The PACE 2025 Parameterized Algorithms and Computational Experiments Challenge: Dominating Set and Hitting Set

Mario Grobler
University of Bremen, Germany

Sebastian Siebertz
University of Bremen, Germany

Abstract

The 10th iteration of the of the Parameterized Algorithms and Computational Experiments challenge (PACE) 2025 was devoted to engineer algorithms solving the Dominating Set problem as well as the Hitting Set problem. In contrast to the last iterations, these problems are (under standard assumptions) *not* fixed-parameter tractable (fpt) in general. However, restricting the structure of the input (e.g. to planar graphs or degenerate graphs for Dominating Set, or to set systems with sets of bounded size for Hitting Set) renders these problems fpt. Following the spirit of the last iterations of the PACE challenge, there is an exact track and a heuristic track for each problem; each track coming with a benchmark set of 100 public instances and 100 private instances. Overall, the PACE 2025 had 71 participants from 25 teams, 13 countries, and 3 continents. In this report, we briefly describe the setup of the challenge, the selection of benchmark instances, as well as the ranking of the participating teams. We also briefly outline the approaches used in the submitted solvers.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Graph algorithms analysis

Keywords and phrases PACE 2025 Report, Dominating Set, Hitting Set, Algorithm Engineering, FPT, Heuristics

Digital Object Identifier 10.4230/LIPIcs.IPEC.2025.32

Acknowledgements The PACE challenge was supported by Networks [28]. The prize money (€4000) was generously provided by Networks [28], an NWO Gravitation project of the University of Amsterdam, Eindhoven University of Technology, Leiden University and the Center for Mathematics and Computer Science (CWI). We are grateful to the whole optil.io team, especially to Jan Badura for the fruitful collaboration and for hosting the competition at the optil.io online judge system. Further, this work was supported by the Data Science Center of the University of Bremen (DSC@UB) funded by the State of Bremen.

1 Introduction

The Parameterized Algorithms and Computational Experiments (PACE) Challenge, now in its 10th iteration, continues its mission of advancing the practical impact of parameterized and fine-grained algorithm design. Since its inception in PACE 2016, which focused on Treewidth and Feedback Vertex Set, the challenge has presented a diverse sequence of computational problems, each selected for its theoretical richness and relevance to real-world algorithm engineering.

20th International Symposium on Parameterized and Exact Computation (IPEC 2025).
Editors: Akanksha Agrawal and Erik Jan van Leeuwen; Article No. 32; pp. 32:1–32:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

PACE 2016	TREEWIDTH and FEEDBACK VERTEX SET	[12];
PACE 2017	TREEWIDTH and MINIMUM FILL-IN	[13];
PACE 2018	STEINER TREE	[8];
PACE 2019	VERTEX COVER and HYPER-TREEWIDTH	[15];
PACE 2020	TREEDEPTH	[26];
PACE 2021	CLUSTER EDITING	[24];
PACE 2022	DIRECTED FEEDBACK VERTEX SET	[19];
PACE 2023	TWINWIDTH	[4];
PACE 2024	ONE-SIDED CROSSING MINIMIZATION	[25].

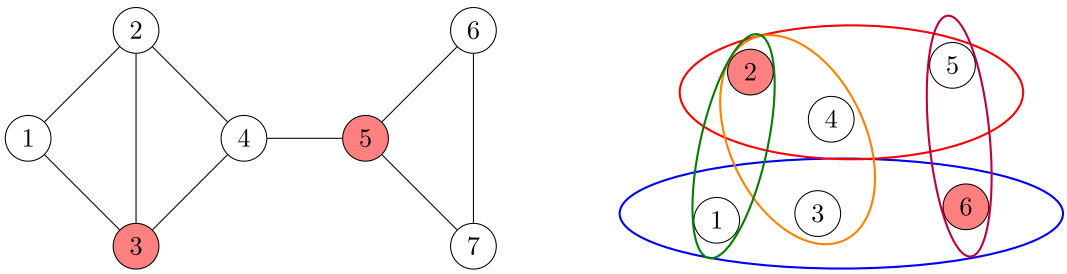
The overarching goal of PACE 2025 remains the same: to rigorously evaluate state-of-the-art algorithms for NP-complete problems, blending provable algorithmic guarantees with thorough experimental evaluation. This year's chosen problems, DOMINATING SET and HITTING SET, are canonical in complexity theory, embodying both theoretical depth and experimental challenge.

2 This year's problems

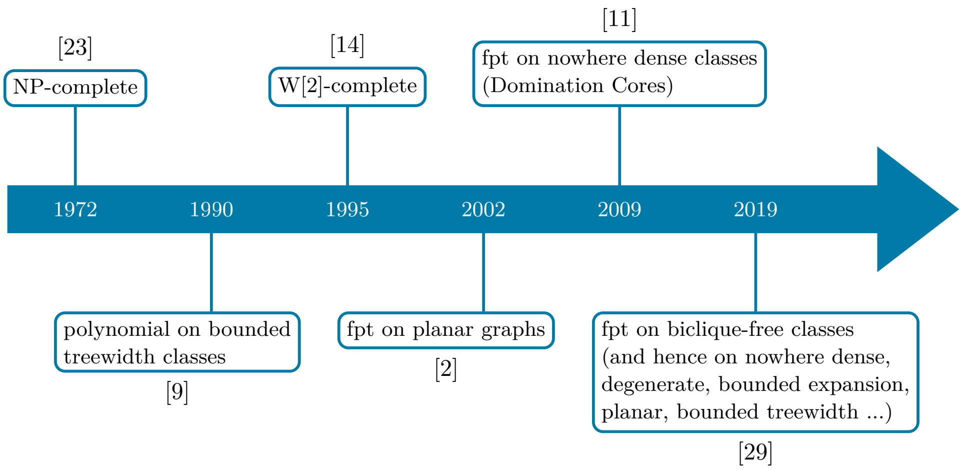
We briefly introduce the notation used. A *graph* G consists of its non-empty vertex set $V(G)$ and its edge set $E(G) \subseteq \binom{V(G)}{2}$. In particular, all graphs in this report are undirected, simple, and without loops. We define the (open) neighborhood of a vertex $v \in V(G)$ by $N(v) = \{u \in V(G) \mid \{u, v\} \in E(G)\}$. Similarly, we define the closed neighborhood of v by $N[v] = N(v) \cup \{v\}$. A *set system* $\mathcal{S}$ consists of its non-empty universe $V(\mathcal{S})$ and its set of sets $E(\mathcal{S}) \subseteq 2^{V(\mathcal{S})}$.

