18th International Symposium on Parameterized and Exact Computation

IPEC 2023, September 6–8, 2023, Amsterdam, The Netherlands

Edited by

Neeldhara Misra

Magnus Wahlström

LIPICS

Editors

Neeldhara Misra
IIT Gandhinagar, India
neeldhara.m@iitgn.ac.in

Magnus Wahlström
Royal Holloway, University of London, UK
Magnus.Wahlstrom@rhul.ac.uk

ACM Classification 2012
Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Approximation algorithms analysis; Theory of computation → Graph algorithms analysis; Theory of computation → Mathematical optimization; Theory of computation → Algorithm design techniques

ISBN 978-3-95977-305-8

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at https://www.dagstuhl.de/dagpub/978-3-95977-305-8.

Publication date
December, 2023

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at https://portal.dnb.de.

Digital Object Identifier: 10.4230/LIPIcs.IPEC.2023.0

ISBN 978-3-95977-305-8 **ISSN 1868-8969** **https://www.dagstuhl.de/lipics**

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 1868-8969

https://www.dagstuhl.de/lipics

Contents

Regular Papers

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

PACE Solver Descriptions

Preface

The International Symposium on Parameterized and Exact Computation (IPEC, formerly IWPEC) is a series of international symposia covering research in all aspects of parameterized and exact algorithms and complexity. It started in 2004 as a biennial workshop and became an annual event in 2009. Previous iterations of the symposium were:

- 2004 Bergen, Norway
- 2006 Zürich, Switzerland
- 2008 Victoria, Canada
- 2009 Copenhagen, Denmark
- 2010 Chennai, India
- 2011 Saarbrücken, Germany
- 2012 Lubljana, Slovenia
- 2013 Sophia Antipolis, France
- 2014 Wrocław, Poland
- 2015 Patras, Greece
- 2016 Aarhus, Denmark
- 2017 Vienna, Austria
- 2018 Helsinki, Finland
- 2019 Munich, Germany
- 2020 virtual / Hong Kong, China
- 2021 virtual / Lisbon, Portugal
- 2022 Potsdam, Germany

This volume contains the papers presented at IPEC 2023: the 18th International Symposium on Parameterized and Exact Computation. IPEC 2023 was held on September 6—8 (Wed to Fri) as part of ALGO 2023, and took place in Amsterdam, the Netherlands at Centrum Wiskunde & Informatica (CWI). In response to the call for papers, 85 extended abstracts were registered, of which 10 were withdrawn or otherwise failed to submit a full version. The resulting number of 75 full submissions represents a significant increase in interest in the conference compared to previous years. 34 papers were ultimately selected for presentation at the conference and inclusion in these proceedings. The reviews were performed in a double-blind fashion, and there were 106 external reviews out of a total of 223 reviews.

The **Best Paper Award** was given to Hans L. Bodlaender (Utrecht University), Isja Mannens (Utrecht University), Jelle Oostveen (Utrecht University), Sukanya Pandey (Utrecht University) and Erik Jan van Leeuwen (Utrecht University) for their paper "The Parameterised Complexity of Integer Multicommodity Flow". The **Best Student Paper Award** was given to Stefan Kratsch (Humboldt-Universität zu Berlin) and Pascal Kunz (Humboldt-Universität zu Berlin) for their paper "Approximate Turing kernelization and lower bounds for domination problems". The **EATCS-IPEC Nerode Prize** was given to Marek Cygan (University of Warsaw and Nomagic), Jesper Nederlof (Utrecht University), Marcin Pilipczuk (University of Warsaw), Michał Pilipczuk (University of Warsaw), Johan M. M. van Rooij (Utrecht University) and Jakub Onufry Wojtaszczyk (Google) for their paper "Solving Connectivity Problems Parameterized by Treewidth in Single Exponential Time". IPEC 2023 hosted an award ceremony with a talk given jointly by Michał Pilipczuk and Johan M. M. van Rooij. The Nerode Prize committee consisted of Fedor Fomin (chair; University of Bergen), Thore Husfeldt (IT University of Copenhagen) and Sang-il Oum (Korea Advanced Institute of Science and Technology). Tuukka Korhonen (University of Bergen) presented an **invited tutorial** on "New methods in FPT algorithms for treewidth". Finally, IPEC 2023 hosted the award ceremony of the eighth Parameterized Algorithms and Computational Experiments (PACE) challenge. These proceedings contain a report on the **PACE 2023** challenge and brief communications of the winners about their solvers.

We thank the program committee and the external reviewers for their commitment in the paper selection process. We also thank all the authors who submitted their work. We are grateful to the local organizers of ALGO 2023 for the local arrangements.

Neeldhara Misra and Magnus Wahlström
Gandhinagar and London, October 2022

Program Committees

IPEC 2023 Program Committee

- Akanksha Agrawal (IIT Madras, India)
- Cristina Bazgan (Paris Dauphine University, France)
- Robert Bredereck (TU Clausthal, Germany)
- Eduard Eiben (Royal Holloway, University of London, UK)
- Archontia Giannopoulou (University of Athens, Greece)
- Pallavi Jain (IIT Jodhpur, India)
- Bart M. P. Jansen (Eindhoven University of Technology, Netherlands)
- Mark Jones (TU Delft, Netherlands)
- Christian Knauer (Universität Bayreuth, Germany)
- Dusan Knop (Czech Technical University, Czech Republic)
- Bingkai Lin (Nanjing University, China)
- Neeldhara Misra (IIT Gandhinagar, India) (co-chair)
- André Nichterlein (TU Berlin, Germany)
- Sebastian Ordyniak (University of Leeds, UK)
- Fahad Panolan (IIT Hydrebad, India)
- Daniel Paulusma (Durham University, UK)
- R.B. Sandeep (IIT Dharwad, India)
- Magnus Wahlström (Royal Holloway, University of London, UK) (co-chair)

IPEC 2023 Steering Committee

- Holger Dell (2021-24, chair)
- Fedor Fomin (2021-24)
- Petr Golovach (2020-23)
- Łukasz Kowalik (2022-25)
- Daniel Marx (2020-23)
- Neeldhara Misra (2022-25)
- Jesper Nederlof (2021-24)
- Magnus Wahlström (2022-25)
- Meirav Zehavi (2020-23)

PACE 2023 Program Committee

- Max Bannach (Universität zu Lübeck)
- Sebastian Berndt (Universität zu Lübeck)

List of External Reviewers

- Duncan Adamson
- Jungho Ahn
- Dhanyamol Antony
- Emmanuel Arrighi
- Pradeesha Ashok
- Rémy Belmonte
- Matthias Bentert
- Steffen van Bergerem
- Magnus Bordewich
- Sergio Cabello
- Huairui Chu
- Alexis de Colnet
- Alex Crane
- Radu Curticapean
- Argyrios Deligkas
- Riccardo Dondi
- Pål Grønås Drange
- Foivos Fioravantes
- Till Fluschnik
- Vincent Froese
- Harmender Gahlawat
- Jaroslav Garvardt
- Serge Gaspers
- Panos Giannopoulos
- Tiger-Lily Goldsmith
- Petr Golovach
- Maximilian Gorsky
- Siddharth Gupta
- Sushmita Gupta
- Thekla Hamm
- Tesshu Hanaka
- Samuel Hand
- Klaus Heeger
- Leo van Iersel
- Tanmay Inamdar
- Yuni Iwamasa
- Lars Jaffke
- Satyabrata Jana
- Rasmus Ibsen-Jensen
- Andrzej Kaczmarczyk
- Lawqueen Kanesh
- Anjeneya Swami Kare
- Tomohiro Koana
- Christian Komusiewicz
- Stefan Kratsch
- Simon Krogmann
- Ariel Kulik
- O-Joung Kwon
- Noleen Köhler
- Matyáš Křišťan
- Junjie Luo
- Jayakrishnan Madathil
- Diptapriyo Majumdar
- Barnaby Martin
- Andrés López Martínez
- Rogers Mathew
- Filippos Mavropoulos
- Marcelo Garlet Milani
- Pranabendu Misra
- Matthias Mnich
- Hendrik Molter
- Nils Morawietz
- Amer Mouawad
- Anthony Perez
- Théo Pierron
- Evangelos Protopapas
- Lars Rohwedder
- Sanjukta Roy
- Abhishek Sahu
- Saket Saurabh
- Sanjay Seetharaman
- Sebastian Siebertz
- Kirill Simonov
- Fiona Skerman
- Ramanujan M. Sridharan
- Ramanujan Sridharan
- Giannos Stamoulis
- Fabian Stehn
- Céline Swennenhuis
- Prafullkumar Tale
- Ioan Todinca
- Vikash Tripathi
- Oxana Tsidulko
- Rao B V
- Ruben F.A. Verhaegh
- Christopher Weyand
- Hongxun Wu
- Karol Węgrzycki
- Jie Xue
- Yongjie Yang
- Chihao Zhang
- Dimitris Zoros

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

List of Authors

Emmanuel Arrighi (1, 2, 39)
University of Bergen, Norway; University of Trier, Germany

Max Bannach (3, 35)
European Space Agency, Advanced Concepts Team, Noordwijk, The Netherlands

Matthias Bentert (2, 4)
University of Bergen, Norway

Sebastian Berndt (35)
Institute for Theoretical Computer Science, University of Lübeck, Germany

Gaétan Berthe (38)
LIRMM, CNRS, Université de Montpellier, France

Sriram Bhyravarapu (5)
The Institute of Mathematical Sciences, HBNI, Chennai, India

Hans L. Bodlaender (6, 7)
Utrecht University, The Netherlands

Édouard Bonnet (7, 8, 40)
LIP, ENS Lyon, France

Guilherme C. M. Gomes (9)
Department of Computer Science, Federal, University of Minas Gerais, Belo Horizonte, Brazil

