Temporal matching in link stream: kernel and approximation

Julien Baste, Binh-Minh Bui-Xuan

HAL Id: hal-02075831

https://hal.sorbonne-universite.fr/hal-02075831

Submitted on 21 Mar 2019

Temporal matching in link stream: kernel and approximation

Julien Baste[1]   Binh-Minh Bui-Xuan[1]

Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris, France
[julien.baste,buixuan]@lip6.fr

Abstract

A link stream is a sequence of pairs of the form $(t, \{u, v\})$, where $t \in \mathbb{N}$ represents a time instant and $u \neq v$. Given an integer γ, the γ-edge between vertices u and v, starting at time t, is the set of temporally consecutive edges defined as $\{(t', \{u, v\}) \mid t' \in [\![t, t + \gamma - 1]\!]\}$. We introduce the notion of temporal matching of a link stream to be a set of pairwise non overlapping γ-edges belonging to the link stream. Unexpectedly, the problem of computing a temporal matching of maximum size turns out to be NP-hard. We provide a kernelization algorithm parameterized by the solution size for the problem. As a byproduct we also depict a 2-approximation algorithm.

Keywords : *graph, parameterized algorithm, link stream.*

1 Introduction

The question of mining data stemming from human activities not only comes from an old and well studied topic of social science. In the recent years, mining human data is also moved by the sheer amount of applications in web analytics, graph mining statistics, criminology graph visualization and so on. An important, yet not quite well understood, feature of graph data collected by such tools comes from the time dimension: edges here are timestamped edges. They come ordered by the time interval where they are effectively active. We call this kind of data a link stream, in the sense of [3, 4].

A link stream L is a sequence of pairs of the form $(t, \{u, v\})$ where $\{u, v\}$ is an edge, in the sense of classical loopless undirected simple graphs, and $t \in \mathbb{N}$ is an integer representing a discretized time instant. If every pair $(t, \{u, v\})$ in L satisfies $t = t_0$ for some fixed t_0, then we say that link stream L is constant. A constant link stream is equivalent to the formalism of a graph in the classical sense. Given an integer γ, a time instant t, and two distinct vertices u and v, we define the γ-edge between u and v starting at time t as the set $\{(t', \{u, v\}) \mid t' \in [\![t, t + \gamma - 1]\!]\}$. Two γ-edges are compatible when they are not overlapping (cf. formal definition in Section 2). Finally, a γ-matching of link stream L is a set of compatible γ-edges where each γ-edge contains exclusively edges from L. We consider the problem of computing a maximum γ-matching of an input link stream, that we call γ-MATCHING. When $\gamma = 1$, this problem can be solved by a slight extension of the notorious polynomial time algorithm given in [2].

Unfortunately, we found that γ-MATCHING on arbitrary input is NP-hard, as soon as $\gamma > 1$. We address the question of pre-processing, in polynomial time, an input instance of γ-MATCHING, in order to reduce it to an equivalent instance of smaller size, in the sense of kernelization algorithms introduced by [1]. We show that γ-MATCHING when parameterized by the solution size admits a quadratic kernel. On the way to do so, we also depict a procedure which turns out to define a 2-approximation algorithm for γ-MATCHING. Our paper is organised as follows. We first introduce the notion of temporal matching (Section 2), before presenting our algorithmic tools (Section 3) in order to obtain our main result, the kernelization algorithm (Section 4). We close the paper with concluding remarks and directions for further research (Section 5).

2 Temporal matching

We denote by $\mathbb{N}$ the set of nonnegative integer. Given two integers p and q, we denote by $[\![p, q]\!]$ the set $\{r \in \mathbb{N} \mid p \leq r \leq q\}$. A *link stream* L is a triple (T, V, E) such that $T \subseteq \mathbb{N}$, V is a set, and $E \subseteq T \times \binom{V}{2}$. The elements of V are called *vertices* and the elements of E are called *(timed) edges*. A *temporal vertex* of L is a pair (t, v) such that $t \in T$ and $v \in V$.

