Algorithms for Route Planning in Transportation Networks

Dorothea Wagner
Motivation

An important application, e.g.,
- navigation systems for cars,
- Google Maps, Bing Maps, . . . ,
- timetable information.

Many commercial systems
- use heuristic methods,
- consider “reasonable” part of the network,
- have no quality guarantees.

Find methods for route planning in transportation networks with provably optimal solutions regarding the quality of the routes.
Problem

request:
- find the best connection in a transportation network

idea:
- network as graph $G = (V, E)$
- edge weights are travel times
- shortest paths in $G$ equal quickest connections
- classic problem (Dijkstra)

problems:
- transport networks are huge
- Dijkstra too slow ($> 1$ second)
Speed-Up Techniques

observations:
- Dijkstra visits all nodes closer than the target
- unnecessary computations
- many requests in a hardly changing network

idea:
- two-phase algorithm:
  - offline: compute additional data during preprocessing
  - online: speed-up query with this data
- 3 criteria: preprocessing time and space, speed-up over Dijkstra
Showpiece of Algorithm Engineering

- Design
- Experiment
- Implement
- Analyze

Algorithmics

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks

Institute for Theoretical Informatics
Chair Algorithmics
Showpiece of Algorithm Engineering

- Design
- Experiment
- Implement
- Analyze

- Falsifiable Hypotheses

- Realistic machine models
- Real-world Data

Performance guarantees & algorithm dependability

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks
Shortcuts

[SWW99, SS05, GSSD08]

observation:

- nodes with low degree are not important

contract graph

- iteratively remove such nodes
- add shortcuts to preserve distances between non-removed nodes

query:

- bidirectional
- prune edges heading less important nodes
Contraction Hierarchies [GSSD08]

**idea:** solely use contraction

**approach:**
- heuristically order nodes by “importance”
- contract nodes in that order
- node $v$ contracted by

1. **forall edges** $(u, v)$ and $(v, w)$ **do**
2.  
3.  
4. **query only looks at edges to more important nodes**
Example: CH Preprocessing

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks

Institute for Theoretical Informatics
Chair Algorithmics
Example: CH Preprocessing
Example: CH Preprocessing
Example: CH Preprocessing
Example: CH Preprocessing
Example: CH Preprocessing
modified bidirectional Dijkstra

upward graph \( G^\uparrow := (V, E^\uparrow) \) with \( E^\uparrow := \{ (u, v) \in E : u < v \} \)

downward graph \( G^\downarrow := (V, E^\downarrow) \) with \( E^\downarrow := \{ (u, v) \in E : u > v \} \)

forward search in \( G^\uparrow \) and backward search in \( G^\downarrow \)
CH Query

- modified bidirectional Dijkstra
- upward graph \( G^{\uparrow} := (V, E^{\uparrow}) \) with \( E^{\uparrow} := \{ (u, v) \in E : u < v \} \)
- downward graph \( G^{\downarrow} := (V, E^{\downarrow}) \) with \( E^{\downarrow} := \{ (u, v) \in E : u > v \} \)
- forward search in \( G^{\uparrow} \) and backward search in \( G^{\downarrow} \)
CH Query

- modified bidirectional Dijkstra
- upward graph $G_{\uparrow} := (V, E_{\uparrow})$ with $E_{\uparrow} := \{(u, v) \in E : u < v\}$
- downward graph $G_{\downarrow} := (V, E_{\downarrow})$ with $E_{\downarrow} := \{(u, v) \in E : u > v\}$
- forward search in $G_{\uparrow}$ and backward search in $G_{\downarrow}$
CH Query

- modified bidirectional Dijkstra
- upward graph $G_{\uparrow} := (V, E_{\uparrow})$ with $E_{\uparrow} := \{(u, v) \in E : u < v\}$
- downward graph $G_{\downarrow} := (V, E_{\downarrow})$ with $E_{\downarrow} := \{(u, v) \in E : u > v\}$
- forward search in $G_{\uparrow}$ and backward search in $G_{\downarrow}$
Question: What is a good contraction order?
- up to now: solely heuristical [GSSD08]
- no guarantees

WeakCH [BCRW13]
- balanced separator nodes are important
  → resulting CH is called weak
- $O(n^\alpha)$ separators $\rightarrow O(n^\alpha)$ nodes in the search space
- order is independent of metric
From Practice to Theory: WeakCH

