AN ANALYSIS OF THE WEIGHTED FIREFIGHTER PROBLEM

CHRISTOPHER DUFFY, DR. GARY MACGILLIVRAY
DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF VICTORIA, CANADA

ABSTRACT. For the Firefighter Process with weights on the vertices, we show that the problem of deciding whether a subset of vertices of a total weight can be saved from burning remains NP-complete when restricted to binary trees. In addition, we show that a greedy algorithm that defends the vertex of highest degree adjacent to a burning vertex is not an ϵ*-approximation* algorithm for any $\epsilon \in (0,1]$ for the problem of determining the maximum weight that can be saved. This closes an open problem posed by MacGillivray and Wang.

1. INTRODUCTION

We consider the following discrete-time process: at $t = 0$ some vertex of a simple graph begins burning. At each subsequent timestep we defend a vertex from burning and the fire spreads from all burning vertices to all undefended neighbours. This process was originally introduced by Hartnell [4] at the *25th Manitoba Conference on Combinatorial Mathematics and Computing* at the University of Manitoba in 1995. In 2009, MacGillivray and Finbow published a survey on the results to date [3].

A first question to ask about this process is given a graph and a vertex, if the fire starts at that vertex, how many of the vertices can be prevented from burning. The problem of deciding whether a given quantity of vertices can be saved is NP-complete even when restricted to graphs with maximum degree three [6]. Consequently, algorithms which approximate the maximum number of vertices that can be saved are of particular interest. In [5] Hartnell and Li prove the existence of a $\frac{1}{2}$ *– approximation* algorithm for this quantity for trees.

In [7], MacGillivray and Wang examine a greedy algorithm for trees, which defends the vertex of highest degree adjacent to a burning vertex. They leave open the existence of an $\epsilon \in (0,1]$ such that this algorithm is

2010 Mathematics Subject Classification: 05C57, 05C85
Keywords: Graph Theory, Algorithms, Complexity, Firefighter

an ϵ – *approximation* algorithm for the maximum number of vertices that can saved. We will show that no such ϵ exists.

In addition to examining approximation algorithms, we will consider the generalisation of the problem of deciding the maximum number of vertices that can be saved formed by introducing weights on the vertices. This generalisation was among the open problems listed in [3]. It is known that the problem of deciding whether a given quantity of vertices can be saved, named FIREFIGHTER, is NP-complete even when restricted to trees with maximum degree three. The problem can be solved in polynomial time for graphs of maximum degree three in which the fire breaks out at a vertex of degree two [6]. On the other hand, the problem of determining whether a given set of vertices can be saved, named SFIRE, is NP-complete even when restricted to graphs of maximum degree three when the fire breaks out at a vertex of degree two [2]. We will show that when a weight is given to each of the vertices, the problem of determining whether a subset of vertices of a total weight can be saved, named WFIRE, is NP-complete even when restricted to binary trees.

2. Preliminaries and Definitions

If G is a graph and $r \in V$, we call the ordered pair (G, r), a *rooted graph* and r the *root*. This will often be shortened to *a rooted graph* (G, r). Consider a rooted graph (G, r). The *firefighter process* proceeds as follows: At time $t = 0$ a fire breaks out at r. At each subsequent timestep, one unburned vertex of G may be defended from burning and the fire spreads to each undefended vertex adjacent to a burning vertex. Once a vertex is defended it remains defended for the remainder of the process. Similarly, once a vertex is burning it remains burning for the remainder of the process. The process ends when every burning vertex has all of its neighbours either burning or defended. At the conclusion of the process any vertex that is neither burned nor defended is called *protected*. Together, the defended and the protected vertices are the *saved* vertices. We define $MVS(G, r)$ as the maximum number of vertices that can be saved. We call the sequence of defended vertices $D = (d_1, d_2, \ldots, d_t)$, where d_i is defended at time i, a *strategy*.

