A New Temporal Interpretation of Cluster Editing

Cristiano Bocci[1] [0000−0002−7874−4419], Chiara Capresi[1] [0000−0002−8037−6862], Kitty Meeks[2] [0000−0001−5299−3073], and John Sylvester[2] [0000−0002−6543−2934]

[1] Dipartimento di Ingegneria dell'Informazione e Scienze Matematiche, Università degli Studi di Siena
cristiano.bocci@unisi.it, capresi3@student.unisi.it
[2] School of Computing Science, University of Glasgow
kitty.meeks@glasgow.ac.uk, john.sylvester@glasgow.ac.uk

Abstract. The NP-complete graph problem Cluster Editing seeks to transform a static graph into disjoint union of cliques by making the fewest possible edits to the edge set. We introduce a natural interpretation of this problem in the setting of temporal graphs, whose edge-sets are subject to discrete changes over time, which we call Editing to Temporal Cliques. This problem is NP-complete even when restricted to temporal graphs whose underlying graph is a path, but we obtain two polynomial-time algorithms for special cases with further restrictions. In the static setting, it is well-known that a graph is a disjoint union of cliques if and only if it contains no induced copy of P_3; we demonstrate that there can be no universal characterisation of cluster temporal graphs in terms of subsets of at most four vertices. However, subject to a minor additional restriction, we obtain a characterisation involving forbidden configurations on five vertices. This characterisation gives rise to an FPT algorithm parameterised simultaneously by the permitted number of modifications and the lifetime of the temporal graph, which uses a simple search-tree strategy.

Keywords: Temporal graphs, cluster editing, graph clustering, parameterised complexity

1 Introduction

The Cluster Editing problem encapsulates one of the simplest and best-studied notions of graph clustering: given a graph G, the goal is to decide whether it is possible to transform G into a disjoint union of cliques – a *cluster graph* – by adding or deleting a total of at most k edges. While this problem is known to be NP-complete in general [2,12,16,21], it has been investigated extensively through the framework of parameterised complexity, and admits efficient parameterised algorithms with respect to several natural parameters [1,3,7–9,13,17] (for more details see Section 1.1).

Motivated by the fact that many real-world networks of interest are subject to discrete changes over time, there has been much research in recent years into the complexity of graph problems on *temporal graphs*, which provide a natural model for networks exhibiting these kinds of changes in their edge-sets. A first attempt to generalise Cluster Editing to the temporal setting was made by Chen, Molter, Sorge and Suchý [10], who recently introduced the problem Temporal Cluster Editing: here the goal is to ensure that graph appearing at each timestep is a cluster graph, subject to restrictions on both the number of modifications that can be made at each timestep and the differences between the cluster graphs created at consecutive timesteps. A dynamic version of the problem, Dynamic Cluster Editing, has also recently been studied by Luo, Molter, Nichterlein and Niedermeier [18]: here we are given a solution to a particular instance, together with a second instance (that which will be encountered at the next timestep) and are asked to find a solution for the second instance that does not differ too much from the first. One drawback of previous approaches is that they require each snapshot to be a cluster graph. In the static case, the notion of cluster graph is far too rigid for any meaningful application to community detection [23], as it is unreasonable that all pairs in a community are linked by an edge. For temporal graphs this assumption is even more restrictive.

We take a different approach, using a notion of temporal clique that already exists in the literature [14, 22]. Under this notion, a temporal clique is specified by a vertex-set and a time-interval, and we require that each pair of vertices is connected by an edge frequently, but not necessarily continuously, during the time-interval. An example could be emails within a company, where vertices are employees and there is an edge at time t between two employees if they are senders/recipients of an email at time t. Pairs of employees may correspond more or less frequently, however each pair is included in regular company-wide circular emails. We say that a temporal graph is a *cluster temporal graph* if it is a union of temporal cliques that are pairwise *independent*: here we say that two temporal cliques are independent if either their vertex sets are disjoint, or their time intervals are sufficiently far apart (similar to the notion of independence used to define temporal matchings [20]). Equipped with these definitions, we introduce a new temporal interpretation of cluster editing, which we call Editing to Temporal Cliques (ETC): the goal is to add/delete a collection of at most k edge appearances so that the resulting graph is a cluster temporal graph.

We prove that ETC is NP-hard, even when the underlying graph is a path; this reduction, however, relies on edges appearing at many distinct timesteps, and we show that, when restricted to paths, ETC is solvable in polynomial time when the maximum number of timesteps at which any one edge appears in the graph is bounded. It follows immediately from our hardness reduction that the variant of the problem in which we are only allowed to delete, but not add, edge appearances, is also NP-hard in the same setting. On the other hand, the corresponding variant in which we can only add edges, which we call Completion to Temporal Cliques (CTC), admits a polynomial-time algorithm on arbitrary inputs.

In the static setting, a key observation – which gives rise to a simple FPT search-tree algorithm for Cluster Editing parameterised by the number of modifications – is the fact that a graph is a cluster graph if and only if it contains no induced copy of the three-vertex path P_3 (sometimes referred to as a *conflict triple* [6]). We demonstrate that cluster temporal graphs cannot be fully characterised by any local condition that involves only sets of at most four vertices; however, in the most significant technical contribution of this paper, we prove that (subject to a minor additional restriction on the relationship between the spacing parameters that define temporal cliques and independence) a temporal graph is a cluster temporal graph if and only if every subset of at most five vertices induces a cluster temporal graph. Using this characterisation, we obtain an FPT algorithm for ETC parameterised simultaneously by the number of modifications and the lifetime (# of timesteps) of the input temporal graph.

