Karlsruhe Institute of Technology

Time-Dependent Route Planning with Generalized Objective Functions

ESA 2012 – Gernot Veit Batz and Peter Sanders – *{batz,sanders}@kit.edu*

Institute of Theoretical Informatics, Algorithmics II

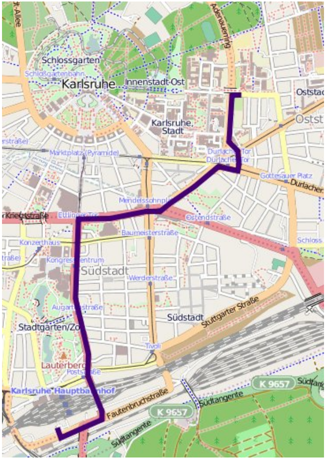
Time-Dependent Route Planning

Motivation

From Karlsruhe Main Station
to Karlsruhe Computer Science Building

At 3:00 at night:

- Empty streets
- Through the city center.



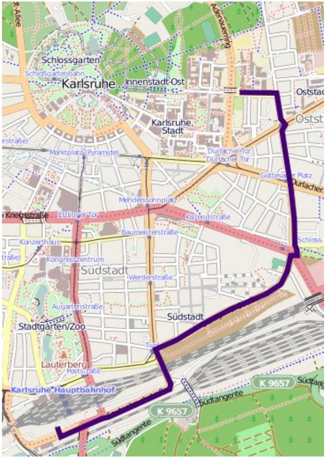
Time-Dependent Route Planning

Motivation

From Karlsruhe Main Station
to Karlsruhe Computer Science Building

At 8:00 in the morning:

- Rush hour
- Avoid crowded junctions.

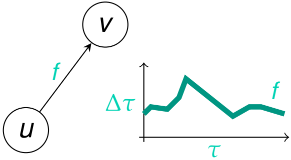


Time-Dependent Route Planning

State of the Art: Only Travel Times

Edge weights are travel time functions

- f: point in time $\mapsto$ Δ travel time
- piecewise linear
- FIFO-property – waiting not beneficial



Earliest arrival query:

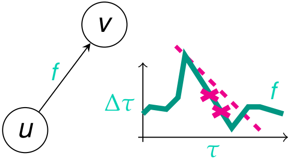
- minimum travel time route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) +$ is minimal amongst all routes

Time-Dependent Route Planning

State of the Art: Only Travel Times

Edge weights are travel time functions

- f: point in time $\mapsto \Delta$ travel time
- piecewise linear
- FIFO-property – waiting not beneficial



Earliest arrival query:

- minimum travel time route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) +$ is minimal amongst all routes

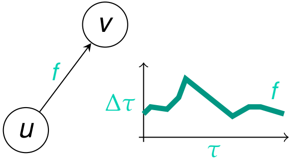
G.V. Batz and P. Sanders:

Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

State of the Art: Only Travel Times

Edge weights are travel time functions

- f: point in time $\mapsto \Delta$ travel time
- piecewise linear
- FIFO-property – waiting not beneficial



Earliest arrival query:

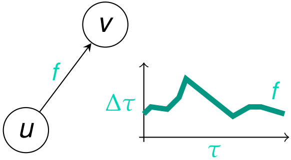
- minimum travel time route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) +$ is minimal amongst all routes

Time-Dependent Route Planning

State of the Art: Only Travel Times

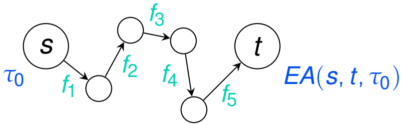
Edge weights are travel time functions

- f: point in time $\mapsto \Delta$ travel time
- piecewise linear
- FIFO-property – waiting not beneficial



Earliest arrival query:

- minimum travel time route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) +$ is minimal amongst all routes

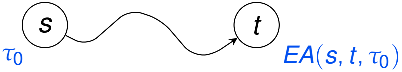


Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

Time-Dependent Route Planning

State of the Art: Only Travel Times

Selected Results:



Earliest Arrival Queries

Algorithm	Space Ovh. [B/n]	Speedup of Dijkstra	Maximum Error [%]	Citation
TCH	899	1428	–	[Batz et al. 2009]
ATCH	144	857	–	[Batz et al. 2010]
ATCH	23	685	–	[Batz et al. 2010]
SHARC	155	60	–	[Delling et al. 2008]
SHARC	68	1 177	0.61	[Brunel et al. 2010]
SHARC	14	491	0.61	[Brunel et al. 2010]

Optimizing Only Travel Time...

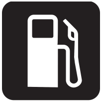
...is not Enough

Highly practical aspects stay unconsidered:

- energy efficient routes
- tolls
- avoid large detours (related to energy efficient)
- avoid inconvenient routes
- ...

Optimizing Only Travel Time...

...is not Enough

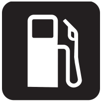


Highly practical aspects stay unconsidered:

- energy efficient routes
- tolls
- avoid large detours (related to energy efficient)
- avoid inconvenient routes
- ...

Optimizing Only Travel Time...

...is not Enough

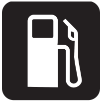


Highly practical aspects stay unconsidered:

- energy efficient routes
- tolls
- avoid large detours (related to energy efficient)
- avoid inconvenient routes
- ...

Optimizing Only Travel Time...

...is not Enough

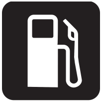


Highly practical aspects stay unconsidered:

- energy efficient routes
- tolls
- avoid large detours (related to energy efficient)
- avoid inconvenient routes
- ...

Optimizing Only Travel Time...

...is not Enough



Highly practical aspects stay unconsidered:

- energy efficient routes
- tolls
- avoid large detours (related to energy efficient)
- avoid inconvenient routes
- ...

