2^{nd} *Parameterized Algorithms & Computational Experiments* Challenge

Where it came from, how it went, who won, and what's next

September 6th, IPEC 2017, Vienna, Austria

NETWORKS

Program committee track A, treewidth

Holger Dell	Saarland University & Cluster of Excellence

Program committee track B, minimum fill-in

Christian Komusiewicz*	Friedrich-Schiller-University Jena
Nimrod Talmon	Weizmann Institute of Science
Mathias Weller	LIRMM Montpellier

Steering committee

Holger Dell	Saarland University & Cluster of Excellence
Bart M. P. Jansen	Eindhoven University of Technology
Thore Husfeldt	ITU Copenhagen and Lund University
Petteri Kaski	Aalto University
Christian Komusiewicz	Friedrich-Schiller-University Jena
Frances A. Rosamond*	University of Bergen

WHERE PACE CAME FROM

History of PACE

- PACE was conceived in fall 2015 when many FPT researchers gathered at the Simons institute
- Born from a feeling that parameterized algorithmics should have a greater impact on practice
- Partially inspired by the success of SAT-solving competitions in neighboring communities
- First iteration in 2015-2016
  - Track A: Treewidth (heuristically & exact)
  - Track B: Feedback Vertex Set

Goals

Investigate the applicability of algorithmic ideas from parameterized algorithmics

1. provide bridge between algorithm design&analysis theory and algorithm engineering practice
2. inspire new theoretical developments
3. investigate the competitiveness of analytical and design frameworks developed in the communities
4. produce universally accessible libraries of implementations and repositories of benchmark instances
5. encourage dissemination of the findings in scientific papers

Publications following the first PACE

htd – A Free, Open-Source Framework for (Customized) Tree Decompositions and Beyond

Michael Abseher(✉), Nysret Musliu, and Stefan Woltran

Institute of Information Systems, TU Wien,
184/2, Favoritenstraße 9–11, 1040 Vienna, Austria
{abseher,musliu,woltran}@dbai.tuwien.ac.at

Publications following the first PACE

htd – A Free, Open-Source Framework

itions

oltran

{abseher,musliu,woltran}@dbai.tuwien.ac.at

Answer Set Solving with Bounded Treewidth Revisited*

Johannes K. Fichte[†] Michael Morak, Markus Hecher and Stefan Woltran
TU Wien, Austria
lastname@dbai.tuwien.ac.at

Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, Markus Hecher, Michael Morak, Stefan Woltran
TU Wien, Vienna, Austria
lastname@dbai.tuwien.ac.at

Answer Set Solving with Bounded Treewidth Revisited*

Johannes K. Fichte†, Michael Morak, Markus Hecher and Stefan Woltran
TU Wien, Austria
lastname@dbai.tuwien.ac.at

Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, Markus Hecher, Michael Morak, Stefan Woltran
TU Wien, Vienna, Austria
lastname@dbai.tuwien.ac.at

mework
itions

oltran

Answer Set Solving with Bounded Treewidth Revisited*

Johannes K. Fichte[†] Michael Morak, Markus Hecher and Stefan Woltran
TU Wien, Aus
lastname@dbai.tuw

SAT-Encodings for Special Treewidth and Pathwidth

Neha Lodha, Sebastian Ordyniak(✉), and Stefan Szeider
Algorithms and Complexity Group, Tu Wien, Vienna, Austria
{neha,ordyniak,sz}@ac.tuwien.ac.at

Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, Markus Hecher, Michael Morak, Stefan Woltran
TU Wien, Vienna, Austria
lastname@dbai.tuwien.ac.at

Answer Set Solving with Bounded Treewidth Revisited*

Johannes K. Fichte†, Michael Morak, Markus Hecher and Stefan Woltran
TU Wien, Aus
lastname@dbai.tuw

mework
itions

oltran

SAT-Encodings for Special Treewidth and Pathwidth

Ordyniak(✉), and Stefan Szeider
ity Group, Tu Wien, Vienna, Austria
ak,sz}@ac.tuwien.ac.at

Linear-time Kernelization for Feedback Vertex Set

Yoichi Iwata
National Institute of Informatics
yiwata@nii.ac.jp

Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, Markus Hecher, Michael Morak, Stefan Woltran
TU Wien, Vienna, Austria
lastname@dbai.tuwien.ac.at

mework
itions

oltran

Answer Set Solving with Bounded Treewidth Revisited*

Johannes K. Fichte[†] Michael Morak, Markus Hecher and Stefan Woltran
TU Wien, Aus
lastname@dbai.tuw

