Link Stream Edition: Sparse Split and Bi-Sparse Split ⋆

Binh-Minh Bui-Xuan [a] Clémence Magnien [a] Pierre Meyer [b]
Thi Ha Duong Phan [c]

[a] *Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6.*
`[buixuan,clemence.magnien]@lip6.fr`

[b] *École Normale Supérieure de Lyon.* `pierre.meyer@ens-lyon.fr`

[c] *Institute of Mathematics, Vietnam Academy of Science and Technology.*
`phanhaduong@math.ac.vn`

Abstract

A link stream is a sequence of timed edges, that are objects of the form (I, uv) where uv is an edge and I a time interval. We give a generic approach to devise fixed parameter tractable algorithms for link stream edition problems and exemplify on two particular problems: sparse split and bi-sparse split edition.

A link stream is sparse split if there exist interval I and vertex subset A such that it consists exactly in $\{(I, uv) : u, v \in A\}$. A link stream is bi-sparse split if there exist consecutive intervals I, J and vertex subsets A, B such that it consists exactly in $\{(I, uv) : u, v \in A \setminus B\} \cup \{(J, uv) : u, v \in B \setminus A\} \cup \{(I \cup J, uv) : u, v \in A \cap B\}$. Problem sparse split (resp. bi-sparse split) link stream edition asks to transform an arbitrary link stream into a sparse split (resp. bi-sparse split) link stream by performing at most some given number of modifications on its timed edges. From a structural point of view, sparse split link streams are equivalent to graphs. We show that a known result on graph edition [F. Hüffner, C. Komusiewicz, and A. Nichterlein, *WADS*, 2015], which is based on a greedy pruning approach, can be directly adapted to give a fixed parameter tractable algorithm for sparse split link stream edition. Bi-sparse split link streams are intrinsically different from graphs. After remarking that the class of bi-sparse split link streams fails an important structural property for the previously mentioned greedy paradigm to operate on them, we devise a loose version of it and obtain a fixed parameter tractable algorithm for bi-sparse split link stream edition. Finally, we revisit and generalise the algorithmic framework to flexibly adapt it to a broad class of edition problems.

Key words: link stream, edge edition, FPT algorithm

⋆ This work encompasses results in [5] while the third author was a Licence stu-

1 Introduction

Problem SparseSplitGraphEdition [7,11,10] consists, given a graph G and an integer k, in transforming G into a single clique plus isolated vertices after at most k edition operations (*i.e.,* edge additions and removals). This problem and its variants have applications in data mining and machine learning [7] and in particular in the correlation clustering problem [1] and the detection of core/periphery structures in networks [3]. This problem has been shown to be NP-complete, with a kernelisation algorithm producing a linear vertex kernel [11,10].

On the other hand, interactions over time, such as phone calls, computer communications, physical proximity between individuals, shopping, and so on, have been studied for a long time and have been captured in the *link stream* framework [12]. Intuitively, a link stream is a sequence of pairs of the form (I, uv) where uv is an edge (in the sense of classical loopless undirected simple graphs) and I a time interval.

A link stream is sparse split if there exist interval I and vertex subset A such that it consists exactly in $\{(I, uv) : u, v \in A\}$. In this paper, we address the problem of edge-editing an arbitrary link stream in order to obtain a sparse split link stream. As already said, the equivalent graph problem has been shown to have implications in data mining and machine learning. Moreover, maximal cliques in link streams have been defined and studied, and have been shown to be also relevant for data mining and shedding intuition on the structure of social interactions [?] and IP traffic [?]. The corresponding SparseSplitLinkStreamEdition problem therefore has applications in data mining and in particular to the problems requiring to take into account the spatio-temporal structure of data, rather than just its structural aspects.

We show that it is possible to generalise the greedy paradigm developed in [10] for link streams. As a byproduct we give a linear vertex kernelization algorithm for a constrained variant of sparse split link stream edition. This result further implies a fixed parameter tractable algorithm for SparseSplitLinkStreamEdition. However, note from a structural point of view

---

dent of *École Normale Supérieure de Lyon*. For financial support, we are grateful to: *Centre National de la Recherche Scientifique*, project INS2I.GraphGPU; *Thales Communications & Security*, project TCS.DJ.2015-432; *Agence Nationale de la Recherche Technique*, project 2016.0097; *European Commission* H2020 FET-PROACT 2016-2017 program, grant 732942 ODYCCEUS; *Agence Nationale de la Recherche*, grant ANR-15-CE38-0001 AlgoDiv; *Ile-de-France Region* FUI21 program, grant 16010629 iTRAC; *Vietnam National Foundation for Science and Technology Development*, NAFOSTED program, grant 101.99-2016.16; *Vietnam Institute for Advanced Study in Mathematics*.

that sparse split link streams are equivalent to classical graphs. Our generalization therefore very closely resembles the solution developed in [10] for graphs. We accordingly raise the question of editing link streams into a target class of link streams that is intrinsically different from a graph.

A simple extension of sparse split link streams to a class of non graph-equivalent link streams lies in the notion of bi-sparse split link streams, *i.e.,* a stream for which there exist consecutive intervals I, J and vertex subsets A, B such that the link stream consists exactly in $\{(I, uv) : u, v \in A \setminus B\} \cup \{(J, uv) : u, v \in B \setminus A\} \cup \{(I \cup J, uv) : u, v \in A \cap B\}$. Unfortunately, the greedy approach cannot be directly extended to bi-sparse split link stream edition. We cope with the inconvenience by devising new kernelization procedures which result in similar performance as with sparse split edition. On the way to do this we revisit and generalize all our techniques so that they can be flexibly adapted to fit into many link stream edition situations. We exemplify our generic properties by confronting them to a list of edition problems in Fig. 1 at the end of the paper.

Our manuscript is organised as follows. Section 2 introduces generic properties of link streams and builds the basis for the rest of the paper. In particular, we explicit a sufficient condition to be verified on a given class of link streams so that some greedy algorithms can become kernelization algorithms for the edition problem associated to that class. Then, in Section 3, we show that the class of sparse split link streams satisfies this sufficient condition, which, as a direct consequence, provides a kernelization algorithm for sparse split edition. In Section 4, we extend the result to bi-sparse split link streams. We then close the paper by summing up our algorithmic ideas and confronting them to a list of edition problems. We also give in the same section concluding remarks and open questions for further investigation.

2 A general framework for Link Stream Edition

Graphs in this paper are simple, undirected and loopless. When, and only when, two vertices u and v are distinct, we denote an edge between u and v indifferently by $uv = vu = \{u, v\}$. We denote the set of all such pairs by $V \otimes V$. A *link stream* L is a triple $L = (T, V, E)$ where T is an interval, V a finite set of vertices, and $E \subseteq T \times V \otimes V$. In this paper we focus on the case where links exist for a continuous duration of time, *i.e.,* there exist intervals $[a, b]$ and vertices u, v such that $\forall t \in [a, b], (t, uv) \in E$. For all $uv \in V \otimes V$, we call $\mathcal{I}_{uv}$ the set of maximal such intervals. In the remaining, we consider only these intervals and we suppose that E is given in the form of interval edges: $E = \{(I, uv) : uv \in V \otimes V \wedge I \in \mathcal{I}_{uv}\}$. Moreover, we suppose that $\mathcal{I}_{uv}$ is finite for every edge uv, and, consequently that E is finite. Note that, by

Fig. 1. Addition operations: (left) addition; (center) extension; (right) merging.

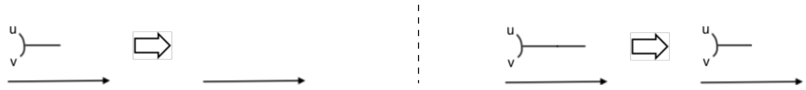

Fig. 2. Deletion operations: (left) deletion; (right) shortening.

construction, edges defined in this way satisfy the *non-overlapping property*: $(I, uv) \in E \wedge (J, uv) \in E \wedge I \neq J \Rightarrow I \cap J = \emptyset$. The size of link stream L is denoted by $|L| = n + m$ where $n = |V|$ and $m = |E|$, with E given in interval edges form. We also denote $T(L) = T$, $V(L) = V$ and $E(L) = E$. The elements of E are called *timed edges*.

Link stream L is *sparse split* if there exist interval I and vertex subset A such that $E(L) = \{(I, uv) : uv \in A \otimes A\}$. Link stream L is *bi-sparse split* if there exist consecutive time intervals I, J, and vertex subsets A, B such that $E(L) = \{(I, uv) : uv \in (A \setminus B) \otimes (A \setminus B)\} \cup \{(J, uv) : uv \in (B \setminus A) \otimes (B \setminus A)\} \cup \{(I \cup J, uv) : uv \in (A \cap B) \otimes (A \cap B)\}$. Note that being sparse split implies being bi-sparse split (where $B = \emptyset$).

