PERCOLATION AND THE FIREFIGHTER PROBLEM

ETHAN KELLY

Ethan Kelly
Email: E.Kelly.1@research.gla.ac.uk
Supervisor: Jessica Enright
Email: Jessica.Enright@glasgow.ac.uk

Department of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QN

1. Introduction

The Firefighter Problem was first discussed by Hartnell in 1995 [5] and models an outbreak of fire with a firefighter strategically blocking its path. We will discuss this original formulation of the problem and explain how it can be used in applications from the spread of disease to the spread of viral content on social media.

The other topic we will address is Percolation Theory, and how it proves very promising in answering questions about the Firefighter Problem. We will outline the original conception and the context it was considered and explain how a slight variant will prove far more useful for our purposes in considering the Firefighter Problem.

2. Background

2.1. Firefighter.

The Firefighter Problem, which we refer to as simply Firefighter, is the focus and motivation of our discussion; we, in particular, hope to show that Percolation can provide many answers to questions asked about Firefighter. We first formalise Firefighter as follows: at $t = 0$, fire breaks out at some vertex v_0 of graph G. The firefighter then 'protects' another vertex[1] of G. A protected vertex is protected for the remainder of the game; it is inflammable *ad infinitum.* Similarly, a vertex that has been on fire is 'burnt' for the rest of the game and cannot ignite again. The fire spreads to any immediate neighbouring vertices that are neither protected nor burnt. Then, the firefighter may protect another vertex, the fire spreads again and so on.[2] The following is a decision formulation given by Finbow and MacGillivray for Firefighter on a tree [2]:

[1] This has been extended to n vertices in some research, but here we wish to illustrate for simplicity the original form of the problem.

[2] It may be tempting to put the firefighting move and the fire spread into two distinct time steps, t and $t+1$ - this is beneficial in some applications (such as two player versions of Firefighter) but in general will make computation more time consuming.

FIREFIGHTER
INSTANCE: A rooted graph (G, r) and an integer $k \geq 1$.
QUESTION: Is there a finite sequence $d_1, d_2, \dots d_t$ of vertices of the graph G such that:

- i d_i is neither burned nor defended at time i,
- ii At time t, no undefended vertex is adjacent to a burning vertex, and
- iii At least k vertices are saved at the end of time t?

Common questions to ask in FIREFIGHTER include: how do we minimise the number of vertices that will be burnt? In a given class of trees, what is the average number of burnt vertices? Is the problem NP-complete?

There are many natural contextualisations of FIREFIGHTER - for instance, let each vertex be an individual and let the edges between them represent social contact. This is a simple model for disease infection. If instead we think of these edges representing virtual contact between individuals on social media, we have a model for the spread of viral internet memes [9].

2.2. **Percolation Theory.** Widely known and used in physics, statistics and mathematics, Percolation theory involves modelling scenarios as n-dimensional graphs, so application to FIREFIGHTER is not entirely unexpected. The edges between vertices in the graph can be either 'open' or 'closed' with probability p and $1 - p$ respectively. We can think of percolation problems as liquid being poured onto a porous material and whether there is a path from hole to hole along open paths through the material. Note that removing more and more edges mov†es us towards a critical point at which removing further edges would cause the graph to fall apart into smaller clusters of vertices and edges that have no access to each other [3]. This is known as 'bond' percolation, as edges correspond to bonds in many of its applications.
Several authors have suggested percolation as a possible approach to FIREFIGHTER [2].

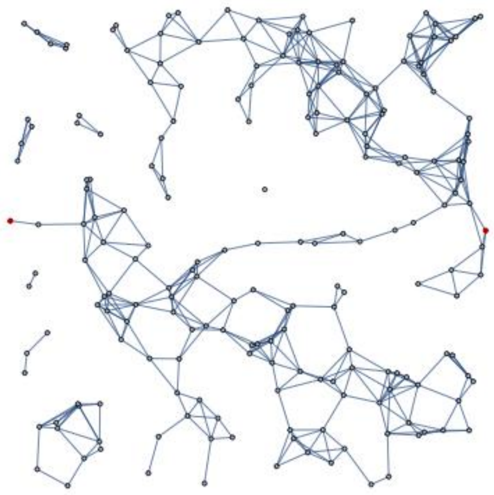


FIGURE 1. A graph at its percolation threshold [6] (Mathematica code in in Appendix A).

