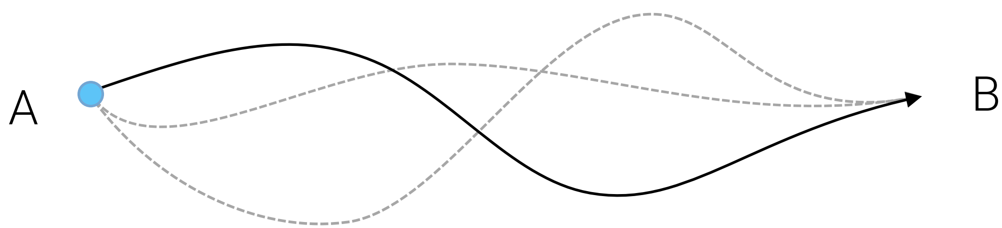
A Heuristic for Multi-modal Route Planning

Dominik Bucher, David Jonietz and Martin Raubal

LBS 2016, 15. November 2016

ETH zürich
geoinformation engineering ikg eth zurich

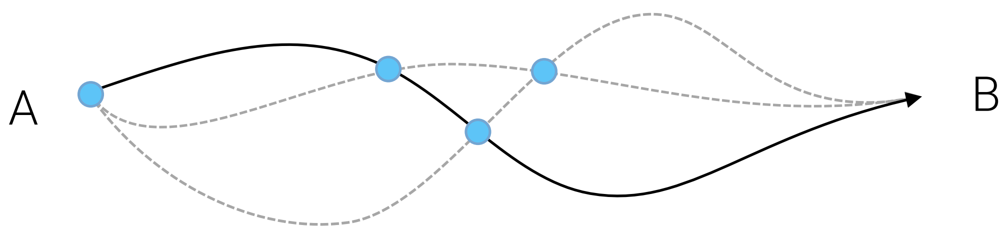
Introduction



A journey from A to B

Walk, Car, Bike,
Public Transport

Introduction



A journey from A to B

Walk, Car, Bike,
Public Transport

Carpooling? Carsharing? Bikesharing?
Ride-hailing? Bus-on-demand?
An arbitrary combination?
Personalization?