The DOMINATING SET and HITTING SET problems are classical NP-complete problems. In particular, HITTING SET was one of Karp's 21 NP-complete problems [23], and DOMINATING SET is NP-complete via its immediate equivalence to SET COVER [18], the dual of HITTING SET.

An instance of DOMINATING SET consists of an (undirected) graph G and an integer k, and the goal is to decide whether D admits a dominating set of size at most k, that is, a set $D \subseteq V(G)$ with $|D| \leq k$ such that $\bigcup_{v \in D} N[v] = V(G)$. DOMINATING SET is an important NP-complete problem known to be W[2]-complete in general [14] but fixed-parameter tractable on several restricted graph classes, see e.g. [2, 9, 11, 29]. In general, there is an algorithm that, for each k, counts the number of dominating sets of size k in $\mathcal{O}(1.5048^n)$ time [32].

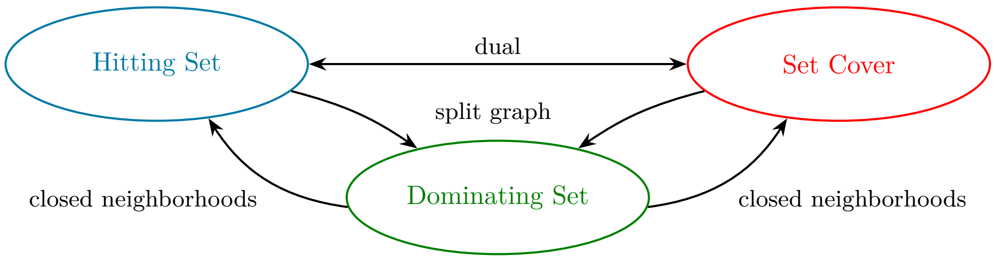


(a) A dominating set. **(b)** A hitting set.



An instance of Hitting Set consists of a set system $\mathcal{S}$ and an integer k, and the goal is to decide whether $\mathcal{S}$ admits a hitting set of size at most k, that is, a set $H \subseteq V(\mathcal{S})$ such that every set $S \in E(\mathcal{S})$ contains at least one vertex from H, that is, for all $S \in E(\mathcal{S})$ we have $S \cap H \neq \varnothing$. Hitting Set generalizes problems like Vertex Cover, Feedback Vertex Set, Dominating Set, and many other problems. It is both NP-complete [23] and W[2]-complete [14], and yet tractable under certain structural restrictions. For example, for 3-Hitting Set (i.e., Hitting Set where every set has cardinality 3), the problem is fpt; the currently fastest known algorithm is due to Tsur [31], yielding a runtime of $\mathcal{O}^*(2.0409^k)$ (where k is the solution size), which generalizes to an $\mathcal{O}^*(d-1+0.0409^k)$-time algorithm for d-Hitting Set (i.e., Hitting Set where every set has cardinality d), which is faster for $d \leq 6$ compared to the the algorithm of Fernau [17].

An instance of Set Cover consists of a set system $\mathcal{S}$ and an integer k, and the goal is to decide whether $\mathcal{S}$ admits a set cover of size at most k, that is, a subset $C \subseteq E(\mathcal{S})$ such that $\bigcup_{S\in C} S = V(\mathcal{S})$. Hitting Set, Dominating Set, and Set Cover are easily seen to be equivalent by standard polynomial-time reductions.



First, Set Cover and Hitting Set are equivalent: one can construct a Hitting Set instance from a Set Cover instance simply by swapping the roles of elements and sets, and vice versa, ensuring that feasible solutions correspond one-to-one. Second, Dominating Set reduces to Set Cover via the reduction that takes the universe to be the vertices of the graph and, for each vertex, introduces a subset consisting of its closed neighborhood. Then dominating sets correspond directly to set covers (of identical size). Finally, given a Set Cover instance $\mathcal{S}$, build the split graph $G(\mathcal{S})$ with vertex set $V(\mathcal{S}) \cup E(\mathcal{S})$ where $E(\mathcal{S})$ induces a clique, $V(\mathcal{S})$ is an independent set, and $S \in E(\mathcal{S})$ is adjacent to $u \in V(\mathcal{S})$ iff $u \in S$. Then dominating sets of $G(\mathcal{S})$ may be assumed to be subsets of $V(\mathcal{S})$ and correspond one-to-one to set covers of $\mathcal{S}$. Note that these correspondences are L-reductions (linear reductions), which also preserve approximation properties. Dominating Set, Hitting Set, and Set Cover parameterized by solution size are among the most frequently cited W[2]-complete problems in parameterized complexity [14].

3 The Setup of PACE 2025

The PACE 2025 competition followed the established multi-track format. The exact track demanded optimal solutions within strict resource limits, while the heuristic track emphasized solution quality under time pressure. Rigor in correctness was paramount: a small test set and verifier were released early, and a verification phase ensured solver reliability.