Tom C. van der Zanden (9)
Department of Data Analytics and Digitalisation, Maastricht University, The Netherlands

Juhi Chaudhary (10)
Ben-Gurion University of the Negev, Beersheba, Israel

Florian Chudigiewitsch (3)
Universität zu Lübeck, Germany

Yoann Coudert-Osmont (38)
Université de Lorraine, CNRS, Inria, LORIA, France

Konrad K. Dabrowski (11)
School of Computing, Newcastle University, UK

Holger Dell (37)
Goethe University Frankfurt, Germany

Alexander Dobler (38)
Algorithms and Complexity Group, TU Wien, Austria

Riccardo Dondi (12)
Università degli studi di Bergamo, Italy

Ilan Doron-Arad (13)
Computer Science Department, Technion, Haifa, Israel

Pål Grønås Drange (2, 14, 39)
University of Bergen, Norway

Maël Dumas (15)
Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans, France

Julien Duron (8, 40)
Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

David Dursteler (36)
University of Utah, Salt Lake City, UT, USA

Eduard Eiben (16)
Department of Computer Science, Royal Holloway, University of London, Egham, UK

Barış Can Esmer (17)
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany; Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus, Germany

Vinicius F. dos Santos (9)
Department of Computer Science, Federal, University of Minas Gerais, Belo Horizonte, Brazil

Fedor V. Fomin (1)
University of Bergen, Norway

Harmender Gahlawat (10)
Ben-Gurion University of the Negev, Beersheba, Israel

Robert Ganian (16, 18)
Algorithms and Complexity Group, TU Wien, Austria

Jaroslav Garvardt (19, 20)
Philipps-Universität Marburg, Germany; Friedrich-Schiller-Universität Jena, Germany

Tatsuya Gima (21)
JSPS Research Fellow, Nagoya University, Japan

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Petr A. Golovach (1)
University of Bergen, Norway

Patrick Greaves (14)
Birkbeck, University of London, UK

Siddharth Gupta (22, 23)
BITS Pilani, Goa Campus, India

Anselm Haak (37)
Goethe University Frankfurt, Germany

Klaus Heeger (24)
Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Irene Heinrich (25)
Technische Universität Darmstadt, Germany

Danny Hermelin (24)
Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Lars Jaffke (7, 26)
University of Bergen, Norway

Satyabrata Jana (5)
The Institute of Mathematical Sciences, HBNI, Chennai, India

Bart M. P. Jansen (27, 28, 29)
Eindhoven University of Technology, The Netherlands

Mark Jones (30)
TU Delft, The Netherlands

Peter Jonsson (11)
Department of Computer and Information Science, Linköping University, Sweden

Frank Kammer (37)
THM, University of Applied Sciences , Mittelhessen, Gießen, Germany

Iyad Kanj (16)
School of Computing, DePaul University, Chicago, IL, USA

Liana Khazaliya (18)
Technische Universität Wien, Austria

Eun Jung Kim (21)
Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Dušan Knop (7)
Czech Technical University in Prague, Czech Republic

Ioannis Koutis (31)
New Jersey Institute of Technology, NJ, USA

Stefan Kratsch (32)
Algorithm Engineering, Humboldt-Universität zu Berlin, Germany

Ariel Kulik (13, 17)
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Pascal Kunz (32)
Algorithm Engineering, Humboldt-Universität zu Berlin, Germany

Noleen Köhler (21)
Université Paris-Dauphine, PSL University, CNRS UMR7243, LAMSADE, Paris, France

Manuel Lafond (12)
Université de Sherbrooke, Canada

Kenneth Langedal (39)
University of Bergen, Norway

Clément Legrand-Duchesne (9)
LaBRI, CNRS, Université de Bordeaux, France

Alexander Leonhardt (37)
Goethe University Frankfurt, Germany

Paloma T. Lima (7)
IT University of Copenhagen, Denmark

Reem Mahmoud (9)
Virginia Commonwealth University, Richmond, VA, USA

Isja Mannens (6)
Utrecht University, The Netherlands

Mathieu Mari (33)
Institute of Informatics, University of Warsaw, Poland; IDEAS-NCBR, Warsaw, Poland

Dániel Marx (17)
CISPA Helmholtz Center for Information Security, Saarbrücken, Germany

Johannes Meintrup (37)
THM, University of Applied Sciences , Mittelhessen, Gießen, Germany

Nikolaos Melissinos (21)
Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Ulrich Meyer (37)
Goethe University Frankfurt, Germany

Martin Milanič (7)
FAMNIT and IAM, University of Primorska, Koper, Slovenia

Yosuke Mizutani (36)
University of Utah, Salt Lake City, UT, USA

Nils Morawietz (20)
Institute of Computer Science, Friedrich Schiller University Jena, Germany

Laure Morelle (26, 38)
LIRMM, Université de Montpellier, CNRS, France

Amer E. Mouawad (9)
Department of Computer Science, American University of Beirut, Beirut, Lebanon

Irene Muzi (14)
Birkbeck, University of London, UK

Daniel Neuen (17)
University of Bremen, Germany

André Nichterlein (20)
Technische Universität Berlin, Germany

Yoshio Okamoto (9)
Graduate School of Informatics and Engineering, The University of Electro-Communications, Chofu, Japan

Jelle J. Oostveen (6)
Utrecht University, The Netherlands

Sebastian Ordyniak (7, 11, 16)
University of Leeds, UK

George Osipov (11)
Department of Computer and Information Science, Linköping University, Sweden

Sukanya Pandey (6, 7)
Utrecht University, The Netherlands

Manuel Penschuck (37)
Goethe University Frankfurt, Germany

Anthony Perez (15)
Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans, France

Timothé Picavet (33)
ENS de Lyon, France; Aalto University, Finland

Marcin Pilipczuk (11)
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland; IT University Copenhagen, Denmark

Michał Pilipczuk (33)
Institute of Informatics, University of Warsaw, Poland

Simon Raßmann (25)
Technische Universität Darmstadt, Germany

Felix Reidl (14)
Birkbeck, University of London, UK

Amadeus Reinald (38)
LIRMM, CNRS, Université de Montpellier, France

Malte Renken (19)
Technische Universität Berlin, Germany

Mathis Rocton (38)
Algorithms and Complexity Group, TU Wien, Austria

Shivesh K. Roy (28, 29)
Eindhoven University of Technology, The Netherlands

Guy Sa'ar (22, 23)
Ben Gurion University of the Negev, Beersheba, Israel

Ignasi Sau (26)
LIRMM, Université de Montpellier, CNRS, France

Saket Saurabh (5)
The Institute of Mathematical Sciences, HBNI, Chennai, India; University of Bergen, Norway

Jannik Schestag (4, 19, 30)
Faculteit Elektrotechniek, Wiskunde en Informatica, TU Delft, The Netherlands; Fakultät für Mathematik und Informatik, Friedrich-Schiller-Universität Jena, Germany

Dvir Shabtay (24)
Department of Industrial Engineering and Management, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Hadas Shachnai (13)
Computer Science Department, Technion, Haifa, Israel

Roohani Sharma (5, 11, 17)
Max Planck Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany

Kirill Simonov (18)
Hasso Plattner Institute, Universität Potsdam, Germany

Frank Sommer (4)
Fakultät für Mathematik und Informatik,
Friedrich-Schiller-Universität Jena, Germany

Ondřej Suchý (7)
Czech Technical University in Prague, Czech
Republic

Blair D. Sullivan (2, 36)
University of Utah, Salt Lake City, UT, USA

Stefan Szeider (16)
Algorithms and Complexity Group, TU Wien,
Austria

Hisao Tamaki (34)
Meiji University, Kawasaki, Japan

Till Tantau (3)
Universität zu Lübeck, Germany

Dimitrios M. Thilikos (26)
LIRMM, Université de Montpellier, CNRS,
France

Farhad Vadiee (39)
University of Bergen, Norway

Bart van der Steenhoven (27)
Eindhoven University of Technology, The
Netherlands

Erik Jan van Leeuwen (6)
Utrecht University, The Netherlands

Manolis Vasilakis (21)
Université Paris-Dauphine, PSL University,
CNRS UMR7243, LAMSADE, Paris, France

Martin Vatshelle (39)
University of Bergen, Norway

Mathias Weller (19, 20)
Technische Universität Berlin, Germany

Petra Wolf (1, 2, 39)
University of Bergen, Norway

Michal Włodarczyk (10)
University of Warsaw, Poland

Michał Włodarczyk (31)
University of Warsaw, Poland

Meirav Zehavi (10, 22, 23, 31)
Ben-Gurion University of the Negev, Beersheba,
Israel

An FPT Algorithm for Temporal Graph Untangling

Riccardo Dondi
Università degli studi di Bergamo, Italy

Manuel Lafond
Université de Sherbrooke, Canada

Abstract

Several classical combinatorial problems have been considered and analysed on temporal graphs. Recently, a variant of Vertex Cover on temporal graphs, called MinTimelineCover, has been introduced to summarize timeline activities in social networks. The problem asks to cover every temporal edge while minimizing the total span of the vertices (where the span of a vertex is the length of the timestamp interval it must remain active in). While the problem has been shown to be NP-hard even in very restricted cases, its parameterized complexity has not been fully understood. The problem is known to be in FPT under the span parameter only for graphs with two timestamps, but the parameterized complexity for the general case is open. We settle this open problem by giving an FPT algorithm that is based on a combination of iterative compression and a reduction to the Digraph Pair Cut problem, a powerful problem that has received significant attention recently.

