

# Distance oracles for timetable graphs

Dištančné orákula pre grafy reprezentujúce cestovné poriadky

František Hajnovič

FMFI UK

April 6, 2013

Supervisor: *doc. RNDr. Rastislav Kráľovič PhD.*

# Content

## 1 Introduction

## 2 Contribution

- Underlying shortest paths
- USP-OR
- USP-OR-A
- Results and comparison
- Neural networks

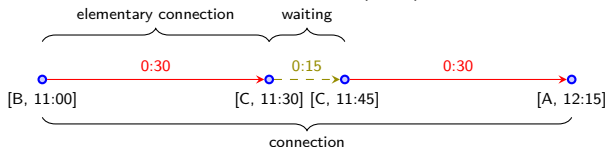
## 3 Conclusion

# Introduction

## Introduction

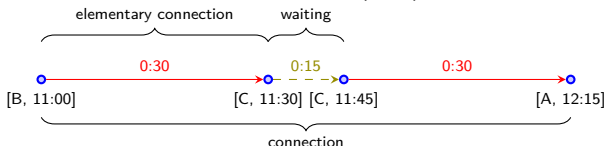
# What is it about?

- Given a timetable, we query -  $(a, t, b)$  - for
  - Earliest arrival (EA)** -  $t_{(a,t,b)}^*$
  - Optimal connection (OC)** -  $c_{(a,t,b)}^*$

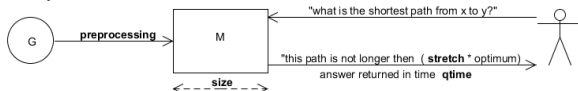


# What is it about?

- Given a timetable, we query -  $(a, t, b)$  - for
  - Earliest arrival (EA)** -  $t_{(a,t,b)}^*$
  - Optimal connection (OC)** -  $c_{(a,t,b)}^*$



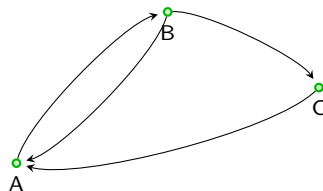
- Motivation: large-scale timetable search engines (*cp.sk*, *imhd.sk...*)
- Approach: (distance) oracle-based approach [TZ05] - pre-computation



# Timetable and underlying graph

Place		Time	
From	To	Departure	Arrival
A	B	10:00	10:45
B	C	11:00	11:30
B	C	11:30	12:10
B	A	11:20	12:30
C	A	11:45	12:15

**Table : Timetable** - a set of **elementary connections** (between pairs of **cities**)

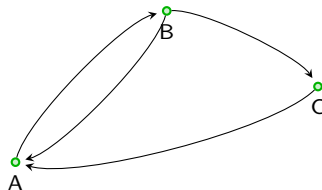


**Figure : Underlying graph**

# Timetable and underlying graph

Place		Time	
From	To	Departure	Arrival
A	B	10:00	10:45
B	C	11:00	11:30
B	C	11:30	12:10
B	A	11:20	12:30
C	A	11:45	12:15

**Table : Timetable** - a set of **elementary connections** (between pairs of **cities**)



**Figure : Underlying graph**

- Goals:
  - Devise methods to tackle EA/OC problem
  - Analyse properties of timetables

# Contribution

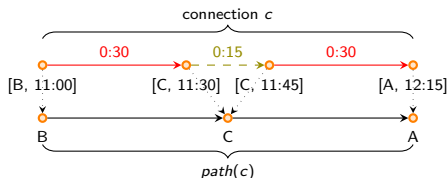
## Contribution



## Underlying shortest paths

## Idea

- “Usually we go through the same sequence of cities”