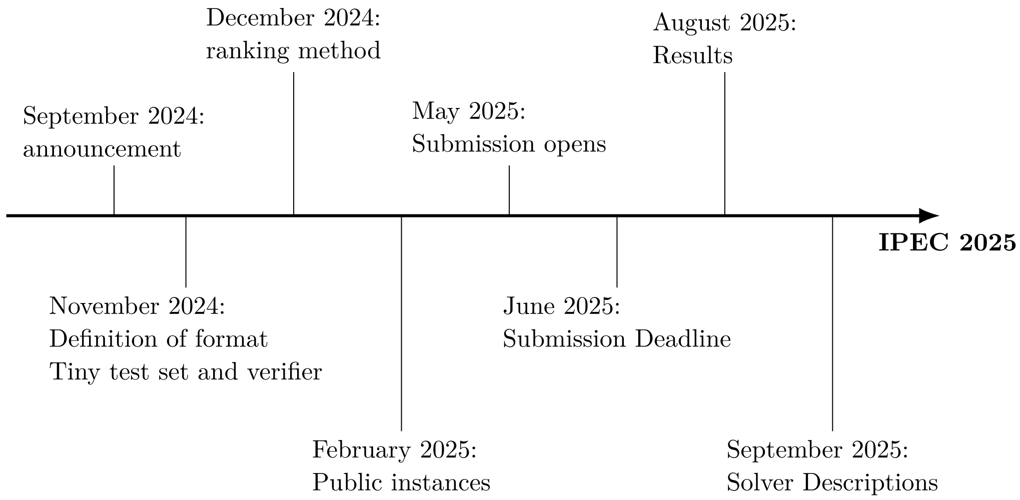


Figure 2 The timeline of PACE 2025.

3.1 Exact Tracks

Each exact track of PACE 2025 consisted of 200 benchmark instances, split into 100 public and 100 private instances. While the public instances were released for solver development and testing, the official evaluation was performed solely on the private set. Each solver was given a time limit of 30 minutes per instance, with an additional mercy time of 25 seconds, and was executed under a memory cap of 16 GB RAM, increased from 8 GB in previous years. To ensure reliability, an additional verification phase was conducted to verify correctness of submitted solvers. As in earlier editions, only solvers implementing provably correct algorithms were permitted, although formal proofs of correctness were not required. The winner of each track was determined by the number of solved private instances; in case of ties, the cumulative running time served as the tie-breaker.

3.2 Heuristic Tracks

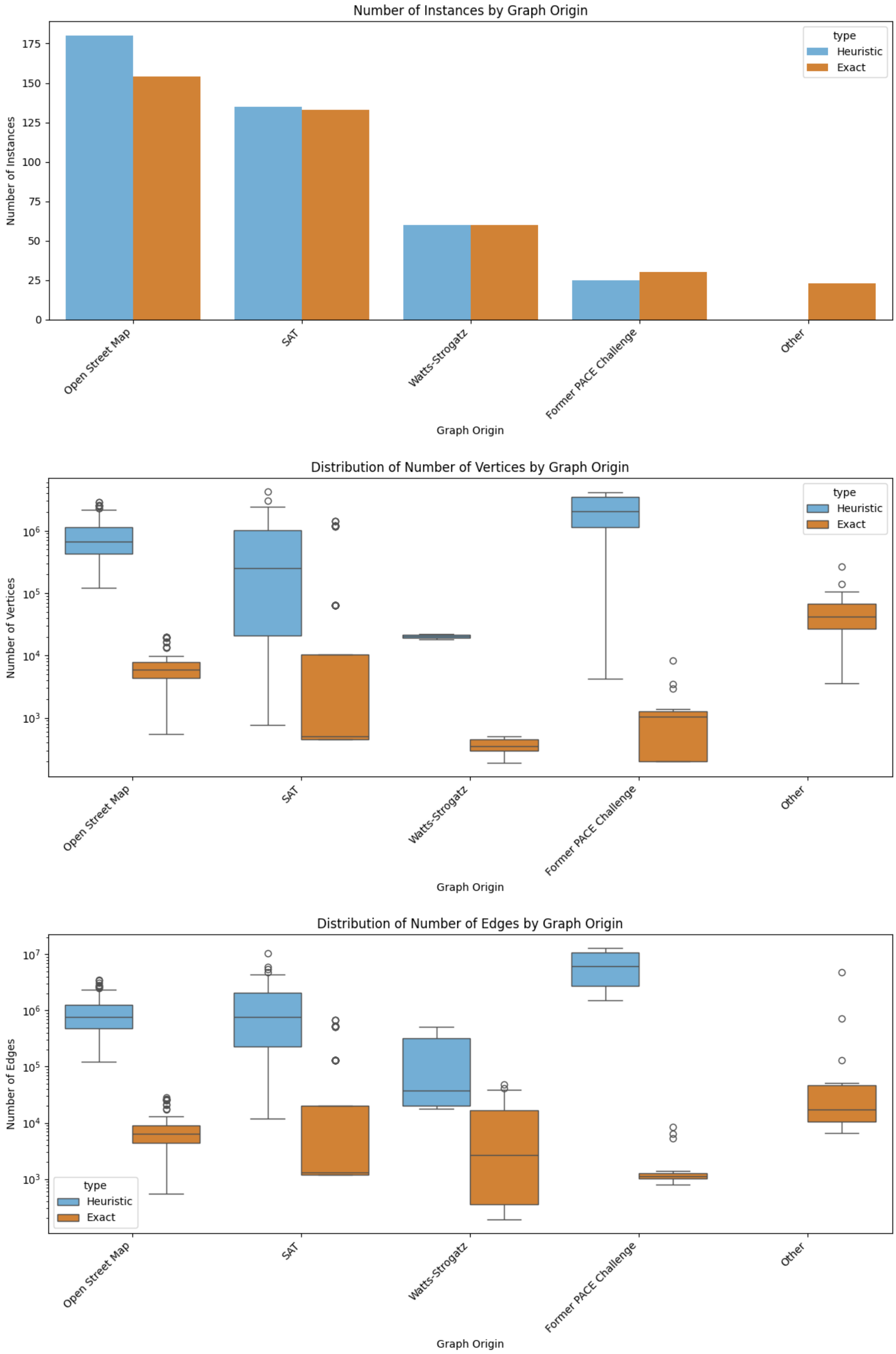
The heuristic tracks of PACE 2025 followed a setup similar to the exact tracks, with 200 benchmark instances in total, split into 100 public and 100 private instances. Each solver was given a time limit of 5 minutes per instance, after which a SIGTERM signal was sent, followed by an additional 20 seconds mercy time before a final SIGKILL was sent. The mercy time was primarily intended to ensure that solvers had a sufficient amount of time to write their output after termination was requested. As in the exact tracks, each solver was executed under a memory cap of 16 GB RAM, increased from 8 GB in previous years. Submissions were ranked by the sum over all instances of the scores computed by the function $f(k) = \left(\frac{u-k}{u-k^*}\right)^2$, where $u = \min\{n, 2k^*\}$, k^* is the best known solution value for the instance and k is the produced solution value. Hence, improving already good solutions was more beneficial than improving bad solutions, and instances where k^* is much smaller than n did not automatically yield a score of close-to-one.