Multi-modal Route Planning

Difficulties

Car runs on Street Network

Every point is reachable

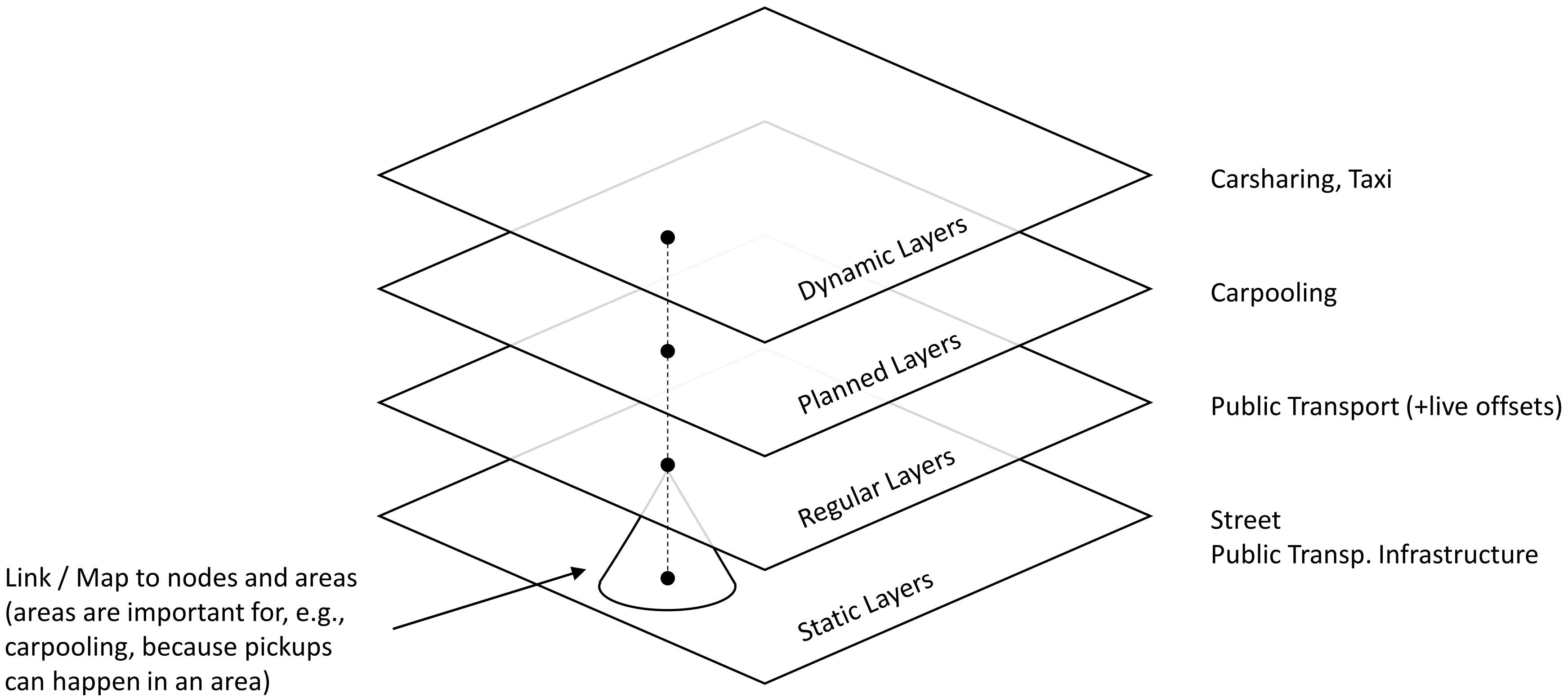
Public Transport runs between Stops

Only certain «transfer nodes» are reachable

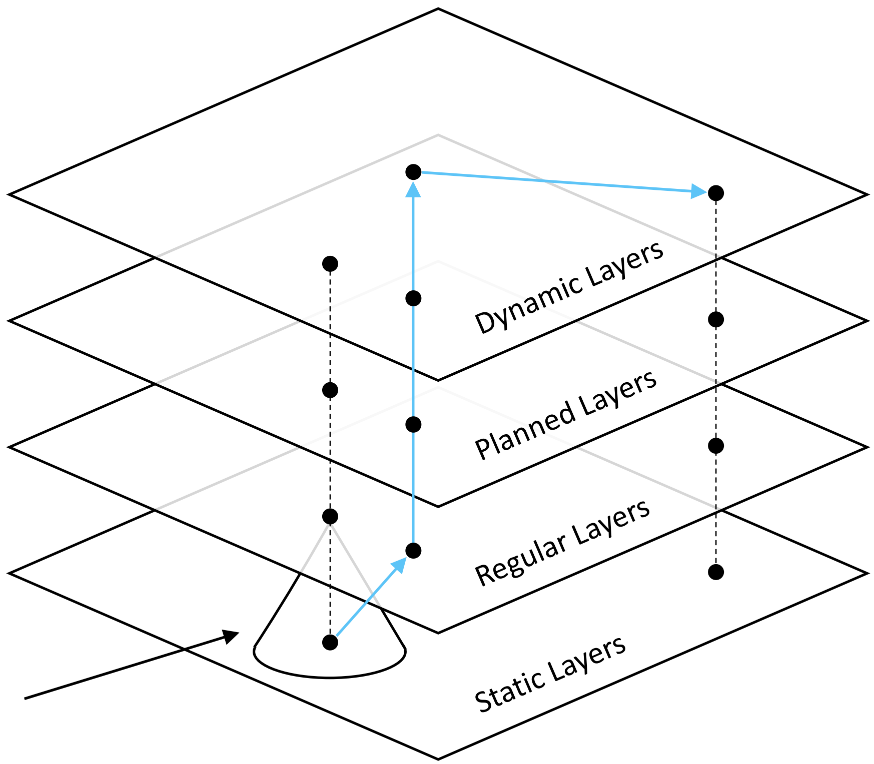
More Difficulties

- Richer user profiles lead to **computationally expensive** graphs and edge weights.
- **Weight coefficients** are not well-suited for representing hard restrictions.
- Traditionally, **all possible paths** taken into consideration by the routing algorithm.
- **System adaption** to new requirements difficult.

Transport Layers



Transport Layers



Carsharing, Taxi

Carpooling

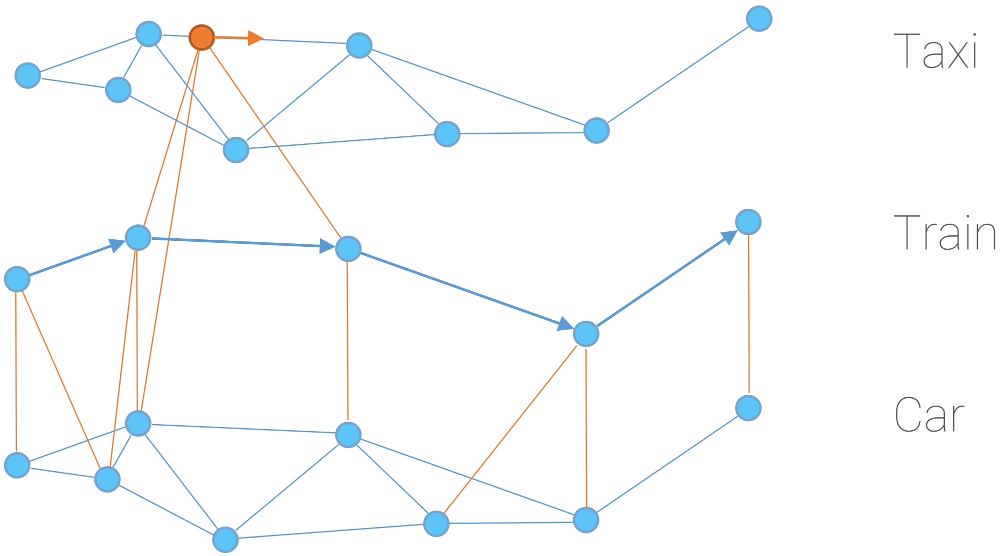
Public Transport (+live offsets)

Street
Public Transp. Infrastructure

Link / Map to nodes and areas (areas are important for, e.g., carpooling, because pickups can happen in an area)

Difficulties

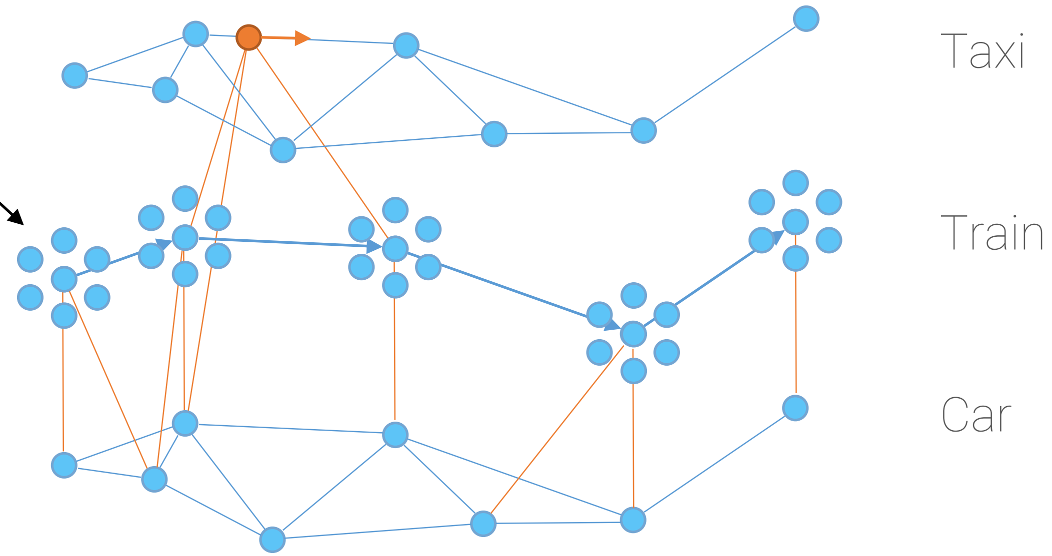
A large dynamic graph of heavily interconnected nodes