1.1 Related Work

Cluster Editing is known to be NP-complete [2, 12, 16, 21], even for graphs with maximum degree six and when at most four edge modifications incident to each vertex are allowed [15]. On the positive side, the problem can be solved in polynomial time if the input graph has maximum degree two [5] (recently improved to degree three [3]) or is a unit interval graph [19]. Further complexity results and heuristic approaches are discussed in a survey article [6].

Variations of the problem in which only deletions or additions of edges respectively are allowed have also been studied. The version in which edges can only be added is trivially solvable in polynomial time, since an edge must be added between vertices u and v if and only if u and v belong to the same connected component of the input graph but are not already adjacent. The deletion version, on the other hand, remains NP-complete even on 4-regular graphs, but is solvable in polynomial time on graphs with maximum degree three [15].

Cluster Editing has received substantial attention from the parameterised complexity community, with many results focusing on the natural parameterisation by the number k of permitted modifications. Fixed-parameter tractability with respect to this parameter can easily be deduced from the fact that a graph is a cluster graph if and only if it contains no induced copy of the three-vertex path P_3, via a search tree argument; this approach has been refined repeatedly in non-trivial ways, culminating in an algorithm running in time $\mathcal{O}(1.76^k + m + n)$ for graphs with n vertices and m edges [7]. More recent work has considered as a parameter the number of modifications permitted above the lower bound implied by the number of modification-disjoint copies of P_3 (copies of P_3 such that no two share either an edge or a non-edge) [17]. Other parameters that have been considered include the number of clusters [13] and a lower bound on the permitted size of each cluster [1].

1.2 Organisation of the Paper

We begin in Section 2 by introducing some notation and definitions, and formally defining the ETC problem. In Section 3 we collect several results and fundamental lemmas which are either used in several other sections or may be of independent interest. In Section 4 we restrict to temporal graphs which have a path as the underlying graph: in Section 4.1 we show that ETC is NP-hard even in this setting, however in Section 4.2 we then show that if we further restrict temporal graphs on paths to only have a bounded number of appearances of each edge then ETC is solvable in polynomial time. In Section 5 we consider a variant of the ETC problem where edges can only be added, and show that this can be solved in polynomial time on any temporal graph. Finally in Section 6 we present our main result which gives a characterisation of cluster temporal graphs by induced temporal subgraphs on five vertices. We prove this result in Section 6.1 before applying it in Section 6.2 to show that (subject to minor additional restrictions) ETC is in FPT when parameterised by the lifetime of the temporal graph and number of permitted modifications. Due to space constraints, many proofs are omitted but can be found in the full version of the paper [4].

2 Preliminaries

In this section we first give basic definitions and introduce some new notions that are key to the paper, before formally specifying the ETC problem.

2.1 Notation and Definitions

Elementary Definitions. Let $\mathbb{N}$ denote the natural numbers (with zero) and $\mathbb{Z}^+$ denote the positive integers. We refer to a set of consecutive natural numbers $[i,j] = \{i, i+1, \ldots, j\}$ for some $i, j \in \mathbb{N}$ with $i \leq j$ as an *interval*, and to the number $j - i + 1$ as the *length* of the interval. Given an undirected (static) graph $G = (V, E)$ we denote its vertex-set by $V = V(G)$ and edge-set by $E = E(G) \subseteq \binom{V}{2}$. We work in the word RAM model of computation, so that arithmetic operations on integers represented using a number of bits logarithmic in the total input size can be carried out in time $\mathcal{O}(1)$. We use standard notions from parameterised complexity, following the notation of [11].

Temporal Graphs. A *temporal graph* $\mathcal{G} = (G, \mathcal{T})$ is a pair consisting of a static (undirected) underlying graph $G = (V, E)$ and a labeling function $\mathcal{T} : E \to 2^{\mathbb{Z}^+} \setminus \{\emptyset\}$. For a static edge $e \in E$, we think of $\mathcal{T}(e)$ as the set of time appearances of e in $\mathcal{G}$ and let $\mathcal{E}(\mathcal{G}) := \{(e,t) \mid e \in E \text{ and } t \in \mathcal{T}(e)\}$ denote the set of edge appearances, or *time-edges*, in a temporal graph $\mathcal{G}$. We consider temporal graphs $\mathcal{G}$ with *finite lifetime* given by $T(\mathcal{G}) := \max\{t \in \mathcal{T}(e) \mid e \in E\}$, that is, there is a maximum label assigned by $\mathcal{T}$ to an edge of G. We assume w.l.o.g. that $\min\{t \in \mathcal{T}(e) \mid e \in E\} = 1$. We denote the lifetime of $\mathcal{G}$ by T when $\mathcal{G}$ is clear from the context. The *snapshot* of $\mathcal{G}$ *at time* t is the static graph on V with edge set $\{e \in E \mid t \in \mathcal{T}(e)\}$. Given temporal graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, let $\mathcal{G}_1 \triangle \mathcal{G}_2$ be the set of time-edges appearing in exactly one of $\mathcal{G}_1$ or $\mathcal{G}_2$. For the purposes of computation, we assume that $\mathcal{G}$ is given as a list of (static) edges together with the list of times $\mathcal{T}(e)$ at which each static edge appears, so the size of $\mathcal{G}$ is $\mathcal{O}(\max\{|\mathcal{E}|, |V|\}) = \mathcal{O}(|V|^2 T)$.