3.3 Benchmark Set

The benchmark set of PACE 2025 was selected with the objective of providing instances with nice structural properties, rather than relying solely on random or synthetic constructions. The collection was drawn from a diverse set of sources:

- **OpenStreetMap (OSM)**. These instances are (nearly) planar graphs derived from road networks. Interestingly, many solvers were able to compute exact solutions very efficiently on such instances, even for comparatively large graphs.
- **SATLIB**. Graph instances obtained from reductions of SATLIB benchmarks [21], which inherit structural features of the underlying SAT formulas. Again, a large fraction of solvers was capable of producing exact solutions within short running times
- **Watts-Strogatz random graphs**. The Watts-Strogatz model generates graphs by starting from a regular ring lattice and randomly rewiring edges with a given probability, thereby interpolating between regular and small-world structures. In contrast to the previous categories, these instances proved particularly challenging: even relatively small graphs remained intractable for most solvers.
- **Previous PACE challenges.** A small subset of instances was reused from earlier editions, in particular from PACE 2019 (Vertex Cover) and PACE 2021 (Cluster Editing).
- **Other.** The remaining instances include e.g. social network graphs derived from real-world social networks, such as Facebook. Despite their large size, many of these graphs could be solved quickly, with optimal solution sizes typically being very small. Other instances orginate from action sequences or were obtained by taking disjoint unions of the aforementioned instances with additional random edges between the components.

The benchmark set was employed uniformly across both Dominating Set and Hitting Set, and across both exact and heuristic tracks. For the heuristic tracks, larger instances were selected. Where necessary, appropriate reductions were applied to adapt instances to the respective problem formulations. For example, SAT instances were reduced to Dominating Set or Hitting Set, either directly or via intermediate reductions such as Vertex Cover. This ensured that structurally rich instances could be shared across problems and tracks. We note that the public benchmark set did not include any Watts-Strogatz instances. All other categories were approximately balanced between public and private sets. As the other categories turned out to be quite easy for many solvers, there was a high chance that many teams would achieve perfect scores on the public instances. As Watts-Strogatz graphs turned out to be considerably more difficult, they were eventually incorporated into the private test set. Importantly, in the exact tracks of both Dominating Set and Hitting Set, every non-Watts-Strogatz instance was solved by at least one solver. The only unsolved cases were precisely the Watts-Strogatz instances: 14 in the Dominating Set track and 15 in the Hitting Set track, underlining their role as the most challenging instances.


Number of Instances by Graph Origin
Number of Instances
0
25
50
75
100
125
150
175
type
Heuristic
Exact
Open Street Map
SAT
Watts-Strogatz
Former PACE Challenge
Other
Graph Origin
Distribution of Number of Vertices by Graph Origin
Number of Vertices
10^3
10^4
10^5
10^6
type
Heuristic
Exact
Open Street Map
SAT
Watts-Strogatz
Former PACE Challenge
Other
Graph Origin
Distribution of Number of Edges by Graph Origin
Number of Edges
10^3
10^4
10^5
10^6
10^7
type
Heuristic
Exact
Open Street Map
SAT
Watts-Strogatz
Former PACE Challenge
Other
Graph Origin

3.4 Verification Phase and Verification Set

For the first time in the history of the PACE challenge, a dedicated verification phase was introduced. In this phase, all submitted solvers were made publicly available to the community, and participants were explicitly invited to test each other's implementations in order to identify possible bugs. Moreover, all solvers were evaluated against a large verification dataset, which was not used for scoring but served as an additional safeguard: solvers that produced incorrect solutions on any of these instances were allowed to be repaired in case of minor fixes and otherwise disqualified from the competition.

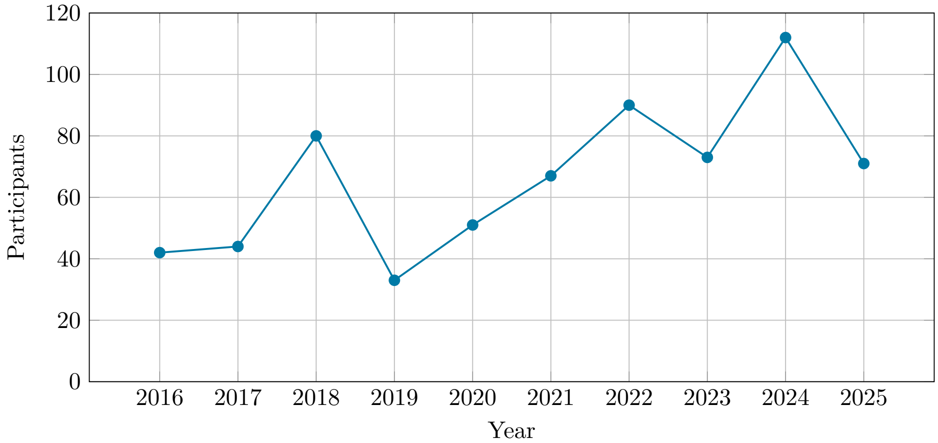
The verification set included large numbers of random graphs generated under different generative models, such as the Erdős-Rényi model [16], the Barabási-Albert model [5], the Watts-Strogatz model [33], and power-law cluster graphs following the algorithm of Holme and Kim [20], as well as uniform intersection graphs and several additional random graph models. In addition, the dataset contained named graphs such as grids, trees, ladders, cliques, paths, and wheels. Finally, a randomly selected subset of the large-scale STRIDE dataset (Sharing and Testing Repository for Instance Deployment and Evaluation, available at `https://domset.algorithm.engineering/`), which is maintained by Manuel Penschuck.

