Reducing Reachability in Temporal Graphs: Towards a More Realistic Model of Real-World Spreading Processes*

Kitty Meeks[1][0000−0001−5299−3073]

School of Computing Science, University of Glasgow, Glasgow G12 8RZ, UK
kitty.meeks@glasgow.ac.uk

Abstract. In many settings there is a need to reduce the spread of something undesirable, such as a virus, through a network. Typically, the network in which the spreading process takes place is not fixed but is subject to discrete changes over time; a natural formalism for such networks is that of *temporal graphs*. In this paper we survey three types of modifications that have been proposed in order to reduce reachability in temporal graphs, as well as the computational complexity of identifying optimal strategies for reducing reachability using each type of modification. We then go on to discuss several limitations of the current frameworks as models for intervention against real-world spreading processes, and suggest how these might be addressed in future research.

Keywords: Reachability · Temporal graphs · Spreading processes · Computational complexity · Parameterized algorithms

1 Introduction

Reachability is a crucial concept in understanding the spread of all kinds of things – good and bad – through networks. Sometimes we would like to restrict the spread of something undesirable, be it a virus or fake news, through a network; in other cases it is desirable to maximise the reachability subject to certain constraints, for example when preparing an advertising campaign or designing a transportation schedule. Here we focus on the problem of reducing reachability in the presence of an undesirable spreading process.

Many of the networks in which we are concerned with reachability – be these networks representing physical or virtual social contact, or transport networks – are inherently temporal: trains depart at specified times, and people do not interact continuously with all of their contacts. The relative timing of connections in the network clearly has an important impact on the reachability of individual vertices, as illustrated in Figure 1. These observations motivate the study of

* The work described here has been supported by EPSRC grants EP/T004878/1 and EP/V032305/1, and a Royal Society of Edinburgh Personal Research Fellowship (funded by the Scottish Government).

reachability in temporal graphs, whose edge-sets are subject to discrete changes over time.

Formally, following the foundational work of Kempe et al. [14], we define a *temporal graph* to be a pair (G, λ) where G is a (static) graph, often called the *underlying graph*, and $\lambda : E(G) \to 2^{\mathbb{N}} \setminus \{\emptyset\}$ maps edges of G to non-empty sets of times. A pair (e, t) where $e \in E(G)$ and $t \in \lambda(e)$ is called a *time-edge* of $\mathcal{G}$. We define the *lifetime* of $\mathcal{G} = (G, \lambda)$ to be the maximum time assigned to any edge by λ; throughout, we will consider only those temporal graphs whose lifetime is finite.

Crucial to the notion of reachability in temporal graphs is the notion of a temporal path. A *temporal path* in $\mathcal{G} = (G, \lambda)$ is a sequence of time-edges $(e_1, t_1), \ldots, (e_p, t_p)$ such that the edges $e_1, \ldots, e_p$ form a path in G and, for $1 \leq i \leq p - 1$, we have $t_i \leq t_{i+1}$. If all inequalities are strict, we say this is a *strict temporal path*.

Armed with these notions, we can define the concept of reachability sets in temporal graphs. The *(strict) temporal reachability set* of the vertex v in $\mathcal{G}$ is the set of all vertices u such that there exists a (strict) temporal path from v to u in $\mathcal{G}$; by convention, we assume that each vertex v also reaches itself. For a set of vertices S, the temporal reachability set of S is the union of the temporal reachability sets of vertices in S. The *temporal reachability* of a vertex (or set of vertices) is the size of its temporal reachability set, and the *maximum* (respectively *minimum*) *temporal reachability* of a temporal graph is the maximum (respectively minimum) temporal reachability of any vertex in the graph.

