Probabilistic Turn-By-Turn Directions for Public Transport[1]

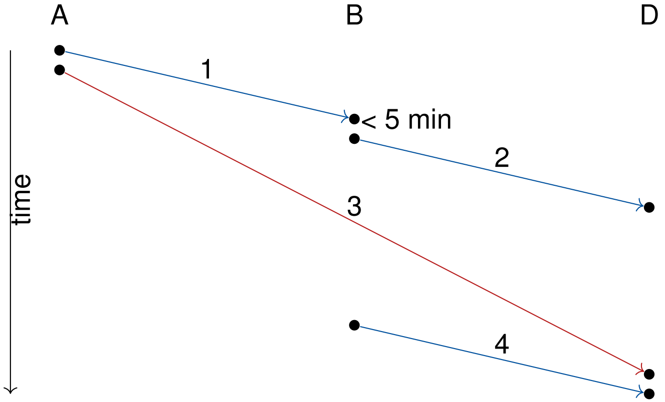
[1]Based on Durner, Robin. "Stochastic strategies for public transport journeys based on realtime delay predictions." University of Stuttgart, 2024. `http://dx.doi.org/10.18419/opus-14802`

Only Showing the Next Step

Just like when going by car, no need to provide the user with an entire, fixed journey

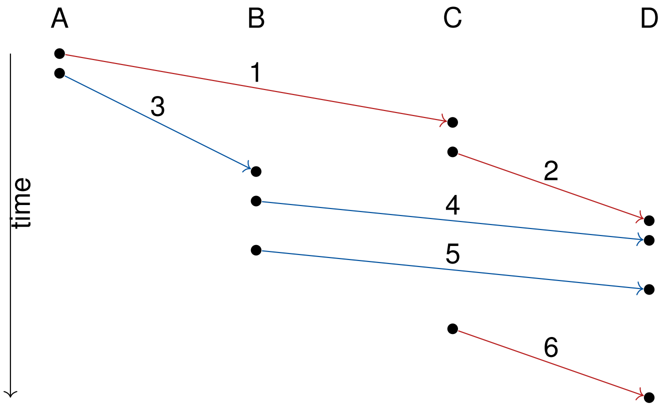
Sinnbild Autobahnausfahrt, StVO 1990. Source: Zehnte Verordnung zur Änderung der Straßenverkehrs-Ordnung. In: Bundesgesetzblatt, 1989, Teil 1, Nr. 52, S. 1976–1979; hier: S. 1978. `https://commons.wikimedia.org/wiki/File:Sinnbild_Autobahnausfahrt,_StVO_1990.svg`

Risk-hedged Transfers



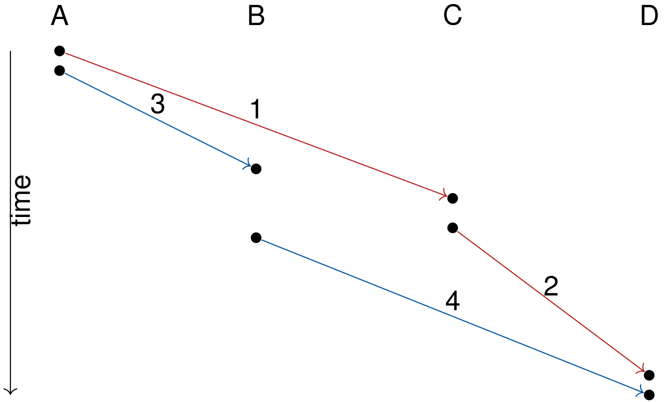
Trying to catch tight connection 2 may result in much earlier arrival in the best case and only slightly later arrival in the worst case compared to connection 3

Backup Transfers



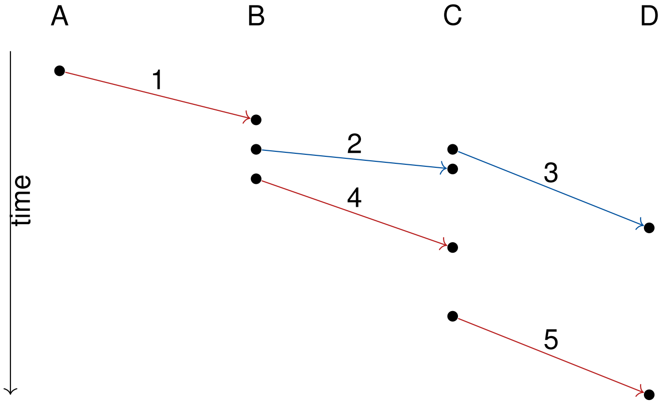
Initially taking connection 3, because it has a better backup connecting service, even though in the best case, one arrives slightly later