Following [3], we can consider any time in the process as the start of a new process on a reduced graph. For example, consider the process on the first graph shown in Figure 1. The second graph in this figure shows the state of the process after defending x and y in the first two timesteps. At this point, if we delete all the vertices that have been defended, as well as those that can never burn, and identify all vertices that are burning in to a single vertex, we have a smaller graph and a single burning vertex. We

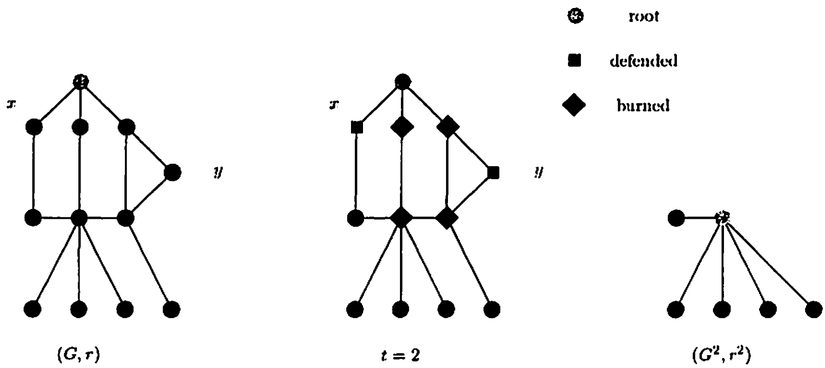

FIGURE 1

may consider this a new rooted graph. This is the third graph shown in the figure. We will denote the new rooted graph formed in this manner after time k as (G^k, r^k)

For a rooted graph (G, r) and a weight function $w : V(G) \to \mathbb{Z}$ we define $MVS_w(G, r)$ to be the maximum sum of the weights of a subset of vertices that can be saved and b_w^D the sum of the weights of the burned vertices under a particular strategy, D. Following [3], we define the Weighted Firefighter Decision Problem as follows:

WFIRE
INSTANCE: A rooted graph (G, r), an integer k and a weight function $w : V(G) \to \mathbb{Z}$.
QUESTION: Is $MVS_w(G, r) \geq k$? That is, is there a strategy $D = (d_1, d_2, \ldots, d_t)$ such that if the fire breaks out at r, then

- after time t no undefended vertex is adjacent to a burning vertex, and
- the sum of the weights of the non-burning vertices after time t is at least k?

In addition to Weighted Firefighter Decision Problem we also consider the Weighted Firefighter Optimization Problem:

OPT-WFIRE
INSTANCE: A rooted graph (G, r), and a weight function $w : V(G) \to \mathbb{Z}$.
PROBLEM: Over all strategies $D = (d_1, d_2, \ldots, d_t)$, such that if the fire breaks out at r, where

- after time t no undefended vertex is adjacent to a burning vertex,

MAXIMISE: $\sum_{v \in V} w(v) - b_w^D$.

This definition of WFIRE allows us to frame the problem of deciding whether a given number of vertices can be saved as an instance of WFIRE. For a rooted graph (G, r), consider a weight function which assigns weight 1 to each of the vertices. If the answer is yes for WFIRE the answer for the corresponding instance of FIREFIGHTER must also be yes. This gives directly that WFIRE is NP-complete, even when restricted to graphs with maximum degree three (see [3]).

3. Greedy Strategies for the Weighted Firefighter Problem

Since the problem of deciding whether a given number of vertices can be saved can be represented as an instance of WFIRE, we will consider the problem of trying to approximate $MVS_w(T, r)$

In [7] MacGillivray and Wang examine the Degree Greedy Algorithm. This algorithm defends, at each step, the vertex of highest degree adjacent to a burning vertex. They show that this strategy finds an optimum solution for caterpillars, but not for arbitrary trees. They leave open the existence of a constant $c \in (0, 1]$ such that the algorithm saves at least $c \cdot MVS(T, r)$ vertices [7]. Here we show that such a constant does not exist.