We Generalize the Objective Function

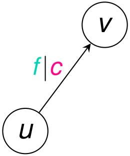
...Using Additional Time-Invariant Costs

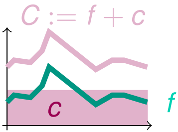
Karlsruhe Institute of Technology

Edge weights are pairs $f|c$

- travel time function f
- time-invariant cost $c \in \mathbb{R}_{\geq 0}$

$\Rightarrow$ time-dependent total cost $C := f + c$





Minimum Cost query:

- minimum total cost route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) + c_4 + \cdots + c_4$ is minimal amongst all routes

Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

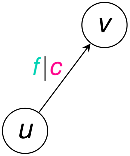
We Generalize the Objective Function

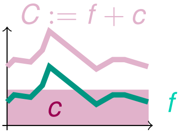
...Using Additional Time-Invariant Costs

Edge weights are pairs $f|c$

- travel time function f
- time-invariant cost $c \in \mathbb{R}_{\geq 0}$

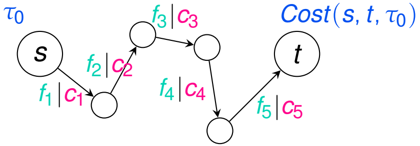
$\Rightarrow$ time-dependent total cost $C := f + c$





Minimum Cost query:

- minimum total cost route...
- ...for given departure time τ_0
- $(f_4 + id) \circ \cdots \circ (f_1 + id)(\tau_0) + c_4 + \cdots + c_4$ is minimal amongst all routes

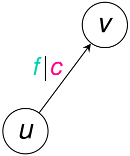


G.V. Batz and P. Sanders:

Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

Practical Applications

...of Time-Dependent Minimum Cost Route Planning



$$C = f + c$$

- **Energy efficient routes:**
  $c \propto$ distance (only approximation of energy)
- **Modeling tolls:**
  $c \propto$ toll charge
- **Avoiding inconvenient routes:**
  $c =$ penalty when narrow, steep, bumpy,...

And combinations: $c = c_1 + c_2 + c_3 + \ldots$

Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard

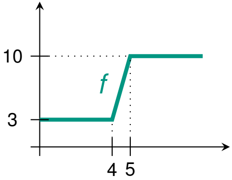
Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

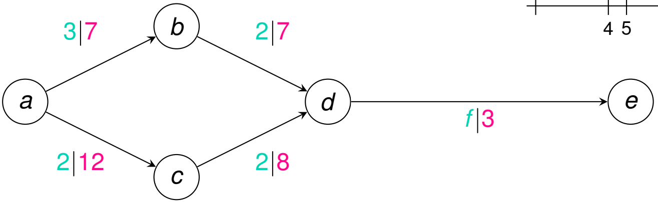
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard





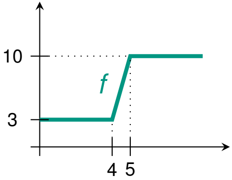
Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

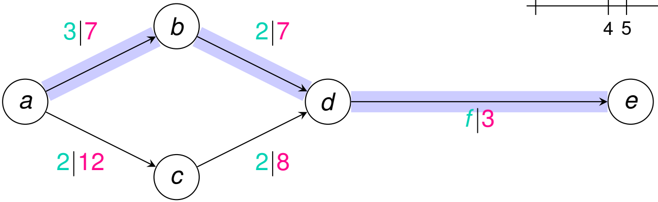
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



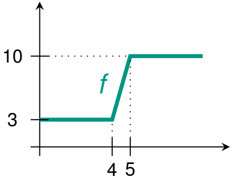


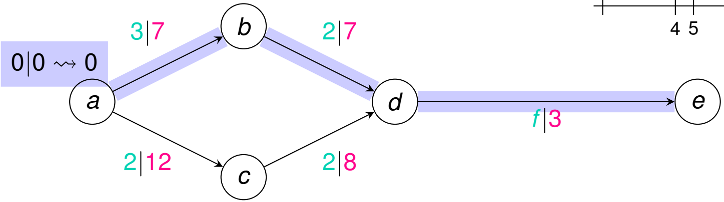
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



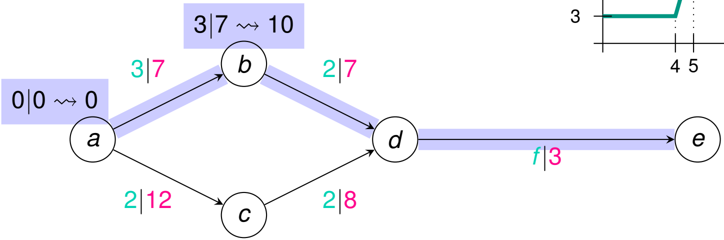


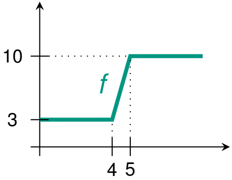
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard





G.V. Batz and P. Sanders:
Time-Dependent Route Planning with Generalized Objective Functions

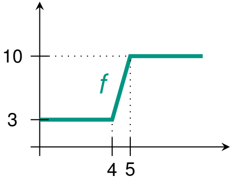
Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

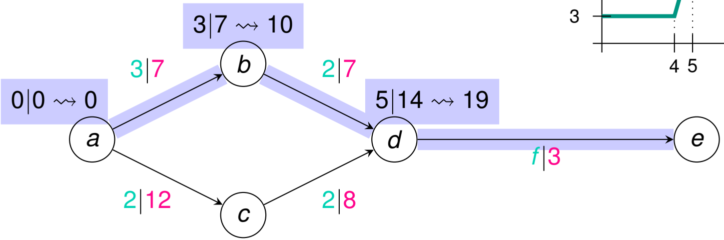
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