SAT-Encodings for Special Treewidth and Pathwidth

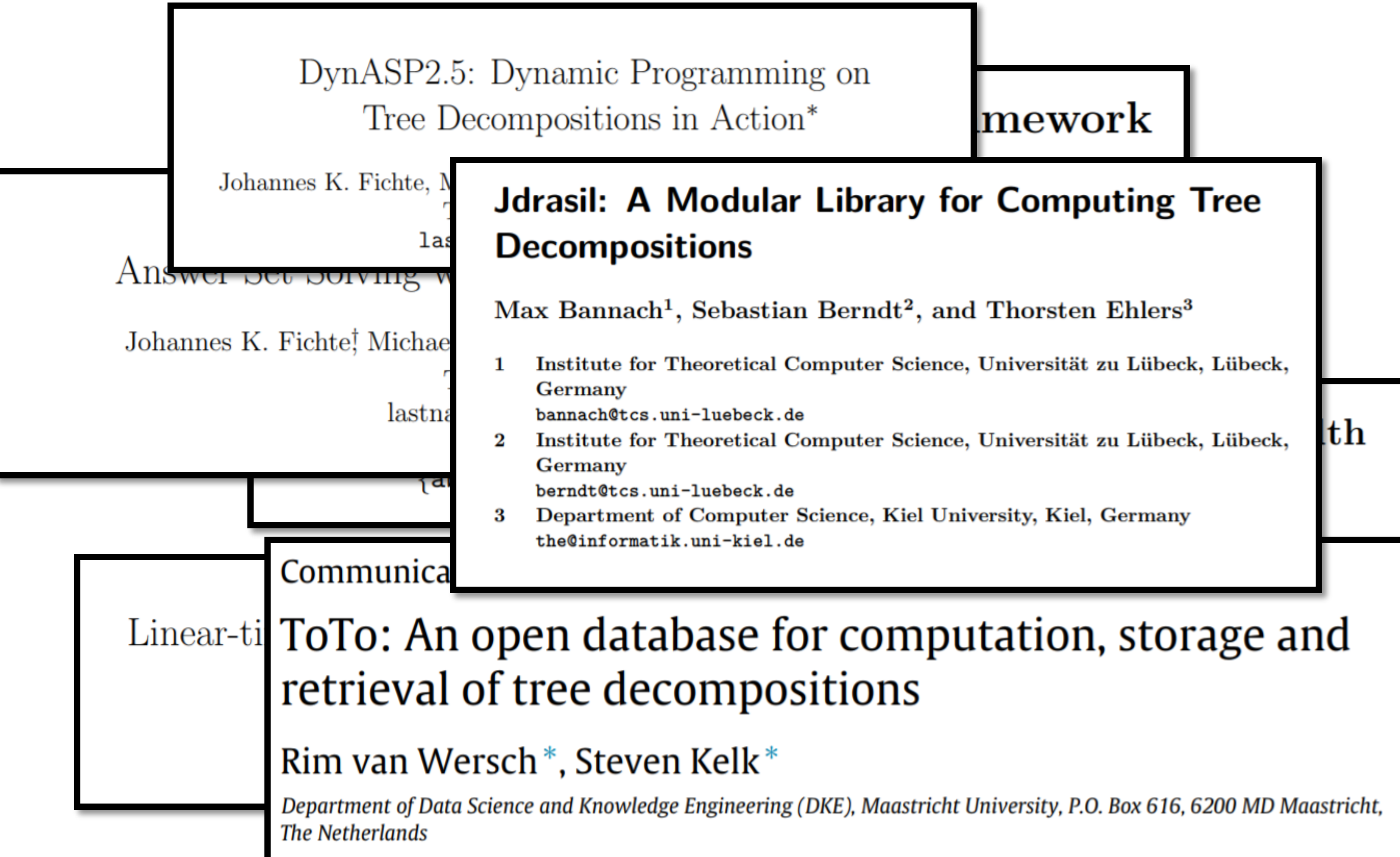
Linear-ti

Communication

ToTo: An open database for computation, storage and retrieval of tree decompositions

Rim van Wersch*, Steven Kelk*
Department of Data Science and Knowledge Engineering (DKE), Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Publications following the first PACE



DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, M

las

…mework

Ans… Set Solving w…

Johannes K. Fichte† Michae

lastna

…th

{a

Jdrasil: A Modular Library for Computing Tree Decompositions

Max Bannach[1], Sebastian Berndt[2], and Thorsten Ehlers[3]

1 Institute for Theoretical Computer Science, Universität zu Lübeck, Lübeck, Germany
bannach@tcs.uni-luebeck.de

2 Institute for Theoretical Computer Science, Universität zu Lübeck, Lübeck, Germany
berndt@tcs.uni-luebeck.de

3 Department of Computer Science, Kiel University, Kiel, Germany
the@informatik.uni-kiel.de

Communica

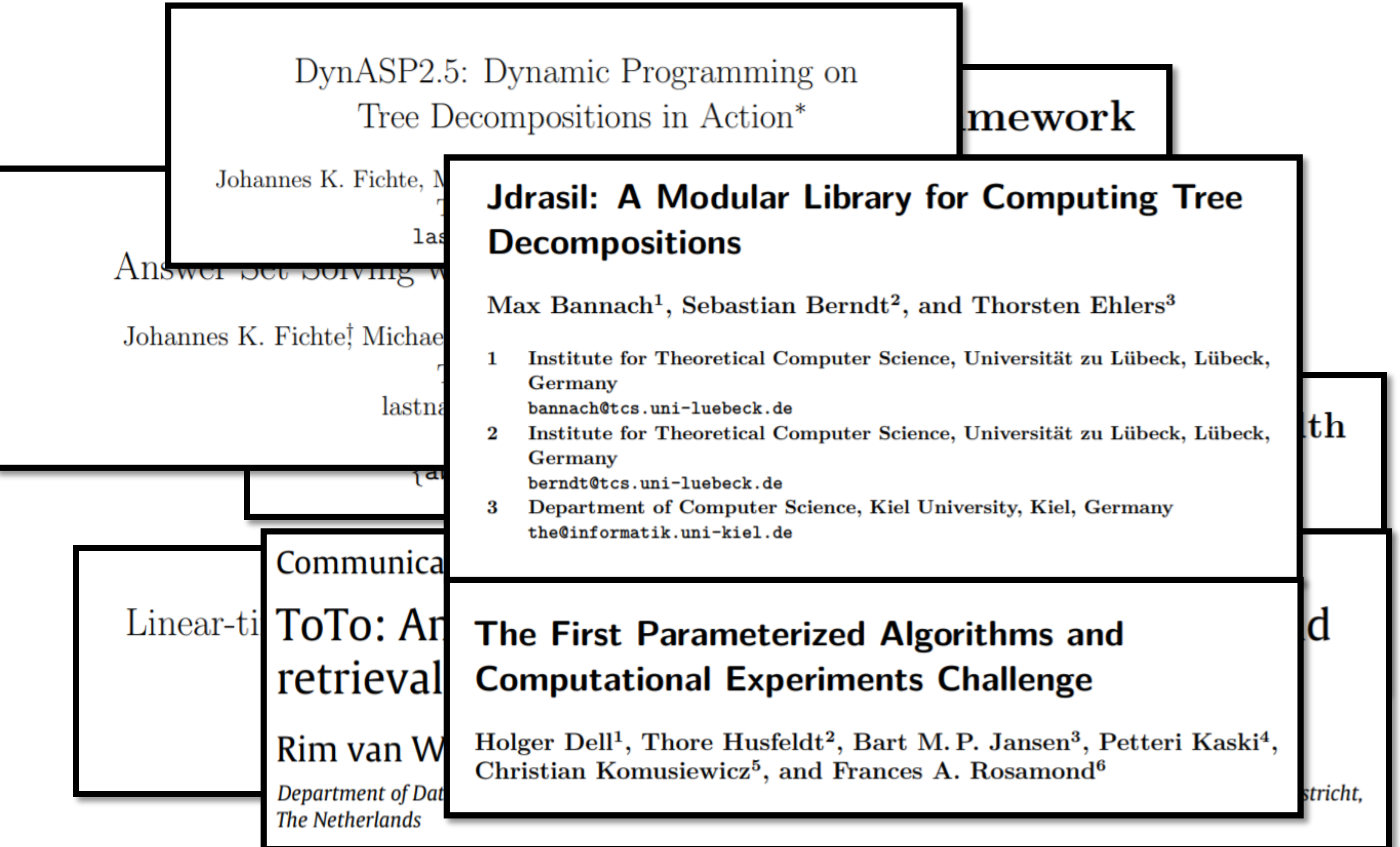
Linear-ti

ToTo: An open database for computation, storage and retrieval of tree decompositions