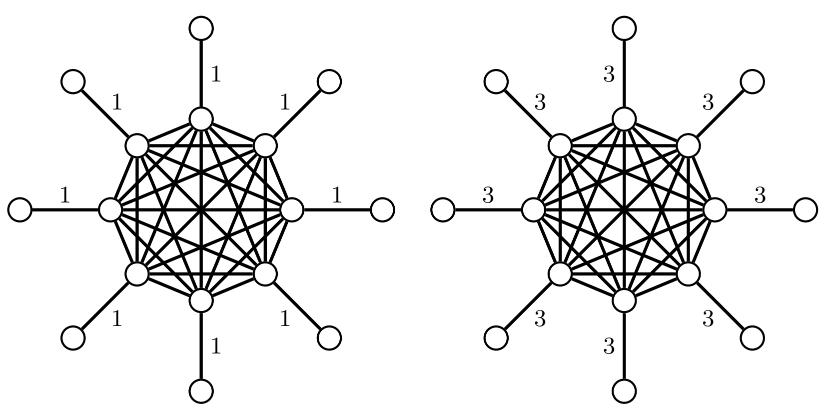

Fig. 1. Changing the relative order of edges can have an arbitrarily large impact on the maximum temporal reachability. Assuming that all edges within the clique are active only at time 2 in both examples (with times on the other edges as indicated), every vertex of the clique in the right-hand example reaches the entire graph, whereas no vertex in the left-hand example reaches any vertex outside the clique other than itself.

While there is clear motivation for studying reachability in the temporal setting, there are many associated challenges. Concepts that are well-understood or even trivial in the static setting become much more complex in the temporal world. For example, reachability is no longer symmetric: often, in the temporal setting, we will find that vertex u is reachable from vertex v even when v is not reachable from u. Classical graph theoretic results on connectivity, including Menger's Theorem, also fail to hold in the temporal setting [14] (although, in the case of Menger's Theorem, a more subtle temporal analogue has been developed [15]).

Motivated by the observation above that it is often desirable to decrease the extent to which something can spread through a network, a major recent focus for research in the temporal reachability setting has been on identifying optimal modifications to decrease the maximum or average temporal reachability of some set of nodes. Unsurprisingly, these kinds of optimisation problems typically turn out to be computationally intractable except in very specific cases.

In Section 2, we will introduce the three main types of modifications that have been studied in the literature to date, and summarise the key results regarding the computational complexity of using these modifications to minimise reachability. In Section 3 we then go on to discuss some limitations of the current models and how these might be extended to better encode real-world scenarios, before concluding in Section 4 with some thoughts on the main challenges for future work in this area.

2 Modifying Temporal Graphs to Reduce Reachability

Three main types of modification to reduce reachability have so far been proposed in the literature: edge deletions, reordering of edges, and delaying of edges. In recognition of the fact that modifications are likely costly, some kind of restriction is typically placed on the extent to which the graph can be modified (for example, by limiting the number of permitted modifications), but more sophisticated requirements that the modifications are not too disruptive to the network have not been investigated (see Section 3.4 for a more detailed discussion of this issue). In this section we discuss each of the types of modification in turn.

2.1 Deleting Edges

Enright, Meeks, Mertzios and Zamaraev [8] introduce the problem TR Edge Deletion: given a temporal graph (G, λ), together with natural numbers k and h, the goal is to determine whether it is possible to delete at most k time-edges[1] so that the maximum temporal reachability of the resulting temporal graph is at most h. This is a direct temporal analogue of earlier work by Enright and Meeks which considered edge deletion in static graphs as a means of limiting the size of

[1] The work in [8] focusses on the special case in which each $|\lambda(e)| = 1$ for every edge e, so deletion of time-edges is equivalent to deletion of static edges in the underlying graph.

an epidemic in cattle trade networks [10]; "deleting" an edge in this setting might correspond to enforcing additional checks or quarantine periods when animals are moved a certain route, thereby reducing the risk of disease transmission along this route to something close to zero. This static version of the problem is already NP-complete [10], so it is unsurprising that TR Edge Deletion remains intractable even under strong restrictions: the problem is NP-hard even when h, the maximum degree of the underlying input graph, and the lifetime of the input temporal graph are all bounded by constants. Moreover, assuming the Exponential Time Hypothesis, there is no hope achieving a running time in terms of the input size and the number k of permitted deletions that improves significantly on a brute-force approach; from the parameterised perspective, this also demonstrates that the problem is W[1]-hard with respect to the parameter k.

