The Firefighter Problem For Cubic Graphs

Andrew King and Gary MacGillivray

Abstract

We show that the firefighter problem is NP-complete for cubic graphs. We also show that given a rooted tree of maximum degree three in which every leaf is the same distance from the root, it is NP-complete to decide whether or not there is a strategy that protects every leaf from the fire, which starts at the root. By contrast, we describe a polynomial time algorithm to decide if it is possible to save a given subset of the vertices of a graph with maximum degree three, provided the fire breaks out at a vertex of degree at most two.

1 Introduction

We consider a discrete-time, dynamic problem introduced by B. Hartnell in 1995 [9]. Let (G, r) be a connected rooted graph. At time 0, a fire breaks out at r. At each subsequent time interval, the firefighter *defends* some vertex which is not on fire, and then the fire spreads to all undefended neighbours of each *burning* (i.e., on fire) vertex. Once burning or defended, a vertex remains so for all time intervals. The process ends when the fire can no longer spread. The firefighter optimization problem is to determine the maximum number of vertices that can be *saved*, i.e., that are not burning when the process ends. The firefighter decision problem is stated formally below:

FIREFIGHTER
INSTANCE: A rooted graph (G, r) and an integer $k \geq 1$.
QUESTION: If the fire breaks out at r, is there a strategy under which at most k vertices burn? That is, does there exist a finite sequence $d_1, d_2, \ldots, d_t$ of vertices of G such that, if the fire breaks out at r, then,

(i) vertex d_i is neither burning nor defended at time i,
(ii) at time t no undefended vertex is adjacent to a burning vertex, and
(iii) at most k vertices are burned at the end of time t.

We now briefly review some previous work on the firefighter problem. Algorithms for the optimization version of the problem on two and three dimensional grid graphs are presented in [14]. Bounds and some exact values for the maximum number of vertices that can be saved for two dimensional grids are given in [12]. In the same paper the problem is shown to be NP-complete for bipartite graphs, and its restriction to trees is considered. Exponential algorithms for solving the firefighter problem on trees are described (one of these runs in linear time for binary trees), and a polynomial-time algorithm for a subclass of trees related to perfect graphs is given. It is proved in [11] that the greedy algorithm is a 2-approximation algorithm on trees, that is, the maximum number of vertices saved is never more than twice the number saved using the greedy algorithm. The firefighter problem on infinite grids is considered in [5, 13]. Other aspects of the firefighter problem are studied in [3] and, very recently, [1]. The latter manuscript contains a good survey of previous work mentioned, but not discussed in detail, here. Related topics are examined in [2, 7, 8, 10].

It was proved in [4] that the firefighter problem is NP-complete for trees of maximum degree three and, by contrast, that the problem can be solved in polynomial time for graphs of maximum degree three if the fire breaks out at a vertex of degree two. The main result of the first part of this paper makes the dividing line a little bit sharper: we show that is the firefighter problem is NP-complete for cubic graphs. In the second part of the paper we consider the question of deciding whether a specified set of vertices can be saved. This problem is shown to be NP-compete. In particular, we prove that for trees of maximum degree three, it is NP-complete to decide if it is possible to save every leaf, even if all of the leaves are at the same distance from the root. By contrast, we show that if the fire breaks out at a vertex of degree at most two, then there is a polynomial time algorithm to decide if it is possible to save a given subset of the vertices of a graph with maximum degree three.

2 Review of previous work on trees with maximum degree three

In this section we review the constructions from [4], as our proofs in subsequent sections of the paper require detailed information about them. The first step in proving that the firefighter problem is NP-complete for trees of maximum degree three is to show that the problem is NP-complete for trees in which all vertices except the one where the fire starts have degree at most three. This construction is subsequently adapted to prove the main result of that paper. We describe both constructions and note the points from the proofs that are important in our current work. The decision problems that are considered are formally stated first.

3-T′-FIRE
INSTANCE: A rooted tree (T, r) such that $d(r) = 2^m + 2$ for some positive integer m and every other vertex in T has degree at most 3, and a positive integer k.
QUESTION: If the fire breaks out at r, is there a strategy such that at most k vertices burn?

3-T-FIRE
INSTANCE: A rooted tree (T, r) with maximum degree $\Delta(T) \leq 3$ and a positive integer k.
QUESTION: If the fire breaks out at r, is there a strategy such that at most k vertices burn?