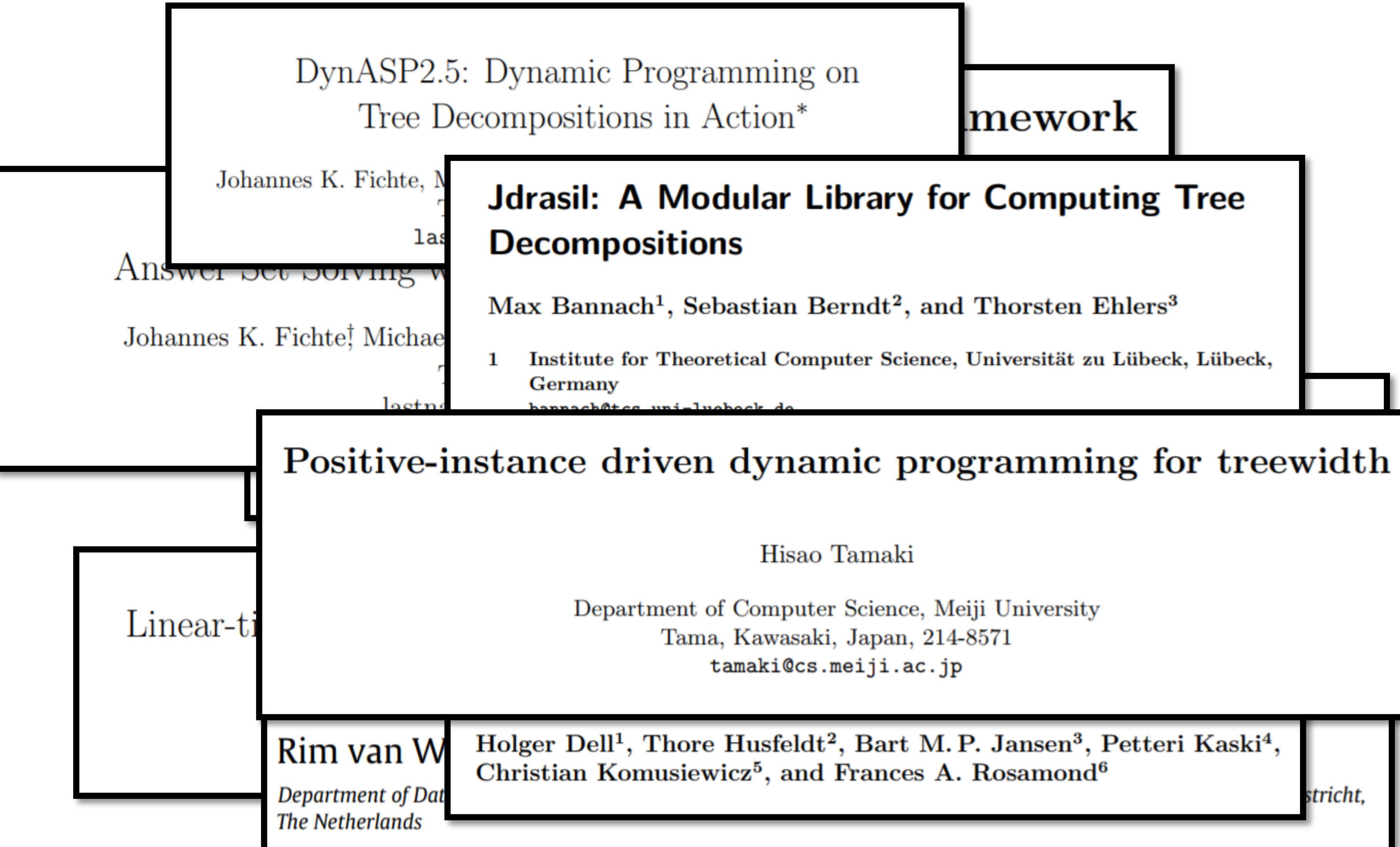
Rim van Wersch*, Steven Kelk*

Department of Data Science and Knowledge Engineering (DKE), Maastricht University, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Publications following the first PACE



Publications following the first PACE



DynASP2.5: Dynamic Programming on Tree Decompositions in Action*

Johannes K. Fichte, M

las

...mework

Answer Set Solving w

Johannes K. Fichte† Michae

lastn

Jdrasil: A Modular Library for Computing Tree Decompositions

Max Bannach[1], Sebastian Berndt[2], and Thorsten Ehlers[3]

1 Institute for Theoretical Computer Science, Universität zu Lübeck, Lübeck, Germany

Positive-instance driven dynamic programming for treewidth

Hisao Tamaki

Department of Computer Science, Meiji University
Tama, Kawasaki, Japan, 214-8571
tamaki@cs.meiji.ac.jp

Linear-ti

Rim van W

Department of Dat
The Netherlands

Holger Dell[1], Thore Husfeldt[2], Bart M. P. Jansen[3], Petteri Kaski[4], Christian Komusiewicz[5], and Frances A. Rosamond[6]

stricht,

Publications following the first PACE

DynASP2.5: Dynamic Programming on
Tree Decompositions in Action*

Johannes K. Fichte, M

mework

Jdrasil: A Modular Library for Computing Tree

Acknowledgment

The author thanks Hiromu Ohtsuka for his help in implementing the block sieve data structure. He also thanks Yasuaki Kobayashi for helpful discussions and especially for drawing the author's attention to the notion of safe separators. This work would have been non-existent if not motivated by the timely challenges of PACE 2016 and 2017. The author is deeply indebted to their organizers, especially Holger Dell, for their dedication and excellent work.

Department of Computer Science, Meiji University
Tama, Kawasaki, Japan, 214-8571
tamaki@cs.meiji.ac.jp

Linear-ti

Rim van W
Department of Dat
The Netherlands

Holger Dell[1], Thore Husfeldt[2], Bart M. P. Jansen[3], Petteri Kaski[4], Christian Komusiewicz[5], and Frances A. Rosamond[6]

stricht,

PACE timeline in 2016-2017

1. Treewidth track
2. Track for computing minimum fill-in (chordal completion)

Time schedule

- November 1st 2016: Announcement of problems and inputs
- March 1st 2017: Submission of prototype program
- May 1st 2017: Submission of final program
- June 1st 2017: Result are communicated to participants
- September 6th 2017: Award ceremony at IPEC

Sponsor for prizes & travel

NETWORKS is a project of
University of Amsterdam
Eindhoven University of Technology
Leiden University
Center for Mathematics and
Computer Science (CWI)

NETWORKS

The NETWORKS project generously sponsors PACE with € 4000
1^{st} prize (€ 500), 2^{nd} prize (€ 300) and 3^{rd} prize (€ 200)

Three subcategories in the competition, with €1000 travel award

`thenetworkcenter.nl`

PACE timeline in 2017-2018

- PACE will focus on a single challenge problem next year

Time schedule

- Today: Announcement of the problem
- November 1st 2017: Detailed problem setting and inputs
- March 1st 2018: Submission of prototype program
- May 1st 2018: Submission of final program
- June 1st 2018: Result are communicated to participants
- August 20-24 2018: Award ceremony at IPEC

The third iteration of PACE

PACE 2017-2018 program committee

Édouard Bonnet — Middlesex University, London

Florian Sikora — University Paris Dauphine

Steiner Tree

How it went and who won

TRACK A: TREEWIDTH

PACE 2017
Track A: Treewidth

Holger Dell

Treewidth Applications (outside of FPT)

- Register allocation in compilers
  (e.g., Thorup 1998)

- Preprocessing for shortest path
  (e.g., Chatterjee Ibsen-Jensen Pavlogiannis 2016)

- Treewidth of specific graph families
  (e.g., Kiyomia Okamotob Otachic 2015)

- Preprocessing for probabilistic inference
  (e.g., Otten Ihler Kask Dechter 2011)

PACE: submission requirements

- repository on github.com
- “edge list” input format
- Output: tree decomposition

Heuristic treewidth competition

Benchmark instances

100 public + 100 secret instances:

35%	graphs from the **UAI 2014 competition** (probabilistic inference)
35%	incidence graphs of **SAT competition** instances
16%	graphs from **treedecomposition.com**
7%	**road graphs**
7%	**transit networks**

	number of edges	treewidth (upper bound)
median	**14k**	**93**
mean	**991k**	**13k**

