

4th Symposium on Algorithmic Foundations of Dynamic Networks

SAND 2025, June 9–11, 2025, Liverpool, GB

Edited by

Kitty Meeks

Christian Scheideler



Editors

Kitty Meeks 

University of Glasgow, UK
Kitty.Meeks@glasgow.ac.uk

Christian Scheideler 

Paderborn University, Germany
scheideler@upb.de

ACM Classification 2012

Theory of computation; Mathematics of computing; Networks

ISBN 978-3-95977-368-3

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/978-3-95977-368-3>.

Publication date

June, 2025

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>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.SAND.2025.0

ISBN 978-3-95977-368-3

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.

Editorial Board

- Luca Aceto (Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Roberto Di Cosmo (Inria and Université Paris Cité, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Holger Hermanns (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)
- Daniel Král' (Leipzig University, DE and Max Planck Institute for Mathematics in the Sciences, Leipzig, DE)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN – Chair)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (Nanyang Technological University, SG)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Pierre Senellart (ENS, Université PSL, Paris, France)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Kitty Meeks and Christian Scheideler</i>	0:vii–0:viii

Regular Papers

The Expressive Power of Uniform Population Protocols with Logarithmic Space	
<i>Philipp Czerner, Vincent Fischer, and Roland Guttenberg</i>	1:1–1:18
Dynamic Debt Swapping in Financial Networks	
<i>Henri Froese, Martin Hoefer, and Lisa Wilhelm</i>	2:1–2:16
Temporal Connectivity Augmentation	
<i>Thomas Bellitto, Jules Bouton Popper, and Bruno Escoffier</i>	3:1–3:16
On b-Matching and Fully-Dynamic Maximum k-Edge Coloring	
<i>Antoine El-Hayek, Kathrin Hanauer, and Monika Henzinger</i>	4:1–4:23
Undecidability of the Emptiness Problem for Weak Models of Distributed Computing	
<i>Flavio T. Principato, Javier Esparza, and Philipp Czerner</i>	5:1–5:14
Dismountability in Temporal Cliques Revisited	
<i>Daniele Carnevale, Arnaud Casteigts, and Timothée Corsini</i>	6:1–6:18
Better Late, Then? The Hardness of Choosing Delays to Meet Passenger Demands in Temporal Graphs	
<i>David C. Kutner and Anouk Sommer</i>	7:1–7:18
Matching and Edge Cover in Temporal Graphs	
<i>Lapo Cioni, Riccardo Dondi, Andrea Marino, Jason Schoeters, and Ana Silva</i>	8:1–8:16
Spanner Enumeration for Temporal Graphs	
<i>Kazuhiro Kurita, Andrea Marino, Jason Schoeters, and Takeaki Uno</i>	9:1–9:21
Fault Detection and Identification by Autonomous Mobile Robots	
<i>Stefano Clemente and Caterina Feletti</i>	10:1–10:20
Hardness of Traversing Gadget Systems with Small Bandwidth	
<i>MIT Gadgets Group, Erik D. Demaine, Jenny Diomidova, Timothy Gomez, Markus Hecher, and Jayson Lynch</i>	11:1–11:16
Restless Exploration and Token Dissemination in Vertex-Permuted Temporal Graphs	
<i>Kamran Ayoubi and Lata Narayanan</i>	12:1–12:17
Self-Stabilizing Weakly Byzantine Perpetual Gathering of Mobile Agents	
<i>Jion Hirose, Ryota Eguchi, and Yuichi Sudo</i>	13:1–13:18
Fractals in Seeded Tile Automata	
<i>Asher Hawn, Ryan Knobel, Adrian Salinas, Ramiro Santos, Robert Schweller, and Tim Wylie</i>	14:1–14:17

4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2025).

Editors: Kitty Meeks and Christian Scheideler



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

On the Runtime of Local Mutual Exclusion for Anonymous Dynamic Networks <i>Anya Chaturvedi, Joshua J. Daymude, and Andréa W. Richa</i>	15:1–15:16
Temporal Dominating Set and Temporal Vertex Cover Under the Lense of Degree Restrictions <i>Anton Herrmann, Christian Komusiewicz, Nils Morawietz, and Frank Sommer</i> ...	16:1–16:18
Brief Announcements	
Brief Announcement: Broadcast via Mobile Agents in a Dynamic Network: Interplay of Graph Properties & Agents <i>William K. Moses Jr., Amanda Redlich, and Frederick Stock</i>	17:1–17:5
Brief Announcement: The Shortest Temporal Exploration Problem <i>Stefan Balev, Éric Sanlaville, and Antoine Toullalan</i>	18:1–18:5
Brief Announcement: Anonymous Distributed Localisation via Spatial Population Protocols <i>Leszek Gąsieniec, Lukasz Kuszner, Ehsan Latif, Ramvijas Parasuraman, Paul Spirakis, and Grzegorz Stachowiak</i>	19:1–19:5
Brief Announcement: Efficient Distributed Algorithms for Shape Reduction via Reconfigurable Circuits <i>Nada Almalki, Siddharth Gupta, Othon Michail, and Andreas Padalkin</i>	20:1–20:6
Brief Announcement: Directed Temporal Tree Realization for Periodic Public Transport: Easy and Hard Cases <i>Julia Meusel, Matthias Müller-Hannemann, and Klaus Reinhardt</i>	21:1–21:5
Brief Announcement: Exploring Word-Representable Temporal Graphs <i>Duncan Adamson</i>	22:1–22:6
Brief Announcement: Reachability in Deletion-Only Chemical Reaction Networks <i>Bin Fu, Timothy Gomez, Ryan Knobel, Austin Luchsinger, Aiden Massie, Marco Rodriguez, Adrian Salinas, Robert Schweller, and Tim Wylie</i>	23:1–23:6
Brief Announcement: Intrinsic Universality in Seeded Active Tile Self-Assembly <i>Tim Gomez, Elise Grizzell, Asher Haun, Ryan Knobel, Tom Peters, Robert Schweller, and Tim Wylie</i>	24:1–24:6

■ Preface

This volume contains the papers that were presented at the 4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND), held in Liverpool, UK, June 9–11, 2025. Running annually since 2022, SAND is a primary venue for original research on fundamental aspects of computing in dynamic networks and computational dynamics, bringing together researchers from computer science and related areas. SAND seeks important contributions from all viewpoints, including theory and practice, characterized by a marked algorithmic aspect and addressing or being motivated by the role of dynamics in computing. It welcomes both conceptual and technical contributions, as well as novel ideas and new problems that will inspire the community and facilitate the growth of the area. The program committee of SAND 2025 consisted of:

- Kitty Meeks, University of Glasgow, UK (chair)
- Christian Scheideler, Paderborn University, Germany (chair)
- Eleni Akrida, Durham University, UK
- John Augustine, IIT Madras, India
- Petra Berenbrink, University of Hamburg, Germany
- Quentin Bramas, University of Strasbourg, France
- Arnaud Casteigts, University of Geneva, Switzerland
- Bogdan Chlebus, Augusta University, USA
- Andrea Clementi, University of Rome Tor Vergata, Italy
- Joshua Daymude, Arizona State University, USA
- David Doty, University of California, Davis, USA
- Yuval Emek, Technion, Israel
- Jessica Enright, University of Glasgow, UK
- Sándor Fekete, University of Braunschweig, Germany
- Laurent Feuilloley, CNRS and University of Lyon, France
- Paola Flocchini, University of Ottawa, Canada
- Seth Gilbert, University of Singapore, Singapore
- Olga Goussevskaia, Federal University of Minas Gerais, Brazil
- Taisuke Izumi, Osaka University, Japan
- Amos Korman, University of Haifa, Israel
- Frederik Mallmann-Trenn, King’s College London, UK
- Othon Michail, University of Liverpool, UK
- Yoann Pigné, University of Le Havre, France
- Maria Potop-Butucaru, Sorbonne University, France
- Giuseppe Prencipe, University of Pisa, Italy
- Joel Rybicki, Humboldt University of Berlin, Germany
- Jared Saia, University of New Mexico, USA
- Robert Schweller, University of Texas Rio Grande Valley, USA
- Ana Silva, Federal University of Ceará, Brazil, and University of Florence, Italy
- George Skretas, Hasso Plattner Institute, Germany
- Paul Spirakis, University of Liverpool, UK
- Sébastien Tixeuil, Sorbonne University, France
- Robin Vacus, Bocconi University of Milan, Italy
- Tiphaine Viard, Telecom Paris, France

4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2025).
Editors: Kitty Meeks and Christian Scheideler



Leibniz International Proceedings in Informatics
LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We are also very grateful to the non-PC-member reviewers who helped us evaluating some of the submissions. Namely, Duncan Adamson, Hind Almahmoud, Kristijan Atanasov, Divya Bajaj, Stefan Balev, Thomas Bellitto, Michelle Döring, Yuval Gil, Valentin Gledel, Timothy Gomez, Thorsten Götte, Kiya Hironori, Naoki Kitamura, Ramin Kosfeld, David Kutner, Stefano Leucci, Raul Lopes, Hendrik Molter, Nils Morawietz, Théo Morel, Francesco Piselli, Malin Rau, Eric Sanlaville, Luca Pepè Sciarria, Jason Schoeters, Olivier Stietel, Jukka Suomela, Georg Tennigkeit, and Sébastien Zeitoun.

