On the Firefighter Problem

Gary MacGillivray*
Department of Mathematics and Statistics
University of Victoria
Victoria, British Columbia, Canada
e-mail: gmacgill@math.uvic.ca

Ping Wang*
Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University, Antigonish, Nova Scotia, Canada
e-mail: pwang@stfx.ca

Abstract

We consider the firefighter problem. We begin by proving that the associated decision problem is NP-complete even when restricted to bipartite graphs. We then investigate algorithms and bounds for trees and square grids.

1 Introduction

We consider a dynamic problem introduced by B. Hartnell in 1995 [5]. Let G be a graph which is rooted at a vertex $r \in V(G)$. At time 0, a fire breaks out at vertex r. At each subsequent time interval, the firefighter *defends* some vertex which is not yet on fire, and then the fire spreads to all undefended neighbours of each *burning* (i.e., on fire) vertex. Once a vertex is defended, it 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.

As described above, the problem involves a single fire and a single firefighter. The general version of the problem involves fires breaking out at each vertex belonging to a set F, and being defended by a set D of firefighters each of whom can, at each time interval, defend a vertex which is not burning (see [3]). We

*Research supported by The Natural Sciences and Engineering Council of Canada.

do not consider the general version of the problem, apart from determining its complexity and the results we survey below.

A number of papers investigating different aspects of the firefighter problem have appeared in the literature. The design of optimal graphs is developed in [3]. Two approaches are considered. One of these is to design graphs such that one can minimize the expected number of vertices burned when the fire breaks out at a random subset of vertices. Stars are proved to be optimal graphs when there is one firefighter, regardless of the number of fires. The second approach is to design graphs where the firefighters can minimize the damage (measured as the number of vertices that eventually burn) given that the fires start at vertices for which the damage is maximized. Related topics are examined in [2, 4, 6, 7]. Other papers consider algorithms, bounds and heuristics [9, 11]. Algorithms for two and three dimensional grid graphs are presented in [11]. When the fire starts at a corner vertex of a grid, the value of the optimum solution is established. Otherwise, the algorithms lead to bounds on the maximum number of vertices that can be saved and, in turn, these lead to some asymptotic results. For an instance (G, r) of the firefighter problem, let $R(G, r)$ be the ratio of the number of vertices that can be saved to the total number of vertices of G. It is proved that $\frac{1}{4} + \varepsilon \leq R(P_n \times P_n, v) \leq 1$, where $\varepsilon = \frac{1}{2n}$ if n is even and $\varepsilon = -\frac{1}{2n^2}$ if n is odd. For the three dimensional grid, it is conjectured that $\lim_{n\to\infty} R(P_n \times P_n \times P_n, r) = 0$ for every vertex r. In [9] it is proved 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. (It need not be the case that the number of vertices burned under a greedy strategy is at most twice the number of vertices burned under an optimum strategy.)

We begin, in Section Two, with some observations and easy cases of the firefighter problem. In Section Three, we state the decision version of the firefighter problem and show that it is NP-complete when the input is restricted to bipartite graphs. This implies NP-completeness of the general version of the problem for bipartite graphs. A variety of results concerning the firefighter problem on trees are proved in Section Four. The section begins with some observations about the location of the vertices that must be defended at time i in an optimal strategy. A linear time algorithm that solves the optimization problem for binary trees, and works for all trees though not in polynomial time, is described next. It is then demonstrated how to formulate the problem as a 0-1 integer program, and shown how the integrality condition can be relaxed if some non-linear constraints are added. The final subsection describes a polynomial time algorithm, obtained via linear programming and a connection to perfect graphs, that works for a subclass of trees. A rooted tree (T, r) is in the subclass if the graph $P(T, r)$, obtained by joining a vertex to all of its descendants and to all vertices at the same level, is perfect. A forbidden substructure characterization of these trees is also given. In the final section of the paper, bounds and some exact values for the maximum number of vertices that can be saved in a square grid are developed. The results in this section improve those in [11].

2 Observations and easy cases

Graph theoretic notation not defined here follows [1].

