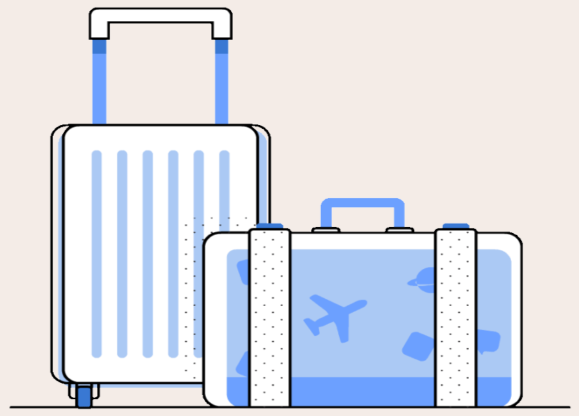
A robust public transport route planner

How to put one up against SBB

Benjamin Franklin *1789*

Nothing is certain except death, taxes and public transport being late.*

**Disclaimer: Real quote slightly different*



Passenger mains issues

- Often difficult to find the shortest path due to different transportation companies

- **Smalls** delays can have **huge** impact when multiple connections

Some numbers :

" 49% of the total passenger trip delays were the result of delayed flights (average trip delay 37 minutes), (...). 14% missed connections (average trip delay **4.4. hours**)."

Source : PASSENGER TRIP DELAY STATISTICS FOR 2010

Our Solution

- *SUGGEST FASTEST ROUTES*
- *TAKE INTO ACCOUNT PROBABLE DELAYS*
- *COMBINES MULTIPLES COMPANIES*

What route from A to B is the fastest at least Q% of the time if I want to arrive at B before instant T ?

Scope focused on Zurich

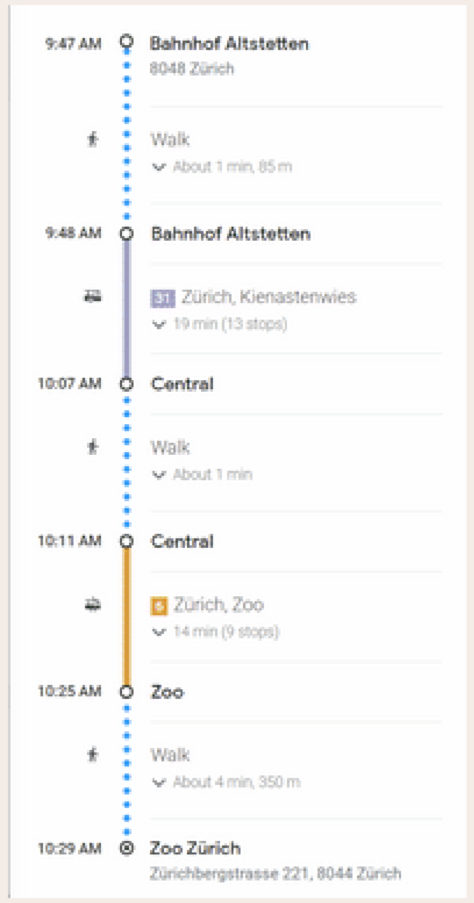
1. Zürich area
2. Evaluated on 2018 data
3. Users are willing to walk up to 500 meters
4. Sometimes you want to table a more lucky route

