Being an Influencer is Hard: The Complexity of Influence Maximization in Temporal Graphs with a Fixed Source

Argyrios Deligkas
Royal Holloway, University of London
Egham, United Kingdom
argyrios.deligkas@rhul.ac.uk

Eduard Eiben
Royal Holloway, University of London
Egham, United Kingdom
eduard.eiben@rhul.ac.uk

Tiger-Lily Goldsmith
Royal Holloway, University of London
Egham, United Kingdom
tigerlily.goldsmith@gmail.com

George Skretas
Hasso Plattner Institute, University of Potsdam
Potsdam, Germany
georgios.skretas@hpi.de

ABSTRACT

We consider the influence maximization problem over a temporal graph, where there is a single fixed source. We deviate from the standard model of influence maximization, where the goal is to choose the set of most influential vertices. Instead, in our model we are given a fixed vertex, or source, and the goal is to find the best time steps to transmit so that the influence of this vertex is maximized. We frame this problem as a spreading process that follows a variant of the susceptible-infected-susceptible (SIS) model and we focus on three objective functions. In the MaxSpread objective, the goal is to maximize the total number of vertices that get infected at least once. In the MaxViral objective, the goal is to maximize the number of vertices that are infected at the same time step. Finally, in MaxViralTstep, the goal is to maximize the number of vertices that are infected at a given time step. We perform a thorough complexity theoretic analysis for these three objectives over three different scenarios: (1) the unconstrained setting where the source can transmit whenever it wants; (2) the window-constrained setting where the source has to transmit at either a predetermined, or a shifting window; (3) the periodic setting where the temporal graph has a small period. We prove that all of these problems, with the exception of MaxSpread for periodic graphs, are intractable even for very simple underlying graphs.

KEYWORDS

Influence Maximization, Social Networks, Temporal Graphs, Computational Complexity, Parameterized Complexity

ACM Reference Format:
Argyrios Deligkas, Eduard Eiben, Tiger-Lily Goldsmith, and George Skretas. 2023. Being an Influencer is Hard: The Complexity of Influence Maximization in Temporal Graphs with a Fixed Source. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

"When is the right time to post our content online? Which days should we place our advertisements so our video goes viral? What's the best strategy to maximize the word-of-mouth effect?" These are some of the questions every advertiser, political party, and individual influencer try to answer. In every case, the goal is the same: maximize their *influence* over their social network.

By now, influence maximization is a well-established problem in Computer Science. The seminal paper of Kempe et al. [24] introduced the basic mathematical model of influence maximization and became the foundation for a plethora of follow-up models. There, the input consists of a static network and the task is to find a set of initial spreaders, or sources, that maximizes the expected outbreak size of a spreading process occurring over the network.

In real-world networks though, many situations and applications exhibit an inherited *temporal structure*: a person will check their social media just a few times during their day; a user will go through their favorite webpages every couple of days; people meet with their friends every few days. Furthermore, the effectiveness of word-of-mouth for a specific post/product/advert deteriorates as time passes if there are no interactions, or discussions, about it. Arguably, if a person has not recently seen a post from their favourite artist, they are unlikely to discuss with their friends a post that the artist made a long time ago. Motivated by the increase of social media influence in marketing strategies and the belief that the data provided will become more detailed in the future, the base model of [24] was suitably augmented in order to capture the above mentioned scenarios.

Temporal graphs [21, 23] form a solid basis that naturally captures the temporal structure of the aforementioned instances; in a temporal graph there is a fixed set of vertices and the connections between them change between consecutive time steps. Furthermore, when a temporal graph is coupled with a susceptible-infected-susceptible (SIS) spreading process, then it becomes an excellent framework to analyze the influence maximization problems described previously. So, now the task is not only to choose the vertices that will become the sources, but in addition decide *when*, i.e., at which time step, each source will become active [17]. In fact, marketing companies are very interested about the *"when-to-post"* problem, since different times of posting new content can yield vastly different results. This problem has already been studied both within computer science [30, 35] and in the field of marketing [22].

Since the influence maximization problem on temporal graphs is a generalization of the base model of [24], the NP-hardness results derived for the more constrained setting immediately apply here too. On the other hand, all the models so far assume that we can

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

choose each vertex only *once* [17]. Thus, the implied intractability of the problem comes from the choice of the sources and not from the activation times. In reality though, the set of possible sources someone can choose from is rather limited and not the whole set of vertices of the graph. In fact, there are several cases where there is only a single, *fixed* source; for example "youtubers" are independent and the only power they have is just to choose the time they will release their videos online. In this case, the NP-hardness from [24] does not apply any more. Our goal is to remedy this situation by establishing the tractability frontier for this scenario.

1.1 Our contribution

Our contribution is two-fold. Firstly, we formally define three different influence-maximization objectives on temporal graphs with a fixed source, where each variant captures a different aspect of the problem. Then, we perform a complexity-theoretic analysis for them, under three different variants that arise in real-life scenarios.

The input in each problem consists of a temporal graph, known in advance, a fixed source, and budget of allowed "posts", or *transmissions*. The goal is to choose a *transmission schedule*, i.e., the time steps the source transmits, in order to maximize an influence-related objective. We follow an SIS-style spreading process: an *inactive*, i.e., not currently influenced, vertex becomes *active* at time step $t + 1$ if at time step t is adjacent to an active vertex. Every active vertex is associated with a *counter* which shows for how many time steps it will remain active. If at time step t a vertex is adjacent to an active vertex, then its counter resets to a fixed number δ. The parameter δ resembles the maximum time an agent will be under the influence of the source, and thus the vertex will diffuse the information it has about the source.

The first objective, termed MaxSpread, aims to maximize the number of different vertices that became influenced at some point in time. This is probably the most natural objective, since it aims to maximize the exposure of the source to different customers. The goal of the second objective, termed MaxViral, is to create a "viral effect": maximize the number of simultaneously-active vertices. Finally, the target of the objective MaxViralTstep is to maximize the number of active vertices at a predetermined time step. For example, this objective is desirable when a political party wants to maximize the number of active voters on elections day, or a company wants to maximize its active customers on a product-launch day.

The first scenario we study is when there are no constraints on the transmission schedule, i.e., the source can transmit at any time step. In the second scenario, we consider transmission schedules that have to follow *window constraints*, a type of constraints that actually occur in real life. When a company chooses an influencer, they specify strict times and dates where each individual post has to be done. The timing mainly depends on abstract analytics that specify the influencer's highest engagement in the day but the data currently provided are not pinpoint and they are bundled in 3-hour slots. We consider two cases of window constraints. In the *fixed* window case, the time is split into intervals of length w and the source has to transmit exactly once in each window. In the (x, y)-*shifting* window case, any two consecutive transmissions of the source have to be at least x time steps apart and at most y time steps apart. In the last scenario, we consider *periodic* graphs. In this case the temporal graph has infinite lifetime with period $t_{\max}$. This means that the edges that are available at time step $i \in [1, t_{\max}]$, appear at time step $i + j \cdot t_{\max}$, for every $j \in \mathbb{N}$.