**Question:** What is a good contraction order?
- up to now: solely heuristical [GSSD08]
- no guarantees

**WeakCH** [BCRW13]
- balanced separator nodes are important
  → resulting CH is called *weak*
- $O(n^\alpha)$ separators $\rightarrow O(n^\alpha)$ nodes in the search space
- order is independent of metric
New Challenges

realistic customizable routes:
- user customizable metrics
e.g., height restrictions, avoid freeways, eco-friendliness, . . .
- fast customization time per metric
- very small space overhead

timetable information:
- consider public transportation networks
- develop new techniques
- robustness towards the input?

multi-modal routes:
- change the type of transportation during the journey
- allow only “reasonable” transfers
- several constraints to the shortest path
idea:
- CH topology is the same regardless of metric
- quickly introduce new metric
From Theory to Practice: Customizable Contraction Hierarchies

[DSW14]

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

some arc in the CH
From Theory to Practice: Customizable Contraction Hierarchies

[DSW14]

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

establish lower triangle inequality
idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

establish lower triangle inequality
idea:

- CH topology is the same regardless of metric
- quickly introduce new metric

do this for all lower triangles
From Theory to Practice: Customizable Contraction Hierarchies

[DSW14]

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
From Theory to Practice: Customizable Contraction Hierarchies

[DSW14]

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
From Theory to Practice: Customizable Contraction Hierarchies

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
From Theory to Practice: Customizable Contraction Hierarchies

[DSW14]

idea:
- CH topology is the same regardless of metric
- quickly introduce new metric

process arcs increasing by order
What is a Timetable?

Karlsruhe / 10 min

8:00 → 8:31

Mannheim / 9 min

Rome / 10 min

8:31 → 11:00

Milan / 12 min

Frankfurt / 12 min

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks
What is a Timetable?

Karlsruhe / 10 min

8:00 → 8:31

Mannheim / 9 min

8:31 → 9:08

Rome / 10 min

8:31 → 11:00

Frankfurt / 12 min

Milan / 12 min
What is a Timetable?

Karlsruhe / 10 min

8:00 → 8:31

Mannheim / 9 min

Rome / 10 min

8:31 → 11:00

Milan / 12 min

8:31 → 9:08

Frankfurt / 12 min
What is a Timetable?

Karlsruhe / 10 min

8:00 → 8:31

Mannheim / 9 min

8:31 → 9:08

Rome / 10 min

8:31 → 11:00

Milan / 12 min

Frankfurt / 12 min
Existing Approaches

list of connections and stops

query
Existing Approaches

list of connections and stops

Time Expanded

[PSWZ08]

query
Existing Approaches

- Time Expanded
  - [PSWZ08]
- Time Dependent
  - [PSWZ08]
- complex arc weights

query

list of connections and stops
Existing Approaches

list of connections and stops

Time Expanded
[PSWZ08]

Time Dependent
[PSWZ08]

query
Existing Approaches

list of connections and stops

Time Expanded [PSWZ08]

Time Dependent [PSWZ08]

RAPTOR [DPW12a]
### Earliest Arrival Time Problem

**Input:** ordered connection list, source stop, source time, target stop  
**Output:** earliest arrival time

<table>
<thead>
<tr>
<th>Connections</th>
<th>depstop</th>
<th>arrstop</th>
<th>deptime</th>
<th>arrtime</th>
<th>tripid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Is trip reachable?</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Connection Scan [DPSW13]**

<table>
<thead>
<tr>
<th>Stop ID</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>+∞</td>
</tr>
<tr>
<td></td>
<td>+∞</td>
</tr>
<tr>
<td></td>
<td>+∞</td>
</tr>
<tr>
<td></td>
<td>+∞</td>
</tr>
<tr>
<td></td>
<td>+∞</td>
</tr>
</tbody>
</table>
### Earliest Arrival Time Problem