Ranking by Preferential Voting

Instances=Voters

"Ballot" for instance he166.gr:

submission	width after 30 minutes
B	672
E	957
A	994
C	33279

→ Use Schulze method to combine votes

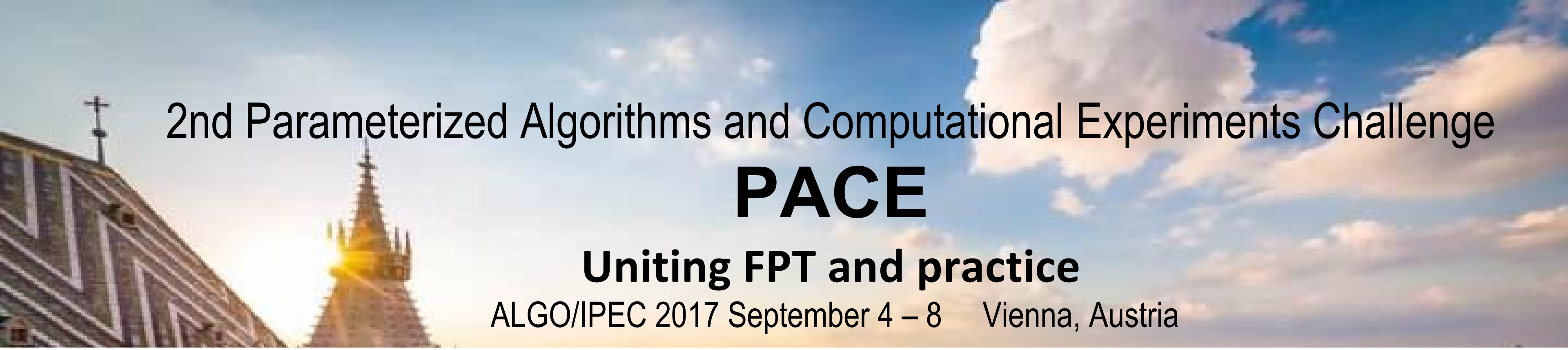
Participants

6 submissions:

- 3 new teams
- 3 teams from last year

Honorable mentions

Rank 4	Max Bannach Sebastian Berndt Thorsten Ehlers	(University of Lübeck), (University of Lübeck), (University of Kiel)
Rank 5	Philippe Jégou Hanan Kanso Cyril Terrioux	(Aix-Marseille Université, LSIS)
Rank 6	Lukas Larisch Felix Salfelder	(King-Abdullah University of Science and Engineering) (University of Leeds)



2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8   Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Michael Abseher, Nysret Musliu, Stefan Woltran

TU Wien, Institute of Information Systems

as the

Third Place Winners in Heuristic Treewidth Decomposition

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8   Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Ben Strasser

Karlsruhe Institute of Technology

as the

Second Place Winner in the Heuristic Treewidth Decomposition Challenge

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8   Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Keitaro Makii, Hiromu Ohtsuka, Takuto Sato, Hisao Tamaki

Meiji University

as the

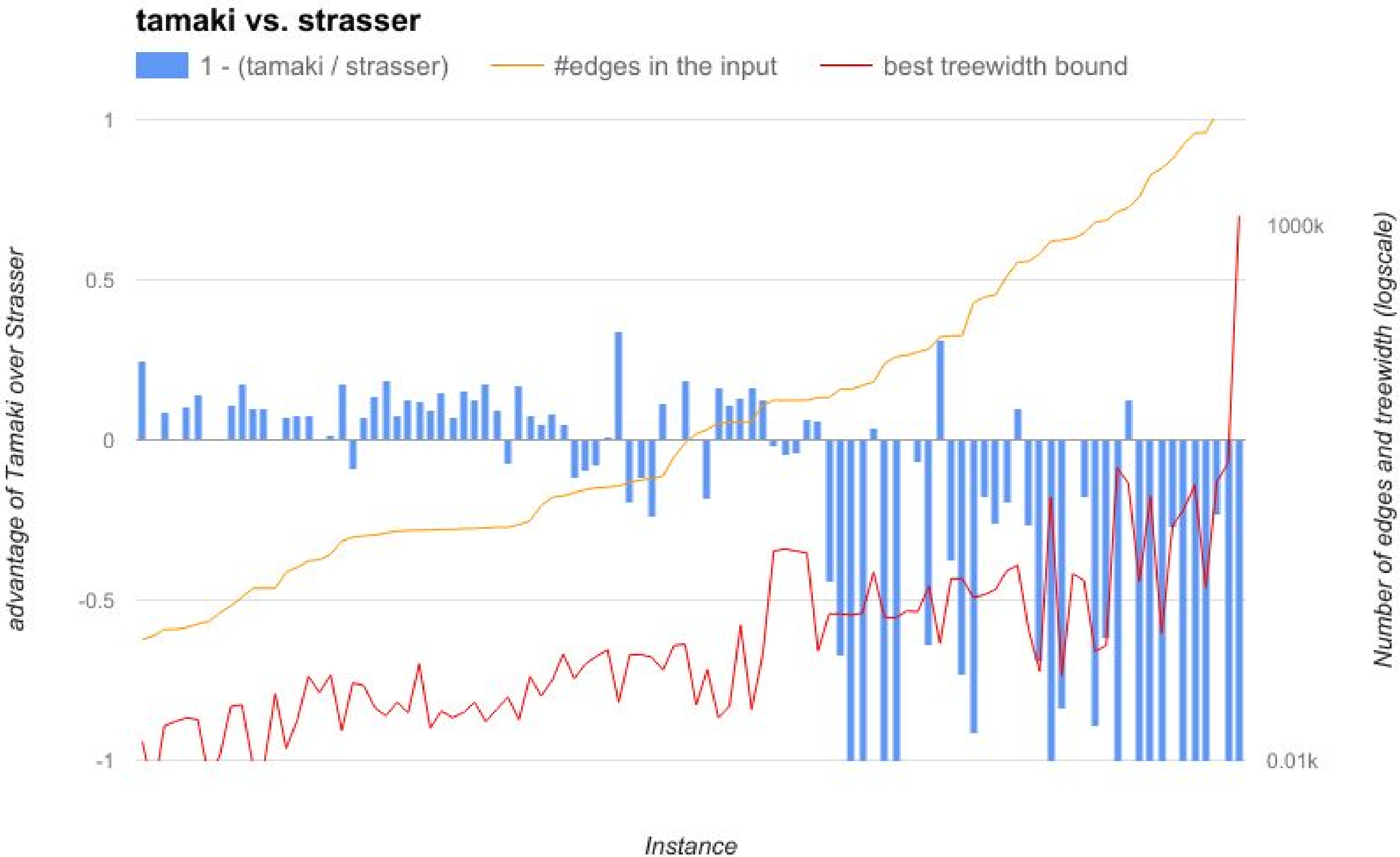
First Place Winners in Heuristic Treewidth Decomposition

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

Plot: Winner vs. Second



Exact treewidth competition

Benchmark instances

100 public + 100 secret instances

Grow balls in graphs from heuristic challenge

Use CPU months to test “instance difficulty” by running last year’s winning solver

