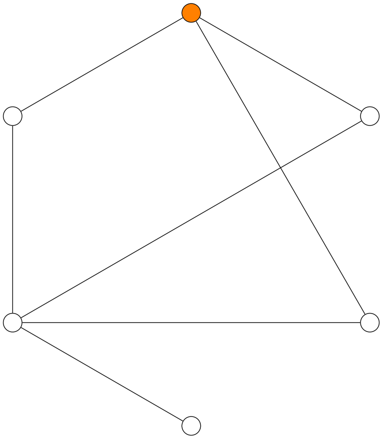
The Temporal Firefighter Problem

Samuel Hand, Jessica Enright, Kitty Meeks

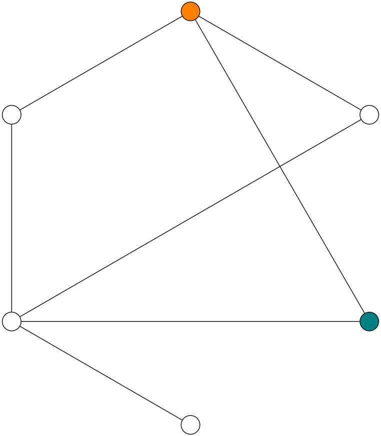
July 2022

School of Computing Science, University of Glasgow

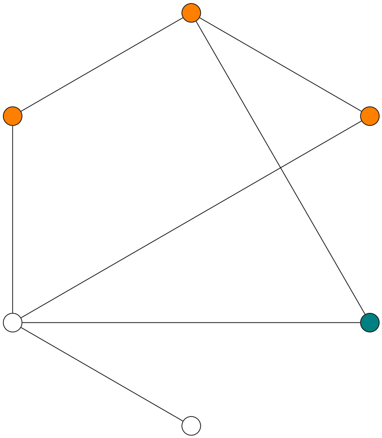
Firefighter (Hartnell 1995)



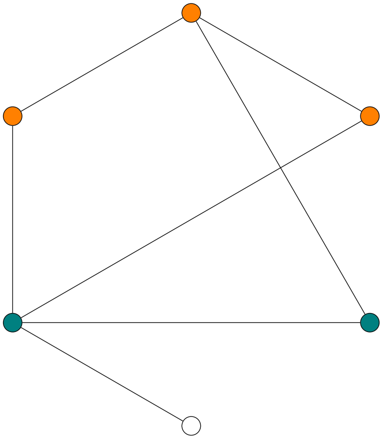
Firefighter (Hartnell 1995)



Firefighter (Hartnell 1995)



Firefighter (Hartnell 1995)



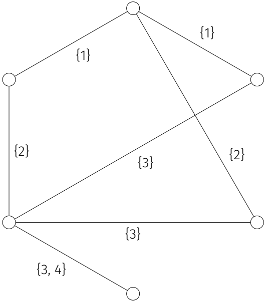
Firefighter (Hartnell 1995)

Input: A rooted graph (G, r) and an integer k.
Output: Can we save at least k vertices if the fire starts burning at r?

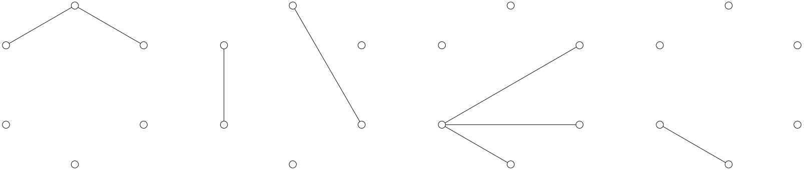
NP-Complete on arbitrary graphs, but in P for:

- Interval graphs, permutation graphs, P_k-free graphs for $k > 5$, split graphs, cographs (Fomin, Heggernes, and Leeuwen 2016).
- Graphs of maximum degree three with $\deg(r) = 2$. (Finbow et al. 2007).