SAND 2024 received 34 submissions. The review process was double-blind and each paper was assigned to at least three members of the program committee with relevant expertise and eventually reviewed by them and/or by additional reviewers whenever needed. The program committee accepted 16 papers as regular papers, and 8 as brief announcements. These papers cover a wide range of topics, including dynamic networks and distributed algorithms, mobile computing and robotics, programmable matter, and temporal and dynamic graph algorithms. Keynote talks were given by distinguished researchers, to whom we are grateful: Eric Demaine (MIT), Jessica Enright (University of Glasgow), and Bernhard Haeupler (Sofia University and ETH Zürich).

This year, the Best Paper Award was given to Philipp Czerner, Vincent Fischer and Roland Guttenberg for their paper “The Expressive Power of Uniform Population Protocols with Logarithmic Space”, and the Best Student Paper Award was given to Flavio Principato, Javier Esparza and Philipp Czerner for their paper “Undecidability of the Emptiness Problem for Weak Models of Distributed Computing”.

We wish to thank the members of the various committees of SAND as well as its advisory board, for all their hard work in establishing this relatively new conference. All have been supportive throughout. We are grateful to the program committee members and to the additional reviewers for devoting time and effort in order to come up with a strong conference program. A special thanks goes to the chairs of the organizing committee: Leszek Gasieniec, Othon Michail, Igor Potapov, Paul Spirakis, Prudence Wong, and Viktor Zamaraev. We are also indebted to the chair of the SAND steering committee, Othon Michail, for all of his support. Above all, we thank the authors for submitting their work to SAND 2025 and providing substantial contributions to our knowledge on the role of dynamics in computing. We do believe that this volume will inspire further work and will contribute to the further growth of this exciting research area.

June 2025

Kitty Meeks and Christian Scheideler

Matching and Edge Cover in Temporal Graphs

Lapo Cioni  

Università degli Studi di Firenze, Italy

Riccardo Dondi  

Università degli Studi di Bergamo, Italy

Andrea Marino  

Università degli Studi di Firenze, Italy

Jason Schoeters  

Università degli Studi di Firenze, Italy

Ana Silva  

Universidade Federal do Ceará, Fortaleza, Brasil

Abstract

Temporal graphs are a special class of graphs for which a temporal component is added to edges, that is, each edge possesses a set of times at which it is available and can be traversed. Many classical problems on graphs can be translated to temporal graphs, and the results may differ.

In this paper, we define the TEMPORAL EDGE COVER and TEMPORAL MATCHING problems and show that they are NP-complete even when fixing the lifetime or when the underlying graph is a tree. We then describe two FPT algorithms, with parameters lifetime and treewidth, that solve the two problems. We also find lower bounds for the approximation of the two problems and give two approximation algorithms which match these bounds. Finally, we discuss the differences between the problems in the temporal and the static framework.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Mathematics of computing → Matchings and factors; Mathematics of computing → Approximation algorithms; Theory of computation → Problems, reductions and completeness; Theory of computation → Fixed parameter tractability

Keywords and phrases graphs, temporal graphs, edge cover, matching, parameterized algorithm, approximation algorithm

Digital Object Identifier 10.4230/LIPIcs.SAND.2025.8

Related Version *arxiv preprint*: <https://arxiv.org/abs/2504.06762> [4]

Funding This paper was partially funded by Italian PNRR CN4 Centro Nazionale per la Mobilità Sostenibile, NextGeneration EU - CUP, B13C22001000001. MUR of Italy, under PRIN Project n. 2022ME9Z78 – NextGRAAL: Next-generation algorithms for constrained GRAPh visuALization, PRIN PNRR Project n. P2022NZPJA – DLT-FRUIT: A user centered framework for facilitating DLTs FRUITion.

1 Introduction

A temporal graph is a graph where the edges are available only at prescribed moments. More formally, a *temporal graph* with *lifetime* τ is a pair $\mathcal{G} = (G, \lambda)$ where G is a graph (called the *underlying graph*) and λ is the *time labelling* that assigns to each edge a finite non-empty subset of $[\tau]$. Alternatively, a temporal graph can be seen as a finite sequence of spanning subgraphs of G called *snapshots*. A *temporal vertex* is an occurrence of a vertex in time, i.e. an element of $V(G) \times [\tau]$, and a *temporal edge* is an occurrence of an edge in time, i.e. (e, t) with $e \in E(G)$ and $t \in \lambda(e)$. They appear in the literature under many distinct names (temporal networks [8], edge-scheduled networks [2], dynamic networks [16], time-varying



© Lapo Cioni, Riccardo Dondi, Andrea Marino, Jason Schoeters, and Ana Silva; licensed under Creative Commons License CC-BY 4.0

4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2025).

Editors: Kitty Meeks and Christian Scheideler; Article No. 8; pp. 8:1–8:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

8:2 Matching and Edge Cover in Temporal Graphs

■ **Table 1** Temporal variations of edge cover.

covered by covered	TEMPORAL EDGE	EDGE
TEMPORAL VERTEX	polynomial	NP-complete (Theorem. 3)
VERTEX	polynomial	polynomial

■ **Table 2** Temporal variations of matching.

taking not sharing	TEMPORAL EDGE	EDGE
TEMPORAL VERTEX	polynomial	NP-complete (Theorem 4)
VERTEX	polynomial	polynomial

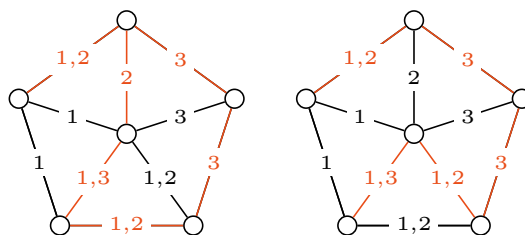
graphs [3], stream graphs, link streams [11], etc). We refer the reader to [8] for a plethora of applications. In the recent years, many papers have focused on studying how well-known problems in static graph theory translate into the temporal setting. In this paper we focus on edge covering and matching problems.

A *matching*¹ is a set of edges such that no two edges share a common vertex. An *edge cover* is a set of edges ensuring that every vertex in the graph is incident to at least one edge in the set. The *maximum matching problem* seeks to find a matching of the largest possible size, while the *minimum edge cover problem* aims to determine the smallest edge cover². These are fundamental problems in graph theory, known to be dual to each other and solvable in polynomial time. To illustrate their duality, consider a maximum matching M in a graph G . A minimum edge cover S of size $|V(G)| - |M|$ can be obtained from M by greedily adding edges until all vertices in G are covered. Applying similar combinatorial reasoning, one can obtain a maximum matching from a minimum edge cover, bringing us to the equality $\alpha'(G) + \beta'(G) = |V(G)|$ [7], where $\alpha'(G)$ is the size of a maximum matching and $\beta'(G)$ is the size of a minimum edge cover. This is known as Gallai's Theorem.

