Delay-Robustness of Transfer Patterns in Public Transportation Route Planning

ATMOS 2013, Sophia Antipolis

Hannah Bast, Jonas Sternisko, Sabine Storandt

Department of Computer Science
Albert-Ludwigs-University Freiburg

September 5, 2013

13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization and Systems

Transfer pattern of a route

$:=$

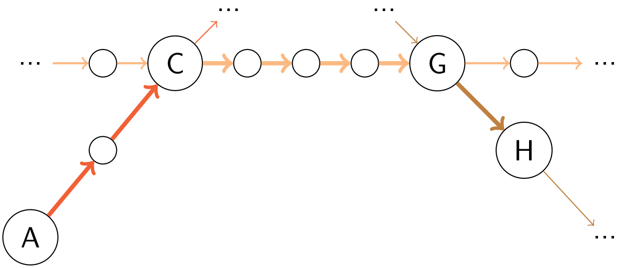
Sequence of stations where vehicle is changed

Transfer pattern of a route

:=

Sequence of stations where vehicle is changed

Example: Route A $\rightarrow$ H has transfer pattern ACGH.



Transfer Pattern Routing

Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(1) Construct query graph from patterns

Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

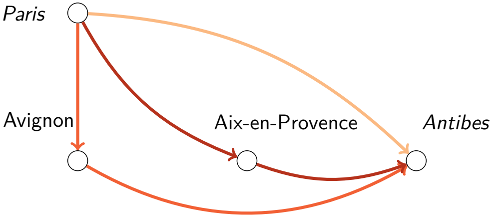
(1) Construct query graph from patterns

Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(1) Construct query graph from patterns

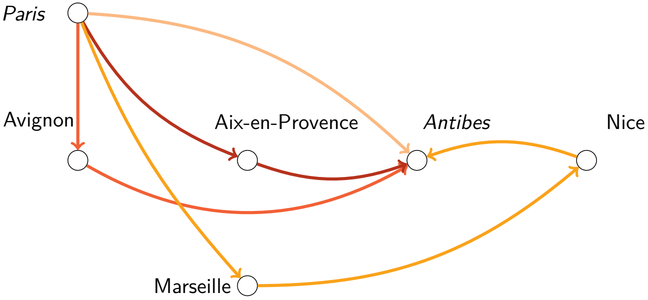


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(1) Construct query graph from patterns