Given an integer γ, a γ-edge between two vertices u and v at time t, denoted $\Gamma_\gamma(t, u, v)$, is the set $\{(t', \{u, v\}) \mid t' \in [\![t, t + \gamma - 1]\!]\}$. We say that a γ-edge Λ is incident to temporal vertex (t, v) if there exists a vertex $u \in V$ such that $(t, \{u, v\}) \in \Lambda$. We say that two γ-edges are compatible if there is no temporal vertex (t, v) that is incident to both of them. A γ-matching $\mathcal{M}$ of a link stream L is a set of pairwise compatible γ-edges. We say that a γ-edge Λ is incident with a vertex $v \in V$ if there exist a vertex $u \in V$ and an integer $t \in T$ such that $\Lambda = \Gamma_\gamma(t, u, v)$. We say that an edge $e \in E$ is in a γ-matching $\mathcal{M}$ if there exists $\Lambda \in \mathcal{M}$ such that $e \in \Lambda$.

This paper focus on the following problem.

γ-MATCHING
Input: A link stream L and an integer k.
Output: A γ-matching of L of size k or a correct answer that such a set does not exist.

Property 1 γ-MATCHING *is NP-hard.*

Sketch of the proof. We reduce from 3-SAT, that is well known to be NP-hard. Let φ be a formula with n variables $x_1, \ldots, x_n$ and m clauses $C_0, \ldots, C_{m-1}$ such that each clauses is of size at most 3. Without loss of generality, we assume that a clause does not contain twice the same variable. We call X the set containing the n variables and $\mathcal{C}$ the set containing the m clauses.

We define the linkstream $L = (T, V, E)$ in the following way: $T = [\![0, (m+1)\gamma - 1]\!]$, $V = \{x^-, x^=, x^+ \mid x \in X\} \cup \{x_t^{++}, x_t^{--} \mid x \in X, t \in [\![0, m-1]\!]\} \cup \{c\}$, and $E = E_{var} \cup E_{cla}$ where:

$$
\begin{aligned}
E_{var} \;=\; & \{(t, \{x^=, x^+\}), (t, \{x^=, x^-\}) \mid t \in [\![0, (m+1)\gamma - 1]\!], x \in X\} \\
\cup \;& \left\{(t, \{x^+, x_i^{++}\}), (t, \{x^-, x_i^{--}\}) \mid t \in [\![1, m\gamma]\!], x \in X, i = \left\lfloor \frac{t-1}{\gamma} \right\rfloor \right\} \\
E_{cla} \;=\; & \{(t, \{c, x_i^{++}\}) \mid t \in [\![i\gamma + 1, (i+1)\gamma]\!], i \in [\![0, m-1]\!], x \in X, x \text{ appears positively in } C_i\} \\
\cup \;& \{(t, \{c, x_i^{--}\}) \mid t \in [\![i\gamma + 1, (i+1)\gamma]\!], i \in [\![0, m-1]\!], x \in X, x \text{ appears negatively in } C_i\}.
\end{aligned}
$$

We depict in Figure 1 the linkstream build for $\gamma = 3$ and $\varphi = (\overline{w} \vee x \vee \overline{y}) \wedge (\overline{w} \vee \overline{x} \vee z)$. In order to finish the proof of the property, it is sufficient to show the equivalence between above gadget and the original instance of 3-SAT, that is:

Claim 1 *There is an assignment of the variables that satisfies φ if and only if L contains a γ-matching of size $(2m+1)n + m$.*