Difficulties

A large dynamic graph of heavily interconnected nodes

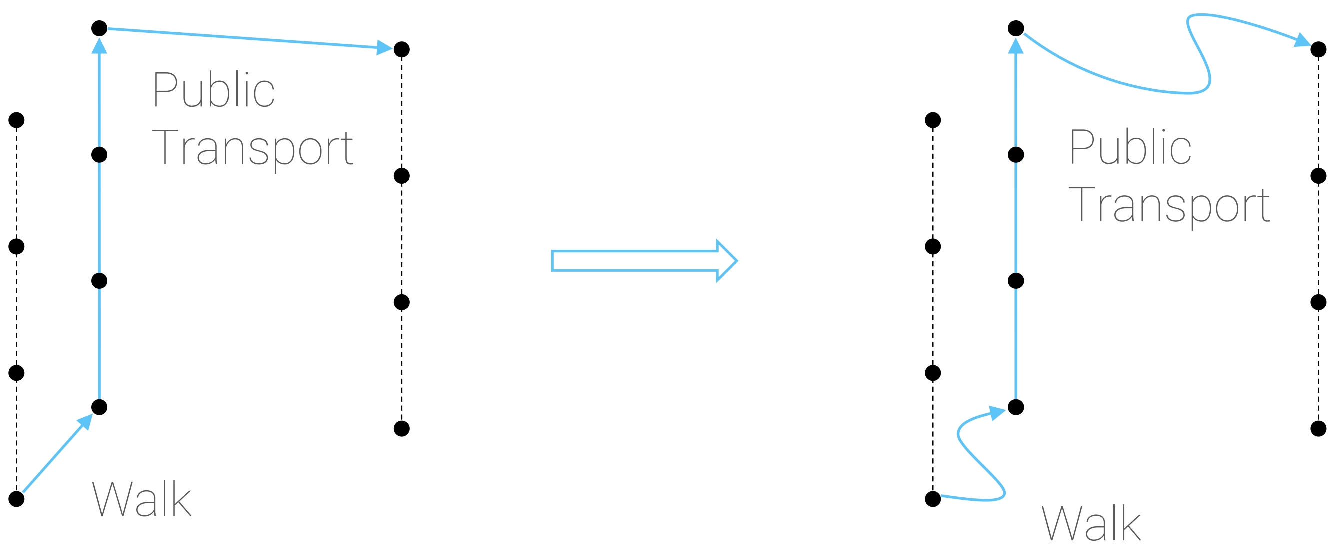
Time «unrolling» into a lot of states



Two Steps

1. Use dynamic «transfer graph» to compute possible multi-modal routes from A to B

2. Compute feasibility of trip legs using a conventional route planner for the respective mode



Rule Base for Transport Modes

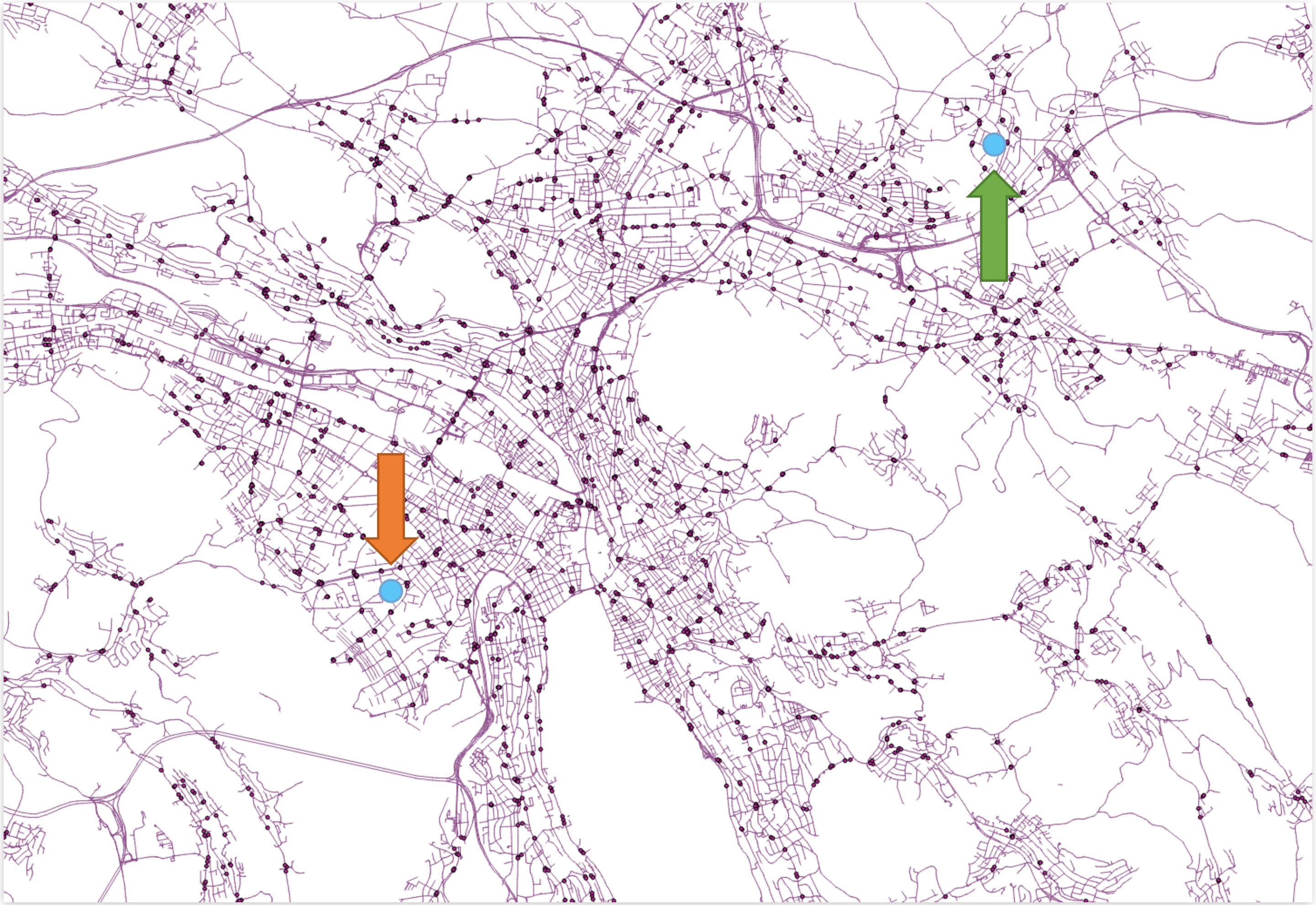
Rules (cf. agent planning)

O[condition] → M[condition] → D[condition]: [outcomes]