Each of the two problems just mentioned is proved to be NP-complete using a reduction from a variant of Not All Equal 3-SAT:

RESTRICTED NAE 3-SAT (see [4])
INSTANCE: An ordered pair (B, C) consisting of a set B of boolean variables and a set C of clauses over B each of which is a disjunction of three literals, where $|B| = 2^m$ for some integer $m \geq 2$, exactly $|C|/2$ clauses have no negated literals, and the remaining clauses are obtained from these by replacing each literal with its negation.
QUESTION: Is there a truth assignment for B such that every clause in C contains at least one true literal and at least one false literal?

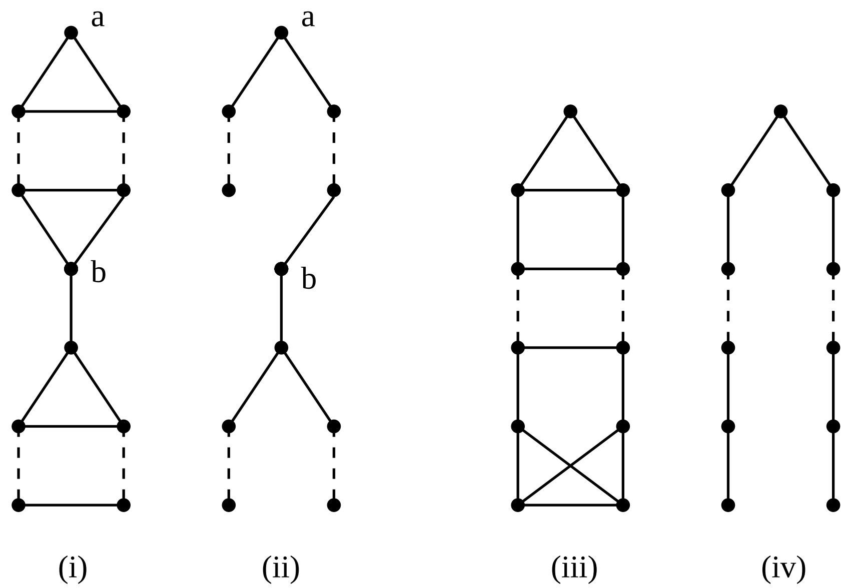

Figure 1: (i) Snake (ii) Snake tree (iii) Ladder (iv) Ladder tree

We now describe two types of graphs that are important to the constructions throughout this paper. A graph of the type shown in Figure 1 (i), and rooted at the vertex a, is called a *snake*. A snake of diameter n, with distance m from a to b, is denoted by $\mathcal{S}(n,m)$. A *snake tree* is a spanning tree of a snake of the form shown in Figure 1 (ii). A snake tree of diameter n, with distance m from a to b, is denoted by $\mathcal{S}^T(n,m)$. A graph of the type shown in Figure 1 (iii), with at least five vertices and and rooted at the vertex of degree two belonging to the triangle, is called a *ladder*. The notation $\mathcal{L}(n)$ is used to denote a ladder of diameter n. A *ladder tree* is a spanning tree of a ladder of the form shown in Figure 1 (iv). The notation $\mathcal{L}^T(n)$ is used to denote a ladder of diameter n.

The constructions described below make use of binary trees that are both full and complete. A rooted tree (T,r) is called *full* if all leaves occur at the same level (*i.e.* all leaves are at the same distance from the root). A binary tree (T,r) is called *complete* if every internal vertex has exactly two children. Thus, a complete and full binary tree of height h has exactly $2^{h+1}-1$ vertices of which 2^h are leaves, each of which is at distance h from r.

Let x be a vertex of a graph G, and let (T, r) be a rooted graph. When we *root a copy of T at x*, we construct a new graph from the disjoint union of G and T by identifying the vertices x and r. In the construction described below, we will normally root either complete and full binary trees, or paths, at vertices of other graphs. We always assume that the root vertex of a path is of degree one in the path.

We now describe the polynomial time transformation that reduces RESTRICTED NAE 3-SAT to 3-T′-FIRE. Suppose an instance (B, C) of RESTRICTED NAE 3-SAT, where $B = \{b_1, b_2, \ldots, b_b\}$, the integer m is defined by $b = 2^{m-1}$, and $C = \{c_1, c_2, \ldots, c_n\}$, is given. Assume that $n > b \geq 4$, (clauses can be duplicated to ensure $n > b$; this assumption will simplify analysis). Also assume that for $i = 1, 2, \ldots, n/2$, the clause c_{2i} arises from negating each variable in the clause c_{2i-1}. Let $p = \lceil \log_2(n) \rceil + 2$. The construction of the rooted tree (T', r') has two phases. First, we construct a full rooted tree (T_1, r') with height $b + p$ in which the degree of r is $2^m + 2$. We subsequently augment (T_1, r'), without changing the degree of r', to construct our final rooted tree.