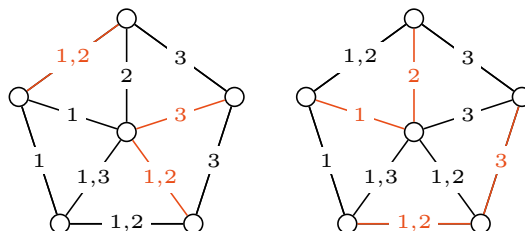
The above concepts naturally extend to temporal graphs in multiple ways, depending on whether we aim to cover or saturate vertices versus temporal vertices, and whether we achieve this using edges or temporal edges. This distinction gives rise to four possible variations, as summarized in Tables 1 and 2. It is straightforward to show that most of these variations reduce to solving the corresponding minimum edge cover or maximum matching problem in static graphs. Indeed, whenever vertices are considered, the temporal component of the edges does not play a role in the problems, and the solutions are the same as those of the corresponding static problems on the underlying graph. On the other hand, if both temporal edges and temporal vertices are considered, then the snapshots of the temporal graph are independent and can be solved as they were static graphs (the resulting graph is called static expansion of a temporal graph [13]). For this reason, we focus on the cases highlighted in pink. In the following, we formally define the relevant concepts. We say that a temporal vertex (v, t) is *isolated* if $t \notin \lambda(uv)$ for every $u \in N(v)$ (in other words, if v is isolated in snapshot G_t).

¹ The definitions for matching and edge cover, as well as their relationship, can be found in most graph theory books. We refer to [15].

² We assume that the graph G has no isolated vertices.



■ **Figure 1** Two minimal temporal edge covers of a temporal graph. The one on the right has minimum cardinality.



■ **Figure 2** Two maximal temporal matchings of a temporal graph. The one on the right has maximum cardinality.

► **Definition 1** (Temporal Edge Cover). *Given a temporal graph $\mathcal{G} = (G, \lambda)$, a temporal edge cover of \mathcal{G} is a subset $S \subseteq E(G)$ such that, for every non-isolated $(v, t) \in V(G) \times [\tau]$, there exists an edge $e \in S$ incident to v such that $t \in \lambda(e)$.*

Examples of temporal edge cover are shown in Figure 1. Observe that the temporal edge covers presented are minimal, with the one on the right having the smallest cardinality among all edge covers of that temporal graph.

► **Definition 2** (Temporal Matching). *Given a temporal graph $\mathcal{G} = (G, \lambda)$, a subset $M \subseteq E(G)$ is a temporal matching of \mathcal{G} if for every $e, e' \in M$, $e \neq e'$, either $e \cap e' = \emptyset$ or $\lambda(e) \cap \lambda(e') = \emptyset$.*

Examples of temporal matching are shown in Figure 2. Observe that the temporal matchings are maximal, with the one on the right having maximum cardinality among all temporal matchings.

We call TEMPORAL EDGE COVER (resp. TEMPORAL MATCHING) the problem of, given a temporal graph \mathcal{G} and a nonnegative integer k , deciding whether there exists a temporal edge cover (resp. temporal matching) of \mathcal{G} of size at most (resp. at least) k .

Our Contributions. Our results are summarized in Theorems 3 and 4. We prove that both problems are NP-complete, even when $\tau = 2$ or when the underlying graph is a tree. This implies that both problems are para-NP-complete when parameterized by either the lifetime or the treewidth of the underlying graph. We then show that combining these parameters allows us to obtain FPT algorithms. It is worth noting that the apparent similarity between the two problems is not due to shared proof techniques; rather, all proofs are independent. Finally, the problems differ in terms of approximation: while TEMPORAL EDGE COVER can be approximated within a logarithmic factor, TEMPORAL MATCHING cannot. In particular, note that our approximation factors are asymptotically optimal.

► **Theorem 3.** TEMPORAL EDGE COVER

1. is NP-complete even if $\tau = 2$;
2. is NP-complete even if the underlying graph is a tree;

3. is FPT parameterized by τ plus the treewidth of the underlying graph;
4. cannot be approximated within factor $b \log \tau$ for any b with $0 < b < 1$, unless $P=NP$;
5. can be approximated within factor $O(\log \tau)$.

► **Theorem 4.** TEMPORAL MATCHING

1. is NP-complete even if $\tau = 2$;
2. is NP-complete even if the underlying graph is a tree;
3. is FPT parameterized by τ plus the treewidth of the underlying graph;
4. cannot be approximated within factor $\tau^{1-\varepsilon}$, for any $\varepsilon > 0$.
5. can be approximated within factor τ .

As previously noted, despite the apparent similarity, the proofs of Theorems 3 and 4 are fundamentally different. This independence arises from the fact that the size of a minimum temporal edge cover is unrelated to the size of a maximum temporal matching, unlike the case of static graphs. In fact, in Section 7 we prove a stronger result, namely that having a minimum temporal edge cover does not facilitate the computation of a maximum temporal matching, and vice versa. More specifically, we show that, given a temporal matching of maximum cardinality, finding a minimum temporal edge cover remains NP-complete. Likewise, given a minimum temporal edge cover, finding a maximum temporal matching is also NP-complete. Observe that this implies that a temporal version of Gallai’s Theorem cannot hold unless $P = NP$. A complete version of this paper can be found in [4].

Related Works. Many variations of the temporal matching problem have been explored in the literature. The first definition of a temporal matching appears in [14], where it is defined as a set of temporal edges $\{(e_1, t_1), \dots, (e_q, t_q)\}$ such that $\{e_1, \dots, e_q\}$ forms a matching in the underlying static graph, and all timestamps are distinct. This constraint can be quite restrictive, as it permits selecting at most one edge per snapshot.

A relaxation of this constraint was introduced in [12] with the concept of a Δ -temporal matching. In this variation, temporal edges incident to the same vertex must have timestamps that differ by at least Δ . This concept arises from the idea of analyzing the graph through temporal windows of size Δ , which led to the definition of several Δ -related problems, summarized in [10]. In the latter work, they also introduce the notion of a Δ -edge cover, leaving open the related problem.

A closely related concept is that of a γ -matching in a link stream, introduced in [1], where γ is a fixed positive integer. Using our terminology, this corresponds to a set of temporal edges $\{(e_1, t_1), \dots, (e_q, t_q)\}$ such that $\{t_i, \dots, t_i + \gamma - 1\} \subseteq \lambda(e_i)$ for each $i \in [q]$, and whenever $|t_i - t_j| < \gamma$, then $e_i \cap e_j = \emptyset$. Observe that this is a special case of Δ -temporal matching.

2 Preliminaries

A (undirected, loopless) graph G is an ordered pair (V, E) , where V is a finite set and $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$. The elements of V are called *vertices* and the elements of E are called *edges*. Sometimes we use $V(G)$ and $E(G)$ to refer to the set of vertices and edges of G , respectively. Also, for simplicity, we write the elements of $E(G)$ as uv instead of $\{u, v\}$, while still using the notation $u \in uv$. Given $v \in V(G)$, let $\delta_G(v) = \{e \in E(G) \mid v \in e\}$ be the set of edges incident to v in G . Given a graph G , a positive integer τ and a function $\lambda : E(G) \rightarrow \mathcal{P}([\tau])$, with $\mathcal{P}([\tau])$ being the power set of $\{1, \dots, \tau\}$, such that each edge is assigned a finite non-empty subset of $[\tau]$. Then $\mathcal{G} = (G, \lambda)$ is a *temporal graph* with *lifetime*

τ . We can see the vertices and edges of \mathcal{G} in two ways. One is to see them as just the vertices and edges of G . The other is to add a temporal component to them. In this way, we have *temporal vertices* in the form $(v, i) \in V(G) \times [\tau]$, and *temporal edges* in the form (e, j) with $e \in E(G)$ and $j \in \lambda(e)$.

We recall some NP-complete problems that we use in the reductions of this paper.

3-SAT(2,2): given an input boolean formula F in conjunctive normal form, where each clause has three literals and each variable appears four times, of which exactly two times is negated, decide whether F is satisfiable and, if so, give an assignment that satisfies it.

SET COVER: given a pair (U, \mathcal{S}) and a nonnegative integer k , where $U = [n]$ for some n and $\mathcal{S} = \{S_1, \dots, S_m\}$ is a collection of subsets of U , determine (if it exists) a subcollection of at most k subsets S_{i_1}, \dots, S_{i_k} such that $U \subset \bigcup_{j=1}^k S_{i_j}$.