Templates and Cliques. For an edge $e \in E(G)$ in the underlying graph of a temporal graph $\mathcal{G} = (G, \mathcal{T})$, an interval $[a, b]$, and $\Delta_1 \in \mathbb{Z}^+$, we say that e is Δ_1-*dense* in $[a, b]$ if for all $\tau \in [a, \max\{a, b - \Delta_1 + 1\}]$ there exists a $t \in \mathcal{T}(e)$ with $t \in [\tau, \tau + \Delta_1 - 1]$. This formalises the idea of two vertices being connected 'frequently, but not continuously' from the introduction. We define a *template* to be a pair $C = (X, [a, b])$ where X is a set of vertices and $[a, b]$ is an interval. For a set S of time-edges we let $V(S)$ denote the set of all endpoints of time-edges in S, and the *lifetime* $L(S) = [s, t]$, where $s = \min\{s : (e, s) \in S\}$ and $t = \max\{t : (e, t) \in S\}$. We say that S *generates* the template $(V(S), L(S))$. A set $S \subset \mathcal{E}(\mathcal{G})$ forms a Δ_1-*temporal clique* if for every pair $x, y \in V(S)$ of vertices in the template $(V(S), L(S))$ generated by S, the edge xy is Δ_1-dense in $L(S)$. We can assume that the lifetime of any template generated by a set S is minimal, that is, a time-edge from S occurs at each end-point of $L(S)$.

Independence and Cluster Temporal Graphs. For $\Delta_2 \in \mathbb{Z}^+$ we say that two templates $(X, [a, b])$ and $(Y, [c, d])$ are Δ_2-*independent* if

$$X \cap Y = \emptyset \quad \text{ or } \quad \min_{s \in [a,b], t \in [c,d]} |s - t| \geq \Delta_2.$$

Thus, two templates are independent if they share no vertices, or their time intervals are at least Δ_2 time steps apart. Let $\mathfrak{T}(\mathcal{G}, \Delta_2)$ be the class of all collections of pairwise Δ_2-independent templates where each $(X, [a, b]) \in \mathfrak{T}(\mathcal{G}, \Delta_2)$ satisfies $X \subseteq V(G)$ and $1 \leq a \leq b \leq T(\mathcal{G})$. Two sets S_1, S_2 of time-edges are Δ_2-*independent* if the templates they generate are Δ_2-independent. As a special case of this, two time-edges (e, t), (e', t') are Δ_2-*independent* if $e \cap e' = \emptyset$ or $|t - t'| \geq \Delta_2$. A temporal graph $\mathcal{G}$ *realises a collection* $\{(X_i, [a_i, b_i])\}_{i \in [k]} \in \mathfrak{T}(\mathcal{G}, \Delta_2)$ of pairwise Δ_2-independent templates if

- for each $(xy, t) \in \mathcal{E}$ there exists $i \in [k]$ such that $x, y \in X_i$ and $t \in [a_i, b_i]$,
- for each $i \in [k]$ and $x, y \in X_i$, the edge xy is Δ_1-dense in $[a_i, b_i]$.

The first condition specifies that every time-edge of $\mathcal{G}$ is contained in a single template. The second states that for any template, and any pair of vertices in vertex set of the template, there is a time edge between the vertices contained in any time window of length Δ_1 contained in the lifetime of the template.

If there exists some $\mathcal{C} \in \mathfrak{T}(\mathcal{G}, \Delta_2)$ such that $\mathcal{G}$ realises $\mathcal{C}$ then we call $\mathcal{G}$ a (Δ_1, Δ_2)-*cluster temporal graph.* Throughout we assume that $\Delta_2 > \Delta_1$, since if $\Delta_2 \leq \Delta_1$ then one Δ_1-temporal clique can realise many different sets of Δ_2-independent templates. For example, if $\Delta_2 = \Delta_1$ then the two time-edges (e, t) and $(e, t+\Delta_1)$ are Δ_2-independent but also e is Δ_1-dense in the interval $[t, t+\Delta_1]$.

Induced, Indivisible, and Saturated Sets. Let S be a set of time edges and A be a set of vertices, then we let $S[A] = \{(xy,t) \in S : x,y \in A\}$ be the set of all the time-edges in S *induced* by A. Similarly, given a temporal graph $\mathcal{G}$ and $A \subset V$, we let $\mathcal{G}[A]$ be the temporal graph with vertex set A and temporal edges $\mathcal{E}[A]$. For an interval $[a,b]$ we let $\mathcal{G}|_{[a,b]}$ be the temporal graph on V with the set of time-edges $\{(e,t) \in \mathcal{E}(\mathcal{G}) : t \in [a,b]\}$. We will say that a set S of time-edges is Δ_2-*indivisible* if there does not exist a pairwise Δ_2-independent collection $\{S_1,\dots,S_k\}$ of time-edge sets satisfying $\cup_{i\in[k]} S_i = S$. A Δ_2-indivisible set S is said to be Δ_2-*saturated in* $\mathcal{G}$ if after including any other time-edge of $\mathcal{E}(\mathcal{G})$ it would cease to be Δ_2-indivisible.

2.2 Problem Specification

Editing to Temporal Cliques. We can now introduce the ETC problem which, given as input a temporal graph $\mathcal{G}$ and natural numbers $k, \Delta_1, \Delta_2 \in \mathbb{Z}^+$, asks whether it is possible to transform $\mathcal{G}$ into a (Δ_1,Δ_2)-cluster temporal graph by applying at most k modifications (addition or deletion) of time-edges. Given any temporal graph $\mathcal{G}$, the set Π of time-edges which are added to or deleted from $\mathcal{G}$ is called the *modification set*. We note that the modification set Π can be defined as the symmetric difference between the time-edge set $\mathcal{E}(\mathcal{G})$ of the input graph and that of the same graph after the modifications have been applied. More formally, our problem can be formulated as follows.