More Reliable Transfers



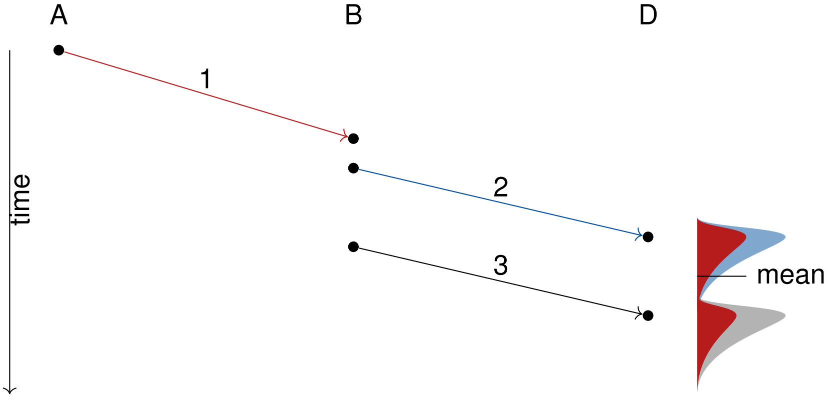
Taking connection 3 and 4 with a more reliable transfer, even though they arrive slightly later at destination

(Negative) Adaptive Transfers



Trying to take connection 2 instead of 4, if possible, in order to have a chance to catch connection 3, and in order to increase reliability of transfers in the future

Weighted Averages of Destination Arrival Distributions

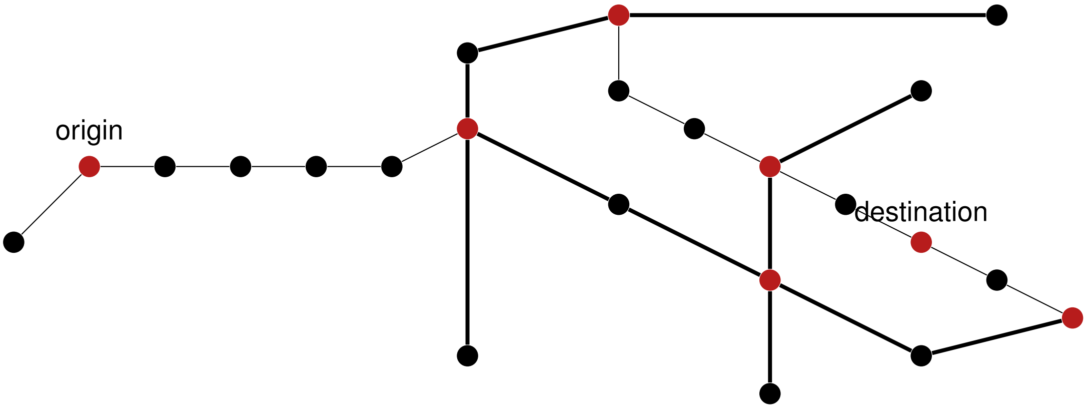


Destination arrival distributions of connections 2 and 3 are weighted to give the distribution for connection 1. The mean is not a good intuitive representation of the distribution.