**Input:** ordered connection list, source stop, source time, target stop

**Output:** earliest arrival time

<table>
<thead>
<tr>
<th>stop ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival time</td>
<td>+∞</td>
<td>+∞</td>
<td>+∞</td>
<td>+∞</td>
<td>+∞</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>connections ordered by departure time</th>
</tr>
</thead>
<tbody>
<tr>
<td>dep: 1</td>
</tr>
</tbody>
</table>

| is trip reachable? | F | F | ... |
## Earliest Arrival Time Problem

**Input:** ordered connection list, source stop, source time, target stop

**Output:** earliest arrival time

### Connections

<table>
<thead>
<tr>
<th>Stop ID</th>
<th>Arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>1</td>
<td>8:00</td>
</tr>
<tr>
<td>2</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>3</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>4</td>
<td>$+\infty$</td>
</tr>
</tbody>
</table>

### Orders

<table>
<thead>
<tr>
<th>Connections</th>
<th>Ordered by Departure Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>dep: 1</td>
<td>arr: 3</td>
</tr>
<tr>
<td>9:00</td>
<td>9:25</td>
</tr>
<tr>
<td>tripid</td>
<td>dep: 3</td>
</tr>
<tr>
<td></td>
<td>arr: 4</td>
</tr>
<tr>
<td></td>
<td>9:15</td>
</tr>
<tr>
<td></td>
<td>9:45</td>
</tr>
<tr>
<td></td>
<td>tripid</td>
</tr>
<tr>
<td></td>
<td>dep: 3</td>
</tr>
<tr>
<td></td>
<td>arr: 4</td>
</tr>
<tr>
<td></td>
<td>9:25</td>
</tr>
<tr>
<td></td>
<td>9:55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Is Trip Reachable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
</tr>
<tr>
<td>F</td>
</tr>
</tbody>
</table>

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks

Institute for Theoretical Informatics
Chair Algorithmics
Earliest Arrival Time Problem

input: ordered connection list, source stop, source time, target stop
output: earliest arrival time

connections
ordered by
departure time

stop ID
arrival time

is trip reachable?

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks
Connection Scan \cite{DPSW13}

Earliest Arrival Time Problem

input: ordered connection list, source stop, source time, target stop
output: earliest arrival time

<table>
<thead>
<tr>
<th>stop ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival time</td>
<td>+∞</td>
<td>8:00</td>
<td>+∞</td>
<td>9:25</td>
<td>+∞</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>connections</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>orderd by departure time</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

is trip reachable? | | | |
| T | F | | |
# Connection Scan [DPSW13]

## Earliest Arrival Time Problem

**input:** ordered connection list, source stop, source time, target stop  
**output:** earliest arrival time

### Connections

<table>
<thead>
<tr>
<th>stop ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival time</td>
<td>+∞</td>
<td>8:00</td>
<td>+∞</td>
<td>9:25</td>
<td>+∞</td>
<td>...</td>
</tr>
</tbody>
</table>

### Connections ordered by departure time

<table>
<thead>
<tr>
<th>connections</th>
<th>dep: 1</th>
<th>arr: 3</th>
<th>9:00</th>
<th>9:25</th>
<th>tripid</th>
<th>9:15</th>
<th>9:45</th>
<th>tripid</th>
<th>dep: 3</th>
<th>arr: 4</th>
<th>9:25</th>
<th>9:55</th>
<th>tripid</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>is trip reachable?</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks

Institute for Theoretical Informatics  
Chair Algorithmics
**Connection Scan** [DPSW13]

**Earliest Arrival Time Problem**

- **Input:** ordered connection list, source stop, source time, target stop
- **Output:** earliest arrival time

<table>
<thead>
<tr>
<th>stop ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival time</td>
<td>+∞</td>
<td>8:00</td>
<td>+∞</td>
<td>9:25</td>
<td>+∞</td>
<td>...</td>
</tr>
</tbody>
</table>

**Connections ordered by departure time**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>9:00</td>
<td>3</td>
<td>4</td>
<td>9:15</td>
<td>3</td>
<td>4</td>
<td>9:45</td>
<td>3</td>
<td>4</td>
<td>9:55</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Is trip reachable?**

<table>
<thead>
<tr>
<th>...</th>
<th>T</th>
<th>...</th>
<th>F</th>
<th>...</th>
</tr>
</thead>
</table>