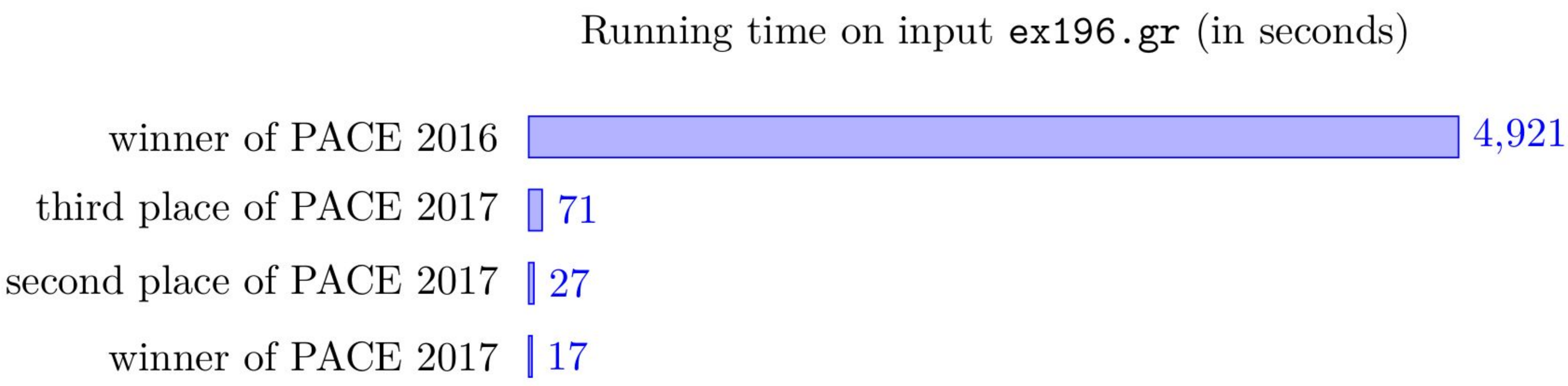
	number of edges	treewidth
median	**730**	**11**
mean	**7300**	**31**

Outcome

3 submissions:

1 new team

2 teams from last year



Everyone was 100x faster than last year!

2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8 Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Max Bannach, Sebastian Berndt, Thorsten Ehlers

University of Lübeck, University of Lübeck, University of Kiel

as the

Third Place Winners in the Optimal Treewidth Decomposition Competition

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8 Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Hiromu Ohtsuka and **Hisao Tamaki**

Meiji University

as the

Second Place Winners in the Optimal Treewidth Decomposition Competition

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8   Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Lukas Larisch and **Felix Salfelder**

King-Abdullah University of Science and Engineering   University of Leeds

as the

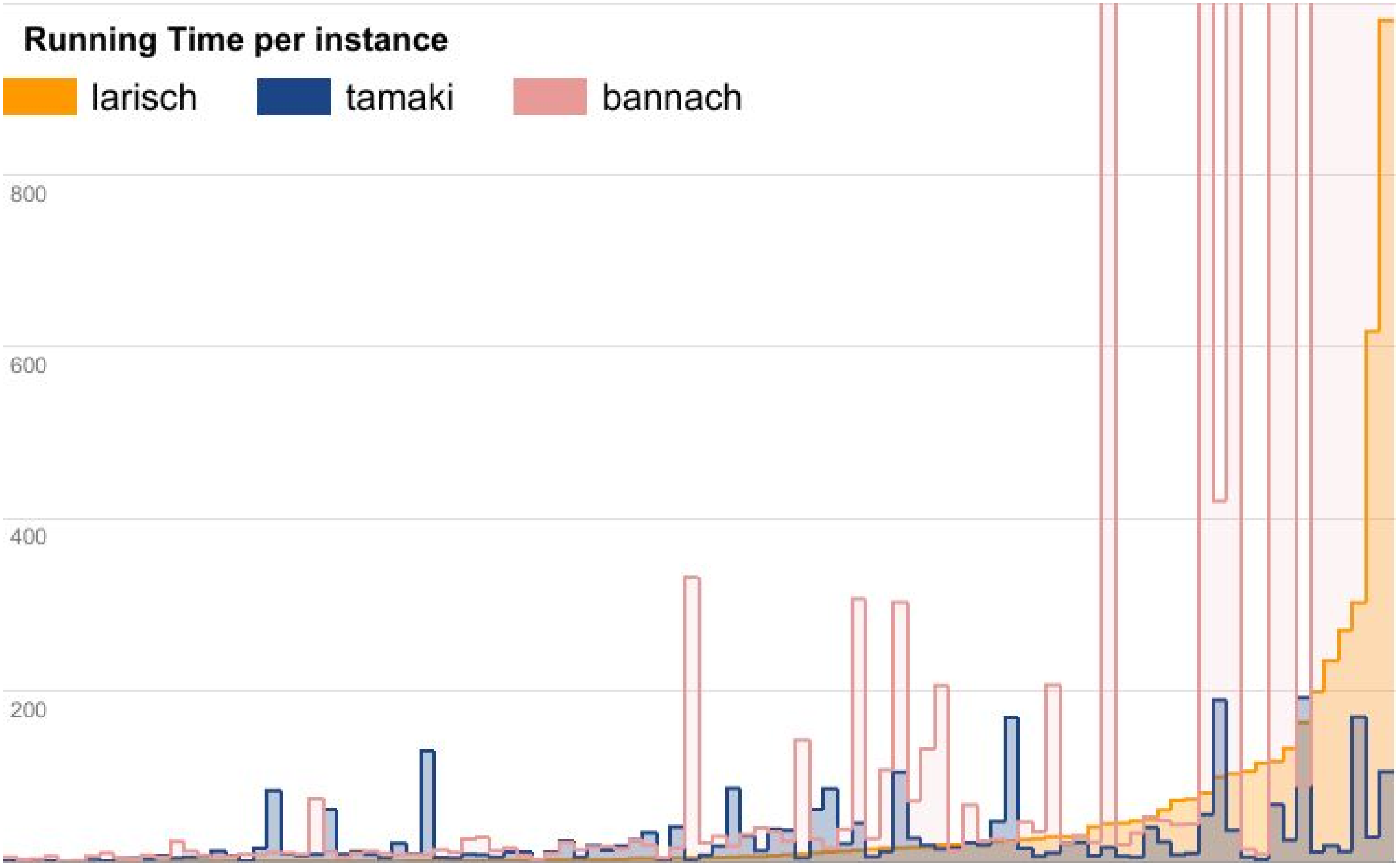
First Place Winners in the Optimal Treewidth Decomposition Competition

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

Exact treewidth: Plot



Treewidth competition future

New instance set for exact treewidth:

- Supports 1000x speed improvements over PACE 2017
- Persistent competition on optil.io

tdlib – PACE 2017

Lukas Larisch, Felix Salfelder

IPEC 2017

About tdlib, goals

- Tree decomposition (and related) algorithms
  - Free (libre) heuristic/exact implementations
  - Pre/post processing
  - As C++ library

About tdlib, goals

- Tree decomposition (and related) algorithms
  - Free (libre) heuristic/exact implementations
  - Pre/post processing
  - As C++ library
- Explore theoretic results in practice
  - Register allocation (sdcc)
    P. K. Krause, L. Larisch: The Treewidth of C, (SCOPES'15)

About tdlib, goals

- Tree decomposition (and related) algorithms
  - Free (libre) heuristic/exact implementations
  - Pre/post processing
  - As C++ library
- Explore theoretic results in practice
  - Register allocation (sdcc)
    P. K. Krause, L. Larisch: The Treewidth of C, (SCOPES'15)
  - Treewidth bounds in large instances
    (e.g. `maxsat`, up to 1.e7/4.e11 vertices/edges, 7% rel. err)

About tdlib, goals

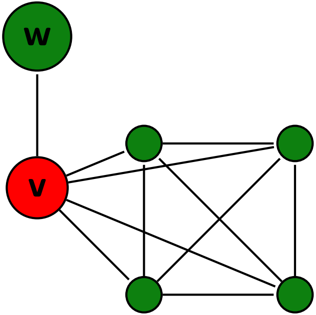
- Tree decomposition (and related) algorithms
  - Free (libre) heuristic/exact implementations
  - Pre/post processing
  - As C++ library
- Explore theoretic results in practice
  - Register allocation (sdcc)
    P. K. Krause, L. Larisch: The Treewidth of C, (SCOPES'15)
  - Treewidth bounds in large instances
    (e.g. `maxsat`, up to 1.e7/4.e11 vertices/edges, 7% rel. err)
  - Python bindings
  - A Sagemath package