USER EXPERIENCE - Demonstration Break

Competitive route generation

Zurich Altstetten ⟶ Zoo, Zurich

Google Maps



9:47 AM Bahnhof Altstetten
8048 Zürich

Walk
About 1 min, 85 m

9:48 AM Bahnhof Altstetten

31 Zürich, Kienastenwies
19 min (13 stops)

10:07 AM Central

Walk
About 1 min

10:11 AM Central

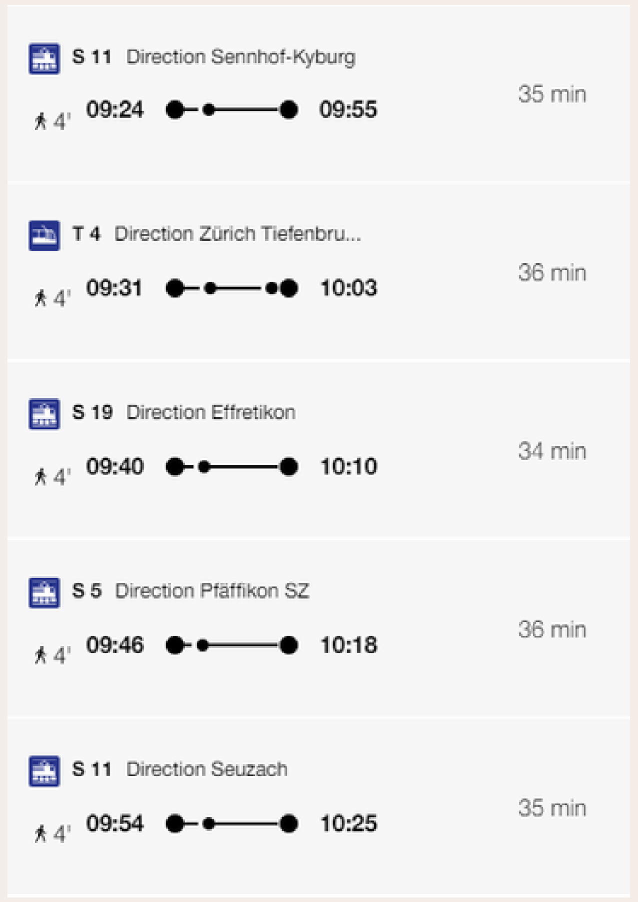
6 Zürich, Zoo
14 min (9 stops)

10:25 AM Zoo

Walk
About 4 min, 350 m

10:29 AM Zoo Zürich
Zürichbergstrasse 221, 8044 Zürich

SBB



S 11 Direction Sennhof-Kyburg
4' 09:24 09:55 35 min

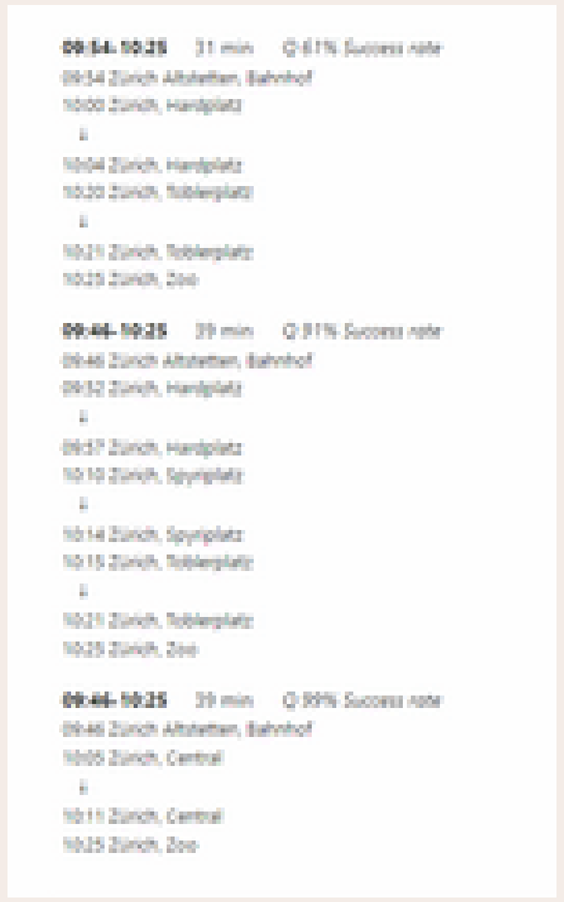
T 4 Direction Zürich Tiefenbru...
4' 09:31 10:03 36 min

S 19 Direction Effretikon
4' 09:40 10:10 34 min

S 5 Direction Pfäffikon SZ
4' 09:46 10:18 36 min

S 11 Direction Seuzach
4' 09:54 10:25 35 min

Our solution



09:54-10:25 31 min 61% Success rate
09:54 Zürich Altstetten, Bahnhof
10:00 Zürich, Hardplatz
↓
10:04 Zürich, Hardplatz
10:20 Zürich, Toblerplatz
↓
10:21 Zürich, Toblerplatz
10:25 Zürich, Zoo

