Summary of ”Making Life More Confusing for Firefighters”

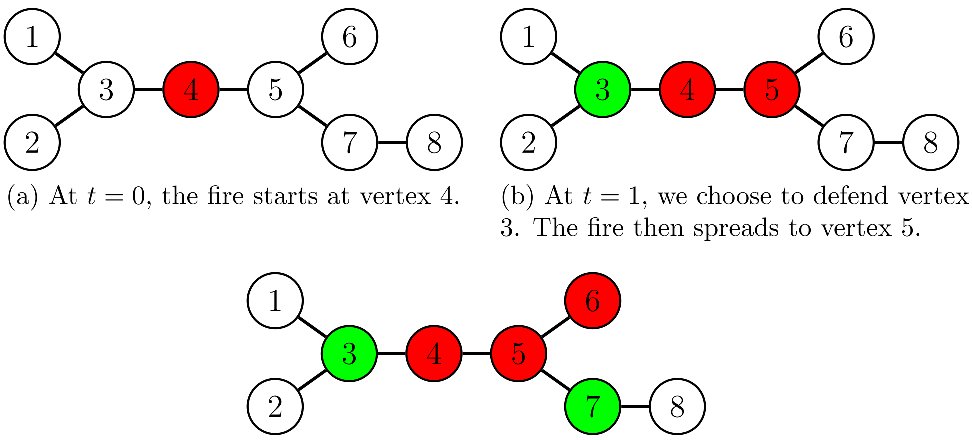
Oskar Ådahl Juni Drakengren Tim Erixon Simon Lindblom
John Enbäck

1 Introduction

In this project, we give a summary of the paper *Making Life More Confusing for Firefighters* written by Samuel D. Hand, Jessica Enright, and Kitty Meeks, all from the University of Glasgow. [3] In the paper, they investigate whether the Firefighter problem (the F problem) on certain classes of temporal graphs is NP-complete or not.

To even understand the problem they address, we first introduce some of the core terminology. In Section 2 follows a description of the main methods and ideas used in the paper's proofs. Finally, in Section 3, we present the results achieved by the authors.

The F problem is known to be NP-complete on arbitrary graphs, NP meaning that the answer is either yes or no, and that it is possible to verify the solution within polynomial time, while the completeness part means that every other NP, in polynomial time, can be reduced to the problem.[1] Since there are no known generalised algorithms for NP-complete problems, the authors are interested in for what classes of graphs or what conditions on lambda for temporal graphs it is possible to find an algorithm solving the problem within a reasonable time; polynomial time, that is tractable problems, and which classes of graphs have NP-complete problems. Therefore, in chapters 3 and 4, they enforce restrictions on the graphs, and put constraints on the lifespan of temporal graphs respectively, studying whether the problems now are NP-complete or tractable.



(a) At $t = 0$, the fire starts at vertex 4.

(b) At $t = 1$, we choose to defend vertex 3. The fire then spreads to vertex 5.

(c) At $t = 2$, we choose to defend vertex 7. The fire then spreads to vertex 6. Now, the fire cannot spread anymore, so the process terminates.

Figure 1: An example of a graph G going through the fire with a strategy of deploying two firefighters, saving a total of 5 vertices.

A sequence of vertices $v_1, v_2, ..., v_n$ such that v_i is a valid defence at $t = i$, i.e., v_i is neither burning nor defended at $t = i$, is called a *strategy.* If we are given a rooted graph (G, r) and an integer k, F asks whether there exists a strategy that saves at least k vertices from burning when the fire starts at r. In Figure 1 above, our strategy is $3, 7$, and we save 5 vertices. Hence, we have solved F for the given rooted graph and $k = 5$.

A *temporal graph* is a pair (G, λ), where G is a graph and λ is a function on the set of edges of G that assigns to each edge a set of timesteps at which it is active. That an edge e is active at timestep i means that the fire can spread along e at that time. Similarly, when e is not active, the fire cannot spread along it. For example, a normal graph is a temporal graph where $\lambda(e) = \mathbb{N}$ for all edges e in G. An important concept related to temporal graphs is the *lifetime*, which is the last timestep at which any edge is active.

The *Temporal Firefighter Problem* (TF) is very similar to F, with the sole difference that we let the rooted graph (G, r) be a temporal rooted graph $((G, r), \lambda)$ instead.