Example: Drive a Bike

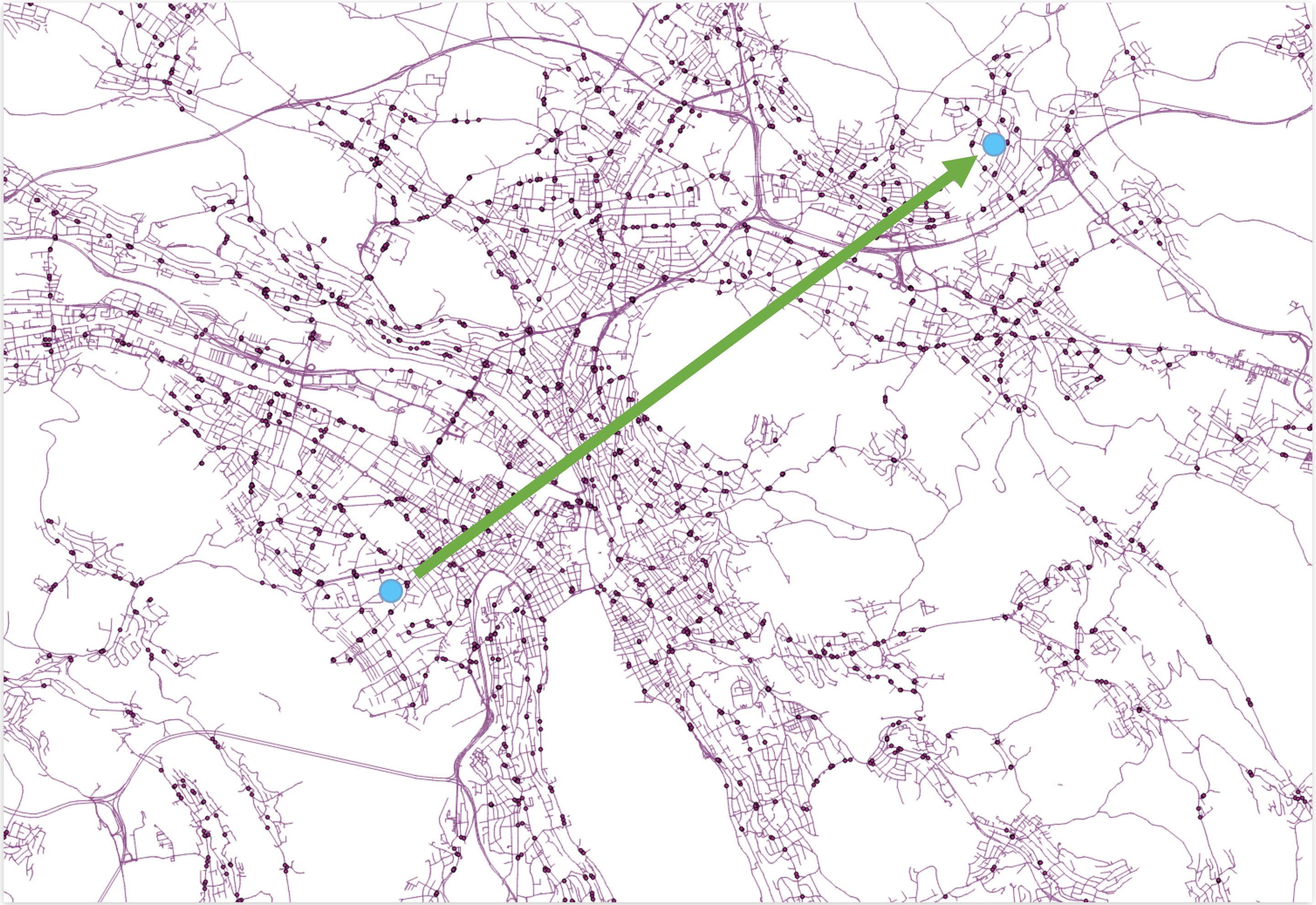
A[user[bikeLocation] = A] → BIKE[context[weather] != "rain"] → B[bikeParking = true]:
user[bikeLocation] = B, user[distBiked] += dist(A, B)