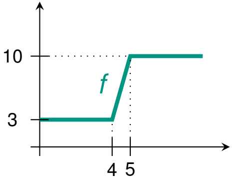


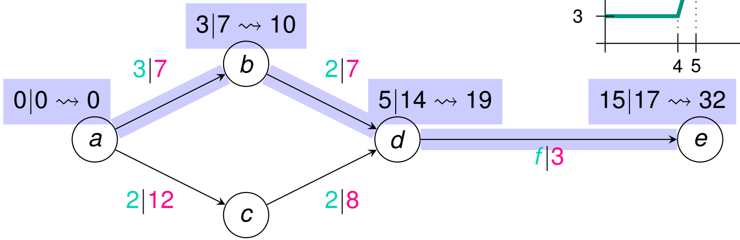
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



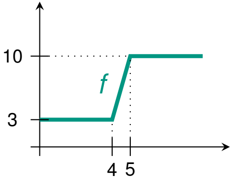


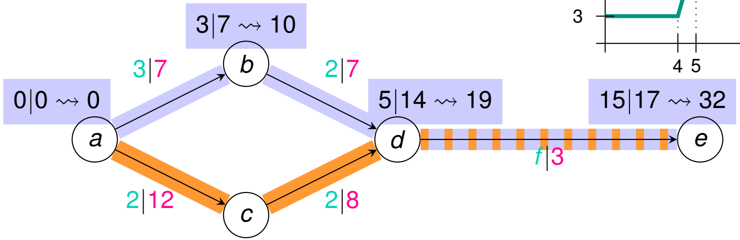
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



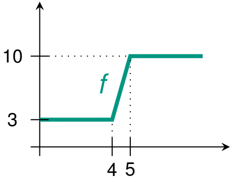


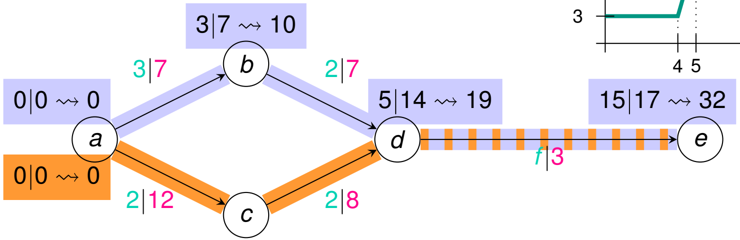
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard





G.V. Batz and P. Sanders:
Time-Dependent Route Planning with Generalized Objective Functions

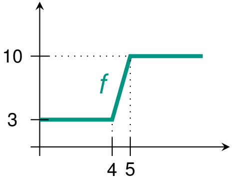
Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

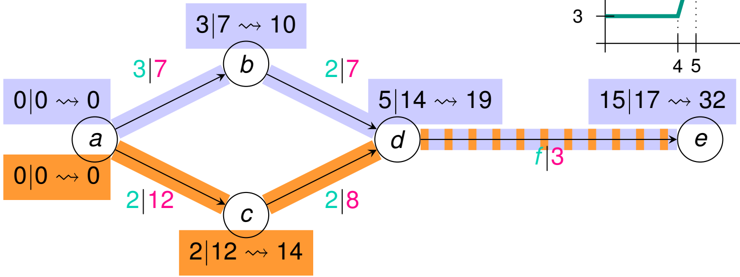
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



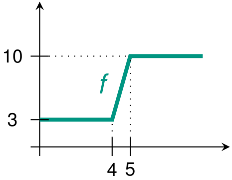


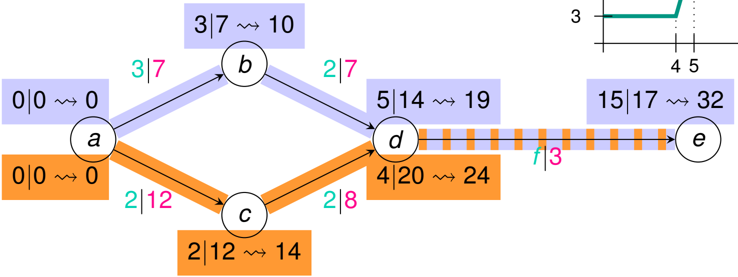
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



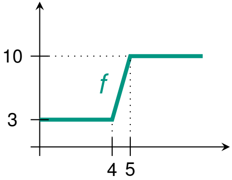


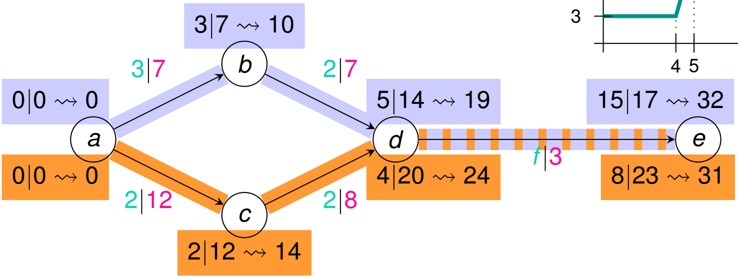
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard



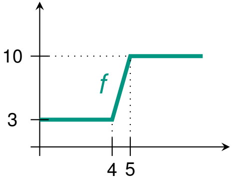


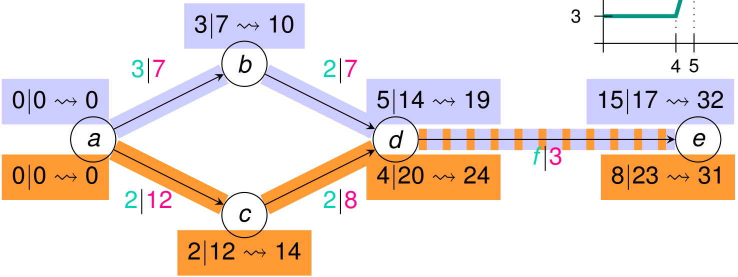
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard





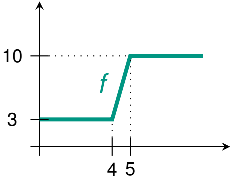
⇒ The only optimal route has a non-optimal prefix!