09:46-10:25 39 min 91% Success rate
09:46 Zürich Altstetten, Bahnhof
09:52 Zürich, Hardplatz
↓
09:57 Zürich, Hardplatz
10:10 Zürich, Spyriplatz
↓
10:14 Zürich, Spyriplatz
10:15 Zürich, Toblerplatz
↓
10:21 Zürich, Toblerplatz
10:25 Zürich, Zoo

09:46-10:25 39 min 99% Success rate
09:46 Zürich Altstetten, Bahnhof
10:05 Zürich, Central
↓
10:11 Zürich, Central
10:25 Zürich, Zoo

- State of the art route generation
- Includes novel "feeling lucky" route

TECHNICAL IMPLEMENTATION AND CHALLENGES

Tasks and challenges to parallelize work

Main Tasks

1. Routing algorithm
   (60 % of the team)

   - Modelling the time table for routing engine
   - Developing custom time depended Dijkstra

2. Delays computation and confidence level design
   (40 % of the team)

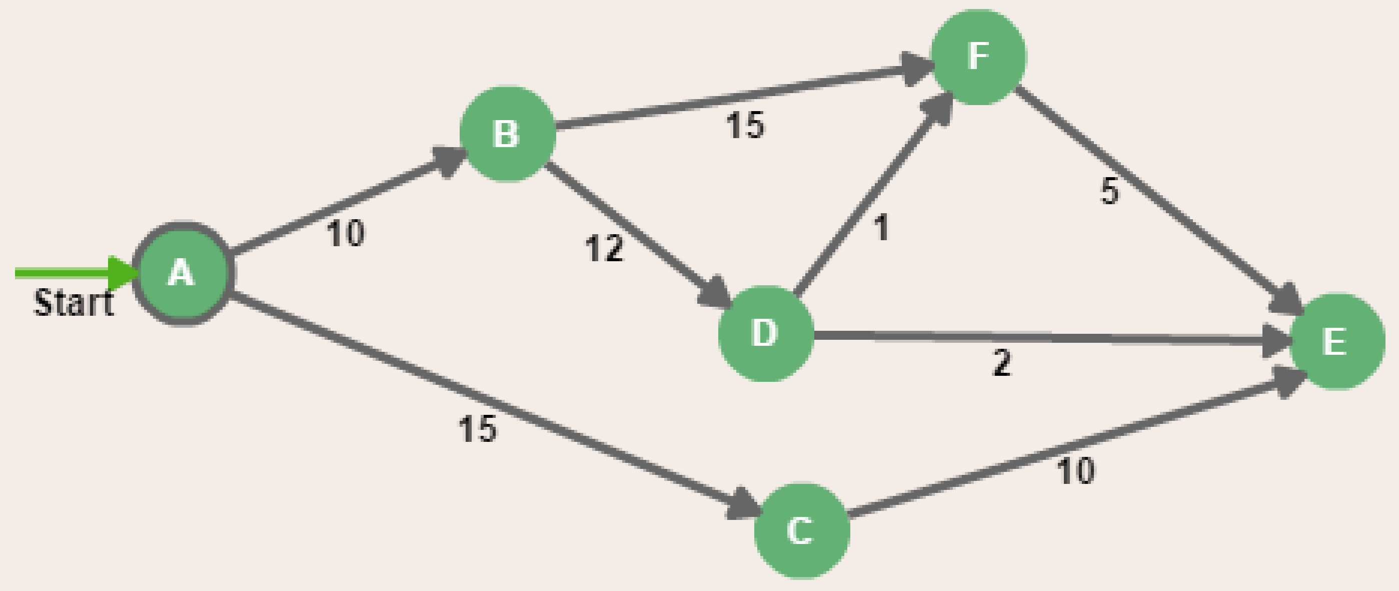
   - Keys compatibility problem (Trip_Id, Bpuic)
   - Expensive operation in terms of computation (Joins), difficulties to combine Spark with Python

3. User interface
   (40 % of the team)

   - UX for presenting generated routes
   - Inline JavaScript enabled user responsiveness

Related Challenges

Modeling the problem



Directional weighted graph

The public transport can be modelled as a graph.