Approach

- no upfront, fixed journey plan
- full country- or planet-scale timetable
- earliest mean destination arrival
- satisfy scenarios from motivation
- calculate entire probability distributions of destination arrival
- arrival and departure distributions based on non-stochastic realtime delay predictions and historical data
- cancellation probabilities
- push destination arrival distributions through the network

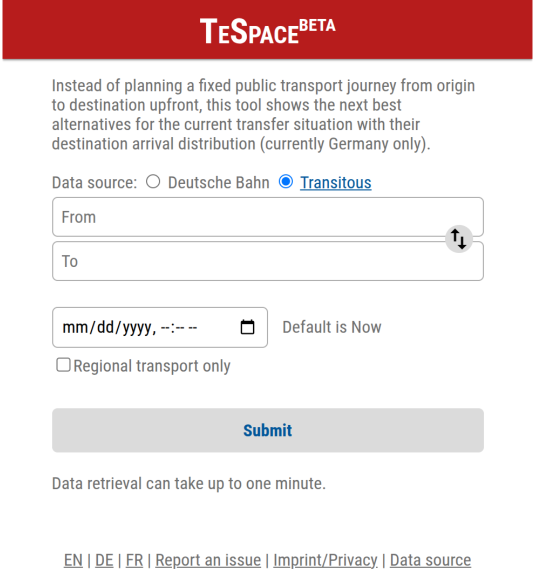
The Heuristic of Relevant Stops



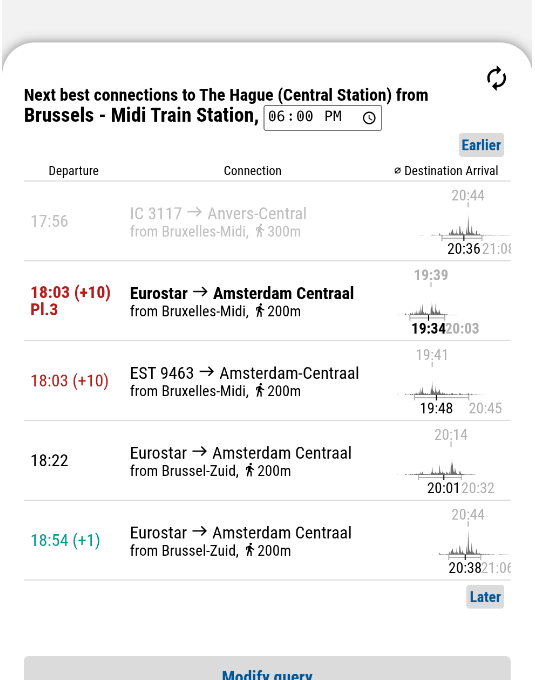
For a given query from origin to destination, which stops are relevant for transferring?