Theorem 3.1. *If $Greedy_d(T, r)$ denotes the sum of weight of vertices saved using the Degree Greedy Algorithm, then there is no $c \in (0, 1]$ such that for all rooted trees (T, r)*

$$Greedy_d(T, r) \geq c \cdot MVS_w(T, r).$$

Proof. Let J be a full and complete binary tree of height k rooted at v. Let P_k be the path of length k with vertex sequence $x_0, x_1, \ldots, x_k$. For $0 \leq i \leq k$, let S_i be a copy of the star $K_{1,3}$ with centre vertex w_i. Construct a tree T_k with $2^{k+1} + 5k - 1$ vertices by joining x_0 to v, and x_i to w_i for all $i \leq k$. (See Figure 2)

Consider OPT-WFIRE on (T_k, x_0) where $w(u) = 1$ for all $u \in V$. With strategy $D = (v, x_2, z_1)$ a total weight of 9 burns. This strategy is optimal for $k > 4$. Thus $MVS_w(T_k, r) = 2^{k+1} + 5k - 10$. The Degree Greedy Algorithm yields the strategy $D' = (w_0, w_1, \ldots, w_k)$. Under D' every vertex of J will burn along with every vertex on P_k. Thus $Greedy_d(T_k, r) = 4k$.

$$\lim_{k \to \infty} \frac{Greedy_d(T_k, r)}{MVS_w(T_k, r)} = \lim_{k \to \infty} \frac{4k}{2^{k+1} + 5k - 10} = 0.$$

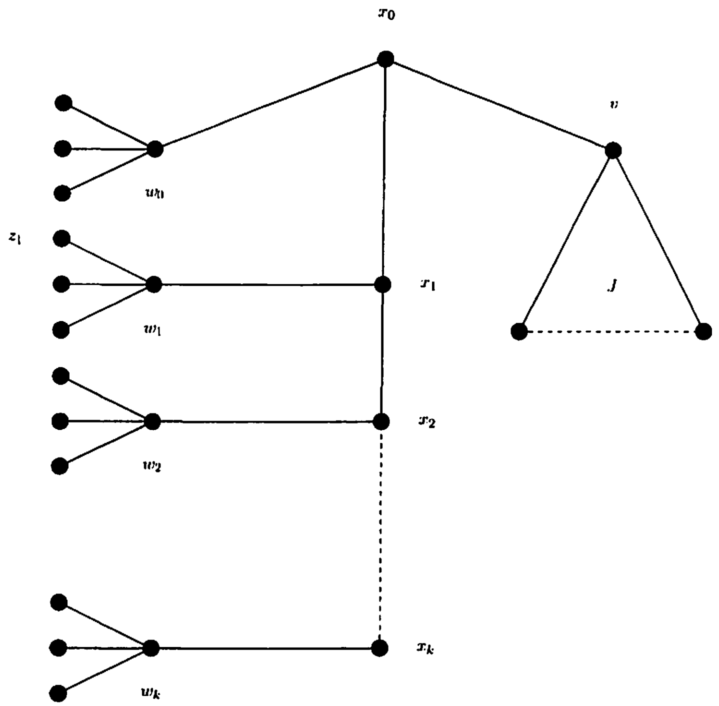

FIGURE 2

There is no $c \in (0,1]$ such that the Degree Greedy Algorithm saves at least $c \cdot MVS_w(T,r)$ vertices. □

Corollary 3.2. *For any $\epsilon > 0$, the Degree Greedy Algorithm is not an ϵ - approximation algorithm for OPT-WFIRE.*

This closes the problem posed by MacGillivray and Wang in [7].

In [5], Hartnell and Li discuss a greedy algorithm for trees that defends the vertex adjacent to the fire with the most successors. This algorithm, the Weighted Greedy Algorithm, is a $\frac{1}{2}$ – *approximation* algorithm for the problem of determining the maximum number of vertices that can be saved [5]. Modifying the algorithm to defend the vertex whose defence would save a subset of vertices with the greatest total weight yield a $\frac{1}{2}$ – *approximation* algorithm for OPT-WFIRE. This can be shown using the same argument as in [5] with minor variations to account for the weights.