Editing to Temporal Cliques (ETC):
Input: A temporal graph $\mathcal{G} = (G, \mathcal{T})$ and positive integers $k, \Delta_1, \Delta_2 \in \mathbb{Z}^+$.
Question: Does there exist a set Π of time-edge modifications, of cardinality at most k, such that the modified temporal graph is a (Δ_1,Δ_2)-cluster temporal graph?

We begin with a simple observation about the hardness of ETC which shows we can only hope to gain tractability in settings where the static version is tractable. However, we shall see in Section 4.1 that ETC is hard on temporal graphs with paths as their underlying graphs, and thus the converse is false.

Proposition 1. *Let $\mathcal{C}$ be a class of graphs on which* Cluster Editing *is* NP-*complete. Then* ETC *is* NP-*complete on the class of temporal graphs* $\{(G,\mathcal{T}) : G \in \mathcal{C}\}$.

3 Basic Observations on ETC

In this section we collect many fundamental results on the structure of temporal graphs and the cluster editing problem. We will use many of these results frequently throughout the proofs of results in this paper; we include all lemma statements here as they provide some insight into the behaviour of (Δ_1, Δ_2)-cluster temporal graphs and may be of use in the further study of cluster editing in the temporal setting.

Lemma 1 shows that there is a way to uniquely partition any temporal graph.

Lemma 1. *For any $\Delta_2 \in \mathbb{Z}^+$, any temporal graph $\mathcal{G}$ has a unique decomposition of its time-edges into Δ_2-saturated subsets.*

The next three elementary lemmas are useful for relating indivisible sets to Δ_1-temporal cliques to clusters in the proof of the characterisation, Theorem 4.

Lemma 2. *If two Δ_2-indivisible sets S_1 and S_2 of time-edges satisfy $S_1 \cap S_2 \neq \emptyset$, then $S_1 \cup S_2$ is Δ_2-indivisible.*

Lemma 3. *Let $\mathcal{G}$ be a temporal graph, $S \subseteq \mathcal{E}(\mathcal{G})$ be a Δ_2-saturated set of time-edges, and $\mathcal{K}$ a Δ_1-temporal clique such that $\mathcal{K} \subseteq \mathcal{E}(\mathcal{G})$ and $\mathcal{K} \cap S \neq \emptyset$. Then $\mathcal{K} \subseteq S$.*

Lemma 4. *Let $\mathcal{G}$ be any (Δ_1, Δ_2)-cluster temporal graph. Then, any Δ_2-indivisible set $S \subseteq \mathcal{E}(\mathcal{G})$ must be contained within a single Δ_1-temporal clique.*

However, the first application of these lemmas is the following result, which shows that the partition from Lemma 2 can be found in polynomial time.

Lemma 5. *Let $\mathcal{G} = (G = (V, E), \mathcal{T})$ be a temporal graph, and let $\mathcal{E} = \{(e, t) : e \in E, t \in \mathcal{T}(e)\}$ be the set of time-edges of $\mathcal{G}$. Then, there is an algorithm which finds the unique partition of $\mathcal{E}$ into Δ_2-saturated subsets in time $\mathcal{O}(|\mathcal{E}|^3|V|)$.*

Since any temporal graph has a unique decomposition into Δ_2-saturated sets by Lemma 1, and using the fact that any pair of Δ_2-saturated sets is Δ_2-independent by definition, we obtain the following corollary to Lemma 4.

Lemma 6. *A temporal graph $\mathcal{G}$ is a (Δ_1, Δ_2)-cluster temporal graph if and only if every Δ_2-saturated set of time-edges forms a Δ_1-temporal clique.*

Lemmas 5 and 6 allow us to deduce the following result.

Lemma 7. *Let $\mathcal{G} = (G = (V, E), \tau)$ be a temporal graph, and let $\mathcal{E} = \{(e, t) : e \in E, t \in \mathcal{T}(e)\}$ be the set of time-edges of $\mathcal{G}$. Then, we can determine in time $\mathcal{O}(|\mathcal{E}|^3|V|)$ whether $\mathcal{G}$ is a (Δ_1, Δ_2)-cluster temporal graph.*

The next three Lemmas concern induced structures in cluster temporal graphs.

Lemma 8. *Let $\mathcal{G}$ be a (Δ_1, Δ_2)-cluster temporal graph and $S \subseteq V(\mathcal{G})$. Then, $\mathcal{G}[S]$ is also a (Δ_1, Δ_2)-cluster temporal graph.*

Lemma 9. *Let $\mathcal{G} = (G, \mathcal{T})$ be a temporal graph, and $\mathcal{C} \in \mathfrak{T}(\mathcal{G}, \Delta_2)$ be a collection minimising $\min_{\mathcal{G}_\mathcal{C} \text{ realises } \mathcal{C}} |\mathcal{G} \triangle \mathcal{G}_\mathcal{C}|$. Then, for any template $C = (X, [a, b]) \in \mathcal{C}$, the static underlying graph of $\mathcal{G}[X]|_{[a,b]}$ is connected.*

Lemma 10. *Let $\mathcal{G}$ be a temporal graph. Then, there exists a (Δ_1, Δ_2)-cluster temporal graph $\mathcal{G}'$, minimising the edit distance $|\mathcal{G} \triangle \mathcal{G}'|$ between $\mathcal{G}$ and $\mathcal{G}'$, such that the lifetime of $\mathcal{G}'$ is a subset of the lifetime of $\mathcal{G}$.*

4 ETC on Paths

Throughout P_n will denote the path on $V(P_n) = \{v_1, \dots, v_n\}$ with $E(P_n) = \{v_i v_{i+1} : 1 \leq i < n\}$. Define $\mathfrak{F}_n$ be the set of all temporal graphs $\mathcal{P}_n = (P_n, \mathcal{T})$ on n vertices which have the path P_n as the underlying static graph.