Our results. We perform a thorough complexity-theoretic study for the three objectives in each of the aforementioned scenarios. With the exception of MaxSpread for periodic graphs, we prove that the problems are intractable even for very restricted settings! In particular, we prove the following results. For the unconstrained setting, all three problems are NP-complete and W[2]-hard when parameterized by the number of transmissions of the source, even when the underlying graph is a binary tree. For the fixed window setting, all problems are NP-complete even when the window has size two. In addition, for MaxSpread the underlying graph is a star, at most three edges appear at any time step, and every edge appears at most twice. For MaxViral and MaxViralTstep the underlying graph is a subdivision of a star. For the shifting window setting, we prove that the problems are NP-complete for every $(2\delta, 4\delta)$-shifting window and the underlying graph is a star, or a subdivision of a star. Finally, for periodic graphs our results are as follows. For MaxSpread, we derive a fixed parameter tractable algorithm parameterized by $t_{\max}$; in other words, the problem is polynomial-time solvable when the period is constant. For problems MaxViral and MaxViralTstep, we prove that they are NP-hard even when $t_{\max} = 2$, i.e., the graph has period 2.

1.2 Related Work

After the seminal paper of Kempe et al. [24], influence maximization problems have received a tremendous amount of attention; see for example the surveys [4, 7, 20, 26, 31, 32, 37] and the references therein. More related, but still quite different, to our model are *time-aware* diffusion models, that study both discrete-time settings [8, 25, 27], or continuous-time models [28, 29, 34].

Closer to our work are the papers that studied networks that change over time - these networks/graphs can be either temporal, like ours, or graphs that evolve according to different procedures. Aggarwal et al. [1] propose efficient heuristics for finding a set of sources that maximises influence at a later time step. Zhang et al. [36] assume that the temporal graph is unknown and changes can only be detected periodically. Zhu et al. [39] introduce a continuous-time Markov chain model, an information diffusion model based on the independent cascade model [18, 19] and their experiments illustrate how to find a small set of sources. Wang and Street [33] focus on tracking the diffusion and aggregation of influence through the network in the context of viral marketing. They provide heuristics for finding a good initial set of influencers (referred to in their work as adopters). Another line of research considers location-aware influence maximisation, where the diffusion is focused on vertices located in certain areas rather than a general spread. Zhou et al. [38] focus on a distance-aware weighted model and go beyond generic influence maximisation algorithms, providing a location-based influence maximisation algorithm.

Recently, a new line of work established to capture the nature of real world social networks moving from static graphs to temporal graphs [6]. They study the complexity of competitive diffusion games [3, 15] and Voronoi games [2, 12].

In more recent work, Erkol et al. [13] studies influence maximisation under the susceptible-infected-recovered (SIR) model and analyses the performance of approximation algorithms over several temporal networks; in addition they study a special temporal network setting where the influence function is not submodular. In [14], the same set of authors answer some of their open questions and they find that greedy optimization is an effective method for finding a set of sources that has very high performance.

Statements where proofs or details are omitted due to space constraints are marked with ⋆. A version containing all proofs and details is provided as supplementary material.

2 PRELIMINARIES

For $n \in \mathbb{N}$, we denote $[n] := \{1, 2, \ldots, n\}$. A *temporal graph* $\mathcal{G} := \langle G, \mathcal{E} \rangle$ is defined by an *underlying graph* $G = (V, E)$ and a sequence of edge-sets $\mathcal{E} = (E_1, E_2, \ldots, E_{t_{\max}})$. It holds that $E = E_1 \cup E_2 \cup \cdots \cup E_{t_{max}}$. The *lifetime* of $\mathcal{G}$ is $t_{\max}$. An edge $e \in E$ has label i, if it is available at time step i, i.e., $e \in E_i$. In addition, an edge has k *labels*, if it appears in k edge-sets.

Spreading process. We follow a spreading process that resembles the *Susceptible Infected Susceptible* (SIS) model. In our model, we are given a fixed vertex $s \in V$, called the *source*. In addition, at every time step each vertex has a *state*, which is *active* or *inactive* and it is determined by a *counter* that gets integer values in $[0, \delta]$, for some given natural $\delta > 0$. The state of a vertex is active if $\delta > 0$ and inactive otherwise. Initially, every vertex is inactive. The source can choose which time steps to *transmit*. If s decides to transmit at time step t, then it sets its counter to δ, i.e. becomes active and it remains active until time step $t + \delta$. The remaining vertices evolve as follows.

- If at time step t there is a vertex $v \neq s$ that has an edge with an active vertex u, i.e., $uv \in E_t$, then at time step $t + 1$ the counter of v is set to δ.
- If at time step t there is a vertex v that is active and all of its adjacent vertices are inactive, i.e., all $u \in V$ with $uv \in E_t$ are inactive, then at time step $t + 1$ the counter of vertex v decreases by 1.

Observe that the procedure above allows "renewal" of the active state for a vertex. In other words, it resets the counter of a vertex to δ after any time step it is adjacent to an active vertex. In contrast, the standard SIS model does not allow renewals. We discuss the differences between our model and SIS in Section 6.

A *transmission schedule* $\mathcal{T} = (\tau_1, \tau_2, \ldots, \tau_b)$ is a set of time steps where the source transmits. Let $(\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)$ denote the set of active vertices at time step t under the transmission schedule $\mathcal{T}$ for source s, when the counter is δ.

Problems. We study three problems. In every case, the input is a temporal graph $\mathcal{G}$, a source s, a budget b and positive integers δ, and k.

- **MaxSpread.** Is there a transmission schedule $\mathcal{T}$, such that $\left|\bigcup_{t \in [t_{\max}]} (\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)\right| \geq k$ and $|\mathcal{T}| \leq b$? Put simply, the goal for MaxSpread is to maximize the number of different vertices that become active.
- **MaxViral.** Is there a transmission schedule $\mathcal{T}$, such that $\max_{t \in [t_{\max}]} |(\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)| \geq k$ and $|\mathcal{T}| \leq b$? So, the goal for MaxViral is to maximize the number of active vertices at any time step.
- **MaxViralTstep.** Here, we have in addition a time step $t^* \in [t_{\max}]$. Is there a transmission schedule $\mathcal{T}$, such that $|(\delta, \mathcal{T}) - \texttt{active}_{t^*}(\mathcal{G}, s)| \geq k$ and $|\mathcal{T}| \leq b$? In other words, the goal for MaxViralTstep is to maximize the number of active vertices at a given time step t^*.

Parameterized complexity. We refer to the standard books for a basic overview of parameterized complexity theory [9, 11]. At a high level, parameterized complexity studies the complexity of a problem with respect to its input size, n, and the size of a parameter k. A problem is *fixed parameter tractable* by k, if it can be solved in time $f(k) \cdot \texttt{poly}(n)$, where f is a computable function. If a problem is W[2]-hard with respect to k, then it is unlikely to be fixed parameter tractable by k.

SetCover. An instance of SetCover consists of a collection $S = \{S_1, S_2, \ldots, S_m\}$ of subsets of a set N of n elements, and a positive integer b. The task is to decide if there are b sets $T_1, T_2, \ldots, T_b$ in S, such that $T_1 \cup T_2 \cup \cdots \cup T_b = N$. SetCover is known to be W[2]-hard when parameterized by b [10].

3 UNCONSTRAINED SCHEDULES

In this section we study the complexity of the three objectives under unconstrained transmission schedules. We show that all of them are intractable, even on trees of maximum degree three. For clarity of exposition, when we write that "vertex u influences vertex y" we mean that vertex y became active via a sequence of vertex activations that includes vertex s.

3.1 MaxSpread

We prove that MaxSpread is NP-complete and W[2]-hard when parameterized by b. Containment in NP is straightforward, since given a transmission schedule, we can simulate the process and check whether every vertex becomes active at least once. Now, we describe the construction and prove some key properties our construction satisfies. Then we provide the proof of the main theorem of the section.