In this context, we could determine the critical point to see how we might contain the fire to a smaller cluster that cannot spread to the wider graph. For FIREFIGHTER, site percolation is more applicable: rather than considering open or closed *edges* ('bonds') between vertices as in bond percolation, we consider each *vertex* ('site') as being 'occupied' or 'unoccupied' with probability p and $1 - p$ respectively.

Formally, we consider a point lattice $\mathbb{L}$ and denote the open cluster as $C(x)$, where $x \in \mathbb{L}$ is the local origin of the cluster. This cluster $C(x)$ is defined as the set of all vertices that can be reached from open paths beginning at the nucleation site, x. Then, we are particularly interested in the *percolation probability*:

$$\theta(p) = \mathbb{P}_p(\,|C(0)| = \infty\,),$$

and the *critical probability* (or *percolation threshold*):

$$p_c = \sup\{\, p \mid \theta(p) = 0\,\}.$$

Here, $\mathbb{P}_p$ is the product measure given by:

$$\mathbb{P}_p = \prod_{v \in \mathbb{L}^d} \mu_v$$

where μ_v is the *Bernoulli measure*, which returns p when v is open and $1 - p$ when v is closed [8, p. 28]. Analytically, others have shown that in the case of a two-dimensional regular point lattice, the critical probability is $p_c = 1/2$ [7].

3. Potential Uses of Percolation in The Firefighter Problem

We have identified three main avenues that may be pursued in FIREFIGHTER using Percolation: the firefighter may use percolation in order to defend the graph, the fire might spread with percolation probability p or we might use percolation on the graph to form a more useful model (i.e. one that can more accurately represent an irregular population density). The former might be used when the firefighter can save more than one vertex at each turn; the latter may be more useful when modelling disease spread.

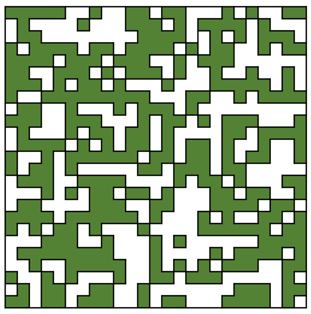


FIGURE 2. A regular (25×25) graph, percolated with probability $p = 2/3$.

3.1. **Better than random.** One potential use for the firefighter using percolation as a method of defence would be as a baseline test: in most scenarios, a method for obtaining defence strategies should be at least as effective as a random defence sequence. We could find such a random sequence using percolation for comparison purposes. Consider a sequence of vertices in graph G, written as $d_1, d_2, \ldots, d_t$. An optimal defence sequence could be found using integer programming as provided by Finbow and MacGillivray [2]:

$$\begin{array}{lll} \text{Maximise} & \displaystyle\sum_{v \in V(G)} d_v w(v) & \text{for each level } i \\ \text{subject to} & \displaystyle d_v + \sum_{\text{level}(v)=i} d_v \leq 1 & \text{for each level } i \\ & \displaystyle d_v + \sum_{u \succ v} d_u \leq 1 & \text{for every outer vertex } v \text{ of } T, \\ & d_v \in \{0,1\}. & \end{array}$$