Given a link stream L, an *edition operation* over L is either an addition operation or a deletion operation, which are defined as follows. An *addition operation* over L transforms L into L', where $T(L') = T(L)$, $V(L') = V(L)$, $E(L')$ satisfies the non-overlapping property, and $E(L')$ is modified in one of the three following ways (cf. Figure 1):

- add (I, uv): there are interval I and vertices u, v such that $E(L') = E(L) \cup \{(I, uv)\}$;
- extend from $(J, uv) \in E(L)$ to (I, uv): there are intervals I, J and vertices u, v such that $J \subseteq I$, $I \setminus J$ is a non-empty interval, and $E(L') = \big(E(L) \setminus \{(J, uv)\}\big) \cup \{(I, uv)\}$;
- merge $(J, uv) \in E(L)$ and $(K, uv) \in E(L)$ into (I, uv): there are intervals I, J, K and vertices u, v such that $J \subseteq I$, $K \subseteq I$, $J \cap K = \emptyset$, $\min I = \min J$, $\max I = \max K$, and $E(L') = \big(E(L) \setminus \{(J, uv), (K, uv)\}\big) \cup \{(I, uv)\}$.

A *deletion operation* over L transforms L into L', where $T(L') = T(L)$, $V(L') = V(L)$, and $E(L')$ is modified in one of the two following ways (cf. Figure 1):

- delete $(I, uv) \in E(L)$: there are interval I and vertices u, v such that $E(L') = E(L) \setminus \{(I, uv)\}$;
- shorten $(I, uv) \in E(L)$ downto (J, uv): there are intervals I, J and vertices u, v such that $J \subseteq I$, $I \setminus J$ is a non-empty interval, and $E(L') = \big(E(L) \setminus$

$\{(I, uv)\}\big) \cup \{(J, uv)\}$.

Notice that there exists a third deletion operation, that is the opposite of merge. However, we do not address it in this paper, as it is not relevant with respect to the problems we consider.

We omit the formal argument for the following property's correctness, which is straightforward.

Property 1 *On input two link streams L and L' it is possible to compute in polynomial time the minimum number of edition operations transforming L into L'. It is also possible to compute such a smallest series of edition operations transforming L into L' in polynomial time.*

Proof: Computing $E(L)\Delta E(L')$ is polynomial. □

Given a class $\mathcal{L}$ of link streams, we consider the following decision problem.

$\mathcal{L}$-LinkStreamEdition ($\mathcal{L}$-LSE):

Input: L a link stream and $k \in \mathbb{N}$ an integer.
Question: Is there a series of at most k consecutive edition operations transforming L into L' such that $L' \in \mathcal{L}$?

By SS-LSE and BiSS-LSE, we refer to problem $\mathcal{L}$-LinkStreamEdition when $\mathcal{L}$ is the class of sparse split link streams and when $\mathcal{L}$ is the class of bi-sparse split link streams, respectively.

A sparse split graph consists in a clique and isolated vertices. The sparse split graph edition problem asks to transform an input graph into a sparse split graph by performing at most a given number of edge graph editions. An edge graph edition is either the addition of a non-existing edge or the removal of an existing edge. Basically, it follows from the NP-completeness of sparse split graph edition [11] that SS-LSE and BiSS-LSE are NP-complete. Formally however, problem SS-LSE does not exactly encompass sparse split graph edition: although any target sparse split link stream is structurally equivalent to a sparse split graph (i.e. at the end of the edition, we obtain essentially the same thing), an arbitrary input link stream to the link stream edition problem might have nothing to do with any input of the graph edition problem. We consequently need to prove the following property. However, the proof is very procedural.

Property 2 *Both* SparseSplitLinkStreamEdition *(SS-LSE) and* BiSparseSplitLinkStreamEdition *(BiSS-LSE) are NP-complete.*

Proof: SS-LSE and BiSS-LSE are in NP because the membership testing problem is trivially polynomial for both classes of sparse split and bi-sparse split link streams. The only thing we will prove is the NP-completeness of SS-LSE, because it implies that of BiSS-LSE. We do this using the NP-completeness of sparse split graph edition [11], with the following direct reduction. Let (G, k) be an instance of sparse split graph edition. We first compute link stream L based on graph G as follows: $V(L) = V(G)$, $T(L) = [0, 0]$, and $E(L) = \{(T(L), uv) : uv \in E(G)\}$. We then return instance (L, k) of SS-LSE. Here, the size of L clearly satisfies $|L| = O(|G|)$ and L can trivially be computed from G in polynomial time. We now need to prove that the answer to SS-LSE on (L, k) is positive if and only if the answer to sparse split graph edition on (G, k) is positive. Since L is essentially defined based on G, it is straightforward to check that the condition is sufficient, namely that: if the answer to sparse split graph edition on (G, k) is positive, then the answer to SS-LSE on (L, k) is positive.

In order to prove that the condition is necessary, suppose that the answer to SS-LSE on input (L, k) is positive. Then, there exists a sparse split link stream which can be obtained after at most k edition operations on L. Let L' be a sparse split link stream that has been obtained from L after a minimum number of edition operations (hence, this number must be at most k). By definition of edition operations on link streams, we have $T(L') = T(L)$ and $V(L') = V(L)$. Now, the main property here is that: since $|T(L)| = 1 = |T(L')|$, every edition operation transforming L into L' can only be of the form (add, I, uv) or (delete, I, uv); there cannot be neither merging, shortening nor extension. Since L' is a sparse split link stream, there exists $A \subseteq V(L)$ such that $E(L') = \{(T(L), uv) : uv \in A \otimes A\}$. Define now G' such that $V(G') = V(L)$ and $E(G') = A \otimes A$. Note that, as with G', graph G also satisfies $V(G) = V(L)$. Then, by performing on G similar edition operations as on L, we can obtain G' from G after the same number of graph edition operations as the number of operations needed to transform L into L'. Hence, the answer to sparse split graph edition on input (G, k) is positive. $\square$

We consider the parameterized version of $\mathcal{L}$-LSE where integer k becomes the parameter and no longer belongs to the input, in the sense of [8,9,6].

PARAMETERIZED-$\mathcal{L}$-LINKSTREAMEDITION:

INPUT: L a link stream.
PARAMETER: $k \in \mathbb{N}$ an integer.
QUESTION: Is there a series of at most k consecutive edition operations transforming L into L' such that $L' \in \mathcal{L}$?

The parameterized version of LSE is *fixed parameter tractable (FPT)* if there exist a computable function $f : \mathbb{N} \to \mathbb{N}$, a constant $c \in \mathbb{N}$ and an algorithm $\mathcal{A}$ such that on any input L with parameter k, algorithm $\mathcal{A}$ correctly answers problem LSE in $O(f(k) \times |L|^c)$ worst case time. In the remaining of this section, we give generic ideas for designing a FPT algorithm $\mathcal{A}$ for LSE. The overall goal of the paper is to prove that:

Main Result 1
The parameterized versions of SPARSESPLIT-LINKSTREAMEDITION *and* BISPARSESPLIT-LINKSTREAMEDITION *are fixed parameter tractable.*

Proof: follows from Property 3 below and subsequent Main Result 2. □

2.1 Graph likeness

A link stream L is said to be *graph-like* if there are time instants $\alpha \leq \omega$ of $T(L)$ such that every timed edge $(I, uv) \in E(L)$ satisfies $I \subseteq [\alpha, \omega]$, plus the fact that either $\min I = \alpha$ or $\max I = \omega$ (or both). We stress that both classes of sparse split link streams and bi-sparse split link streams are graph-like. Graph-likeness is very useful to reduce a link stream edition instance to (mostly) a graph edition instance. This is done through a boxed version of link stream edition problems.

When a timed edge (I, uv) satisfies $I \subseteq [\alpha, \omega]$, plus the fact that either $\min I = \alpha$ or $\max I = \omega$ (or both), we say that (I, uv) is (α, ω)*-graph like.* A link stream which only contains timed edges that are boxed by α and ω is said to be (α, ω)*-graph like.*

(α, ω)-$\mathcal{L}$-LINKSTREAMEDITION:

INPUT: L a link stream and $k \in \mathbb{N}$ an integer.
QUESTION: Is there a series of at most k consecutive edition operations transforming L into $L' \in \mathcal{L}$ that is (α, ω)-graph like?

By parameterized version of (α, ω)-$\mathcal{L}$-LSE, we refer to the one where integer k becomes the parameter and no longer belongs to the input, as above.

