2nd Parameterized Algorithms & Computational Experiments Challenge

Where it came from, how it went, who won, and what’s next
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
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 for (Constraint) Tree-Decompositions

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
{absener,musliu,woltran}@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

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

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, Austria
lastname@dbai.tuwien.ac.at

**SAT-Encodings for Special Treewidth and Pathwidth***

Johannes Ordyniak, and Stefan Szeider
Computer Science Institute Group, TU Wien, Vienna, Austria
sak, 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

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

SAT-Encodings for Special Treewidth and Pathwidth

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, Michael Fink

Answer Set Solving with DynASP2.5

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

Jdrasil: A Modular Library for Computing Tree Decompositions

Max Bannach¹, Sebastian Berndt², and Thorsten Ehlers³

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
Publications following the first PACE
Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action*
Johannes K. Fichte, Michael Franz, and Stefan Szeider
Answer Set Solving with DynASP 2.5

Jdrasil: A Modular Library for Computing Tree Decompositions
Max Bannach¹, Sebastian Berndt², and Thorsten Ehlers³
¹ Institute for Theoretical Computer Science, Universität zu Lübeck, Lübeck, Germany
² iwr, University of Heidelberg, Germany
³ Algorithms and Complexity Institute, Vienna University of Technology, Austria

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

Rim van Wijk¹, Holger Dell², Thore Husfeldt², Bart M. P. Jansen³, Petteri Kaski⁴, Christian Komusiewicz⁵, and Frances A. Rosamond⁶
¹ Department of Data Science, University of Twente, The Netherlands
Publications following the first PACE

DynASP2.5: Dynamic Programming on Tree Decompositions in Action

Johannes K. Fichte, Marko Lacki

Jdrasil: A Modular Library for Computing Tree Framework

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.

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

Linear-time FPT Algorithms for Computing Tree Decomposition Width

Hans L. Bodlaender, Hans Bodlaender, Willem F. L. de Roever
Department of Computer Science, University of Groningen
The Netherlands

Rim van Wijngaarden
Department of Computing Science, University of Tartu
Tartu, Estonia

Holger Dell1, Thore Husfeldt2, Bart M. P. Jansen3, Petteri Kaski4, Christian Komusiewicz5, and Frances A. Rosamond6

1 Saarland University, Saarbrücken, Germany
2 Stockholm University, Stockholm, Sweden
3 University of Twente, Enschede, The Netherlands
4 Aalto University, Espoo, Finland
5 Université Paris 13, Sorbonne Paris Cité
6 University of South Australia, Adelaide, Australia
PACE timeline in 2016-2017

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

Time schedule

– November 1\textsuperscript{st} 2016: Announcement of problems and inputs
– March 1\textsuperscript{st} 2017: Submission of prototype program
– May 1\textsuperscript{st} 2017: Submission of final program
– June 1\textsuperscript{st} 2017: Result are communicated to participants
– September 6\textsuperscript{th} 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)

The NETWORKS project generously sponsors PACE with € 4000
1\textsuperscript{st} prize (€ 500), 2\textsuperscript{nd} prize (€ 300) and 3\textsuperscript{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 1\(^{st}\) 2017: Detailed problem setting and inputs
– March 1\(^{st}\) 2018: Submission of prototype program
– May 1\(^{st}\) 2018: Submission of final program
– June 1\(^{st}\) 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
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 Okamoto Otachi 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](http://treedecomposition.com)
- 7% **road graphs**
- 7% **transit networks**

<table>
<thead>
<tr>
<th></th>
<th>number of edges</th>
<th>treewidth (upper bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td>median</td>
<td>14k</td>
<td>93</td>
</tr>
<tr>
<td>mean</td>
<td>991k</td>
<td>13k</td>
</tr>
</tbody>
</table>
Ranking by Preferential Voting

Instances=Voters

“Ballot” for instance he166.gr:

<table>
<thead>
<tr>
<th>submission</th>
<th>width after 30 minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>672</td>
</tr>
<tr>
<td>E</td>
<td>957</td>
</tr>
<tr>
<td>A</td>
<td>994</td>
</tr>
<tr>
<td>C</td>
<td>33279</td>
</tr>
</tbody>
</table>