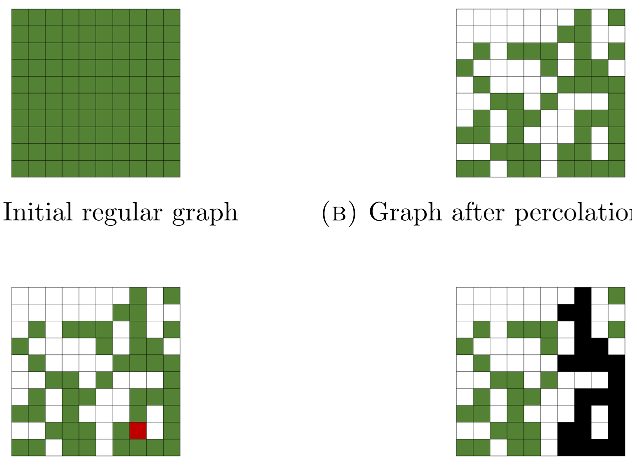
where $u \succ v$ indicates that u is an ancestor of v. The optimal strategies provided for different classes and densities of graphs here will provide an upper bound (which may indeed be impossible to attain in some cases) for success of a given strategy. We can find a lower bound using percolation, and so we have a range of success values as a starting point: if some strategy is better than random percolation, then it is worth considering, but below the particular expected optimal solution from integer programming and we can improve or find a better strategy.

We conjecture that, at the lowest graph densities, the random strategy will be very close to the optimal strategy and thus finding an improvement is at once difficult and lacking in great utility. At the very highest graph densities, while random strategies will have a very low expected best-case scenario, as will most strategies as the constraint on the firefighter that they have only one vertex to save per turn does not go as far when vertices are much better connected. Thus, finding improvements on random will again not prove very useful as random and best-case scenarios serve to show us that there is a small range of possible, not good outcomes.

3.2. **Reproduction rate.** We now focus our attention on the fire spread being determined by percolation (rather than the firefighter's defence sequence). Diseases, when there is a large enough sample size, have a basic reproduction rate associated with them, denoted R_0: for instance, measles has a basic reproduction rate $12 \leq R_0 \leq 18$ [4] and the Influenza strain responsible for the 1918 pandemic has a basic reproduction rate of $1.4 \leq R_0 \leq 2.8$ [1]. These values correspond to a basic expectation

Where we wish to consider vertices as individuals and edges as the connections between them, percolation may give us a more useful model for disease spread when we do not assume the population is well mixed and instead introduce probability functions to correspond to the likelihood one vertex is connected to another.

3.3. **Irregular population density.** A great deal of the literature surrounding Firefighter assumes a regular graph - that is, in the context of disease we assume a well-mixed population where everyone has equal probability of coming into contact with their neighbours. Of course, this is a significant simplification of reality: some individuals are very well connected and have lots of contact with others, whereas some people have significantly less contact with others. In the context of a forest fire, the density of a forest is irregular and there is a probability in the unit interval that fire can spread between two trees depending on their proximity (among other factors): an example of this can be seen in figure 2. Thus, percolation on regular grids to more closely resemble populations or forest density could lead to far more useful and realistic modelling results.

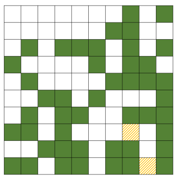

(A) Initial regular graph (B) Graph after percolation ($p = 1/2$)

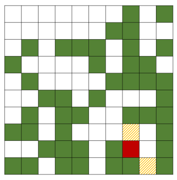
(C) Initial outbreak ($t = 0$) (D) End of outbreak ($t = 12$)

Figure 3. Outbreak of fire on a percolated graph

To illustrate this idea, we will consider a similar (perhaps precursor) to Firefighter, called The Firebreak Problem or simply Firebreak. Rather than have a firefighter reactively combatting the fire on each turn, we begin with an allocated amount of funding with which to mitigate the expected damage of the fire, and spend all of that before the fire begins. Generally, this means we have a number of edges to remove (trees to cut down or place a barrier in between) before the fire begins. Figure 3 depicts the use of percolation on a graph to model a forest fire. We can see that the model would have little utility if we began the outbreak on the graph in 3a (all of the graph would be burnt), but percolation has made this a more instructive and realistic model in figure 3b. We can see where the fire has spread in 3d, and we can see where - if we had a finite amount of resources - we must focus our efforts and funding in introducing a firebreak.