Construction. We reduce from SetCover. In what follows we will assume, that $n = 2^k$ for some positive integer k. Observe that this is without loss of generality, since we can augment any instance by adding a dummy set that contains the required number of elements, and by asking for a solution of size $b + 1$. We construct a perfect binary tree P_1 with n leaves, where the root of the binary tree is the source, s, and its leaves are $x_1, x_2, \ldots, x_n$ from "left to right" and let $h = \log 2n$ be the current height of the tree. Note that such a tree always has $2n$ vertices and is uniquely defined. For every $j \in [m]$ and for every $l \in [h]$, we add label $(l - 1)\delta + (j - 1)(\delta + 1) + 1$ to every edge of the tree between levels l and $l + 1$. Then we construct vertices $y_1, y_2, \ldots, y_n$ and for every $i \in [n]$, we add edge $x_i y_i$. Note now that P_1 has $h + 1$ levels. For every $i \in [n]$ and $j \in [m]$, if $i \in S_j$, we add label $h\delta + (j - 1)(\delta + 1) + 1$ to edge $x_i y_i$; see Figure 1.

Intuitively, the goal of this construction is to have the following properties:

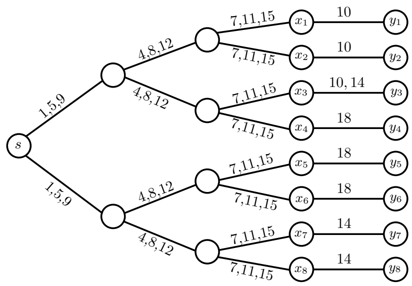

Figure 1: An example of the construction used in Theorem 3.4, when $\delta = 3$. The subsets of the SetCover problem are $S_1 = \{u_1, u_2, u_3\}$, $S_2 = \{u_3, u_7, u_8\}$, $S_3 = \{u_4, u_5, u_6\}$ and $\delta = 3$

- Each transmission of vertex s acts like a monotone wave, that goes from the vertex to the leaves, i.e., no vertex can influence its parent.
- For every $i \in [n]$ and $j \in [m]$, a transmission of vertex s will influence a vertex y_i only if s transmits at a time step $t_j = (j-1)(\delta+1)+1$, such that set S_j includes element i in the SetCover instance.

In the next part, we formally list these properties which are vital for the proof of theorem 3.4.

Lemma 3.1 (★). *Consider constructed graph P_1. Let $t_j = (j-1)(\delta+1)+1$. If vertex s transmits between time steps $t_j - (\delta - 1)$ and t_j, then every vertex $v \in C$ at level l will become active at time step $(l-2)\delta + (j-1)(\delta+1)+2$. Additionally the vertices become active once per such a transmission and an active vertex v in level l_1 can never influence a vertex w in level l_2, such that $l_2 < l_1$ (the transmission goes from the root of the tree towards the leaves).*

Proposition 3.2 (★). *Consider constructed graph P_1. For every $j \in [m]$, if vertex s transmits at time step t, such that $t_j - (\delta-1) \leq t \leq t_j$, then no other vertex will become active by this transmission.*

Lemma 3.3 (★). *Consider constructed graph P_1. For every $j \in [m]$ and $i \in [n]$, a vertex y_i will become active by the transmission of vertex s, only if the transmission starts between time steps $t_j - (\delta-1)$ and t_j, and only if $i \in S_j$.*

Theorem 3.4. *For any $\delta \geq 1$, MaxSpread is NP-hard and W[2]-hard when parameterized by b on tree graphs with degree 3.*

Proof. We claim that there exists a solution to MaxSpread on constructed graph P_1, such that s influences $3n$ vertices, if s transmits at $|b|$ time steps at most in the constructed graph, if and only if there is a solution $T = T_1 \cup T_2 \cup \cdots \cup T_b$ to the original SetCover instance.

Let $t_j = (j-1)(\delta+1)+1$. Assume that we have solution T to the SetCover set problem. We can construct a solution for MaxSpread by having vertex s transmit at each time step t_j, such that $S_j \in T$. We can guarantee that this is a solution for MaxSpread since every $i \in [n]$ is included in at least one set S_j and by Lemma 3.3 every vertex y_i will become active by at least one transmission of vertex s.

For the reverse direction, consider that we have a solution to MaxSpread which is a transmission schedule $\mathcal{T} = (\tau_1, \tau_2, \ldots, \tau_b)$. This means that if vertex s transmits at time steps $\tau_1, \tau_2, \ldots, \tau_b$, then every vertex in the graph will become active including vertices $y_1, y_2, \ldots, y_n$. To construct a solution for the SetCover problem we do the following. For every $t \in [b]$ and every $j \in [m]$, if $t_j - (\delta-1) \leq \tau_t \leq t_j$, we add set S_j to solution T of the SetCover problem (ignoring duplicate additions). By definition, the size of T is at most b. Note also that every element $i \in [n]$ is included in T since every vertex y_i becomes active by at least one transmission of vertex s and by Proposition 3.2 and Lemma 3.3 this only happens when $i \in S_j$. This completes the proof. □

3.2 MaxViral and MaxViralTstep

We will now show that MaxViral is NP-complete and W[2]-hard when parameterized by b on tree graphs with degree 3. Containment in NP is straightforward, since given a transmission schedule, we can simulate the process and check the maximum active vertices for any time step. We will use a similar construction to the previous theorem and prove some key properties that we need; the construction differentiates in several parts in order to accommodate the different objectives. The proof will be again via a reduction from SetCover. Note also that the following construction/proofs can also be easily modified to show that MaxViralTstep is NP-hard and W[2]-hard and as such, we will not provide one.

Construction. We reduce from SetCover. We construct a perfect binary tree P_2 with n leaves, where the root of the binary tree is called s and the leaves of the binary tree are called $x_1, x_2, \ldots, x_n$ from "left to right" and let $h = \log 2n$ be the height of the current tree. Note that we will not update the value of h once more vertices are added to the tree. Also, note that such a tree always has $2n-1$ vertices and is uniquely defined. Let set C contain every vertex of the current graph. For every $j \in [m+1]$ and for every $l \in [h]$, we add label $(l-1)\delta + h(j-1)(\delta+1)+1$ to every edge of the tree between levels l and $l+1$. Then we construct vertices $y_1, y_2, \ldots, y_n$ and vertices $z_1, z_2, \ldots, z_n$ and for every $i \in [n]$, we add edge x_iy_i and y_i, z_i. For every $i \in [n]$ and $j \in [m+1]$, if $i \in S_j$, we add label $h\delta + h(j-1)(\delta+1)+1$ to edge x_iy_i. For every $i \in [n]$ and $j \in [h(m+1)]$, we add labels $(j-1)(\delta+1)+1$, $(j-1)(\delta+1)+2$ to edge y_iz_i. This completes the construction; see Fig. 2 for an example.

Intuitively, the goal of this construction is to have the following properties:

- Each transmission of vertex s acts like a monotone wave, that goes from the vertex to the leaves, i.e., no vertex can influence its parent.
- For every $i \in [n]$ and $j \in [m]$, a transmission of vertex s will influence a vertex y_i only if s transmits at a time step $t_j = h(j-1)(\delta+1)+1$, such that set S_j includes element i in the SetCover instance.
- For every $i \in [n]$ and $j \in [m+1]$, once vertex y_i is influenced at some time step $h\delta + h(j_1-1)(\delta+1)+1$ by a transmission from vertex s, vertex y_i, z_i will always be active at time step $h\delta + h(j-1)(\delta+1)+3$, where $j_1 < j$.