Algorithm



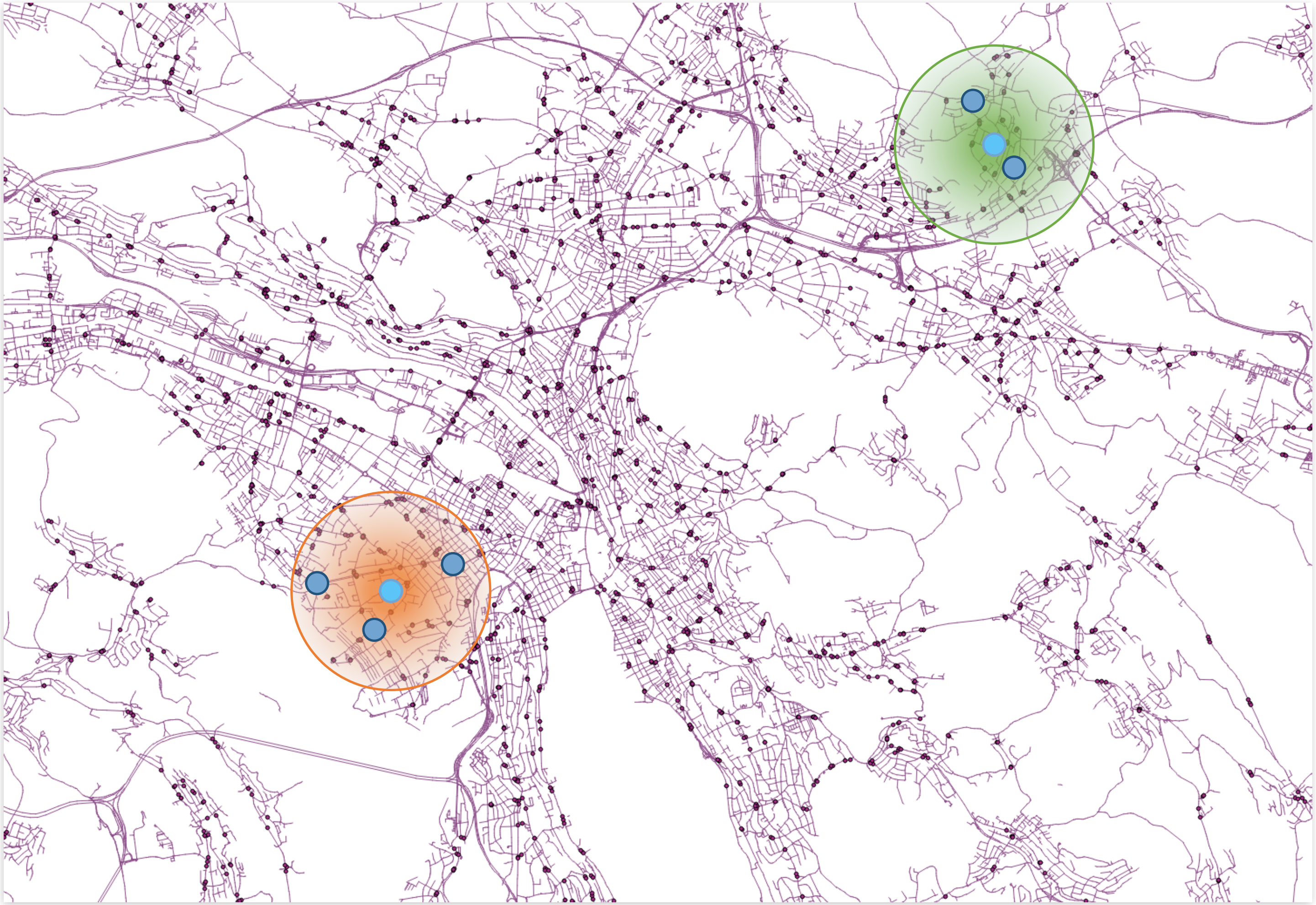
Algorithm

1. Directly Reachable?



Algorithm

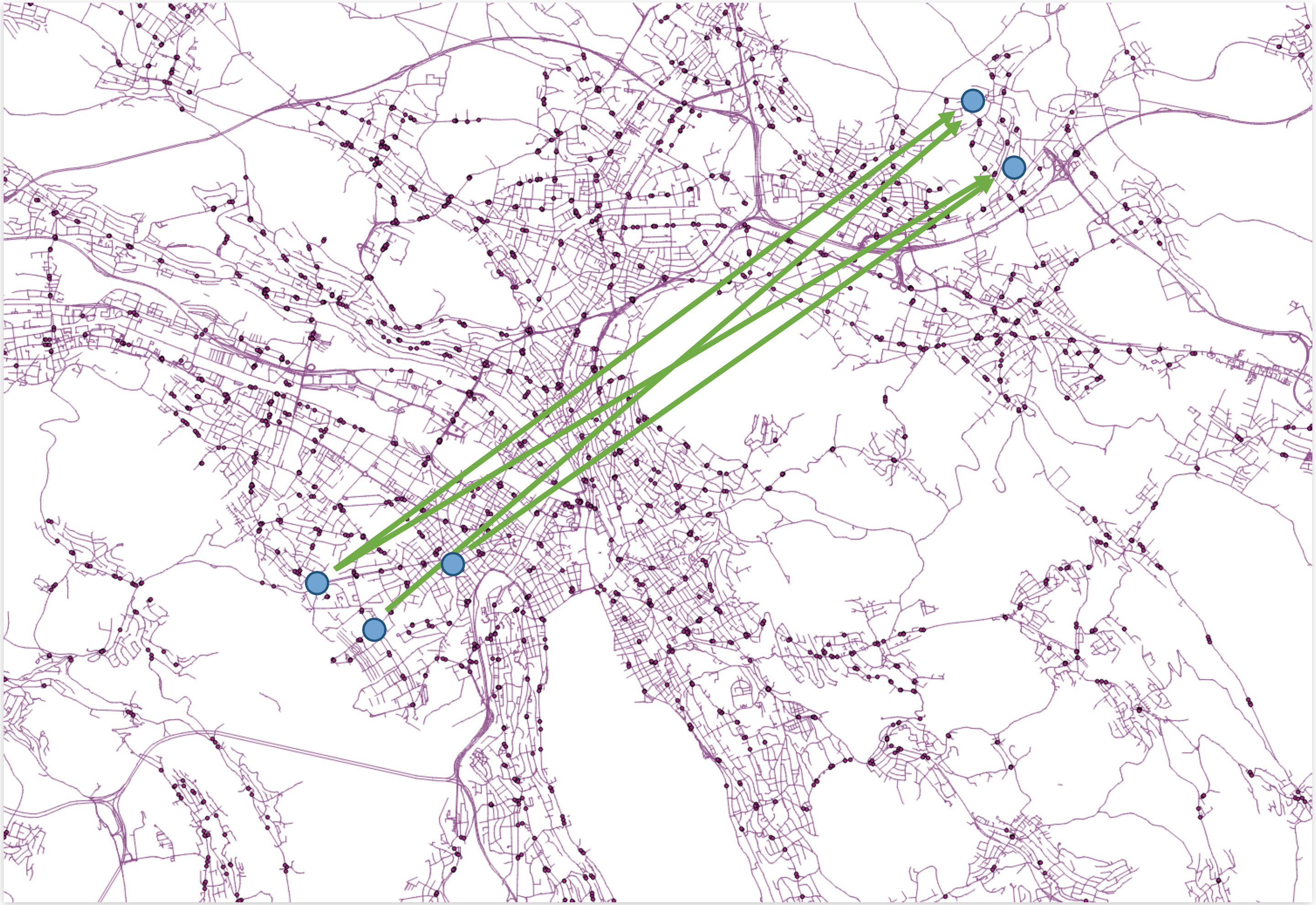
1. Directly Reachable?
2. Find Transfer Locations