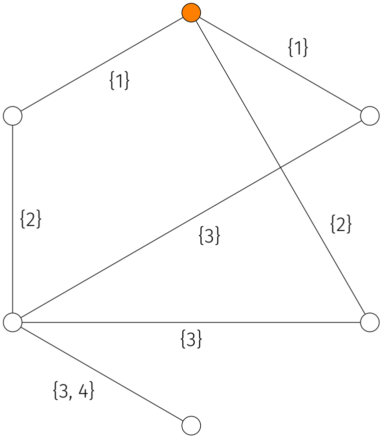
(G, λ):



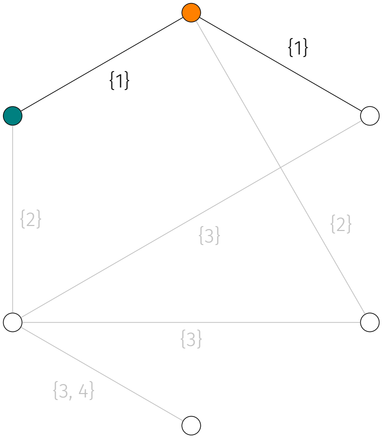
Lifetime = $\Lambda = 4$



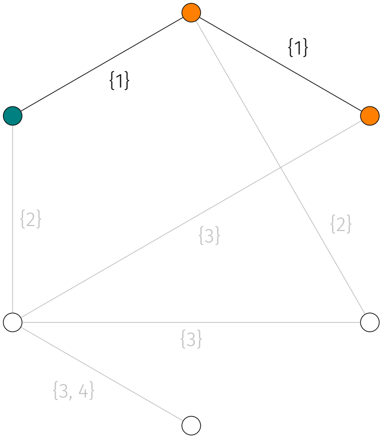
$t = 0$



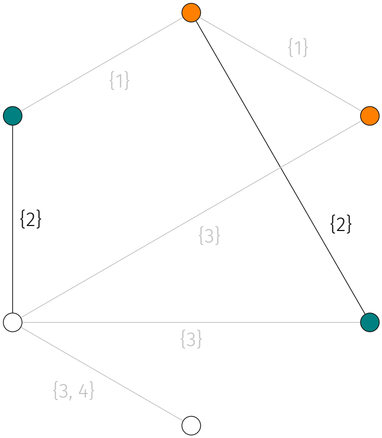
$t = 1$



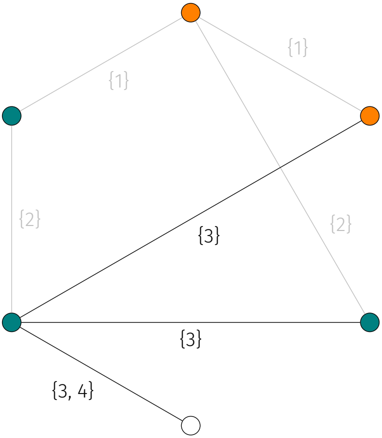
$t = 1$



$t = 2$



$t = 3$



Temporal Firefighter

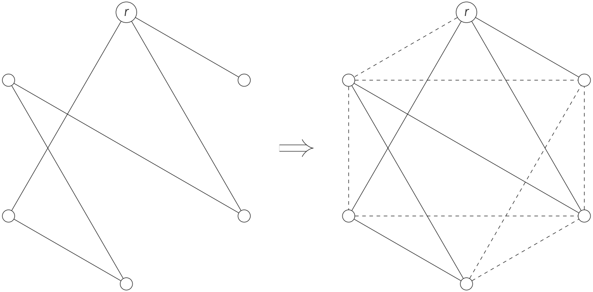
Input: A rooted temporal graph $((G, \lambda), r)$ and an integer k.
Output: Can we save at least k vertices if the fire starts burning at r?

NP-Complete whenever the underlying graph belongs to a class for which Firefighter is NP-Complete.

Remains in P for graphs of maximum degree 3 providing $\deg(r) = 2$, the good news seems to end there...

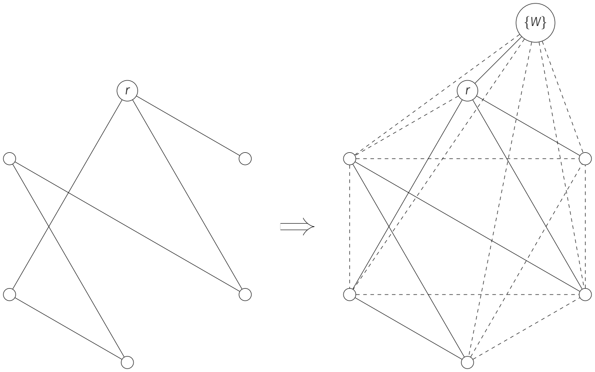
Our Hardness Results

NP-Complete on interval graphs, permutation graphs, P_k-free graphs for $k > 5$, split graphs, and cographs.