SET PACKING: given a collection of sets $\mathcal{S} = \{S_1, \dots, S_m\}$ and a nonnegative integer k , determine (if it exists) a subcollection of at least k pairwise disjoint sets in \mathcal{S} .

Finally, we recall the definition of *nice tree decomposition*, that we use for the FPT algorithms.

A *tree decomposition* of a graph G is a pair $(T, \{X_t\}_{t \in V(T)})$, where T is a tree and $\{X_t\}_{t \in V(T)}$ is a collection of subsets of $V(G)$ (called bags), such that the following three conditions hold:

1. Every vertex of G appears in at least one bag:

$$\bigcup_{t \in V(T)} X_t = V(G).$$

2. For every edge $(u, v) \in E(G)$, there exists a bag X_t such that both u and v are in X_t :

$$\forall (u, v) \in E(G), \exists t \in V(T) \text{ such that } \{u, v\} \subseteq X_t.$$

3. For every vertex $v \in V(G)$, the set of nodes $\{t \in V(T) \mid v \in X_t\}$ forms a subtree of T .

The *width* of a tree decomposition is defined as $\max_{t \in V(T)} |X_t| - 1$, i.e., the size of the largest bag minus one. The *treewidth* of a graph G is the minimum width over all possible tree decompositions of G .

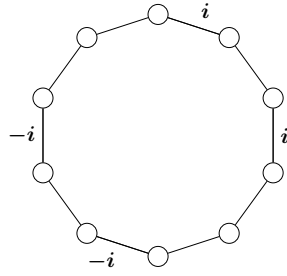
A tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of G is a *nice tree decomposition* if:

1. T is a rooted tree (call r its root), and each node $t \in V(T)$ is one of the following types:
 - *Leaf node*: t is a leaf of T , and $X_t = \emptyset$.
 - *Introduce node*: t has exactly one child t' , and $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$. We say that t *introduces* v .
 - *Forget node*: t has exactly one child t' , and $X_t = X_{t'} \setminus \{v\}$ for some $v \in X_{t'}$. We say that t *forgets* v .
 - *Join node*: t has exactly two children t_1 and t_2 , and $X_t = X_{t_1} = X_{t_2}$.
2. $B_r = \emptyset$.

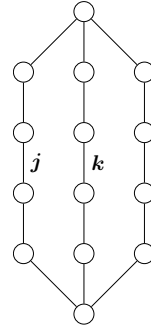
It is largely known that a nice tree decomposition can be obtained from a tree decomposition without increasing the width. We refer the reader to [5] for a very good introduction about how to obtain algorithms that run in FPT time when parameterized by the treewidth.

3 Hardness and Tractability of TEMPORAL EDGE COVER

In this section, we study the complexity of TEMPORAL EDGE COVER. Specifically, we show that TEMPORAL EDGE COVER is NP-complete when the lifetime τ of graph is 2, and then we show that it is NP-complete even when the underlying graph is a tree. This suggests that



■ Figure 3 Graph L_i .



■ Figure 4 Graph $T_{j,k,l}$.

both the lifetime τ and the treewidth w of a graph play an important role in the complexity of TEMPORAL EDGE COVER. Indeed, we describe an FPT algorithm in τ and w which solves the problem.

3.1 Hardness for $\tau = 2$

We prove that TEMPORAL EDGE COVER restricted to $\tau = 2$ is NP-complete by giving a reduction from 3-SAT(2,2). We first describe some (non temporal) graphs needed by our reduction, that have some edges marked (note that the marking is not the time labelling λ).

► **Definition 5.** Let i be a positive integer. We define the graph $L_i = (V_i, E_i)$ to be a cycle with 10 edges such that (1) two edges of E_i are marked i and two edges of E_i are marked $-i$ and (2) there is one unmarked edge between edges with the same marking, and two unmarked edges between edges of opposite marking.

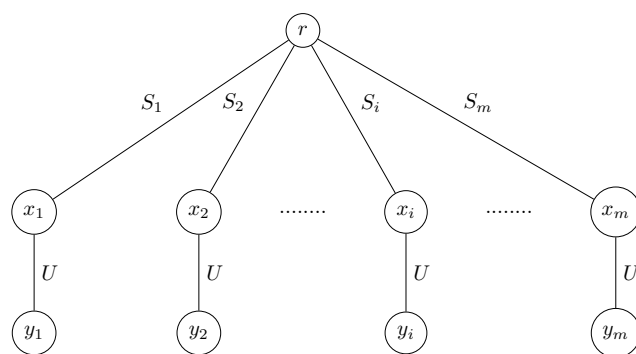
Graph L_i is shown in Figure 3. Note that, since it has ten vertices and ten edges, its vertices can be covered using five edges in two ways, denoted by $E'_{i,1}$ and $E'_{i,2}$:

- $E'_{i,1}$ contains both edges marked by i and no edge marked by $-i$
- $E'_{i,2}$ contains both edges marked by $-i$ and no edge marked by i

Given three integers j, k, l we define the graph $T_{j,k,l}$, with edges marked j, k, l as in Figure 4.

We use the graphs L_i and $T_{i,j,k}$ to define an instance of TEMPORAL EDGE COVER with lifetime 2 corresponding to an instance of 3-SAT(2,2). Consider an instance F of 3-SAT(2,2) consisting of clauses C_1, \dots, C_m over n variables x_1, \dots, x_n . Recall that each $C_j, j \in [m]$ has three literals and each variable $x_i, i \in [n]$, appears in exactly two clauses as a positive literal and in exactly two clauses as a negative literal. We construct a corresponding temporal graph \mathcal{G} , with lifetime $\tau = 2$, associated with F as follows:

- At time 1, \mathcal{G} is defined as a graph G_1 that contains, for each variable $x_i, i \in [n]$, a cycle L_i , as defined in Definition 5. Note that these cycles are all vertex disjoint.
- At time 2, \mathcal{G} is defined as a graph G_2 that contains, for each clause $C_p, p \in [m]$, over variables x_i, x_j, x_k , with $i, j, k \in [n]$, a graph $T_{j,k,l}^p$ isomorphic to $T_{j,k,l}$. The marked edges of $T_{j,k,l}^p$ are defined as follows. First, $T_{j,k,l}^p$ shares marked edges with $L_q, q \in \{j, k, l\}$, in G_1 . For each $q \in \{j, k, l\}$, if x_q is positive in C_p , then $T_{j,k,l}^p$ and L_q share an edge marked q , if x_q is negative in C_p , then $T_{j,k,l}^p$ and L_q share an edge marked $-q$. Note that we define a one-to-one correspondence between the marked edges of graphs $T_{j,k,l}^q$ and of the graphs L_i , since each L_i has two edges marked i and two edges marked $-i$, and



■ **Figure 5** The temporal graph obtained from an instance of SET COVER.

a formula in 3-SAT(2,2) has precisely two positive occurrences of each variable x_i and two occurrences of its negation. Thus, two distinct edges of L_i with the same marking corresponds to two distinct edges of some $T_{i,j,k}^p, T_{i,j',k'}^r$.

The resulting temporal graph can be constructed in polynomial starting from an instance F of 3-SAT(2,2). Using this reduction, we can prove that F is satisfiable if and only if there exists an edge cover of \mathcal{G} having at most $5n + 6m$ edges. The idea behind the proof is that each L_i can be covered with 5 edges by $E'_{i,1}$ or $E'_{i,2}$, while each $T_{j,k,l}^p$ must be covered using at least 6 non marked edges, with 6 being achieved only if at least one marked edge is part of the covering. Depending on which $E'_{i,a}$ is used for the covering, true or false is assigned to the corresponding variable x_i .

► **Theorem 6.** TEMPORAL EDGE COVER for graphs of lifetime 2 is NP-complete.

3.2 Hardness when the Underlying Graph is a Tree

We show that TEMPORAL EDGE COVER is NP-complete when the underlying graph is a tree by giving a reduction from SET COVER to TEMPORAL EDGE COVER.