Start with the single vertex r'. For $i = 1, 2, \ldots, b$, root two paths of length i at r and call the vertices of degree one in the resulting graph b_i and $\bar{b}_i$. These vertices correspond to the literals in the instance of RESTRICTED NAE 3-SAT in the obvious way. At each of b_i and $\bar{b}_i$, root a complete and full binary tree of height p. From each leaf of these trees root a path of the appropriate length so that the vertex of degree one in the resulting graph is distance $b+p$ from r'. (The paths rooted at b_b and $\bar{b}_b$ have length zero.) Call these leaves $t_{b_i,1}, t_{b_i,2}, \ldots, t_{b_i,2^p}$ and $t_{\bar{b}_i,1}, t_{\bar{b}_i,2}, \ldots, t_{\bar{b}_i,2^p}$ respectively. Next, root two paths of length $b + 1$ at r' and call the resulting vertices of degree one b_0 and $\bar{b}_0$. From these vertices, root complete and full binary trees of height $p-1$, calling their leaves $t_{b_0,1}, t_{b_0,2}, \ldots, t_{b_0,2^{p-1}}$ and $t_{\bar{b}_0,1}, t_{\bar{b}_0,2}, \ldots, t_{\bar{b}_0,2^{p-1}}$, respectively. The tree constructed so far is (T_1, r'). Note that r' has degree $2b + 2 = 2^m + 2$.

The number of vertices of T_1 is

$$\begin{aligned} |V(T_1)| &= 1 + 2(1 + 2 + \cdots + b) + 2b(2^{p+1} - 2) + \\ &\quad 2 \cdot 2^p((b-1) + (b-2) + \cdots + 0) + 2(b+1) + 2(2^p - 2). \end{aligned}$$

The number of these that are leaves is $2b \cdot 2^p + 2 \cdot 2^{p-1} = (2b+1)2^p$.

We now construct (T', r') by augmenting (T_1, r') as follows. For $1 \leq j \leq 2^{p-1}$, add children x_j and y_j from vertex $t_{b_0,j}$, and children $\bar{x}_j$ and $\bar{y}_j$ from $t_{\bar{b}_0,j}$. At each of the vertices just added, root a copy of $\mathcal{L}^T(3n + 1)$. For each

i and j with $1 \le i \le b$ and $1 \le j \le n$, do the following: if b_i is in clause c_j, root a copy of $\mathcal{S}^T(3n+2, 3j-1)$ at $t_{b_i,j}$ and a copy of $\mathcal{S}^T(3n+2, 3j)$ at $t_{\bar{b}_i,j}$. If $\bar{b}_i$ is in clause c_j, root a copy of $\mathcal{S}^T(3n+2, 3j-1)$ at $t_{\bar{b}_i,j}$ and a copy of $\mathcal{S}^T(3n+2, 3j)$ at $t_{b_i,j}$. At each remaining unaltered leaf of T_1, root a copy of $\mathcal{L}^T(3n+2)$. This completes the construction.

The number of vertices of T' is $|V(T_1)| + 2^p(6n+3) + 2b \cdot 2^p(6n+4) - 12b$. The height of (T', r') is $d = b + p + 3n + 2$.

To complete the instance of 3-T′-FIRE, the integer k' is defined as:

$$\begin{aligned} k' &= |V(T)| - \left(\sum_{i=1}^{b} [2^p(6n+6+b-i) - 1] + \right. \\ &\left. + \sum_{i=0}^{p} [2^{p-i}(6n+4) - 1] + 9n^2 + \frac{15n}{2} + 1 \right). \end{aligned} \quad (1)$$

It is proved in [4] that at least k' vertices are burned no matter what strategy is used to defend the tree, and that the answer for the given instance RESTRICTED NAE 3-SAT is YES if and only if there is a strategy for the firefighter problem on (T', r') under which exactly k' vertices are burned. We next describe such a strategy.

Let τ be a truth assignment for the variables in B. The strategy $f(\tau)$ for the firefighter problem is defined as follows: For $i = 1, 2, \ldots, b$, if b_i is true, defend b_i at time i and otherwise defend $\bar{b}_i$ at time i. At time $b+1$, defend $\bar{b}_0$. From time $b+2$ to $b+p$, defend the unprotected descendant of b_0 which is not on the path from r to x_1. At time $b+p+1$, defend x_1. For time $i = b+p+2$ to $b+p+3n+2$, defend the tree greedily, that is, at time i defend a vertex at level i with the largest number of descendants. Note that, in case of a tie, the subtrees rooted at each such vertex are isomorphic. Assuming a predetermined tie-breaking scheme, the function $f(\tau)$ is well defined.