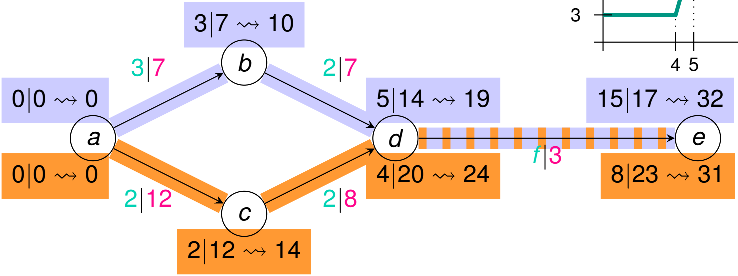
Complexity

...of Time-Dependent Minimum Cost Route Planning

Surprisingly, minimum cost queries are...

- ...very hard to answer
- ...much harder than earliest arrival queries
- ...even NP-hard





$\Rightarrow$ The only optimal route has a non-optimal prefix!

$\Rightarrow$ Sometimes all optimal routes have non-optimal subroutes.

NP-hardness

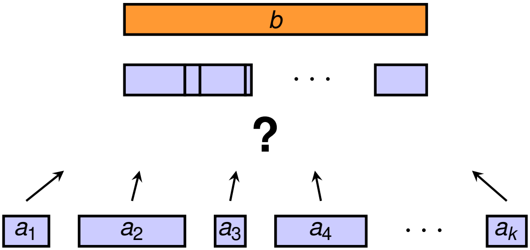
...of Time-Dependent Minimum Cost Queries

Proof:

Reducing **Number partitioning:**

Given: $a_1, \ldots, a_k, b \in \mathbb{N}_{>0}$

Question: Do $x_1, \ldots, x_k \in \{0, 1\}$ exist s.t. $b = x_1 a_1 + \cdots + x_2 a_k$?



(proof inspired by [Ahuja et al. 2003])

NP-hardness

...of Time-Dependent Minimum Cost Route Planning

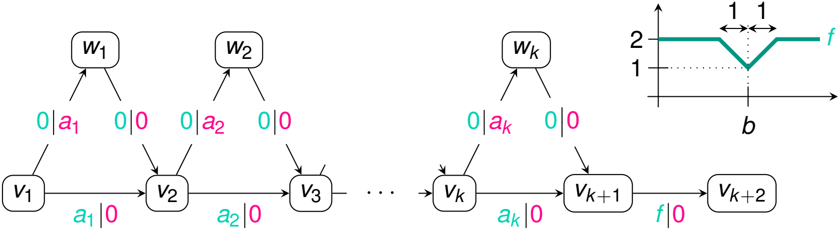
Proof:

Reducing **Number partitioning:**

Given: $a_1, \ldots, a_k, b \in \mathbb{N}_{>0}$

Question: Do $x_1, \ldots, x_k \in \{0, 1\}$ exist
s.t. $b = x_1 a_1 + \cdots + x_2 a_k$?

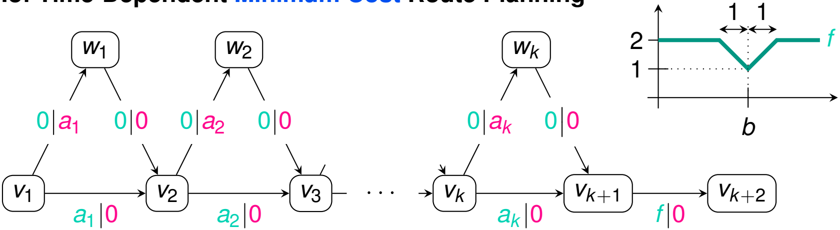
...to **minimum cost query** from v_1 to v_{k+2} departure time 0:



G.V. Batz and P. Sanders:
Time-Dependent Route Planning with Generalized Objective Functions
Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

NP-hardness

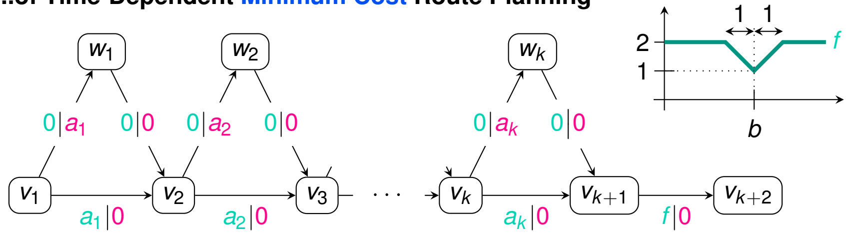
...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning

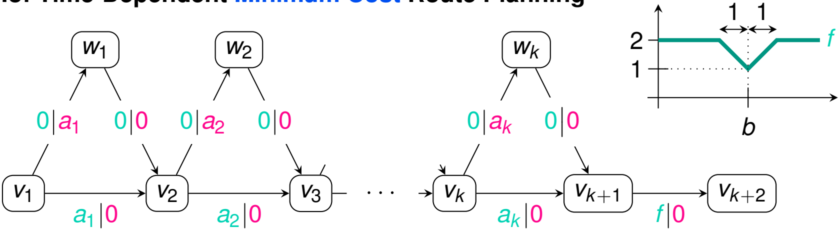


- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

NP-hardness

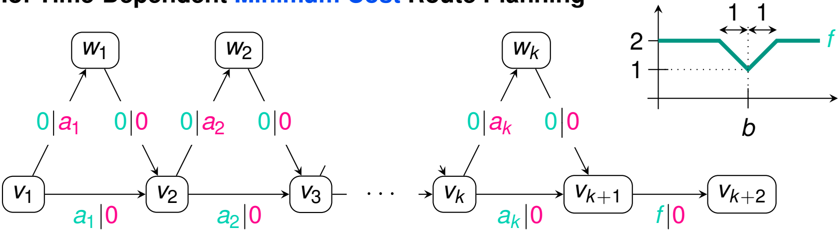
...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

partition problem answers yes

$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0,1\}$ s.t. $b = x_1 a_1 + \cdots + x_k a_k$