Property 3 *Let $\mathcal{L}$ be a class of graph-like link streams. If the parameterized version of (α, ω)-$\mathcal{L}$-LSE is fixed parameter tractable for all $\alpha, \omega \in T$, then the parameterized version of $\mathcal{L}$-LSE is fixed parameter tractable.*

Proof: On input (L, k) of $\mathcal{L}$-LSE we are asked to transform arbitrarily given link stream L into a link stream of $\mathcal{L}$ (using at most k operations). Let $L' \in \mathcal{L}$ be any target link stream into which we wish to transform L. Since $L' \in \mathcal{L}$ is graph-like, there exist α' and ω' such that any timed edge of L' is (α', ω')-graph like. Although we do not know in advance the values of α' and ω', we can find them by an exhaustive search because there are only a polynomial number of possibilities for them. Indeed, we have:

- Let $Ext(L) = \{t \in T | \exists (I, uv) \in E(L) s.t. t = \min I \vee t = \max I\}$.

- Assume that $\alpha' \notin Ext(L)$, we will prove that there exists another solution that can be reach from L with at most k operations such that the corresponding α value belongs to $Ext(L)$.
- Suppose there exists $b_0 = \max\{t \in T | \exists (I, uv) \in E(L) \wedge t = \min I \wedge t < \alpha'\}$. Then, for every timed edge $l = ([\alpha', e'], uv) \in L'$, there are only three edition operations that can create l: add, extend or shorten (and not merge nor delete). If l was the result of an addition operation, then we can instead add $([b_0, e'], uv)$ for the same cost. If l was the result of an extension operation, then we can instead extend further to $([b_0, e'], uv)$ for the same cost. If l was the result of an shortening operation, by definition of b_0, before the shortening, l began at either b_0 or strictly before b_0. Then, we can either leave it unchanged, or shorten it less to obtain $([b_0, e'], uv)$ for the same cost.
- Else, b_0 does not exists, we define $b_1 = \min\{t \in T | \exists (I, uv) \in E(L) \wedge t = \min I \wedge t > \alpha'\}$, and process in a symmetric manner.
- Finally, we process similarly for ω'.

Then, we can w.l.o.g. assume that $\alpha', \omega' \in Ext(L)$. Accordingly, in order to obtain an FPT algorithm for $\mathcal{L}$-LSE it is sufficient to loop through all the $O(|E(L)|^2)$ possible values of $\alpha', \omega' \in Ext(L)$ and solve (α', ω')-$\mathcal{L}$-LSE using any FPT algorithm for the latter problem. □

The parameterized version of (α, ω)-$\mathcal{L}$-LSE has a *kernel* if there exist a computable function $f : \mathbb{N} \to \mathbb{N}$ and a polynomial time algorithm $\mathcal{A}$ which takes as input an instance (L, k) of (α, ω)-$\mathcal{L}$-LSE and produces an instance (L', k') such that: $k' \leq k$; $|L'| \leq f(k)$; and (L', k') yields a positive answer for (α, ω)-$\mathcal{L}$-LSE if and only if (L, k) yields a positive answer for (α, ω)-$\mathcal{L}$-LSE. In this case, algorithm $\mathcal{A}$ is called a *kernelization algorithm* for (α, ω)-$\mathcal{L}$-LSE. Having a kernel implies being fixed parameter tractable [8].

In Sections 3 and 4, we depict the details of our approach and will prove the following result. When combined with Property 3, they imply Main Result 1 as a direct consequence.

Main Result 2
The parameterized version of (α, ω)-SparseSplit-LinkStreamEdition *has a linear vertex kernel.*

Proof: follows from Theorem 2 (Section 3) and Theorem 3 (Section 4). □

We say that an algorithm is *safe* with respect to (α, ω)-$\mathcal{L}$-LSE if on input (L, k) it produces in polynomial time an instance (L', k') such that $k' \leq k$ and that (L', k') yields a positive answer for (α, ω)-$\mathcal{L}$-LSE if and only if (L, k) yields a positive answer for (α, ω)-$\mathcal{L}$-LSE. One popular way to obtain kernelization algorithms is the successive applications of safe algorithms, in such a way that it leads to an instance of sufficiently small size, for instance polynomial or linear in the value of the parameter.

As artificial terminal cases, let NEGATIVE be a link stream not belonging to $\mathcal{L}$, and POSITIVE be a link stream belonging to $\mathcal{L}$. This way, $(\text{NEGATIVE}, 0)$ and $(\text{POSITIVE}, 0)$ are trivial negative and positive instances of (α, ω)-$\mathcal{L}$-LSE, respectively.

Procedure 1 On input instance (L, k) of (α, ω)-$\mathcal{L}$-LSE, remove from L every timed edge (I, uv) with

$$\{\min I, \max I\} \cap \{\alpha, \omega\} = \emptyset,$$

and decrease k with the number of such timed edges.

Lemma 1 *Procedure 1 is safe with respect to* (α, ω)*-*$\mathcal{L}$*-LSE.*

Proof: We proceed by exhaustive case analysis. Let (L', k') be the output of Procedure 1. If (L', k') yields a positive answer to (α, ω)-$\mathcal{L}$-LSE, it is then trivial to prove that (L, k) yields a positive answer to (α, ω)-$\mathcal{L}$-LSE: Procedure 1 minimally transforms L into L'; from that point, we use k' operations to transform L' into some $L'' \in \mathcal{L}$ with all the desired properties; the total number of operations cannot exceed k by minimality of the transformation of L into L'; hence, with at most k operations, we can also transform L into L'', which is known to own all the desired properties.

Suppose now that (L, k) yields a positive answer to (α, ω)-$\mathcal{L}$-LSE and let $L'' \in \mathcal{L}$ be (α, ω)-graph like which can be obtained from L after k editions. In what follows we will try to permute the k edition operations transforming L into L'' in such a way that they will first transform L into L', and only then, they will complete the transformation by transforming L' into L''. Let (I, uv) be a timed edge of L such that $\{\min I, \max I\} \cap \{\alpha, \omega\} = \emptyset$. We distinguish two cases: either $([\alpha, \omega], uv)$ belongs to L'' or not. In the latter case, (I, uv) needs to be removed from L on the way to L''; we can then do that removal first among the edition operations transforming L to L''. In the former case, (I, uv) needs to be transformed into $([\alpha, \omega], uv)$ and for this transformation we need to use at least two edition operations in any case (extend twice; extend then shorten; shorten then extend; add then remove; remove then add; and so on). Hence, we can replace the two corresponding operations with: first remove (I, uv) then add $([\alpha, \omega], uv)$; the first of these two operations can be done at the beginning, and the second at the end of the operations transforming L into L''. We have just argued that L can first be transformed into L', then L' can be transformed into L'', using in total the same number, which is at most k, of edition operations. Moreover, the number of timed edges of L of the form (I, uv) where $\{\min I, \max I\} \cap \{\alpha, \omega\} = \emptyset$ is exactly the minimum number of operations needed to transform L into L'. Besides, this number corresponds to $k - k'$, by definition of Procedure 1. We just have proved that (L', k') yields a positive answer to (α, ω)-$\mathcal{L}$-LSE. □

In the sequel we revisit and generalize the notion of neighbourhood reconstruction in Ref. [10] so that they can apply to a broad scope of edition problems. We employ the following generic formalism for a later use in Procedure 2. Its essence can be seen as a formalization of the greedy approach on link streams. A vertex u of a link stream L is said to be *isolated* if every timed edge (I, vw) of $E(L)$ satisfies $u \neq v$ and $u \neq w$.

Definition 1 (σ-greedy editable link streams) Given an integer σ, a class $\mathcal{L}$ of link streams is said to be *σ-greedy (α, ω)-editable* if there exists an algorithm $\mathcal{A}$, called σ-greedy (α, ω)-edition algorithm for $\mathcal{L}$, which on input (L, u) with L being an arbitrary link stream and u a vertex of $V(L)$, produces in polynomial time a set of link streams $\mathcal{A}(L, u)$ satisfying the following three conditions.

- $\mathcal{A}(L, u) \subseteq \mathcal{L}$.
- every link stream in $\mathcal{A}(L, u)$ is (α, ω)-graph like.
- for every integer k, we have the following property: **if** $\Big($there exists a series of k edition operations, among which at most σ involving u, transforming L into $L' \in \mathcal{L}$ in such a way that L' is (α, ω)-graph like and that u is not isolated in $L'\Big)$ **then** $\Big($there exists a series of k edition operations, among which σ involving u, transforming L into $L' \in \mathcal{A}(L, u)$ in such a way that u is not isolated in $L'\Big)$.

Essentially, $\mathcal{A}(L, u)$ would be a class of link streams that can be obtained from L after local modifications around vertex u. For that local behaviour of $\mathcal{A}$ we qualify it as greedy. We note furthermore that $|\mathcal{A}(L, u)|$ is polynomial in the size of L because $\mathcal{A}$ terminates in polynomial time. If a link stream edition problem admits such an algorithm $\mathcal{A}$, we show in the sequel a straightforward safe procedure in order to kernelize the problem. One illustration is given in Section 3 with Property 4.