The F problem is known to be NP-complete on arbitrary graphs, NP meaning that the answer is either yes or no, and that it is possible to verify the solution within polynomial time, while the completeness part means that every other NP, in polynomial time, can be reduced to the problem.[1] Since there are no known generalised algorithms for NP-complete problems, the authors are interested in for what classes of graphs or what conditions on lambda for temporal graphs it is possible to find an algorithm solving the problem within a reasonable time; polynomial time, that is tractable problems, and which classes of graphs have NP-complete problems. Therefore, in chapters 3 and 4, they enforce restrictions on the graphs, and put constraints on the lifespan of temporal graphs respectively, studying whether the problems now are NP-complete or tractable.

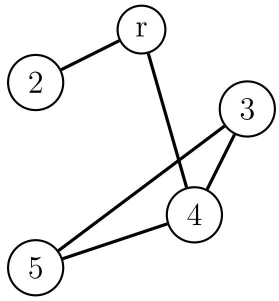
2 Method

2.1 Restricting the Underlying Graph

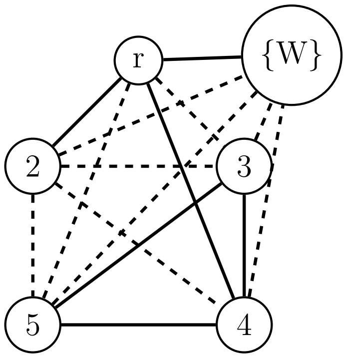
This chapter's main focus is on the difference between the Firefighter Problem (F) and the Temporal Firefighter Problem (TF), and the complexity that arises with TF. That leads to several classes of graphs that are solvable in Polynomial (P) time for the Firefighter Problem, but NP-complete if it is instead "put into" the Temporal Firefighter Problem, if the underlying graph is a clique, where "underlying graph is a clique" means that the graph the problem plays out on is a complete graph, however the edges does only need to be temporally active at at least one timestep (see Figure 2).

However, in the paper, they prove the stronger statement: *For any constant $c \in \mathbb{N}$, TF is NP-complete when restricted to temporal graphs whose underlying graph is a clique and whose lifetime is at most $n^{\frac{1}{c}}$, where n is the number of vertices in the graph.*

They prove it by reducing a static graph (G, r) into $((G', \lambda), r)$ with a set W containing the number of vertices $|V(G)|^c - |V(G)|$ (where c is a positive integer) added and connected by a "static"/temporally active edge throughout the spreading of the fire, to r. Using the same strategy S used for the F problem with $((G, r), k)$, leads to all those vertices in W burning up at $t = 1$, and the rest of the vertices (2 through 5 in Figure 2) being able to save k vertices by the same strategy S used in (G, r) which also saved k vertices. If you would instead save a vertex in W (at $t = 1$ since no other time is possible), then at least one (or more vertices) would burn from the original G, saving at the very most the same number of vertices, but probably less, leading to strategy S saving more or at the very least the same number of vertices.



(a) The static graph $((G, r), k)$, with root r and have at least one strategy that saves k vertices.



(b) The graph $(((G', \lambda), r), k)$, with the vertex marked $\{W\}$ being a set W containing all vertices $|V(G)|^c - |V(G)|$ added (meaning that W contains that amount of vertices), and a strategy saving k vertices iff there was a strategy in $((G, r), k)$ that did so/was a yes-instance.

$\lambda(-) = \{1, 2, ..., |V(G)| - 1\}$
$\lambda(- - -) = \{|V(G)| - 1\}$

Figure 2: An example of the transformation or reduction from static (G, r) to temporal $((G', \lambda), r)$, with time-labelling function λ such that $((G', \lambda), r)$ is a clique with edges between all vertices while still acting as the original graph (G, r).