The firefighter problem can be regarded as a one player colouring game. Initially, some vertex of a graph G is painted red. At each turn, the player (firefighter) selects an unpainted vertex and paints it black. Each unpainted neighbour of a red vertex is then painted red. This process continues until, finally, the game ends when no more vertices can be painted red. The object of the game is to minimize the number of red vertices. In the context of this formulation, a vertex is *burning* if it is painted red, *defended* if it is painted black, and *saved* if it is not painted red when the game ends. It is easy to see that the set of burning vertices induces a connected subgraph and, if there are unpainted vertices (neither burning nor defended) when the process ends, then the set of defended vertices is a vertex cut.

Let (G,r) be an instance of the firefighter (optimization) problem. We use $MVS(G,r)$ to denote the maximum number of vertices of G that can be saved if the fire breaks out at r. An *optimum strategy* is one which results in $MVS(G,r)$ vertices being saved. Notice that in any optimum strategy, all defended vertices are adjacent to a burning vertex when the process ends, otherwise, there is a strategy that saves at least one more vertex. Further, since at least one vertex is defended at each step and at least one new vertex burns in all steps except possibly the last, the game ends in at most $\lfloor \frac{n}{2} \rfloor$ steps.

As a way of becoming familiar with the firefighter problem, the reader is invited to establish the following very easy facts.

- $MVS(K_n,r) = 1$.
- If $2 \le m \le n$, then $MVS(K_{m,n},r) = 2$.
- $MVS(C_n,r) = n-2$.
- $MVS(P_n,r) = n-1$ is r is a leaf, and $n-2$ otherwise.

A *caterpillar* is a tree that has a path containing at least one end of every edge. It is not hard to prove that the following greedy strategy works for caterpillars.

Strategy 2.1. *Let (C,r) be a rooted caterpillar. If r is a leaf, then defend its unique neighbour. Otherwise, at time 1 defend the neighbouring vertex of highest degree. If the process has not terminated, then at time 2 defend a highest degree unprotected vertex adjacent to the fire.*

As a final easy but slightly less trivial example, consider the n-cube Q_n. Recall that $V(Q_n)$ is the set of binary sequences of length n, and two vertices are adjacent if and only if they differ in exactly one element.

Proposition 2.2. $MVS(Q_n) = n$.

Proof. Since Q_n is vertex-transitive, we assume the fire breaks out at vertex $r = (0, 0, \ldots, 0)$. We use mathematical induction to show that at time $t \geq 0$ all vertices with at most t ones are either defended or burned. This implies the proposition since the number of vertices that can be defended in t time intervals is t.

The statement is clearly true when $t = 0$. Assume that it holds for $t = k$, and consider $t = k + 1$.

Each vertex with $k + 1$ ones is adjacent to $k + 1$ vertices with k ones (there are $k + 1$ ones that can be changed to a zero). By the induction hypothesis, at least one of these vertices is burning. Hence any undefended vertex with $k + 1$ ones is adjacent to a burning vertex and will burn at time $k + 1$ if it is not immediately defended. This completes the proof. □

It is interesting to note that only n vertices can be saved although Q_n has a large number of vertices (2^n), each of which has a small degree (n).

3 Complexity

In this section we establish NP-completeness of the firefighter decision problem for bipartite graphs. It is formally stated below:

FIREFIGHTER
INSTANCE: A rooted graph (G, r) and an integer $k \geq 1$.
QUESTION: Is $MVS(G, r) \geq k$? That is, is there 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 least k vertices are saved at the end of time t.

The transformation is from the well-known NP-complete problem EXACT COVER BY 3-SETS (X3C, see [10], page 221). A description of X3C is included below for completeness.

EXACT COVER BY 3-SETS (X3C)
INSTANCE: A set X with $|X| = 3q$ and a collection C of 3-element subsets of X.
QUESTION: Does C contain an *exact cover* for X, i.e., is there a subcollection $C' \subseteq C$ such that each element of X occurs in exactly one member of C'?

Theorem 3.1. *FIREFIGHTER is NP-complete for bipartite graphs.*

Proof. FIREFIGHTER is clearly in NP. The transformation is from X3C. Suppose an instance of X3C, a set X with $|X| = 3q$ and a collection C of 3-subsets of X, is given. We construct a rooted bipartite graph (G, r) and a positive integer k such that at least k vertices of G can be saved if and only if there is an exact cover of X by elements of C.

Initially, G consists of the vertex r and $|C|$ vertices $C_1, C_2, \ldots, C_{|C|}$ vertices corresponding to the elements of C, each joined to r by a path of length q. So far, there are $1 + |C|q$ vertices. To complete the construction of G, for each pair of disjoint sets in C, join the corresponding vertices C_i and C_j by $10q^5$ paths of length two. Let P be the set of vertices added in this step. Finally, let $k = q + \binom{q}{2}(10q^5)$. The construction can be accomplished in polynomial time. It is easy to see that G is bipartite.