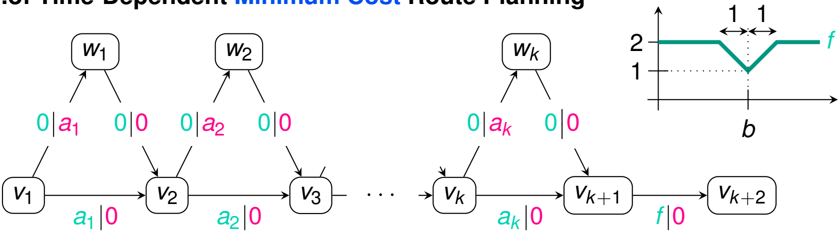
$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0,1\}$ s.t. $b = \sum_{i \in X} a_i$, $X = \{i \mid x_i = 1\} \subseteq \{1, \ldots, k\}$

$\Leftrightarrow \exists$ a path from v_1 to v_{k+1} has travel time b

$\Leftrightarrow$ minimum cost from v_1 to v_{k+2} for departure time 0 is $c_{\text{all}} + 1$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

partition problem answers yes

$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = x_1 a_1 + \cdots + x_k a_k$

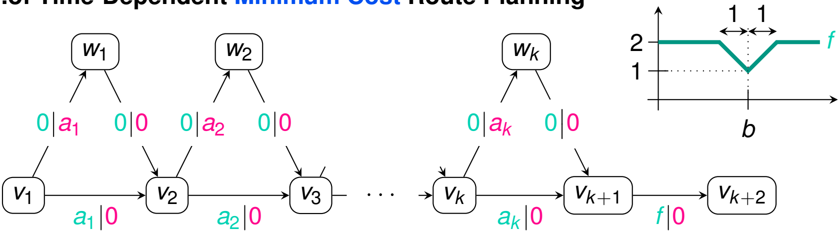
$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = \sum_{i \in X} a_i$, $X = \{i \mid x_i = 1\} \subseteq \{1, \ldots, k\}$

$\Leftrightarrow \exists$ a path from v_1 to v_{k+1} has travel time b

$\Leftrightarrow$ minimum cost from v_1 to v_{k+2} for departure time 0 is $c_{\text{all}} + 1$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

partition problem answers yes

$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = x_1 a_1 + \cdots + x_k a_k$

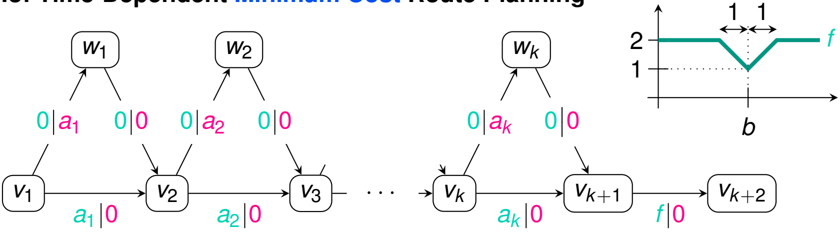
$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = \sum_{i \in X} a_i$, $X = \{i \mid x_i = 1\} \subseteq \{1, \ldots, k\}$

$\Leftrightarrow \exists$ a path from v_1 to v_{k+1} has travel time b

$\Leftrightarrow$ minimum cost from v_1 to v_{k+2} for departure time 0 is $c_{\text{all}} + 1$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

partition problem answers yes

$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = x_1 a_1 + \cdots + x_k a_k$

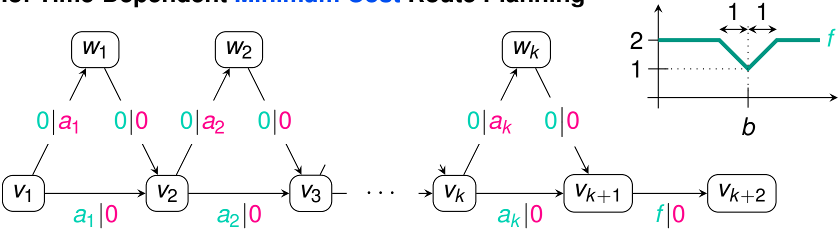
$\Leftrightarrow \exists\, x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = \sum_{i \in X} a_i,\ X = \{i \mid x_i = 1\} \subseteq \{1, \ldots, k\}$

$\Leftrightarrow \exists$ a path from v_1 to v_{k+1} has travel time b

$\Leftrightarrow$ minimum cost from v_1 to v_{k+2} for departure time 0 is $c_{\text{all}} + 1$

NP-hardness

...of Time-Dependent Minimum Cost Route Planning



- 2^k paths from v_1 to v_{k+1}
- all with same total cost $c_{\text{all}} := a_1 + \cdots + a_k$
- but different travel time: $\sum_{i \in X} a_i$ where $X \subseteq \{1, \ldots, k\}$

partition problem answers yes

$\Leftrightarrow \exists\ x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = x_1 a_1 + \cdots + x_k a_k$

$\Leftrightarrow \exists\ x_1, \ldots, x_k \in \{0, 1\}$ s.t. $b = \sum_{i \in X} a_i$, $X = \{i \mid x_i = 1\} \subseteq \{1, \ldots, k\}$