Multi-modal Route Planning

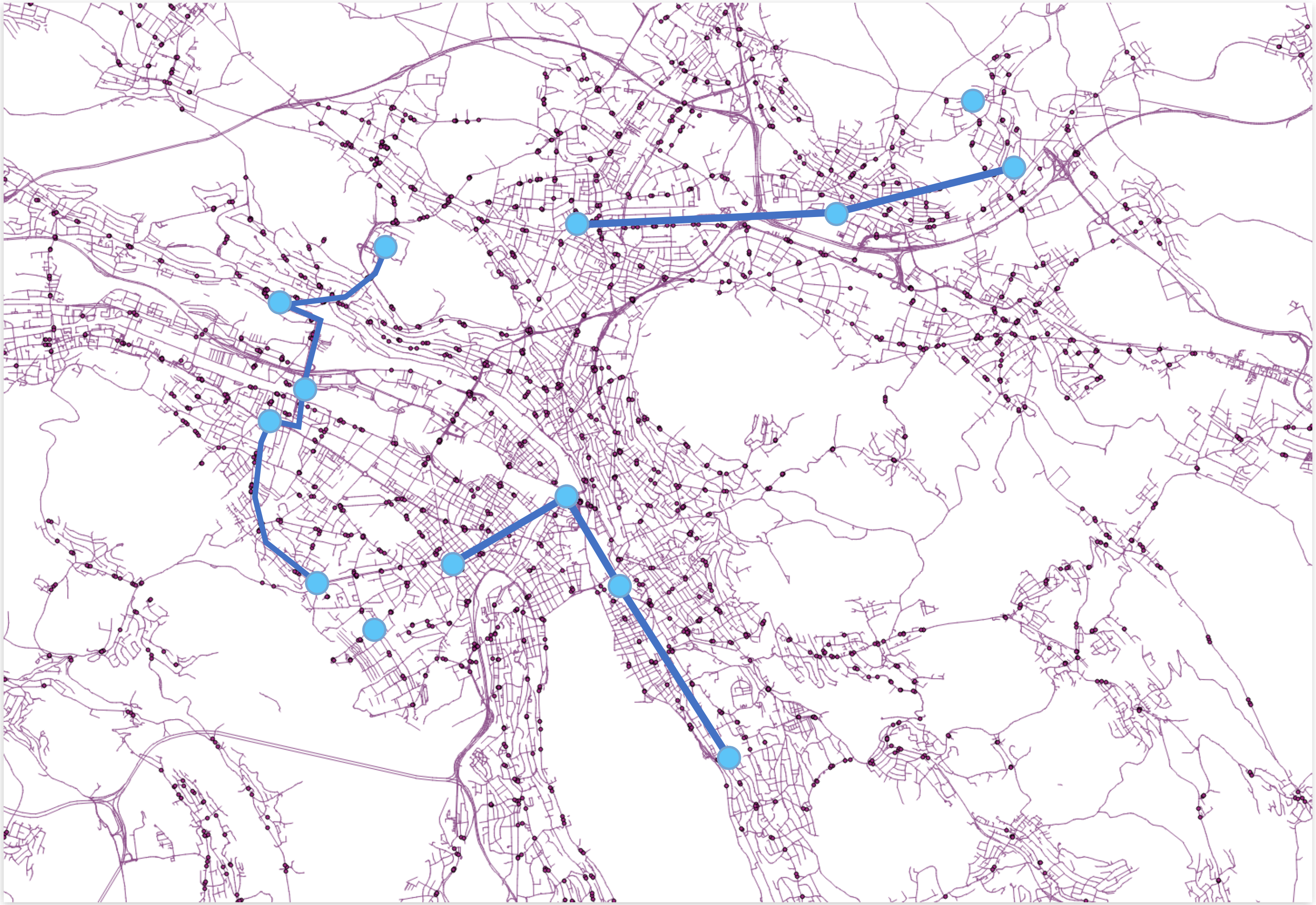
Algorithm

1. Directly Reachable?
2. Find Transfer Locations
3. Check Reachability



Algorithm

1. Directly Reachable?
2. Find Transfer Locations
3. Check Reachability
4. Expand

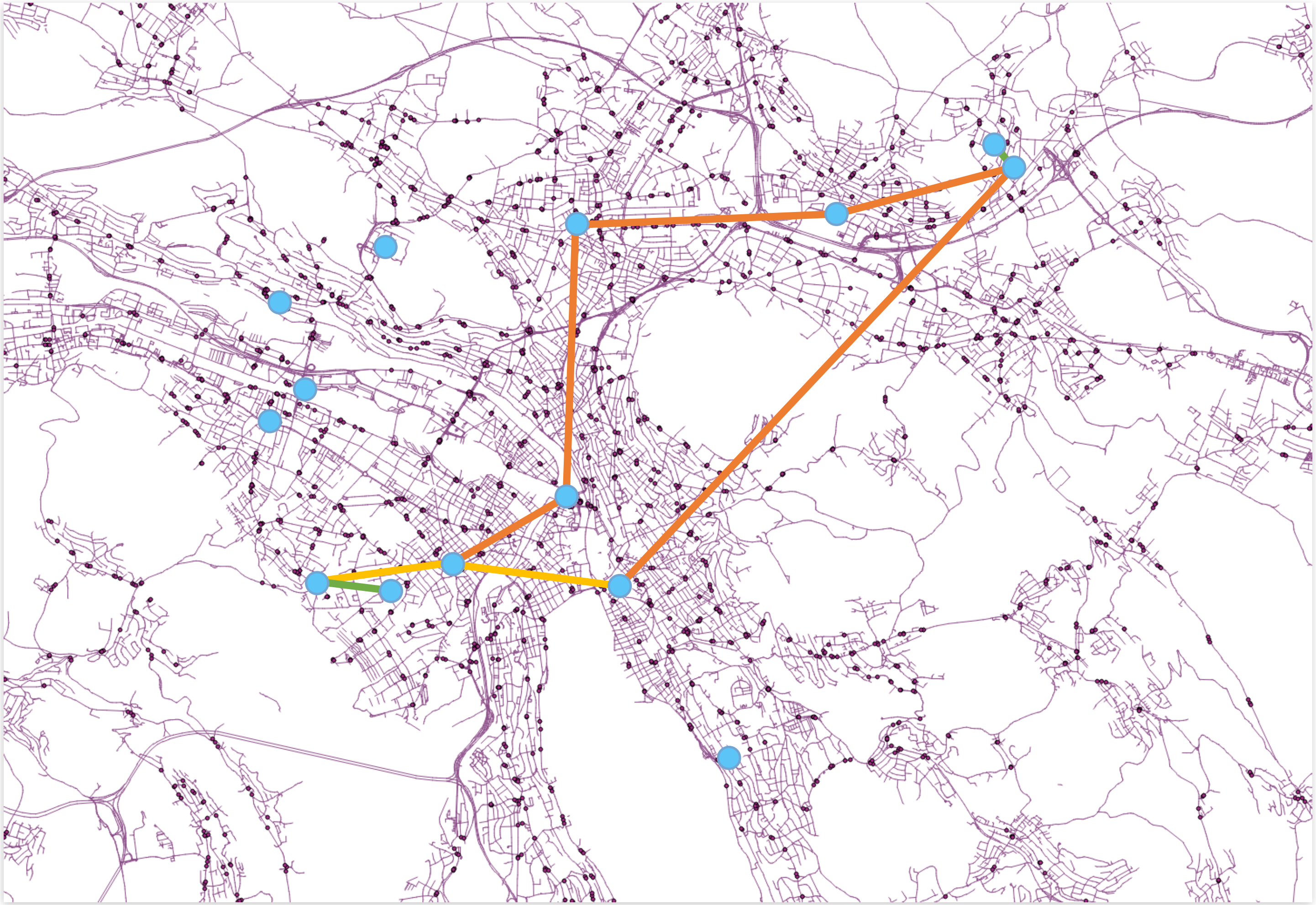


Algorithm

1. Directly Reachable?
2. Find Transfer Locations
3. Check Reachability
4. Expand
5. Check Reachability

Algorithm

1. Directly Reachable?
2. Find Transfer Locations
3. Check Reachability
4. Expand
5. Check Reachability
6. Unfold (e.g., using an energy model)



Implemented Rules

Rules for Walking, Biking, Driving, using Public Transport, Carsharing, Carpooling, Bikesharing

Integrating personalized constraints, contextual factors, mode-dependent restrictions