Next, we formally list these properties which are vital for the proof of Theorem 3.10.

Lemma 3.5 (★). *Consider constructed graph P_2. Let $t_{j_1} = h(j-1)(\delta+1)+1$. If vertex s transmits between time steps $t_{j_1} - (\delta-1)$ and t_{j_1}, then every vertex $v \in C$ at level l will become active at time*

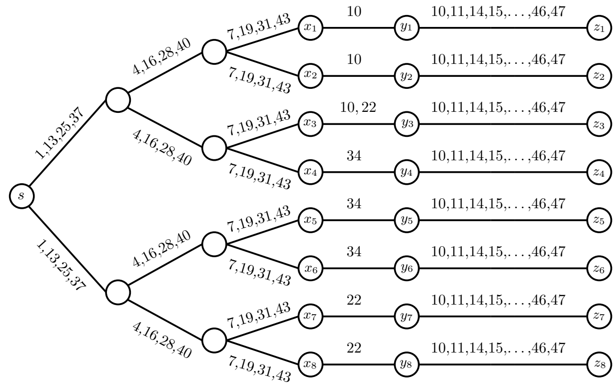

Figure 2: An example of the construction used to prove that MaxViral is W[2]-hard, when $\delta = 3$. The subsets of the SetCover problem are $S_1 = \{u_1, u_2, u_3\}$, $S_2 = \{u_3, u_7, u_8\}$, $S_3 = \{u_4, u_5, u_6\}$,

step $(l-2)\delta + t_{j_1} + 2$. Additionally vertices $v \in C$ become active once per such a transmission and an active vertex v in level l_1 can never influence a vertex w in level l_2, such that $l_2 < l_1$ (the transmission goes from the root of the tree towards the leaves).

Corollary 3.6 ($\star$). *Consider any pair of vertices $v, w \in C \setminus (s)$, where v, w belong to different levels of P_2. There exists no such pair of vertices such that both vertices can be active at the time step.*

Proposition 3.7 ($\star$). *Consider constructed graph P_2. For every $j \in [m+1]$, if vertex s transmits at time step t, such that $t_j - (\delta - 1) \le t \le t_j$, then no other vertex will become active by this transmission.*

Lemma 3.8 ($\star$). *For every $i \in [n]$, vertex y_i cannot influence vertex x_i.*

Lemma 3.9 ($\star$). *Consider constructed graph P_2. Let $t_j = h(j-1)(\delta+1)+1$. For every $j \in [m+1]$ and $i \in [n]$, a vertex y_i will become active at time step $h\delta + t_{j_1} + 3$ by the transmission of vertex s, only if the transmission starts between time steps $t_{j_1} - (\delta - 1)$ and t_{j_1}, where $j_1 < j$, and only if $i \in S_j$. Additionally, once vertex y_i becomes active at time step $h\delta + t_{j_1} + 3$ by the transmission of vertex s, vertices y_i, z_i will be active at every time step $h\delta + t_{j_1} + 3 > h\delta + t_j + 2$, for every $j \in [m+1]$.*

Theorem 3.10. *For any $\delta \ge 1$, MaxViral is* NP*-hard and* W[2]*-hard when parameterized by b on tree graphs with degree 3.*

Proof. We claim that there exists a solution to MaxViral on constructed graph P_2, such that we maximize the number of vertices that are active at any one time step t if s transmits at $|b|$ time steps at most in the constructed graph, if and only if there is a solution $T = T_1 \cup T_2 \cup \cdots \cup T_b$ to the original SetCover instance.

First note, that due to Corollary 3.6, the maximum vertices that can be influenced in the graph is exactly the set that contains vertices x_i, y_i, z_i. Let $t_j = h(j-1)(\delta+1)+1$. Assume that we have solution T to the SetCover set problem. We split, the analysis into two cases: (i) $1 \le \delta \le 2$ and (ii) $\delta \ge 2$. For case (i), we can construct a solution for MaxViral by having vertex s transmit at each time step t_j, such that $S_j \in T$. We can guarantee that this is a solution for MaxViral since every $i \in [n]$ is included in at least one set S_j and by Lemma 3.9, for every $i \in [n]$, every vertex y_i will become active by at least one transmission of vertex s. Additionally, due to Lemma 3.9, every vertex x_i, z_i will be active at every time step $h\delta + t_{j_1} + 3$ for $t_{j+1} > t_j$. Finally, s performs one final transmission at $(m+1-1)(\delta+1)+1$, so that all vertices x_i, y_i, z_i are active at time step $h\delta + h(m+1-1)(\delta+1)+3$. For case (ii), assume that the last set is called S_f. We create the same solution as case (i) but we do not add the final transmission. This is because once vertex s transmits at time step $(f-1)(\delta+1)+1$, at time step $h\delta + h(f-1)(\delta+1)+3$, every vertex x_i will be active by the transmission at time step $(f-1)(\delta+1)+1$, or by a previous transmission. This was not true for case (i), because influence time is small, and vertices x_i will have stopped being active at time step $h\delta + h(f-1)(\delta+1)+3$ by the transmission that fired at time step $(f-1)(\delta+1)+1$.

For the reverse direction, we again split the analysis in two cases: (i) $1 \le \delta \le 2$ and (ii) $\delta \ge 2$. Consider that we have a solution to MaxViral which is a transmission strategy $\mathcal{T} = (\tau_1, \tau_2, \ldots, \tau_b)$. For case (i) this means that if vertex s transmits at time steps $\tau_1, \tau_2, \ldots, \tau_b$, then due to Lemma 3.9, vertices x_i, y_i, z_i in the graph will be active at time step $h\delta + h(m+1-1)(\delta+1)+3$. To construct a solution for the SetCover problem we do the following. For every $t \in [b-1]$ and every $j \in [m]$, if $t_j - (\delta - 1) \le \tau_t \le t_j$, we add set S_j to solution T of the SetCover problem (ignoring duplicate additions). We do not add the last set b because this transmission was used only to influence vertices x_i. By definition the size of T is at most b. Note also that every element $i \in [n]$ is included in T since every vertex y_i, z_i becomes active by at least one transmission of vertex s and by Proposition 3.7 and Lemmas 3.8,3.9 this only happens when $i \in S_j$. For case (ii), we do the exactly the same but we also add set b to the solution. □

3.3 Approximation Algorithm

In this subsection, we show that all of the problems we study admit a constant factor approximation. That is, while the problems are NP-hard, we will show that for each of the problems there exists a polynomial time algorithm that either finds a transmission schedule $\mathcal{T} = (\tau_1, \tau_2, \ldots, \tau_b)$ that achieves at least $1 - \frac{1}{e} \approx 0.632$ fraction of the influence target k, or we correctly output that the influence target k cannot be achieved. More precisely, we give a reduction from each of our problems to MaximumCoverage (the maximization version of SetCover) that preserves both influence target k and budget b; here b will be the number of sets we wish to select and k the number of elements of the universe we wish to cover. Moreover, we show that there is one-to-one correspondence between transmission schedules and the collections of selected sets.

The following lemma serves as the main tool in this section. It basically states that if we have some transmission schedule $\mathcal{T}$, then the set of all vertices active at some time step t is precisely the union over all $\tau \in \mathcal{T}$ of the vertices active at time step t if we only transmit at time step τ.