Theorem 1. [4] *The truth assignment τ is a satisfying truth assignment for a given instance (B, C) of RESTRICTED NAE 3-SAT if and only if the strategy $f(\tau)$ results in exactly k' vertices being burned in the tree (T', r') constructed above. Conversely, any strategy under which at most k' vertices of (T', r') are burned is obtained from some satisfying truth assignment for (B, C).*

In the following theorem, the structure of an instance of RESTRICTED NAE 3-SAT plays an important role. Recall that the clauses come in pairs:

for each clause involving b_i, b_j, and b_k there is a corresponding clause containing $\bar{b}_i, \bar{b}_j$, and $\bar{b}_k$. By construction of T, each of these two clauses gives rise to a snake tree which is rooted at a descendant of b_i, and similarly there are two such snake trees rooted at some descendant of each vertex involved in the clause pair.

Theorem 2. [4] *Let (T', r') and k' be as above. In any strategy under which at most k' vertices burn, the vertex defended at times $b + p + 2$ through $b + p + 3n + 1$ can be chosen to belong to a snake tree. Each such vertex is either of degree three, or is of degree two and the parent of a vertex of degree three. Further, for $i = 1, 2, \ldots, b$, either all leaves of the snake trees which are at level $b + p + 3n + 2$, descendants of b_i, and arising from a clause pair burn, or no such leaf burns. The same statement holds for $\bar{b}_i$.*

We now turn our attention to describing how this construction was adapted to prove NP-completeness of 3-T-FIRE. The reduction is once again from RESTRICTED NAE 3-SAT. The idea is to use (T', r') and k' as described above to obtain a rooted tree (T, r) with maximum degree three and $deg(r) = 3$, and an integer k so that at most k vertices burn in (T, r) if and only if at most k' vertices burn in (T', r').

The construction of T begins with the single vertex r. Join three new vertices to r and, at each of these, root a full, complete binary tree of height $m - 1$ (where m is defined by $b = 2^{m-1}$, as above). For the moment consider the tree constructed so far as being ordered, so that its leaves are ordered from left to right. At each of the first $2^m - 1$ of these leaves, root a copy of a full, complete binary tree F of height $h = \lceil \log_2 |V(T)| \rceil + 3$. Label the remaining leaves from left to right as $r_0, r_1, \ldots, r_b$.

For $i = 0, 1, \ldots, b$, let (R_i, w_i) and $(\bar{R}_i, \bar{w}_i)$ denote the subtree of T' rooted at the unique neighbour of r' on the (r', b_i)-path and $(r', \bar{b}_i)$-path, respectively. Let (S_i, z_i) be the rooted tree constructed from (R_i, w_i) and $(\bar{R}_i, \bar{w}_i)$ by adding a new vertex z_i and joining it to w_i and $\bar{w}_i$.

To complete the construction of T, for $i = 1, 2, \ldots, b$, root a copy of S_i at r_i. Finally, set $k = k' + 2^m - 1 + m$. As before, each satisfying truth assignment leads to a strategy under which exactly k vertices are burned and, conversely, any strategy under which at most k vertices are burned leads to a satisfying truth assignment.

Theorem 3. [4] *Let (T, r) and k be as above. In any strategy under which at most k vertices burn, no vertex among $r_0, r_1, \ldots, r_b$ is defended.*

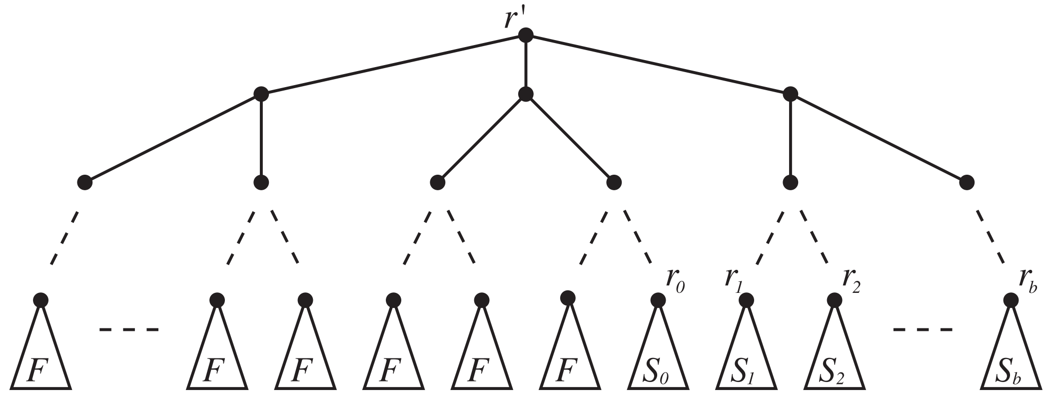

Figure 2: The construction of T' from T

It follows that if there is a satisfying truth assignment, then a strategy which guarantees that at most k vertices will burn is to defend an ancestor of a copy of F at times one through m, and then to employ that same strategy as for 3-T′-FIRE at the remaining times.