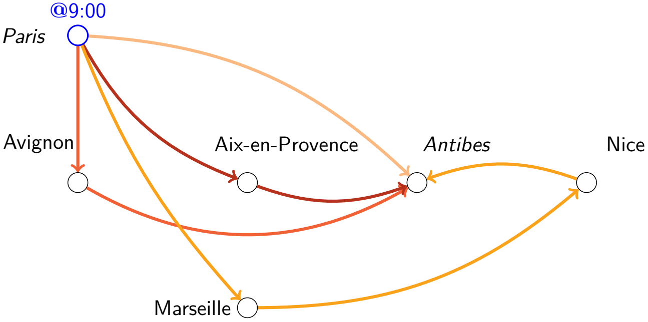


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

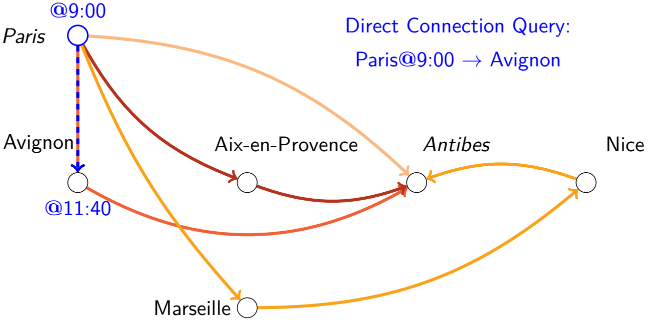


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

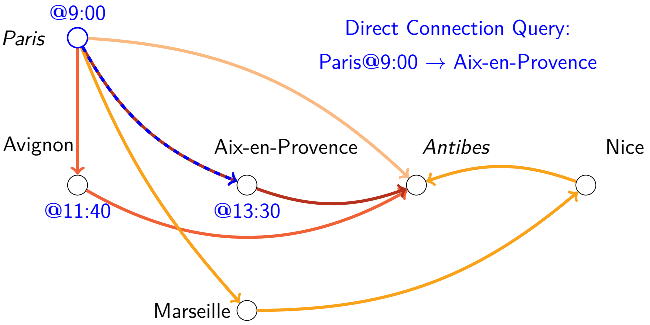


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

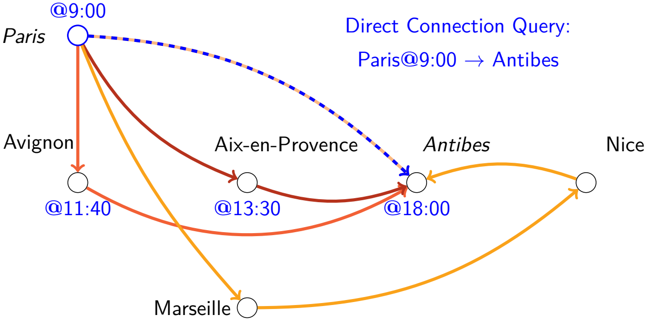


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

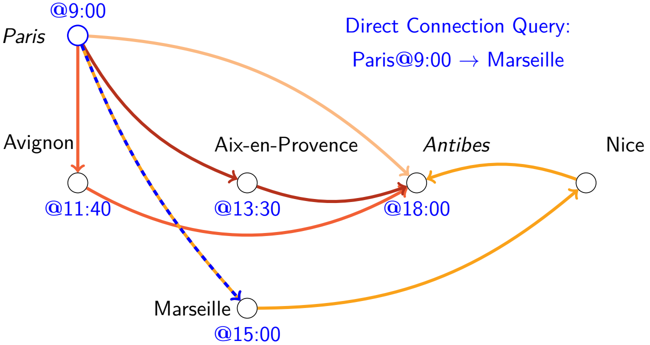


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

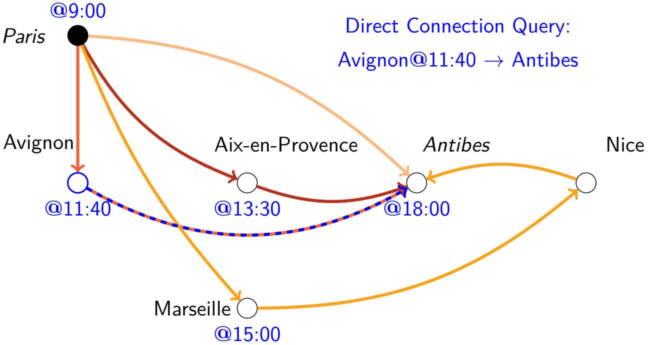


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

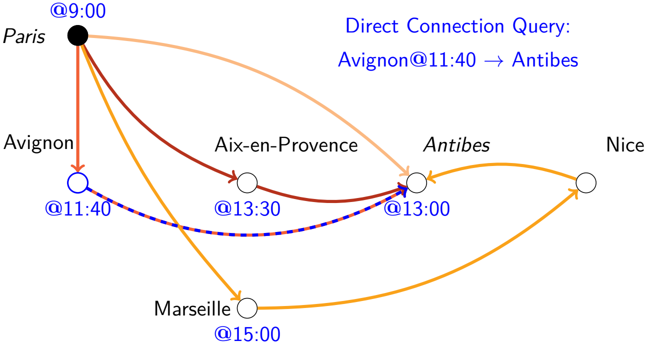


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

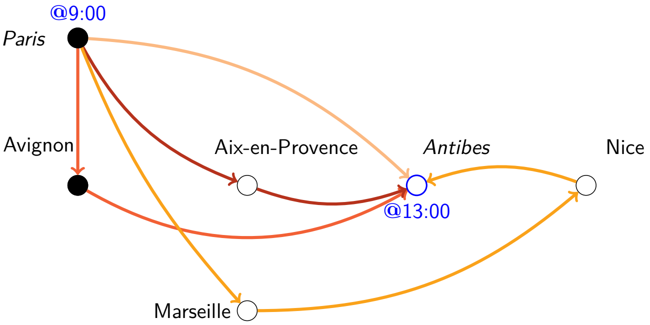


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph

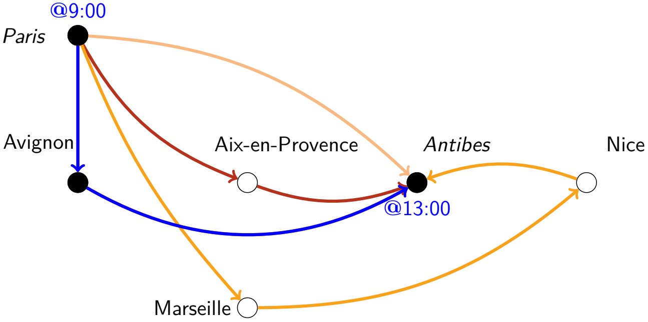


Transfer Pattern Routing

Given the optimal patterns from *Paris* to *Antibes*:

- Paris, Antibes
- P, Avignon, A
- P, Aix-en-Provence, A
- P, Marseille, Nice, A

(2) Search on the graph



Transfer Pattern Routing

Public Transportation Route Planning

- Find a set of Pareto-optimal routes for a query $A@t \rightarrow B$ given a set of timetables tt
- Different structure than Road Networks, different algorithms

Transfer Pattern Routing

- *Bast et al. ESA 2010*
- Given the optimal transfer patterns, search the best route among them