Suppose X admits an exact cover by elements of C. Then, at each time from 1 to q, defend a vertex corresponding to a set in the cover. Since each pair of these vertices is joined to $10q^5$ paths of length two, the number of vertices saved is at least $q + \binom{q}{2}(10q^5)$, as required.

Conversely, suppose at least $q + \binom{q}{2}(10q^5)$ vertices can be saved. Notice that all but at most q vertices corresponding to elements in C are burned after q steps. Thus, at most q such vertices can be saved. If fewer than q of these vertices are saved, then the maximum number of vertices in P that can be saved is $\binom{q-1}{2}(10q^5)$, so that even if every other vertex in the graph were saved, the number of vertices saved is at most $\binom{q-1}{2}(10q^5) + q|C| \leq \binom{q-1}{2}(10q^5) + q\binom{3q}{3} < k$. Therefore, q vertices corresponding to elements of C must be saved. If these q vertices do not correspond to a collection of pairwise disjoint 3-subsets then, reasoning as above, the maximum number of vertices saved is at most $\binom{q}{2}(10q^5) - 10q^5 + q\binom{3q}{3} < k$. Thus, the elements of C corresponding to these q vertices form to an exact cover.

□

Theorem 3.1 implies NP-completeness, for bipartite graphs, of the more general version of the firefighter decision problem with a set F of vertices where the fire breaks out and a set D of defenders, as it establishes NP-completeness of the restriction to $|F| = |D| = 1$.

4 Trees

The result in Section 3 motivates investigating the firefighter problem on classes of bipartite graphs. Trees are a natural choice but, even for this restricted class of graphs, the problem seems difficult. We conjecture that FIREFIGHTER is NP-complete for trees.

The main results in this section include a linear time algorithm that computes the optimum solution for binary trees and works for all trees (though not in polynomial time), an integer programming formulation of the problem on trees, a non-linear optimization version, and a polynomial time algorithm that computes the optimum solution for a subclass of trees. The latter algorithm arises from transforming the firefighter problem in certain trees to the maximum weight independent set problem in a related family of perfect graphs and finding a solution using linear programming.

4.1 Tree Preliminaries

In this subsection we state some definitions and establish a few useful facts about the location of the vertices that are protected, at each time interval, in an optimal strategy.

Let (T, r) be a tree rooted at r, and let v be a vertex of T. The *level* of a vertex v in (T, r) is the length of the unique (r, v)-path. A vertex $w \neq v$ of T is a *descendant* of v if the unique (r, w)-path in T contains v. In this case v is called an *ancestor* of w. Note that the definition excludes the possibility that a vertex is a descendant or ancestor of itself. We use $desc(v)$ to denote the number of descendants of the vertex v. We say that v is the *parent* of w, or that w is a *child* of v, if w is both a descendant of v and adjacent to v. The *subtree rooted at* v is the rooted tree (T', v), where T' is the subtree of T consisting of the vertex v and all of its descendants. A *branch* of (T, r) is a subtree rooted at a child of r. The *length* of a branch B of (T, r) is the largest level (in (T, r)) of a leaf of (T, r) belonging to B. A *stem* is a vertex which is adjacent to a leaf.

Observation 4.1. *Suppose the fire breaks out at vertex r of the tree T. In an optimum strategy, the vertex defended at each time is adjacent to a burning vertex.*

Proof. Suppose that S is an optimum strategy in which, at time i, the vertex x defended is not adjacent to a burning vertex. Since in a tree there is a unique path between any two vertices, the strategy S' which is the same as S except that at time i the ancestor of x which is not burning and is adjacent to a burning vertex is defended, saves more vertices than S, a contradiction. □

Corollary 4.2. *Suppose the fire breaks out at vertex r of the tree T. In an optimum strategy, the vertex defended at time i is at level i.*

Corollary 4.3. *Suppose the fire breaks out at vertex r of the tree T. In an optimum strategy, no two vertices at the same level are defended.*