Lemma 3.11 ($\star$). *Given a temporal graph $\mathcal{G} := \langle G, \mathcal{E} \rangle$ with lifetime $t_{\max}$, a source vertex s, a time step $t \in [t_{\max}]$, integer $\delta > 0$, a transmission schedule $\mathcal{T}$, it holds that $(\delta, \mathcal{T})\text{-}\mathtt{active}_t(\mathcal{G}, s) = \bigcup_{\tau \in \mathcal{T}} (\delta, \tau)\text{-}\mathtt{active}_t(\mathcal{G}, s)$.*

Given the above lemma, it is rather straightforward to construct an instance of SetCover or MaximumCoverage from each of our problems. Before we do so, let us formally define the MaximumCoverage problem. An instance of MaximumCoverage consists of a collection $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ of subsets over some universe U and a positive integers b and k, and we need to decide if there is a sub-collection $\mathcal{T} \subseteq \mathcal{S}$ of size b, such that $|\bigcup_{S\in\mathcal{T}} S| \geq k$. To obtain our results we use the well-known result that for MaximumCoverage, there is a polynomial-time algorithm that either outputs a collection $\mathcal{T} \subseteq \mathcal{S}$ such that $|\mathcal{T}| = b$ and $|\bigcup_{S\in\mathcal{T}} S| \geq (1 - \frac{1}{e}) \cdot k$ or correctly outputs that no collection of at most b sets covers at least k elements.

Theorem 3.12 ($\star$). *There is a polynomial-time algorithm that finds a $(1 - \frac{1}{e})$-approximate solution for MaxSpread, MaxViral, and MaxViralTstep.*

4 WINDOW CONSTRAINED SCHEDULES

In this section we consider two types of so called "window constraint" schedules, where we are only interested in transmission schedules satisfying some additional constraints. First we study fixed-window schedules. There the lifetime of the temporal graph is split into a number of disjoint time intervals and the transmission schedule needs to have exactly one transmission in each of the intervals. Then, we shift our attention to (x, y)-shifting window schedules, where the difference between two consecutive transitions should be between x and y. Both scenarios are associated with a new natural parameter: the *size*, w, of the window.

Note, if we do not restrict size of the window, then the results from the previous section extend rather straightforwardly. To see this, consider the fixed window case. Assume that we have an instance of MaxSpread for the unconstrained setting, on the temporal graph $\mathcal{G} = \langle G, \mathcal{E}\rangle$. Then, we could just create a new temporal graph $\mathcal{G}' = \langle G', \mathcal{E}'\rangle$ such that $G' = G$ and $\mathcal{E}'$ is a concatenation of b copies of the sequence $\mathcal{E}$. We then set the windows to have size $t_{\max}$ each, i.e., first window contains time steps 1 to $t_{\max}$, second window from $t_{\max}$ to $2t_{\max}$ and so on. It is rather straightforward to see that this reduction immediately gives hardness for MaxSpread. It is also not too hard to verify that using similar reductions as in previous section would give hardness for MaxViral and MaxViralTstep, when the size of the window is not bounded by a constant. We can easily get hardness for shifting window schedules, using an argument that is nearly the same. This time, between the two consecutive copies of $\mathcal{E}$, we introduce $t_{\max}$ many time steps without any edge, and we let $x = t_{\max}$ and $y = 2t_{\max}$.

On the other hand, if both the budget, b, and window size, w, were parameters, then we would immediately get that the lifetime is bounded by b times the (max) window size (or $y + 1$ in the case of (x, y)-shifting windows), so an exhaustive search already gives an algorithm running in time $\binom{b \cdot w}{b} \cdot \texttt{poly}(|V(G)| \cdot t_{\max})$, where w is the size of the window. For this reason, in the rest of the section we will consider the cases, where the budget is large (or unrestricted) and the size of the window is small. We will show that, unfortunately, the problem remains NP-hard even for constant size windows.

4.1 Fixed Window Schedules

In this section we prove that all three problems are NP-hard even for very restricted settings.

Theorem 4.1 ($\star$). *MaxSpread with fixed window constraints is NP-hard for every $\delta \geq 1$ even when window size is 2δ, in every time step there are at most 3 active edges, every edge is active at most twice, and the underlying graph is a star with center the source s.*

Proof. We show this by reduction from VertexCover on graphs with maximum degree three [16]. In the VertexCover problem, we are given a graph $H = \langle V, E\rangle$ and an integer ℓ, and the question is whether there exists a set of vertices S such that $|S| \leq \ell$ and for all $e \in E$ we have $|e \cap S| \geq 1$.

Now, let $\langle H, \ell\rangle$ be an instance of VertexCover such that the degree of every vertex $h \in V(H)$ is at most three. We construct a temporal graph $\mathcal{G} = \langle G, \mathcal{E}\rangle$ as follows. First, for the sake of presentation of the proof, let $|V(H)| = n$, $|E(H)| = m$, and let us order the vertices and edges of H in an arbitrary but fixed order. That is let $V(H) = \{h_1, h_2, \ldots, h_n\}$ and $E(H) = \{e_1, e_2, \ldots, e_m\}$.

The vertices of $\mathcal{G}$ are as follows:

- the source vertex s;
- the set V_E containing m vertices, such that for every $e_j \in E(H)$ there is a vertex $v_j \in V_E$;
- the set U containing n vertices, such that for every $h_i \in V(H)$, there is a vertex $u_i \in U$.

The underlying graph G is then a star with center the vertex s and $t_{\max} = 2\delta \cdot (|V(H)| + 1)$. Every window will consist of 2δ time steps and will be associated with a single vertex of $V(H)$. So, for every $i \in [n]$ the window from time step $2\delta \cdot (i - 1) + 1$ to time step $2\delta \cdot i$ is associated with vertex h_i. The last window is a "dummy" window that is only necessary in the case of $\delta = 1$. In this case, we might want to transmit at the time step $2\delta \cdot |V(H)|$ to activate some vertices, which we can only do if there is one more time step to activate the vertices adjacent to s at time step $2\delta|V(H)|$. Consider now the window associated with h_i. At the first step inside this window, i.e. at $2\delta(i-1)+1$, there will be edges between s and every $v_j \in V_E$ such that h_i is incident with the edge e_j. Note that since the degree of h_i is at most 3, at most 3 edges have label $2\delta(i-1)+1$. Moreover, in $(\delta+1)$-th time step inside the window $(2\delta(i-1)+\delta+1)$ there is an edge between s and u_i. That is exactly one edge – the edge su_i – has the label $2\delta(i - 1) + \delta + 1$.

Given the above construction of the temporal graph $\mathcal{G}$, we let $k = n+m-\ell$ and we claim that there exists a window constraint transmission schedule $\mathcal{T}$ such that $\left|\bigcup_{t\in[t_{\max}]}(\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)\right| \geq k$ if and only if H admits a vertex cover with at most ℓ vertices. To see this, we only need to show that it only makes sense to transmit in 1st or $\delta + 1$st step in each window. Given this, we then observe that transmitting in the 1st time steps in the windows associated with the vertices of a vertex cover S and in $\delta + 1$st windows in the remaining time steps activates all but $|S|$ many vertices in U (precisely the vertices associated with the vertices in S). ◻

Similarly as in the previous section, we can rather straightforwardly modify the above proof to obtain the NP-hardness for MaxViral and MaxViralTstep.

Theorem 4.2 (★). *MaxViral and MaxViralTstep with fixed window constraints are both* NP*-hard even when the window size is* 2δ*, for every* $\delta \geq 1$.