On the positive side, the authors obtain two polynomial-time algorithms which approximate the minimum number of deletions needed: one computes an h-approximation on arbitrary graphs, while the other computes a c-approximation whenever the underlying input graph has cutwidth at most c. While at first sight these approximation ratios seem to leave a lot of room for improvement, it is also shown that the problem is unlikely to admit a polynomial-time constant-factor approximation in general, even when restricted to temporal graphs with lifetime two. From the parameterised perspective, the problem is also shown to admit an exact FPT algorithm when parameterised simultaneously by h, the maximum degree of the underlying input graph, and the treewidth of the underlying input graph. Many of these results are also extended to a setting in which restrictions are placed on the permitted waiting time at each vertex along a temporal path.

2.2 Reordering Edges

A different method for reducing reachability in temporal graphs was proposed by Enright, Meeks and Skerman [11]: given a static underlying graph as input, the goal is to assign times to the edges so as to minimise the maximum temporal reachability of the resulting temporal graph. Formally, they introduce the problem Min-Max Reachability Temporal Ordering, which takes as input a static graph $G = (V, E)$, a list $\mathcal{E} = \{E_1, \ldots, E_\ell\}$ of subsets of E, and a positive integer h, and asks whether there is a bijective function $t : \mathcal{E} \to [\ell]$ such that the maximum temporal reachability of the temporal graph in which every edge in E_i is active at time $t(E_i)$ is at most k. The requirement that the time-labelling function t be bijective is to rule out trivial solutions in the case of strict reachability (since, in this case, reachability can always be minimised by making all edges active at the same time); for non-strict reachability, there is always an optimal solution in which no two edge-sets are assigned the same time since, if E_1 and E_2 are initially assigned the same time, altering the assignment so that they are given consecutive times (in arbitrary order, and without changing the relative order of any other sets) will not increase the reachability of any vertex. This model was once again inspired by the spread of disease in livestock: many livestock trades are mediated through markets, with named events (e.g. "Spring

Bull Sale") taking place on specific dates; thus certain sets of trades will take place on the same day, but the relative order of this set of trades and another set (of another class of animal, but potentially involving many of the same farms) could be changed by an auction company.

The general version of this problem turns out to be extremely computationally challenging: Min-Max Reachability Temporal Ordering is NP-complete even on trees and DAGs, and from the parameterised perspective is W[1]-hard parameterised by the vertex-cover number of the input graph, even when the input graph is required to be a tree. The only positive result in this setting is a constant-factor approximation to the optimisation problem when the input satisfies several strong structural restrictions.

In the quest for more positive results, the authors consider a special case – further removed from the motivating application – in which each edge set E_i is a singleton (so that all edges can be reordered independently). In this case the problem is still NP-complete on general graphs, but does admit a linear-time constant-factor approximation algorithm when restricted to graphs of bounded maximum degree. This restriction of the problem can also be solved exactly in polynomial time on DAGs, and on trees it admits an FPT algorithm parameterised by either the maximum permitted reachability h or the maximum degree.

2.3 Delaying Edges

A third family of modifications that could be used to reduce temporal reachability was introduced by Deligkas and Potapov [7]: they consider the operations of either *merging* a sequence of consecutive timesteps (so that all edges previously active at any timestep in the sequence are now only active at the last timestep in the sequence), or *delaying* individual edges by some number of timesteps. They propose that, in some settings, these merging and delaying operations are less disruptive to the infrastructure than the deletion or more general reordering approaches discussed above. Again, the goal[2] is to reduce the reachability of the resulting temporal graph: in addition to considering the maximum temporal reachability, they also consider the problems of minimising the average temporal reachability over all vertices and of minimising the temporal reachability of some specific set of vertices given as part of the input. Determining whether it is possible to achieve a specified bound for any of the three notions of temporal reachability using merging operations turns out to be NP-complete, even on trees of maximum degree three. In the setting of delaying operations, the authors adapt a reduction from [8] to show that, when the number of permitted delaying operations is bounded by k, all three problems are NP-complete and W[1]-hard with respect to the parameter k. On the positive side, however, they give a polynomial-time algorithm to optimise all three notions of temporal reachability when an arbitrary number of delaying operations can be applied.