$\Leftrightarrow \exists$ a path from v_1 to v_{k+1} has travel time b

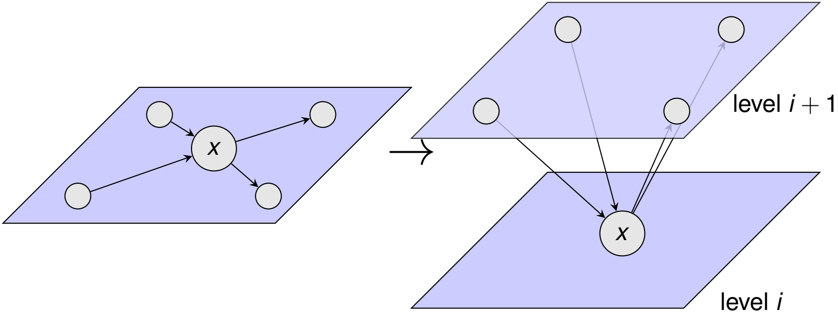
$\Leftrightarrow$ minimum cost from v_1 to v_{k+2} for departure time 0 is $c_{\text{all}} + 1$

Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts

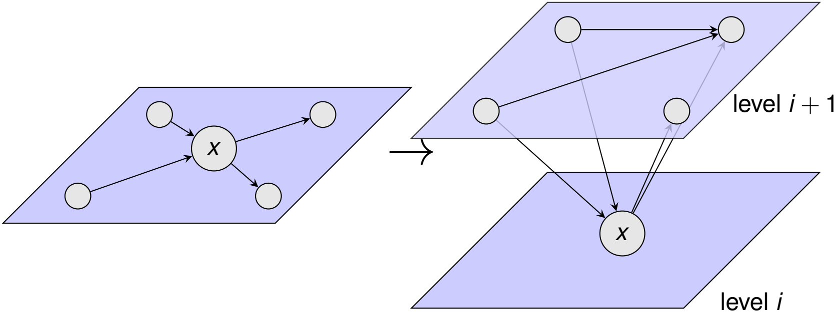


Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts

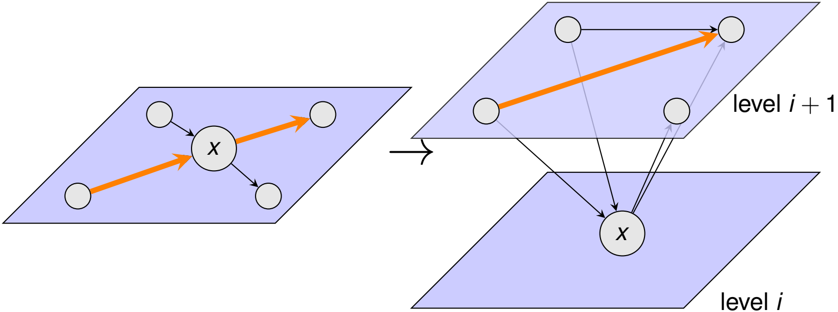


Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts

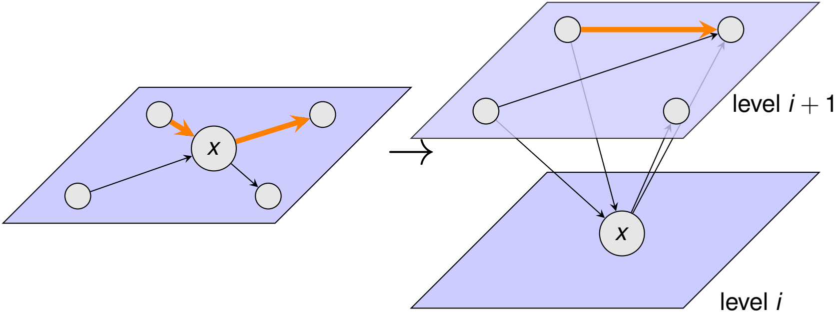


Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts

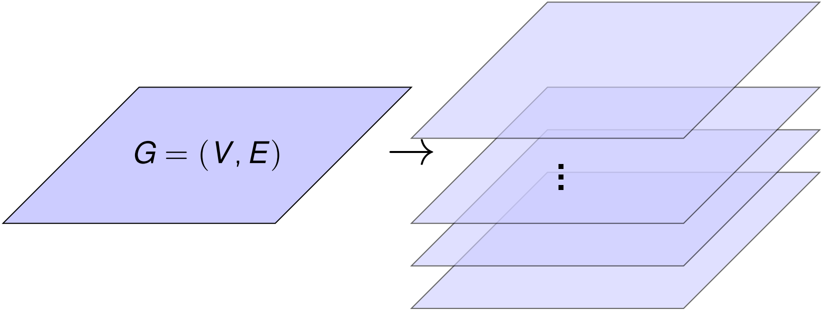


Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts

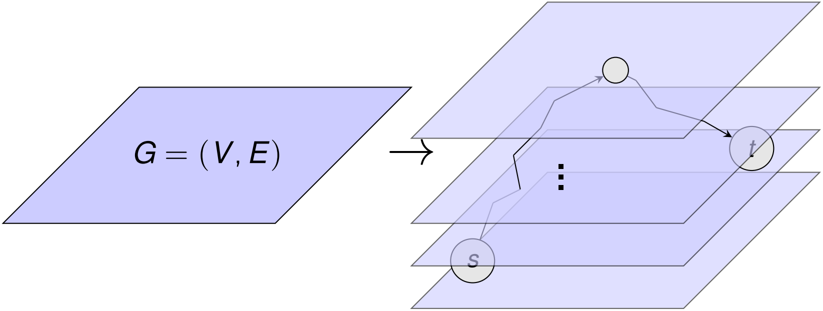


Contraction Hierarchies (CH) – Idea

[Geisberger et al. 2008]

Construct a hierarchy in a preprocessing step:

- Order nodes by importance
- Obtain next level by contracting next node
- Preserve optimal routes by inserting shortcuts



$\Rightarrow$ There is always an optimal up-down-route.

Heuristic Minimum Cost Queries...