4.2 Shifting Window Schedules

The hardness results for the shifting windows case follow rather easily from the proofs in the previous section. For MaxSpread, we follow the same reduction as in Theorem 4.1, but: we add additional δ time steps between any two consecutive windows; each time step will be empty, i.e. it will not contain any edges; we ask for $(2\delta, 4\delta)$-shifting window. This guarantees that we can always transmit in every window of size 3δ, either at time step $3i\delta + 1$ or at time step $3i\delta + \delta + 1$, which are the only two time steps within the window than contain some edges. Moreover, the lower value of the window, 2δ, guarantees that s cannot be active during both of the time steps within one window.

Theorem 4.3. *MaxSpread with* $(2\delta, 4\delta)$*-shifting window constraints is* NP*-hard, for every* $\delta \geq 1$*, even when in every time step there are at most* 3 *active edges, every edge is active at most twice, and the underlying graph is a star with the center in the source* s.

Following a similar argument to Theorem 4.2 , we get.

Theorem 4.4. *MaxViral and MaxViralTstep* $(2\delta, 4\delta)$*-shifting window constraints is* NP*-hard for every* $\delta \geq 1$.

5 SCHEDULES ON PERIODIC GRAPHS

In this section we investigate the complexity of our three problems on periodic temporal graphs. A periodic temporal graph is given to us in the exactly same way as the temporal graph defined in Section 2. The only difference is that after reaching the time step $t_{\max}$ instead of stopping the time, the temporal graph repeats itself from time step 1. More precisely, given a *temporal graph* $\mathcal{G} := \langle G, \mathcal{E} \rangle$, where $\mathcal{E} = (E_1, E_2, \ldots, E_{t_{\max}})$, if we say that $\mathcal{G}$ is a periodic graph, then edge with label $i \in [t_{\max}]$ is available not only in time step i but in every time step $i + j \cdot t_{\max}$, where $j \in \mathbb{N}$. In this case, we call $t_{\max}$ the *period* of $\mathcal{G}$.

Note that while in previous sections we only needed to simulate the spreading process until time step $t_{\max}$, this is now not the case and the spreading process continues infinitely. However, our questions still make sense and we can even upper bound the number of steps we need to simulate. It is not so difficult to see that there are at most $(\delta + 1)^{|V(G)|}$ different combinations of counter values on the vertices of the graph G. Thus, after simulating at most $(\delta + 1)^{|V(G)|} \cdot t_{\max}$ steps of the spreading process, some combination of counters will appear at time steps $i + j \cdot t_{\max}$ and $i + j' \cdot t_{\max}$ for some $i \in [t_{\max}]$. Since the process is deterministic, if there is no new transmission between these two time steps, we end up repeating exactly the same sequence of combinations of counter values (and hence active vertices), as we have between these two time steps. Therefore, the time between any two consecutive transmissions in a solution to any of our problems, does not need to be more than $(\delta+1)^{|V(G)|} \cdot t_{\max}$ steps apart. Thus, for any such solution we need to simulate $(\delta+1)^{|V(G)|} \cdot t_{\max} \cdot (b+1)$ steps, where b is the budget we are given.

In what follows, we consider the situation when the period $t_{\max}$ is small, i.e., we study the parameterized complexity of the problem

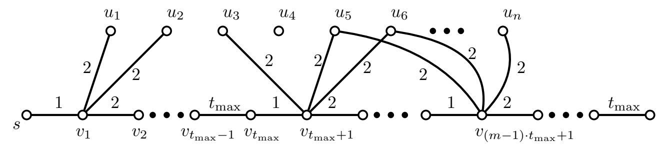

Figure 3: Illustration of the reduction described in Theorem 5.2. There is a path $s, v_1, \ldots, v_{mt_{\max}}$ **with cycling labels** $1, 2, \ldots, t_{\max}, 1, 2 \ldots$ **and vertices** $u_1, \ldots, u_n$**. Vertex** v_1 **is adjacent to** u_1 **and** u_2 **with label** 2 **on the two edges representing** $S_0 = \{1, 2\}$**. Similarly,** $v_{t_{\max}+1}$ **is adjacent to** u_3, u_5, u_6 **with label** 2**, representing** $S_1 = \{u_3, u_5, u_6\}$ **and so on.**

with respect to the parameter $t_{\max}$. Before we prove our results, we would like to highlight that restricting the period is necessary if we hope to get any positive results for the problem. This is because adding $\delta + 1$ new labels $E_{t_{\max}+1}, E_{t_{\max}+2}, \ldots, E_{t_{\max}+\delta+1}$ where $E_j = \emptyset$ for all j between $t_{\max} + 1$ and $t_{\max} + \delta + 1$, gives us a periodic graph in which all vertices (other than s) are always inactive when the period starts. Hence, the idea described above effectively reduces the problem on periodic temporal graphs to the problem when we are given non-periodic temporal graphs.

Let us now consider MaxSpread. Since we only care about which vertices are activated at least once, but do not care about synchronicity, we immediately get from the application of Lemma 3.11 that we only need to consider transmitting within the first period. So, if $b \geq t_{\max}$, then clearly an optimal solution is to just transmit $t_{\max}$ times in total – in every time step within the first period. On the other hand, if $b < t_{\max}$, then we can enumerate all $\binom{t_{\max}}{b} < 2^{t_{\max}}$ possibilities to transmit in b different time steps within the first period. However, there is still one issue to resolve: our upper bound $(\delta + 1)^{|V(G)|} \cdot t_{\max} \cdot (b + 1)$ on the number of steps we need to simulate is too large. In the next theorem we prove that, if we only care about whether a vertex has been activated or not, then we can overcome this large upper bound and show that the number of steps we need to simulate is actually only at most $(|V(G)| + 1) \cdot t_{\max} \cdot \delta$.

Theorem 5.1 (★). *MaxSpread can be solved in* $2^{t_{\max}} \cdot \texttt{poly}(|V(G)|)$ *time on a periodic temporal graph* $\mathcal{G} = \langle G, \mathcal{E} \rangle$ *with period* $t_{\max}$.

Notice that if $\delta > t_{\max}$, then every vertex with at least one neighbor in the underlying graph will remain active forever after being activated. Therefore, in this case, both MaxViral and MaxViralTstep always require budget at most two: transmitting at time step 1 will make every connected component of $G - \{s\}$ active by time step $|V(G)| \cdot t_{\max}$; then transmit one more time to activate the singletons in $G - \{s\}$. On the other hand, perhaps surprisingly, we will show that the problem is NP-hard whenever $\delta < t_{\max}$.

Theorem 5.2 (★). *MaxViral and MaxViralTstep are* NP*-hard and* W[2]*-hard parameterized by the budget* b*. Moreover, this holds for every pair of* $\delta, t_{\max}$*, such that* $\delta < t_{\max}$*,* $t_{\max} \geq 2$ *and* $\delta \geq 1$.

Proof. Let $\langle \mathcal{S}, N, b \rangle$ be an instance of SetCover such that $\mathcal{S} = \{S_0, S_1, \ldots, S_{m-1}\}$. Without loss of generality let us assume that $N = [n]$ for some $n \in \mathbb{N}$. Given any $\delta, t_{\max}$, such that $\delta < t_{\max}$, $t_{\max} \geq 2$, $\delta \geq 1$, we are going to construct an input $\langle \mathcal{G}, s, \delta, b, k \rangle$ for MaxViral and MaxViralTstep such that if $\mathcal{S}$ admits a set cover of size at most b, then there exists a transmission schedule

$\mathcal{T}$ of size at most b such that $|(\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)| \geq k$, where $t = (m-1) \cdot t_{\max} + \delta + 2$ and $\max_{t \in [t_{\max}]} |(\delta, \mathcal{T}) - \texttt{active}_t(\mathcal{G}, s)| < k$ otherwise. We set $k = n + b \cdot \delta$.