- Multi-criteria
- Very fast even on huge networks (e.g. Northern America)
- State-of-the-art, Google Maps

Transfer Pattern Routing

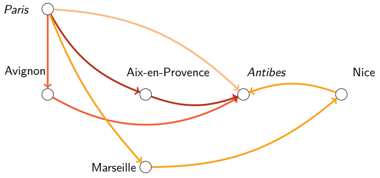
Components

Data computed from timetables tt

- Transfer patterns TP_{tt}
- Direct connection datastructure DC_{tt}

At query time

- Dijkstra on query graph built from patterns
- Relax arcs using direct connection queries



Transfer Patterns vs. Real-Time Updates

Time-consuming precomputation

- Computation of patterns in $O(N^2)$
- Heuristics: important stations, limits
- Still very long

Route planning applications

- Steady flow of updates during operation
- Example: Delay of trips
  - Traffic jam, strike, blocked rails, ...

No guarantee for optimal responses when trips are delayed

- Transfer patterns are computed on the original network
- Any update may introduce a new optimal pattern

Handling real-time updates with transfer patterns

Observation

- Typical updates introduce only small changes to the timetables, $tt \to tt'$
- Computing $TP_{tt'}$ is expensive, $DC_{tt'}$ not.

Main idea

- Update only the direct connection data $DC_{tt'}$
- Search on the original transfer patterns TP_{tt}

Expectation

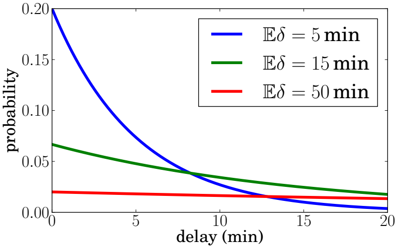
- Correct responses in most cases
- What happens elsewhere?

How much does the quality of results suffer?