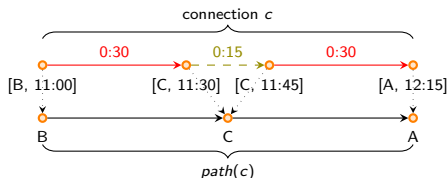
Odchod	Prichod	Dĺžka cesty*	Použité linky	Zóny	Cena*
21:59	22:10	11 min	95	-	0,70 €
22:09	22:20	11 min	95	-	0,70 €
22:19	22:30	11 min	95	-	0,70 €
22:29	22:40	11 min	95	-	0,70 €
22:39	22:50	11 min	95	-	0,70 €

- $p$  is **USP**  $\iff \exists t : path(c_{(a,t,b)}^*) = p$
- we have USP  $\rightarrow$  reconstruct  $c_{(a,t,b)}^*$

## Underlying shortest paths

## Idea

- “Usually we go through the same sequence of cities”



- $p$  is **USP**  $\iff \exists t : path(c_{(a,t,b)}^*) = p$
- we have USP  $\rightarrow$  reconstruct  $c_{(a,t,b)}^*$
- Overtaking** [MHSWZ07] causes problems, but can be easily removed

Odchod	Prichod	Dĺžka cesty*	Použité linky	Zóny	Cena*
21:59	22:10	11 min	95	-	0,70 €
22:09	22:20	11 min	95	-	0,70 €
22:19	22:30	11 min	95	-	0,70 €
22:29	22:40	11 min	95	-	0,70 €
22:39	22:50	11 min	95	-	0,70 €

Name	Overtaken edges (%)
air01	1%
cpru	2%
cpza	2%
montr	1%
sncf	2%
sncf-ter	2%
sncf-inter	8%
zsr	0%

## USP-OR

*USP-OR*

- Pre-compute *all conn.* - space  $\mathcal{O}(h n^2 \gamma)$ 
  - $\gamma$  - the average OC size
  - daily height usually  $200 < h < 800$

## USP-OR

## USP-OR

- Pre-compute *all conn.* - space  $\mathcal{O}(h n^2 \gamma)$ 
  - $\gamma$  - the average OC size
  - daily height usually  $200 < h < 800$
- Pre-compute *all USPs* - space  $\mathcal{O}(\tau n^2 \gamma)$ 
  - $\tau(x, y)$  - # of USPs between  $x$  and  $y$

Name	avg $\tau$	max $\tau(x, y)$	$\gamma$
<i>air01</i>	5.8	30	3.0
<i>cpru</i>	7.0	64	13.8
<i>cpza</i>	5.1	42	11.1
<i>montr</i>	4.3	30	20.3
<i>sncf</i>	4.3	24	10.5
<i>sncf-inter</i>	0.6	19	7.9
<i>sncf-ter</i>	6.1	33	10.8
<i>zsr</i>	2.5	19	13.7

Table : Daily, 200 station timetables

USP-OR	prep	size	qtime	stretch
guaranteed	$\mathcal{O}(hn^2(\log n + \delta))$	$\mathcal{O}(\tau n^2 \gamma)$	avg. $\mathcal{O}(\tau \gamma)$	1
$\tau$ const., $\gamma \leq \sqrt{n}$ , $\delta \leq \log n$	$\mathcal{O}(hn^2 \log n)$	$\mathcal{O}(n^{2.5})$	avg. $\mathcal{O}(\sqrt{n})$	1

Table :  $\delta$  - the UG density  $\frac{m}{n}$

## USP-OR

## USP-OR

- Pre-compute *all conn.* - space  $\mathcal{O}(h n^2 \gamma)$ 
  - $\gamma$  - the average OC size
  - daily height usually  $200 < h < 800$
- Pre-compute *all USPs* - space  $\mathcal{O}(\tau n^2 \gamma)$ 
  - $\tau(x, y)$  - # of USPs between  $x$  and  $y$

Name	avg $\tau$	max $\tau(x, y)$	$\gamma$
<i>air01</i>	5.8	30	3.0
<i>cpru</i>	7.0	64	13.8
<i>cpza</i>	5.1	42	11.1
<i>montr</i>	4.3	30	20.3
<i>sncf</i>	4.3	24	10.5
<i>sncf-inter</i>	0.6	19	7.9
<i>sncf-ter</i>	6.1	33	10.8
<i>zsr</i>	2.5	19	13.7