3.5 Evaluation Infrastructure

For baseline testing, the organizers employed an internal solver combining a set of problem-specific reduction rules with the commercial ILP solver CPLEX [10]. For both problems, dedicated verifiers were made available to participants; these were implemented as Java/Maven projects and ensured the correctness of submitted solutions (however, optimality was not checked). All solvers were executed inside Docker containers, using a common template that was also released publicly. Each container provided an identical environment, based on Ubuntu with 16 GB RAM and a single CPU core, thereby guaranteeing portability and reproducibility. This homogeneous setup allowed all solvers to be executed seamlessly on the compute cluster of the Data Science Center at the University of Bremen (`https://www.dsc-ub.de/`), ensuring fair and consistent evaluation across teams.

4 Participants

PACE 2025 had 71 participants from 25 teams, 13 countries, and 3 continents. We list the results below. A graduation cap (🎓) indicates a team where students are the main contributors.



5 Results

5.1 Dominating Set – Exact Track

5.1.1 Ranking

Rank	Team	Score	Time	Link
1	**UzL** Max Bannach, Florian Chudigiewitsch, Marcel Wienöbst *European Space Agency, University of Lübeck*	81	46111 s	
2	**Bad Dominating Set Maker** Alexander Dobler, Simon D. Fink, Mathis Rocton *TU Wien*	80	50253 s	
3	**OBLX** Jona Dirks, Enna Gerhard, Victoria Kaial, Lucas Lorieau *Université Clermont-Auvergne, University of Bremen*	80	64579 s	
4	**Alpaca** Tobias Röhr *Hasso Plattner Institute, University of Potsdam*	76	50057 s	
5	**Shadoks** Guilherme D. da Fonseca, Fabien Feschet, Yan Gerard *Aix-Marseille Université, Université Clermont-Auvergne*	73	65713 s	
6	**PaceYourself** Lukas Geis, Alexander Leonhardt, Johannes Meintrup, Ulrich Meyer, Manuel Penschuck *Goethe University Frankfurt, Technische Hochschule Mittelhessen*	71	83392 s	
7	**Swats** Sylwester Swat *Poznań University of Technology*	64	70325 s	
8	**Floris** Floris van der Hout *Utrecht University*	62	92309 s	
9	**AEG Heidelberg** Adil Chhabra, Marlon Dittes, Ernestine Großmann, Kenneth Langedal, Henrik Reinstädtler, Christian Schulz, Darren Strash, and Henning Woydt *Heidelberg University, Hamilton College*	48	107050 s	
10	**DSHunter** Paweł Putra *University of Warsaw*	22	142228 s	
11	**fu2025_pace** Yi-Hui Fan, Arthur Gisbrecht, Rashid Harvey, Vinzent Jörß, Bernadette Keßler, Oliver Seitz *FU Berlin*	19	163971 s	

12	**Elias & Thor** 🎓 Elias Rasmussen, Thor Vejen Eriksen *University of Copenhagen*	18	156974 s	GitHub
DQ	**Marina, Viktoriia & Fedor** 🎓 Marina Fridman, Viktoriia Krivogornitsyna, Fedor Kurmazov *Saint Petersburg State University of Economics*			GitHub

5.1.2 Short Description of the Dominating Set Exact Track Winning Solvers

For the **UzL** solver and the **Bad Dominating Set Maker**, we refer to Section 5.3.2, as these solvers are also the winners of the Hitting Set Exact Track, and there are only minor differences in the strategies employed for Dominating Set and Hitting Set.

The **OBLX** solver uses an extended version of Dominating Set and applies a large set of reduction rules, both well-known rules and rules especially designed for planar graphs. It then reduces the resulting instance to Hitting Set and further to weighted Max-Sat, which is then solved using the `EvalMaxSat` solver [3]. Using the result, a postprocessing routine to revert the modifications of the previously applied reduction rules is applied on the solution. This step is necessary in order to obtain the solution for the original Dominating Set instance.

5.2 Dominating Set – Heuristic Track

5.2.1 Ranking

Rank	Team	Score	Time	Link
1	**Florian & Guillaume** Florian Fontan, Guillaume Verger	99.81	28553 s	GitHub
2	**Root** Canhui Luo, Zhipeng Lv, Zhouxing Su, Qingyun Zhang *Huazhong University of Science and Technology*	99.64	30035 s	GitHub
3	**Swats** Sylwester Swat *Poznań University of Technology*	99.35	22062 s	GitHub
4	**Shadoks** Guilherme D. da Fonseca, Fabien Feschet, Yan Gerard *Aix-Marseille Université, Université Clermont-Auvergne*	99.02	31545 s	GitHub
5	**AEG Heidelberg** Adil Chhabra, Marlon Dittes, Ernestine Großmann, Kenneth Langedal, Henrik Reinstädtler, Christian Schulz, Darren Strash, and Henning Woydt *Heidelberg University, Hamilton College*	98.06	30014 s	GitHub

6	**PaceYourself** Lukas Geis, Alexander Leonhardt, Johannes Meintrup, Ulrich Meyer, Manuel Penschuck *Goethe University Frankfurt, Technische Hochschule Mittelhessen*	96.92	30024 s	
7	**Samuel** Samuel Füßinger *Eberhard Karls Universität Tübingen*	96.80	30017 s	
8	**Viacheslav** Viacheslav Khrushchev *HSE University Moscow*	96.27	30024 s	
9	**Hui, Bo, Yexin & Xinyun** Hui Kong, Bo Peng, Yexin Peng, Xinyun Wu *Hubei University of Technology, Southwestern University of Finance and Economics*	96.16	30025 s	
10	**Greeduce** Adam Polak, Jonas Schmidt *Bocconi University*	95.82	30016 s	
11	**Aslam, Ashwin, Sahil, Nithin, Pankaj & Edwin** Aslam Ameen, Ashwin Jacob, Sahil Muhammed, Nithin R, Pankaj Kumar R, Edwin Thomas *National Institute of Technology Calicut*	88.99	150 s	
12	**Zhaojie, Zhipeng, Zhouxing & Qingyun** Zhaojie Liu, Zhipeng Lv, Zhouxing Su, Qingyun Zhang *Huazhong University of Science and Technology*	87.90	1054 s	
13	**Marcin** Marcin Mennemann *TU Dortmund*	75.06	30019 s	
14	**fu2025_pace** Yi-Hui Fan, Arthur Gisbrecht, Rashid Harvey, Vinzent Jörß, Bernadette Keßler, Oliver Seitz *FU Berlin*	31.11	18421 s	
DQ	**fu2025_pace** Yi-Hui Fan, Arthur Gisbrecht, Rashid Harvey, Vinzent Jörß, Bernadette Keßler, Oliver Seitz *FU Berlin*			

