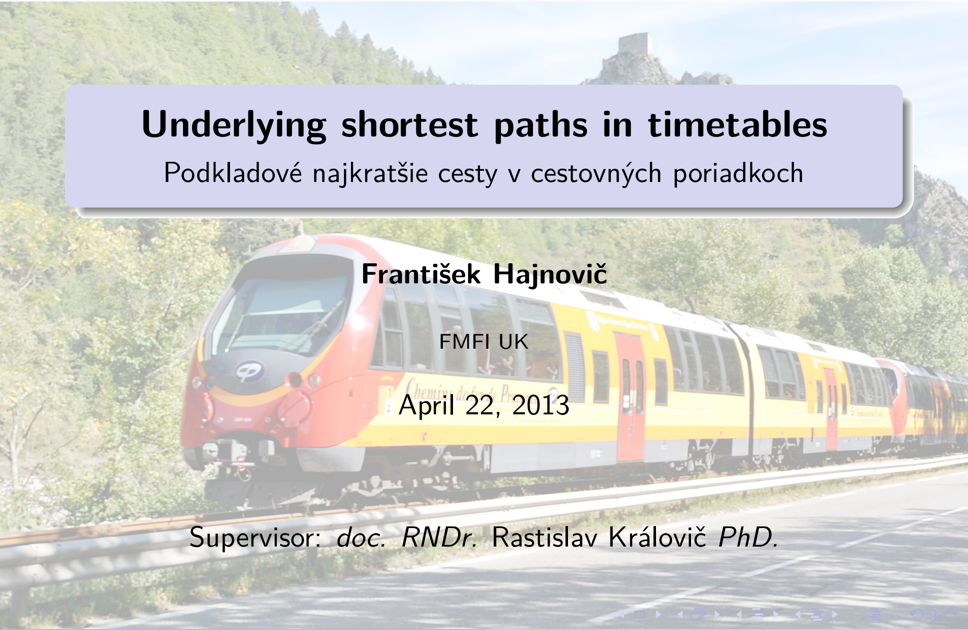


Underlying shortest paths in timetables

Podkladové najkratšie cesty v cestovných poriadkoch

František Hajnovič

FMFI UK

April 22, 2013

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

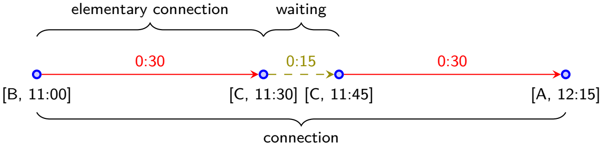
Content

Introduction

Introduction

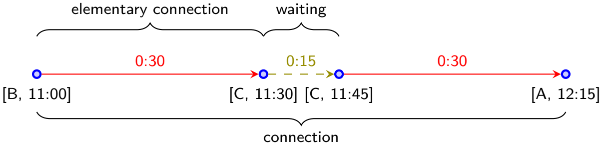
What is it about?

- Given a timetable, we query for **optimal connection** (OC): $(a, t, b) \rightarrow c^*_{(a,t,b)}$

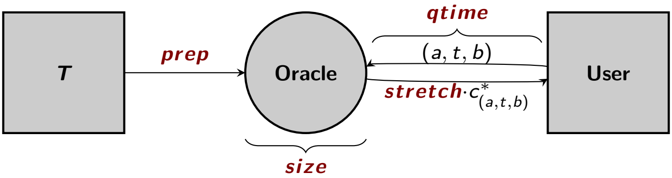


What is it about?

- Given a timetable, we query for **optimal connection** (OC): $(a, t, b) \rightarrow c^*_{(a,t,b)}$



- 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**).

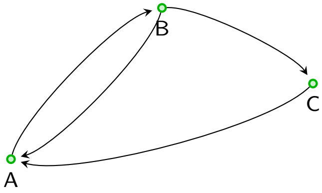


Figure : **Underlying graph**.

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**).

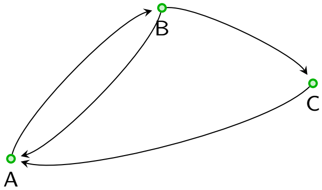


Figure : **Underlying graph**.