Demo

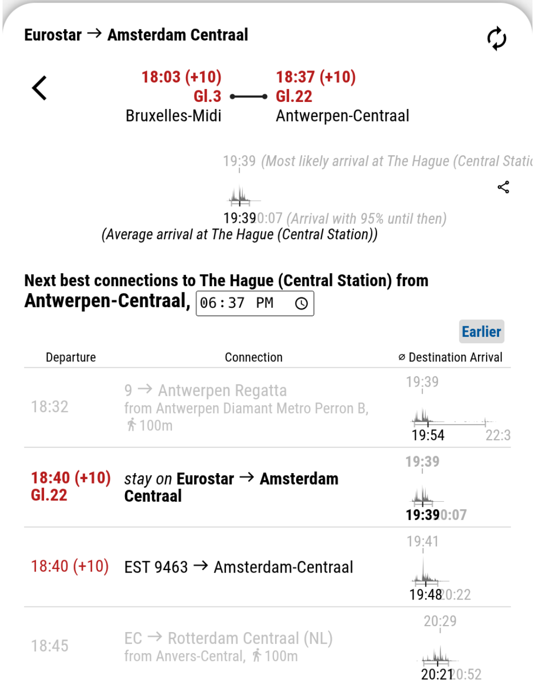
`https://tespace.traines.eu`



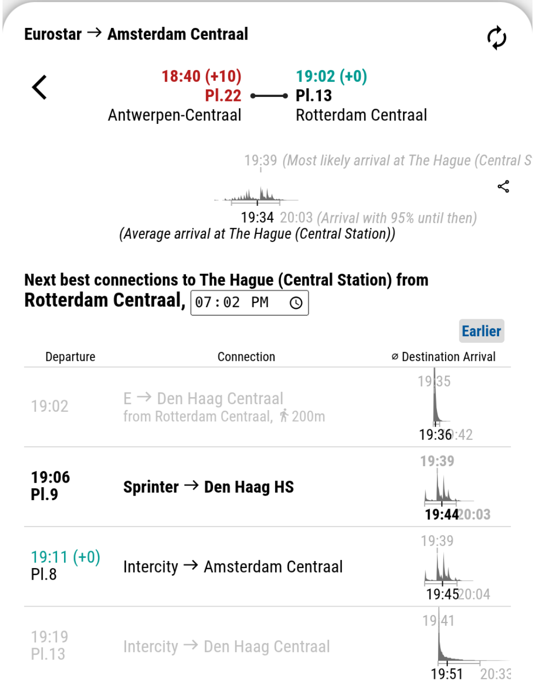
Starting just like a normal journey planner search with origin and destination, e.g. from Brussels Midi to The Hague Central. Let's use the global timetable data from Transitous (`https://transitous.org`), employing the relevant stops heuristic.



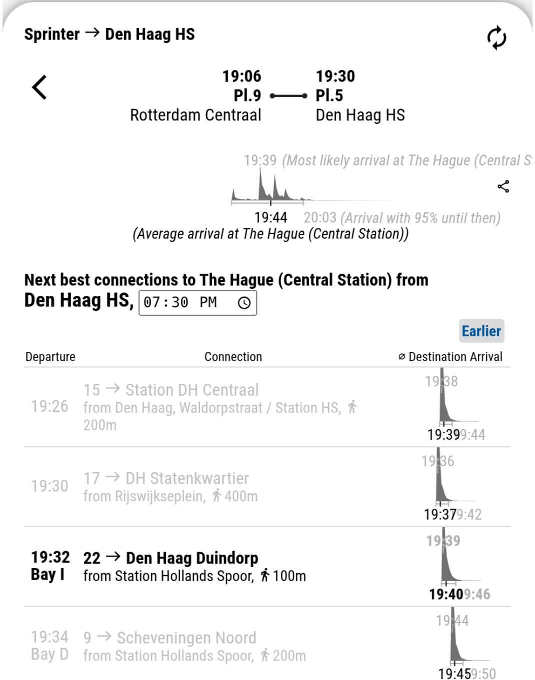
The first step: Taking the next best train departing from Brussels Midi in direction of The Hague, without needing to know the steps down the line.



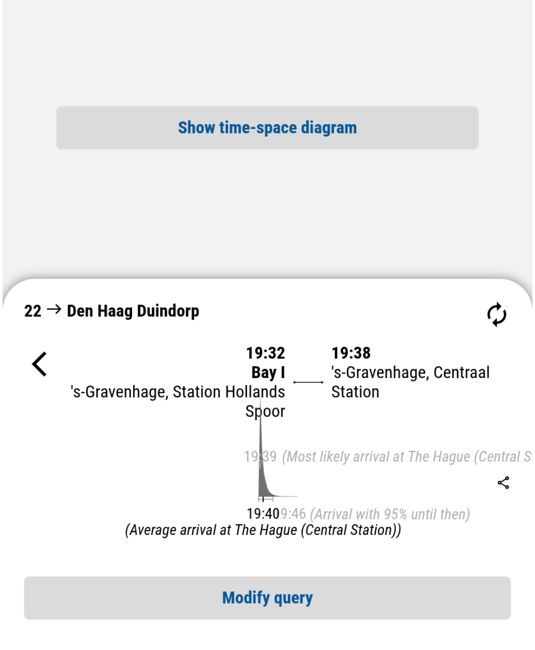
Let's take the 18:03 Eurostar and select it. Next relevant stop is Antwerp. We're always given multiple alternatives for how to continue. For each connection, average destination arrival time, most likely arrival time, and 95% arrival time are displayed together with the histogram of the destination arrival distribution of the user at The Hague Central.



Let's stay on the Eurostar. Next relevant stop is Rotterdam at 19:02. There, we could be lucky and catch a train departing also at 19:02 going directly to The Hague Central. Or, the more likely option, we take the Sprinter at 19:06. We have backups with the Intercitys at 19:11 and 19:19.