Given a σ-greedy (α, ω)-editable class $\mathcal{L}$ of link streams, we show in Lemma 2 that the following algorithm is safe.

Procedure 2 On input instance (L, k) of (α, ω)-$\mathcal{L}$-LSE, where $\mathcal{L}$ is σ-greedy (α, ω)-editable, let $\mathcal{A}$ be a σ-greedy (α, ω)-edition algorithm for $\mathcal{L}$, we process as follows. For every vertex $u \in V(L)$, compute $\mathcal{A}(L, u)$ by calling $\mathcal{A}$ on input (L, u). Then, for every link stream $L' \in \mathcal{A}(L, u)$, if there exists a series of k edition operations transforming L into L', then return $(\textsc{Positive}, 0)$. After all looping processes, if we still have not returned anything yet, then return original input (L, k).

Lemma 2 *For every σ-greedy (α, ω)-editable class $\mathcal{L}$ of link streams, Procedure 2 is safe with respect to (α, ω)-$\mathcal{L}$-LSE.*

Proof: On input (L, k) Procedure 2 returns either $(\textsc{Positive}, 0)$ or (L, k). Hence, the output value for the parameter is at most k. As for complexity issues, after producing $\mathcal{A}(L, u)$, which is in polynomial time by definition of $\mathcal{A}$, Procedure 2 does in total $\sum_{u \in V(L)} |\mathcal{A}(L, u)|$ checks whether there exists a series of k edition operations transforming L into some given L'. Because both L and L' in each case are known, each checking process can be done in polynomial time, cf. Property 1. By definition of greedy edition, $\sum_{u \in V(L)} |\mathcal{A}(L, u)|$ is polynomial in the size of L. Therefore, Procedure 2 terminates in polynomial time.

Now, suppose that (L, k) yields a positive answer to (α, ω)-$\mathcal{L}$-LSE. Since Procedure 2 only returns either $(\textsc{Positive}, 0)$ or (L, k), which both yield positive answer to (α, ω)-$\mathcal{L}$-LSE, we have indeed that the input instance and the outputted instance of Procedure 2 yield the same answer to (α, ω)-$\mathcal{L}$-LSE.

On the other hand, suppose that (L, k) yields a negative answer to (α, ω)-$\mathcal{L}$-LSE. Here, if Procedure 2 returns (L, k) then everything is fine and there is no more thing to prove. Let us suppose by contradiction that Procedure 2 returns $(\textsc{Positive}, 0)$. This return instruction must have been done when examining some $u \in V(L)$ and some $L' \in \mathcal{A}(L, u)$. By definition of the return instruction in Procedure 2, L' can be obtained from L after k edition operations. By definition of $\mathcal{A}(L, u)$, L' is (α, ω)-graph like. Furthermore, since $\mathcal{A}(L, u) \subseteq \mathcal{L}$, we have $L' \in \mathcal{L}$. But then, L' would be a certificate that (L, k) yields a positive answer to (α, ω)-$\mathcal{L}$-LSE. Contradiction. □

The main advantage of greedy edition lies rather in the following property.

Lemma 3 *Let $\mathcal{L}$ be a σ-greedy (α, ω)-editable class of link streams. Then, on input (L, k) Procedure 2 outputs (L, k) (and not $(\textsc{Positive}, 0)$) if and only if the following property holds. If $L' \in \mathcal{L}$ is (α, ω)-graph like, and L' can be obtained from L after at most k edition operations, then, for every vertex u of $V(L)$, either u is an isolated vertex in L' or there must be at least $\sigma + 1$ edition operations involving u on the way from L to L'.*

Proof: By definition of Procedure 2, when the output of the procedure is (L, k) (and not $(\textsc{Positive}, 0)$), the following property must fail: $\big($there exists a series of k edition operations, among which σ involving u, transforming L into $L' \in \mathcal{A}(L, u)$ in such a way that u is not isolated in $L'\big)$. Then, by definition of σ-greedy (α, ω)-edition algorithm $\mathcal{A}$, the following property must fail: $\big($there exists a series of k edition operations, among which σ involving u, transforming L into $L' \in \mathcal{L}$ in such a way that L' is (α, ω)-graph like and that u is not isolated in $L'\big)$. Now, let $L' \in \mathcal{L}$ be (α, ω)-graph like, and L' can be obtained from L after at most k edition operations. According to the failure of the previous property, for every vertex u of $V(L)$, the only possibilities left for u are: either u is an isolated vertex in L'; or there must be at least $\sigma + 1$ edition operations involving u on the way from L to L'. □

2.3 Hereditary link streams and isolation

A class $\mathcal{L}$ of link streams is *hereditary* if it is closed under vertex deletion and under addition of isolated vertex. Obviously, both classes of sparse split link streams and of bi-sparse split link streams are hereditary. We can now conclude by the following formalism. The (open) *neighbourhood* of u in L is defined as $N_L(u) = \{v \in V(L) : \exists (I, uv) \in E(L)\}$. Besides, we define $\mathbb{0}^L_{uv}$ as the minimum number of edition operations required to remove all timed edges between u and v from L, that is, to transform L into L' where $V(L) = V(L')$, $T(L) = T(L')$, and $E(L') = E(L) \setminus \{(I, uv) : I \subseteq T(L)\}$. We also define $\mathbb{0}^L_u = \sum_{v \neq u} \mathbb{0}^L_{uv}$.

Procedure 3 On input instance (L, k) of (α, ω)-$\mathcal{L}$-LSE, initialize (L', k') with $(L', k') \leftarrow (L, k)$. For every vertex $u \in V(L')$ such that $\mathbb{0}^{L'}_u \leq \sigma$, isolate then remove u, that is, perform $(L', k') \leftarrow (L'', k' - \mathbb{0}^{L'}_u)$, where $V(L'') = V(L') \setminus \{u\}$, $T(L'') = T(L')$ and $E(L'') = E(L') \setminus \{(I, uv) : I \subseteq T(L') \wedge v \in V(L')\}$. At the end of the process, return (L', k').

Lemma 4 *Let $\mathcal{L}$ be a hereditary and σ-greedy (α, ω)-editable class of graph-like link streams. Let (L_0, k_0) be an arbitrary instance of (α, ω)-$\mathcal{L}$-LSE. Apply Procedure 2 and let (L, k) be the output. Then, on input instance (L, k), Procedure 3 is safe with respect to (α, ω)-$\mathcal{L}$-LSE.*

Proof: Let (L', k') be the output of Procedure 3. Clearly, Procedure 3 runs in polynomial time and can only decrease the value of the parameter, that is, we have $k' \leq k$.

Suppose that (L', k') yields a positive answer to (α, ω)-$\mathcal{L}$-LSE, that is, there exists $L'' \in \mathcal{L}$ which is (α, ω)-graph like and which can be obtained from L' after at most k' edition operations. Now, for every $u \in V(L)$ such that $\mathbb{0}^L_u \leq \sigma$, add a new isolated vertex to L'', and obtain L'''. By heredity, $L''' \in \mathcal{L}$. Furthermore, it is clear that L''' is (α, ω)-graph like. Here, the important point is that we have constructed L''' in such a way that: as far as L is concerned, and after we isolate every vertex u in L such that $\mathbb{0}^L_u \leq \sigma$, then we can mimic the k' operations transforming L' into L'' in order to transform the remaining of L into L''' with k' similar operations. Hence, we have that L''' can be obtained from L after isolating every u with $\mathbb{0}^L_u \leq \sigma$, plus k' edition operations. Finally, by definition of (L', k') in Procedure 3, the total number of these edition operations is exactly k. Hence, in this case the input instance

(L, k) of Procedure 3 yields a positive answer to (α, ω)-$\mathcal{L}$-LSE.

Suppose now that (L, k) yields a positive answer to (α, ω)-$\mathcal{L}$-LSE, that is, there exists $L'' \in \mathcal{L}$ which is (α, ω)-graph like and which can be obtained from L after at most k edition operations. Then, by Lemma 3, for every vertex u of $V(L)$, either u is an isolated vertex in L'' or there must be at least $\sigma + 1$ edition operations involving u on the way from L to L''. Let $u \in V(L)$ such that $\mathbb{0}_u^L \leq \sigma$ (which implies that u is isolated in L'). If u is an isolated vertex in L'' then the minimum number of edition operations to isolate u from input instance L, which is $\mathbb{0}_u^L$, is indeed what we have use in Procedure 3 as long as u is concerned. If u is not an isolated vertex in L'' then we know there are at least $\sigma+1$ edition operations required in order to transform the neighbourhood of u in L into the neighbourhood of u in L''. However, let us consider L''', the link stream that is obtained by removing u from L''. Firstly, L''' is (α, ω)-graph like. Secondly, L''' belongs to $\mathcal{L}$ because of heredity. Lastly, as long as u is concerned, we can obtain L''' from L' by at most the same edition operations as obtaining L'' from L. Basically, the fact that L can be transformed into L'' by k edition operations guarantees that L' can be transformed into L''' by at most k' edition operations. Therefore, the outputted instance (L', k') of Procedure 3 yields a positive answer to (α, ω)-$\mathcal{L}$-LSE. □

Procedure 4 On input instance (L, k) of (α, ω)-$\mathcal{L}$-LSE, if $|V(L)| > \frac{2\times k}{\sigma+1}$ then we output (Negative, 0). Otherwise, we output (L, k).