Table : Daily, 200 station timetables

USP-OR	prep	size	qtime	stretch
guaranteed	$\mathcal{O}(hn^2(\log n + \delta))$	$\mathcal{O}(\tau n^2 \gamma)$	avg. $\mathcal{O}(\tau \gamma)$	1
$\tau$ const., $\gamma \leq \sqrt{n}$ , $\delta \leq \log n$	$\mathcal{O}(hn^2 \log n)$	$\mathcal{O}(n^{2.5})$	avg. $\mathcal{O}(\sqrt{n})$	1

Table :  $\delta$  - the UG density  $\frac{m}{n}$ 

- Space  $\mathcal{O}(n^{2.5})$  too big anyway

## USP-OR

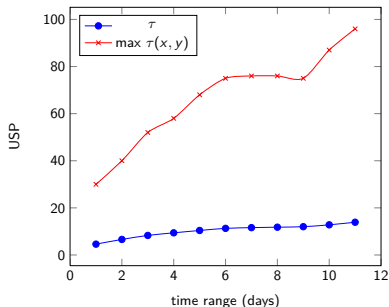
*USP-OR* -  $\tau$  evolution

Figure : Changing of  $\tau$  with increased time range in *air01* dataset. 1 day = about 800 in height

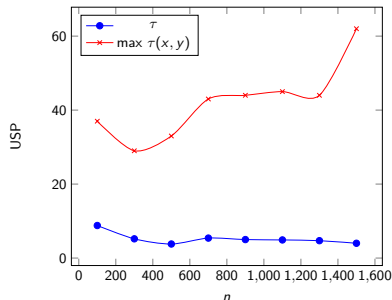
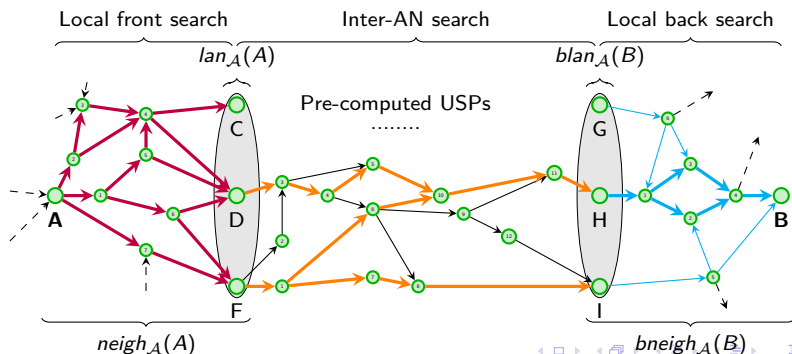


Figure : Changing of  $\tau$  with increased # of stations in *snCF* dataset

## USP-OR-A

## USP-OR-A

- Pre-compute USPs only among *some* cities in UG:  $(r_1, r_2, r_3)$  access node set
  - Small size:  $|\mathcal{A}| \leq r_1 \cdot \sqrt{n}$
  - Small node neighbourhoods:  $(\text{avg } |\text{neigh}(x)||)^2 \leq r_2 \cdot n$
  - Few local access nodes:  $|\text{lan}_{\mathcal{A}}(x)| \leq r_3$



## USP-OR-A

*USP-OR-A* and access nodes

<i>USP-OR-A</i>	guaranteed	$\tau, r_1, r_2, r_3$ const., $\gamma \leq \sqrt{n}$ , $\delta \leq \log n$
<i>prep</i>	$\mathcal{O}(f(n) + (r_1 + r_2)(\delta + \log n)hn^{1.5})$	$\mathcal{O}(f(n) + hn^{1.5} \log n)$
<i>size</i>	$\mathcal{O}(r_2 n^{1.5} + r_1^2 \tau_{\mathcal{A}} \gamma_{\mathcal{A}} n)$	$\mathcal{O}(n^{1.5})$
<i>qtime</i>	avg. $\mathcal{O}(r_2 r_3 \sqrt{n} (\log(r_2 n) + \delta) + r_3^2 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
<i>stretch</i>	1	1