Due to space limit, we omit the formal proof of Claim 1. Intuitively, the edge between $(0, \{x^=, x^+\})$ and $(0, \{x^=, x^-\})$ that is in the requested γ-matching determines if the variable x is set to true or false. Moreover, the size of the requested γ-matching ensures that if the edge $(0, \{x^=, x^+\})$ (resp. $(0, \{x^=, x^-\})$) is in the γ-matching, then every edge $(t, \{x^=, x^+\})$ (resp. $(t, \{x^=, x^-\})$), $t \in [\![0, (m+1)\gamma - 1]\!]$, and every edge $(t, \{x^-, x_i^{--}\})$ (resp. $(t, \{x^+, x_i^{++}\})$), $t \in [\![1, m\gamma]\!]$, $i = \left\lfloor \frac{t-1}{\gamma} \right\rfloor$, are in the γ-matching as well. Finally, during the time interval $[\![i\gamma + 1, (i+1)\gamma]\!]$, we check that the clause C_i is satisfied.

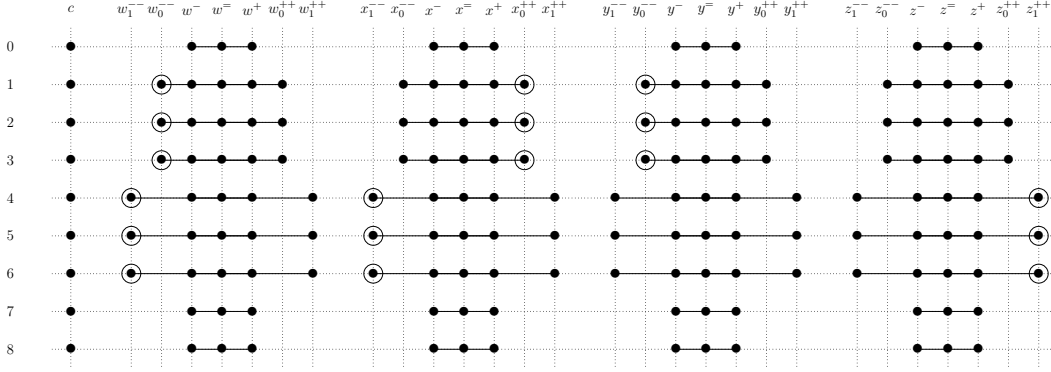FIG. 1: The constructed linkstream L when $\varphi = (\overline{w} \vee x \vee \overline{y}) \wedge (\overline{w} \vee \overline{x} \vee z)$ and $\gamma = 3$. Here $T = [\![0, 8]\!]$ and an edge $(t, \{u, v\})$ of L is depicted by a bold line, following the horizontal line corresponding to time t, going from the vertical line corresponding to u to the vertical line corresponding to v. For readability, the edges incident with c are not drown. Instead, we have circled the temporal vertices that are neighbors of c.

3 Approximation algorithm

In this section, we adopt the greedy approach in order to provide a $2-$approximation algorithm for γ-MATCHING. Let $L = (T, V, E)$ be a link stream. Let $\mathcal{P}$ be the set of every γ-edges of L. Let $\preceq$ be an arbitrary total ordering on the elements of $\mathcal{P}$ such that for any two elements of $\mathcal{P}$, $\Lambda_1 = \Gamma_\gamma(t_1, u_1, v_1)$ and $\Lambda_2 = \Gamma_\gamma(t_2, u_2, v_2)$ such that $t_1 < t_2$, we have $\Lambda_1 \preceq \Lambda_2$.

We denote by $\mathcal{A}$ the following greedy algorithm. The algorithm starts with $\mathcal{M} = \emptyset$, $\mathcal{Q} = \mathcal{P}$, and a function $\rho : V \times T \to \{0, 1\}$ such that for each $(t, v) \in T \times V$, $\rho(t, v) = 0$. The purpose of ρ is to keep track of the temporal vertices that are incident to a γ-edge of $\mathcal{M}$. As long as $\mathcal{Q}$ is not empty, the algorithm selects Λ, the γ-edge of $\mathcal{Q}$ that is minimum for $\preceq$, and removes it from $\mathcal{Q}$. Let K be the set of the 2γ temporal vertices that are incident to Λ. If, for each $(t, v) \in K$, $\rho(t, v) = 0$, then the algorithm adds Λ to $\mathcal{M}$ and for each $(t, v) \in K$, it sets $\rho(t, v)$ to 1, otherwise it does nothing at this step. When $\mathcal{Q} = \emptyset$, the algorithm returns $\mathcal{M}$.