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks

Institute for Theoretical Informatics
Chair Algorithmics
Earliest Arrival Time Problem

input: ordered connection list, source stop, source time, target stop
output: earliest arrival time

```
         stop ID
         arrival time
         ··· 0     1     2     3     4     ···
         +∞ 8:00 +∞ 9:25 9:55 ···

         connections
         ordered by
         departure time
         ··· dep.: 1 9:00 9:25 9:15 9:45 ···
         arr.: 3 25 45
         ··· tripid dep.: 3 ···
         arr.: 4 ···
         ··· tripid
         dep.: 3 9:25 9:55 ···
         arr.: 4 55
         ··· tripid
         ···

         is trip reachable?
         ··· T     F     ···
```
### Connection Scan [DPSW13]

#### Earliest Arrival Time Problem

- **Input:** Ordered connection list, source stop, source time, target stop
- **Output:** Earliest arrival time

<table>
<thead>
<tr>
<th>stop ID</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>arrival time</td>
<td>+∞</td>
<td>8:00</td>
<td>+∞</td>
<td>9:25</td>
<td>9:55</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>connections ordered by departure time</th>
</tr>
</thead>
<tbody>
<tr>
<td>dep:</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>is trip reachable?</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
</tbody>
</table>
Vision: Navi for the World

Worldwide network composed of car, rail, flight, …
Multi-Modal Routing

up to now, research mostly on uni-modal routing
- restricted to one transportation network
- time-independent and time-dependent (separately)

what we really want is planning a journey by

- choosing source and destination
- desired means of transportation (car, train, flight, . . .)
- . . . in a mixed network
“Classic” Shortest Paths

main challenge for multi-modal route planning:

shortest path

desirable path

a shortest $s$-$t$-path could require too many transfers
### Definition (LABEL CONSTRAINED SHORTEST PATH PROBLEM)

**Input:**
- weighted graph \( G = (V, E) \), source \( s \), and target \( t \)
- a vehicle type per edge \( f : E \rightarrow \Sigma \)
- formal language \( L \subseteq \Sigma^* \)

**Output:**
- a \( s-t \)-path in \( G \)
- minimize the edge weight sum
- word along the path must be in \( L \)

### Theorem

*The LABEL CONSTRAINED SHORTEST PATH PROBLEM (LCSPP) is solvable in polynomial time, if \( L \) is a regular language.*
Adapting Speed-Up Techniques

[Paj09], [KLPC11]

- **bidirectional search**
  easily adaptable (time-dependency is hard)

- **goal-directed search**
  ALT adaptable but low speed-ups,
  Arc-Flags turns out difficult

- **contraction**
  adaptable with some restrictions
  - contracted graph is called the **core**

Two promising approaches:

- **access-node routing (ANR)**
  adapting ideas from transit-node routing (table lookups)

- **user-constrained CH (UCCH)**
  augmenting contraction hierarchies
User-constrained Shortest Paths

- optimality of multi-modal paths depends on user-choice
  desired modes of transport, constraints on the sequence of modes
- user-constraints are an additional input to the query
- preprocessing should respect these query-time constraints
  ANR predetermines the constraint automaton during preprocessing

question: can CH be adapted to this setting?
User-constrained Shortest Paths

- optimality of multi-modal paths depends on user-choice
- desired modes of transport, constraints on the sequence of modes
- user-constraints are an additional input to the query
- preprocessing should respect these query-time constraints
  ANR predetermines the constraint automaton during preprocessing

**question:** can CH be adapted to this setting?
User-constrained Shortest Paths

- optimality of multi-modal paths depends on user-choice
- desired modes of transport, constraints on the sequence of modes
- user-constraints are an additional input to the query
- preprocessing should respect these query-time constraints
  ANR predetermines the constraint automaton during preprocessing

question: can CH be adapted to this setting?
User-constrained Shortest Paths

- optimality of multi-modal paths depends on user-choice desired modes of transport, constraints on the sequence of modes
- user-constraints are an additional input to the query
- preprocessing should respect these query-time constraints
  ANR predetermines the constraint automaton during preprocessing

**question:** can CH be adapted to this setting?
User-constrained Shortest Paths

- optimality of multi-modal paths depends on user-choice
  desired modes of transport, constraints on the sequence of modes
- user-constraints are an additional input to the query
- preprocessing should respect these query-time constraints
  ANR predetermines the constraint automaton during preprocessing

question: can CH be adapted to this setting?
User-constrained CH (UCCH) [DPW12b]

multi-modal CH:
- contraction introduces shortcuts with label sequences
- witness search depends on constraints
  requires a-priori knowledge of the constraint automata

idea: do not contract nodes with incident link-edges.