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

Keywords and phrases Temporal Graphs, Vertex Cover, Graph Algorithms, Parameterized Complexity

Digital Object Identifier 10.4230/LIPIcs.IPEC.2023.12

Acknowledgements We thank the referees of the paper that help us to improve the presentation.

1 Introduction

Temporal graphs are emerging as one of the main models to describe the dynamics of complex networks. They describe how relations (edges) change in a discrete time domain [12, 11], while the vertex set is not changing. The development of algorithms on temporal graphs has mostly focused on finding paths or walks and on analyzing graph connectivity [12, 20, 21, 7, 22, 8, 3, 17, 1, 5]. However, several classical problems in computer science have been recently extended to temporal graphs and one of the most relevant problems in graph theory and theoretical computer science, Vertex Cover, has been considered in this context [2, 10, 19].

In particular, here we study a variant of Vertex Cover, called Network Untangling, introduced in [19]. Network Untangling has applications in discovering event timelines and summarizing temporal networks. It considers a sequence of temporal interactions between entities (e.g. discussions between users in a social network) and aims to explain the observed interactions with few (and short) *activity intervals* of entities, such that each interaction is covered by at least one of the two entities involved (i.e. at least one of the two entities is active when an interaction between them is observed).

Network Untangling can be seen as a variant of Vertex Cover, where we search for a minimum cover of the interactions, called temporal edges. The size of this temporal vertex cover is based on the definition of *span* of a vertex, that is the length of vertex activity. In particular, the span of a vertex is defined as the difference between the maximum and minimum timestamp where the vertex is active. Hence, if a vertex is active in exactly one

18th International Symposium on Parameterized and Exact Computation (IPEC 2023).
Editors: Neeldhara Misra and Magnus Wahlström; Article No. 12; pp. 12:1–12:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

timestamp, it has a span equal to 0. This models the idea that each vertex is present in the network because we know that they interacted at least once, but that sustained periods of interaction are relatively rare.

Four combinatorial formulations of Network Untangling have been defined in [19], varying the definition of vertex activity (a single interval or $h \geq 2$ intervals) and the objective function (minimization of the sum of vertex spans or minimization of the maximum vertex span). Here we consider the formulation, denoted by MinTimelineCover, where vertex activity is defined as a single interval and the objective function is the minimization of the sum of vertex spans. Hence, given a temporal graph, MinTimelineCover asks for a cover of the temporal edges that has minimum span and such that each vertex is active in one time interval.

We focus on this specific problem, since it is not known to be FPT or not, while the variant of the problem where vertex activity is defined as two intervals is known to be NP-hard when the span is equal to 0 [9]. Hence it is unlikely that this problem variant admits an FPT algorithm for parameter the span. The MinTimelineCover problem is known to be NP-hard also in very restricted cases, when each timestamp contains at most one temporal edge [4], when each vertex has at most two incident temporal edges in each timestamp and the temporal graph is defined over three timestamps [4], and when the temporal graph is defined over two timestamps [9]. MinTimelineCover is also known to be approximable within factor $O(T \log n)$, where n is the number of vertices and T is the number of timestamps of the temporal graph [6]. Note that, since the span of a vertex activity in exactly one timestamp is equal to 0, MinTimelineCover is trivially in P when the temporal graph is defined on a single timestamp, since in this case any solution of the problem has span 0. Furthermore, deciding whether there exists a solution of MinTimelineCover that has span equal to 0 can be decided in polynomial time via a reduction to 2-SAT [19].

MinTimelineCover has been considered also in the parameterized complexity framework. The definition of span leads to a problem where the algorithmic approaches applied to Vertex Cover cannot be easily extended for the parameter span of the solution. Indeed, in Vertex Cover for each edge we are sure that at least one of the endpoints must be included in the solution, thus at least one of the vertices contributes to the cost of the solution. This leads to the textbook FPT algorithm of branching over the endpoints of any edge. For MinTimelineCover, a vertex with span 0 may cover a temporal edge, as the vertex can be active only in the timestamp where the temporal edge is defined. This makes it more challenging to design FPT algorithms when the parameter is the span of the solution. In this case, MinTimelineCover is known to admit a parameterized algorithm only when the input temporal graph is defined over two timestamps [9], with a parameterized reduction to the Almost 2-SAT problem. However, the parameterized complexity of MinTimelineCover for the span parameter on general instances has been left open [9, 4]. The authors of [9] have also analyzed the parameterized complexity of the variants of Network Untangling proposed in [19], considering other parameters in addition to the span of the solution: the number of vertices of the temporal graph, the length of the time domain, and the number of intervals of vertex activity.

Our contributions. We solve the open question on the parameterized complexity of MinTimelineCover by showing that the problem is FPT in parameter k, the span of a solution, even if the number of timestamps is unbounded. Our algorithm takes time $O^*(2^{5k \log k})$, where the O^* notation hides polynomial factors. Our algorithm is divided into two phases, each using a different technique. First, given a temporal graph G, we use a variant of iterative compression, where we start from a solution S of span at most k on a subgraph of G induced by a subset of vertices (taken across all timestamps), and then try to maintain such a solution after adding a new vertex of G to the graph under consideration. This requires us to reorganize which vertices involved in S should be in the solution or not, and in which timestamps. One challenge is that since the number of such timestamps is unbounded, there are too many ways to choose how to include or not include the vertices that are involved in S. We introduce the notion of a *feasible assignment*, which allows us to compute how the vertices in S can be reorganized (see Def. 8 for the formal definition). There are only $2^{O(k \log k)}$ ways of reorganizing the vertices in S. We try each such feasible assignments X, and we must then find a temporal cover of the whole graph G that "agrees" with X.

This leads to the second phase of the algorithm, which decides if such an agreement cover exists through a reduction to a variant of a problem called Digraph Pair Cut. In this problem, we receive a directed graph and forbidden pairs of vertices, and we must delete at most k arcs so that a specified source vertex does not reach both vertices from a forbidden pair. It is known that the problem can be solved in time $O^*(2^k)$. In this work, we need a version where the input specifies a set of deletable and undeletable arcs, which we call Constrained Digraph Pair Cut. The Digraph Pair Cut problem and its variants have played an important role in devising randomized kernels using matroids [16] and, more recently, in establishing a dichotomy in the complexity landscape of constraint satisfaction problems [13, 15]. Here, the problem is useful since it can model the implications of including a vertex in the solution or not and, in a more challenging way, allows implementing the notion of cost using our definition of span. We hope that the techniques developed for this reduction can be useful for other variants of temporal graph cover.

Overview of the algorithm. Our approach is loosely inspired by some ideas from the FPT algorithm for two timestamps, which is a reduction to Almost 2-SAT [9]. In the latter, one is given a set of clauses with at most two variables each and must delete a minimum number of clauses so that those remaining are satisfiable. We do not use Almost 2-SAT directly, but its usage for two timestamps may help understand the origins of our techniques and the relevance of our reduction to Digraph Pair Cut.

The reduction from MinTimelineCover on two timestamps to Almost 2-SAT associates each vertex v_i with a variable $x(v_i)$, which is true when one should include v_i in a temporal cover and false otherwise; each edge $u_i v_i$ is associated with a clause $x(u_i) \vee x(v_i)$ (here, v_i represents the occurrence of vertex v at timestamp $i \in \{1, 2\}$). This corresponds to enforcing the inclusion of u_i or v_i in our vertex cover, and we can include enough copies of this clause to make it undeletable. Since our goal is to minimize the number of base vertices v with both v_1 and v_2 in the cover, we also add a clause $\neg x(v_1) \vee \neg x(v_2)$. Then there is a temporal cover of G of span at most k if and only if one can delete at most k clauses of the latter form to make all remaining clauses satisfiable.

For $T \geq 3$ timestamps, the clauses of the form $x(u_i) \vee x(v_i)$ can still be used to model the vertex cover requirements, but there seems to be no obvious way to model the span of a cover. One would need to devise a set of clauses of size two such that choosing an interval of t vertices in a cover corresponds to deleting $t - 1$ negative clauses. Our idea is to extend current FPT algorithms for Almost 2-SAT to accommodate our cost function. In [18], the authors propose an iterative compression FPT algorithm that starts from a solution that deletes $k + 1$ clauses, and modifies it into a solution with k clauses, if possible. The algorithm relies on several clever, but complicated properties of the dependency graph of the clauses (in which vertices are literals and arcs are implications implied by the clauses). This algorithm seems difficult to adapt to our problem. To our knowledge, the only other FPT algorithm for Almost 2-SAT is that of [16]. The algorithm of [16] employs a parameterized reduction to Digraph Pair Cut. At a high level, the idea is to start from an initial guess of assignment for a well-chosen subset of variables, then to construct the dependency graph of the clauses. A certain chain of implications is enforced by our initial guess, the vertex pairs to separate correspond to contradictory literals, and deleting arcs corresponds to deleting clauses. It turns out that, with some work, we can skip the Almost 2-SAT formulation and reduce MinTimelineCover to (a variant of) Directed Pair Cut directly by borrowing some ideas from this reduction. This is not immediate though. The first challenge is that the aforementioned "well-chosen initial guess" idea cannot be used in our context, and we must develop new tools to enumerate a bounded number of initial guesses from a partial solution (which we call feasible assignment). The second challenge is that our reduction to our variant of Directed Pair Cut needs a specific gadget to enforce our cost scheme, while remaining consistent with the idea of modeling the dependency graph of the SAT instance corresponding to the vertex cover problem at hand.

Some of the proofs are omitted due to page limit.