If, for instance, we could only remove two vertices as a firebreak, we ought to make those three the ones shown in yellow in figure 4. This is an example of how percolating a graph for use in Firefighter (and indeed Firebreak) can assist in generating a more


(A) Position of Firebreak (in yellow)


(B) Fire onset with firebreak ($t = 0$)

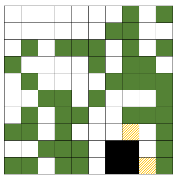

(C) End of outbreak ($t = 3$)

FIGURE 4. Outbreak of fire on a defended percolated graph

realistic and useful model for contagion. There is much interest in the regular graph examples, but our modelling contexts of interest call for a more sporadic graph density and so we, in future research, will try and prove conjectures regarding the minimum number of vertex defences required in FIREBREAK for percolated graphs of given percolation threshold and dimension (or infinite graphs) and analogous questions in FIREFIGHTER.

APPENDIX A. PERCOLATION IN MATHEMATICA

Below is some sample Mathematica code for producing a helpful tool to visualise percolation on a network and finding the percolation threshold and is based on a Mathematica Stack Exchange question [6].

```
* Create sample dataset *
points = RandomReal[10 {-1, 1}, {200, 2}];

ListPlot[pts, AspectRatio -> Automatic,
 Epilog -> {Red, Point[pts[[63]]], Point[pts[[90]]]}]

* Define distance matrix between points *
dst = Outer[EuclideanDistance, pts, pts, 1]; // Timing

* Set of possible edges in graph, sorted by length *
edges = Subsets[Range@Length[pts], {2}];
edges = SortBy[edges, Extract[dst, #] &];

* Index furthest left and right vertices as start and end *
start = First@Ordering[pts[[All, 1]], 1];
end = First@Ordering[pts[[All, 1]], -1];
```

```
* Get minimal length edge that we need to include *
idx = Module[{f}, Do[Set @@ f /@ (edges[[i]]);
   If[f[start] === f[end], Return[i]], {i, Length[edges]}]]

Extract[dst, edges[[idx]]]

* Manipulate to add edges one by one in increasing
   length until we have percolation threshold *
Manipulate[
 HighlightGraph[
  Graph[Range@Length@pts, UndirectedEdge @@@ Take[edges, i],
   VertexCoordinates -> pts], {start, end}], {i, 1, idx, 1}]
```

References

[1] N. M. Ferguson, D. A. Cummings, C. Fraser, J. C. Cajka, P. C. Cooley, and D. S. Burke, *Strategies for mitigating an influenza pandemic*, Nature, 442 (2006), pp. 448–452.

[2] S. Finbow and G. MacGillivray, *The firefighter problem: A survey of results, directions and questions*, The Australasian Journal of Combinatorics, 43 (2009).

[3] G. Grimmett, *Percolation*, Springer-Verlag, Berlin, 2 ed., 1999.

[4] F. M. Guerra, S. Bolotin, G. Lim, J. Heffernan, S. L. Deeks, Y. Li, and N. S. Crowcroft, *The basic reproduction number (r_0) of measles: a systematic review*, The Lancet Infectious Diseases, 17 (2017), pp. 420–428.

[5] B. L. Hartnell, *Firefighter! an application of domination*, in 25th Manitoba Conference on Combinatorial Mathematics and Computing, University of Manitoba in Winnipeg, Canada, 1995.

[6] S. Horvát, *Mathematica stack exchange answer.* mathematica.stackexchange.com/a/5173, October 2017.

[7] H. Kersten, *The critical probability of bond percoation on the square lattice equals 1/2*, Communications in mathematical physics, 74 (1980), pp. 41–59.

[8] A. Klenke, *Probability Theory*, Springer, Berlin Heidelberg, 2 ed., 2014.

[9] J. D. O'Brien, I. K. Dassios, and J. P. Gleeson, *Spreading of memes on multiplex networks*, New Journal of Physics, 21 (2019), p. 025001.

Ethan Kelly
Email: E.Kelly.1@research.gla.ac.uk

Supervisor: Jessica Enright
Email: Jessica.Enright@glasgow.ac.uk

Department of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QN