Nodes are stops.

Edges are transports and possible transfers.

Preprocessing

SBB provides:
- Trips
- Transfers between platforms
- Station location

SBB does not provide:
- Stations reachable on foot from other stations

Routing

A modified version of dijkstra's shortest path algorithm

- Start from the arrival time
- Find a path in the graph that maximizes the departure time at each step.

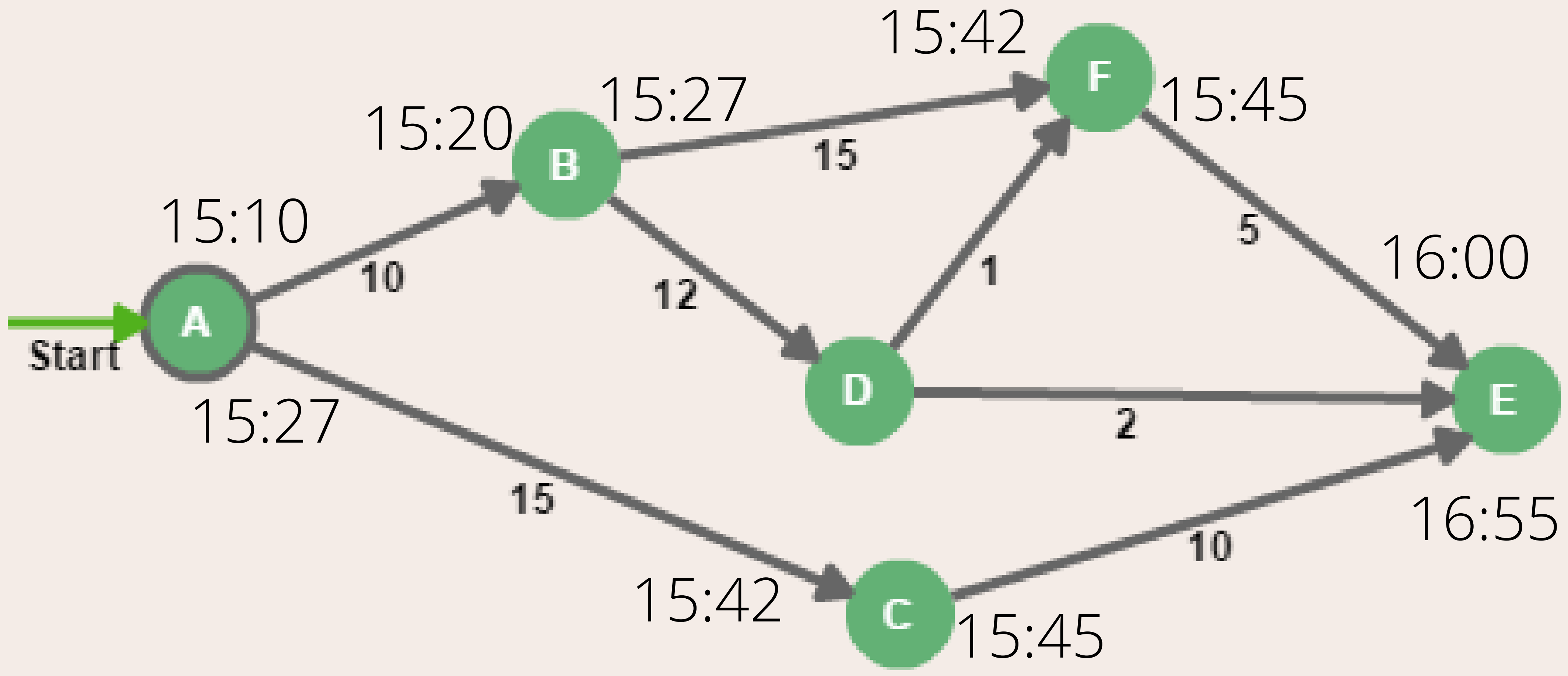
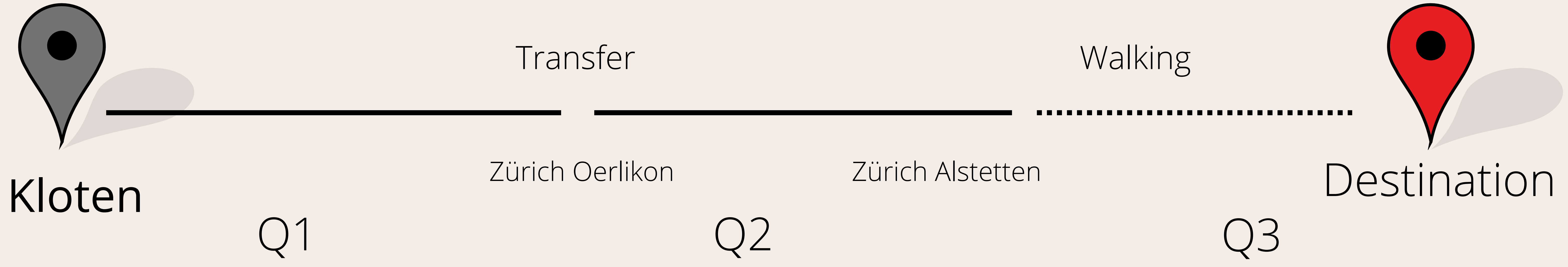