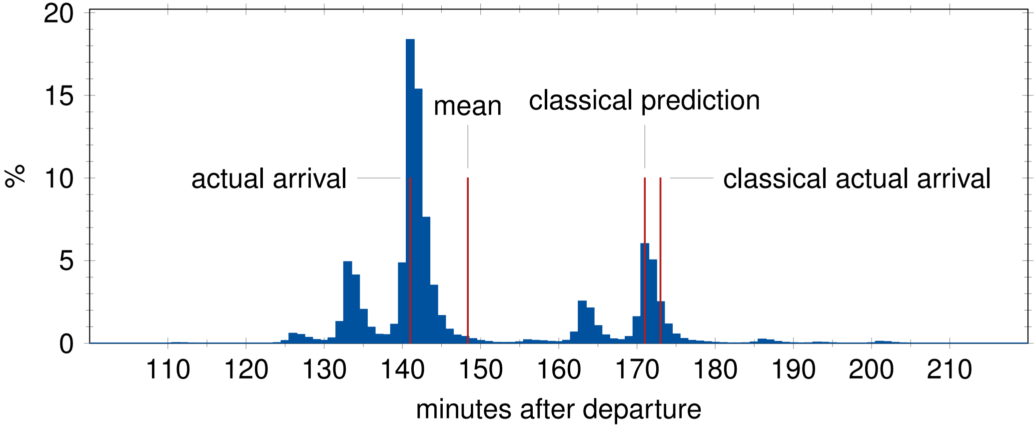


Let's board the Sprinter. We'll need another transfer at The Hague HS. We can just take one of the buses or trams leaving next.



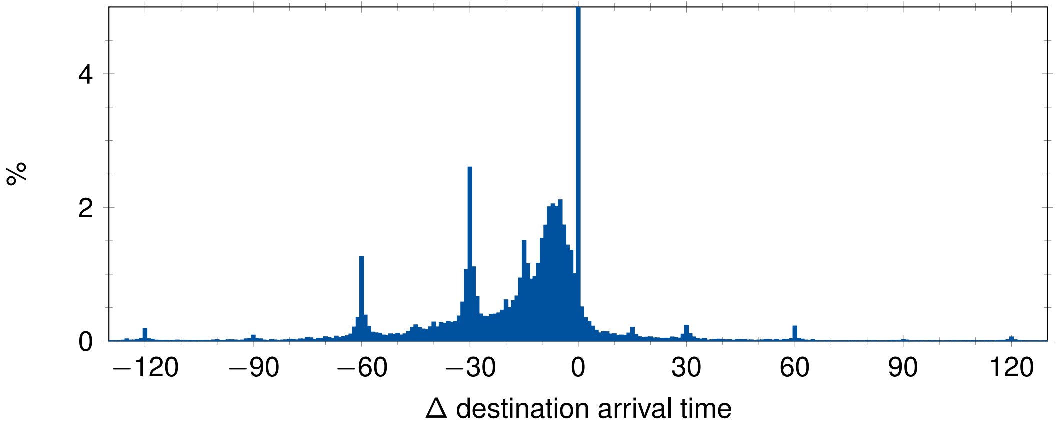
We are about to arrive at The Hague Central.

A Typical Destination Arrival Distribution



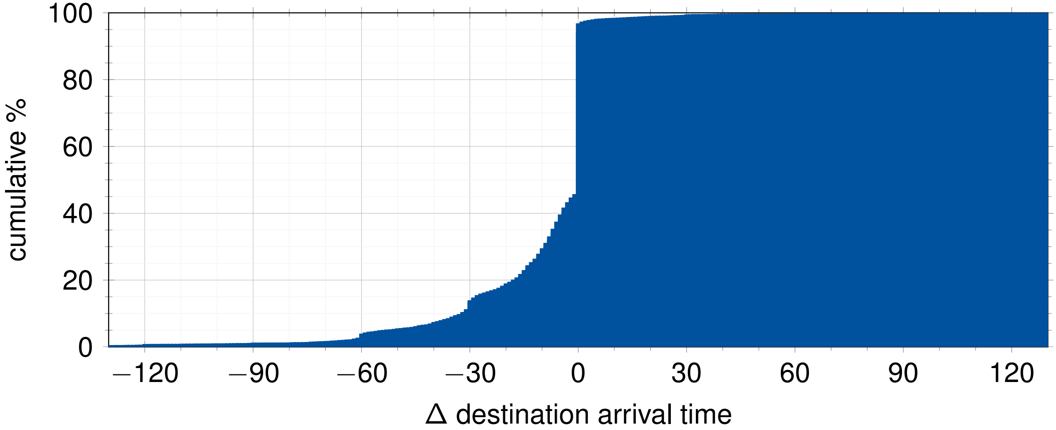
The mean does not intuitively represent the destination arrival distribution