Theorem 4. [4] *The truth assignment τ is a satisfying truth assignment for a given instance (B, C) of RESTRICTED NAE 3-SAT if and only if the strategy of defending an ancestor of a copy of F at times one through m, and then defending the same vertices as $f(\tau)$, results in exactly k vertices being burned in the tree (T, r) constructed above. Conversely, any strategy under which at most k vertices of (T, r) are burned is obtained from some satisfying truth assignment for (B, C).*

3 Cubic Graphs

We now use the results described in the previous section to prove NP-completeness of the following problem:

3-FIRE
Instance: A rooted cubic graph (G, r) and a positive integer k.
Question: If the fire breaks out at r, is there a strategy such that at most k vertices burn?

Theorem 5. *3-FIRE is NP-complete.*

Proof. The transformation is from RESTRICTED NAE 3-SAT. Suppose an instance (B, C) of this problem is given. Let (T, r) and k be the instance of 3-T-FIRE of which the construction is described in the previous section.

We first construct a rooted cubic graph (G, r) from (T, r) by adding edges. Notice that the only vertices of T that do no have degree three are leaves of a copy of F, or belong to one of $S_0, S_1, \ldots, S_b$. In each copy of F, add a cycle of whose vertex set is the set of leaves of that copy of F. It remains to describe the modifications to $S_i, 1 \leq i \leq b$. First, join each vertex of degree two on the (z_i, b_i)-path with the vertex on the $(z_i, \bar{b}_i)$-path which is at the same level. Second, for $j = 1, 2, \ldots, 2^{p-1}$, join each vertex of degree two on the $(b_i, t_{b_{i,j}})$-path to the vertex of degree two on the $(b_i, t_{b_{i,j+2^{p-1}}})$-path which is at the same level in T. Do the same for vertices of degree two on $(\bar{b}_i, t_{\bar{b}_{i,j}})$ paths. Third, replace each snake tree with a snake and each ladder tree with a ladder. (This amounts to adding edges. Also, it is important to observe that each ladder tree used in the construction has at least 12 vertices, so this substitution is possible; ladders must have at least five vertices.) At this point the only vertices which do not have degree three are the ones in snakes at the maximum distance from the root a of the snake. These currently have degree two. Recalling the structure of RESTRICTED NAE 3-SAT, for $i = 1, 2, \ldots b$, and each vertex b_i (resp. $\bar{b}_i$), we can add two edges to form a 4-cycle among the four degree two vertices that are descendants of b_i (resp. $\bar{b}_i$) and which belong to the two snakes arising from the clause pair involving b_i and $\bar{b}_i$. This completes the construction of (G, r).

The integer k in our instance of 3-FIRE is the same k as in our instance of 3-T-FIRE. The transformation can clearly be carried out in polynomial time.

Since (T, r) is a spanning subgraph of (G, r), no strategy can save more vertices of (G, r) than it does on (T, r). Thus, by Theorem 4, it suffices to prove that there is a strategy under which exactly k vertices are burned. It turns out that the same strategy that was used for 3-T-FIRE works.

Suppose there is a satisfying truth assignment, and consider the strategy for (T, r) described in the previous section. We analyze this strategy carefully and argue that no more vertices burn in G than burn in T. The analysis makes substantial use of Theorem 3.

At times one through m the vertex defended is an ancestor of a copy of F. By construction of (G, r), each such vertex is a cut vertex of G, so every descendant of such a vertex that is saved in T is also saved in G.

At times $m+1$ through $m+p$ the vertex defended is a descendant of b_0 which of degree three in T. By construction of (G, r), each such vertex is a cut vertex of G, so every descendant of such a vertex that is saved in T is also saved in G.

For $i = 1, 2, \ldots, b$, at time $m+p+i$ the vertex defended is either b_i or $\bar{b}_i$. Again, each such vertex is a cut vertex of G, so every descendant of such a vertex that is saved in T is also saved in G.

At time $m+p+b+1$ the vertex $\bar{b}_0$ is defended. It is also a cut vertex of G, and so all if its descendants in T are also saved in G.