- contraction and witness search are limited to each modality
  ⇒ preprocessing independent of mode sequence constraints
Example: UCCH Preprocessing
preprocessing
- linked nodes are not contracted thus contained in the core
- shortcuts between core nodes preserve distances
  allows using the road network between rail stations

query
- CH search on the component
- label constrained search on the core
- engineering yields further improvement
**mmRAPTOR [DDPWW13]**

**idea:** compute multicriteria, multimodale Pareto sets

- optimize arrival time plus
- various (dependent on the vehicle type) "convenience criteria" for example # transfers, walking time, taxi costs, etc.

**known problem:** Pareto sets quickly grow in the # criteria.
mmRAPTOR [DDPWW13]

**idea:** compute multicriteria, multimodale Pareto sets

- optimize arrival time plus
- various (dependent on the vehicle type) "convenience criteria" for example # transfers, walking time, taxi costs, etc.

**known problem:** Pareto sets quickly grow in the # criteria.
Relevant Journeys

approach:

- compute full Pareto set with multicriteria-aware algorithm
- weight all journeys using fuzzy logic [FA04]
- the highest journeys are the most relevant ones
Relevant Journeys

**approach:**

- compute full Pareto set with multicriteria-aware algorithm
- weight all journeys using fuzzy logic [FA04]
- the highest journeys are the most relevant ones

Dorothea Wagner – Algorithms for Route Planning in Transportation Networks
Relevant Journeys

approach:

- compute full Pareto set with multicriteria-aware algorithm
- weight all journeys using fuzzy logic [FA04]
- the highest journeys are the most relevant ones
Outlook

- big steps towards “solving” timetable & multi-modal routing, but not yet there
  - scalability to huge networks is still a problem
  - delay robustness: How to avoid future delays?
  - real-time updates: How to react to known delays?
  - result-diversity
    - good because no single journey fits all needs
    - bad because generating all “optimal” is too expensive
Outlook

- big steps towards “solving” timetable & multi-modal routing, but not yet there
- scalability to huge networks is still a problem
- delay robustness: How to avoid future delays?
- real-time updates: How to react to known delays?
- result-diversity
  - good because no single journey fits all needs
  - bad because generating all “optimal” is too expensive
Reinhard Bauer, Tobias Columbus, Ignaz Rutter, and Dorothea Wagner.
Search-space size in contraction hierarchies.

Chris Barrett, Riko Jacob, and Madhav V. Marathe.
Formal-language-constrained path problems.

Daniel Delling, Julian Dibbelt, Thomas Pajor, Dorothea Wagner, and Renato F. Werneck.
Computing multimodal journeys in practice.

Julian Dibbelt, Thomas Pajor, Ben Strasser, and Dorothea Wagner.
Intriguingly simple and fast transit routing.

Daniel Delling, Thomas Pajor, and Renato F. Werneck.
Round-based public transit routing.

Julian Dibbelt, Thomas Pajor, and Dorothea Wagner.
User-constrained multi-modal route planning.
Julian Dibbelt, Ben Strasser, and Dorothea Wagner.
Customizable contraction hierarchies.
Technical report, ITI Wagner, Department of Informatics, Karlsruhe Institute of Technology (KIT), 2014.
http://arxiv.org/abs/1402.0402

Marco Farina and Paolo Amato.
A fuzzy definition of “optimality” for many-criteria optimization problems.

Robert Geisberger, Peter Sanders, Dominik Schultes, and Daniel Delling.
Contraction hierarchies: Faster and simpler hierarchical routing in road networks.

Dominik Kirchler, Leo Liberti, Thomas Pajor, and Roberto Wolfler Calvo.
UniALT for regular language constraint shortest paths on a multi-modal transportation network.

Thomas Pajor.
Multi-modal route planning.

Evangelia Pyrga, Frank Schulz, Dorothea Wagner, and Christos Zaroliagis.
Efficient models for timetable information in public transportation systems.
Peter Sanders and Dominik Schultes.
Highway hierarchies hasten exact shortest path queries.

Frank Schulz, Dorothea Wagner, and Karsten Weihe.
Dijkstra’s algorithm on-line: An empirical case study from public railroad transport.