Given an instance (U, \mathcal{S}, k) of SET COVER, where $U = [n]$ and \mathcal{S} consists of m sets S_1, \dots, S_m ($S_i \subseteq [n]$, for each $i \in [m]$), we construct a corresponding temporal graph \mathcal{G} (see Figure 5). \mathcal{G} has an underlying graph G which is a tree rooted in r ; r has m children x_1, \dots, x_m , and each x_i has a single child y_i , with $i \in [m]$. Function λ associates time label to each edge as follows: $\lambda(x_i y_i) = S_i$ and $\lambda(r x_i) = U$, for each $i \in [m]$. The idea of the reduction is that each edge $x_i y_i$, $i \in [m]$, must be in a temporal edge cover, and that the temporal vertices (r, j) , $j \in [m]$, are covered by edges incident in r that encode a set cover.

► **Theorem 7.** TEMPORAL EDGE COVER is NP-complete even when the underlying graph is a tree.

3.3 FPT algorithm in τ and treewidth for TEMPORAL EDGE COVER

In this subsection we present an FPT algorithm that finds the minimum cardinality of a temporal edge cover of \mathcal{G} . Note that we can assume, without loss of generality, that each temporal vertex of the temporal graph \mathcal{G} can be covered by at least one edge. That is, we can assume that there are no independent temporal vertex in \mathcal{G} , since those would not need to be covered and can be ignored during the computation.

8:8 Matching and Edge Cover in Temporal Graphs

Let $\mathcal{G} = (G, \lambda)$ be a temporal graph and consider a nice tree decomposition $(T, \{X_t\}_{t \in V(T)})$, with T rooted at r , of a G . For each $t \in V(T)$, let G_t be the subgraph of G containing all the vertices $v \in X_{t'}$ for any t' in the subtree rooted at t . Also, for any $X \subseteq V(G)$, let $E(X)$ denote the set of edges with some endpoint in X (formally, $E(X) = \{uv \in E(G) \mid u, v \in X\}$).

Given $R \subseteq V(G)$, we denote by $V^T(R)$ the set of temporal vertices $R \times [\tau]$; for simplicity, we write $V^T(G)$ to denote $V^T(V(G))$. Given $S \subseteq E(G)$, we denote by $V^T(S)$ the set of temporal vertices which are endpoints of S , i.e., $V^T(S) = \bigcup_{e \in S} \{(u, i) \mid i \in \lambda(e), u \in e\}$. Additionally, given $S \subseteq E(G)$ and $(u, i) \in V^T(G)$, we say that S covers (u, i) if there exists $e \in S$ such that $u \in e$ and $i \in \lambda(e)$. Observe that S covers $V^T(S)$.

As is usual the case when using tree decomposition, we work with partial solutions, i.e., with sets of edges that only partially cover the temporal vertices of G_t . This is because we might cover some temporal vertex $(u, i) \in X_t \times [\tau]$ only with an edge introduced later, i.e., with an edge uv such that $v \notin V(G_t)$. Therefore, for each node of T , we keep track of the temporal vertices within $X_t \times [\tau]$ that are covered and of the edges within $E(X)$ that are chosen. Formally, given $t \in V(T)$, for each $S \subseteq E(X_t)$ and each $C \subseteq X_t \times [\tau]$ with $V^T(S) \subseteq C$, we define:

If there is no such set S' , then $T_t(S, C) = +\infty$. Essentially, the function gives the minimum cardinality of a partial edge cover S' for the temporal graph $(G_t, \lambda \upharpoonright_{E(G_t)})$ such that:

- S is exactly the set of edges in $E(X_t)$ that are selected by S' ;
- C is exactly the set of temporal vertices in $X_t \times [\tau]$ covered by S' . Observe that these must include the endpoints of the temporal edges related to the edges selected in S , and this is why we ask for $V^T(S)$ to be contained in C ; and
- Each temporal vertex related to some vertex in $G_t \setminus X_t$ must be covered by S' .

Observe that $T_r(\emptyset, \emptyset)$ gives us the minimum cardinality of a temporal edge cover for \mathcal{G} . In what follows, we show how to recursively compute $T_t(S, C)$ for each $t \in V(T)$, $S \subseteq E(X_t)$, and $C \subseteq (X_t \times [\tau])$ with $V^T(S) \subseteq C$, depending of the type of node t .

- leaf: if t is a leaf, then $T_t(\emptyset, \emptyset) = 0$;
- introduce node: let $v \in V(G)$ be the vertex introduced by t and t' be its only child. Also, let D be the set of temporal vertices (u, i) with $u \neq v$ covered by some edge incident to v . Formally, $D = V^T(S \cap \delta_G(v)) \setminus (\{v\} \times [\tau])$. Additionally, let $S' = S \setminus \delta_G(v)$, $C' = C \setminus (\{v\} \times [\tau])$, and $k = |S \cap \delta_G(v)|$. We have that:

$$T_t(S, C) = \begin{cases} k + \min_{\hat{D} \subseteq D} T_{t'}(S', C' \setminus \hat{D}) & , \text{ if } V^T(S) \cap (\{v\} \times [\tau]) = C \cap (\{v\} \times [\tau]) \\ +\infty & , \text{ otherwise} \end{cases}$$

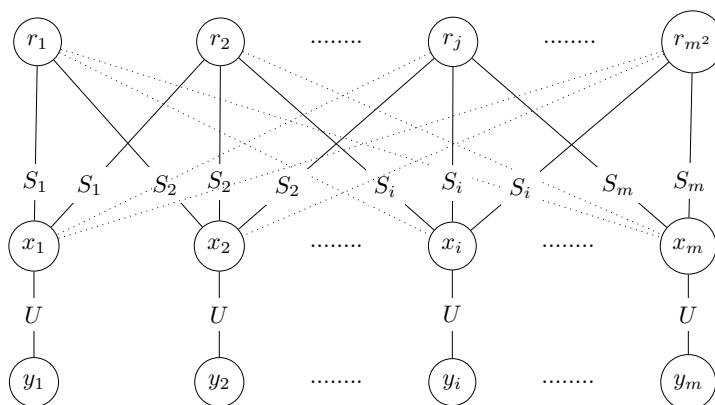
- forget node: let $v \in V(G)$ be the vertex forgotten by t and let t' be its only child. Also, let $S' = \delta_G(v) \cap E(X_{t'})$. Then:

$$T_t(S, C) = \min_{\hat{S} \subseteq S'} T_{t'}(S \cup \hat{S}, C \cup (\{v\} \times [\tau])).$$

- join node: let t_1 and t_2 be the two children of t . By definition, we know that $X_{t_1} = X_{t_2}$. Then:

$$T_t(S, C) = -|S| + \min\{T_{t_1}(S, C_1) + T_{t_2}(S, C_2) \mid C_1 \cup C_2 = C \text{ and } V^T(S) \subseteq C_1 \cap C_2\}.$$

► **Theorem 8.** TEMPORAL EDGE COVER can be computed in time $O^*(2^{w^2} \cdot 8^{w \cdot \tau})$.



■ **Figure 6** The temporal graph obtained from an instance of SET COVER. Some edges are dotted for readability, and we are not showing the labels on those edges for the same reason.

4 Approximation of TEMPORAL EDGE COVER

In this section we consider the approximability of TEMPORAL EDGE COVER. First, we show a bound on the approximation ($b \log \tau$, for any constant $0 < b < 1$), then we present an approximation algorithm of factor $O(\log \tau)$.

4.1 Inapproximability

We show that TEMPORAL EDGE COVER cannot be approximated within factor $b \log \tau$, for any constant $0 < b < 1$. We prove this result by giving an approximation preserving reduction from the SET COVER problem³. Consider an instance (U, \mathcal{S}) of SET COVER, where $U = \{u_1, \dots, u_n\}$ and $\mathcal{S} = \{S_1, \dots, S_m\}$. We can assume $U = [n]$, therefore each S_i , $i \in [m]$ is a subset of $[n]$. We define a corresponding instance $\mathcal{G} = (G, \lambda)$ of TEMPORAL EDGE COVER as follows (see Figure 6):

$$\begin{aligned} V(G) &= \{r_i \mid i \in [m^2]\} \cup \{x_i \mid i \in [m]\} \cup \{y_i \mid i \in [m]\} \\ E(G) &= \{r_i x_j \mid i \in [m^2], j \in [m]\} \cup \{x_i y_i \mid i \in [m]\} \\ \lambda : E(G) &\rightarrow \mathcal{P}([n]), \quad \lambda(e) = \begin{cases} S_j & \text{if } e = r_i x_j \text{ for some } i \in [m^2], j \in [m], \\ U & \text{if } e = x_i y_i \text{ for some } i \in [m]. \end{cases} \end{aligned}$$

Note that \mathcal{G} has lifetime $\tau = n$. We now show the main properties of the reduction.