Lemma 5 *Let $\mathcal{L}$ be a hereditary and σ-greedy (α, ω)-editable class of graph-like link streams. Let (L_0, k_0) be an arbitrary instance of (α, ω)-$\mathcal{L}$-LSE. Iteratively apply (Procedure 2 then Procedure 3) until the output is constant, and let (L, k) be that output. Then, on input instance (L, k), Procedure 4 is safe with respect to (α, ω)-$\mathcal{L}$-LSE.*

Proof: Let (L', k') be the output of Procedure 4. Clearly, Procedure 4 runs in polynomial time and can only decrease the value of the parameter, that is, we have $k' \leq k$.

First, remark that Procedure 4 outputs (Negative, 0) if and only if the value of (L', k') yields a negative answer to (α, ω)-$\mathcal{L}$-LSE: indeed, L' will then only contain vertices where, for each of them, we must spend at least $\sigma + 1$ edition operations involving the vertex; but also L' will then have a lot of vertices, namely $|V(L')| > \frac{2\times k'}{\sigma+1}$. However, with only k' edition operations, since each can only involve two vertices, we cannot involve more than $\frac{2\times k'}{\sigma+1}$ vertices when each of them needs to be involved at least $\sigma+1$ times. Hence, when Procedure 4 outputs (Negative, 0), the value of (L', k') yields a negative answer to (α, ω)-$\mathcal{L}$LSE. It is consequently sufficient to prove that (L, k) and (L', k') yield the same answer to (α, ω)-$\mathcal{L}$-LSE. □

Theorem 1 *Let $\mathcal{L}$ be a hereditary and σ-greedy (α, ω)-editable class of graph-like link streams. First apply Procedure 1. Then, iteratively apply (Procedure 2 then Procedure 3) until the output is constant. Finally apply Procedure 4. This process results in a kernelization algorithm for (α, ω)-$\mathcal{L}$-LSE. Moreover, the number of vertices in the kernel is linear in the parameter.*

Proof: By Lemmas 1, 2, 4, and 5, the procedure given in the statement is safe. The only thing left to prove is that the size of the output (L', k') of the procedure is bounded by a function of k'. This is clearly the case if the output of Procedure 2 is $(\textsc{Positive}, 0)$, or the output of Procedure 4 is $(\textsc{Negative}, 0)$.

Otherwise, let (L, k) be the output obtained at the end of the iteration of (Procedure 2 and Procedure 3). Note that (L, k) is also the input of Procedure 4. Since Procedure 4 does not return $(\textsc{Negative}, 0)$, we have that $|V(L)| \leq \frac{2\times k}{\sigma+1}$, and that $(L, k) = (L', k')$.

Finally, after Procedure 1, the maximum number of possible edges is obtained if there are two timed edges for each pair of vertices (u, v): one involving α and one involving ω. Therefore, $|E(L')| = O(|V(L')|^2)$. □

3 Sparse Split Link Stream Edition

In order to show that sparse split link stream edition is fixed parameter tractable, cf. Main Results 1 and 2, the only thing left to show is:

Property 4 *The class $\mathcal{L}$ of sparse split link streams is a hereditary and σ-greedy (α, ω)-editable class of graph-like link streams.*

Proof: Graph-likeness and heredity are clearly true. The only thing to prove is about greedy edition. For this we design algorithm $\mathcal{A}$, which we prove to be a σ-greedy (α, ω)-edition algorithm. On input (L, u) with L being an arbitrary link stream and u a vertex of L, we define the set $\mathcal{A}(L, u)$ of sparse split link streams as follows. Let us consider $N_L(u)$. With at most σ edition operations involving u, there are only a polynomial number of possibilities of transforming $N_L(u)$ into a neighbourhood of u with exclusively timed edges of the form $([\alpha, \omega], uv)$ for some $v \in V(L)$. We denote all these possibilities by $N_\sigma(u) = \{N \subseteq V(L) : N$ is the neighbourhood of u with exclusively timed edges of the form $([\alpha, \omega], uv)$ after at most σ edition operations involving u from original link stream $L\}$. We note $N_\sigma[u] = \{N \cup \{u\} : N \in N_\sigma(u)\}$. We then define
$\mathcal{A}(L, u) = \{L' : V(L') = V(L) \wedge T(L') = T(L) \wedge E(L') = \{([\alpha, \omega], vw) : vw \in A \otimes A\} \wedge A \in N_\sigma[u]\}$.

We have that:

- $\mathcal{A}(L, u)$ is composed of a polynomial number [1] of sparse split link streams
- $\mathcal{A}(L, u)$ can trivially be computed in polynomial time.
- every link stream in $\mathcal{A}(L, u)$ is (α, ω)-graph like.
- finally, let k be an arbitrary integer and suppose that the following property holds: $\Big($there exists a series of k edition operations, among which σ involving u, transforming L into $L' \in \mathcal{L}$ in such a way that L' is (α, ω)-graph like and that u is not isolated in $L'\Big)$. Let L' be defined as in the statement. We will show that $L' \in \mathcal{A}(L, u)$. Let us consider $N_{L'}[u] = N_{L'}(u) \cup \{u\}$. Since L' is (α, ω)-graph like and by definition of $N_\sigma[u]$ we have that $N_{L'}[u] \in N_\sigma[u]$. Hence, $\{([\alpha, \omega], vw) : vw \in N_{L'}[u] \otimes N_{L'}[u]\} \subseteq E(L')$. What's more, $L' \in \mathcal{L}$, that is, L' is sparse split. Hence, there exist time interval I and vertex subset A such that $E(L') = \{(I, vw) : vw \in A \otimes A\}$. Therefore $I = [\alpha, \omega]$. A second consequence is that $N_{L'}[u] \subseteq A$. We will show that they are equal. Suppose there exists some vertex $v \in A \setminus N_{L'}[u]$. Then, $v \neq u$ and v is a neighbour of u in L' (by definition of L' being sparse split). In other words, $v \in N_{L'}[u]$, which is a contradiction. Therefore, $N_{L'}[u] = A$ and $A \in N_\sigma[u]$. We have just proved that $L' \in \mathcal{A}(L, u)$. As a consequence, the following property holds: $\Big($there exists a series of k edition operations, among which σ involving u, transforming L into $L' \in \mathcal{A}(L, u)$ in such a way that u is not isolated in $L'\Big)$.

We conclude that $\mathcal{A}$ is a σ-greedy (α, ω)-edition algorithm for sparse split link streams. □

Theorem 2 *The parameterized version of (α, ω)-SS-LSE has a linear vertex kernel.*

Proof: follows from Property 4 and Theorem 1. □

Corollary 1 *The parameterized version of SS-LSE is fixed parameter tractable.*

Proof: follows from Property 3 and Theorem 2. □

4 Bi-Sparse Split Link Stream Edition

Unfortunately, the class of bi-sparse split link streams does not seem to be σ-greedy editable, for any integer σ. We cope with this inconvenience by two main ideas. First we use a loose version of greedy editing (cf. Procedure 7). Second, we must control the behaviour of high degree vertices: we remark that when a vertex has a degree higher than the parameter k, then it cannot be

[1] There are at most $\binom{|V(L)|-1}{\sigma}$ choices of $N \in N_\sigma(u)$. Once $A = N \cup \{u\}$ is chosen, L' is entirely defined.

isolated in any link stream that can be obtained after k edition operations on L. Hence, that vertex must belong to one of the two clique-like vertex sets in any bi-sparse split link stream. Formally, we implement our ideas by considering the following version of BiSS-LSE, that we call BiSS-LSE with VIP vertices.

VIP-BiSS-LSE:

INPUT: L a link stream, $k \in \mathbb{N}$, and $X, Y \subseteq V(L)$ vertex subsets called VIP.
QUESTION: Is there a series of at most k edition operations transforming L into L' such that there exist consecutive time intervals I, J, and vertex subsets $A \supseteq X$, $B \supseteq Y$ such that $E(L) = \{(I, uv) : uv \in (A \setminus B) \otimes (A \setminus B)\} \cup \{(J, uv) : uv \in (B \setminus A) \otimes (B \setminus A)\} \cup \{(I \cup J, uv) : uv \in (A \cap B) \otimes (A \cap B)\}$?