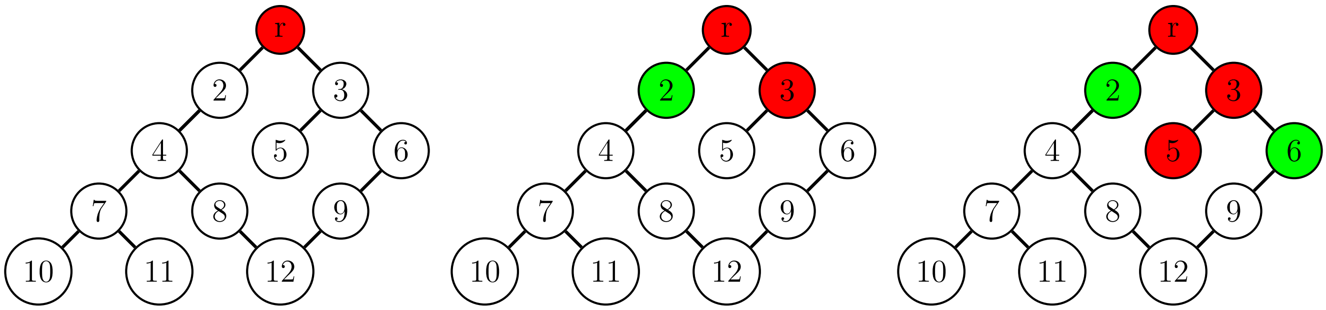
This means that if there is a strategy S that follows from F in graph $((G, r), k)$, then that same strategy will also save k vertices in $(((G', \lambda), r), k)$ if we use the reduction seen in Figure 2.

That theorem leads to the corollary that several types/classes of graphs that are solvable in P time in F, are instead NP-complete in TF: split graphs, unit interval graphs, cographs, P_k-free graphs for $k > 2$ and AT-free graphs (the last one is of interest since it is the only one in the list that has not been proven to be solvable in P time, but is shown to be NP-complete in TF).

However, there is a class of graphs that is solvable in P time for the F problem (see [2]), and also solvable in P time for the TF problem: the class of graphs of maximum degree 3, with a root of degree at most two ($G3R2$), an example of which can be seen in Figure 1.

To prove the statement that $G3R2$ is solvable in P time on a rooted temporal graph $((G, \lambda), r)$ the paper uses the same structure as in [2], while modifying it for the temporal structure. For $G3R2$ to be solvable in P time, we need to show that there is an optimal strategy, S, that always defends next to the fire (like in Figure 1) while at the same time creating the shortest possible path of fire.

The paper proves it by creating a function $f(u)$ that checks the temporal distances (meaning how many timesteps it takes between the two vertices, including the timesteps "waiting" for the edge to be temporally active) between the root r and vertex u (where u would be the last vertex to burn), and then showing that S minimises the function $f(u)$ by showing that when the fire reaches u, there are only two possible cases, as of the degree restriction of $G3R2$ leading to only two other vertices being connected to u. Then, for the fire to stop at u, either the two vertices are already defended (case 1), which would mean that we had time to defend those two before the fire came to u (which means that $f(u)$ is not optimal and that we could have minimised it by defending next to the fire). For the second case, that one of the vertices had already been defended, and the other one would have been defended at the next timestep, means that S is the optimal strategy and the $G3R2$ is solvable in P time for the TF problem.



(a) A $G3R2$, with root r and a cycle (which is allowed), at $t = 0$.

(b) Defending vertex 2, since $2 \sim 10$ is the longest possible path, and $r \sim 3$is the shortest possible leaf, leading to vertex 3 burning at $t = 1$

(c) Defending vertex 6 since it $6 \sim 12$ is longer than $3 \sim 5$ (which is the shortest possible leaf), stopping the spread of the fire at $t = 2$.

Figure 3: The F problem on a $G3R2$-graph, using the optimal strategy.

We can see an example of this strategy S in action in figure 3, as we minimise the longest possible "leaf" (and/or cycle), leading to us finding $r \sim 3 = P$ (P stands for the optimal Path) and defending the vertices outside of P but that is directly next to the fire, "leading" the fire to vertex 5.

Note that while figure 3 is a Firefighter problem (since it is easier to draw and is basically the same process), the TF-problem can only shorten the optimal path, and the strategy is equally easy to calculate as $f(u)$ now just checks the **temporal** length/distance between r and u and finding the shortest one in a temporal space, instead of the shortest distance in a static graph.

