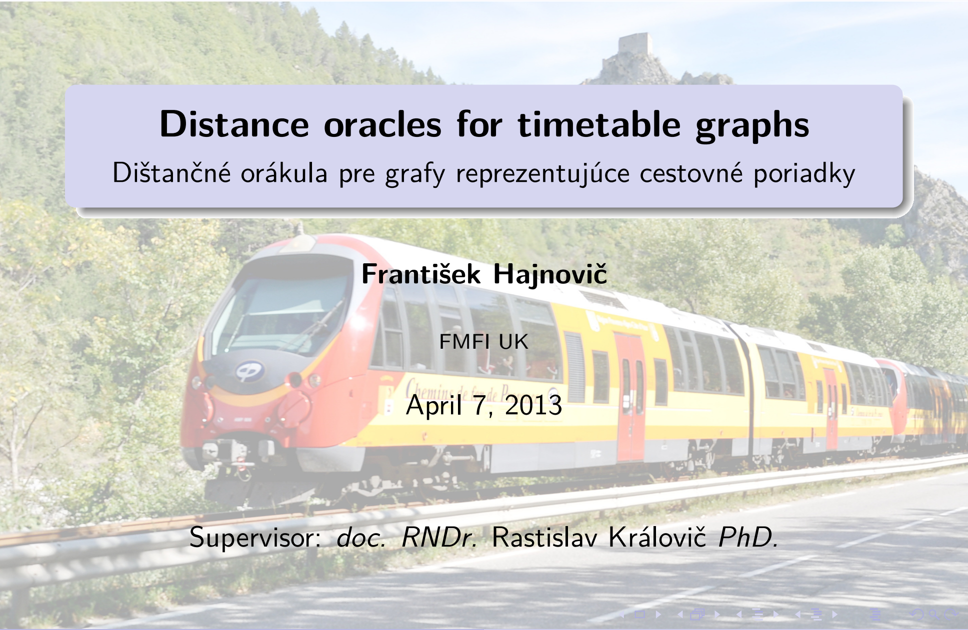


Distance oracles for timetable graphs

Dištančné orákula pre grafy reprezentujúce cestovné poriadky

František Hajnovič

FMFI UK

April 7, 2013

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

Content

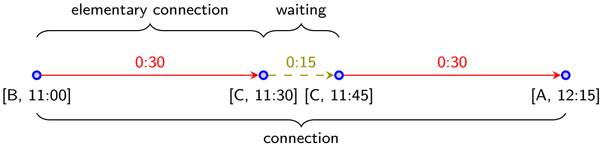
1. Introduction
2. Contribution
   - Underlying shortest paths
   - USP-OR
   - USP-OR-A
   - Results and comparison
   - Neural networks
3. Conclusion

Introduction

Introduction

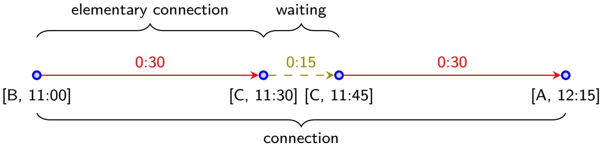
What is it about?

- Given a timetable, we query - (a, t, b) - for
  - **Earliest arrival** (EA) - $\boldsymbol{t^*_{(a,t,b)}}$
  - **Optimal connection** (OC) - $\boldsymbol{c^*_{(a,t,b)}}$

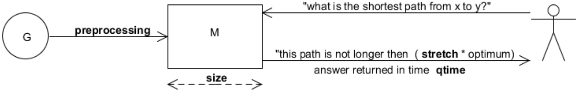


What is it about?

- Given a timetable, we query - (a, t, b) - for
  - **Earliest arrival** (EA) - $\boldsymbol{t^*_{(a,t,b)}}$
  - **Optimal connection** (OC) - $\boldsymbol{c^*_{(a,t,b)}}$



- 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**)

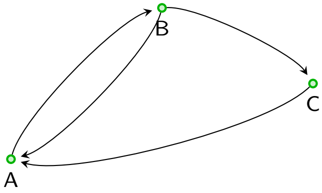


Figure : **Underlying graph**

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**)

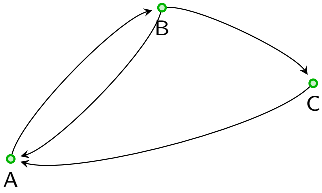


Figure : **Underlying graph**

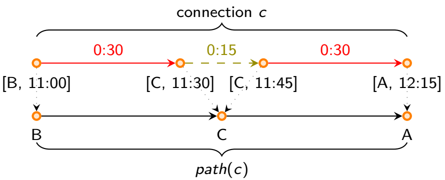
- Goals:
  - Devise methods to tackle EA/OC problem
  - Analyse properties of timetables