Clearly, solving BiSS-LSE on input (L, k) is equivalent to solving VIP-BiSS-LSE on input $(L, k, \emptyset, \emptyset)$. As before, we consider the following boxed version of BiSS-LSE, where I and J are two given consecutive time intervals.

VIP-(I, J)-BiSS-LSE:

INPUT: L a link stream, $k \in \mathbb{N}$, and $X, Y \subseteq V(L)$ vertex subsets called VIP.
QUESTION: Is there a series of at most k edition operations transforming L into L' such that there exist vertex subsets $A \supseteq X$, $B \supseteq Y$ such that $E(L) = \{(I, uv) : uv \in (A \setminus B) \otimes (A \setminus B)\} \cup \{(J, uv) : uv \in (B \setminus A) \otimes (B \setminus A)\} \cup \{(I \cup J, uv) : uv \in (A \cap B) \otimes (A \cap B)\}$?

Procedure 5 (Pruning the VIPs) On input instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, we will modify (X, Y), not (L, k). We initialize $(X', Y') \leftarrow (X, Y)$ and process as follows. For every vertex $u \in V(L)$ such that $N_L(u) > 2 \times k$:

- if after k edition operations u cannot be isolated on the interval I, then perform $X' \leftarrow X' \cup \{u\}$.
- if after k edition operations u cannot be isolated on the interval J, then perform $Y' \leftarrow Y' \cup \{u\}$.

At the end of the looping process, return instance (L, k, X', Y').

The two following lemmas are clearly true.

Lemma 6 *Procedure 5 is safe with respect to VIP-(I, J)-BiSS-LSE.*

Lemma 7 *Any output (L', k', X', Y') of Procedure 5 satisfies: for every $u \in V(L') \setminus (X' \cup Y')$, we have $|N_{L'}(u) \setminus (X' \cup Y')| \leq 2 \times k'$.*

As with Procedure 1, we first restrict our instance to the targetted time intervals (I, J).

Procedure 6 On input instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, remove from L every timed edge (K, uv) with

$$\{\min K, \max K\} \cap \{\min I, \max I = \min J, \max J\} = \emptyset,$$

and decrease k with the number of such timed edges.

Lemma 8 *Procedure 6 is safe with respect to VIP-(I, J)-BiSS-LSE.*

Proof: Straightforward by an exhaustive case analysis, similarly as what was done for Lemma 1. □

Let us now present our loose version of greedy edition for BiSS-LSE. Let (L, k, X, Y) be an instance of VIP-(I, J)-BiSS-LSE. For every pair of distinct vertices $u, v \in V(L) \setminus (X \cup Y)$, we will define the following set $\mathcal{A}(L, X, Y, u, v)$ of bi-sparse split link streams. Let us consider $N_L(u)$. With at most σ edition operations involving u, there are only a polynomial number of possibilities of transforming $N_L(u)$ into a neighbourhood of u with exclusively timed edges of the form (I, uw) for some $w \in V(L)$. Among these possibilities, we rule out those which do not contain all vertices from X. We denote all the remaining possibilities by $N_{\sigma,X,I}(u) = \{N : X \subseteq N \subseteq V(L) \wedge N$ is the neighbourhood of u with exclusively timed edges of the form (I, uw) after at most σ edition operations involving u from original link stream $L\}$. We note $N_{\sigma,X,I}[u] = \{N \cup \{u\} : N \in N_{\sigma,X,I}(u)\}$. Similarly, we define $N_{\sigma,Y,J}[v]$. We then define $\mathcal{A}(L, X, Y, u, v) = \{L' : V(L') = V(L) \wedge T(L') = T(L) \wedge E(L') = \{(I, st) : st \in (A \setminus B) \otimes (A \setminus B)\} \cup \{(J, st) : st \in (B \setminus A) \otimes (B \setminus A)\} \cup \{(I \cup J, st) : st \in (A \cap B) \otimes (A \cap B)\} \wedge A \in N_{\sigma,X,I}[u] \wedge B \in N_{\sigma,Y,J}[v]\}$.

Procedure 7 On input instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, we process as follows. For every pair of distinct vertices $u, v \in V(L) \setminus (X \cup Y)$, we compute the above defined set $\mathcal{A}(L, X, Y, u, v)$. Then, for every link stream $L' \in \mathcal{A}(L, X, Y, u, v)$, if there exists a series of k edition operations transforming L into L', then return $(\textsc{Positive}, 0, \emptyset, \emptyset)$. After all looping processes, if we still have not returned anything yet, then return original input (L, k, X, Y).

Lemma 9 *Procedure 7 is safe with respect to VIP-(I, J)-BiSS-LSE.*

Proof: On input (L, k, X, Y) Procedure 7 returns either $(\textsc{Positive}, 0, \emptyset, \emptyset)$ or (L, k, X, Y). Hence, the output value for the parameter is at most k. As for complexity issues, after producing $\mathcal{A}(L, X, Y, u)$, which is in polynomial time by definition of $\mathcal{A}$, Procedure 7 does in total $\sum_{u,v \in V(L)\setminus(X\cup Y)} |\mathcal{A}(L, X, Y, u, v)|$ checks whether there exists a series of k edition operations transforming L into some given L'. Because both L and L' in each case are known, each checking process can be done in polynomial time, cf. Property 1. By definition of $\mathcal{A}(L, X, Y, u, v)$, this set has at most $\binom{|V(L)|-1}{\sigma} \times \binom{|V(L)|-1}{\sigma}$ members. Hence, $\sum_{u,v \in V(L)\setminus(X\cup Y)} |\mathcal{A}(L, X, Y, u, v)|$ is polynomial in the size of L. Therefore, Procedure 7 terminates in polynomial time.

Now, suppose that (L, k, X, Y) yields a positive answer to VIP-(I, J)-BiSS-LSE. Since Procedure 7 only returns either $(\text{POSITIVE}, 0, \emptyset, \emptyset)$ or (L, k, X, Y), which both yield positive answer to VIP-(I, J)-BiSS-LSE, we have indeed that the input instance and the outputted instance of Procedure 7 yield the same answer to VIP-(I, J)-BiSS-LSE.

On the other hand, suppose that (L, k, X, Y) yields a negative answer to VIP-(I, J)-BiSS-LSE. Here, if Procedure 7 returns (L, k, X, Y) then everything is fine and there is no more thing to prove. Let us suppose by contradiction that Procedure 7 returns $(\text{POSITIVE}, 0, \emptyset, \emptyset)$. This return instruction must have been done when examining some pair of distinct vertices $u, v \in V(L) \setminus (X \cup Y)$ and some $L' \in \mathcal{A}(L, X, Y, u, v)$. By definition of the return instruction in Procedure 7, L' can be obtained from L after k edition operations. By definition of $\mathcal{A}(L, X, Y, u, v)$, L' is a bi-sparse split link stream with timed edges exclusively of the form (I, st) or (J, st), where the "clique" at time interval I, resp. J, include all vertices from X, resp. Y. But then, L' would be a certificate that (L, k, X, Y) yields a positive answer to VIP-(I, J)-BiSS-LSE. Contradiction. □

Among the vertices in the set $V(L)\setminus(X\cup Y)$ we will distinguish two categories: those having a degree of at least $\sigma + 1$ and the other ones. Let S_σ denote the former ones, and n_σ their number: $S_\sigma = \{u \in V(L)\setminus(X\cup Y) : |N_L(u)| \geq \sigma+1\}$ and $n_\sigma = |S_\sigma|$.

Now let L and L' be two arbitrary link streams. We note $S_\sigma^{L\to L'} = \{u \in S_\sigma :$ there are at most σ edition operations involving u on the way from L to $L'\}$. Notice that vertices in $S_\sigma^{L\to L'}$ are not isolated in L'.

Finally, we note $n_\sigma^{L\to L'} = |S_\sigma^{L\to L'}|$.

Lemma 10 *Let (L, k, X, Y) be an output of the successive application of Procedure 5 then Procedure 6. On input (L, k, X, Y) Procedure 7 outputs (L, k, X, Y) (and not $(\text{POSITIVE}, 0, \emptyset, \emptyset)$) if and only if the following property holds. If L' is a certificate that (L, k, X, Y) yields a positive answer to VIP-(I, J)-BiSS-LSE, and if there exists $u \in V(L) \setminus (X \cup Y)$ such that u is not isolated in L' and there have been at most σ edition operations involving u on the way from L to L', then $n_\sigma^{L\to L'} \leq 2 \times k + \sigma + 1$.*

Proof: We have in particular that L' is a bi-sparse split link stream. Let A and B be such that $E(L') = \{(I, st) : st \in (A \setminus B) \otimes (A \setminus B)\} \cup \{(J, st) : st \in (B \setminus A) \otimes (B \setminus A)\} \cup \{(I \cup J, st) : st \in (A \cap B) \otimes (A \cap B)\}$. Since u is not isolated in L', $u \in (A \cup B) \setminus (X \cup Y)$. W.l.o.g. suppose that $u \in A$.