Experimental Setup

- Compute transfer patterns
- Apply delay scenario to dataset ($tt \rightarrow tt'$)
- Answer N queries using TP_{tt}, $DC_{tt'}$
- Compare results with ground truth

Delay Scenario	Average delay $\mathbb{E}\delta$ 5 min	15 min	50 min
Low	25%	-	-
Medium	-	25%	-
High	-	-	25%
Mix Low	10%	3%	1%
Mix Normal	20%	10%	5%
Mix Chaos	40%	40%	20%



Experiment 1

Classification of responses

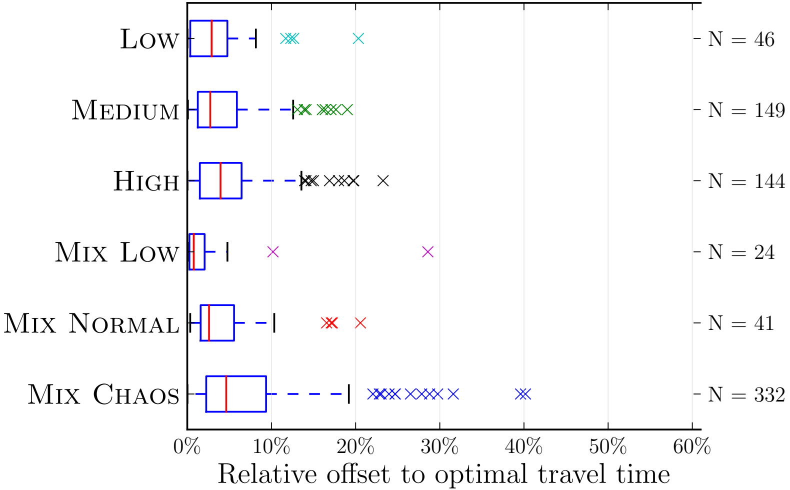
Example: 50,000 queries on New York City (2.3 M departures)

	OPTIMAL	ALMOST A	ALMOST B	BAD
	$d = 0$	$d \leq 5\ min$	$d \leq 10\ min$	
		$\wedge \frac{d}{c^*} \leq 0.05$	$\wedge \frac{d}{c^*} \leq 0.10$	
Null	99.99%	0.01%	0.00%	0.00%
Low	99.87%	0.09%	0.00%	0.04%
Medium	99.68%	0.19%	0.03%	0.10%
High	99.72%	0.17%	0.02%	0.09%
Mix Low	99.93%	0.05%	0.00%	0.02%
Mix Normal	99.89%	0.07%	0.01%	0.03%
Mix Chaos	99.19%	0.57%	0.07%	0.17%

$c :=$ *travel time*, $d := c^* - c$

Experiment 1

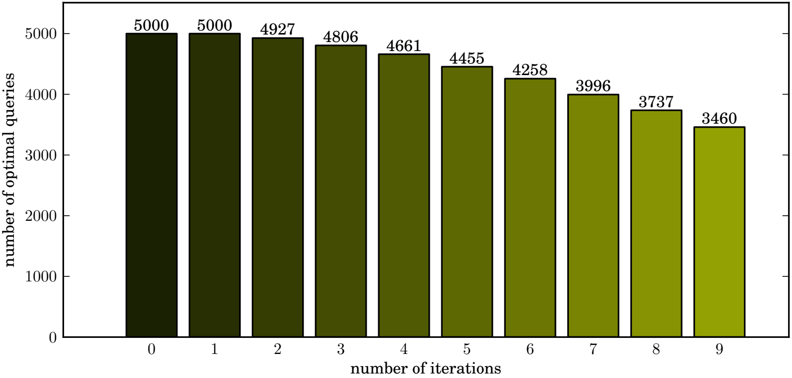
Suboptimal responses



Experiment 2

Directed Delay

- Repeat a fixed query
- Delay the vehicles of the optimal connections
- Until the response becomes suboptimal



Summary

Delay-Robustness of Transfer Patterns

Contribution

- Strategy to adapt to timetable updates
- Empirical proof for robustness to delay

In the paper

- Updating the direct connection data in real-time
- Detailed experimental evaluation
- Analysis of reasons for robustness
- Discussion of possible improvements

Find it here: `http://drops.dagstuhl.de/opus/volltexte/2013/4243/pdf/5.pdf`