5.2.2 Short Description of the Dominating Set Heuristic Track Winning Solvers

Florian & Guillaume solve both Dominating Set and Hitting Set instances as Set Cover instances. In a first step they apply a set of reduction rules. They then compute greedily an initial solution, where a greedy heuristic is chosen depending on an inital very fast estimation of the solution size. Following up, a large neighborhood search is perfomed to very quickly improve the solution. Finally, a local search approach is used to improve the current solution to get the best possible solution.

Also **Root** unifies Dominating Set and Hitting Set by transforming them into Set Cover and applies a small set of classical reduction rules. It then generates a high-quality starting solution by applying a multi-round greedy heuristic guided by element frequencies. Finally, it iteratively improves the solution by local search with with an adaptive weighting approach.

Swats formulates both Dominating Set and Hitting Set as a generalized Dominating Set problem. It first applies a wealth of reduction rules, both known rules, as well as a variety of new reduction rules that were designed, optimized and implemented to be applicable even for very large graph instances. It then proceeds in an counter-example guided abstraction refinement (CEGAR) fashion. Iteratively it adds a subset of unsatisfied constraints and updates a solution to satisfy these constraints. This solution is found by a local search approach and subsequently improved by local search. If the size of the solution could not be improved for a few subsequent iterations, a state perturbation is applied to try to get out of a found local optimum.

5.3 Hitting Set – Exact Track

5.3.1 Ranking

Rank	Team	Score	Time	Link
1	**UzL** Max Bannach, Florian Chudigiewitsch, Marcel Wienöbst *European Space Agency, University of Lübeck*	79	49854 s	
2	**Bad Dominating Set Maker** Alexander Dobler, Simon D. Fink, Mathis Rocton *TU Wien*	79	57783 s	
3	**André** André Schidler *Albert-Ludwigs-Universität Freiburg*	78	49565 s	
4	**Alpaca** Tobias Röhr *Hasso Plattner Institute, University of Potsdam, Germany*	76	50947 s	
5	**Shadoks** Guilherme D. da Fonseca, Fabien Feschet, Yan Gerard *Aix-Marseille Université, Université Clermont-Auvergne*	74	60729 s	
6	**Swats** Sylwester Swat *Poznań University of Technology*	64	67148 s	
7	**AEG Heidelberg** Adil Chhabra, Marlon Dittes, Ernestine Großmann, Kenneth Langedal, Henrik Reinstädtler, Christian Schulz, Darren Strash, and Henning Woydt *Heidelberg University, Hamilton College*	63	77045 s	

5.3.2 Short Description of the Hitting Set Exact Track Winning Solvers

The **UzL** solver starts by applying four well-known (and fast) reduction rules before reformulating the problem as a (partial) Max-Sat problem. Afterwards, the Max-Sat solver `EvalMaxSat` [3] is used to solve the resulting Max-Sat instance. If the instance originates from the Dominating Set benchmark, an additional proprocessing step involving the preprocessor `maxpre2` [22] is done. The internal parameters of `EvalMaxSat` have been tweaked; in particular, the time spent on conflict minimization has been bounded. If the resulting Hitting Set instance is an instance of 2-Hitting Set (i.e., a Vertex Cover instance), similar approaches to the ones utilized in the PACE 2019 have been employed.

The **Bad Dominating Set Maker** first converts the input instance (both, Dominating Set and Hitting Set) into an equivalent Directed Constrained Domination instance. Afterwards, a large set of reduction rules is applied. Then, `hdt` [1] is used to compute a tree decomposition. If the resulting decomposition is of width at most 13, a dynamic programming approach is used to compute a solution. Otherwise, the solver checks whether the instance is a Vertex Cover instance, in which case the PACE 2019 solver `peaty` [30] is employed. If all the aforementioned condition do not hold, the Max-Sat solver `EvalMaxSat` [3] with some adjustments is used.

The solver of **André Schidler** first applies two well-known (and fast) reduction rules. Afterwards, based on the size of the instance, either a Max-Sat solver or a Mixed Linear Programm (MLP) solver is utilized: if the reduced instance has no more than 250 elements, the instances is directly translated into an MLP instances and solved using the MLP solver `SCIP` [6]. Otherwise a self-implemented Max-Sat solver is used, which relies on the OLL algorithm [27] and internally uses the Sat solver Cadical 2.1.3. [7]. The key to good OLL performance relies on a strong initial lower bound which is obtained by finding a large set of disjoint sets. To speed up things further, both, the MLP solver as well as the Max-Sat solver are warm-started with an upper bound which is obtained using a local search approach.