4.1 NP-Completeness

Clearly, ETC is in NP because, for any input instance $(\mathcal{G}, \Delta_1, \Delta_2, k)$, a non-deterministic algorithm can guess (if one exists) the modification set Π and, using Lemma 7, verify that that the modified temporal graph is a (Δ_1, Δ_2)-cluster temporal graph in time polynomial in k and the size of $\mathcal{G}$. We show that ETC is NP-hard even for temporal graphs with a path as underlying graph.

Theorem 1. ETC *is* NP*-complete, even if the underlying graph G of the input temporal graph $\mathcal{G}$ is a path.*

To prove this result we construct a reduction to ETC from the NP-complete problem TEMPORAL MATCHING. For a fixed $\Delta \in \mathbb{Z}^+$, a *Δ-temporal matching* $\mathcal{M}$ of a temporal graph $\mathcal{G}$ is a set of time-edges of $\mathcal{G}$ which are pairwise Δ-independent. It is easy to note that if $\mathcal{G} = \mathcal{M}$, then $\mathcal{G}$ is a (Δ_1, Δ)-cluster temporal graph for any value of $\Delta_1 \geq 1$, because then each time-edge in $\mathcal{G}$ can be considered as a Δ_1-temporal clique with unit time interval, and these cliques are by definition Δ-independent. We can now state this problem formally.

TEMPORAL MATCHING (TM):
Input: A temporal graph $\mathcal{G} = (G, \mathcal{T})$ and two positive integers $k, \Delta \in \mathbb{Z}^+$.
Question: Does $\mathcal{G}$ admit a Δ-temporal matching $\mathcal{M}$ of size k?

It was shown in [20] that TEMPORAL MATCHING is NP-complete even if $\Delta = 2$ and the underlying graph G is a path. The reduction fixes $\Delta_1 = 1$ and $\Delta_2 = 5$. It then takes our input temporal graph $\mathcal{P}_n$ and transforms it into an new instance $\mathcal{P}'_n$ by adding empty "filler" snapshots between each snapshot $\mathcal{P}_n$, see Figure 1. It is shown that a matching in the original instance corresponds to a $(1, 5)$-cluster temporal graph, which gives one direction of the reduction. We then show that, if enough filler snapshots are added, then there exists an optimal solution to ETC where time-edges are only deleted from $\mathcal{P}'_n$. We can then deduce from this that, since the underlying graph is a path, a solution to ETC using only deletions must be a matching of the required size.

4.2 Bounding the Number of Edge Appearances

We now show that, if additionally the number of appearances of each edge in $\mathcal{P}_n$ is bounded by a fixed constant, then ETC can be solved in time polynomial in the size of the input temporal graph.

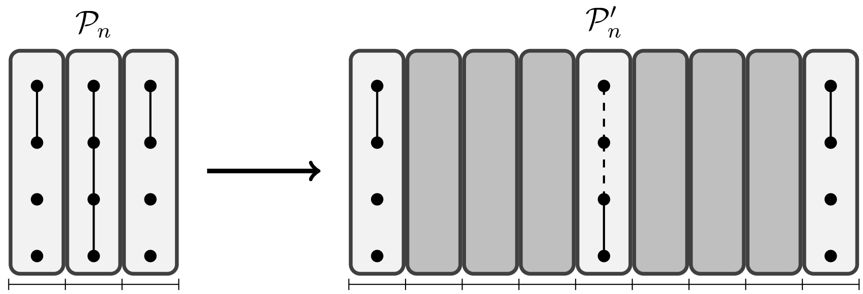

Fig. 1. An instance $\mathcal{P}$ of TM is shown on the left and the stretched graph $\mathcal{P}'_n$ on which we solve Editing to Temporal Cliques is on the right. Non-filler snapshots are shown in white and filler snapshots are grey. Dotted edges show edges that were removed to leave a $(1,5)$-temporal cluster graph (which is also a 5-temporal matching, and corresponds to a 2-temporal matching in $\mathcal{P}$).

Theorem 2. *Let $(\mathcal{P}_n, \Delta_1, \Delta_2, k)$ be any instance of* ETC *where $\mathcal{P}_n \in \mathfrak{F}_n$ and there exists a natural number σ such that $|\mathcal{T}(e)| \leq \sigma$ for any $e \in E(P_n)$. Then,* ETC *on $(\mathcal{P}_n, \Delta_1, \Delta_2, k)$ is solvable in time $\mathcal{O}(T^{4\sigma}\sigma^2 \cdot n^{2\sigma+1})$.*

This theorem is proved using a fairly standard dynamic programming approach, where we go along the underlying path P_n uncovering one vertex in each step. In particular, at the i^{th} vertex we try to extend the current set of templates on the first $i-1$ vertices of the path to an optimal set of templates also including the i^{th} vertex.

5 Completion to Temporal Cluster Graphs

In this section we consider the following variant of ETC, in which we are only allowed to add time-edges.