Preprocessing

- Rule based complete reduction for treewidth 4
  islet, twig, buddy, series, cube. c.f. tdlib documentation

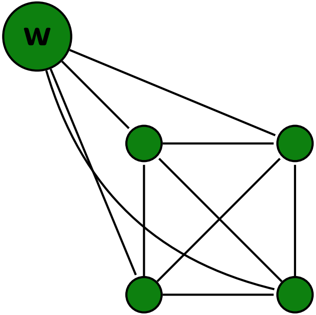
Preprocessing

- Rule based complete reduction for treewidth 4
- (Almost) simplicial vertex elimination rules



Preprocessing

- Rule based complete reduction for treewidth 4
- (Almost) simplicial vertex elimination rules



tdlib and PACE'16

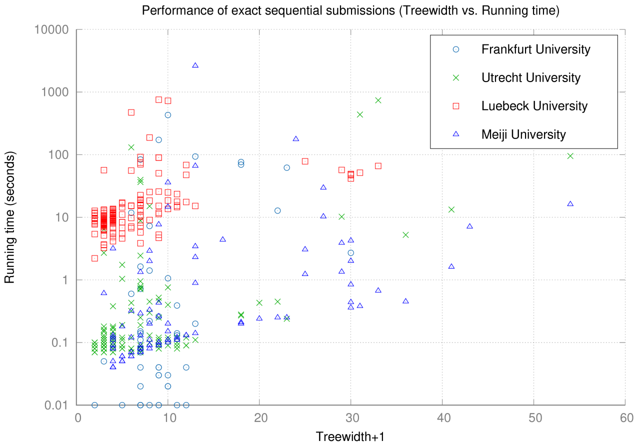
- refactoring: C++11, generic programming

tdlib and PACE'16

- refactoring: C++11, generic programming
- structural/algorithmic improvements

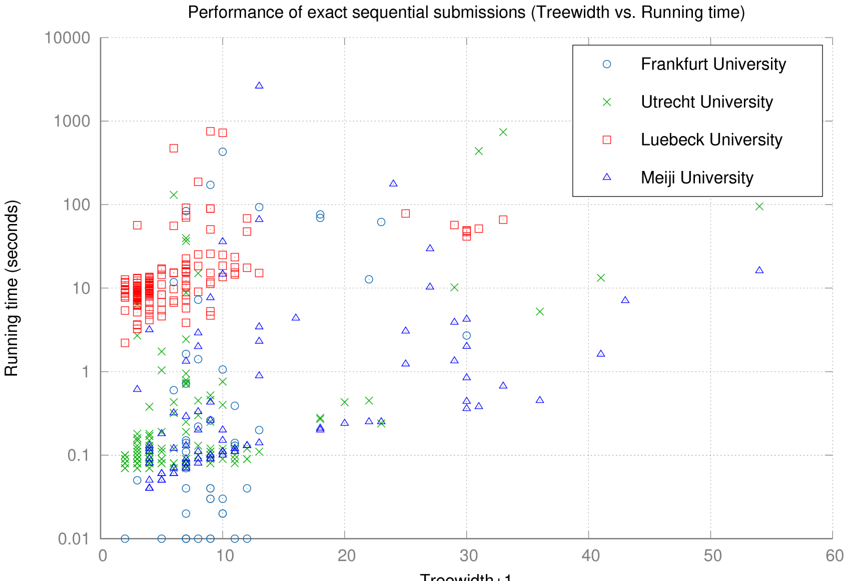
tdlib and PACE'16

- refactoring: C++11, generic programming
- structural/algorithmic improvements
- reference implementations, exact & heuristic



tdlib and PACE'16

- reference implementations, exact & heuristic



PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

Exact algorithm, recycling

- Rule based preprocessor

PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

Exact algorithm, recycling

- Rule based preprocessor
- Exact kernel inspired by PACE'16 (Tamaki)
  - .. implementing Arnborg, Corneil, Proskurowski + more ideas.

PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

Exact algorithm, recycling

- Rule based preprocessor
- Exact kernel inspired by PACE'16 (Tamaki)
  - .. implementing Arnborg, Corneil, Proskurowski + more ideas.
  - Restructured, object oriented

PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

Exact algorithm, recycling

- Rule based preprocessor
- Exact kernel inspired by PACE'16 (Tamaki)
  - .. implementing Arnborg, Corneil, Proskurowski + more ideas.
  - Restructured, object oriented
  - Ported to tdlib/gala

PACE'17 submission

Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- $\rightsquigarrow$ interruptible exact algorithm
- Postprocessing

Exact algorithm, recycling

- Rule based preprocessor
- Exact kernel inspired by PACE'16 (Tamaki)
  - .. implementing Arnborg, Corneil, Proskurowski + more ideas.
  - Restructured, object oriented
  - Ported to tdlib/gala
- Optimised for speed
- $\rightsquigarrow$ pretty fast on small instances

Thank You.

How it went and who won

TRACK B: MINIMUM FILL-IN

The 2nd Parameterized Algorithms and Computational Experiments Challenge: Track B Minimum Fill-In

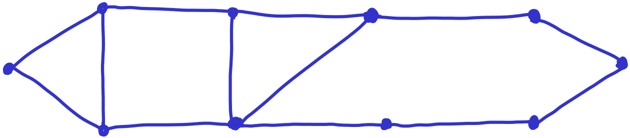
Christian Komusiewicz
Friedrich-Schiller-Universität Jena

Nimrod Talmon
Weizmann Institute of Science

Mathias Weller
LIRMM, Université de Montpellier II

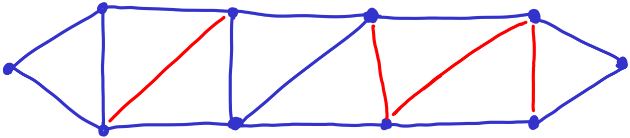
Challenge Problem

Minimum Fill-In
Input: An undirected graph $G = (V, E)$.
Task: Find a minimum-size edge set F such that $(V, E \cup F)$ is chordal.



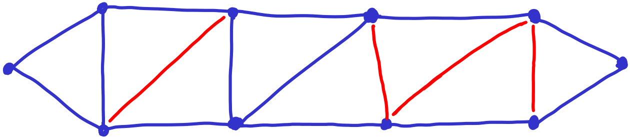
Challenge Problem

Minimum Fill-In
Input: An undirected graph $G = (V, E)$.
Task: Find a minimum-size edge set F such that $(V, E \cup F)$ is chordal.



Challenge Problem

Minimum Fill-In
Input: An undirected graph $G = (V, E)$.
Task: Find a minimum-size edge set F such that $(V, E \cup F)$ is chordal.