► **Lemma 9.** *Consider an instance (U, \mathcal{S}) of SET COVER and a corresponding instance \mathcal{G} of TEMPORAL EDGE COVER. Given a solution \mathcal{S}' of SET COVER on instance (U, \mathcal{S}) , we can compute in polynomial time a solution of TEMPORAL EDGE COVER on instance \mathcal{G} that consists of at most $m + |\mathcal{S}'|m^2$ edges.*

► **Lemma 10.** *Consider an instance (U, \mathcal{S}) of SET COVER and a corresponding instance \mathcal{G} of TEMPORAL EDGE COVER. Given a solution E' of TEMPORAL EDGE COVER on instance \mathcal{G} , then there exists a positive integer k such that $|E'| = m + km^2$. Then we can compute in polynomial time a solution of SET COVER on instance (U, \mathcal{S}) that consists of at most k sets.*

³ In this section we consider the optimization version of SET COVER, thus we omit k from the instance of the problem.

8:10 Matching and Edge Cover in Temporal Graphs

■ **Algorithm 1** Approximation algorithm for TEMPORAL EDGE COVER

Input: a temporal graph $\mathcal{G} = (G, \lambda)$ with lifetime τ .

Output: an edge cover E' of \mathcal{G} of approximation factor $O(\log \tau)$

```

Mark each temporal vertex  $(v_i, t)$ ,  $i \in [n], t \in [\tau]$  as uncovered
 $i \leftarrow 1$ ;
 $E' \leftarrow \emptyset$ ;
foreach  $i \in [n]$  do
    Define an instance  $(U^i, \mathcal{S}^i)$  of SET COVER corresponding to  $v_i$ ;
    Compute (via a greedy approximation algorithm) an approximated solution  $\mathcal{C}^i$  of
    SET COVER on instance  $(U^i, \mathcal{S}^i)$ ;
    Compute an approximation edge cover  $E'_i$ , by adding an edge  $e_h$  to  $E'_i$  if and only
    if  $S^i_h \in \mathcal{C}^i$ ;
     $E' \leftarrow E' \cup E'_i$ ;
    Mark each temporal vertex covered by  $E'_i$  as covered;
     $i \leftarrow i + 1$ ;
end
Output  $E'$ 

```

► **Theorem 11.** TEMPORAL EDGE COVER *cannot be approximated within factor $b \log \tau$, for any b with $0 < b < 1$, unless $P = NP$.*

4.2 A $O(\log \tau)$ -Approximation Algorithm

In this section we present an approximation algorithm for TEMPORAL EDGE COVER of factor $O(\log \tau)$. Given a temporal graph $\mathcal{G} = (G, \lambda)$, with lifetime τ and $G = (V, E)$, the algorithm assumes that the vertices are ordered – the specific order is not relevant – so we denote them as v_1, \dots, v_n . The approximation algorithm, described in Algorithm 1, computes an edge cover E' by greedily covering the uncovered temporal vertices of each vertex v_i , $i \in [n]$, following the order (first it covers the uncovered temporal vertices of v_1 , then of v_2 and so on, until all the temporal vertices are covered). In order to cover the temporal vertices of each v_i , it applies the greedy algorithm of SET COVER on an instance that contains an element for each uncovered temporal vertex (v_i, t) and a set, for each edge $v_i v_j \in E$, that covers (v_i, t) for each $t \in \lambda(v_i v_j)$.

More precisely, consider the i -th iteration, $i \in [n]$, of Algorithm 1. Given the set E' of edges computed by the first $i - 1$ -iterations of the algorithm, we define an instance (U^i, \mathcal{S}^i) of SET COVER, where U^i is the universe set and \mathcal{S}^i is a collection of sets over U^i . For each $i \in [n]$, the universe set U^i is defined as

$$U^i = \{t \in [\tau] \mid (v_i, t) \text{ is not covered by } E' \text{ and there exists a } v_i v_j \in E \text{ such that } t \in \lambda(v_i v_j)\}.$$

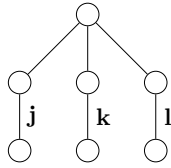
The collection of sets \mathcal{S}^i is defined as $\mathcal{S}^i = \{S^i_{e_1}, \dots, S^i_{e_z}\}$, where e_1, \dots, e_z are the edges incident in v_i and each $S^i_h \subseteq [\tau]$ is defined as $S^i_h = \{t \in [\tau] \mid t \in \lambda(e_h)\}$.

Algorithm 1 marks each temporal vertex as *covered* when it adds to solution E' an edge that covers it.

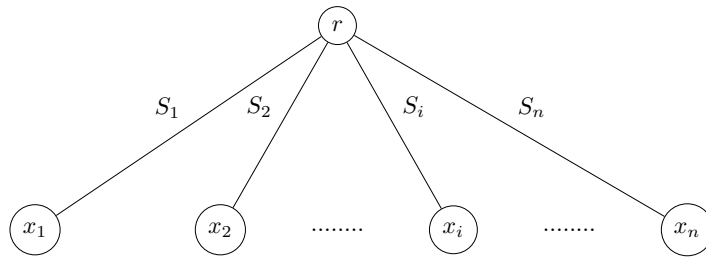
Now, we show the correctness of Algorithm 1.

► **Lemma 12.** *Let E' be a solution computed by Algorithm 1. Then, denoted by E^* an optimal solution of TEMPORAL EDGE COVER on instance \mathcal{G} , it holds that*

1. E' is an edge cover of \mathcal{G}
2. $|E'| \leq \log \tau |E^*|$.



■ **Figure 7** Graph $C_{j,k,l}$.



■ **Figure 8** Temporal graph associated to an instance of SET PACKING.

5 TEMPORAL MATCHING: Hardness and Tractability

In this section we consider the TEMPORAL MATCHING problem and provide hardness results and tractability. The outline is the same as TEMPORAL EDGE COVER; we show that TEMPORAL MATCHING is NP-complete when the lifetime τ of graph is 2, and then we show that it is NP-complete even when the underlying graph is a tree. Finally, we describe an FPT algorithm in τ and w (treewidth) which solves TEMPORAL MATCHING.

5.1 Hardness for $\tau = 2$

We show the NP-hardness of TEMPORAL MATCHING restricted to lifetime 2 with reduction from 3-SAT(2,2) similar to the one given in Section 3.1. This reduction follows the same idea as that of Theorem 6. Indeed, we still use the graph L_i defined in Definition 5 and showed in Figure 3. For this reduction we do not encode clauses with graphs isomorphic to $T_{j,k,l}$, but graphs isomorphic to $C_{j,k,l}$, shown in Figure 7. $C_{j,k,l}$ has three edges marked with integers j, k, l .

Let F be an instance of 3-SAT(2,2), with n variables x_1, \dots, x_n and m clauses C_1, \dots, C_m . We construct an associated temporal graph with lifetime $\tau = 2$ defined in the following way. At time 1, \mathcal{G} contains a graph G_1 consisting of the disjoint union of cycles $L_i, i \in [n]$, one for each variable x_i . At time 2, \mathcal{G} contains a graph G_2 that for each clause C_p over variables x_j, x_k and $x_l, j, k, l \in [n]$, contains graph $C_{j,k,l}^p$ isomorphic to $C_{j,k,l}$. As in Section 3.1, the marked edges of $C_{j,k,l}^p$ are shared with cycles $L_i, i \in \{j, k, l\}$ that encode the variables x_j, x_k and x_l . The shared marked edge between $C_{j,k,l}^p$ and L_i has mark $-i$ if x_i is negated in the clause, i if the variable is positive in the clause. Note that $C_{j,k,l}^p$'s are build so that the marked edges of L_i are in one-to-one correspondence with marked edges of $C_{j,k,l}$'s.