The temporal graph $\mathcal{G}$ is constructed as follows. See also Figure 3 for the illustration of the reduction. The vertex set of $\mathcal{G}$ consists of: 1) source vertex s; 2) $q = m \cdot t_{max}$ many vertices $V_1 = \{v_1, v_2, \dots, v_q\}$; 3) vertices $V_2 = \{u_1, u_2, \dots, u_n\}$, and the edges of $\mathcal{G}$ are: 1) edge sv_1 with label 1; 2) for every $i \in [q-1]$, edge $v_i v_{i+1}$ with label (i mod $t_{\max}$) + 1; 3) for every set $S_j \in \mathcal{S}$ and every $i \in S_j$, there is an edge between $v_(j \cdot t_{\max}) + 1$ and u_i with label 2.

Since edges $v_j u_i$ have all label 2 and $\delta < t_{\max}$, it is easy to see that vertices in V_2 can only be activated in time step $\ell \cdot t_{\max} + 3$ for some $\ell \in \mathbb{N}$ and it will be active in time steps $\ell \cdot t_{\max} + 3, \ell \cdot t_{\max} + 4, \dots, \ell \cdot t_{\max} + \delta + 2$. Similarly, v_i for $i \in [q]$ can only be activated in time step $\ell \cdot t_{\max} + (i \mod t_{\max}) + 1$ from v_{i-1} (where $v_0 = s$). It is also straightforward to verify that we can assume that the source s transmits only at time steps τ such that $\tau = \ell \cdot t_{\max} + 1$. Now, if s transmits at $\tau = \ell \cdot t_{\max} + 1$, then at time step t, only the vertices $v_{t-\tau-\delta}, \dots, v_{t-\tau-1}$ in V_1 are active. Moreover, if $t - \tau = j \cdot t_{\max} + r$, where $r \in \{2, 3, \dots, \delta + 1\}$, then at time step t, the vertices $u_i \in V_2$ such that $i \in S_j$ are also active at the time step t and these are the only vertices of V_2 that are activated at time step t by the transmission at the time step τ.

Given the above discussion and Lemma 3.11, we can quite easily argue that a transmission schedule $\mathcal{T} = \{\tau_1, \dots, \tau_b\}$ that activates at least $k = n + b \cdot \delta$ many vertices at time step t, then we can construct a set cover of $\mathcal{S}$ of size b. Namely, it follows that if we transmit at most b times, then at most $b \cdot (\delta - 1)$ many vertices in V_1 can be active at the same time and in order for $k = n + b \cdot \delta$ many vertices to be active at time step t, all the vertices in V_2 have to be active. It follows that $\mathcal{T} = \{\tau_1, \dots, \tau_b\}$ such that for all $i \in [b]$, $t - \tau_i = j_i \cdot t_{\max} + r_i$, where $r_i \in \{2, 3, \dots, \delta + 1\}$, and $\bigcup_{i \in [b]} S_{j_i} = [n]$, therefore $\mathcal{S}$ admits a set cover of size at most b.

On the other hand, we can easily revert the above construction and postpone the first transmission such that the active vertices on the path are in the blocks representing set cover exactly at time step $t = (m-1) \cdot t_{\max} + \delta + 2$, at this time step all vertices in U activated by the first vertex in these blocks are still active. □

6 DISCUSSION

In this paper we have explored the complexity of influence maximization on temporal graphs with a single fixed source. We have focused on three objectives, MaxSpread, MaxViral, and MaxViralTstep, under three different settings, which are naturally motivated by real life scenarios. We have proved that in almost every case, the problem is intractable.

In this section we discuss the connections our model has with some other problems arising in temporal graphs; we compare our spreading dynamics with the "standard" SIS model; and we highlight some open questions that deserve extra study.

Comparison with the SIS model. As we have explained in Section 2, the spreading dynamics we consider allow, "renewal" of the influence. In other words, a vertex will reset its counter to δ every time it is adjacent to an active vertex, even if it is *already* active. In contrast, the original SIS model does not allow this; a vertex u becomes active at time step $t + 1$ only if at time step t, vertex u

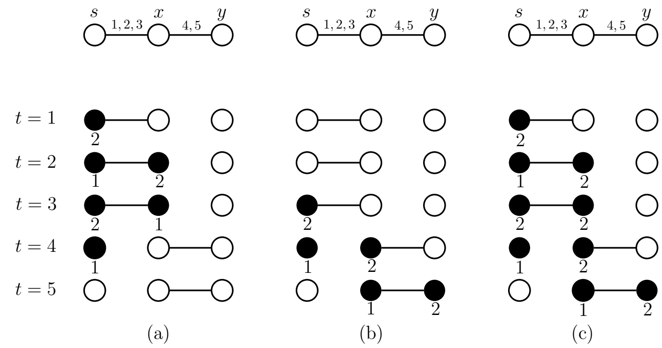

Figure 4: Comparison between transmission schedules for SIS and our spreading dynamics where $\delta = 2$, on the temporal path with vertices s, x, y. (a) SIS dynamics with (a) $\mathcal{T} = (1, 3)$; (b) SIS dynamics with $\mathcal{T} = (3)$; (c) our dynamics with $\mathcal{T} = (1, 3)$. Solid vertices depict active vertices. The number below each vertex is its counter at the specific timestep.

is (a) inactive and (b) adjacent to an active vertex. This difference makes the two processes behave very differently. In fact, under the SIS model, the size of active vertices does not monotonically increase with the number of transmissions; Figure 4 demonstrates this. It is not too difficult to verify that all of our hardness results, both NP-hardness and W[2]-hardness, apply under the SIS spreading model. This is because the structure of our instances is designed in a way that does not allow for renewal. On the other hand though, our positive result for periodic graphs is no longer valid under SIS model. The complexity of the problem is an interesting question, mainly because of the non-monotonicity of the active vertices when SIS is used.

Open Problem 1. What is the complexity of MaxSpread for periodic graphs under the SIS spreading model?

Connection to restless temporal walks. A *temporal walk* in $\langle G, \mathcal{E} \rangle$ from vertex v_1 to vertex v_w is a sequence of edges $W = (v_i v_{i+1}, t_i)_{i=1}^{w-1}$ such that for every $i \in [w]$ it holds that $v_i v_{i+1} \in E_{t_i}$, i.e. $v_i v_{i+1}$ is available at time step t_i and time steps are strictly increasing, i.e. if $i < j$ then $t_i < t_j$. Observe that a vertex v can appear multiple times in a temporal walk; in a temporal path this is *not* allowed. A *temporal walk* $W = (v_i v_{i+1}, t_i)_{i=1}^{w-1}$ is called δ-restless if $t_i < t_{i+1} < t_i + \delta$, for every $i \in [w]$. Then observe that (δ, τ)-$\texttt{active}_t(\mathcal{G}, s)$ is equal to the set of vertices for which there exists a (δ, τ)-restless temporal walk from s with arrival time between t and $t - \delta$. Restless walks have been studied in the past [5], albeit from a different point of view compared to ours.