- Much depends on choosing good AN set

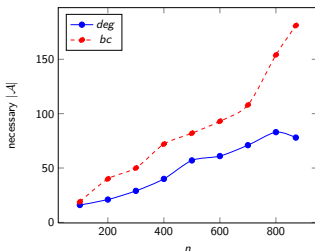
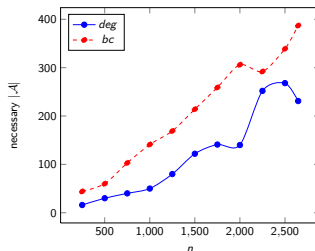


## USP-OR-A

## USP-OR-A and access nodes

USP-OR-A	guaranteed	$\tau, r_1, r_2, r_3$ const., $\gamma \leq \sqrt{n}$ , $\delta \leq \log n$
prep	$\mathcal{O}(f(n) + (r_1 + r_2)(\delta + \log n)hn^{1.5})$	$\mathcal{O}(f(n) + hn^{1.5} \log n)$
size	$\mathcal{O}(r_2 n^{1.5} + r_1^2 \tau_{\mathcal{A}} \gamma_{\mathcal{A}} n)$	$\mathcal{O}(n^{1.5})$
qtime	avg. $\mathcal{O}(r_2 r_3 \sqrt{n}(\log(r_2 n) + \delta) + r_3^2 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
stretch	1	1

- Much depends on choosing good AN set
- Minimize  $|\mathcal{A}|$  s.t.  $\forall x : |\text{neigh}_{\mathcal{A}}(x)| \leq \sqrt{n} \rightarrow \text{NP-complete}$
- Choose ANs based on degree/betweenness centrality

Figure : Dataset *cpru* ( $\sqrt{n} \approx 30$ )Figure : Dataset *snf* ( $\sqrt{n} \approx 51$ )

# Locsep

- Select AN that locally separates many vertices

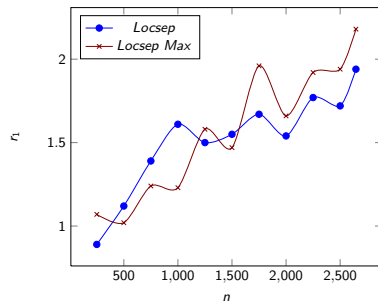
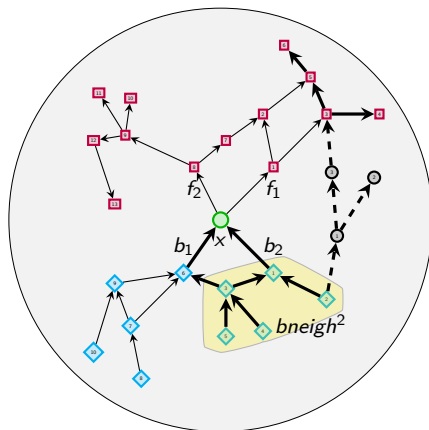


Figure : Dataset *snctf*. Found AN set with size  $|\mathcal{A}| = r_1 \sqrt{n} \approx r_1 51$ .

## Results and comparison

## Results

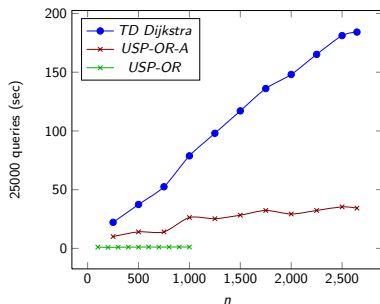


Figure : USP-OR-A + Locsep vs. USP-OR vs. TD Dijkstra on *sncf* dataset. Changing  $n$ .