Theorem 3.3. *If $Greedy_w(T,r)$ denotes the number of vertices saved using the Weighted Greedy Algorithm, then*

$$\frac{1}{2} \leq \frac{Greedy_w(T,r)}{MVS_w(T,r)}.$$

That is, the Weighted Greedy Algorithm is a $\frac{1}{2}$-approximation algorithm for OPT-WFIRE on trees.

4. Complexity Results for the Weighted Firefighter Problem

We turn now to complexity results for the Weighted Firefighter Problem. For FIREFIGHTER, a sharp dividing line exists based on maximum degree of the graph and the degree of the root. However this is not the same when we consider the problem of deciding whether a given set of vertices can be saved from burning.

Following [3], we define the problem of deciding whether a given set of vertices can be saved as follows:

SFIRE
INSTANCE: A rooted graph (G,r) and a subset $S \subseteq V(G) - \{r\}$.
QUESTION: If the fire breaks out at r, is there a strategy under which no vertices in S burn? That is, does there exist a strategy $D = (d_1, d_2, \ldots, d_t)$ such that if the fire breaks out at r, then

- after time t no undefended vertex is adjacent to a burning vertex, and
- no vertex in S is burned after time t?

Theorem 4.1. [2] *SFIRE is NP-complete even when restricted to graphs with maximum degree three where the fire starts at a vertex of degree two. The problem is in P when restricted to binary trees.*

As we saw in Section 2, we can use WFIRE to represent an instance of FIREFIGHTER by setting all weights to 1. Similarly, we can also use WFIRE to represent an instance of SFIRE. For an instance of SFIRE on (G,r) with set S we can form an instance of WFIRE on (G,r) where

$$w(v) = \begin{cases} 0 & : v \notin S \\ 1 & : v \in S \end{cases}$$

and by setting $k = |S|$ [3].

Using this method represent an instance of SFIRE as an instance of WFIRE, we are able to gain a further complexity results for WFIRE.

Corollary 4.2. *WFIRE is NP-complete even when restricted to graphs with maximum degree three rooted at a vertex of degree two.*

Proof. An instance of SFIRE on (G, r) with set S can be expressed as an instance of WFIRE. SFIRE is NP-complete even when restricted to graphs with maximum degree three rooted at a vertex of degree two [2]. Therefore WFIRE is NP-complete by restriction. □

Since SFIRE can be decided in polynomial time for binary trees, we will consider the restricion of WFIRE to binary trees. The restriction of SFIRE to binary trees requires that we only consider sets S where S is the leaves of the tree. Surprisingly, knowing when the leaves of an arbitary tree can be saved helps us establish a complexity result for WFIRE on binary trees. To that end, we consider the following decision problem:

3FLFIRE
INSTANCE: A rooted tree (T, r) with $\Delta(T) \leq 3$.
QUESTION: If the fire breaks out at r, is there a strategy under which no leaf of T burns? That is, does there exist a strategy $D = (d_1, d_2, \ldots, d_t)$ such that if the fire breaks out at r, then

- after time t no undefended vertex is adjacent to a burning vertex, and
- no leaf in T is burned after time t.

Theorem 4.3. [6] *3FLFIRE is NP-complete.*

Theorem 4.4. *The restriction of WFIRE to binary trees is NP-complete.*

Proof. The problem is clearly in NP. The transformation is from 3FLFIRE. Suppose an instance of 3FLFIRE, a rooted tree (T', r') with $deg(r) = 3$, is given. We construct an instance (T, r) of WFIRE, in which (T, r) is a binary tree, as follows.

Let G be the binary tree shown in Figure 3. Suppose that the three neighbours of r' in T' are u', v' and w'. Delete r', and identify u', v' and w' with the vertices u, v and w of G. This completes the construction of (T, r), which is a binary tree. We complete the transformation by defining the weight function $w : V(T) \to \{-1, 0, 1\}$ and integer k. Let L be the set of leaves of T. Set $w(y) = w(z) = -1$, $w(a) = 1$ for all $a \in L$, and $w(a) = 0$ otherwise. Finally set $k = |L| + 1$. The transformation can be accomplished in polynomial time.