Table 1 A selection of rules implemented in our prototype system

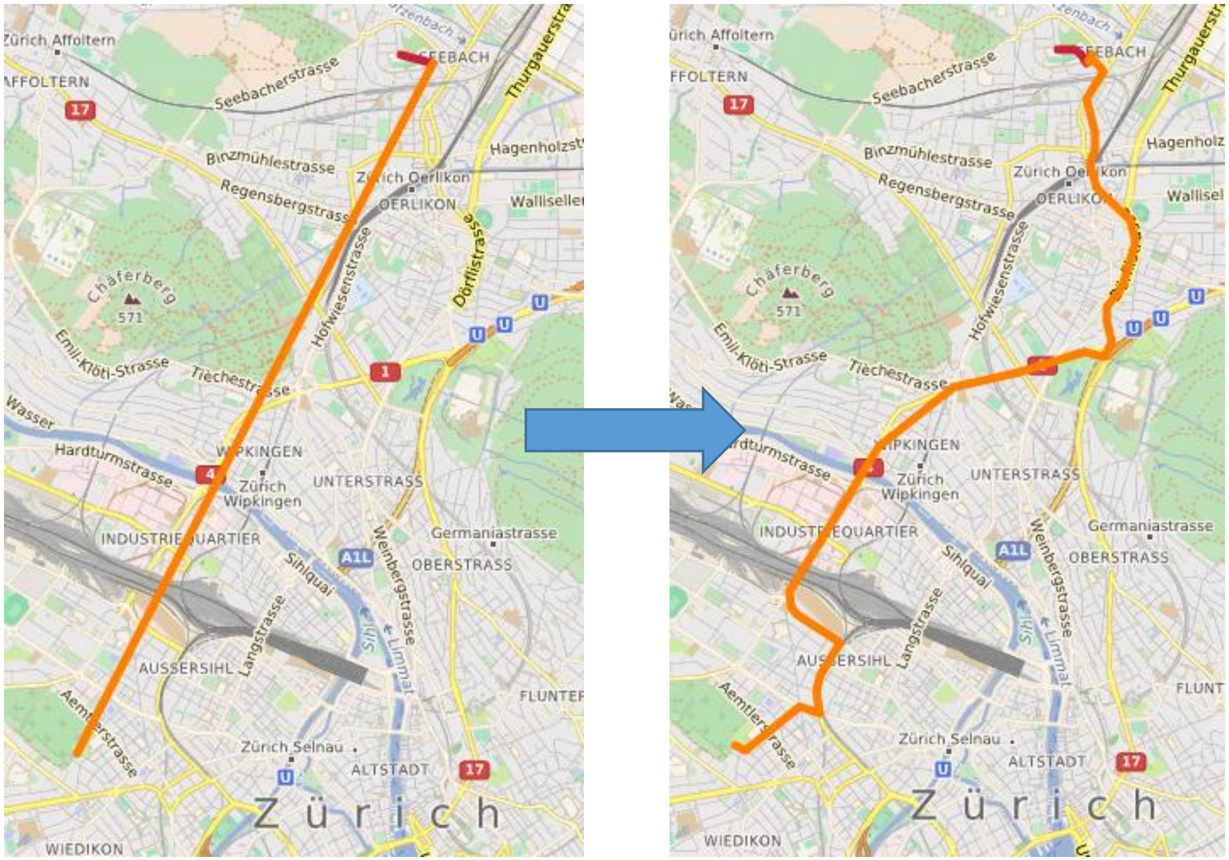
Mode	Rule	Description
Walk	$d_w = user[distWalked] + dist(A, B)$ $A[\emptyset] \rightarrow WALK[(d_w < user[maxDist])$ and (NOT $context[rainyWeather]$ OR $(d_w < user[maxDistRain]))$ and $(context[currentTime]$ IN $user[acceptableTimeIntvlWalk])]$ $\rightarrow B[\emptyset]$: $user[distWalked \mathrel{+}= dist(A, B)]$, $context[time \mathrel{+}= time(A, B)]$	Every node provides walking, however, a user can only walk up to a maximal distance (which gets decreased if it is raining), and if the current time is within an accepted time interval for walking. As a result of walking, the total walked distance is updated as well as the context
Bike	$A[user[bikeLocation] = A] \rightarrow BIKE[($NOT $context[rainyWeather])$ AND $(context[currentTime]$ IN $user[acceptableTimeIntvlWalk])] \rightarrow B[bikeParking = true]$: $user[bikeLocation] = B$, $user[distBiked] \mathrel{+}= dist(A, B)$, $context[time \mathrel{+}= time(A, B)]$	A user can only take the bike, if her bike currently is at the location. Further, the destination needs to have a bike parking spot available. Concerning contextual variables similar to walking
Car	$A[user[carLocation] = A] \rightarrow CAR[\phi] \rightarrow B[\#parkingSpots > 0]$: $user[carLocation] = B$, $context[time \mathrel{+}= time(A, B)]$	Taking the car is only possible from the location where the user currently has parked her car to locations with a parking spot available. As a result, the car is at location *B*
Train	$A[connectsLineX = true] \rightarrow TRAIN[\phi] \rightarrow B[connectsLineX = true]$: $[\phi]$, $context[time \mathrel{+}= time(A, B)]$	Similar to bus
Tram	$A[connectsLineX = true] \rightarrow TRAM[\phi] \rightarrow B[connectsLineX = true]$: $[\phi]$, $context[time \mathrel{+}= time(A, B)]$	Similar to bus
Carshare	$A[carSharing = true, \#cars > 0] \rightarrow CARSHARE[\phi] \rightarrow B[\#parkingSpots > 0]$: $A[\#cars -= 1]$, $context[time \mathrel{+}= time(A, B)]$	Carsharing is possible from carsharing locations, where enough cars are available. The destination needs to have free parking spots
Carpool	$A[intersects(A, C) = true] \rightarrow CARPOOL[\phi] \rightarrow A[intersects(B, D) = true]$: $[\phi]$	Carpooling is possible from locations that intersect with a spatio-temporal corridor of a carpooler
Bike-Share	$A[bikeSharing = true, \#bikes > 0] \rightarrow BIKESHARE[context[weather] \mathrel{!}= \text{"rain"}] \rightarrow B[bikeParking = true]$: $A[\#bikes -= 1]$, $user[distBiked] \mathrel{+}= dist(A, B)$, $context[time \mathrel{+}= time(A, B)]$	Bikesharing is possible from bikesharing locations, where enough bikes are available

Benefits

1. The heuristic only has to consider a graph consisting of transfer nodes (mostly public transport stops).
2. For validation of individual route segments, only the respective sub-graphs have to be queried.