Minimum Fill-In is

- fixed-parameter tractable e.g. parameterized by solution size $|F|$,
- admits subexponential-time algorithms

Challenge Setup

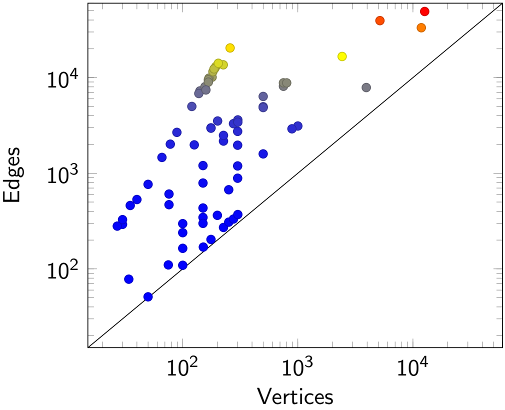
Benchmark Instances: 100 public + 100 hidden instances

Instance origin: Systems of linear equations, phylogenetic networks, social networks, molecular interaction networks

Challenge Setup

Benchmark Instances: 100 public + 100 hidden instances

Instance origin: Systems of linear equations, phylogenetic networks, social networks, molecular interaction networks



Ranking: # solved hidden instances within 30 minutes (each)

CKOMUS SIGN OUT

OPTIL.io

ABOUT CONTESTS PROBLEMS STANDING AUTHOR

MINIMUM FILL IN: PACE 2017 B

By Christian Komusiewicz[1]

DESCRIPTION RUNS MY RUNS STANDING PRIVATE STANDING SUBMIT DISCUSS

TLE = Time Limit Exceeded, **WA** = Wrong Answer, **RTE** = Runtime Error, **MLE** = Memory Limit Exceeded, **OLE** = Output Limit Exceeded, **PLE** = Processes Limit Exceeded, more help...

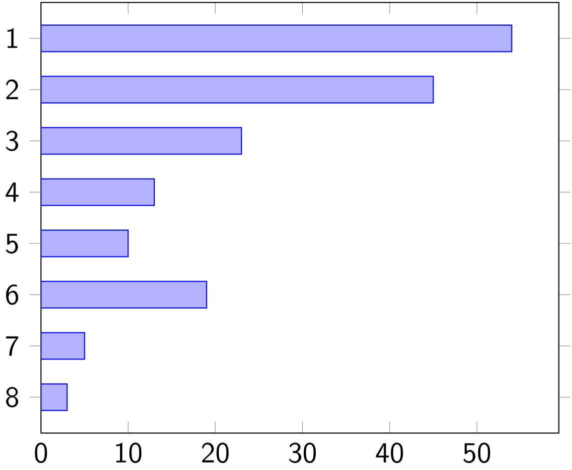
First 1 Last

20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
3.00	WA	527.00	1,017.00	1,652.00	79.00	3.00	16.00	578.00	1,754.00	63.00	320.00	WA	134.00	144.00	WA	439.00	426.00	712.00	WA	101.00
3.00	TLE	521.00	965.00	1,408.00	78.00	3.00	15.00	559.00	1,570.00	63.00	307.00	TLE	122.00	144.00	2,278.00	423.00	387.00	654.00	TLE	100.00
3.00	MLE	521.00	965.00	1,408.00	78.00	3.00	15.00	559.00	1,570.00	63.00	307.00	TLE	122.00	144.00	2,278.00	423.00	387.00	TLE	TLE	100.00
3.00	TLE	TLE	965.00	TLE	TLE	3.00	15.00	TLE	TLE	63.00	TLE	MLE	MLE	144.00	MLE	TLE	TLE	TLE	MLE	MLE
3.00	TLE	TLE	TLE	TLE	TLE	3.00	15.00	TLE	TLE	63.00	TLE	TLE	TLE	144.00	TLE	TLE	TLE	TLE	TLE	TLE
3.00	TLE	TLE	TLE	TLE	78.00	3.00	15.00	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	100.00
3.00	TLE	TLE	TLE	TLE	78.00	3.00	15.00	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	100.00
3.00	TLE	TLE	MLE	TLE	TLE	3.00	15.00	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE
3.00	RTE	TLE	TLE	TLE	WA	WA	WA	TLE	TLE	TLE	TLE	TLE	WA	WA	TLE	TLE	TLE	TLE	TLE	WA
WA	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE	TLE
RTE	TLE	MLE	MLE	TLE	TLE	RTE	RTE	MLE	TLE	TLE	MLE	MLE	TLE	TLE	RTE	MLE	MLE	MLE	MLE	TLE

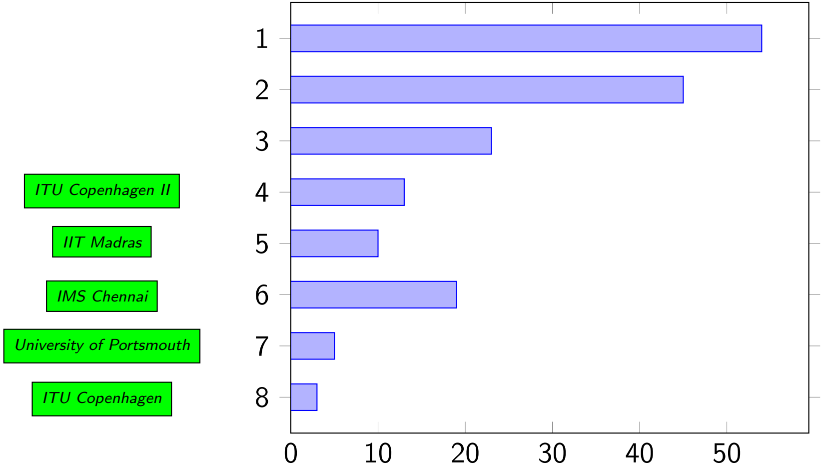
The National Centre for Research and Development

The project is co-financed by The National Centre for Research and Development within the LIDER programme.

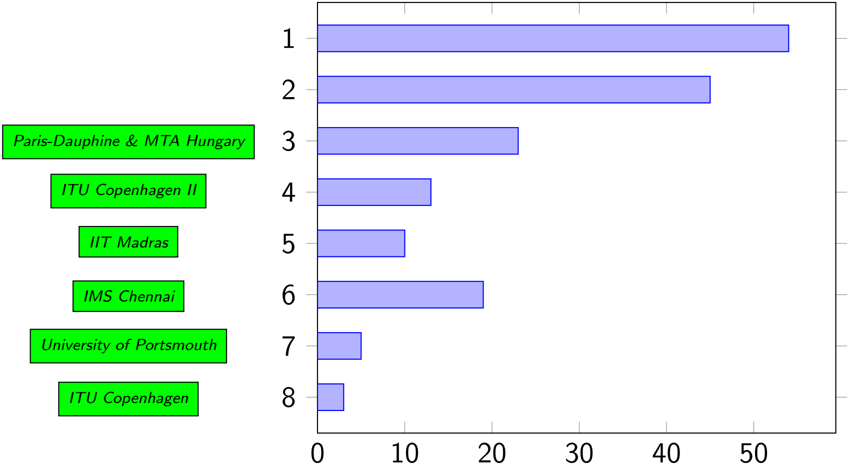
Results



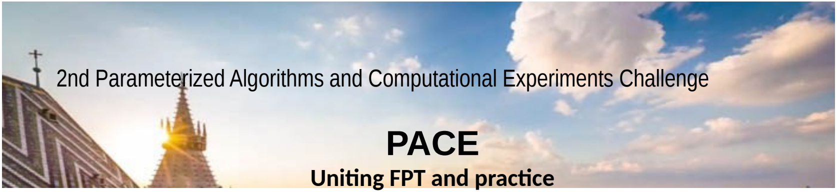
Results


ITU Copenhagen II
IIT Madras
IMS Chennai
University of Portsmouth
ITU Copenhagen
1
2
3
4
5
6
7
8
0
10
20
30
40
50

