</td>
<td>0.02 21.231 323.97</td>
</tr>
<tr>
<td>Time spent walking [min]</td>
<td>2.00 22.394 149.00</td>
<td>2.00 22.476 149.00</td>
</tr>
<tr>
<td>Time spent waiting [min]</td>
<td>0.00 3.432 217.02</td>
<td>0.00 3.492 217.02</td>
</tr>
<tr>
<td>Trips per passenger</td>
<td>1.00 1.771 6.00</td>
<td>1.00 1.746 8.00</td>
</tr>
<tr>
<td>Connections per passenger</td>
<td>1.00 9.396 109.00</td>
<td>1.00 9.474 97.00</td>
</tr>
<tr>
<td>Passengers per connection</td>
<td>0.00 12.740 <strong>1290.10</strong></td>
<td>0.00 12.847 <strong>1233.60</strong></td>
</tr>
</tbody>
</table>
Thank you for your attention!