2 Preliminaries

For an integer n, we denote $[n] = \{1, \dots, n\}$ and for two integers i, j, we denote $[i, j] = \{i, i+1, \dots, j-1, j\}$ (which is the empty st if $i > j$). Temporal graphs are defined over a discrete time domain $\mathcal{T}$, which is a sequence $1, 2 \dots, T$ of timestamps. A temporal graph is also defined over a set of vertices, called *base vertices*, that do not change in the time domain and are defined in all timestamps, and are associated with *vertices*, which are base vertices defined in specific timestamps. We use subscripts to denote the timestamp to which a vertex belongs to, so, for a base vertex v and $t \in [T]$, we use v_t to denote the occurrence of v in timestamp t. A *temporal edge* connects two vertices, associated with distinct base vertices, that belong to the same timestamp.

▶ **Definition 1.** *A temporal graph $G = (V_B, E, \mathcal{T})$ consists of*

1. *A time domain $\mathcal{T} = \{1, 2 \dots, T\}$;*
2. *A set V_B of* base vertices*; V_B has a corresponding set $V(G)$ of* vertices*, which consists of base vertices in specific timestamps, defined as follows:*

$$V(G) = \{v_t : v \in V_B \wedge t \in [T]\}.$$

3. *A set $E = E(G)$ of temporal edges, which satisfies:*

$$E \subseteq \{u_t v_t : u, v \in V_B, t \in [T] \wedge u \neq v\}.$$

For a directed (static) graph H, we denote by (u, v) an arc from vertex u to vertex v (we consider only directed static graphs, not directed temporal graphs).

Given a temporal graph $G = (V_B, E, \mathcal{T})$ and a set of base vertices $B \subseteq V_B$, we define the set $\tau(B)$ of all vertices of B across all times:

$$\tau(B) = \{v_t : v \in B \wedge t \in [T]\}.$$

If $B = \{v\}$, we may write $\tau(v)$ instead of $\tau(\{v\})$.

For a subset $W_B \subseteq V_B$ of base vertices, we denote by $G[W_B]$ the subgraph induced by $\tau(W_B)$, that is, the graph whose vertex set is $\tau(W_B)$ and whose edge set is $\{u_t v_t \in E : u_t, v_t \in \tau(W_B)\}$. We also use the notation $G - W_B = G[V_B \setminus W_B]$. Observe that $G[W_B]$ and $G - W_B$ are temporal graphs over the same time domain as G.

In order to define the problem we are interested in, we need to define the *assignment* of a set of base vertices.

▶ **Definition 2.** *Consider a temporal graph $G = (V_B, E, \mathcal{T})$ and a set $W_B \subseteq V_B$ of base vertices. An* assignment *of W_B is a subset $X \subseteq \tau(W_B)$ such that if $u_p \in X$ and $u_q \in X$, with $p, q \in [T]$, then $u_t \in X$, for each $t \in [p, q]$. For a base vertex $u \in W_B$ such that there exists $t \in [T]$ with $u_t \in X$, we denote by $\delta(u, X)$, $\Delta(u, X)$, respectively, the minimum and maximum timestamp, respectively, such that $u_{\delta(u,X)}, u_{\Delta(u,X)} \in X$. If u_t does not exist, then $\delta(u, X) = \Delta(u, X) = 0$.*

If W_B is clear from the context or not relevant, then we may say that X is an assignment, without specifying W_B. Note that, given an assignment X and a set $\tau(v)$, for some $v \in V_B$, then $X \cap \tau(v) = \{v_t : v_t \in X \wedge v_t \in \tau(v)\}$ contains vertices for v that belong to a contiguous interval of timestamps. Consider a set $I \subseteq [T]$ of timestamps. An assignment X *intersects* I if there exists $v_t \in X$ such that $t \in I$.

Now, we give the definition of *temporal cover*.

▶ **Definition 3.** *Given a temporal graph $G = (V_B, E, \mathcal{T})$ a* temporal cover *of G is an assignment X of V_B such that the following properties hold:*

1. *For each $v \in V_B$ there exists at least one $v_t \in X$, for some $t \in \mathcal{T}$.*
2. *For each $u_t v_t \in E$, with $t \in [T]$, at least one of u_t, v_t is in X.*

For a temporal cover X of G, the *span* of v in X is defined as: $sp(v, X) = \Delta(v, X) - \delta(v, X)$. Note that if a temporal cover X contains, for a base vertex $v \in V_B$, a single vertex v_t, then $sp(v, X) = 0$. The span of X, denoted by $sp(X)$, is then defined as:

$$sp(X) = \sum_{v \in V_B} sp(v, X).$$

The definition of temporal cover requires that for each base vertex at least one of its associated vertices belongs to the cover. This is not strictly necessary, since it might be possible to cover every temporal edge without this condition. However, this condition simplifies some of the definitions and proofs below. Note that if an assignment of a base vertex is not needed to cover temporal edges, we can assign the vertex to some timestamp without increasing the span.

Now, we are able to define MinTimelineCover (an example is presented in Fig. 1).

▶ **Problem 4.** (*MinTimelineCover*)
Input: *A temporal graph $G = (V_B, \mathcal{T}, E)$, an integer k.*
Question: *Does there exist a temporal cover of G of span at most k?*

A temporal cover $S \subseteq V(G)$ of span at most k will sometimes be called a *solution*. Our goal is to determine whether MinTimelineCover is FPT in parameter k.

3 An FPT Algorithm

In this section we present our FPT algorithm, which consists of two parts:

1. The iterative compression technique.
2. A reduction to the Constrained Digraph Pair Cut problem.

Before presenting the main steps of our algorithm, we present the main idea and some definitions. Recall that our parameter, that is the span of a solution of MinTimelineCover, is denoted by k.

Consider a temporal graph G and assume we have a temporal cover S of span at most k of the subgraph $G - \{w\}$, for some base vertex $w \in V_B$. The idea of the iterative compression step is, starting from S, to show how to decide in FPT time whether there

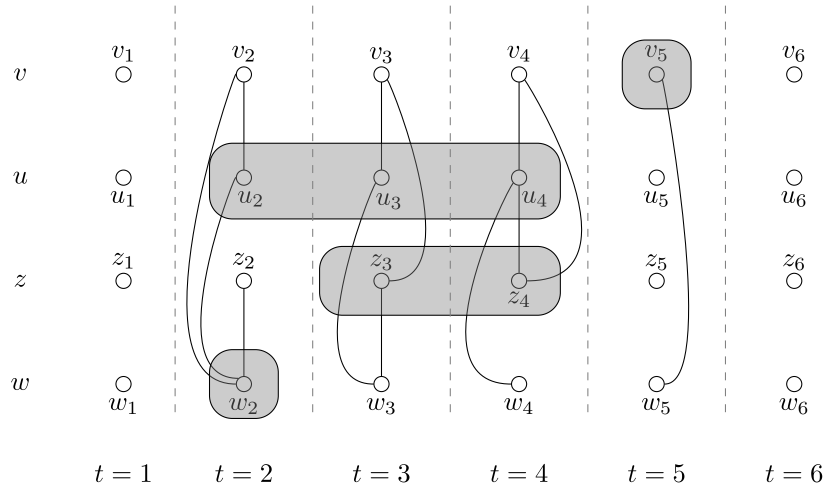

Figure 1 An example of MinTimelineCover on a temporal graph G consisting of four base vertices and six timestamps. For each timestamp, we draw the temporal edges of G, for example for $t = 2$, the temporal edges are v_2u_2, v_2w_2, u_2w_2, z_2w_2. Also note that in $t = 1$ and $t = 6$ no temporal edge is defined. A temporal cover $X = \{v_5, u_2, u_3, u_4, z_3, z_4, w_2\}$ is represented with grey rectangles. Note that $\delta(v, X) = \Delta(v, X) = 5$, $\delta(u, X) = 2$, $\Delta(u, X) = 4$, $\delta(z, X) = 3$, $\Delta(z, X) = 4$, $\delta(w, X) = \Delta(w, X) = 2$. It follows that $sp(X) = 3$.

exists a solution of MinTimelineCover for G. This is done by solving a subproblem, called Restricted Timeline Cover, where we must modify S to consider w. A solution to this subproblem is computed by branching on the assignments of base vertices having a positive span in S and on w, and then reducing the problem to Constrained Digraph Pair Cut. Restricted Timeline Cover is defined as follows.

▶ **Problem 5.** (*Restricted Timeline Cover*)

Input: *A temporal graph $G = (V_B, E, \mathcal{T})$, a vertex $w \in V_B$, an integer k, a temporal cover S of $G - \{w\}$ of span at most k.*

Output: *Does there exist a temporal cover of G of span at most k?*

For technical reasons that will become apparent later, we will assume that the temporal graph contains no edge at timestamps 1 and T, i.e. for every $u_tv_t \in E$, we have $t \in [2, T-1]$ (as in Fig. 1). In particular, this avoids us to consider different gadget definitions in the reduction to Constrained Digraph Pair Cut, as the cases where a base vertex is assigned the first or the last of its associated vertex behaves somehow differently. It is easy to see that if this is not already the case, we can add two such "dummy" timestamps, where G does not contain any temporal edge. Indeed, since there are no temporal edges in these two timestamps, then G has a temporal cover of span at most k if and only if the same graph with dummy timestamps has a temporal cover of span at most k.

Informally, if we are able to solve Restricted Timeline Cover in FPT time, then we can obtain an FPT algorithm for MinTimelineCover as well. Indeed, we can first compute a temporal cover on a small subset of base vertices (for example a single vertex), and then we can add, one at a time, the other vertices of the graph. This requires at most $|V_B|$ iterations, and each time a vertex is added, we compute a solution of Restricted Timeline Cover to check whether it is possible to find a temporal cover of span at most k after the addition of a vertex.

Iterative Compression

We now present our approach based on iterative compression to solve the RESTRICTED TIMELINE COVER problem. Given a solution S for $G - \{w\}$, we focus on the vertices of V_B that have a positive span in S and vertex w. An example of our approach, that illustrates the sets of base vertices and vertices used by the algorithm, is presented in Fig. 2.