→ Use Schulze method to combine votes
Participants

6 submissions:

3 new teams

3 teams from last year
Honorable mentions

Rank 4
Max Bannach (University of Lübeck),
Sebastian Berndt (University of Lübeck),
Thorsten Ehlers (University of Kiel)

Rank 5
Philippe Jégou
Hanan Kanso (Aix-Marseille Université, LSIS)
Cyril Terrioux

Rank 6
Lukas Larisch (King-Abdullah University of Science and Engineering)
Felix Salfelder (University of Leeds)
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
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
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

tamaki vs. strasser

1 - (tamaki / strasser)  #edges in the input  best treewidth bound

advantage of Tamaki over Strasser

Number of edges and treewidth (logscale)

Instance
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

<table>
<thead>
<tr>
<th></th>
<th>number of edges</th>
<th>treewidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>median</td>
<td>730</td>
<td>11</td>
</tr>
<tr>
<td>mean</td>
<td>7300</td>
<td>31</td>
</tr>
</tbody>
</table>
Outcome

3 submissions:

1 new team

2 teams from last year

Running time on input ex196.gr (in seconds)

winner of PACE 2016  4,921
third place of PACE 2017  71
second place of PACE 2017  27
winner of PACE 2017  17

Everyone was 100x faster than last year!
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
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**
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

Running Time per instance
- larisch
- tamaki
- bannach

Y-axis: Running Time per instance
X-axis: Instances
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 $1.0 \times 10^7 / 4.0 \times 10^{11}$ 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

![Performance of exact sequential submissions (Treewidth vs. Running time)](image)
Heuristic "anytime" algorithm

- Guided elimination order brute forcing
- \(\rightsquigarrow\) interruptible exact algorithm
- Postprocessing
Heuristic “anytime” algorithm
  ▶ Guided elimination order brute forcing
  ▶ INTERRUPTIBLE interruptible exact algorithm
  ▶ Postprocessing
Exact algorithm, recycling
  ▶ Rule based preprocessor
Heuristic "anytime" algorithm
- Guided elimination order brute forcing
- \( \mapsto \) 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
- ⟷ 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
  ➤ \(\mapsto\) 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
  ➤ \(\mapsto\) 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)
**MINIMUM FILL IN: PACE 2017 B**

By Christian Komusiewicz

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

<table>
<thead>
<tr>
<th>First</th>
<th>1</th>
<th>Last</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
</tr>
<tr>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>26</td>
<td>27</td>
<td>28</td>
</tr>
<tr>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>35</td>
<td>36</td>
<td>37</td>
</tr>
<tr>
<td>38</td>
<td>39</td>
<td>40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3.00</th>
<th>WA</th>
<th>527.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1,017.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,052.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>79.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>16.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>578.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,754.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>320.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>134.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>439.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>426.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>712.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WA</td>
</tr>
<tr>
<td></td>
<td></td>
<td>101.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>965.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,408.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>559.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,570.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>122.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,278.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>423.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>387.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>654.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>965.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,408.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>559.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,570.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>122.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,278.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>423.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>387.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>654.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>965.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,408.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>559.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,570.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>122.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,278.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>423.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>387.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>654.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>965.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,408.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>559.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,570.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>122.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,278.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>423.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>387.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>654.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>521.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>965.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,408.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>559.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1,570.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>63.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>307.00</td>
</tr>
<tr>
<td></td>
<td>TLE</td>
<td>122.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>144.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2,278.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>423.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>387.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>654.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLE</td>
</tr>
<tr>
<td></td>
<td></td>
<td>100.00</td>
</tr>
</tbody>
</table>

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
Results
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

1. University of Helsinki
2. Paris-Dauphine & MTA Hungary
3. ITU Copenhagen II
4. IIT Madras
5. IMS Chennai
6. University of Portsmouth
7. ITU Copenhagen
This is to certify that the 2017 PACE Program Committee has selected Jeremias Berg, Matti Järvisalo, Tuukka Korhonen 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
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)

Yasuaki Kobayashi  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 $|\text{miss}(S)| = 1$, where miss$(S)$ is the set of missing edges in $G[S]$. Then, there is an optimal solution that contains 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!