$\lambda(\text{—}) = \{1..|V(G)| - 1\}$

$\lambda(\text{---}) = \{|V(G)| - 1\}$



$\lambda(\text{—}) = \{1..|V(G)| - 1\}$

$\lambda(\text{- - -}) = \{|V(G)| - 1\}$

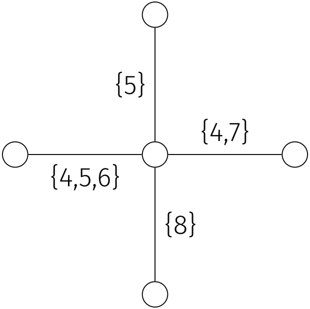
Using the underlying structure of the graph doesn't seem to fruitful, what else can we try?

Make use of the times! We look for a parameter of the temporal structure that will give us fixed parameter tractability.

Reminder: we say a problem is fixed parameter tractable if we can solve it in time $f(k) \cdot \text{poly}(n)$.

Temporal Firefighter is fixed parameter tractable when parameterised by vertex-interval-membership-width.

Measures the maximum number of vertices that are *relevant* at any given timestep (Bumpus and Meeks 2021).



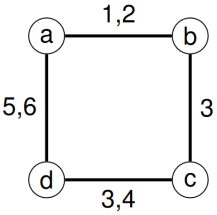
In particular the vertex interval membership width is the maximum size of an entry in the vertex interval membership sequence.

This is a sequence of sets of vertices $(F_t)_{t \in [\Lambda]}$ - one set for each timestep.

For each time t we include a vertex v in F_t if its active interval contains t:

$$F_t := \{u \in V(G) : \exists i \leq t \leq j, \exists v, w \in V(G). i \in \lambda(vu) \wedge j \in \lambda(wu)\}$$

Vertex Interval Membership Width



U_1	U_2	U_3	U_4	U_5	U_6
a	a	a	a	a	a
b	b	b			
		c	c		
		d	d	d	d

Figure 1: A graph with vimw $\omega = 4$, the vertex interval membership sequence is displayed to the right.

Then the vertex interval membership sequence of a graph G is the integer $\omega := \max_{t \in [\Lambda]} |F_t|$

Easier to work with a related problem Temporal Firefighter Reserve, in which defences can be delayed and added to a budget.

Can save the exact same number of vertices as in Temporal Firefighter.

All defences can be delayed until right before the vertex to be defended would burn.

Dynamically program - for each timestep t iteratively compute sets L_t containing every possible state (D, B, g, c) where:

- D is the set of *relevant* defended vertices
- B is the set of *relevant* burnt vertices
- g is the budget
- c is the total number of burnt vertices

Begin by initialising $L_0 = \{(\emptyset, \{r\}, 1, 1)\}$. Then consider every possible defence (including not defending at all) temporally adjacent to the fire, and create entries in L_1 accordingly. Continue iterating over the timesteps in this manner.

Runtime

The number of relevant burnt and defended vertices is bounded by the vertex interval membership width.

Thus there are $O(4^\omega \omega \Lambda^2)$ entries in each L_t. Additionally for each entry in L_t there are at most 2^ω defences to consider in order to compute the entries in L_{t+1}.

We do this for every timestep, so we obtain a runtime of $O(8^\omega \omega \Lambda^3)$.

Conclusions

Just restricting the structure of the underlying graph in Temporal Firefighter is not very useful.

However when we consider restricting the temporal structure, we get FPT!

What other parameters can we devise? For what problem do they work?

Recently we have considered the complexity of TEMPORAL FIREFIGHTER where we bound the number of edges per timestep - we believe it remains hard, even on trees.

Limit the edge activity and the maximum degree?

Approximations?

Related problems: graph burning, graph flooding etc.

Questions?

Bumpus, Benjamin Merlin and Kitty Meeks (2021). “Edge exploration of temporal graphs”. In: *CoRR* abs/2103.05387. arXiv: 2103.05387.

Finbow, Stephen et al. (2007). “The firefighter problem for graphs of maximum degree three”. In: *Discret. Math.* 307.16, pp. 2094–2105. DOI: 10.1016/j.disc.2005.12.053. URL: https://doi.org/10.1016/j.disc.2005.12.053.

References II

Fomin, Fedor V., Pinar Heggernes, and Erik Jan van Leeuwen (2016). "The Firefighter problem on graph classes". In: *Theor. Comput. Sci.* 613, pp. 38–50. DOI: *10.1016/j.tcs.2015.11.024*. URL: *https://doi.org/10.1016/j.tcs.2015.11.024*.

Hartnell, Bert (1995). "Firefighter! an application of domination". In: *the 24th Manitoba Conference on Combinatorial Mathematics and Computing, University of Minitoba, Winnipeg, Cadada, 1995.*