Open questions. We have resolved the complexity of the three objectives for almost every class of graphs. Though, there exist some intriguing questions that will complete the complexity-landscape of the problem.

Open Problem 2. Do MaxViral and MaxViralTstep on periodic graphs belong to NP, or are they complete for some other class, like PSPACE?

Observe that the infinite lifetime of the graph makes the problem of verifying objectives MaxViral and MaxViralTstep non trivial. An intermediate question is the following, since our hardness reduction cannot be trivially extended in order to resolve it.

Open Problem 3. Are MaxViral and MaxViralTstep on periodic graphs NP-hard when $\delta = t_{\max}$, i.e. when δ equals the period of the graph?

The last question is concerned about the case where the underlying graph is a path. Some initial observations show that MaxSpread in the unconstrained setting is tractable. However, for the remaining combinations of objectives and transmission schedules, the problem remains wide open.

Open Problem 4. What is the complexity of the three objectives when the underlying graph is a path?

REFERENCES

[1] Charu C Aggarwal, Shuyang Lin, and Philip S Yu. 2012. On influential node discovery in dynamic social networks. In *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 636–647.

[2] Hee-Kap Ahn, Siu-Wing Cheng, Otfried Cheong, Mordecai Golin, and Rene Van Oostrum. 2004. Competitive facility location: the Voronoi game. *Theoretical Computer Science* 310, 1-3 (2004), 457–467.

[3] Noga Alon, Michal Feldman, Ariel D Procaccia, and Moshe Tennenholtz. 2010. A note on competitive diffusion through social networks. *Inform. Process. Lett.* 110, 6 (2010), 221–225.

[4] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the myths of influence maximization: An in-depth benchmarking study. In *Proceedings of the 2017 ACM international conference on management of data*. 651–666.

[5] Matthias Bentert, Anne-Sophie Himmel, André Nichterlein, and Rolf Niedermeier. 2020. Efficient computation of optimal temporal walks under waiting-time constraints. *Applied Network Science* 5, 1 (2020), 1–26.

[6] Niclas Boehmer, Vincent Froese, Julia Henkel, Yvonne Lasars, Rolf Niedermeier, and Malte Renken. 2021. Two Influence Maximization Games on Graphs Made Temporal. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Zhi-Hua Zhou (Ed.). International Joint Conferences on Artificial Intelligence Organization, 45–51. https://doi.org/10.24963/ijcai.2021/7 Main Track.

[7] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. 2013. Information and influence propagation in social networks. *Synthesis Lectures on Data Management* 5, 4 (2013), 1–177.

[8] Wei Chen, Wei Lu, and Ning Zhang. 2012. Time-critical influence maximization in social networks with time-delayed diffusion process. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[9] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer. https://doi.org/10.1007/978-3-319-21275-3

[10] Rod G Downey and Michael R Fellows. 1995. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on computing* 24, 4 (1995), 873–921.

[11] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of parameterized complexity*.

[12] Christoph Dürr and Nguyen Kim Thang. 2007. Nash equilibria in Voronoi games on graphs. In *European Symposium on Algorithms*. Springer, 17–28.

[13] Şirag Erkol, Dario Mazzilli, and Filippo Radicchi. 2020. Influence maximization on temporal networks. *Physical Review E* 102, 4 (2020), 042307.

[14] Şirag Erkol, Dario Mazzilli, and Filippo Radicchi. 2022. Effective submodularity of influence maximization on temporal networks. *Physical Review E* 106, 3 (2022), 034301.

[15] Naoka Fukuzono, Tesshu Hanaka, Hironori Kiya, Hirotaka Ono, and Ryogo Yamaguchi. 2020. Two-player competitive diffusion game: Graph classes and the existence of a Nash equilibrium. In *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 627–635.

[16] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman.

[17] Nathalie TH Gayraud, Evaggelia Pitoura, and Panayiotis Tsaparas. 2015. Diffusion maximization in evolving social networks. In *Proceedings of the 2015 acm on conference on online social networks*. 125–135.

[18] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters* 12, 3 (2001), 211–223.

[19] Jacob Goldenberg, Barak Libai, and Eitan Muller. 2001. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review* 9, 3 (2001), 1–18.

[20] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. 2013. Information diffusion in online social networks: A survey. *ACM Sigmod Record* 42, 2 (2013), 17–28.

[21] Petter Holme and Jari Saramäki. 2012. Temporal networks. *Physics reports* 519, 3 (2012), 97–125.

[22] Vamsi Kanuri, Yixing Chen, and Shrihari Sridhar. 2018. Scheduling Content on Social Media: Theory, Evidence, and Application. *Journal of Marketing* 86 (11 2018), 89–108. https://doi.org/10.1177/0022242918805411

[23] David Kempe, Jon Kleinberg, and Amit Kumar. 2002. Connectivity and inference problems for temporal networks. *J. Comput. System Sci.* 64, 4 (2002), 820–842.

[24] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 137–146.

[25] Jinha Kim, Wonyeol Lee, and Hwanjo Yu. 2014. CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing. *Knowledge-Based Systems* 62 (2014), 57–68.

[26] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1852–1872.

[27] Su-Chen Lin, Shou-De Lin, and Ming-Syan Chen. 2015. A learning-based framework to handle multi-round multi-party influence maximization on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 695–704.

[28] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. 2011. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697* (2011).

[29] Kevin Scaman, Rémi Lemonnier, and Nicolas Vayatis. 2015. Anytime influence bounds and the explosive behavior of continuous-time diffusion networks. *Advances in Neural Information Processing Systems* 28 (2015).

[30] Nemanja Spasojevic, Zhisheng Li, Adithya Rao, and Prantik Bhattacharyya. 2015. When-to-post on social networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2127–2136.

[31] Jimeng Sun and Jie Tang. 2011. A survey of models and algorithms for social influence analysis. In *Social network data analytics*. Springer, 177–214.

[32] V Tejaswi, PV Bindu, and P Santhi Thilagam. 2016. Diffusion models and approaches for influence maximization in social networks. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 1345–1351.

[33] Wenjun Wang and W Nick Street. 2018. Modeling and maximizing influence diffusion in social networks for viral marketing. *Applied network science* 3, 1 (2018), 1–26.

[34] Miao Xie, Qiusong Yang, Qing Wang, Gao Cong, and Gerard De Melo. 2015. Dynadiffuse: A dynamic diffusion model for continuous time constrained influence maximization. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[35] Ali Zarezade, Utkarsh Upadhyay, Hamid R. Rabiee, and Manuel Gomez-Rodriguez. 2017. RedQueen: An Online Algorithm for Smart Broadcasting in Social Networks *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 51–60. https://doi.org/10.1145/3018661.3018684

[36] Huiyuan Zhang, Thang N Dinh, and My T Thai. 2013. Maximizing the spread of positive influence in online social networks. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 317–326.

[37] Huiyuan Zhang, Subhankar Mishra, My T Thai, J Wu, and Y Wang. 2014. Recent advances in information diffusion and influence maximization in complex social networks. *Opportunistic Mobile Social Networks* 37, 1.1 (2014), 37.

[38] Tao Zhou, Jiuxin Cao, Bo Liu, Shuai Xu, Ziqing Zhu, and Junzhou Luo. 2015. Location-based influence maximization in social networks. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 1211–1220.

[39] Tian Zhu, Bai Wang, Bin Wu, and Chuanxi Zhu. 2014. Maximizing the spread of influence ranking in social networks. *Information Sciences* 278 (2014), 535–544.