...With Time-Dependent Contraction Hierarchies (TCH)

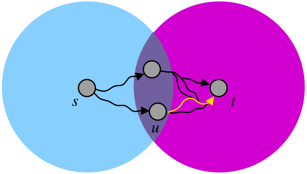
Phase 1: Bidirectional **upward** search:

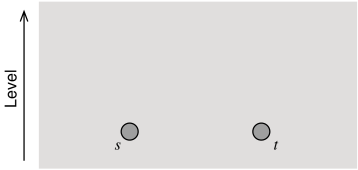
- Forward: multi-label search
- Backward: interval search

⇝ meet in candidate nodes

Phase 2: Downward search

- Forward: muti-label search
- Uses only edges touched by backward/upward search
- $Cost(s, t, \tau_0) = \tau_t + \gamma_t$ first “settled” label of t





Heuristic Minimum Cost Queries...

...With Time-Dependent Contraction Hierarchies (TCH)

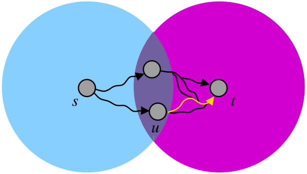
Phase 1: Bidirectional **upward** search:

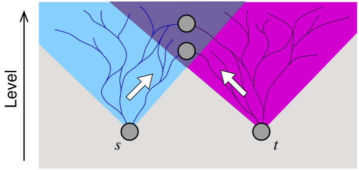
- Forward: multi-label search
- Backward: interval search

⇝ meet in candidate nodes

Phase 2: Downward search

- Forward: muti-label search
- Uses only edges touched by backward/upward search
- $Cost(s, t, \tau_0) = \tau_t + \gamma_t$ first “settled” label of t





Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

Heuristic Minimum Cost Queries...

...With Time-Dependent Contraction Hierarchies (TCH)

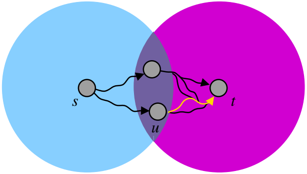
Phase 1: Bidirectional **upward** search:

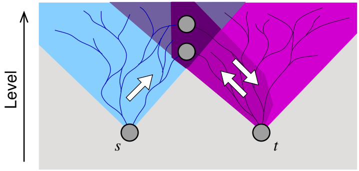
- Forward: multi-label search
- Backward: interval search

⇝ meet in candidate nodes

Phase 2: Downward search

- Forward: muti-label search
- Uses only edges touched by backward/upward search
- $Cost(s, t, \tau_0) = \tau_t + \gamma_t$ first “settled” label of t



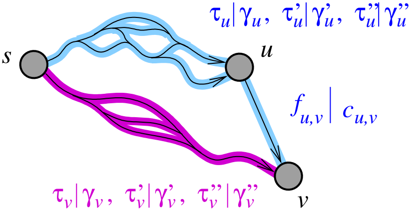


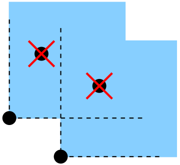
Department of Informatics
Institute of Theoretical Informatics, Algorithms II

Multi-Label search

- Computes all Pareto optimal paths from node s
- Multiple labels per node
- Node labels are pairs $\tau_u|\gamma_u$
- Labels in priority queue instead of nodes

Edge relaxation: $\tau_{\text{new}}|\gamma_{\text{new}} := \tau_u + f_{uv}(\tau_u) \mid \gamma_u + c_{uv}$

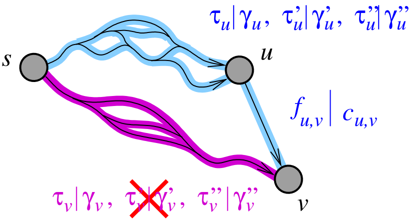


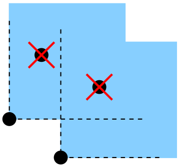


Multi-Label search

- Computes all Pareto optimal paths from node s
- Multiple labels per node
- Node labels are pairs $\tau_u | \gamma_u$
- Labels in priority queue instead of nodes

Edge relaxation: $\tau_{\text{new}} | \gamma_{\text{new}} := \tau_u + f_{uv}(\tau_u) \mid \gamma_u + c_{uv}$



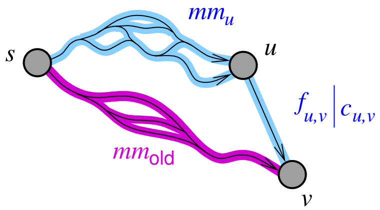


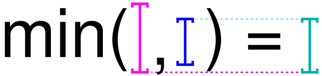
Interval Search

- Dijkstra-like search
- Computes upper and lower bounds of total cost
- Node labels are intervals $mm_u := [a_u, b_u]$

Edge relaxation:

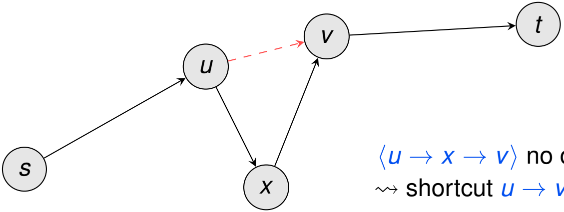
$$mm_{\text{new}} := \min(mm_{\text{old}}, mm_u + [c_{uv} + \min f_{uv},\ c_{uv} + \max f_{uv}])$$





Why is Minimum Cost Query with CH Heuristic?