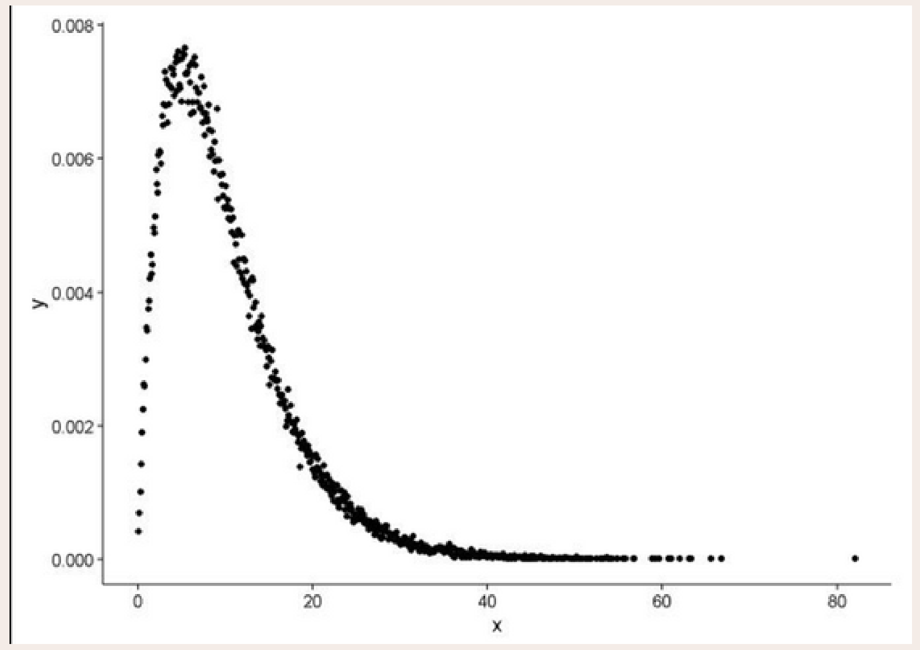
Image source: https://www.baeldung.com/java-dijkstra