If $S_\sigma^{L\to L'}$ is an empty set, then $n_\sigma^{L\to L'} = 0$, and there is no more thing to show. We assume the contrary and let $v \in S_\sigma^{L\to L'}$. By definition, $N_L(v) \geq \sigma + 1$ and there are at most σ edition operations involving v on the way from L to L'.

Hence, v cannot be isolated in L'. Now, if $v \in B \setminus (X \cup Y)$, we have in particular that $v \neq u$; moreover, Procedure 7 would have returned $(\textsc{Positive}, 0, \emptyset, \emptyset)$ when examining $\mathcal{A}(L, X, Y, u, v)$. Hence, $v \in A \setminus (X \cup Y)$. We have just proved that $S_\sigma^{L \to L'} \subseteq A \setminus (X \cup Y)$.

Finally, since (L, k, X, Y) is an output of Procedure 5, by Lemma 7, and by the fact $u \notin X \cup Y$, we have $|N_L(u) \setminus (X \cup Y)| \leq 2 \times k$. Besides, we know that there are at most σ edition operations involving u on the way from L to L'. Hence, $|N_{L'}(u) \setminus (X \cup Y)| \leq 2 \times k + \sigma$. Since L' is a bi-sparse split link stream, we have that $u \in S_\sigma^{L \to L'}$. Hence, $u \in A \setminus (X \cup Y)$, and $A = N_{L'}(u) \cup \{u\}$. Therefore, $|A \setminus (X \cup Y)| \leq 2 \times k + \sigma + 1$.

Since $S_\sigma^{L \to L'} \subseteq A \setminus (X \cup Y)$, we have that $n_\sigma^{L \to L'} \leq 2 \times k + \sigma + 1$. □

Procedure 8 On input instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, if $n_\sigma > \frac{2 \times k}{\sigma + 1} + 2 \times k + \sigma + 1$ then output $(\textsc{Negative}, 0, \emptyset, \emptyset)$. Otherwise, initialize $(L', k') \leftarrow (L, k)$. For every vertex $u \in V(L') \setminus (X \cup Y)$ such that $\mathbb{0}_u^{L'} \leq \sigma$, isolate then remove u, that is, perform $(L', k') \leftarrow (L'', k' - \mathbb{0}_u^{L'})$, where $V(L'') = V(L') \setminus \{u\}$, $T(L'') = T(L')$ and $E(L'') = E(L') \setminus \{(I, uv) : I \subseteq T(L') \wedge v \in V(L')\}$. At the end of the process, return (L', k', X, Y).

Lemma 11 *Let (L, k, X, Y) be an output of the successive application of Procedures 5, 6 then 7. On input (L, k, X, Y) Procedure 8 is safe with respect to VIP-(I, J)-BiSS-LSE.*

Proof: In fact, the situation is very similar to what was done in Procedure 3 in the previous section. We merely adapt the proof of previous Lemma 4 so that it fits the new formalism. We stress that the adaptation is very procedural.

Let (L', k', X, Y) be the output of Procedure 8. Clearly, Procedure 8 runs in polynomial time and can only decrease the value of the parameter, that is, we have $k' \leq k$.

Suppose that (L', k', X, Y) yields a positive answer to VIP-(I, J)-BiSS-LSE, that is, there exists a bi-sparse split link stream L'' such that L'' is made of exclusively timed edges over one of the three time intervals I, J, and $I \cup J$; and such that L'' can be obtained from L' after at most k' edition operations. Now, for every $u \in V(L)$ such that $\mathbb{0}_u^L \leq \sigma$, add a new isolated vertex to L'', and obtain L'''. By heredity, L''' is bi-sparse split. Furthermore, it is clear that L''' is made of exclusively timed edges over one of the three time intervals I, J, and $I \cup J$. Here, the important point is that we have constructed L''' in such a way that: as far as L is concerned, and after we isolate every vertex u in L such that $\mathbb{0}_u^L \leq \sigma$, then we can mimic the k' operations transforming L' into L'' in order to transform the remaining of L into L''' with k' similar operations. Hence, we have that L''' can be obtained from L after isolating every u with $\mathbb{0}_u^L \leq \sigma$, plus k' edition operations. Finally, by definition of

(L', k') in Procedure 8, the total number of these edition operations is exactly k. Hence, in this case the input instance (L, k, X, Y) of Procedure 8 yields a positive answer to VIP-(I, J)-BiSS-LSE.

Suppose now that (L, k, X, Y) yields a positive answer to VIP-(I, J)-BiSS-LSE, that is, there exists a bi-sparse split link stream L'' such that L'' is made of exclusively timed edges over one of the three time intervals I, J, and $I \cup J$; and such that L'' can be obtained from L after at most k edition operations. Then, by Lemma 10, for every vertex u of $V(L)$, either u is an isolated vertex in L'' or there must be at least $\sigma + 1$ edition operations involving u on the way from L to L''. Let $u \in V(L)$ such that $\mathbb{0}_u^L \leq \sigma$ (which implies that u is isolated in L'). If u is an isolated vertex in L'' then the minimum number of edition operations to isolate u from input instance L, which is $\mathbb{0}_u^L$, is indeed what we have use in Procedure 8 as long as u is concerned. If u is not an isolated vertex in L'' then we know there are at least $\sigma + 1$ edition operations required in order to transform the neighbourhood of u in L into the neighbourhood of u in L''. However, let us consider L''', the link stream that is obtained by removing u from L''. Firstly, L''' is made of exclusively timed edges over one of the three time intervals I, J, and $I \cup J$. Secondly, L''' is bi-sparse split because of heredity. Lastly, as long as u is concerned, we can obtain L''' from L' by at most the same edition operations as obtaining L'' from L. Basically, the fact that L can be transformed into L'' by k edition operations guarantees that L' can be transformed into L''' by at most k' edition operations. Therefore, the outputted instance (L', k', X, Y) of Procedure 8 yields a positive answer to VIP-(I, J)-BiSS-LSE. □

Procedure 9 On input instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, if $|V(L) \setminus (X \cup Y)| > \frac{2 \times k}{\sigma + 1}$ then we output $(\text{NEGATIVE}, 0, \emptyset, \emptyset)$. Otherwise, we output (L, k, X, Y).

Lemma 12 *Let (L_0, k_0, X_0, Y_0) be an arbitrary instance of VIP-(I, J)-BiSS-LSE. First apply Procedure 5 and Procedure 6. Then, iteratively apply (Procedure 7 then Procedure 8) until the output is constant, and let (L, k, X, Y) be that output. Then, on input instance (L, k, X, Y), Procedure 9 is safe with respect to VIP-(I, J)-BiSS-LSE.*

Proof: For any solution L' to a positive instance (L, k, X, Y) of VIP-(I, J)-BiSS-LSE, let $T = S_\sigma \setminus S_\sigma^{L \to L'}$. By Lemma 10, $|T| \geq n_\sigma - (2 \times k + \sigma + 1)$. Hence, if Procedure 9 returns $(\text{NEGATIVE}, 0, \emptyset, \emptyset)$, then $|T| > \frac{2 \times k}{\sigma + 1}$. By definition, for any member $u \in T$, we need to spend at least $\sigma + 1$ edition operations involving u. Therefore, with only k operations we cannot succeed in transforming L into L'. Hence the result.

First, remark that, at the last instruction, Procedure 8 outputs $(\text{NEGATIVE}, 0, \emptyset, \emptyset)$ if and only if the value of (L', k', X, Y) yields a negative answer to VIP-

(I, J)-BiSS-LSE: indeed, a part from VIP vertices in X and Y, L' will then only contain vertices where, for each of them, we must spend at least $\sigma + 1$ edition operations involving the vertex; moreover we have in this case that $|V(L') \setminus (X \cup Y)| > \frac{2\times k'}{\sigma+1}$. However, with only k' edition operations, since each can only involve two vertices, we cannot involve more than $\frac{2\times k'}{\sigma+1}$ vertices when each of them needs to be involved at least $\sigma + 1$ times. Hence, when Procedure 8 outputs $(\textsc{Negative}, 0, \emptyset, \emptyset)$ in the last instruction, the value of (L', k', X, Y) yields a negative answer to VIP-(I, J)-BiSS-LSE. It is consequently sufficient to prove that (L, k, X, Y) and (L', k', X, Y) yield the same answer to VIP-(I, J)-BiSS-LSE. □

Corollary 2 *Let (L, k, X, Y) be an arbitrary instance of VIP-(I, J)-BiSS-LSE. First apply Procedure 5. Then, apply Procedure 6. Then, iteratively apply (Procedure 7 then Procedure 8) until the output is constant. Finally apply Procedure 9. This process results in a kernelization algorithm for VIP-(I, J)-BiSS-LSE. Moreover, the number of vertices in the kernel is linear in the parameter.*

Proof: By Lemmas 6, 8, 9, 11, and 12, the procedure given in the statement is safe. The only thing left to prove is that the size of the output (L', k', X', Y') of the procedure is bounded by a function of k'. This is clearly the case if the output of Procedure 7 is $(\textsc{Positive}, 0, \emptyset, \emptyset)$, or the output of Procedure 9 is $(\textsc{Negative}, 0, \emptyset, \emptyset)$.