Contribution

Contribution

Idea

- *"Usually we go through the same sequence of cities"*

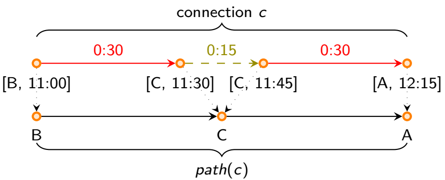


Odchod	Príchod	Dĺžka cesty*	Použité linky	Zóny	Cena*
21:59	22:10	11 min	95	-	0,70 €
22:09	22:20	11 min	95	-	0,70 €
22:19	22:30	11 min	95	-	0,70 €
22:29	22:40	11 min	95	-	0,70 €
22:39	22:50	11 min	95	-	0,70 €

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

Idea

- *"Usually we go through the same sequence of cities"*



Odchod	Príchod	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 $c^*_{(a,t,b)}$
- **Overtaking** [MHSWZ07] causes problems, but can be easily removed

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

USP-OR

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

USP-OR

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

Name	**avg τ**	**max $\tau(x,y)$**	γ
air01	5.8	30	3.0
cpru	7.0	64	13.8
cpza	5.1	42	11.1
montr	4.3	30	20.3
sncf	4.3	24	10.5
sncf-inter	0.6	19	7.9
sncf-ter	6.1	33	10.8
zsr	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
$\boldsymbol{\tau}$ **const.,** $\boldsymbol{\gamma \le \sqrt{n}}$**,** $\boldsymbol{\delta \le \log n}$	$\mathcal{O}(hn^2\log n)$	$\mathcal{O}(n^{2.5})$	avg. $\mathcal{O}(\sqrt{n})$	1

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

USP-OR

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

Name	avg τ	max $\tau(x,y)$	γ
air01	5.8	30	3.0
cpru	7.0	64	13.8
cpza	5.1	42	11.1
montr	4.3	30	20.3
sncf	4.3	24	10.5
sncf-inter	0.6	19	7.9
sncf-ter	6.1	33	10.8
zsr	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
$\boldsymbol{\tau}$ **const.,** $\gamma \le \sqrt{n}$**,** $\delta \le \log n$	$\mathcal{O}(hn^2 \log n)$	$\mathcal{O}(n^{2.5})$	avg. $\mathcal{O}(\sqrt{n})$	1

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

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

USP-OR - τ evolution

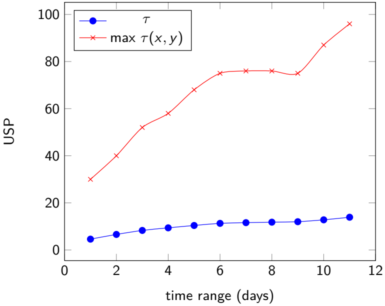


Figure : Changing of τ with increased time range in *air01* dataset. 1 day = about 800 in height

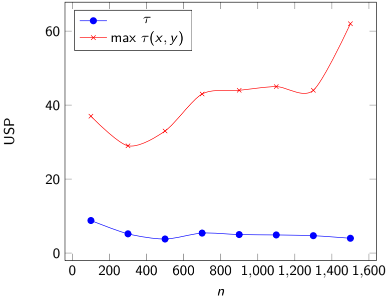
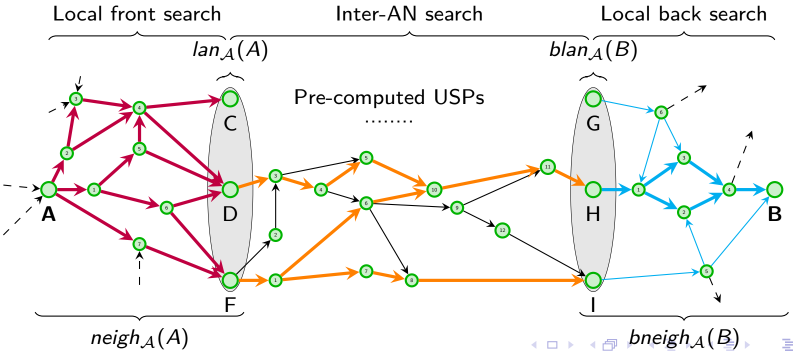


Figure : Changing of τ with increased # of stations in *sncf* dataset

USP-OR-A

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



USP-OR-A and access nodes

USP-OR-A	**guaranteed**	**τ, 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

USP-OR-A and access nodes

USP-OR-A	guaranteed	τ, 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. $\forall x : |neigh_{\mathcal{A}}(x)| \leq \sqrt{n} \rightarrow$ NP-complete
- Choose ANs based on degree/betweenness centrality

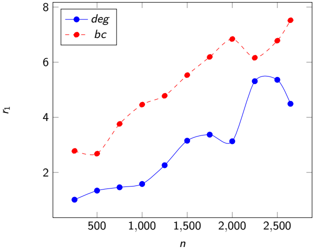


Figure : $|\mathcal{A}| = r_1\sqrt{n} \approx r_1 51$ (*sncf*).