> Completion to Temporal Cliques (CTC):
> *Input:* A temporal graph $\mathcal{G} = (G, \mathcal{T})$ and positive integers $k, \Delta_1, \Delta_2 \in \mathbb{Z}^+$.
> *Question:* Does there exist a set Π of time-edge additions, of cardinality at most k, such that the modified temporal graph is a (Δ_1, Δ_2)-cluster temporal graph?

The main result of this section is to show that the above problem can be solved in time polynomial in the size of the input temporal graph.

Theorem 3. *There is an algorithm solving* Completion to Temporal Cliques *on any temporal graph $\mathcal{G}$ in time $\mathcal{O}\big(|\mathcal{E}(\mathcal{G})|^3|V|\big)$.*

As observed in [21], the cluster completion problem is also solvable in polynomial time on static graphs. In this case the optimum solution is obtained by transforming each connected component of the input graph into a complete graph. The situation is not quite so simple in temporal graphs, however a similar phenomenon holds with $\varDelta_2$-saturated sets taking the place of connected components; our algorithm relies heavily on the fact (Lemma 5) that we can find these $\varDelta_2$-saturated sets efficiently.

6 A Local Characterisation of Cluster Temporal Graphs

In Section 6.1 we give a characterisation of cluster temporal graphs. We then use this characterisation in Section 6.2 to give an FPT algorithm for ETC.

6.1 The Five-Vertex Characterisation

In this section we show the following characterization of $(\varDelta_1, \varDelta_2)$-cluster temporal graphs in terms of their induced five-vertex subgraphs. The characterisation relies on a fairly natural additional condition which says clusters cannot appear too close to each other in time. We discuss the potential to improve this characterisation in more detail in Section 7.

Theorem 4. *Let $\varDelta_2 > 2\varDelta_1$. Then any temporal graph $\mathcal{G}$ is a $(\varDelta_1, \varDelta_2)$-cluster temporal graph if and only if $\mathcal{G}[S]$ is a $(\varDelta_1, \varDelta_2)$-cluster temporal graph for every set $S \subseteq V(\mathcal{G})$ of at most five vertices.*

One direction of Theorem 4 follows easily from Lemma 8. The other direction is far more challenging. The following lemma illustrates a key idea in the proof of this more challenging direction of Theorem 4: the five vertex condition allows us to 'grow' certain sets of time-edges.

Lemma 11. *Let $\mathcal{G}$ be any temporal graph satisfying the property that $\mathcal{G}[S]$ is a $(\varDelta_1, \varDelta_2)$-cluster temporal graph for every set $S \subseteq V(\mathcal{G})$ of at most five vertices. Let $\mathcal{H}$ be a $\varDelta_1$-temporal clique realising the template $(H, [c, d])$, and $x, y \in H$. Suppose that xy is $\varDelta_1$-dense in the set $[a, b] \supseteq [c, d]$ and let $r_1 = \min(\mathcal{T}(xy) \cap [a, b])$ and $r_2 = \max(\mathcal{T}(xy) \cap [a, b])$. Then there exists a $\varDelta_1$-temporal clique $\mathcal{H}'$ which realises the template $(H, [r_1, r_2])$ where $[r_1, r_2] \supseteq [a + \varDelta_1 - 1, b - \varDelta_1 + 1]$.*

We are now ready to prove the final direction of Theorem 4; full details of this proof, including proofs of the claims, can be found in the full version [4].

Lemma 12. *Let $\mathcal{G}$ be any temporal graph such that $\mathcal{G}[S]$ is a $(\varDelta_1, \varDelta_2)$-cluster temporal graph for every set $S \subseteq V(\mathcal{G})$ of at most five vertices. Then $\mathcal{G}$ is a $(\varDelta_1, \varDelta_2)$-cluster temporal graph.*

Proof. Let $\mathcal{P}_{\mathcal{G}}$ be the partition of $\mathcal{E}(\mathcal{G})$ into $\varDelta_2$-saturated subsets; we know that this partition exists and is unique by Lemma 1. Fix any subset $S \in \mathcal{P}_{\mathcal{G}}$ and denote $L(S) = [s, t]$. We want to show that S forms a $\varDelta_1$-temporal clique in $\mathcal{G}$.

To prove this, we introduce a collection $\varkappa_S = \{S_1, \ldots, S_m\}$ of subsets of S, such that each S_i is a $\varDelta_1$-temporal clique for any $i \in \{1, \ldots, m\}$, $S = \bigcup_{i=1}^{m} S_i$

and for any $S_i \in S$ there does not exist any other Δ_1-temporal clique $K \subseteq S$ such that $S_i \subset K$; we will say that each S_i is a *maximal Δ_1-temporal clique within S*. First of all, we note that this collection exists: in fact, because even the singleton set containing any time-edge is a Δ_1-temporal clique, every time-edge in S belongs to at least one Δ_1-temporal clique. Note that the subsets S_i with $i \in \{1, \ldots, m\}$ are not required to be pairwise disjoint.

We will assume for a contradiction that $m \geq 2$. Because S is Δ_2-saturated, it is not possible that all the Δ_1-temporal cliques contained in $\varkappa_S$ are pairwise Δ_2-independent, since this would imply that S is not Δ_2-indivisible. Thus, as we assume $m \geq 2$, let us consider any distinct $S_i, S_j \in \varkappa_S$ that are not Δ_2-independent. We shall then show that they must both be contained within a larger Δ_1-temporal clique, which itself is contained in S, contradicting maximality. It will then follow that $m = 1$ and thus S is itself a Δ_1-temporal clique, which establishes the theorem.