[2] Beyond the scope of this paper, the authors in [7], and a subsequent related work [6], also consider goals related to increasing reachability using the same operations.

Molter, Renken and Szchoche [16] recently investigated the relationship between the complexity of minimising temporal reachability by respectively deleting and delaying edges. In addition to showing that the maximum temporal reachability of any vertex in some specified set of source vertices can, in both cases, be minimised in polynomial time when the underlying graph is a tree, they provide a general reduction from the delaying version to the deletion version of the problem. However, when parameterised by the permitted number of reachable vertices, this relationship is reversed: the delaying version is in FPT while the deletion version remains W[1]-hard.

3 Limitations of Current Models

While all of the work described above draws on real-world applications for motivation, there are a number of common features in all of these models which limit their practical applicability. In this section we will discuss some of these, and the ways in which future research might address the current shortcomings.

3.1 Arbitrary Waiting Times

With the exception of some results on deletion [8], the work described above allows the most flexible notion of temporal paths, whereby there is no restriction on the waiting time at a single vertex. This is not necessarily a realistic model of many spreading processes: for example, in the setting of disease spread, there will typically only be a limited time-window during which an individual is infectious and can potentially pass on infection to their contacts; similarly, when considering the spread of information it seems more likely that an individual will share gossip they have just learnt than that received a year ago. Enright, Meeks, Mertzios and Zamaraev [8] consider the edge-deletion problem in the setting of so-called (α, β)-reachability as a more realistic model for disease spread, in which the waiting time at a vertex must be at least α and at most β timesteps. However, their results implicitly assume a disease for which recovery does not confer immunity (or a network in which vertices correspond to groups of individuals, e.g. animals on a farm, and so multiple infections of different individuals within the group are possible), as (α, β)-reachability is defined in terms of temporal *walks* which satisfy this restriction, meaning that the same vertex may be visited multiple times: this is primarily for pragmatic computational reasons, since the existence or otherwise of such a walk between two vertices can be determined in polynomial time [1], whereas determining the existence or otherwise of a so-called *restless* temporal path, in which the waiting time at each vertex can be at most Δ, is known to be NP-complete [4].

It would be particularly interesting to investigate the impact of restrictions on waiting times in the two models in which times are changed: for example, a delay could either create or destroy restless temporal paths without changing the set of (traditional) temporal paths in the graph. However, this restriction has not yet been considered in either the reordering or delaying model; this seems a fruitful line for future research.

3.2 Deterministic Spreading Only

Perhaps the most obvious limitation of all the models discussed above is that they consider only deterministic worst-case spread, whereas in reality transmission along a single time-edge will be associated with some probability. There are certain applications for worst-case analysis of this kind, for example if all potential contacts of a diseased individual need to be identified and required to isolate, but in other settings we are likely to be more interested in the expected reachability of a vertex (or other measures associated with the random variable denoting the size of the reachable set).

This natural generalisation has been proposed as an extension to current work [11], but probabilistic spreading has not so far been considered within any of the models introduced above. One reason for this is that probabilistic spreading vastly changes the nature of the computation: computing the probability of transmission between a pair of vertices is equivalent to counting the number of weighted temporal paths between the two (where weights are multiplied along the path, and correspond to the transmission probabilities associated with the corresponding time-edges), and even without weights this problem is known to be intractable in many settings [9]. Thus even the problem of verifying whether a given intervention (based on deleting or changing the times on edges) is sufficient to achieve a specified goal (in terms of the expected reachability) will be intractable unless the input is highly restricted. However, it would nevertheless be worthwhile to carry out further research in this direction, even if the only realistic goal is that of obtaining approximation algorithms in restricted settings.

3.3 Only One Type of Modification Allowed

Several different kinds of modification are encapsulated in the various models discussed above, each of which may be realistic in particular settings, but each model considered allows only one type of modification to be applied to a single instance. However, it is credible that, in many scenarios, it might for example be feasible to change the timing of some edges while removing others completely. The complexity of carrying out combinations of these modifications to achieve a specific reachability goal is entirely unexplored.