Change in Actual Arrival Compared to Classical Algorithm



Case Switzerland: On average, a user arrives about 9 minutes or about 5% earlier when using our algorithm, assuming short transfers

Change in Actual Arrival Compared to Classical Algorithm



Case Switzerland: In about 15% of cases, they arrive 30 minutes or more earlier

Conclusion

- it pays off to refresh the journey plan and consider alternatives while travelling
- helpful for “nimble” users in more delay-prone environments
- depends on ticketing, but in many cases, a certain flexibility is actually given (incidents, relation tickets, regional tickets, GAs, Interrail tickets...)
- for tickets limited to specific trains: calculate more reliable, fixed transfers without alternatives (completely different question)
- integrate directly into MOTIS[2] and thereby Transitous[3]?
- additional criteria for itinerary quality/relevance, probabilistic or not?

[2] https://github.com/motis-project/motis
[3] https://transitous.org/

Backup

Terminology

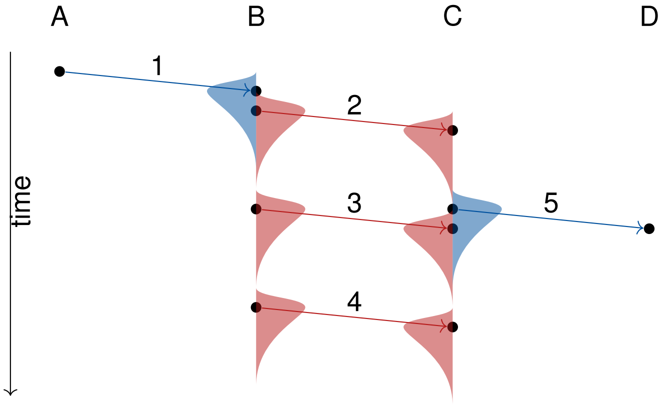
- timetable $(\Pi, \mathcal{S}, \mathcal{C}, \mathcal{T}, \mathcal{W})$
- Π period of operation = time
- $\mathcal{S}$ stops
- $\mathcal{T}$ trips (sequence of stops of a vehicle at a certain time)
- $\mathcal{C}$ connections (pair of stops in a trip)
- $\mathcal{W}$ walking connections between stops (instead: station contraction)
- a query from $s_{\text{source}} \in \mathcal{S}$ to $s_{\text{dest}} \in \mathcal{S}$ at time $t_{\text{source}} \in \Pi$

Recursive Definition

$$p_{T_{\text{dest}}(c)}(t) = \frac{1}{\mathbb{P}_{\text{feasible}}(c)} \sum_{i=0}^{|C_{\text{cont}}(c)|-1} \mathbb{P}_{\text{success}}(c, c_i) \cdot \left(\prod_{j=0}^{i-1} (1 - \mathbb{P}_{\text{success}}(c, c_j)) \right) \cdot p_{T_{\text{dest}}(c_i)}(t)$$

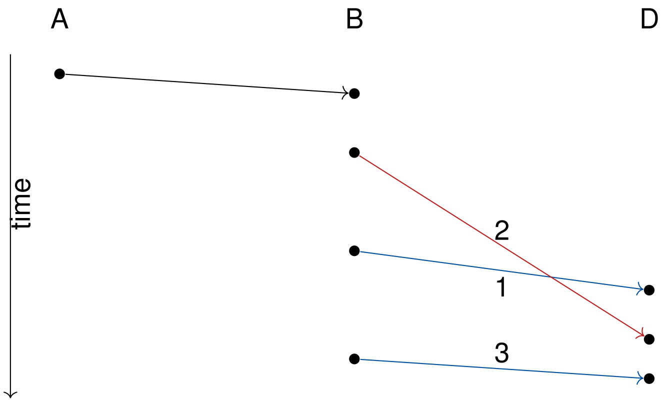
with the continuing connections $c_i \in \mathcal{C}$ ordered ascendingly by mean destination arrival $\mathbb{E}[T_{\text{dest}}(c_i)]$
and $\mathbb{P}_{\text{success}}(c, c_i) := \mathbb{P}_{\text{reachable}}(c, c_i) \cdot \mathbb{P}_{\text{feasible}}(c_i) \cdot \mathbb{P}_{\text{after}}(c, c_i)$