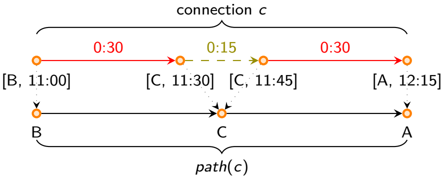
- Goals:
  - Devise methods to tackle EA/OC problem
  - Analyse properties of real-world timetables

Contribution

Contribution

Idea

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

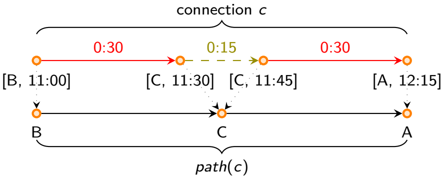


Odchod	Príchod	Dĺžka cesty*	Použité linky	Zóny	Cena*
21:59	22:10	11 min	95	-	0,70 €
22:09	22:20	11 min	95	-	0,70 €
22:19	22:30	11 min	95	-	0,70 €
22:29	22:40	11 min	95	-	0,70 €
22:39	22:50	11 min	95	-	0,70 €

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

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 p: $expand(p) \rightarrow c^*_{(a,t,b)}$
- **Overtaking** [MHSWZ07] causes problems, but can be easily removed

Name	Overtaken edges (%)
air01	1%
cpsk	2%
gb-coach	1%
gb-train	0%
montr	1%
sncf	1%
sncf-ter	1%
sncf-inter	6%
zsr	0%

USP-OR

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

USP-OR

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

Name	$\boldsymbol{\tau}$	γ	$\boldsymbol{\delta}$	$\boldsymbol{h}$
air01	7.5	2.1	32.9	112.7
cpsk	12.1	15.9	3.9	50.7
gb-coach	5.3	5.3	5.6	48
gb-train	10.3	7.6	5.7	129.6
montr	5.1	21.0	1.9	35.0
sncf	5.6	8.6	3.4	42.4
sncf-inter	2.6	13.4	1.1	20.8
sncf-ter	5.6	12.7	3.3	34.0
zsr	3.7	13.5	2.7	21.6

Table : Daily, 200 station timetable subsets.

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.**, $\gamma \leq \sqrt{\boldsymbol{n}}$, $\boldsymbol{\delta \leq \log n}$	$\mathcal{O}(hn^2 \log n)$	$\mathcal{O}(n^{2.5})$	avg. $\mathcal{O}(\sqrt{n})$	1

USP-OR - τ evolution

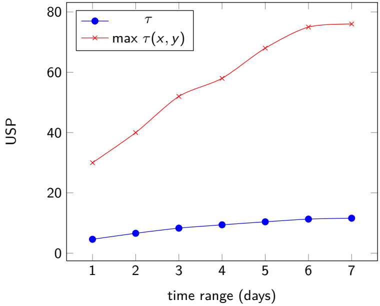


Figure : Changing of τ with increased time range in *gb-sncf-200* dataset.

USP-OR - τ evolution

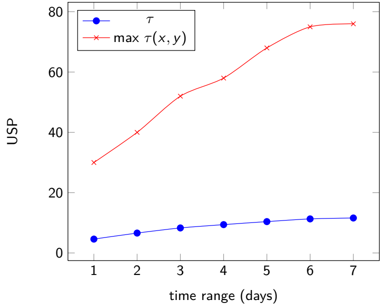


Figure : Changing of τ with increased time range in *gb-sncf-200* dataset.

Segmentation of the timetable to days.

USP-OR - τ evolution

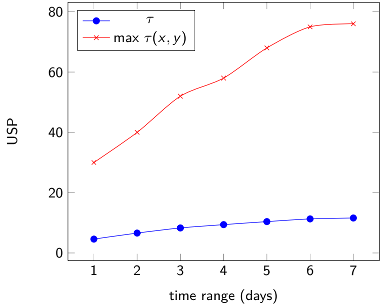


Figure : Changing of τ with increased time range in *gb-sncf-200* dataset.
Segmentation of the timetable to days.

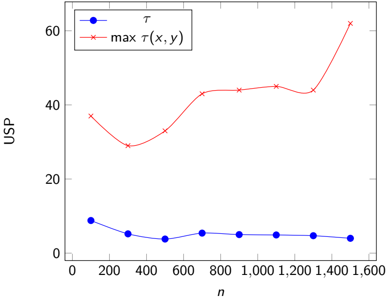


Figure : Changing of τ with increased # of stations in *gb-coach* dataset.

USP-OR - τ evolution

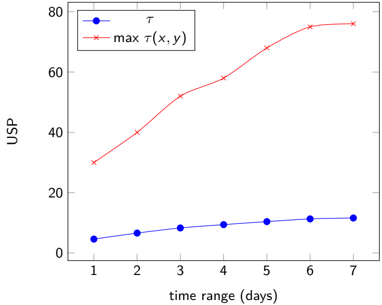


Figure : Changing of τ with increased time range in *gb-sncf-200* dataset.
Segmentation of the timetable to days.