Consider the input of RESTRICTED TIMELINE COVER that consists of a temporal graph $G = (V_B, E, \mathcal{T})$, a vertex $w \in V_B$, and a temporal cover S of $G - \{w\}$ of span at most k. Define the following sets associated with S:

$$V_S = \{v \in V_B : \exists p, q \in [T], p < q, \text{ such that } v_p, v_q \in S \} \cup \{w\}$$
$$V'_S = \{v_t : v_t \in S, v \in V_S \setminus \{w\}\} \cup \{w_t : t \in [T]\}.$$

The set V_S is defined as the set of base vertices having span greater than 0 in S, plus the vertex w. V'_S contains the vertices in $V(G)$ associated with V_S, in particular: (1) the vertices corresponding to the base vertices in $V_S \setminus \{w\}$ that are included in S and (2) vertices corresponding to the base vertex w in every timestamp.

Define the following set I_S of timestamps associated with $V_S \setminus \{w\}$:

$$I_S = \{t \ \in [T] : u_t \in V'_S \text{ for some } u \in V_S \setminus \{w\} \ \}.$$

Essentially, I_S contains those timestamps where the base vertices of $V_S \setminus \{w\}$, that is of span greater than zero, have associated vertices in S. These timestamps are essential for computing a solution of RESTRICTED TIMELINE COVER, that is to compute whether there exists a temporal cover of G of span at most k starting from S. We define now the sets of base vertices and vertices associated with S having a span equal to 0:

$$Z_S = V_B \setminus V_S \qquad Z'_S = S \setminus V'_S.$$

First, we show two easy properties of S and I_S on the temporal graph $G - \{w\}$.

▶ **Lemma 6.** *Let S be a solution of MINTIMELINECOVER on instance $G - \{w\}$ and let I_S be the associated set of timestamps. Then $|I_S| \leq 2k$.*

▶ **Lemma 7.** *Let S be a solution of MINTIMELINECOVER on instance $G - \{w\}$. Then, $sp(Z'_S) = 0$. Moreover, Z'_S covers each temporal edge of $G - \{w\}$ not covered by $V'_S \setminus \tau(w)$.*

Now, we introduce the concept of feasible assignment, which is used to "guess" how S is rearranged in a solution of RESTRICTED TIMELINE COVER. Recall that an assignment X intersects a set I_S of timestamps if there exists $v_t \in X$ such that $t \in I_S$.

▶ **Definition 8** (Feasible assignment)**.** *Consider an instance of RESTRICTED TIMELINE COVER that consists of a temporal graph $G = (V_B, \mathcal{T}, E)$, a vertex $w \in V_B$, a temporal cover S of $G - \{w\}$ of span at most k, and sets V_S, V'_S and I_S associated with S. We say that an assignment $X \subseteq \tau(V_S)$ of V_S is a* feasible assignment *(with respect to G, S, and I_S) if all of the following conditions hold:*

1. *the span of X is at most k;*
2. *every edge of $G[V_S]$ is covered by X;*
3. *$X \cap \tau(w)$ is a non-empty assignment of $\{w\}$;*
4. *for every $v \in V_S \setminus \{w\}$, at least one of the following holds: (1) $X \cap \tau(v)$ is empty; (2) $X \cap \tau(v)$ is an assignment of $\{v\}$ that intersects with I_S; or (3) $X \cap \tau(v)$ contains a vertex v_t such that $v_t w_t \in E$ and $w_t \notin X \cap \tau(w)$.*

Given a feasible assignment X, we denote

$$M_S(X) = \{v \in V_S : X \cap \tau(v) \neq \emptyset\} \qquad N_S(X) = \{v \in V_S : X \cap \tau(v) = \emptyset\}$$

Informally, point 4 considers the possible cases for a feasible assignment of the vertices of a base vertex $v \in V_S \setminus \{w\}$: none of the associated vertices in I_S belongs to the computed solution (case 4.(1)); some of its associated vertices in I_S belongs to the solution (case 4.(2)); or some of the v_t vertices are forced, since they belong to an edge $v_t w_t$ with $t \in I_S$, that we know is not covered by w_t (case 4.(3)). Note that these cases are not necessarily mutually exclusive.

Note that $M_S(X)$ and $N_S(X)$ form a partition of V_S. Also note that G, S, and I_S are fixed in the remainder, so we assume that all feasible assignments are with respect to G, S, and I_S without explicit mention. We now relate feasible assignments to temporal covers.

▶ **Definition 9.** *Let X^* be a temporal cover of G and let X be a feasible assignment. We say that X^** agrees *with X if:*
- *for each $v \in M_S(X)$, $X^* \cap \tau(v) = X \cap \tau(v)$;*
- *for each $v \in N_S(X)$ and each $t \in I_S$, X^* contains every neighbor u_t of v_t such that $u_t \in \tau(Z_S)$.*

The intuition of X^* agreeing with X is as follows. For $v \in M_S(X)$, X "knows" which vertices of $\tau(v)$ should be in the solution, and we require X^* to contain exactly those. For $v \in N_S(X)$, we interpret that X does not want any vertex v_t with $t \in I_S$. Thus, to cover the edges incident to v_t that go outside of V_S, we require X^* to contain the other endpoint. Note an important subtlety: we act "as if" X^* should not contain v_t or other vertices of $N_S(X)$ with timestamp in I_S, but the definition does not forbid it. Hence, X^* *can* contain a vertex of $N_S(X)$ in some timestamps of I_S, as long as X^* contains also its neighbors (in I_S) outside V_S.

The main purpose of feasible assignments and agreement is as follows.

▶ **Lemma 10.** *Let X^* be a temporal cover of G of span at most k. Then there exists a feasible assignment X such that X^* agrees with X.*

Proof. Construct $X \subseteq X^*$ as follows: add $X^* \cap \tau(w)$ to X, and for $v \in V_S \setminus \{w\}$, add $X^* \cap \tau(v)$ to X if and only if $X^* \cap \tau(v)$ intersects with the set I_S, or if it contains a vertex v_t incident to an edge $v_t w_t \in E$ such that $w_t \notin X^* \cap \tau(w)$. Note that since X^* is an assignment of V_B, X is an assignment of V_S.

We first focus on arguing that X satisfies each condition of a feasible assignment (Definition 8). For Condition 1, since X^* has span at most k and $X \subseteq X^*$, it is clear that X also has span at most k. For Condition 3, $X^* \cap \tau(w)$ is non-empty by the definition of a temporal cover, and we added $X^* \cap \tau(w)$ to X. For Condition 4, we explicitly require in our construction of X that for each $v \in V_S \setminus \{w\}$, if $X \cap \tau(v)$ is non-empty, then it is equal to $X^* \cap \tau(v)$ and it either intersects with I_S or covers an edge not covered by $X \cap \tau(w) = X^* \cap \tau(w)$.

Let us focus on Condition 2. Let $u_t v_t \in E(G[V_S])$. If $u = w$, then if we did not add w_t to X, X^* must contain v_t and we added $X^* \cap \tau(v)$ to X, thereby covering the edge. The same holds if $v = w$. Assume $u \neq w, v \neq w$, and suppose without loss of generality that X^* contains u_t to cover the edge. Suppose for contradiction that X does not cover $u_t v_t$. Then we did not add $X^* \cap \tau(u)$ to X, which implies that $X^* \cap \tau(u)$ does not intersect with I_S. In particular, $t \notin I_S$. Recall that S, the temporal cover of $G - \{w\}$, only intersects with $\tau(u)$ and $\tau(v)$ in timestamps contained in I_S. Hence, S cannot cover $u_t v_t$, a contradiction. We deduce that X covers every edge. Therefore, X is a feasible assignment.

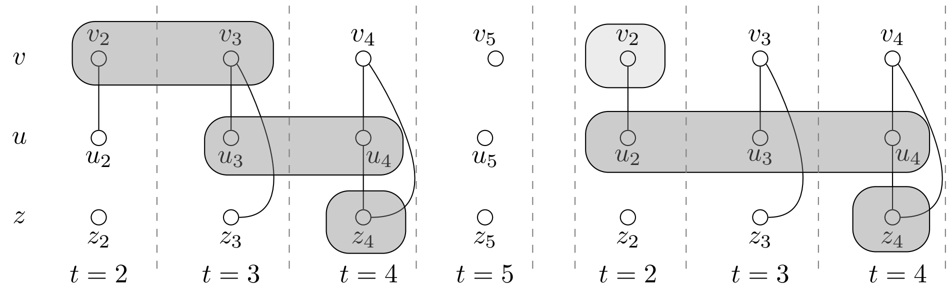

Figure 2 An example of application of iterative compression (timestamps 1 and 6 are not shown as they are edgeless, also vertex w is not shown, its assignment is defined as in Fig. 1). In the left part, we represent solution $S = \{v_2, v_3, u_3, u_4, z_4\}$, where the vertices in S are highlighted with grey rectangles. Note that $I_S = \{2, 3, 4\}$, $V_S = \{v, u\}$, $V'_S = \{v_2, v_3, u_3, u_4\}$, $Z_S = \{z\}$, $Z'_S = \{z_4\}$. In the right part, we represent in grey a feasible assignment X associated with S, containing vertices u_2, u_3, u_4; in light grey we highlight $N'_S = \{v_2\}$. The sets associated with S and X are: $M_S = \{u\}$, $N_S = \{v\}$, $N'_S = \{v_2\}$, $N''_S = \{v_2, v_3, v_4\}$. The reduction to CONSTRAINED DIGRAPH PAIR CUT eventually leads to the solution of MINTIMELINECOVER represented in Fig. 1.