The next claim shows that if two maximal Δ_1-temporal cliques in S are not Δ_2-independent, then there is a small sub-graph witnessing this non-independence.

Claim 1 *Let $S_i, S_j \in \varkappa_S$ be any pair of Δ_1-temporal cliques which are not Δ_2-independent. Then, there exists a set $W \subseteq V(S_i) \cup V(S_j)$ with $|W| \leq 5$ such that $(S_i \cup S_j)[W]$ is Δ_2-indivisible and contains at least one time-edge from each of S_i and S_j.*

Recall that S_i and S_j are both Δ_2-indivisible as they are Δ_1-temporal cliques. It therefore follows from Claim 1 and Lemma 2 that both $S_i[W] \cup S_j$ and $S_j[W] \cup S_i$ are Δ_2-indivisible. As their intersection is $(S_i \cup S_j)[W] \neq \emptyset$, invoking Lemma 2 once again gives that $S_i \cup S_j$ is Δ_2-indivisible.

Claim 2 *Let S_i and S_j be as above with $L(S_i) = [s_i, t_i]$ and $L(S_j) = [s_j, t_j]$. Then, there exists some $K \subseteq V$ and $s', t' \in \mathbb{Z}^+$ such that $\mathcal{G}$ contains a Δ_1-temporal clique $\mathcal{K}$ realising the template $(K, [s', t'])$ where:*

- *$s' \in [s, \min\{s_i, s_j\} + \Delta_1 - 1]$ and $t' \in [\max\{t_i, t_j\} - \Delta_1 + 1, t]$,*
- *there exist $x, y \in K$ and a time $r_i \in [s', t']$ such that $(xy, r_i) \in S_i$, and*
- *there exist $w, z \in K$ and a time $r_j \in [s', t']$ such that $(wz, r_j) \in S_j$.*

Let us now consider $\mathcal{K}$, the Δ_1-temporal clique of Claim 2. From this we want to extend S_i and S_j to a Δ_1-temporal clique with vertex set $V(S_i) \cup V(S_j)$ and lifetime at least $L(\mathcal{K}) \cup L(S_i) \cup L(S_j) = [\min\{s_i, s_j, s'\}, \max\{t_i, t_j, t'\}]$. We do this in two stages, via the following claims.

Claim 3 *There exist $h_1 \leq h_2$ satisfying*

$$[h_1, h_2] \supseteq [\min\{s_i, s_j, s'\} + \Delta_1 - 1, \max\{t_i, t_j, t'\} - \Delta_1 + 1]$$

such that $(V(S_i) \cup V(S_j), [h_1, h_2])$ is a Δ_1-temporal clique.

Claim 4 *$(V(S_i) \cup V(S_j), [\min\{s_i, s_j, s'\}, \max\{t_i, t_j, t'\}])$ is a Δ_1-temporal clique in $\mathcal{G}$.*

Observe that Claim 4 contradicts the initial assumption that S_i and S_j were maximal in S. Thus the assumption that $m \geq 2$ must be incorrect and thus S consists of a single Δ_1-temporal clique. Because S was a generic set of the partition $\mathcal{P}_{\mathcal{G}}$ of the given temporal graph $\mathcal{G}$ into Δ_2-saturated subsets, then $\mathcal{G}$ must be a (Δ_1, Δ_2)-cluster temporal graph by Lemma 6, giving the result. □

6.2 A Search-Tree Algorithm

Using the characterisation from the previous section, we are now able to prove the following result using a standard bounded search tree technique.

Theorem 5. *Let $\Delta_2 > 2\Delta_1$. Then* ETC *can be solved in time* $(10T)^k \cdot T^3|V|^5$.

7 Conclusions and Open Problems

In this paper we introduced a new temporal variant of the cluster editing problem, ETC, based on a natural interpretation of what it means for a temporal graph to be divisible into "clusters". We showed hardness of this problem even in the presence of strong restrictions on the input, but identified two special cases in which polynomial-time algorithms exist: firstly, if underlying graph is a path and the number of appearances of each edge is bounded by a constant, and secondly if we are only allowed to add (but not delete) time-edges. One natural open question arising from the first of these positive results is whether bounding the number of appearances of each edge can lead to tractability in a wider range of settings: we conjecture that the techniques used here can be generalised to obtain a polynomial-time algorithm when the underlying graph has bounded pathwidth, and it may be that they can be extended even further.

Our main technical contribution was Theorem 4, which gives a characterisation of (Δ_1, Δ_2)-cluster temporal graphs in terms of five vertex subsets, whenever the condition $\Delta_2 > 2\Delta_1$ holds. The assumption that $\Delta_2 > 2\Delta_1$ is needed in two places in the proof of Theorem 4, but we believe that with care it may be possible to modify the proof so that this condition is not required. If it is indeed possible to remove this condition on Δ_1 and Δ_2, then the resulting characterisation would be best possible, as the graph illustrated in Figure 2 demonstrates that no such characterisation involving only four-vertex subsets can exist.

In addition to providing substantial insight into the structure of (Δ_1, Δ_2)-cluster temporal graphs, Theorem 4 also gives rise to a simple search tree algorithm, which is an FPT algorithm parameterised simultaneously by the number k of permitted modifications and the lifetime of the input temporal graph. An interesting direction for further research would be to investigate whether this result can be strengthened: does there exist a polynomial kernel with respect to this dual parameterisation, and is ETC in FPT parameterised by the number of permitted modifications alone?