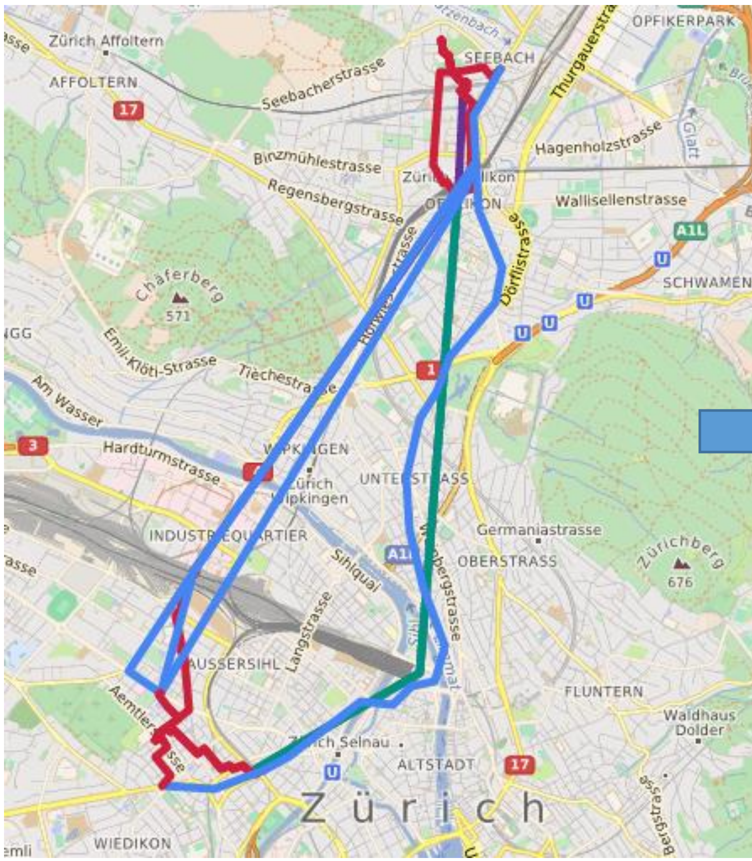
Example Outputs

Raw Output

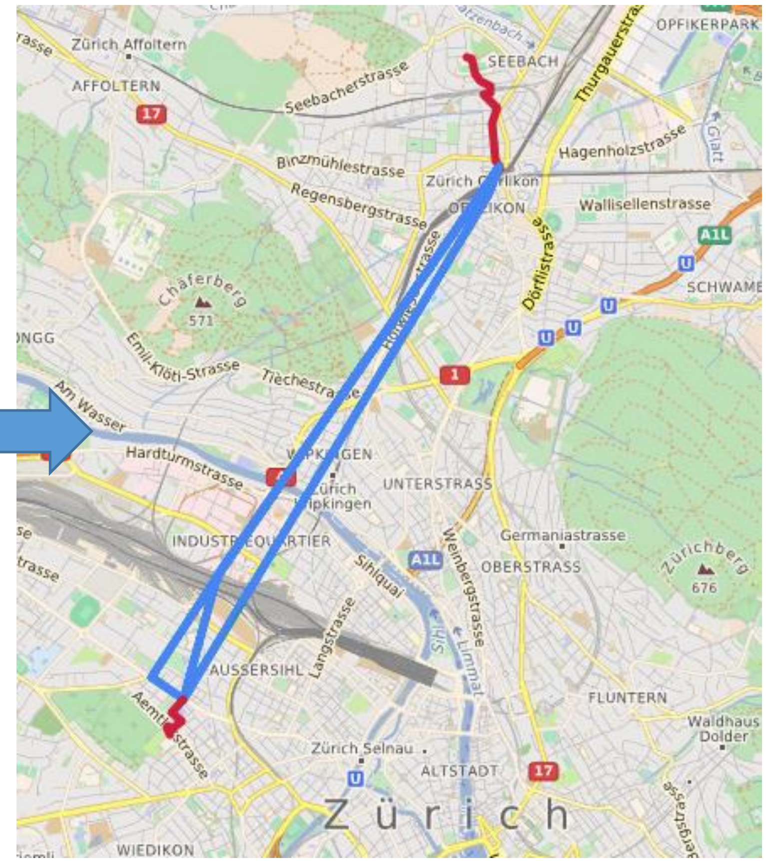


Planned Route

Example Outputs



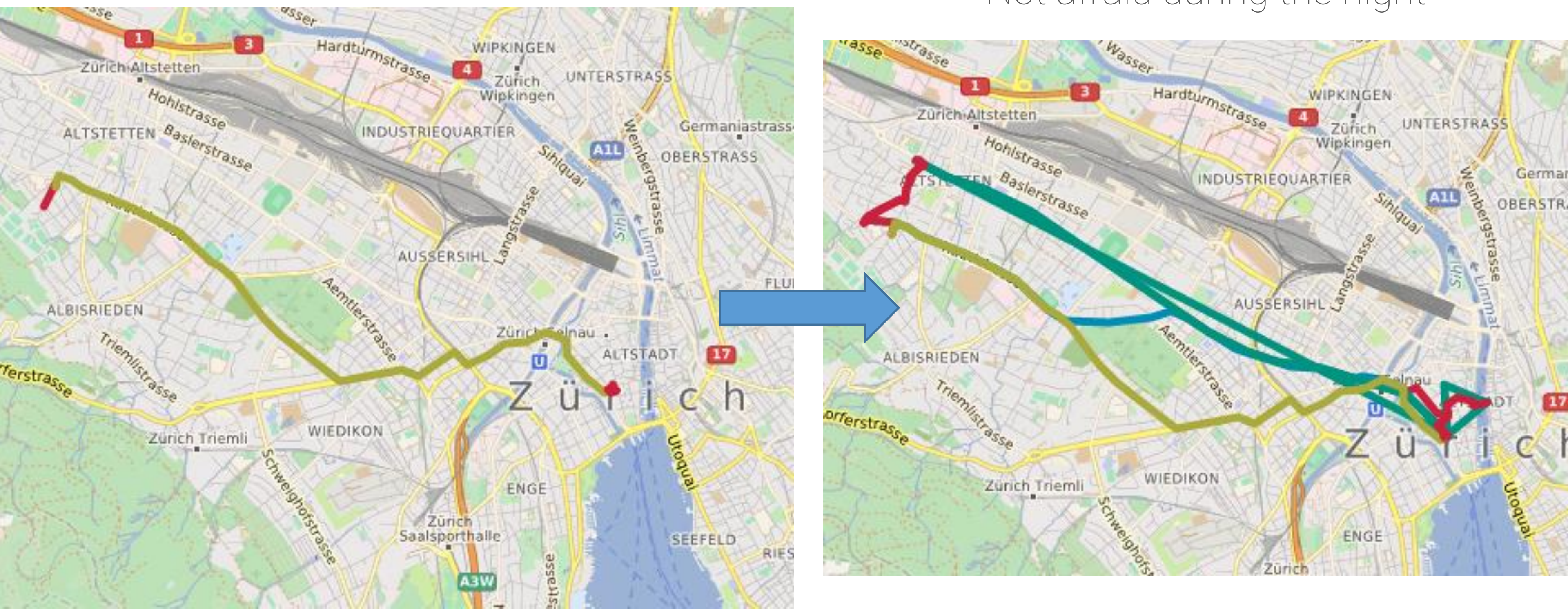
Normal Weather



Rainy Weather

Example Outputs

Not afraid during the night



Afraid during the night

Conclusions

We can bulid a dynamic graph «on the fly»,
and use it to compute multi-modal route options.

We can integrate personalized preferences,
and quickly update the respective rules.

The heuristic can be expanded to include for
example points of interest, as intermediate stops.

Questions

Multi-Modal Routing