2.2 Restricting the Temporal Structure

The next chapter studies a certain limit on the rooted temporal graph, and how it is related to the time (computationally). To do this, they first (for some temporal graph (G, λ)) define a kind of span for the lifetime of a vertex v, its lifespan starts at the first timestep where it is incident to an active edge and ends at the last timestep where it is incident to an active edge. From this is defined a sequence of sets, F_t, where F_t is a set of vertices and v is in F_t iff t is part of v's lifespan. They then define ω as the size of the largest of these sets. This is almost an upper bound on the number of vertices relevant at any timestep, although it does not take into account the possibility to do moves that will be beneficial in the future but not in the present.

Here we are introduced to the temporal firefighter reserve problem (TFR), which is the same as TF but allows us at each turn to, instead of deploying a firefighter, save and deploy it at a later timestep instead. Fixing a rooted TG $((G, \lambda), r)$, the strategies of TF are a subset of the strategies of TFR. In addition, making a move earlier never hurts. This leads to a more concrete lemma, which says that on a rooted TG, being able to save at least k vertices in the TF version is equivalent to being able to save at least k vertices in the TFR version. They henceforth use TFR to also be able to draw a conclusion about TF. A reason why we use TFR is in order to only have to focus on at most ω vertices (and what has happened before) at each timestep.

Most of the rest of the chapter is about designing a fixed parameter tractable (FPT)-algorithm with the goal of determining if it is possible to save k vertices, taking a rooted temporal graph as input. A FPT-algorithm is an algorithm that works in polynomial time if you fix some constant, which here will be ω.[4] It works by recursively computing a sequence of time-dependant sets L_i, where each element in L_i contains a tuple of information, namely: the set of defended vertices in F_i (D), the set of burned vertices in F_i (B), the number of firefighters deployable at time $i + 1$ (g) and the total number of vertices burning at time i (c). The fact that this algorithm works is encapsulated by a theorem. Its idea is to first set L_0, whose only element is $(\emptyset, \{r\}, 1, 1)$ since these are the initial values. They then construct the elements in L_i by, for each element in L_{i-1}, and for some choice of new vertices to defend, checking if a tuple satisfies 4 conditions. These conditions are limits on the new sets of defended and burned vertices, the new budget, and the new number of vertices burning that ensure that we get the situation we would expect, with an element in L_i as our previous step and having chosen vertices to defend. This induction means that a snapshot of the graph is in some L_i iff it is part of some strategy. This gives the endpoint all possible strategies that could be optimal, where we know how many vertices are saved by each strategy and can compare these to conclude if k vertices can be saved.

This leads us to the final theorem, which states that it is possible to solve TF in time $O(8^{\omega}\omega\Lambda^3)$. The main point of this is to say that for a fixed omega, the problem can be solved in polynomial time. Since we only have to focus on at most ω nodes at any timestep, this is what limits the exponential growth of the possible strategies.

3 Result

The paper has shown that Temporal Firefighter is NP-complete if its graph belongs to a class of graphs for which Firefighter is also NP-complete. It has also been shown that Temporal Firefighter is NP-complete for some graphs that are known to be solvable in polynomial time for Firefighter. Graphs for which the maximum degree is three, and for which the root is of degree two, are solvable in polynomial time for both Temporal Firefighter and Firefighter. The authors then produce a FPT-algorithm that restricts the temporal structure, such that the Temporal Firefighter becomes solvable in polynomial time. The authors also highlight areas that could extend their current works, such as figuring out the complexity of a Temporal Firefighter where the number of active edges is maximum bounded for each timestep.

References

[1] Stephen Arthur Cook. *NP-complete problem*. `https://www.britannica.com/science/NP-complete-problem`. Nov. 2023.

[2] Stephen Finbow et al. "The firefighter problem for graphs of maximum degree three". In: *Discrete Mathematics* 307.16 (2007). `https://www.sciencedirect.com/science/article/pii/S0012365X0600776X`, pp. 2094–2105.

[3] Samuel Hand, Jessica Enright, and Kitty Meeks. "Making life more confusing for firefighters". In: *arXiv preprint arXiv:2202.12599* (2022). `https://arxiv.org/pdf/2202.12599.pdf`.

[4] David Karger. "Fixed Parameter Tractability and Treewidth". `https://courses.csail.mit.edu/6.854/06/scribe/s21-fixedParamNP.pdf`. Scrib: Pramook Khungurn, Class: 6.854 Advanced Algorithms, MIT.