Results



Paris-Dauphine & MTA Hungary
ITU Copenhagen II
IIT Madras
IMS Chennai
University of Portsmouth
ITU Copenhagen
1
2
3
4
5
6
7
8
0
10
20
30
40
50



This is to certify that the 2017 PACE Program Committee has selected

Édouard Bonnet, R.B. Sandeep, Florian Sikora

University Paris-Dauphine Hungarian Academy of Sciences University Paris-Dauphine

as the

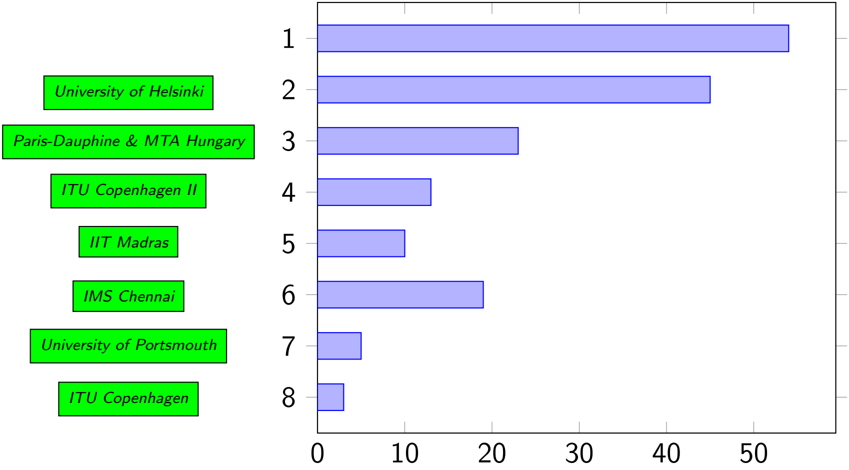
Third Place Winners in the Minimum Fill-In Challenge

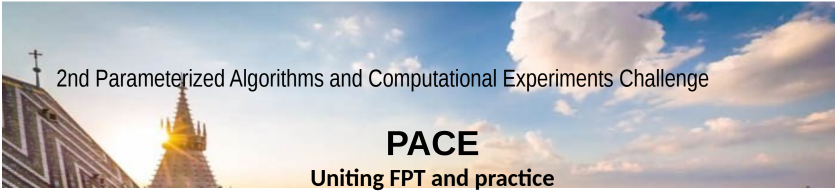
Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

Results


University of Helsinki
Paris-Dauphine & MTA Hungary
ITU Copenhagen II
IIT Madras
IMS Chennai
University of Portsmouth
ITU Copenhagen
1
2
3
4
5
6
7
8
0
10
20
30
40
50



2nd Parameterized Algorithms and Computational Experiments Challenge

PACE

Uniting FPT and practice

ALGO/IPEC 2017 September 4 – 8 Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Jeremias Berg, Matti Järvisalo, Tuukka Korhonen

University of Helsinki

as the

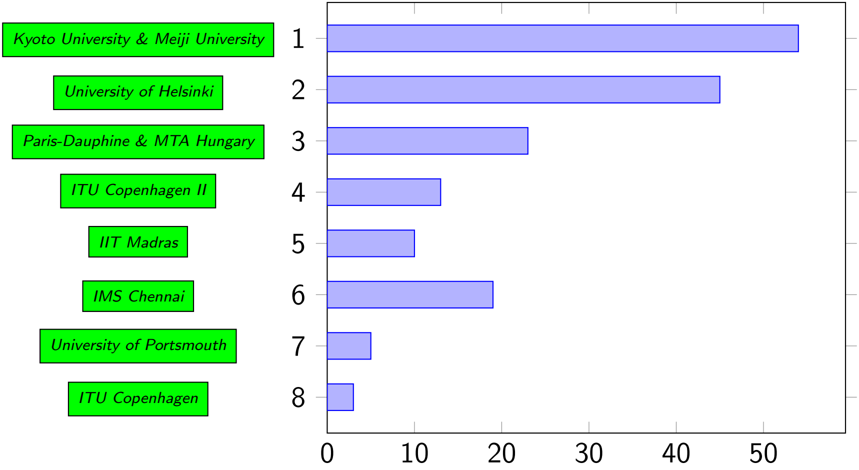
Second Place Winners in the Minimum Fill-In Challenge

Holger Dell, Saarland University. Track A Chair

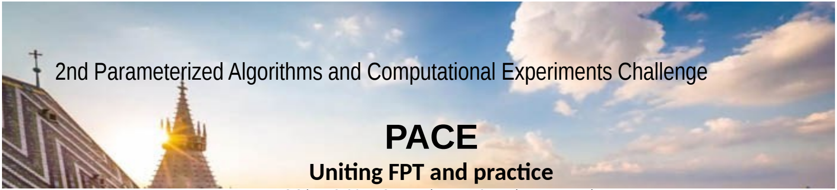
Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

Results



Kyoto University & Meiji University
University of Helsinki
Paris-Dauphine & MTA Hungary
ITU Copenhagen II
IIT Madras
IMS Chennai
University of Portsmouth
ITU Copenhagen
1
2
3
4
5
6
7
8
0
10
20
30
40
50



ALGO/IPEC 2017 September 4 – 8 Vienna, Austria

This is to certify that the 2017 PACE Program Committee has selected

Yasuaki Kobayashi, Hisao Tamaki

Kyoto University Meiji University

as the

First Place Winners in the Minimum Fill-In Challenge

Holger Dell, Saarland University. Track A Chair

Christian Komusiewicz, Friedrich-Schiller-University Jena. Track B Chair

2017 PACE Programme Committee Co-chairs

About our submission (Track B)

<u>Yasuaki Kobayashi</u> Hisao Tamaki

Minimum Fill-In Problem

Given: undirected graph $G = (V, E)$

Task: find a smallest F such that $G' = (V, E \cup F)$ is chordal

Techniques

- A sufficient condition for edges that can be safely added.
- A modified version of “Positive-instance driven dynamic programming for treewidth”.

Edges that can be safely added

Lemma [Bodlaender et al. 2011]:

Let S be a minimal separator of G such that $S \subseteq N(v)$ for some $v \in V$. Suppose $|\mathrm{miss}(S)| = 1$, where $\mathrm{miss}(S)$ is the set of missing edges in $G[S]$. Then, there is an optimal solution that contains $\mathrm{miss}(S)$.

- If G has a minimal separator S that satisfies the above condition, we can decompose G by using S.
- We can generalize this lemma for minimal separators that have more than one missing edges (with some additional conditions).

Positive-Instance Driven DP

- The treewidth and minimum fill-in problem can be solved by DP algorithms based on minimal separators and potential maximal cliques [Bouchitté & Todinca 2011].

- Tamaki developed a positive-instance driven DP for treewidth [Tamaki 2017].

  - applicable to the min fill-in problem with some non-trivial modifications.

Thank you!

https://github.com/TCS-Meiji/PACE2017-TrackB/

CHANGING ROLES


GOODBYE, FRAN!
PACE

NETWORKS is a project of
University of Amsterdam
Eindhoven University of Technology
Leiden University
Center for Mathematics and
Computer Science (CWI)

NETWORKS

OPTIL.io

https://pacechallenge.wordpress.com