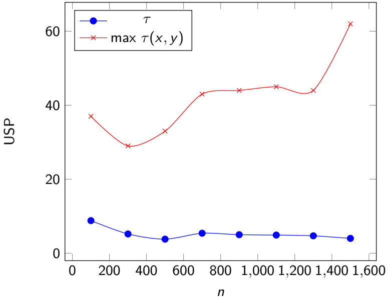
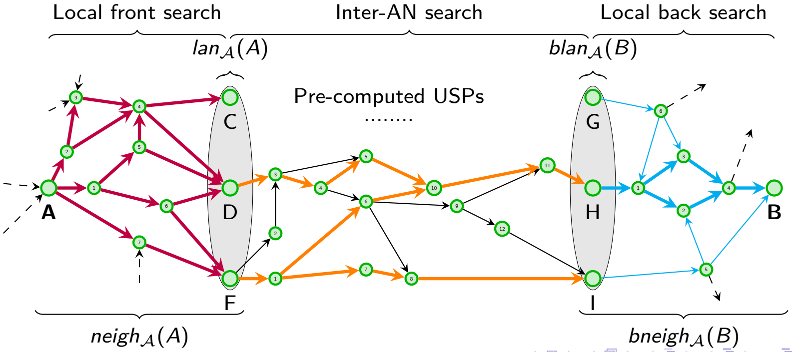


Figure : Changing of τ with increased # of stations in *gb-coach* dataset.

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

USP-OR-A

- Pre-compute USPs only among *some* cities in UG: $(\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3})$ **access node set** $\mathcal{A}$
  - 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: $avg(|lan_{\mathcal{A}}(x)|)^2 \leq r_3$



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. $\mathcal{O}(r_2 r_3 \sqrt{n}(\log(r_2 n) + \delta) + r_3 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
stretch	1	1

USP-OR-A and access nodes

USP-OR-A	**guaranteed**	$\boldsymbol{\tau, r_1, r_2, r_3}$ **const.,** $\boldsymbol{\gamma \leq \sqrt{n}}$**,** $\boldsymbol{\delta \leq \log n}$
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. $\mathcal{O}(r_2 r_3 \sqrt{n}(\log(r_2 n) + \delta) + r_3 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
stretch	1	1

- Minimize $|\mathcal{A}|$ s.t. $\forall x : |neigh_{\mathcal{A}}(x)| \leq \sqrt{n} \rightarrow$ NP-complete

USP-OR-A and access nodes

USP-OR-A	**guaranteed**	$\boldsymbol{\tau, r_1, r_2, r_3}$ **const.,** $\boldsymbol{\gamma \leq \sqrt{n}}$**,** $\boldsymbol{\delta \leq \log n}$
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. $\mathcal{O}(r_2 r_3 \sqrt{n}(\log(r_2 n) + \delta) + r_3 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
stretch	1	1

- Minimize $|\mathcal{A}|$ s.t. $\forall x : |neigh_{\mathcal{A}}(x)| \leq \sqrt{n} \rightarrow$ NP-complete

- Choose ANs based on degree/betweenness centrality

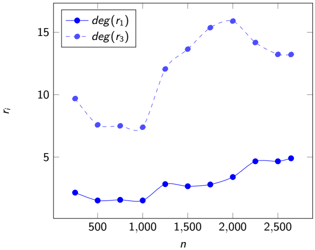


Figure : *sncf*, $r_2 \leq 1$.

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. $\mathcal{O}(r_2 r_3 \sqrt{n}(\log(r_2 n) + \delta) + r_3 \tau_{\mathcal{A}} \gamma_{\mathcal{A}})$	avg. $\mathcal{O}(\sqrt{n} \log n)$
stretch	1	1

- Minimize $|\mathcal{A}|$ s.t. $\forall x : |neigh_{\mathcal{A}}(x)| \leq \sqrt{n} \rightarrow$ NP-complete

- Choose ANs based on degree/betweenness centrality

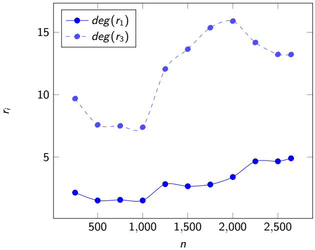
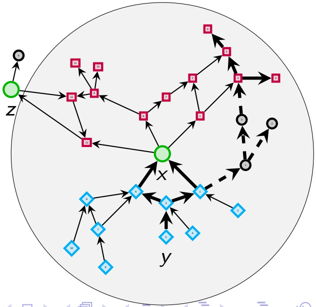


Figure : *sncf*, $r_2 \leq 1$.