The correctness of the reduction follows from the fact that a maximum temporal matching of each $L_i, i \in [n]$, contains five edges, one including positively marked edges and one including negatively marked edges. This encodes an assignment to the variables. The temporal matching of each $C_{j,k,l}^p$ contains at most one unmarked edge. However, a temporal matching M of \mathcal{G} contains one unmarked edge of $C_{j,k,l}^p$ only if there is a marked edge shared by $C_{j,k,l}^p$ and some L_i that does not belong to M . This encodes the fact that at least one literal of each clause must be satisfied. This reduction allows us to prove the following result.

► **Theorem 13.** TEMPORAL MATCHING for graphs with lifetime 2 is NP-complete.

5.2 Hardness when the Underlying Graph is a Tree

We show a reduction from SET PACKING to TEMPORAL MATCHING. Given an instance (U, \mathcal{S}, k) of SET PACKING with $\mathcal{S} = \{S_1, \dots, S_n\}$ a collection of sets over a universe set U , we construct a temporal graph $\mathcal{G} = (G, \lambda)$ such that there exists k disjoint sets in \mathcal{S} if and

8:12 Matching and Edge Cover in Temporal Graphs

only if there exists a temporal matching M of \mathcal{G} of size at least k . Without loss of generality, we assume that $U = [n]$ and that each $S_i \subseteq [n]$, for each $i \in [m]$.

$\mathcal{G} = (G, \lambda)$ is defined as follows (the resulting temporal is presented in Figure 8):

- G is a tree rooted at a vertex r , which has n children x_1, \dots, x_n
- For each $i \in [n]$, $\lambda(rx_i) = S_i$

The idea of the reduction is that since any pair of edges rx_i and rx_j of \mathcal{G} , where $i, j \in [n]$ and $i \neq j$, share vertex r , then they can be in a temporal matching only if they are defined in different times, thus they are related to two disjoint subsets S_i and S_j in an instance of SET PACKING. Then, since SET PACKING is NP-complete [9], we can prove the following result.

► **Theorem 14.** TEMPORAL MATCHING is NP-complete even when the underlying graph G is a tree.

5.3 FPT algorithm in τ and treewidth for TEMPORAL MATCHING

In this subsection we show an algorithm that finds the maximum cardinality of a temporal matching of \mathcal{G} in FPT time when parameterized by τ plus the treewidth. The approach follows the same idea as the TEMPORAL EDGE COVER one (see Section 3.3).

Again, let $\mathcal{G} = (G, \lambda)$ be a temporal graph and consider a nice tree decomposition $(T, \{X_t\}_{t \in V(T)})$ of G , with T rooted at r . We use the same notation as the one used in Section 3.3. Given a matching $M \subseteq E(G)$ and a temporal vertex $(u, i) \in V^T(G)$, observe that (u, i) can be covered by M at most once, i.e., there is at most one edge $e \in M$ such that $u \in e$ and $i \in \lambda$. If such an edge exists, we say that M saturates (u, i) .

We define the dynamic programming table T_t related to each $t \in V(T)$ as follows. For each $N \subseteq E(X_t)$ and $C \subseteq V^T(X_t)$ with $V^T(N) \subseteq C$:

$$T_t(N, C) = \max\{k \mid \exists \text{ a temporal matching } M \subseteq E(G_t) \text{ s.t. } |M| = k, \\ M \cap E(X_t) = N, \text{ and } V^T(M) \cap V^T(X_t) = C\}.$$

If there exists no such set M (e.g., it could happen that no temporal matching saturates exactly C), then $T_t(N, C) = 0$. Essentially, the function gives the maximum cardinality of a temporal matching M for the temporal graph $(G_t, \lambda \upharpoonright_{E(G_t)})$ such that:

- N is exactly the set of selected edges in $E(X_t)$;
- C is exactly the set of temporal vertices in $V^T(X_t)$ saturated by M .

Because $G_r = G$, the value of $T_r(\emptyset, \emptyset)$ tells us the maximum cardinality of a temporal matching for \mathcal{G} . We show how to recursively compute $T_t(N, C)$ for each $t \in V(T)$, $N \subseteq E(X_t)$, and $C \subseteq V^T(X_t)$, depending on the type of node t .

- leaf: if t is a leaf, then $T_t(\emptyset, \emptyset) = 0$;
- introduce node: let $v \in V(G)$ be introduced by t and let t' be its only child. Also, let $F = N \cap \delta_G(v)$ be the set of edges in N incident to v . Then:

$$T_t(N, C) = \begin{cases} T_{t'}(N \setminus F, C \setminus V^T(F)) + |F| & , \text{ if } V^T(F) \cap (\{v\} \times [\tau]) = C \cap (\{v\} \times [\tau]) \\ 0 & , \text{ otherwise.} \end{cases}$$

- forget node: let $v \in V(G)$ be forgotten by t and let t' be its only child. To define the recursive function, let \mathcal{N} contain every $\hat{N} \subseteq \delta_G(v) \cap E(X_{t'})$ such that $V^T(\hat{N}) \setminus (\{v\} \times [\tau]) \subseteq C$ and such that \hat{N} is a matching. In words, it contains every subset of edges of $E(X_{t'})$ incident to v whose other endpoints are temporal vertices within C , while also not having

any edges sharing the same temporal vertices. Also, for any $\hat{N} \in \mathcal{N}$, let $\mathcal{C}_{\hat{N}}$ contain every $\hat{C} \subseteq \{v\} \times [\tau]$ such that $V^T(\hat{N}) \cap (\{v\} \times [\tau]) \subseteq \hat{C}$. Then

$$T_t(N, C) = \max\{T_{t'}(N \cup \hat{N}, C \cup \hat{C}) \mid \hat{N} \in \mathcal{N} \text{ and } \hat{C} \in \mathcal{C}_{\hat{N}}\}.$$

- join node: let t_1 and t_2 be the children of t . Recall that $X_{t_1} = X_{t_2}$. Then:

$$T_t(N, C) = -|N| + \max\{T_{t_1}(N, C_1) + T_{t_2}(N, C_2) \mid C_1 \cap C_2 = V^T(N) \text{ and } C_1 \cup C_2 = C\}.$$

- **Theorem 15.** TEMPORAL MATCHING can be computed in time $O^*(2^{w^2} \cdot 8^{w \cdot \tau})$.

6 Approximation of TEMPORAL MATCHING

In this section we consider the approximability of TEMPORAL MATCHING. We start by discussing a bound on the approximability of the problem. Since the reduction described in Section 5.2 is also approximation preserving (note that it defines $\tau = n$) and since SET PACKING is hard to approximated within factor $O(n^{1-\varepsilon})$ [9, 17], for any $\varepsilon > 0$, unless $P = NP$, then we have the following result.

- **Corollary 16.** TEMPORAL MATCHING cannot be approximated within factor $O(\tau^{1-\varepsilon})$, for any $\varepsilon > 0$, unless $P = NP$.

On the positive side, we can prove that TEMPORAL MATCHING can be easily approximated within factor τ , by computing a maximum matching in each snapshot and returning as approximated solution the one having maximum cardinality.

- **Theorem 17.** TEMPORAL MATCHING can be approximated in polynomial time within factor τ .

Proof. Consider the following approximation algorithm. For each $t \in [\tau]$, the approximation algorithm computes a maximum matching M_t of the static graph G_t , defined as \mathcal{G} restricted to time t (i.e. G_t is the snapshot of \mathcal{G} in t). Then the approximation algorithm returns as an approximated solution, denoted by M , a matching of maximum cardinality among M_t , $t \in [\tau]$.

First, note that M is a feasible solution of TEMPORAL MATCHING. Indeed, since M is a matching in a static graph, each pair of edges in M is vertex disjoint, hence M is also a temporal matching. Now, we prove that the approximation factor is indeed τ . Consider a maximum temporal matching M^* in \mathcal{G} . Consider the set of edges $M_t^* \subseteq M^*$ defined at time t , $t \in [\tau]$. By definition of temporal matching, the edges in M_t^* must be vertex disjoint, thus they must be a matching in G_t . Since for each $t \in [\tau]$ M_t is a maximum matching of G_t , it follows that $|M_t^*| \leq |M_t|$. By construction of M , we have

$$\sum_{t \in [\tau]} |M_t^*| \leq \sum_{t \in [\tau]} |M_t| \leq \tau |M|,$$

thus concluding the proof. ◀

7 Relation between Max Temporal Matching and Min Temporal Edge Cover

In this section, we show that having a minimal temporal edge cover does not facilitate the computation of a maximum temporal matching, and vice versa.