- **Travel time only:**
  There is always an optimal route with only optimal subroutes
  ⇒ Insert shortcut **iff** $\langle u, x, v \rangle$ is optimal route
  ⇒ Decide locally
  ⇒ EA query always finds existing optimal up-down-route



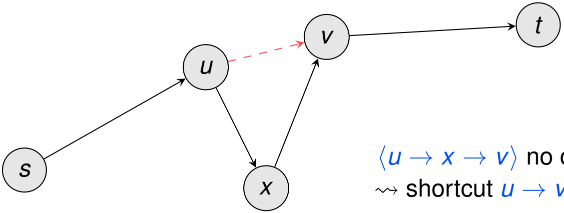
$\langle u \rightarrow x \rightarrow v \rangle$ no optimal route
$\rightsquigarrow$ shortcut $u \rightarrow v$ not inserted

Department of Informatics
Institute of Theoretical Informatics, Algorithmics II

Why is Minimum Cost Query with CH Heuristic?

- **With additional time-invariant costs:**
  Sometimes all optimal s-t-routes have non-optimal subroutes
  $\Rightarrow$ Decide globally or check Pareto optimality

  Both very expensive, so decide locally!
  $\Rightarrow$ Present up-down-routes not necessary optimal
  $\Rightarrow$ Heuristic!



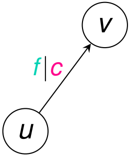
$\langle u \rightarrow x \rightarrow v \rangle$ no optimal route
$\rightsquigarrow$ shortcut $u \rightarrow v$ not inserted

Experiments

Running Time and Error

German road network:

- Nodes: 4.7 million
- Edges: 10.8 million, 7.2 % time-dependent



1. Experiment: Energy consumption

- $c \propto$ distance (estimates energy consumption)
- 1 km costs 0.1€
- 1 hour costs 5€, 10€, or 20€ ($\rightsquigarrow$ three instances)

$\Rightarrow c := \lambda \cdot \mathit{distance}$ where $\lambda \in \{0.72, 0.36, 0.18\}$

2. Experiment: Energy consumption and tolls

- Same as above
- But: motorway edges cost 0.2€ instead 0.1 €

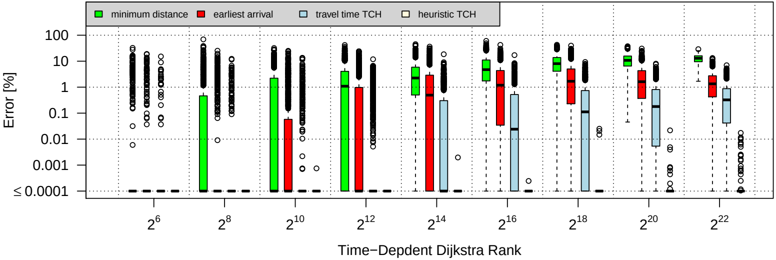
Experiment 1: Energy Consumption

hourly rate [€]	Space [B/n]	Preprocessing (8 cores) [h:m]	Query [ms]	Error [%] max.	Error [%] avg.
5	1 481	0:28	4.92	0.09	0.00
10	1 316	0:26	4.22	0.03	0.00
20	1 212	0:25	3.51	0.01	0.00

- Error compared to multi label A*
  Heuristics obtained from preceding backward interval search
- Very fast query
- Nearly no error
- But: Needs much space

Experiment 1: Energy Consumption

Hourly Rate = 5€



Much smaller error than

- Minimum distance routes
- Earliest arrival routes
- Routes from minimum cost query in travel time TCH

Note: Even some outliers can result in bad publicity!

2. Experiment: With Motorway Tolls

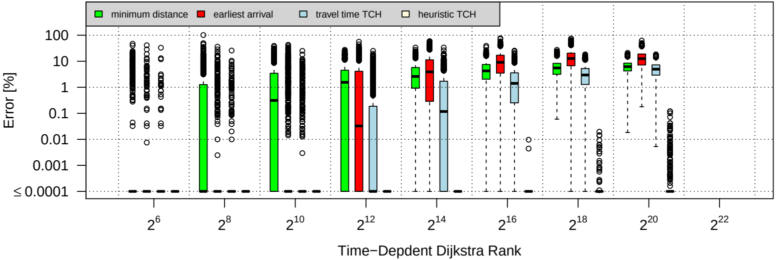
hourly rate [€]	Space [B/n]	Preprocessing (8 cores) [h:m]	Query [ms]	Error [%] max.	Error [%] avg.
5	1 863	1:06	14.96		
10	2 004	1:16	40.96		
20	1 659	0:46	27.90		

Harder instances:

- Multi label A* no longer feasible $\rightsquigarrow$ error unknown
- Slower query (though still not bad)
- Needs even more space

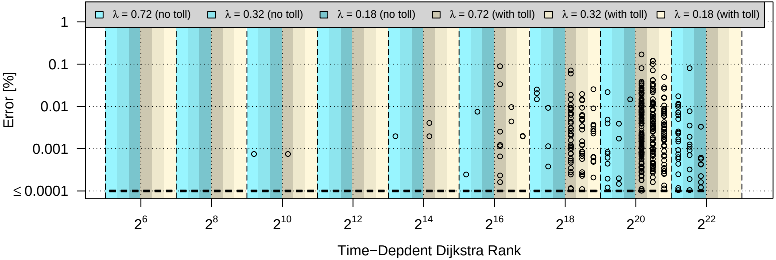
2. Experiment: With Motorway Tolls

Hourly Rate = 10€



- Multi-label A* terminated up to rank 2^{20}
- Very small error
- Again: Minimum distance, earliest arrival, and TCH routes worse

Summary of Measured Errors



- Error not significantly away from 0
- Outliers not serious

Conclusions

- Minimum cost queries NP-hard in theory
- Heuristic TCHs are very fast: 5 ms and 41 ms
- Errors negligible
- But: space consuming
- Multi-label A* needs 2.3 s (no tolls)

Future Work

- Reduce space (techniques from ATCH [Batz et al. 2010])
- Fast heuristic cost profile search
- Exact Hierarchy
- More general objective functions

Future Work

- Reduce space (techniques from ATCH [Batz et al. 2010])
- Fast heuristic cost profile search
- Exact Hierarchy
- More general objective functions

Questions?