5.4 Hitting Set – Heuristic Track

5.4.1 Ranking

Rank	Team	Score	Time	Link
1	**Root** Canhui Luo, Zhipeng Lv, Zhouxing Su, Qingyun Zhang *Huazhong University of Science and Technology*	99.79	30024 s	
2	**Florian & Guillaume** Florian Fontan, Guillaume Verger	99.73	29535 s	
3	**Shadoks** Guilherme D. da Fonseca, Fabien Feschet, Yan Gerard *Aix-Marseille Université, Université Clermont-Auvergne*	99.21	31522 s	
4	**Swats** Sylwester Swat *Poznań University of Technology*	99.18	23089 s	

5	**AEG Heidelberg** Adil Chhabra, Marlon Dittes, Ernestine Großmann, Kenneth Langedal, Henrik Reinstädtler, Christian Schulz, Darren Strash, and Henning Woydt *Heidelberg University, Hamilton College*	98.01	30013 s	
6	**Greeduce** Adam Polak, Jonas Schmidt *Bocconi University*	95.73	30011 s	
7	**Deepak, Syed & Saurabh** Deepak Ajwani, Syed Mahmudul Kabir Ratul, Saurabh Ray *New York University Abu Dhabi, University College Dublin*	82.50	29464 s	
8	**Sebastian, Mirza, Patrick & Mariette** Sebastian Angrick, Mirza Redzic, Patrick Steil, Mariette Vasen *Karlsruhe Institute of Technology*	1.83	32055 s	

5.4.2 Short Description of the Hitting Set Heuristic Track Winning Solvers

Root and **Florian & Guillaume** reduce both Dominating Set and Hitting Set to Set Cover and used the same solver for both problems, see Section 5.2.2.

Shadoks applies a small set of standard reduction rules. It then ties to solve each connected component exactly with an MIP or MAXSAT solver, from small to large, and aborting after a while if no solution is found in due time. It then reduces the problem to MAXSAT and runs an anytime MAXSAT heuristic solver for around a minute. Finally, it iteratively uses a local search improvement until the time runs out.

6 Student Ranking

Besides the global ranking there is a student ranking, which is only eligible for submission where students are the main contributors, for example, teams of student being part of a student project. These student groups may be supervised by non-student people as long as the students are the main contributors. In each trach the three highest ranked student student teams are the winners of the student category.

7 Price Money

The price money of 4000€ was generously provided by Networks [28] and is distributed as follows.

- 1rst Price: 300€
- 2nd Price: 200€
- 3rd Price: 100€

- 1rst Student Price: 200€
- 2nd Student Price: 150€
- 3rd Student Price: 100€

8 PACE Organsition

The program committee of PACE 2025 consisted of Mario Grobler and Sebastian Siebertz, both from the University of Bremen. During the competition, the members of the steering committee were:

- (since 2023) Max Bannach (European Space Agency)
- (since 2023) Sebastian Berndt (Universität zu Lübeck)
- (since 2016) Holger Dell (Goethe University Frankfurt and IT University of Copenhagen)
- (since 2016) Bart M. P. Jansen (chair) (Eindhoven University of Technology)
- (since 2024) Philipp Kindermann (Universität Trier)
- (since 2021) André Nichterlein (Technical University of Berlin)
- (since 2022) Christian Schulz (Universität Heidelberg)
- (since 2024) Soeren Terziadis (TU Eindhoven)

The Program Committee of PACE 2026 will be chaired by Alexander Leonhardt, Manuel Penschuck, and Mathias Weller.

9 Conclusion

We would like to sincerely thank all participants of PACE 2025 for their remarkable contributions, creative approaches, and their patience in the face of occasional technical issues. We look forward to future editions that will continue to build bridges between theory and practice, and we are excited to see how innovative solvers and new ideas will further shape the field.

9.1 Lessons Learned

Organizing the 10th iteration of the PACE Challenge has also provided valuable insights. It turned out to be highly challenging to design a benchmark set that is both structurally rich and well balanced in difficulty; while some classes of instances were solved quickly by nearly all solvers, others (notably the Watts–Strogatz graphs) proved unexpectedly difficult. Closely related, although creating a timeline for the organization is straightforward, adhering to it in practice turned out to be considerably harder. The discrepancy between the public and private instance sets, again caused by the delayed inclusion of Watts–Strogatz graphs, understandably led to surprise and discussion among participants.

On the technical side, the process of setting up Docker containers for all solvers was very time-consuming, but the effort clearly paid off: it ensured portability, reproducibility, and a fair evaluation across the homogeneous infrastructure provided by the Data Science Center at the University of Bremen. Finally, the verification phase to ensure correctness of solvers proved to be extremely valuable. It revealed a range of issues, including small implementation bugs such as failure to handle comment lines in input files, conceptual oversights such as missing coverage of corner cases, and configuration problems such as insufficient numerical precision of external ILP solvers. Thanks to this process, nearly all issues could be resolved before the official evaluation, and in the end only two solvers had to be disqualified for producing incorrect or suboptimal solutions in the exact track. We thank Manuel Penschuck for preparing the STRIDE repository and for his valuable remarks in catching the aforementioned bugs!

9.2 Outlook

Despite their difficulty, the Watts–Strogatz instances also provided particularly valuable insights. In total, 14 instances in Dominating Set and 15 instances in Hitting Set could not be solved by any solver. At the same time, the best-performing solvers solved only 81 instances (DS) and 79 instances (HS), respectively. This implies that even the strongest solvers failed on at least five, resp. six, instances that some other solver was able to handle successfully. Thus, no single solver dominated across the entire instance set, and the comparison revealed genuine diversity in algorithmic strengths.

This observation opens a promising avenue for future research. Since different solvers excelled on different subsets of the benchmark, it becomes natural to ask whether the underlying algorithmic ideas and engineering concepts can be combined. A solver that unifies the complementary strengths of existing methods might be capable of solving all instances that were solved by at least one team.

The experiences from PACE 2025 reinforce the importance of combining strong theoretical foundations with robust engineering practices. Future iterations of the PACE Challenge will continue to build on these lessons, refining the evaluation methodology, strengthening the benchmarking process, and fostering collaboration across different areas of algorithm design. We are looking forward to PACE 2026!

References

1 Michael Abseher, Nysret Musliu, and Stefan Woltran. htd – A free, open-source framework for (customized) tree decompositions and beyond, 2017.

2 Jochen Alber, Hongbing Fan, Michael R. Fellows, Henning Fernau, Rolf Niedermeier, Fran Rosamond, and Ulrike Stege. A refined search tree technique for dominating set on planar graphs. *Journal of Computer and System Sciences*, 71(4):385–405, 2005. doi:10.1016/J.JCSS.2004.03.007.