For $i = 2, 3, \ldots, 3n$, the vertex if T that is defended at time $m+p+b+i$ is chosen according to a greedy strategy. By Theorem 2, each such vertex can be chosen to belong to a snake, and either be a vertex of degree three, or of degree two and a parent of degree three. These are not cut vertices of G because of the 4-cycle created among the degree two vertices belonging to the two snakes arising from each clause pair. Note, however, that there are no other adjacencies between vertices in snakes corresponding to different clauses. By Theorem 2, for each clause pair either all of the vertices in such a 4-cycle burn or none do. If all such vertices burn, then both snake trees burn entirely. If no such vertices burn in T then, by our construction they do not burn in G. This means that every vertex of snake tree that is saved in T is also saved in G. This completes the proof. □

4 Protecting specified sets of vertices

The problems discussed up to this point and in [4] have been concerned with saving as many vertices of the graph as possible. In this section we consider the problem of deciding whether all members of a specified set of vertices can be saved. Formally:

S-FIRE
INSTANCE: A rooted graph (G, r) and a subset $S \subseteq V(G)$.
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 finite sequence $d_1, d_2, \ldots, d_t$ of vertices of G such that, if the fire breaks out at r, then,

(i) vertex d_i is neither burning nor defended at time i,
(ii) at time t no undefended vertex is adjacent to a burning vertex, and
(iii) no vertex in S is burned at the end of time t.

We will show that S-FIRE is NP-complete even when restricted to S being the set of leaves of a full rooted tree of maximum degree three. By contrast, we will also describe a polynomial time algorithm to decide whether it is possible to save a given set S of vertices in a graph with maximum degree three, provided the fire breaks out at a vertex of degree at most two.

We establish the NP-completeness results first. The restricted version of the problem that we consider is stated below.

3-FL-FIRE
INSTANCE: A full rooted tree (T, r) with $\Delta(T) \leq 3$.
QUESTION: When the fire begins at r, is there a strategy such that no leaf burns? That is, does there exist a finite sequence $d_1, d_2, \ldots, d_t$ of vertices of T such that, if the fire breaks out at r, then,
(i) vertex d_i is neither burning nor defended at time i,
(ii) at time t no undefended vertex is adjacent to a burning vertex, and
(iii) no leaf of T is burned at the end of time t.

Our transformation will make use of graphs which we call *forks*, and which are illustrated in Figure 3. Specifically, $\mathcal{F}(n, m)$ denotes the fork with diameter n and distance m from a to b.

Theorem 6. *3-FL-FIRE is NP-complete.*

Proof. The transformation is from RESTRICTED NAE 3-SAT. We will continue the reduction described in Section 2, which is used to prove Theorem 1 in [4]. Consider the instance (T, r, k) of 3T$'$-FIRE generated from an instance (B, C) of RESTRICTED NAE 3-SAT, with the added constraint that for $i = 1, 2, \ldots, b-1$, $b_i > b_i + 1 \geq 1$ (we can add polynomially many clauses to make this true). We will alter T as follows:

- Let $d' = b + p + b2^p + 1 \geq d$.
- Replace every $\mathcal{S}^T(3n+2, i)$ in T with $\mathcal{F}(d' - b - p, i - 1)$ (see Figure 3) for all i.

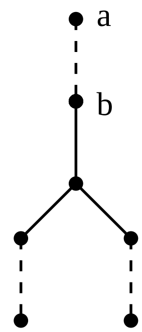

Figure 3: A fork $\mathcal{F}(n,m)$, with diameter n and distance m from a to b.

- Replace every $\mathcal{L}^T(3n+2)$ in T with a path of length $d'-b-p$.
- Replace every $\mathcal{L}^T(3n+1)$ in T with a path of length $d'-b-p-1$.

T is now a full tree with depth d'. Since the tree has been altered, we must redefine the strategy $f(\tau)$ for a given truth assignment τ. The revised strategy, which we call $f'(\tau)$, is the same as $f(\tau)$ down to level $b+p$. On levels $i = b+p+1, \pm p+2, ldots, b+p+3n$, it defends in the copy of $\mathcal{F}$ which replaced the snake tree $f(\tau)$ originally defended at level $i+1$. Below level $b+p+3n$, $f'(\tau)$ greedily defends the maximum number of leaves per turn.

We claim that if a truth assignment τ satisfies (B,C), then $f'(\tau)$ defends every leaf for (T,r). It is easy to see that in $f'(\tau)$, $b2^p+1$ vertices burn at level $b+p$. As in the reduction from Section 2, $f'(\tau)$ saves two leaves per turn by defending at each level i for $i = b+p+1, b+p+2, \ldots, b+p+3n$. This means that at level $b+p+3n$, $b2^p+1-3n$ leaves are unprotected. Since there are $d'-b-p-3n$ levels remaining, it is possible to save every leaf. This proves the claim.

Let l_i be the maximum number of leaves in a subtree rooted at level i. Let l be the number of leaves in T.