To make hybrid models of this kind more realistic and flexible, it would be natural to introduce a cost function that allows different kinds of modifications to be more or less costly (for example, delaying an edge by a short time should intuitively be less costly than removing it altogether). A cost function of this kind could also be used to enforce restrictions on which kinds of modifications are permitted at each edge: for example, there might be some edges that we cannot delete but whose time we are allowed to change.

More generally, cost functions would also be a valuable extension within any of the models that allows only a single type of modification. For example, some edges may be more costly to delete than others. However, the notion of cost is even more powerful in the reordering or delaying setting: here the cost could depend not only on the edge under consideration, but also the duration and/or direction of the time modification (assuming that some initial default configuration is given in the reordering model). This would reflect the intuition that changing the time on an edge by a small amount is likely to be less disruptive than a larger change, and also the fact that in some settings either advancing or delaying will be easier; potentially in some settings, if the change to the time is large enough, it is no less disruptive to delete the edge entirely. Moreover, the relationship between time difference and cost need not be monotonic: it might for example be easier (or indeed only possible) to reschedule a connection to take place on the same day of the week as originally scheduled.

3.4 Independence of Modifications

With the exception of a few results in the reordering setting, the models described here assume that edges can be modified independently from one another; however, this is not necessarily realistic. In the deletion setting, for example, there might be some minimum connectivity requirements that must be maintained – if we were dealing with a transport network, say, we might want to ensure that every vertex is still able to reach some vertex corresponding to a location providing basic essential services. The potential dependencies between different modifications are perhaps even clearer in the delaying setting: again considering a transport network, if particular edges use the same vehicle or driver then delaying one edge may necessarily force other related edges to be delayed; on the other hand, we might not be allowed to increase the gap between two such related edges by too much without violating legal requirements on working hours for the driver.

These observations motivate the development of refined models in which the dependencies between different edges can be captured. Once again, such generalisations will only serve to increase the computational complexity of problems that are already intractable in many cases, but it would be instructive to determine whether any of the restricted settings that admit efficient algorithms in the independent versions of the models also give tractability when some level of dependency is considered. Moreover, investigations in this direction would motivate the development of parameters to capture the structure of the dependencies. This could, for example, be done by considering the structure of the graph whose vertex-set is the set of (time-)edges in the input graph, and in which there is an edge between two (time-)edges if and only if the two cannot be modified independently. The maximum size of a connected component in this auxiliary graph was implicitly considered as a parameter in the reordering model [11], but other parameters such as the maximum degree or treewidth of the auxiliary graph would also likely be of interest.

3.5 Perfect Knowledge

All of the models discussed above assume that (where appropriate) we have complete knowledge of the initial schedule and can carry out precise modifications (so that we specify exactly the new time assigned to an edge after delaying or reordering). However, in real-world scenarios, information and our control of changes is likely to be imperfect, so we may have to operate under uncertainty. For example, we might not know (or be able to specify) the precise time at which a journey will be made, but instead a window in which it will occur; if this is the case for two incident edges in the network, their relative order might not be known. Moreover, a pre-determined schedule might be subject to unplanned disruption, for example in a transport network where a certain number of unforeseen delays can be expected. It would be natural to consider both random and adversarial models of changes to the schedule, with some bound on the number and/or scale of changes.

These observations motivate the development of techniques to deal with uncertainty and identify modification strategies that are in some sense robust to small changes in the schedule, or provide a good approximation to the best strategy regardless of when connections occur during their permitted windows. Very recently, Füchsle, Molter, Niedermeier and Renken [13, 12] made a first step in this direction, considering the simpler goal of determining whether there is a temporal path between two vertices that is robust against some small number of delays (as, for example, when planning a multi-leg train journey). They already encounter intractability in several settings, indicating that optimising reachability in such settings will certainly be challenging.