3 Florent Avellaneda. A short description of the solver evalmaxsat, 2020.

4 Max Bannach and Sebastian Berndt. PACE Solver Description: The PACE 2023 Parameterized Algorithms and Computational Experiments Challenge: Twinwidth. In *18th International Symposium on Parameterized and Exact Computation (IPEC 2023)*, volume 285 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:14, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.IPEC.2023.35.

5 Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

6 Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 8.0. Technical report, Optimization Online, 2021. URL: http://www.optimization-online.org/DB_HTML/2021/12/8728.html.

7 Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. Cadical 2.0. In *Computer Aided Verification*, pages 133–152, Cham, 2024. Springer Nature Switzerland. doi:10.1007/978-3-031-65627-9_7.

8 Édouard Bonnet and Florian Sikora. The PACE 2018 Parameterized Algorithms and Computational Experiments Challenge: The Third Iteration. In *13th International Symposium on Parameterized and Exact Computation (IPEC 2018)*, volume 115 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPICS.IPEC.2018.26`.

9 Bruno Courcelle. The monadic second-order logic of graphs. i. recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. `doi:10.1016/0890-5401(90)90043-H`.

10 IBM ILOG Cplex. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

11 Anuj Dawar and Stephan Kreutzer. Domination Problems in Nowhere-Dense Classes. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 4 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 157–168, Dagstuhl, Germany, 2009. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPICS.FSTTCS.2009.2315`.

12 Holger Dell, Thore Husfeldt, Bart M. P. Jansen, Petteri Kaski, Christian Komusiewicz, and Frances A. Rosamond. The First Parameterized Algorithms and Computational Experiments Challenge. In *11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:9, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPICS.IPEC.2016.30`.

13 Holger Dell, Christian Komusiewicz, Nimrod Talmon, and Mathias Weller. The PACE 2017 Parameterized Algorithms and Computational Experiments Challenge: The Second Iteration. In *12th International Symposium on Parameterized and Exact Computation (IPEC 2017)*, volume 89 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:12, Dagstuhl, Germany, 2018. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPICS.IPEC.2017.30`.

14 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: basic results. *SIAM Journal on Computing*, 24(4):873–921, 1995. `doi:10.1137/S0097539792228228`.

15 M. Ayaz Dzulfikar, Johannes K. Fichte, and Markus Hecher. The PACE 2019 Parameterized Algorithms and Computational Experiments Challenge: The Fourth Iteration. In *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:23, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.IPEC.2019.25`.

16 P. Erdös and A. Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6, 1959.

17 Henning Fernau. A top-down approach to search-trees: Improved algorithmics for 3-hitting set. *Algorithmica*, 57(1):97–118, 2010. `doi:10.1007/S00453-008-9199-6`.

18 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979.

19 Ernestine Großmann, Tobias Heuer, Christian Schulz, and Darren Strash. The PACE 2022 Parameterized Algorithms and Computational Experiments Challenge: Directed Feedback Vertex Set. In *17th International Symposium on Parameterized and Exact Computation (IPEC 2022)*, volume 249 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:18, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPICS.IPEC.2022.26`.

20 Petter Holme and Beom Jun Kim. Growing scale-free networks with tunable clustering. *Phys. Rev. E*, 65, 2002.

21 Holger H. Hoos and Thomas Stützle. Satlib: An online resource for research on sat. In *SAT2000*, pages 283–292. IOS Press, 2000. URL: `www.satlib.org`.

22 Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo. Clause redundancy and preprocessing in maximum satisfiability. In *Automated Reasoning: 11th International Joint Conference, IJCAR 2022, Haifa, Israel, August 8–10, 2022, Proceedings*, pages 75–94. Springer-Verlag, 2022. doi:10.1007/978-3-031-10769-6_6.

23 Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.

24 Leon Kellerhals, Tomohiro Koana, André Nichterlein, and Philipp Zschoche. The PACE 2021 Parameterized Algorithms and Computational Experiments Challenge: Cluster Editing. In *16th International Symposium on Parameterized and Exact Computation (IPEC 2021)*, volume 214 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:18, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.IPEC.2021.26.

25 Philipp Kindermann, Fabian Klute, and Soeren Terziadis. The PACE 2024 Parameterized Algorithms and Computational Experiments Challenge: One-Sided Crossing Minimization. In *19th International Symposium on Parameterized and Exact Computation (IPEC 2024)*, volume 321 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.IPEC.2024.26.

26 Lukasz Kowalik, Marcin Mucha, Wojciech Nadara, Marcin Pilipczuk, Manuel Sorge, and Piotr Wygocki. The PACE 2020 Parameterized Algorithms and Computational Experiments Challenge: Treedepth. In *15th International Symposium on Parameterized and Exact Computation (IPEC 2020)*, volume 180 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:18, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPICS.IPEC.2020.37.

27 Antonio Morgado, Carmine Dodaro, and Joao Marques-Silva. Core-guided maxsat with soft cardinality constraints. In *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Proceedings*, Lecture Notes in Computer Science, pages 564–573. Springer Verlag, 2014.

28 Networks Project, 2017. URL: https://www.thenetworkcenter.nl.

29 J.A. Telle and Y. Villanger. Fpt algorithms for domination in sparse graphs and beyond. *Theoretical Computer Science*, 770:62–68, 2019. doi:10.1016/J.TCS.2018.10.030.

30 James Trimble. jamestrimble/peaty: v1.0.0: Pace challenge 2019 entry, 2019.

31 Dekel Tsur. Faster parameterized algorithms for variants of 3-hitting set. *Journal of Combinatorial Optimization*, 49(4):1–14, 2025.

32 Johan M. M. van Rooij, Jesper Nederlof, and Thomas C. van Dijk. Inclusion/exclusion meets measure and conquer. In *Algorithms - ESA 2009*, pages 554–565. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-04128-0_50.

33 Duncan J. Watts and Steven H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393(6684):440–442, 1998.