As $\mathcal{P}$ can be determined in a sorted way in time $\mathcal{O}(m)$, this algorithm runs in time $\mathcal{O}(n\tau + m)$, where $\tau = |T|$, $n = |V|$, $m = |E|$, and where γ is a constant hidden in the $\mathcal{O}$.

Given a γ-matching $\mathcal{M}$, we define the *bottom temporal vertices* of $\mathcal{M}$, denoted $\mathtt{bot}(\mathcal{M})$, as the set $\{(t + \gamma - 1, u), (t + \gamma - 1, v) \mid \Gamma_\gamma(t, u, v) \in \mathcal{M}\}$. Lemma 1 shows the crucial role of the bottom temporal vertices of the matching returned by $\mathcal{A}$. We omit the proof of Lemma 1 because of the space restriction.

Lemma 1 *Let γ be a positive integer, let L be a link stream, and let $\mathcal{M}$ be a γ-matching returned by $\mathcal{A}$ when applied to L. If $\mathcal{M}'$ is a γ-matching of L, then every γ-edge of $\mathcal{M}'$ is incident to, at least, one temporal vertex of $\mathtt{bot}(\mathcal{M})$.*

Lemma 1 plays a cornerstone role in the proof of subsequent Theorem 2. As a byproduct, we also obtain the following result.

Theorem 1 *$\mathcal{A}$ is a 2-approximation of the γ-MATCHING problem.*

4 Kernelization algorithm

We now show a kernelization algorithm for γ-MATCHING by a direct pruning process based on Lemma 1. The main idea is as follows. First, we compute the set S of all bottom temporal vertices of a γ-matching produced by previously defined algorithm $\mathcal{A}$. Then, we prune the original instance by only keeping edges that belong to a γ-edge incident to a temporal vertex of S. More precisely, we prove the following result.

Theorem 2 *There exists a polynomial-time algorithm that for each instance (L, k), either correctly determines if L contains a γ-matching of size k, or returns an equivalence instance (L', k) such that the number of edges of L' is $2(k-1)(2k-1)\gamma^2$.*

Proof : Let $L = (T, V, E)$ be a link stream and k be an integer. We first run the algorithm $\mathcal{A}$ on L. Let $\mathcal{M}$ be the γ-matching outputed by the algorithm and let $\ell = |\mathcal{M}|$. If $\ell \geq k$, then we already have a solution and then we return a true instance. If $\ell < \frac{k}{2}$, then, by Theorem 1, we know that the instance does not contains a γ-matching of size k, and then we return a false instance. We now assume that $\frac{k}{2} \leq \ell < k$.

Lemma 1 justifies that we are now focusing on the temporal vertices of $\mathtt{bot}(\mathcal{M})$ in order to find the requested kernel. We construct a set $\mathcal{P}$ of γ-edges and we show that any edge e, that is not in a γ-edge of $\mathcal{P}$, is useless when looking for a γ-matching of size k. For each $(t, u) \in \mathtt{bot}(\mathcal{M})$, and for each t' such that $\max(0, t - \gamma + 1) \leq t' \leq t$, we consider the set $\mathcal{S}(t', u)$ of every γ-edge, existing in L, with the form $\Gamma_\gamma(t', u, v)$ with $v \in V$. If the set $\mathcal{S}(t', u)$ is of size at most $2k - 1$, we add every element of $\mathcal{S}(t', u)$ to $\mathcal{P}$. Otherwise, we select $2k - 1$ elements of $\mathcal{S}(t', u)$ that we add them to $\mathcal{P}$. In both cases, we denote by $\mathcal{S}'(t', u)$ the set of elements of $\mathcal{S}(t', u)$ that we have added to $\mathcal{P}$. This finish the construction of $\mathcal{P}$. As $|\mathtt{bot}(\mathcal{M})| = 2\ell$ and for each element of $\mathtt{bot}(\mathcal{M})$ we have added at most $(2k-1)\gamma$ γ-edges to $\mathcal{P}$, we have that $|\mathcal{P}| \leq 2\ell(2k-1)\gamma \leq 2(k-1)(2k-1)\gamma$.