4 Discussion

In spite of much recent work in this area, we have only scratched the surface in understanding the complexity of determining optimal sets of modifications to minimise reachability in realistic temporal settings. There is a strong motivation for generalising the models that have been studied so far to incorporate more realistic features; however, given the extreme computational tractability that has already been encountered in the more basic variants, this will require new approaches to dealing with intractability. We have seen that restricting only the structure of the underlying graph is rarely enough to give rise to efficient algorithms, as many of the problems remain intractable even when the underlying graph is (for example) a tree. A promising research direction that has emerged recently is the development of parameters that describe aspects of the *temporal* structure of temporal graphs in combination with structural properties of the underlying graph. Recent parameters of this kind include the timed feedback-vertex number [5] and the (vertex-)interval-membership-width [2, 3] of temporal graphs; vertex-interval-membership-width has already been exploited in the reachability setting to give an FPT algorithm to minimise the number of vertices reachable from a fixed source under deletion operations [3], but the values of both parameters on real-world networks of interest have yet to be investigated.

References

1. Bentert, M., Himmel, A., Nichterlein, A., Niedermeier, R.: Efficient computation of optimal temporal walks under waiting-time constraints. Appl. Netw. Sci. **5**(1), 73 (2020). https://doi.org/10.1007/s41109-020-00311-0
2. Bumpus, B.M., Meeks, K.: Edge exploration of temporal graphs. In: Flocchini, P., Moura, L. (eds.) Combinatorial Algorithms - 32nd International Workshop, IWOCA 2021, Ottawa, ON, Canada, July 5-7, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12757, pp. 107–121. Springer (2021). https://doi.org/10.1007/978-3-030-79987-8_8
3. Bumpus, B.M., Meeks, K.: Edge exploration of temporal graphs. CoRR **abs/2103.05387** (2021)
4. Casteigts, A., Himmel, A., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. Algorithmica **83**(9), 2754–2802 (2021). https://doi.org/10.1007/s00453-021-00831-w
5. Casteigts, A., Himmel, A., Molter, H., Zschoche, P.: Finding temporal paths under waiting time constraints. Algorithmica **83**(9), 2754–2802 (2021). https://doi.org/10.1007/s00453-021-00831-w
6. Deligkas, A., Eiben, E., Skretas, G.: Minimizing reachability times on temporal graphs via shifting labels. CoRR **abs/2112.08797** (2021)
7. Deligkas, A., Potapov, I.: Optimizing reachability sets in temporal graphs by delaying. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 9810–9817. AAAI Press (2020)
8. Enright, J., Meeks, K., Mertzios, G.B., Zamaraev, V.: Deleting edges to restrict the size of an epidemic in temporal networks. Journal of Computer and System Sciences **119**, 60–77 (2021). https://doi.org/10.1016/j.jcss.2021.01.007
9. Enright, J., Meeks, K., Molter, H.: Counting temporal paths. CoRR **abs/2202.12055** (2022)
10. Enright, J.A., Meeks, K.: Deleting edges to restrict the size of an epidemic: A new application for treewidth. Algorithmica **80**(6), 1857–1889 (2018). https://doi.org/10.1007/s00453-017-0311-7
11. Enright, J.A., Meeks, K., Skerman, F.: Assigning times to minimise reachability in temporal graphs. J. Comput. Syst. Sci. **115**, 169–186 (2021). https://doi.org/10.1016/j.jcss.2020.08.001
12. Füchsle, E., Molter, H., Niedermeier, R., Renken, M.: Delay-robust routes in temporal graphs **219**, 30:1–30:15 (2022). https://doi.org/10.4230/LIPIcs.STACS.2022.30
13. Füchsle, E., Molter, H., Niedermeier, R., Renken, M.: Temporal connectivity: Coping with foreseen and unforeseen delays. CoRR **abs/2201.05011** (2022)
14. Kempe, D., Kleinberg, J., Kumar, A.: Connectivity and inference problems for temporal networks. Journal of Computer and System Sciences **64**(4), 820–842 (2002). https://doi.org/doi.org/10.1006/jcss.2002.1829
15. Mertzios, G.B., Michail, O., Spirakis, P.G.: Temporal network optimization subject to connectivity constraints. Algorithmica **81**(4), 1416–1449 (2019). https://doi.org/10.1007/s00453-018-0478-6
16. Molter, H., Renken, M., Zschoche, P.: Temporal reachability minimization: Delaying vs. deleting. In: Bonchi, F., Puglisi, S.J. (eds.) 46th International Symposium

on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia. LIPIcs, vol. 202, pp. 76:1–76:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.MFCS.2021.76