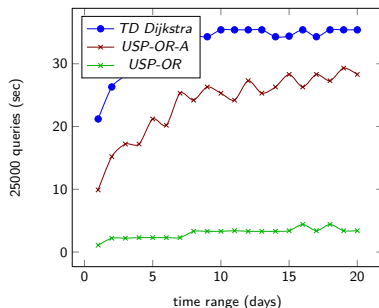


Figure : USP-OR-A + Locsep vs. USP-OR vs. TD Dijkstra on *zsr* dataset. Changing time range.

# Existing methods

- **Time-dependent SHARC**  
[Del08], **Time-dependent CH**  
[BDSV09]

- Speed-ups of about 26 / 1500, respectively (EA only)
- Meant for time-dependent routing in road networks

- **Time-expanded approach**  
[DPW09]

- Speed-ups of about 56
- Remodelling unimportant stations
- Theory vs. practice difference: transfers, cost of travel...

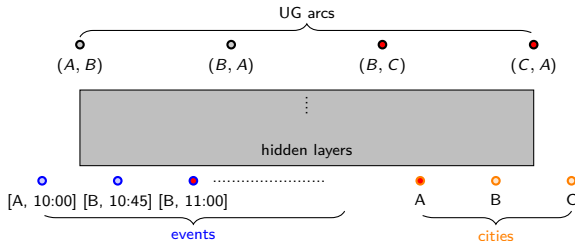
Name	USP-OR	USP-OR-A
<i>cpru</i>	14.5	1.7
<i>cpza</i>	14.3	1.7
<i>montr</i>	8.8	1.5
<i>sncf</i>	64.8	5.4
<i>sncf-inter</i>	27.0	3.6
<i>sncf-ter</i>	78.3	6.3
<i>zsr</i> (daily)	19.3	2.14

**Table :** Speed-up of *USP-OR* and *USP-OR-A* with *Locsep*

## Neural networks

## Neural network approaches

- Multi-layer perceptron, back propagation
- Input layer = events + cities.
- Output layer = arcs of UG  $\rightarrow$  USP



- Tendency to remember USPs
- Long training times

Name	Conn.	Found	Was optimum (%)
air01	931	573	18.7%
cpru	481	281	48%
montr	527	346	86.7%
zsr	672	307	76.2%

Table : Timetables with 30 cities

# Conclusion

## Conclusion

# Conclusion

- Trying out **novel approaches** to find optimal connections in timetables
  - *USP-OR*: Exact and very quick answers (speed-up  $\approx 60$ ) but high space
  - *USP-OR-A*: Exact and quick answers (speed-up  $\approx 6$ ) less space-consuming
  - *NN*: Problem too challenging for NN/try different types of network
- **Application** created to carry out **analysis** of real-world timetables:
  - Degrees, connectivity, BC, high. dimension, overtaking, USPs...
  - Running & evaluating tests of oracles



Figure : It's *blazing* fast!

# Bibliography I

- [BDSV09] Gernot Veit Batz, Daniel Delling, Peter Sanders, and Christian Vetter. Time-dependent contraction hierarchies. In Irene Finocchi and John Hershberger, editors, *ALENEX*, pages 97–105. SIAM, 2009.
- [Del08] Daniel Delling. Time-dependent sharc-routing. In Dan Halperin and Kurt Mehlhorn, editors, *ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 332–343. Springer, 2008. ISBN 978-3-540-87743-1.
- [DPW09] Daniel Delling, Thomas Pajor, and Dorothea Wagner. Engineering time-expanded graphs for faster timetable information. In Ravindra Ahuja, Rolf Möhring, and Christos Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 182–206. Springer Berlin / Heidelberg, 2009. ISBN 978-3-642-05464-8.
- [MHSWZ07] Matthias Müller-Hannemann, Frank Schulz, Dorothea Wagner, and Christos Zaroliagis. *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, chapter Timetable Information: Models and Algorithms, pages 67 – 90. Springer, 2007.
- [TZ05] Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005.



# Thank you for the attention