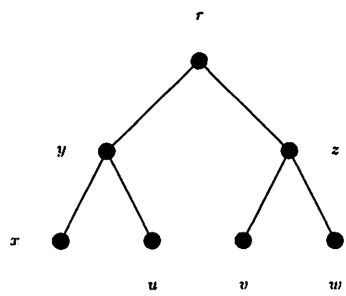

FIGURE 3

Note that in order to save a set of vertices of total weight at least k, the total weight of the burning vertices cannot exceed -2. For this to be possible, both y and z must burn and x must be saved.

Suppose there is a strategy, D, for WFIRE on (T, r) under which the total weight of the burning vertices is at most -2. Since both y and z must burn, we must have $d_1 = x$, or $d_1 \in \{u, v, w\}$ and $d_2 = x$. In the first case, the instance (T^1, r^1) is equivalent to an instance of 3FLFIRE on (T', r'). In the second case, the instance (T^2, r^2) is equivalent to an instance of 3FLFIRE on (T', r') after one of u', v' or w' have been defended. Since the total weight of the burned vertices is -2, D must save all leaves of T'. Hence there is a strategy for 3FLFIRE on (T', r') under which all leaves are saved.

Now suppose there is a strategy, $D' = (d'_1, d'_2, \ldots, d'_t)$, for 3FLFIRE on (T', r') under which all leaves are saved. Consider now applying the the strategy $D = (x, d'_1, d'_2, \ldots, d'_t)$ to (T, r). We see that $b_w^D \leq -2$, as required.

Therefore, the restriction of WFIRE to binary trees is NP-complete. □

We notice that the construction in the above proof relies on vertices whose weight is negative. In such a scenario it is possible that when trying to maximise the weight of the saved vertices it becomes advantageous to have some vertices burn (particularly, negatively weighted ones). By removing this possibility (i.e., by restricting to non-negative weights), the problem can be solved in polynomial time. This result was discovered independently by Bazgan, Chopin and Ries in their work on the Firefighter Problem on Trees using more than one firefighter [1].

Theorem 4.5. *WFIRE is in P when restricted to binary trees and non-negative weights*

Proof. For binary tree (T, r) with non-negative weights on the vertices there exists a strategy D that realises $MVS_w(T, r)$ for which each vertex of D will be adjacent to a burning vertex [3]. As such, at most one new vertex will burn at each timestep. Thus the process will end when a newly burning vertex is either leaf or a vertex of degree two. Let F be the set of such vertices. Let $P_u = rv_1v_2v_3 \dots v_ku$ be the path from r to u on T and let $w(P_u) = \sum_{x \in P_u} w(x)$. Therefore

$$MVS_w(T, r) = min_{u \in F}\{P_u\}.$$

□

Given the relationship between SFIRE and WFIRE it is unsurprising that even when restricting to trees with maximum degree three rooted at a vertex of degree two, WFIRE remains NP-complete. There is a trend to be observed here: restriction of these problems to instances for which the degree of the fire is guaranteed to not exceed two seem to be in P. On the other hand, restriction to instances for which the degree of the fire may exceed two seem to be NP-complete.

REFERENCES

[1] C. Bazgan, M. Chopin, and B. Ries. The firefighter problem with more than one firefighter on trees. *Discrete Applied Mathematics*, 161:899–908, 2011.

[2] C. Duffy and G. MacGillivray. The firefighter problem: Saving sets of vertices of degree three graphs. Manuscript, September 2011.

[3] S. Finbow and G. MacGillivray. The firefighter problem: A survey of results, directions and questions. *Australasian Journal of Combinatorics*, 43:57–77, 2009.

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

[5] B. Hartnell and Q. Li. Firefighting on trees: How bad is the greedy algorithm? *Proceedings of the Thirty-first Southeastern International Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, 145:187–192, 2000.

[6] G. MacGillivray and A.D King. The firefighter problem for cubic graphs. *Discrete Mathematics*, 310:614–621, 2010.

[7] G. MacGillivray and P. Wang. On the firefighter problem. *J. Combin. Math Comput.*, 47:83–96, 2003.