- Select AN that locally separates many vertices



Locsep

- Select AN that locally separates many vertices

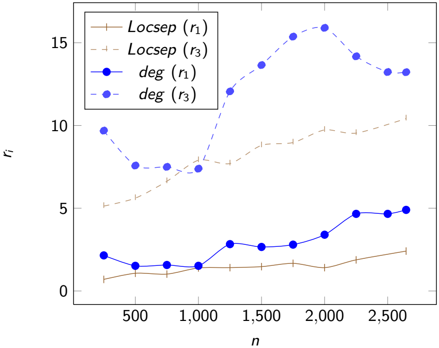


Figure : *sncf*, $r_2 \leq 1$.

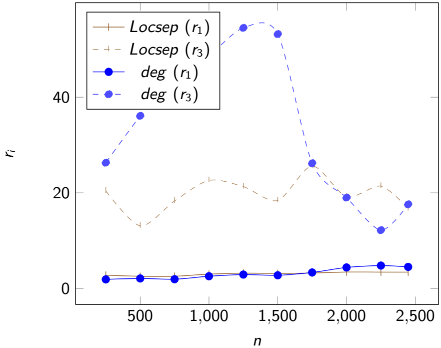


Figure : *gb-coach*, $r_2 \leq 1$.

Performance

- Time-dependent Dijkstra's algorithm with Fibonacci heap priority queue $\mathcal{O}(m + n \log n)$

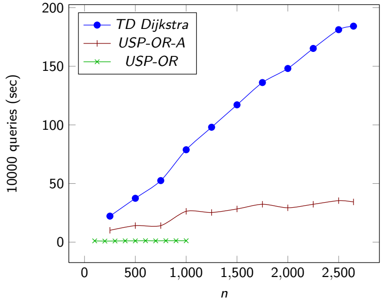


Figure : query time, *sncf*

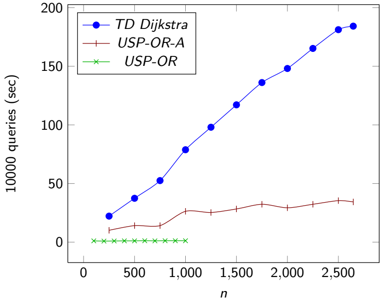


Figure : query time, *gb-coach*

Performance

	Dataset	*sncf*		*gb-coach*		*gb-train*	
	Time range	(1 day)	(1 week)	(1 day)	(1 week)	(1 day)	(1 week)
USP-OR	***n***	1000	700	1000	700	700	500
	Speed-up	0	0	0	0	0	0
	Size-up	0	0	0	0	0	0
USP-OR-A	***n***	2608	2646	2406	2448	2550	2555
	Speed-up	0	6.3	0	8.8	0	2.8
	Size-up	0	7.8	0	10.4	0	5.2

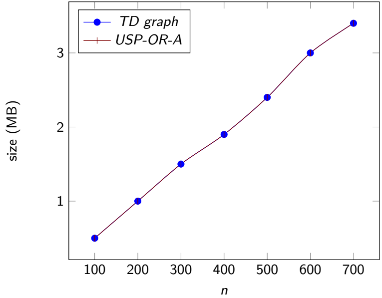


Figure : *USP-OR* oracle size, *sncf*

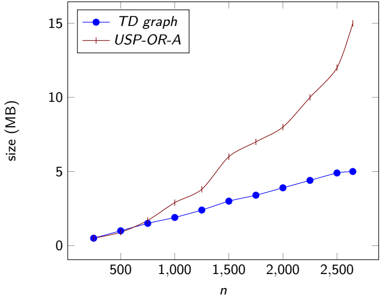


Figure : *USP-OR-A* oracle size, *sncf*

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
- **Engineering time-expanded graphs...** [DPW09]
  - Max speed-up of 56 (Railways with 30000 stations!)
  - Remodelling unimportant stations in TE graphs

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
- **Engineering time-expanded graphs...** [DPW09]
    - Max speed-up of 56 (Railways with 30000 stations!)
    - Remodelling unimportant stations in TE graphs
- Theory vs. practice difference
    - More complicated (transfers, cost of travel...)
    - Focus on one given dataset

Conclusion

Conclusion

Conclusion

- **Application** created to carry out **analysis** of real-world timetables:
  - Degrees, connectivity, BC, highway dimension, overtaking, USPs...
  - Running & evaluating tests of oracles

Conclusion

- **Application** created to carry out **analysis** of real-world timetables:
  - Degrees, connectivity, BC, highway dimension, overtaking, USPs...
  - Running & evaluating tests of oracles
- 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

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 Mohring, 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