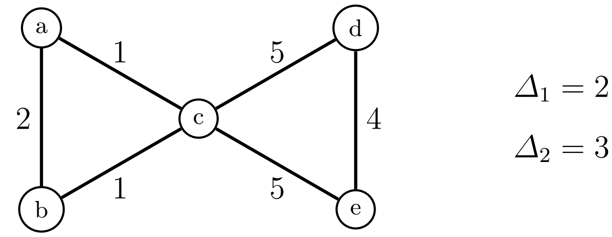

Fig. 2. A temporal graph which is not a $(2,3)$-cluster temporal graph, whose every induced temporal subgraph on at most four vertices is a $(2,3)$-cluster temporal graph.

Acknowledgements

Meeks and Sylvester gratefully acknowledge funding from the Engineering and Physical Sciences Research Council (ESPRC) grant number EP/T004878/1 for this work, while Meeks was also supported by a Royal Society of Edinburgh Personal Research Fellowship (funded by the Scottish Government).

References

1. Faisal N. Abu-Khzam. The multi-parameterized cluster editing problem. In Peter Widmayer, Yinfeng Xu, and Binhai Zhu, editors, *Combinatorial Optimization and Applications - 7th International Conference, COCOA 2013, Chengdu, China, December 12-14, 2013, Proceedings*, volume 8287 of *Lecture Notes in Computer Science*, pages 284–294. Springer, 2013.
2. Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, 2004.
3. Valentin Bartier, Gabriel Bathie, Nicolas Bousquet, Marc Heinrich, Théo Pierron, and Ulysse Prieto. PACE solver description: μsolver - heuristic track. In Petr A. Golovach and Meirav Zehavi, editors, *16th International Symposium on Parameterized and Exact Computation, IPEC 2021, September 8-10, 2021, Lisbon, Portugal*, volume 214 of *LIPIcs*, pages 33:1–33:3. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
4. Cristiano Bocci, Chiara Capresi, Kitty Meeks, and John Sylvester. A new temporal interpretation of cluster editing. *CoRR*, abs/2202.01103, 2022.
5. Sebastian Böcker. A golden ratio parameterized algorithm for cluster editing. *J. Discrete Algorithms*, 16:79–89, 2012.
6. Sebastian Böcker and Jan Baumbach. Cluster editing. In Paola Bonizzoni, Vasco Brattka, and Benedikt Löwe, editors, *The Nature of Computation. Logic, Algorithms, Applications - 9th Conference on Computability in Europe, CiE 2013, Milan, Italy, July 1-5, 2013. Proceedings*, volume 7921 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2013.
7. Sebastian Böcker. A golden ratio parameterized algorithm for cluster editing. *Journal of Discrete Algorithms*, 16:79–89, 2012. Selected papers from the 22nd International Workshop on Combinatorial Algorithms (IWOCA 2011).
8. Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. *Algorithmica*, 64(1):152–169, 2012.
9. Jianer Chen and Jie Meng. A 2k kernel for the cluster editing problem. *J. Comput. Syst. Sci.*, 78(1):211–220, 2012.

10. Jiehua Chen, Hendrik Molter, Manuel Sorge, and Ondřej Suchý. Cluster editing in multi-layer and temporal graphs. In Wen-Lian Hsu, Der-Tsai Lee, and Chung-Shou Liao, editors, *29th International Symposium on Algorithms and Computation, ISAAC 2018, December 16-19, 2018, Jiaoxi, Yilan, Taiwan*, volume 123 of *LIPIcs*, pages 24:1–24:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.
11. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
12. Steven Delvaux and Leon Horsten. On best transitive approximations to simple graphs. *Acta Informatica*, 40(9):637–655, 2004.
13. Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Yngve Villanger. Tight bounds for parameterized complexity of cluster editing with a small number of clusters. *J. Comput. Syst. Sci.*, 80(7):1430–1447, 2014.
14. Anne-Sophie Himmel, Hendrik Molter, Rolf Niedermeier, and Manuel Sorge. Adapting the bron-kerbosch algorithm for enumerating maximal cliques in temporal graphs. *Soc. Netw. Anal. Min.*, 7(1):35:1–35:16, 2017.
15. Christian Komusiewicz and Johannes Uhlmann. Cluster editing with locally bounded modifications. *Discret. Appl. Math.*, 160(15):2259–2270, 2012.
16. Mirko Křivánek and Jaroslav Morávek. *NP* -hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986.
17. Shaohua Li, Marcin Pilipczuk, and Manuel Sorge. Cluster editing parameterized above modification-disjoint P_3-packings. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 49:1–49:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
18. Junjie Luo, Hendrik Molter, André Nichterlein, and Rolf Niedermeier. Parameterized dynamic cluster editing. *Algorithmica*, 83(1):1–44, 2021.
19. Bassel Mannaa. Cluster editing problem for points on the real line: A polynomial time algorithm. *Inf. Process. Lett.*, 110(21):961–965, 2010.
20. George B. Mertzios, Hendrik Molter, Rolf Niedermeier, Viktor Zamaraev, and Philipp Zschoche. Computing maximum matchings in temporal graphs. In Christophe Paul and Markus Bläser, editors, *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*, volume 154 of *LIPIcs*, pages 27:1–27:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
21. Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discret. Appl. Math.*, 144(1-2):173–182, 2004.
22. Tiphaine Viard, Matthieu Latapy, and Clémence Magnien. Computing maximal cliques in link streams. *Theor. Comput. Sci.*, 609:245–252, 2016.
23. Zhao Yang, René Algesheimer, and Claudio J. Tessone. A comparative analysis of community detection algorithms on artificial networks. *Scientific Reports*, (6,30750), 2016.