$\mathbb{P}_{\text{reachable}}$ and $\mathbb{P}_{\text{feasible}}$



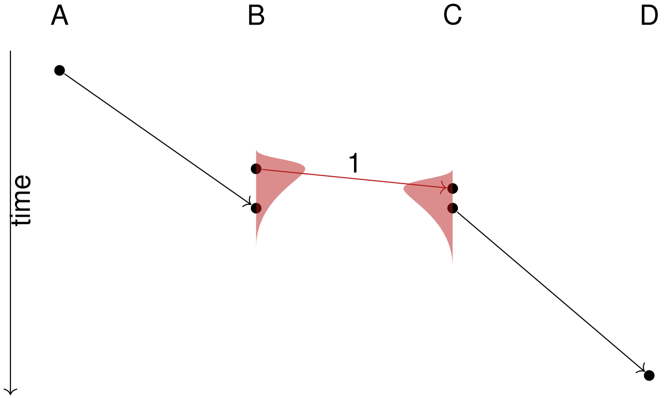
Between connection 1 and 3, the probability of reachability is high, but it is most likely infeasible to arrive at D using connection 3

$\mathbb{P}_{\text{after}}$, Transfer Strategy and Optimality



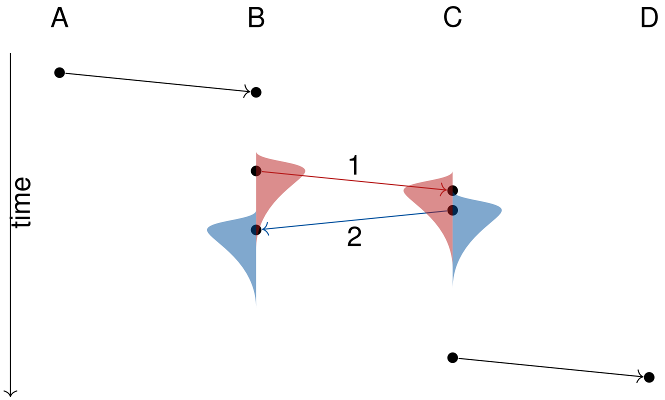
How should connection 2 be weighted?

Time Travel



According to the departure and arrival distributions, connection 1 might arrive at C before it departs from B

Departure and Arrival Distributions Leading to Cycles



Both connection 2 is reachable from connection 1 and connection 1 from connection 2 (albeit with a negative transfer time)

Basic Algorithm

CSA (MEAT)[4] iterates over all connections in descending order of their departure. Instead:

1. Use a depth-first search to find and cut the cycles and construct a topological ordering (preprocessing).
2. Iterate over the topologically ordered connections and push through the destination arrival distributions (query).

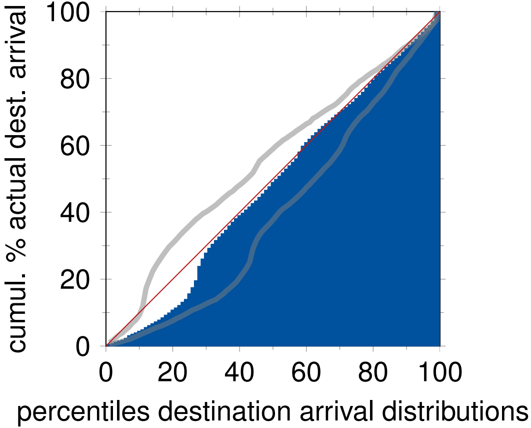
[4] J. Dibbelt, T. Pajor, B. Strasser, D. Wagner. “Connection scan algorithm”. In: Journal of Experimental Algorithmics (JEA) 23 (2018), pp. 1–56. `https://doi.org/10.1145/3274661`.