It remains to show that X^* agrees with X. For $v \in M_S(X)$, $X^* \cap \tau(v) = X \cap \tau(v)$ by the construction of X. For $v \in N_S(X)$, there is no $v_t \in X^*$ with $t \in I_S$, as otherwise we would have added $X^* \cap \tau(v)$ to X. For every such v_t, X^* must contain all of its neighbors in $\tau(Z_S)$ to cover the edges, as required by the definition of agreement. ◀

It remains to show that the number of feasible assignments has bounded size and can be enumerated efficiently. We first show the latter can be achieved through the following steps. Start with X as an empty set and then apply the following steps (checking that the overall span is at most k):

(1) Branch into every non-empty assignment X_w of $\{w\}$ of span at most k. In each branch, add the chosen subset X_w to X;
(2) For every edge $v_t w_t \in E(G[V_S])$ such that $w_t \notin X_w$, add v_t to X;
(3) For every $v \in V_S \setminus \{w\}$, such that $X \cap \tau(v) = \emptyset$ at this moment, branch into $|I_S| + 1$ options: either add no vertex of $\tau(v)$ to X, or choose a vertex v_t and add it to X, where $t \in I_S$;
(4) For every $v \in V_S \setminus \{w\}$ such that $X \cap \tau(v) \neq \emptyset$ at this moment, branch into every assignment X_v of $\{v\}$ of span at most k that contains every vertex of $X \cap \tau(v)$ (if no such assignment exists, abort the current branch). For each such branch, add every vertex of $X_v \setminus X$ to X.

▶ **Theorem 11.** *The above steps enumerate every feasible assignment in time $O(2^{4k \log k} T^2 kn)$, where $n = |V_B|$.*

Reducing to Constrained Digraph Pair Cut

Our objective is now to list every feasible assignment and, for each of them, to verify whether there is a temporal cover that agrees with it. More specifically, consider a feasible assignment $X \subseteq \tau(V_S)$. Our goal is to decide whether there is a temporal cover X^* of span at most k that agrees with X. Since we branch over every possible feasible assignment X, if there is a temporal cover X^* of G of span at most k, then by Theorem 11 our enumeration will eventually consider an X that X^* agrees with, and hence we will be able to decide of the existence of X^*.

We show that finding X^* reduces to the Constrained Digraph Pair Cut problem, as we define it below. For a directed graph H, we denote its set of arcs by $A(H)$ (to avoid confusion with $E(G)$, which is used for the edges of an undirected graph G). For $F \subseteq A(H)$, we write $H - F$ for the directed graph with vertex set $V(H)$ and arc set $A(H) \setminus F$.

▶ **Problem 12.** (*Constrained Digraph Pair Cut*)

Input: *A directed graph $H = (V(H), A(H))$, a source vertex $s \in V(H)$, a set of vertex pairs $P \subseteq \binom{V(H)}{2}$ called* forbidden pairs, *a subset of arcs $D \subseteq A(H)$ called* deletable arcs, *and an integer k'.*

Output: *Does there exist a set of arcs $F \subseteq D$ of H such that $|F| \leq k'$ and such that, for each $\{u, v\} \in P$, at least one of u, v is not reachable from s in $H - F$?*

It is known that Constrained Digraph Pair Cut can be solved in time $O^*(2^{k'})$ [16], but a few remarks are needed before proceeding. In [16], the authors only provide an algorithm for the *vertex-deletion* variant, and do not consider deletable/undeletable arcs. It is easy to make an arc undeletable by adding enough parallel paths between the two endpoints, and we show at the end of the section that our formulation of Constrained Digraph Pair Cut reduces to the simple vertex-deletion variant. The vertex-deletion variant also admits a randomized polynomial kernel, and other FPT results are known for weighted arc-deletion variants [14].

So let us fix a feasible assignment X for the remainder of the section. We will denote $M_S = M_S(X)$ and $N_S = N_S(X)$. We also consider the following set of vertices associated with N_S:

$$N'_S = \{v_2 : v \in N_S\} \qquad N''_S = \{v_t \in \tau(N_S) : t \in I_S\}.$$

For each base vertex $v \in N_S$, we need N'_S to contain any vertex of $\tau(v)$ that belongs to the time interval $[2, T-1]$, so we choose v_2 arbitrarily. Then, N''_S contains those vertices v_t, with $t \in I_S$, not chosen by the feasible assignment X. Note that according to our definition of agreement, a solution X^* should contain all the neighbors of N''_S vertices that are in Z_S. Recall that we have defined $Z_S = V_B \setminus V_S$ and $Z'_S = S \setminus V'_S$. By Lemma 7 we know that Z'_S covers each temporal edge of $G[V_B \setminus \{w\}]$ not covered by $S \cap V'_S$, and that $sp(Z'_S) = 0$. We may assume that for each $v \in Z_S$, there is exactly one $t \in [T]$ such that $v_t \in Z'_S$ (there cannot be more than one since Z'_S has span 0, and if there is no such t, we can add any v_t without affecting the span). Furthermore, we will assume that for each $v \in Z_S$, the vertex v_t in Z'_S is not v_1 nor v_T. Indeed, since we assume that the first and last timestamps of G have no edges, if $v_t = v_1$ or $v_t = v_T$, then v_t covers no edge and we may safely change v_p to another vertex of $\tau(v)$.

The following observation will be useful for our reduction to Constrained Digraph Pair Cut.

▶ **Observation 13.** *Let $u_t v_t \in E(G)$ such that $u \in N_S$ and $v \notin M_S$. Then $v \in Z_S$ and, if $u_t \notin N''_S$, we have $v_t \in Z'_S$.*

Now, given a feasible assignment $X \subseteq \tau(V'_S)$, sets M_S, N_S, N'_S, N''_S, Z_S, and Z'_S, we present our reduction to the Constrained Digraph Pair Cut problem. We construct an instance of this problem that consists of the directed graph $H = (V(H), A(H))$, the set of forbidden (unordered) pairs $P \subseteq \binom{V(H)}{2}$, and the deletable arcs $D \subseteq A(H)$ by applying the following steps. The second step in the construction is the most important and is shown in Figure 3. The intuition of these steps is provided afterwards.

1. add to H the source vertex s;
2. for each $v \in Z_S \cup N_S$, let v_i be the vertex of $Z'_S \cup N'_S$, where $i \in [2, T-1]$. Add to H the vertices $v_1^+, \dots, v_{i-1}^+, v_i^-, v_{i+1}^+, \dots, v_T^+$, the vertices $b_{v,j}, c_{v,j}, d_{v,j}$, for $j \in [T] \setminus \{i\}$, and the set of arcs shown in Figure 3, that is there are arcs $(v_j^+, b_{v,j})$, $(v_j^+, c_{v,j})$, $(c_{v,j}, d_{v,j})$, $(d_{v,j}, v_j^-)$, for each $j \in [T] \setminus \{i\}$ and four directed paths: (1) from $b_{v,i-1}$ to $b_{v,1}$, (2) from $c_{v,1}$ to $c_{v,i-1}$, (3) from $b_{v,i+1}$ to $b_{v,T}$ and (4) from $c_{v,T}$ to $c_{v,i+1}$.
   Add to D the set of deletable arcs $(c_{v,j}, d_{v,j})$, for $j \in [T] \setminus \{i\}$.
   Then add the following pairs to P:
   a. $\{d_{v,h}, b_{v,j}\}$, with $1 \le h < j \le i-1$;
   b. $\{d_{v,h}, b_{v,j}\}$, with $i+1 \le j < h \le T$;
   c. $\{c_{v,h}, d_{v,j}\}$, with $1 \le h \le i-1 \le i+1 \le j \le T$;
   d. $\{c_{v,h}, d_{v,j}\}$, with $1 \le j \le i-1 \le i+1 \le h \le T$.
   Note that we have created $T + 3(T-1) = 4T - 3$ vertices in H in this step. The subgraph of H induced by these vertices will be called the *gadget corresponding to* v.
3. for each temporal edge $u_t v_t \in E(G)$ such that $u_t, v_t \in \tau(Z_S) \cup (\tau(N_S) \setminus N''_S)$, there are three cases. First note that at least one of u_t or v_t is in Z'_S. Indeed, if $u, v \in Z_S$, this is because an element of Z'_S must cover the temporal edge, and if $u \in N_S$, then $v_t \in Z'_S$ by Observation 13 (or if $v \in N_S, u_t \in Z'_S$). The subcases are then:
   a. if $u_t, v_t \in Z'_S \cup N'_S$, add the pair $\{u_t^-, v_t^-\}$ to P;
   b. if $u_t \in Z'_S \cup N'_S, v_t \notin Z'_S \cup N'_S$, add the arc (u_t^-, v_t^+) to H;
   c. if $v_t \in Z'_S \cup N'_S, u_t \notin Z'_S \cup N'_S$, add the arc (v_t^-, u_t^+) to H;
4. for each temporal edge $u_t v_t \in E(G)$ such that $u_t \in (\tau(M_S) \setminus X) \cup N''_S$ and $v_t \in \tau(Z_S)$, there are two cases:
   a. if $v_t \notin Z'_S$, add the arc (s, v_t^+) to H;
   b. if $v_t \in Z'_S$, add the pair $\{s, v_t^-\}$ to P.

Define $k' = k - sp(X)$. This concludes the construction. We will refer to the elements 1, 2, 3, 4 of the above enumeration as the *Steps* of the construction. Note that the only deletable arcs in D are the arcs $(c_{v,j}, d_{v,j})$ introduced in Step 2.