Note that a strategy arising as above from a truth assignment saves l_i leaves which are children of the vertex it protects at level i for $i = 1, \ldots, b+p$ (2^p+2n_i for $i = 1, \ldots, b$ and $2^{p+b-i+1}$ for $i = b+1, \ldots, b+p$). If it is the image under f' of a satisfying truth assignment, this is true for $i = 1, 2, \ldots, d'$.

We claim that if (B,C) has no satisfying truth assignment, then no strategy σ for (T,r) saves every leaf. Suppose to the contrary that there is a strategy σ that saves every leaf, but (B,C) has no satisfying truth assignment. We know from above that $l = \sum_{i=1}^{d'} l_i$. Thus σ saves l_i leaves which are descendants of the vertex it protects at level i, for $i = 1, 2, \ldots, d'$. By the construction of T (with the important fact that $n_i > n_i + 1$), σ must be the same as some truth assignment strategy $f'(\tau)$ on levels 1 to b.

Suppose σ saves neither b_0 nor $\bar{b}_0$, nor any descendant of either of these vertices from time $i = b + 1$ to $b + p$. Then, because it saves l_i vertices, it must be saving forks. This means that it is impossible for σ to save two vertices each turn from $b + p + 1$ to $b + p + 3n$, a contradiction. So without loss of generality (because of isomorphism of subtrees rooted at descendents of b_0 or $\bar{b}_0$ on a given level), assume σ is the same as $f'(\tau)$ from levels 1 to $b + p$.

By Theorems **??** and 2 (and as in the proof of Theorem 1 in [4]), it is impossible for σ to save two vertices each turn from $b + p + 1$ to $b + p + 3n$ because τ does not satisfy (B, C). Therefore, σ protects fewer than l_i leaves which are descendants of the vertex it protects at level on some level i, a contradiction. This proves the claim.

We must now deal with the fact that r has degree $2^m + 2$. We can reduce the problem to a full tree with maximum degree 3 as follows: Instead of using the F described before, we will let F be the subtree of T consisting of r and the subtrees rooted at b_1 and $\bar{b}_1$. With this new F, we create (T', r') as before. From a satisfying truth assignment, we can easily generate a strategy for (T', r') which saves every leaf and which saves l'_i leaves on every turn, where l'_i is the maximum number of leaves that are descendants of a vertex of at level i in T'. We can see that for $i = 1, 2, \ldots, m + 1$, $l'_i = 2^{m+1-i} l_1 + 2^{m+1-i} - 1$, and is achieved uniquely by defending the vertex at level i which saves the most copies of F.

Again, suppose we have no satisfying truth assignment for (B, C), but it is possible to save all the leaves in T' with a strategy σ. Then σ defends an ancestor of a copy of F at every turn from 1 to $m + 1$. This is clearly a contradiction to the previous claim.

To complete the proof, we note that the transformation can be accomplished in polynomial time. This follows, in particular, on noting that F can be constructed in polynomial time (and is therefore of polynomial size). $\square$

Corollary 7. *S-FIRE is NP-complete.*

We conclude this section by establishing a result that provides a sharp diving line between between polynomial and NP-complete cases of S-FIRE.

Theorem 8. *S-FIRE is polynomially solvable when restricted to rooted graphs (G, r) with maximum degree three and in which r has degree at most two.*

Proof. Since the problem is trivial when r has degree one, assume that r has degree two. Since the problem is also trivial if $r \in S$, assume $r \notin S$. And, since it suffices to protect vertices in the component of $G - S$ that contains r, we assume for simplicity of notation that $G - S$ is connected.

Suppose that there is a vertex in $G - S$ which is of degree two in G, and is neither in S nor adjacent to a vertex of S. Then there is such a vertex x which is nearest to r, say at distance d. By the choice of x, all vertices on a shortest $r - x$ path in $G - S$ must have degree three in G. Fix such a path $P : r = x_0 x_1 \ldots x_d = x$. For $i = 1, 2, \ldots, d$, let y_i be the neighbour of x_i (in G) that does not belong to P. At times $t = 1, 2, \ldots, d$, defend y_t. All vertices in S are saved.

A similar argument to the above shows that if there is a vertex in $G - S - \{r\}$ of degree one in $G - S$ which is adjacent to at most one vertex of S, then all vertices in S can be saved. Thus any vertex of degree one in $G - S$ has two neighbours in S.

Another similar argument also shots that if there is a cycle in $G-S$, then all vertices in S can be saved (we can use the above technique to direct the fire around the cycle and put it out).