Otherwise, let (L, k, X, Y) be the output obtained at the end of the iteration of (Procedure 7 and Procedure 8). Note that (L, k, X, Y) is also the input of Procedure 9. Since Procedure 9 does not return $(\textsc{Negative}, 0, \emptyset, \emptyset)$, we have that $|V(L) \setminus (X \cup Y)| \leq \frac{2\times k}{\sigma+1}$, and that $(L, k, X, Y) = (L', k', X', Y')$.

Finally, after Procedure 6, the maximum number of possible edges is obtained if there are three timed edges for each pair of vertices (u, v): one involving $\min I$, one involving $\max I = \min J$ and one involving $\max J$. Therefore, $|E(L')| = O(|V(L')|^2)$.

□

We have proved the following result:

Theorem 3 *The parameterized version of VIP-(I, J)-BiSS-LSE has a linear vertex kernel. The parameterized version of BiSS-LSE is fixed parameter tractable.*

Link stream class	Graph-like	Greedy editable	Edition complexity
1-1-Sparse-Split	Yes	Yes	Boxed kernel (1)
d-1-Sparse-Split	Yes	Yes	Probable boxed kernel (2)
d-1-Cluster	Yes	Yes	Probable boxed kernel (2)
*-1-Sparse-Split	Yes	Unlikely	Possible FPT (3)
*-1-Cluster	Yes	Unlikely	Possible FPT (3)
1-2-Sparse-Split	Yes	Unlikely	Boxed kernel (1)
1-k-Sparse-Split	No	Unlikely	Boxed kernel (1bis)
d-k-Sparse-Split	No	Unlikely	Probable boxed kernel (2)
d-k-Cluster	No	Unlikely	Probable boxed kernel (2)
*-k-Sparse-Split	No	Unlikely	Possible FPT (3)
*-k-Cluster	No	Unlikely	Possible FPT (3)
d-*-Sparse-Split	No	Unlikely	? (4)
d-*-Cluster	No	Unlikely	? (4)
--Sparse-Split	No	Unlikely	? (4)
--Cluster	No	Unlikely	? (4)

Table 1
Classification of some link stream editing problems.

5 Conclusion and Perspectives

We gave generic ideas to solve parameterized versions of edition problems on link streams, that is, to edit an arbitrary given link stream by local edge editions into a link stream belonging to some given class of link streams. In particular, we proved that both sparse split and bi-sparse split link stream edition problems are fixed parameter tractable. Our approach relies on proving that a constrained version of the problem, which is defined by a target time interval, admits a linear vertex kernel.

We believe that our technique can be generalized to solve a broad class of parameterized edition problems on link streams.

5.1 Extension - Multiple "time intervals"

A link stream may be seen as a finite series of graphs. Indeed, if $(t_k)_k$ is the ordered series of timestamps at which at least one timed edge begins or ends, we can build the series of graph $(G_k)_k$ as follows: two vertices are connected in G_k if and only if they are linked between t_k and t_{k+1}. The sparse split link stream class consists of graph equivalent link streams, i.e. their representation as a series of graphs is a single graph. However, we can define new classes with no such restriction. From now on, we shall say that a link stream verifies a certain property (for instance being sparse-split) if and only if all of its graphs verify the same property. We shall call *number of time intervals* in a link stream the number of graphs in its series of graphs.

5.2 Problem Denomination

We give here a classification of different problems in this category. Each problem is characterized by two prefixes and a stem.

The first prefix describes the maximum number of cliques allowed.

- *d*- The maximum number of cliques of size ≥ 2 allowed is d
- ***- Any number of cliques of size ≥ 2 allowed

The second prefix describes the maximum number of time intervals.

- -*k*- The maximum number of neighbouring intervals is k, that is: $\exists (t_i)_{i \in [0,k]}$, $E(L) \subseteq \{([t_i, t_j], uv) : u, v \in V(L) \land 0 \leq i \leq j \leq k\}$
- -***- Any number of neighbouring intervals, that is: $\exists k,\ \exists (t_i)_{i \in [0,k]}$, $E(L) \subseteq \{([t_i, t_j], uv) : u, v \in V(L) \land 0 \leq i \leq j \leq k\}$

The stem describes whether or not we count isolated vertices when counting cliques.

- -*Sparse-Split* On each interval, the link stream is a clique (or d cliques) plus isolated vertices
- -*Cluster* On each interval, the link stream is a set of disjoint cliques

Table 1 presents the resulting classes, whether they are graph-like and indications on the fact that they would be greedy editable or not. In the table, we also give indication of the complexity of the corresponding edition problem. N.B.: (1) our results; (1bis) true by a straightforward generalization of Property 3 and Section 4; (2) probably true, e.g. by a careful extension of our approach; (3) possibly true, e.g. by adapting techniques given in [4,13] to our approach; (4) open, we strongly believe that dealing with the time dimension is the trickier task when trying to solve this kind of problems.

5.3 *Other perspectives*

Our work opens the way to several other interesting research directions.

First, other problems from graph theory have natural equivalent in link streams. For instance, the notion of temporal matching [2] or separators in link streams (called in this case temporal graphs) [**?**] have already been studied. There is no doubt that other notions such as for instance Steiner trees or clique-width have meaningful equivalents in link streams, and assessing whether existing algorithms can be easily extended, or if other procedures have to be designed, is a promising and fruitful direction.

The problems we studied in this paper have direct applications in several data mining problems. Therefore, we believe that it is important to be able to apply our kernelization algorithms to real-world datasets, in order to be able to transform them into sparse split or bi-sparse split link streams.

The preliminary work we have done so far shows that the main hurdle in doing so does not lie in obtaining the solution once we have a kernel, but from finding the correct α and ω values for doing so. Indeed, an exhaustive search leads to applying our kernelization procedures for all pairs of relevant time instants, which is infeasible in practice, even for relatively small datasets. Finding small sets of candidate values for α and ω is therefore a crucial concern.

Another concern in dealing with real-world datasets is that in general, we do not have any precise constraint for the number of desired editions k, but are instead interested in finding the minimum k such that it is possible to transform a given link stream into a sparse split or bi-sparse split one. Therefore, a natural direction would be to design approximation algorithms for this problem, where the goal would be to transform a given link stream into a sparse split or bi-sparse split one with a number of editions that is within a constant factor of the optimal. This kind of approach has already been successfully addressed for the problem of matching [2] and we believe that it is also very promising in our case.

References

[1] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3), 2004.

[2] J. Baste, B.-M. Bui-Xuan, and A. Roux. Temporal matching. *submitted, preliminary results presented at CTW'18*. `https://arxiv.org/abs/1812.08615`.

[3] Stephen P. Borgatti and Martin G. Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, 1999.

[4] N. Bousquet, J. Daligault, and S. Thomassé. Multicut is fpt. In *43rd ACM Symposium on Theory of Computing (STOC'11)*, STOC '11, pages 459–468. ACM, 2011.

[5] B.-M. Bui-Xuan, C. Magnien, and P. Meyer. Kernelization algorithms for some link stream editing problems. In *8th Workshop on Graph Classes, Optimization, and Width Parameters (GROW '17)*, 2017.

[6] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[7] Peter Damaschke and Olof Mogren. Editing the simplest graphs. In *Proceedings of WALCOM*, 2014.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[9] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[10] F. Hüffner, C. Komusiewicz, and A. Nichterlein. Editing graphs into few cliques: complexity, approximation, and kernelization schemes. In *14th International Symposium on Algorithms and Data Structures (WADS'15)*, volume 9214 of *LNCS*, pages 410–421, 2015.

[11] I. Kováč, I. Selečéniová, and M. Steinová. On the clique editing problem. In *39th International Symposium on Mathematical Foundations of Computer Science (MFCS'14)*, volume 8635 of *LNCS*, pages 469–480, 2014.

[12] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. Stream graphs and link streams for the modeling of interactions over time, 2018. https://arxiv.org/abs/1710.04073.

[13] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *43rd ACM Symposium on Theory of Computing (STOC'11)*, STOC '11, pages 469–478. ACM, 2011.