We now prove that if L contains a γ-matching $\mathcal{M}'$ of size k, then it also contains a γ-matching $\mathcal{M}''$ of size k such that $\mathcal{M}'' \subseteq \mathcal{P}$. Let $\mathcal{M}'$ be a γ-matching of L of size k such that $p = |\mathcal{M}' \setminus \mathcal{P}|$ is minimum. We have to prove that $p = 0$. Assume that $p \geq 1$. Let Λ be a γ-edge in $\mathcal{M}' \setminus \mathcal{P}$. Let (t, u) be a temporal vertex of $\mathtt{bot}(\mathcal{M})$ that is incident to Λ. We know by Lemma 1 that this temporal vertex exists. Assume that $\Lambda = \Gamma_\gamma(t', u, v)$ for some $v \in V$ and some t' such that $\max(0, t - \gamma + 1) \leq t' \leq t$. As $\Lambda \notin \mathcal{P}$, we have that $\Lambda \in \mathcal{S}(t', u) \setminus \mathcal{S}'(t', u)$, and so $|\mathcal{S}'(t', u)| = 2k - 1$. Let $N_{\mathcal{S}'}(t', u)$ be the set of vertices w of $V \setminus \{u\}$ such that a γ-edge of $\mathcal{S}'(t', u)$ is incident to w. As $\mathcal{M}' \setminus \{\Lambda\}$ is of size $k - 1$, the γ-edges that it contains can be incident to at most $2k - 2$ vertices. This means that there exists $w \in N_{\mathcal{S}'}(t', u)$ such that no γ-edge of $\mathcal{M}' \setminus \{\Lambda\}$ is incident to w. Thus $(\mathcal{M}' \setminus \{\Lambda\}) \cup \{\Gamma_\gamma(t', u, w)\}$ is a γ-matching of size k. As $\Lambda \notin \mathcal{P}$ and $\Gamma_\gamma(t', u, v) \in \mathcal{P}$, this contradicts the fact that p is minimum.

We now can define the link stream $L' = (T, V, E')$ such that $E' = \{e \in E \mid \exists \Lambda \in \mathcal{P} : e \in \Lambda\}$. As $|\mathcal{P}| \leq 2(k-1)(2k-1)\gamma$ and every element of $\mathcal{P}$ is a γ-edge, we have that $|E'| \leq 2(k-1)(2k-1)\gamma^2$. The theorem follows. □

5   Conclusion and perspectives

We introduce the notion of a temporal matching in a link stream. Unexpectedly, the problem of computing a temporal matching, called γ-MATCHING, turns out to be NP-hard. We then show a kernelization algorithm for γ-MATCHING parameterized by the size of the solution. Our process produces quadratic kernels. On the way to obtaining the kernelization algorithm, we also provide a 2-approximation algorithm for γ-MATCHING. We believe that the same techniques extend to a large class of hitting set problems in the link streams.

References

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

[2] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[3] M. Latapy, T. Viard, and C. Magnien. Stream graphs and link streams for the modeling of interactions over time. 2017. https://arxiv.org/abs/1710.04073.

[4] T. Viard, M. Latapy, and C. Magnien. Computing maximal cliques in link streams. *Theoretical Computer Science*, 609:245–252, 2016.