By the above that, we may assume that $G - S$ is a tree. Hence, if it has at least two vertices then it has at least two vertices of degree one. (One of these might be r.) Initially, for $i \geq 0$, let S_i be the set of vertices in S that are at distance i from r. These sets will be updated in the process described below.

Let (*) be the statement:

$$|\bigcup_{i=1}^{n} S_i| \leq n, \quad \text{for } n = 1, 2, \ldots$$

(Note that the number of possible values of n equals the maximum distance in G from r to a vertex in S.)

Note that if (*) holds, then the answer for the given instance of S-FIRE is YES. All vertices in S can be defended (not just saved) because, for any positive integer n, there are never more than n vertices in S within distance n of r.

Suppose that neither the conditions from the argument above hold, nor (*) holds. Then, some vertex in $G - S$ has two neighbours in S. Note that if such a vertex burns, then on the next turn a vertex in S will necessarily burn. Suppose that the nearest such vertex is at distance k from r. Add all vertices at distance k from r and having two neighbours in S to S_k, and remove their neighbours from S_{k+1} provided they have no neighbours in $G-S$ other than the vertices just added to S_k. After these updates have been made, delete vertices that are not in the same connected component as r in $G-S$ (where $S = S_0 \cup S_1 \cup \cdots$).

Repeat the above process until either a YES is obtained because (*) holds or one of the other conditions above holds, or none of these conditions holds no further updates can be made. We claim that if neither (*) nor one of the other conditions above holds when the process terminates, then $S = S_0 = \{r\}$ and the answer is NO. Since no further updates can be made, no vertex in $G-S$ has two neighbours in S. Also, there is no vertex in $G-S$ which is of degree two in G and is neither in S nor adjacent to a vertex of S, nor is there a vertex in $G-S-\{r\}$ of degree one in $G-S$ which is adjacent to at most one vertex of S. Since $G-S$ is a tree, it follows that $G-S$ is empty and $S=\{r\}$.

We now show that if, eventually, $S=\{r\}$, then the answer is NO. The proof is by induction on the number of vertices of G. The statement is easily seen to hold if G has only one vertex. Suppose it holds for any rooted graph satisfying the hypotheses of the theorem and having n or fewer vertices. Consider an $(n+1)$-vertex rooted graph (G,r) satisfying the hypotheses of the theorem, and such that $S=\{r\}$ when the process above terminates. Since, eventually, $S=\{r\}$, there is a first time when one or more vertices are deleted from G. By the induction hypothesis, the answer is NO for this reduced graph and updated set S. By, as we observed before, if a vertex of an updated set S burns, then a vertex in the original set S burns. This completes the proof. $\square$

References

[1] S. Crosby, A. Finbow, B. Hartnell, R. Moussi, K. Patterson and D. Wattar, Designing Fire Resistant Graphs, manuscript 2005.

[2] A.S. Finbow and B.L. Hartnell, On Designing a Network toDefend Against Random Attacks on Radius Two, *Networks*, Vol. 19 (1989), 771-792.

[3] S. Finbow, B. Hartnell, Q. Li, and K. Schmeisser, On minimizing the effects of fire or a virus on a network, *JCMCC* 33 (2000), 311-322.

[4] S. Finbow, A. D. King, G. MacGillivray and R. Rizzi, The firefighter problem for graphs of maximum degree three, *Discrete Mathematics*, to appear.

[5] P. Fogarty, *Catching the fire on grids*, M.Sc. Thesis (2003), Department of Mathematics, University of Vermont, USA.

[6] M. R. Garey and D. S. Johnson, Computer and intractability, A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, New York, 1979.

[7] G. Gunther and B. Hartnell, Graphs with isomorphic survivor graphs, *Congressus Numerantium 79*, 69-77, 1990.

[8] G. Gunther and B. L. Hartnell, On M-connected and k-neighbour-connected graphs, *Graph Theory, Combinatorics, and Applications*, 585-596, 1991.

[9] B. L. Hartnell, Firefighter! An application of domination. Presentation, *24th Manitoba Conference on Combinatorial Mathematics and Computing*, University of Manitoba in Winnipeg, Canada, 1995.

[10] B. L. Hartnell and W. Kocay, On minimal neighbourhood-connected graphs, *Discrete Mathematics* 92, 95-105, 1991.

[11] 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 (Boca Raton, FL, 2000)*, Congr. Numer. 145 (2000), 187-192..

[12] G. MacGillivray and P. Wang, On the firefighter problem, *JCMCC* 47 (2003), 83-96.

[13] Margaret-Ellen Messinger, *Firefighting on Infinite Grids*, M.Sc. Thesis (2004), Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada.

[14] S. A. Moeller and P. Wang, Fire control on graphs, *JCMCC* 41 (2002), 19 -34.