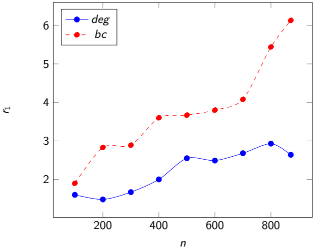
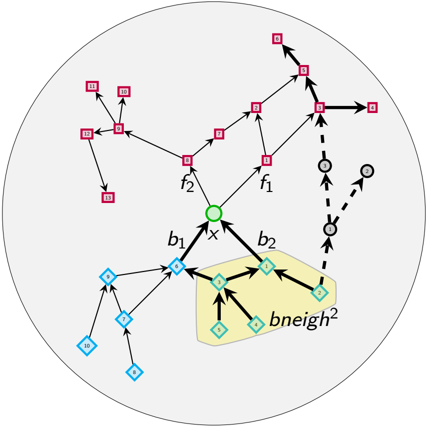


Figure : $|\mathcal{A}| = r_1\sqrt{n} \approx r_1 30$ (*cpru*).

Locsep

- Select AN that locally separates many vertices



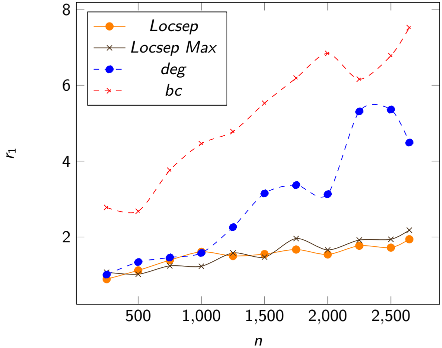


Figure : $|\mathcal{A}| = r_1\sqrt{n} \approx r_1 51$ (*sncf*).

Results

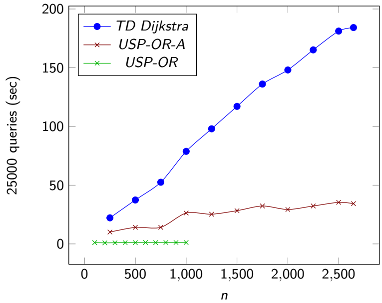


Figure : *USP-OR-A* + *Locsep* vs. *USP-OR* vs. *TD Dijkstra* on *sncf* dataset. Changing *n*.

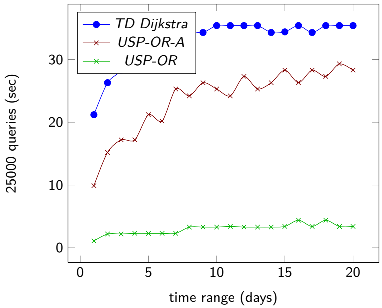


Figure : *USP-OR-A* + *Locsep* vs. *USP-OR* vs. *TD Dijkstra* on *zsr* dataset. Changing time range.

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

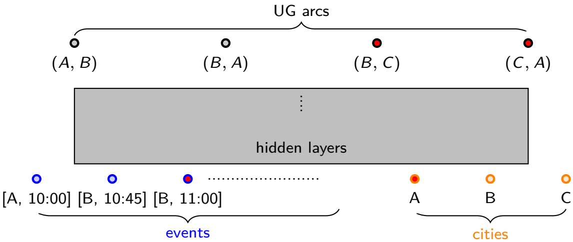
Name	*USP-OR*	*USP-OR-A*
cpru	14.5	1.7
cpza	14.3	1.7
montr	8.8	1.5
sncf	64.8	5.4
sncf-inter	27.0	3.6
sncf-ter	78.3	6.3
zsr (daily)	19.3	2.14

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

- Theory vs. practice difference: transfers, cost of travel...

Neural network approaches

- Multi-layer perceptron, back propagation
- Input layer = events + cities.
- Output layer = arcs of UG → 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 ≈ 60) but high space
  - *USP-OR-A*: Exact and quick answers (speed-up ≈ 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