Confidence level derivation

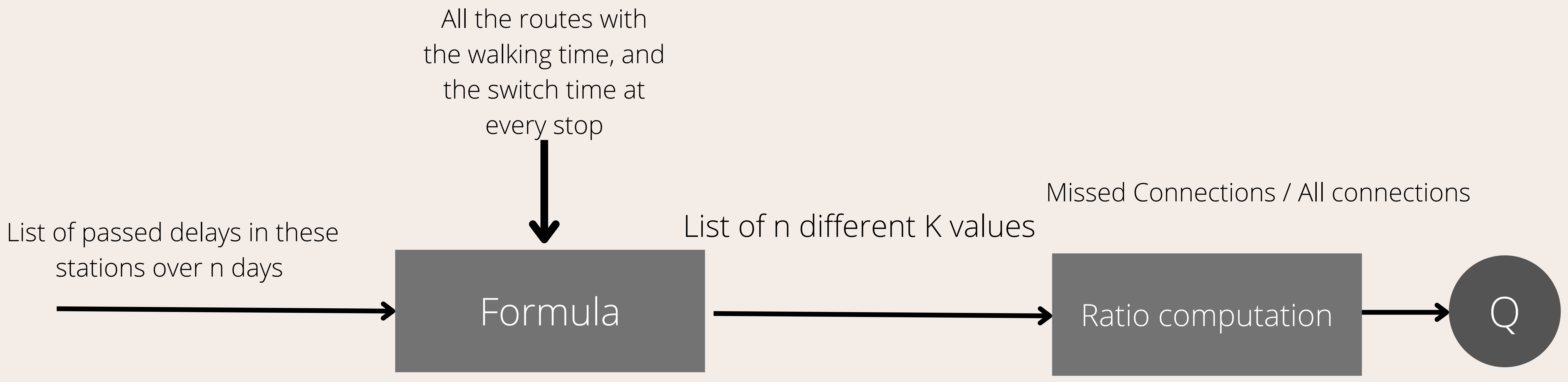


Option 1: Discarded

Fitting Gamma Distributions for each edge graph that would "learn" from the passed data.
Discarded because option 2 is way more explicit for the user



Option 2 : Retained

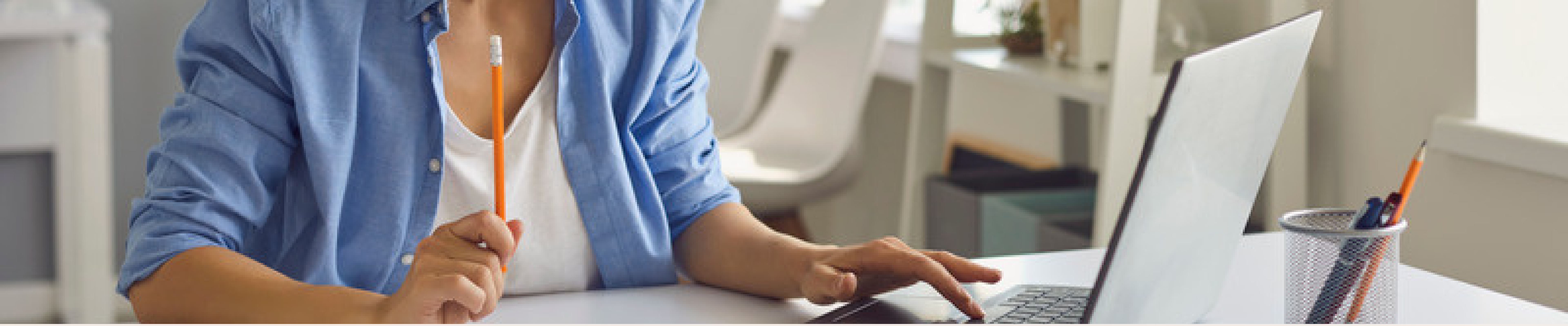

All the routes with the walking time, and the switch time at every stop
List of passed delays in these stations over n days
Formula
List of n different K values
Missed Connections / All connections
Ratio computation
Q

K = SwitchTime - WalkingTime - Delay

If K is negative, then the connection is **missed**

Q is the ratio of successful routes

Global Q = Product of all the subroutes'Q



Future improvements

Walking: We assume that we walk in a straight line regardless of natural obstacles, roads, bridges, etc..

Delays: No interdependency between delays.

Does not not rely on real time data (e.g. live thunderstorm is likely to cause major delay, the night Switzerland wins the '22 world cup there will likely be major delays on routes leading to public places)

Does not take into consideration service announcements