From here, the interpretation of H is that if we delete arc set F, then

(p1) For $v_t \notin Z'_S \cup N'_S$ we should include v_t in X^* if and only if s reaches v_t^+ in $H - F$;

(p2) For $v_t \in Z'_S \cup N'_S$ we should include v_t in X^* if and only if s does *not* reach v_t^- in $H - F$.

The idea behind the steps of the construction is then as follows (and is somewhat easier to describe in the reverse order of steps). Step 4 describes an initial set of vertices that s is forced to reach, which correspond to vertices that are forced in X^*. A vertex v_t in $\tau(Z_S)$ is forced in X^* if there is in an edge $u_t v_t$ and $u_t \in \tau(M_S)$ but $u_t \notin X$. By our definition of agreement, v_t is also forced if $u_t \in N''_S$. Step 4 handles both situations: if $v_t \notin Z'_S$, we force s to reach v_t^+ with the arc (s, v_t^+), which is not deletable. If $v_t \in Z'_S$, then $v_t^- \in V(H)$, and s is forced to *not* reach v_t^- by adding $\{s, v_t^-\}$ to P. By (p1) and (p2), both cases correspond to including v_t in X^*. Then, Step 3 ensures that each temporal edge is "covered": for a temporal edge $u_t v_t$, a pair of the form $\{u_t^-, v_t^-\}$ in P requires that s does not reach one of the two, i.e. that we include one in X^*, and an undeletable arc of the form (u_t^-, v_t^+) enforces that if s reaches u_t^- (i.e. $u_t \notin X^*$), then s reaches v_t^+ (i.e. $v_t \in X^*$). The reason why Z'_S is needed in our construction is that each edge has at least one negative corresponding vertex, so that no other case needs to be considered in Step 3.

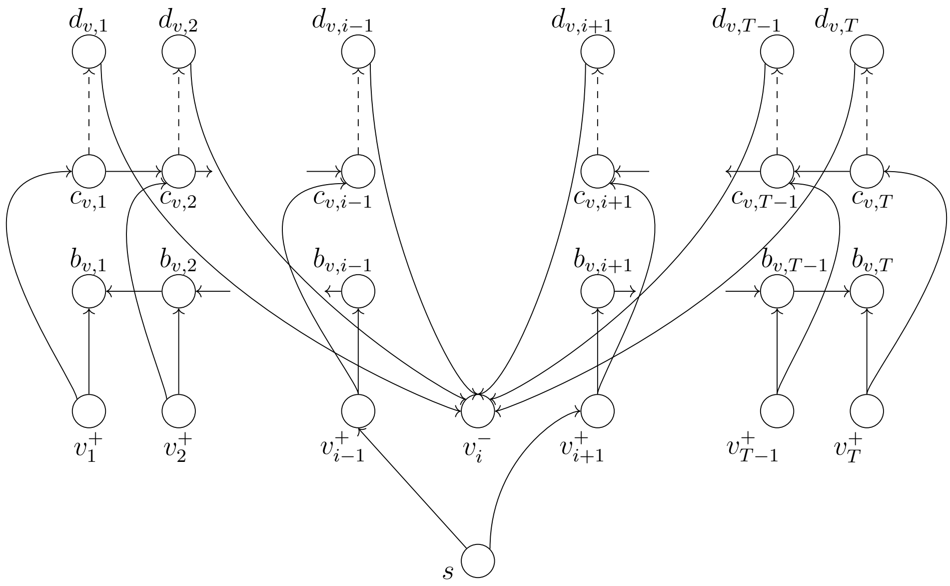

Figure 3 Gadget for $v_i \in Z'_S \cup N'_S$, where $i \in [2, T-1]$. We assume that there exist temporal edges $u_t v_t \in E(G)$, where $t \in \{i-1, i+1\}$, such that $u_t \in (\tau(M_S) \setminus X) \cup N''_S$, $v_t \in \tau(Z_S)$ and $v_t \notin Z'_S$, thus arcs from s to v_t^+ are added.The dashed arcs represent deletable arcs.

Finally, Step 2 enforces the number of deleted arcs to correspond to the span of a solution. That is, it ensures that if we want to add to X^* a set of h vertices of base vertex $v \in Z_S$ to our solution of Restricted Timeline Cover (so with a span equal to $h-1$), then we have to delete $h-1$ deletable arcs of the corresponding gadget of H in order to obtain a solution to Constrained Digraph Pair Cut (and vice-versa). Indeed, consider the gadget in Fig. 3. If v_i is not included in X^*, then in the gadget s reaches h positive vertices $v_l^+, \ldots, v_r^+$ (and v_i^-). It follows that vertices $b_{v,l}, \ldots, b_{v,r}$, $c_{v,l}, \ldots, c_{v,r}$ and $d_{v,l}, \ldots, d_{v,r}$ are all reachable from s. The pairs $\{d_{v,x}, b_{v,y}\}$ defined at Step 2, where either $l \le x \le y \le r-1$ if $r < i$, or $l+1 \le x \le y \le r$ if $l > i$, ensures that arcs $(c_{v,j}, d_{v,j})$, with $j \in [l, r-1]$ in the former case or with $j \in [l+1, r]$ in the latter case, are deleted.

If v_i is included in X^*, then in the gadget s reaches $h-1$ positive vertices $v_l^+, \ldots, v_r^+$, with $i \in [l, r]$, and must not reach negative vertex v_i^-. It follows that vertices $b_{v,l}, \ldots, b_{v,r}$, $c_{v,l}, \ldots, c_{v,r}$ and $d_{v,l}, \ldots, d_{v,r}$ are all reachable from s. Then $h-1$ arcs $(c_{v,j}, d_{v,j})$, with $j \in [l, r] \setminus \{i\}$, must be deleted, due to the pairs $\{d_{v,x}, b_{v,y}\}$, $\{c_{v,x}, d_{v,y}\}$ defined at Step 2.

Note that Step 2 is the reason we added dummy timestamps 1 and T. If v_1 or v_T were allowed to be in $Z'_S \cup N'_S$, we would need a different gadget for these cases.

▶ **Lemma 14.** *There exists a solution of Restricted Timeline Cover that agrees with X if and only if there is $F \subseteq D$ with $|F| \le k'$ such that s does not reach a forbidden pair in $H - F$. Moreover, given such a set F, a solution of Restricted Timeline Cover can be computed in polynomial time.*

Sketch of the proof. ($\Rightarrow$) Suppose that there exists a solution X^* of Restricted Timeline Cover that agrees with X. By definition of Restricted Timeline Cover, X^* has span at most k. Note that for $v \in M_S$, the agreement requires that $X^* \cap \tau(v) = X \cap \tau(v)$, and so the span of v in X^* is the same as the span of v in X. Thus

$$\sum_{v \in Z_S \cup N_S} sp(v, X^*) \le k - sp(X) = k'.$$

We may assume that for every $v \in V_B$, at least one of $v_2, \dots, v_{T-1}$ is in X^*, as otherwise we add one arbitrarily without affecting the span (if only v_1 or v_T is in X^*, remove it first). For each $v \in Z_S \cup N_S$, consider the gadget corresponding to v in H and delete some of its dashed arcs as follows (we recommend referring to Figure 3).

First, if only one of $\tau(v)$ is in X^*, no action is required on the gadget. So assume that $X^* \cap \tau(v)$ has at least two vertices; in the following we denote $v_l = v_{\delta(v,X^*)}$ and $v_r = v_{\Delta(v,X^*)}$ the vertices associated with v having minimum and maximum timestamp, respectively, contained in X^*. We assume that $l, r \in [2, T-1]$ and $l < r$. Note that $X^* \cap \tau(v) = \{v_l, v_{l+1}, \dots, v_r\}$.

Let $v_i \in Z'_S \cup N'_S$, where $i \in [2, T-1]$. Then

- suppose that $l, r \in [2, i-1]$, then: delete every arc $(c_{v,q}, d_{v,q})$, with $l \le q \le r-1$
- suppose that with $l, r \in [i+1, T-1]$, then: delete every arc $(c_{v,q}, d_{v,q})$, with $l+1 \le q \le r$
- suppose that $l \in [2, i]$ and $r \in [i, T-1]$, then: delete every arc $(c_{v,q}, d_{v,q})$, with $l \le q \le i-1$, and delete every arc $(c_{v,q}, d_{v,q})$, with $i+1 \le q \le r$.

We see that by construction for all $v \in Z_S \cup N_S$, the number of arcs deleted in the gadget corresponding to v is equal to the number of vertices in $X^* \cap \tau(v)$ minus one, that is the span of v in X^*. Since these vertices have span at most k', it follows that we deleted at most k' arcs from H. Denote by H' the graph obtained after deleting the aforementioned arcs. We argue that in H', s does not reach a forbidden pair. To this end, we claim the following.

▷ Claim 15. For $v \in Z_S \cup N_S$ and $t \in [T]$, if s reaches v_t^+ in H', then $v_t \in X^*$, and if s reaches v_t^- in H', then $v_t \notin X^*$.

Now, armed with the above claim, we can prove that in H', s does not reach both vertices of a forbidden pair $q \in P$, thus concluding this direction of the proof.

($\Leftarrow$) Suppose that there is a set $F \subseteq D$ with at most k' arcs such that s does not reach a forbidden pair in $H - F$. Denote $H' = H - F$. We construct X^* from F, which will also show that it can be reconstructed from F in polynomial time. Define $X^* \subseteq V(G)$ as follows:

- for each $v \in M_S$, add every element of $X \cap \tau(M_S)$ to X^*;
- for each $v_t \in V(G) \setminus \tau(M_S)$, we add v_t to X^* if and only if one of the following holds: (1) $v_t^+ \in V(H)$ and s reaches v_t^+ in H'; or (2) $v_t^- \in V(H)$, and s does *not* reach v_t^- in H';
- for each $v_j, v_h \in X^*$ with $j < h$, add v_t to X^* for each $t \in [j+1, h-1]$.