For a static graph, the problem of finding the maximum size of a matching and the problem of finding the minimum size of an edge cover are complementary. More specifically, given a graph G on n vertices and denoting the size of a minimum edge cover by $\beta'(G)$ and the size of maximum matching by $\alpha'(G)$, it is known that $\alpha'(G) + \beta'(G) = n$. Indeed, we can even construct a matching from an edge cover, and vice-versa. To see this, let M be a matching of size k . Picking M plus one edge incident to each non-saturated vertex gives us an edge cover of size $k + n - 2k$, thus implying that $\beta'(G) \leq n - \alpha'(G)$. On the other hand, if N is a minimal edge cover of cardinality k , observe that $G' = (V(G), N)$ is a forest of stars. Indeed, G' contains no cycles as removing an edge of a cycle in G' would cover the same vertices. Additionally, if G' contains a path $P = (v_1, v_2, v_3, v_4)$, then $N - v_2v_3$ still covers $V(G)$. Let k' be the number of components of G' and observe that we can construct a matching of size k' by picking one edge of each star of G' . Finally, it is known that a forest on n vertices and k' components has exactly $n - k'$ edges, i.e., $k = n - k'$, from which we get $\alpha'(G) \geq n - \beta'(G)$.

We now see that the temporal variants of matchings and edge covers are not related as in the static case. That is, given a temporal graph \mathcal{G} and a temporal matching of maximum cardinality, the problem of finding a minimum temporal edge cover for \mathcal{G} is still NP-complete. The opposite is also true, which means that if we are given a minimum temporal edge cover then the problem of finding a maximum temporal matching is still NP-complete. To see this, we use some of the reductions presented throughout the paper.

Let S_1, \dots, S_m be an instance of SET COVER. Theorem 7 and Figure 5 detail a reduction to TEMPORAL EDGE COVER, where the resulting temporal graph $\mathcal{G} = (G, \lambda)$ has lifetime $\tau = \max\{k \mid k \in S_i, 1 \leq i \leq m\}$. We now construct a temporal graph $\hat{\mathcal{G}} = (G, \mu)$ with lifetime $\tau + 1$ where $\mu(e) = \lambda(e) \cup \{\tau + 1\}$, for each $e \in E(G)$. That is, we add $\tau + 1$ to each label. Then any temporal matching of maximum cardinality for $\hat{\mathcal{G}}$ contains all the edges x_iy_i , $1 \leq i \leq m$ for each i except for at most one j , and in that case it contains rx_j . This does not depend on the specific instance S_1, \dots, S_m considered. Still, any temporal edge cover of minimum cardinality is a solution for our instance of SET COVER, since the addition of the same element $\tau + 1$ to all labels does not change which edges are a solution. Therefore having a temporal matching of maximum cardinality does not change the complexity of finding a temporal edge cover of minimum cardinality.

On the other hand, suppose that for a temporal graph we know all its temporal edge covers of minimum cardinality, and we want to find a temporal matching of maximum cardinality. Then we can use the reduction from packing set detailed in Theorem 14 and Figure 8. Indeed, the only edge cover takes all the edges of the graph, but the matching depends on the specific sets S_1, \dots, S_n . Thus having a temporal edge cover of minimum cardinality does not change the complexity of finding a temporal matching of maximum cardinality.

8 Conclusion

In this paper, we have investigated the computational complexity of EDGE COVER in temporal graphs. We quickly identified the most interesting case (see again Table 1), which we simply named TEMPORAL EDGE COVER. We presented two NP-completeness results for this problem, one which uses lifetime $\tau = 2$, and another where the underlying graph is a tree (i.e. treewidth equals 1). These results complement our following FPT result, as the parameters considered in our proposed algorithm are τ and treewidth. Then, we have explored approximation of TEMPORAL EDGE COVER and provided an approximation algorithm with an

asymptotically tight approximation factor of $O(\log \tau)$. Inspired by the intrinsic connection between EDGE COVER and MATCHING in (non-temporal) graphs, we also have provided such results for TEMPORAL MATCHING. Surprisingly, even though the problems are shown to be distinct and unrelated to each other in the temporal setting, we have proved very similar results for both (albeit through different reductions and observations).

Although we have presented a comprehensive overview, covering classical complexity, parameterized complexity in terms of lifetime and treewidth, and approximation, we identify the following directions for future research. It may be interesting to identify specific classes of temporal graphs for which tractability of the (non-parametrised) problems is possible, and even more so if these classes correspond to a natural setting for real-life applications of our problems (e.g. TEMPORAL EDGE COVER in planar graphs possibly representing surveillance of a building floor). In terms of parametrized complexity, other parameters can be considered, such as some recently introduced parameters specifically for temporal graphs (see, e.g., the parameters studied and mentioned in [6]).

References

- 1 Julien Baste, Binh-Minh Bui-Xuan, and Antoine Roux. Temporal matching. *Theoretical Computer Science*, 806:184–196, 2020. doi:10.1016/j.tcs.2019.03.026.
- 2 Kenneth A Berman. Vulnerability of scheduled networks and a generalization of Menger’s Theorem. *Networks: An International Journal*, 28(3):125–134, 1996. doi:10.1002/(SICI)1097-0037(199610)28:3<125::AID-NET1>3.0.CO;2-P.
- 3 Arnaud Casteigts, Paola Flocchini, Walter Quattrociocchi, and Nicola Santoro. Time-varying graphs and dynamic networks. *International Journal of Parallel, Emergent and Distributed Systems*, 27(5):387–408, 2012. doi:10.1080/17445760.2012.668546.
- 4 Lapo Cioni, Riccardo Dondi, Andrea Marino, Jason Schoeters, and Ana Silva. Matching and edge cover in temporal graphs. *arXiv e-prints*, 2025. doi:10.48550/arXiv.2504.06762.
- 5 Marek Cygan, Fedor V Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 6 Jessica Enright, Samuel D. Hand, Laura Larios-Jones, and Kitty Meeks. Structural Parameters for Dense Temporal Graphs. In Rastislav Kráľovič and Antonín Kučera, editors, *49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024)*, volume 306 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52:1–52:15, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2024.52.
- 7 Tibor Gallai. Über extreme punkt-und kantenmengen, annales universitatis scientiarum budapestinensis de rolando eotvos nominatae. *Sectio mathematica*, 2:133–138, 1959.
- 8 Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):234, 2015.
- 9 Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. doi:10.1007/978-1-4684-2001-2_9.
- 10 Nina Klobas, George B. Mertzios, and Paul G. Spirakis. Sliding into the Future: Investigating Sliding Windows in Temporal Graphs. In Jérôme Leroux, Sylvain Lombardy, and David Peleg, editors, *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 5:1–5:12, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2023.5.

8:16 Matching and Edge Cover in Temporal Graphs

- 11 Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time. *Social Network Analysis and Mining*, 8(1):61, 2018.
- 12 George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. *Journal of Computer and System Sciences*, 137:1–19, 2023. doi:10.1016/j.jcss.2023.04.005.
- 13 Othon Michail. An introduction to temporal graphs: An algorithmic perspective. *Internet Mathematics*, 12(4):239–280, 2016. doi:10.1080/15427951.2016.1177801.
- 14 Othon Michail and Paul G. Spirakis. Traveling salesman problems in temporal graphs. *Theoretical Computer Science*, 634:1–23, 2016. doi:10.1016/j.tcs.2016.04.006.
- 15 Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.
- 16 B Bui Xuan, Afonso Ferreira, and Aubin Jarry. Computing shortest, fastest, and foremost journeys in dynamic networks. *International Journal of Foundations of Computer Science*, 14(02):267–285, 2003. doi:10.1142/S0129054103001728.
- 17 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(1):103–128, 2007. doi:10.4086/TOC.2007.V003A006.