Runtimes

Mean Only	Relvt. Only	Fuzzy	Epsilon	#	Preproc.		First Query		Query	
					Avg	Max	Avg	Max	Avg	Max
●	○	○	○	100	257e3	353e3	-	-	3939	12191
●	○	○	●	1000	31297	41469	-	-	**534**	33792
●	○	●	●	1000	34475	46306	-	-	849	4291
●	●	○	●	10000	32460	43728	1067	18842	147	30829
●	●	●	●	1000	34867	61043	858	6928	141	6749
○	○	●	●	100	34740	55505	-	-	**5269**	38274
○	●	●	●	1000	32825	44040	1165	10378	**465**	11286

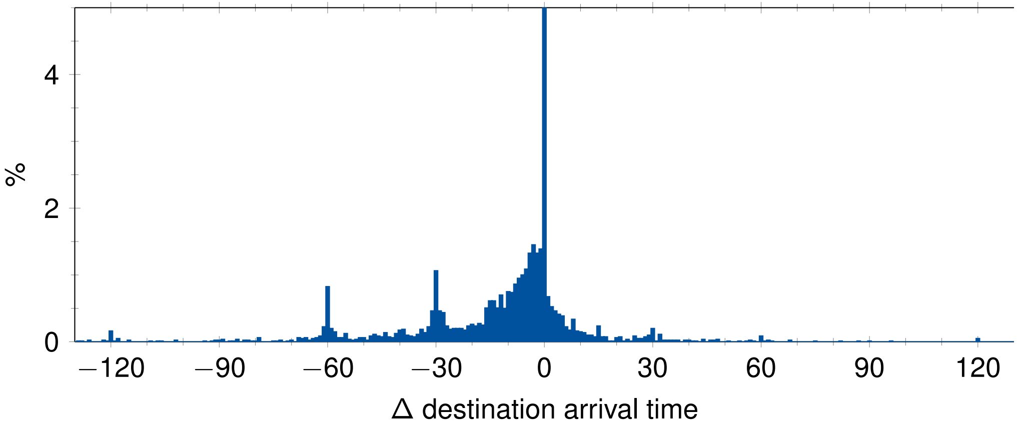
Runtimes in milliseconds on two days of the complete timetable of Switzerland

Accuracy of Calculated Destination Arrival Distributions



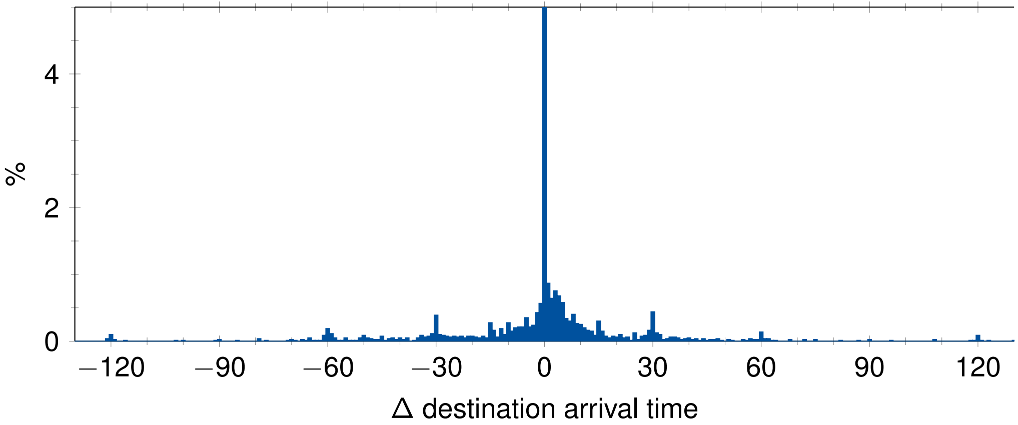
The calculated distributions are very accurate for later arrivals

Change in Actual Arrival with Long Transfers



With long transfers, compared to an offline algorithm, the gains decrease to about 4 minutes on average

Change in Actual Arrival Compared to Online Algorithm



With long transfers and compared to an online algorithm, the gains become negligible