Note that X^* agrees with X. Indeed, for $v \in M_S$, there is no gadget corresponding to v in the construction and thus we only add $X \cap \tau(v)$ to X^*. For $u \in N_S$, consider $u_t \in N''_S$ and a neighbor v_t of u_t in $\tau(Z_S)$. If $v_t \notin Z'_S$, Step 4 adds an undeletable arc from s to v_t^+, hence s reaches that vertex and we put v_t in X^*. If $v_t \in Z'_S$, Step 4 adds $\{s, v_t^-\}$ to P, and thus s does not reach v_t^- in H', and again we add v_t to X^*. Therefore, we add all the $\tau(Z_S)$ neighbors of u_t to X^*, and so it agrees with X. We can prove that X^* covers every temporal edge of G and that $sp(X^*) \le k$. ◀

Wrapping up

Before concluding, we must show that we are able to use the results of [16] to get an FPT algorithm for Constrained Digraph Pair Cut, as we have presented it. As we mentioned, the FPT algorithm in [16] studied the vertex-deletion variant and does not consider undeletable elements, but this is mostly a technicality. Roughly speaking, in our variant, it suffices to replace each vertex with enough copies of the same vertex, and replace each deletable arc (u, v) with a new vertex, adding arcs from the u copies to that vertex, and arcs from that vertex to the v copies. Deleting (u, v) corresponds to deleting that new vertex. For undeletable arcs, we apply the same process but repeat it $k' + 1$ times.

▶ **Lemma 16.** *The* Constrained Digraph Pair Cut *problem can be solved in time $O^*(2^k)$, where k is the number of arcs to delete.*

We are able now to prove the main result of our contribution.

▶ **Theorem 17.** *MinTimelineCover on a temporal graph $G = (V_B, E, \mathcal{T})$ can be solved in time $O^*(2^{5k \log k})$.*

Proof. First, we discuss the correctness of the algorithm we presented. Assume that we have an ordering on the base vertices of G and that v is the first vertex of this ordering. A solution S of MinTimelineCover on $G[\{v\}]$ is equal to $S = \emptyset$.

Then for i, with $i \in [2, |V_B|]$, let G_i be the temporal graph induced by the first i vertices and let w be the $i+1$-th vertex. Given a solution S of MinTimelineCover on instance G_i of span at most k, we can decide whether there exists a solution of MinTimelineCover on instance G_{i+1} by computing whether there exists a solution X^* of the Restricted Timeline Cover problem on instance G_i, w, S. By Lemma 10 and by Theorem 11 if there exists such an X^*, then there exists a feasible assignment X such that X^* agrees with X. By Lemma 14 we can compute, via the reduction to Constrained Digraph Pair Cut, whether there exists a solution of Restricted Timeline Cover on instance on instance G_i, w, S, and if so obtain such a solution (if no such solution X^* exists, then Lemma 14 also says that we will never return a solution, since every feasible assignment X that we enumerate will lead to a negative instance of Constrained Digraph Pair Cut). Thus the Restricted Timeline Cover subproblem is solved correctly, and once it is solved on $G_{|V_B|}$, we have a solution to MinTimelineCover.

Now, we discuss the complexity of the algorithm. We must solve Restricted Timeline Cover $|V_B|$ times. For each iteration, by Theorem 11 we can enumerate the feasible assignments in $O(2^{4k \log k} T^3 n)$ time. For each such assignment, the reduction from Restricted Timeline Cover to Constrained Digraph Pair Cut requires polynomial time, and each generated instance can be solved in time $O^*(2^k)$. The time dependency on k is thus $O^*(2^{4k \log k} \cdot 2^k)$, which we simplify to $O^*(2^{5k \log k})$. ◀

4 Conclusion

We have presented an FPT algorithm for the MinTimelineCover problem, a variant of Vertex Cover on temporal graphs recently considered for timeline activities summarizations. We point out some relevant future directions on this topic: (1) to improve, if possible, the time complexity of MinTimelineCover by obtaining a single exponential time algorithm (of the form $O^*(c^k)$); (2) to establish whether MinTimelineCover admits a polynomial kernel, possibly randomized (which it might, since Constrained Digraph Pair Cut famously admits a randomized polynomial kernel).

References

1 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Christoforos L. Raptopoulos. The temporal explorer who returns to the base. *J. Comput. Syst. Sci.*, 120:179–193, 2021. doi:10.1016/j.jcss.2021.04.001.

2 Eleni C. Akrida, George B. Mertzios, Paul G. Spirakis, and Viktor Zamaraev. Temporal vertex cover with a sliding time window. *J. Comput. Syst. Sci.*, 107:108–123, 2020. doi:10.1016/j.jcss.2019.08.002.

3 Benjamin Merlin Bumpus and Kitty Meeks. Edge exploration of temporal graphs. In Paola Flocchini and Lucia Moura, editors, *Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021, Ottawa, ON, Canada, July 5-7, 2021, Proceedings*, volume 12757 of *Lecture Notes in Computer Science*, pages 107–121. Springer, 2021. doi:10.1007/978-3-030-79987-8_8.

4 Riccardo Dondi. Untangling temporal graphs of bounded degree. *Theor. Comput. Sci.*, 969:114040, 2023. doi:10.1016/j.tcs.2023.114040.

5 Riccardo Dondi and Mohammad Mehdi Hosseinzadeh. Finding colorful paths in temporal graphs. In Rosa María Benito, Chantal Cherifi, Hocine Cherifi, Esteban Moro, Luis M. Rocha, and Marta Sales-Pardo, editors, *Complex Networks & Their Applications X - Volume 1, Proceedings of the Tenth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2021, Madrid, Spain, November 30 - December 2, 2021*, volume 1015 of *Studies in Computational Intelligence*, pages 553–565. Springer, 2021. doi:10.1007/978-3-030-93409-5_46.

6 Riccardo Dondi and Alexandru Popa. Timeline cover in temporal graphs: Exact and approximation algorithms. In Sun-Yuan Hsieh, Ling-Ju Hung, and Chia-Wei Lee, editors, *Combinatorial Algorithms - 34th International Workshop, IWOCA 2023, Tainan, Taiwan, June 7-10, 2023, Proceedings*, volume 13889 of *Lecture Notes in Computer Science*, pages 173–184. Springer, 2023. doi:10.1007/978-3-031-34347-6_15.

7 Thomas Erlebach, Michael Hoffmann, and Frank Kammer. On temporal graph exploration. *J. Comput. Syst. Sci.*, 119:1–18, 2021. doi:10.1016/j.jcss.2021.01.005.

8 Till Fluschnik, Hendrik Molter, Rolf Niedermeier, Malte Renken, and Philipp Zschoche. Temporal graph classes: A view through temporal separators. *Theor. Comput. Sci.*, 806:197–218, 2020. doi:10.1016/j.tcs.2019.03.031.

9 Vincent Froese, Pascal Kunz, and Philipp Zschoche. Disentangling the computational complexity of network untangling. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 2037–2043. ijcai.org, 2022. doi:10.24963/ijcai.2022/283.

10 Thekla Hamm, Nina Klobas, George B. Mertzios, and Paul G. Spirakis. The complexity of temporal vertex cover in small-degree graphs. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10193–10201. AAAI Press, 2022.

11 Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):234, 2015.

12 David Kempe, Jon M. Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. *J. Comput. Syst. Sci.*, 64(4):820–842, 2002. doi:10.1006/jcss.2002.1829.

13 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Directed flow-augmentation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 938–947, 2022.

14 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Directed flow-augmentation. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 938–947. ACM, 2022. doi:10.1145/3519935.3520018.

15 Eun Jung Kim, Stefan Kratsch, Marcin Pilipczuk, and Magnus Wahlström. Flow-augmentation iii: Complexity dichotomy for boolean csps parameterized by the number of unsatisfied constraints. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3218–3228. SIAM, 2023.

16 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. *J. ACM*, 67(3):16:1–16:50, 2020. doi:10.1145/3390887.

17 Andrea Marino and Ana Silva. Königsberg sightseeing: Eulerian walks in temporal graphs. In Paola Flocchini and Lucia Moura, editors, *Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021, Ottawa, ON, Canada, July 5-7, 2021, Proceedings*, volume 12757 of *Lecture Notes in Computer Science*, pages 485–500. Springer, 2021. doi:10.1007/978-3-030-79987-8_34.

18 Igor Razgon and Barry O'Sullivan. Almost 2-sat is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009. doi:10.1016/j.jcss.2009.04.002.

19 Polina Rozenshtein, Nikolaj Tatti, and Aristides Gionis. The network-untangling problem: from interactions to activity timelines. *Data Min. Knowl. Discov.*, 35(1):213–247, 2021. doi:10.1007/s10618-020-00717-5.

20 Huanhuan Wu, James Cheng, Silu Huang, Yiping Ke, Yi Lu, and Yanyan Xu. Path problems in temporal graphs. *Proc. VLDB Endow.*, 7(9):721–732, 2014. doi:10.14778/2732939.2732945.

21 Huanhuan Wu, James Cheng, Yiping Ke, Silu Huang, Yuzhen Huang, and Hejun Wu. Efficient algorithms for temporal path computation. *IEEE Trans. Knowl. Data Eng.*, 28(11):2927–2942, 2016. doi:10.1109/TKDE.2016.2594065.

22 Philipp Zschoche, Till Fluschnik, Hendrik Molter, and Rolf Niedermeier. The complexity of finding small separators in temporal graphs. *J. Comput. Syst. Sci.*, 107:72–92, 2020. doi:10.1016/j.jcss.2019.07.006.