4.2 Binary trees

We now describe a recursive algorithm that computes an optimum solution to the firefighter problem for trees, and works in linear time for binary trees. Suppose the fire breaks out at vertex r of the tree T, and let $x_1, x_2, \ldots, x_k$ be the neighbours of r in T. By Corollary 4.2, we must decide which of $x_1, x_2, \ldots, x_k$ to defend at time one. This will save all vertices in the subtree rooted at x_i. For $i = 1, 2, \ldots, k$, let T_i be the tree obtained from T by deleting r and the subtree containing vertex x_i, and identifying all of the other vertices x_j, $j \neq i$, to obtain a (super-vertex) w_i. For each i between one and k, find $MVS(T_i, w_i)$ recursively. Then, defend a vertex x_i for which $1 + MVS(T_i, w_i) + desc(x_i)$ is largest. The vertices of T_i defended in order to obtain $MVS(T_i, w_i)$ are the remaining vertices of T that should be defended.

It is easy to prove by mathematical induction that the above procedure computes an optimum solution for any tree. It is not, in general, a polynomial-time algorithm. We include here a standard induction argument to show that it requires only a linear number of steps for binary trees: In this case, $k \leq 2$. Let $S(n)$ be the number of steps taken by the algorithm when the input binary tree has n vertices. Then, $S(1)$ is a constant c_1. Let the constant c_2 denote the number of steps necessary to prepare for the recursive calls and choose which vertex to defend at the end of the algorithm. Suppose $S(k) \leq c \cdot k$ for $1 \leq k \leq n-1$, for some constant $c \geq \max\{c_1, c_2\}$. For binary trees, the instructions lead to $S(n) \leq S(n_1) + S(n_2) + c_2$, where n_1 and n_2 are the number of vertices in T_1 and T_2, respectively. Since $n_1 + n_2 = n - 1$, we have by the induction hypothesis that $S(n) \leq cn_1 + cn_2 + c \leq c(n_1 + n_2 + 1) = cn$, as desired.

4.3 Integer Programming

In this subsection we describe a 0-1 integer linear program to find $MVS(T, r)$ for any rooted tree (T, r). For each vertex $v \neq r$ of T, let $w_v = desc(v) + 1$. The integer w_v is the number of vertices that can be saved by defending vertex v. Let x_v be a boolean decision variable such that $x_v = 1$ if and only if vertex v is defended ($x_v = 0$ otherwise). We will want to maximize the objective function $\sum_{v \in V} x_v w_v$, subject to constraints that guarantee that at most one vertex is defended at every level (Corollary 4.3), and that at most most one ancestor of each vertex is defended (so each saved vertex is counted once in the objective function). The latter condition will be satisfied if and only if exactly one ancestor of each leaf is defended. The resulting integer linear program is shown in Figure 1.

We have observed that the LP relaxation of this integer linear program often has integral optima. There are, however, rooted trees for which this is not the case.

It is possible to guarantee an integer optimum solution to the relaxation of

$$\text{Max} \sum_{v \in T-\{r\}} x_v w_v$$

Subject to:

$$\begin{cases} \sum_{level(v)=i} x_v \leq 1 & \text{for each level } i \\ x_v + \sum_{\text{all ancestors of } v} x_u \leq 1 & \text{for every leaf } v \text{ of } T \\ x_v \in \{0,1\} & \end{cases}$$

Figure 1: A 0-1 integer program for the firefighter problem on a tree.

the above integer linear program by adding some non-linear constraints. For each vertex $v \neq r$ and each descendant w of v, add the constraint $x_v x_w = 0$. The constraints corresponding to ancestors can be deleted as these new ones are stronger. We now show that there is an integer valued optimum solution. Let i be the highest level at which some variable x_v is not integral, if such a level exists. Let m be a vertex at this level with $0 < x_m < 1$ and w_m maximum over all vertices v at level i with $x_v > 0$. By the constraints $x_v x_u = 0$ it follows that if $x_v > 0$ then $x_u = 0$ for each vertex $u \neq r$ which is an ancestor or descendant of v. Hence, the value of x_m can be increased by an amount ϵ so long as x_v is decreased by ϵ for some vertex v at level i. Since the objective function is maximized, equality must occur in the constraint corresponding to vertices at the level i. Further, every vertex p at level i with $x_p > 0$ must have $w_p = w_m$, otherwise the solution obtained by setting, for any such x_p, $x_m := x_m + x_p$ and $x_p := 0$ has a greater value of the objective function. But then setting $x_m = 1$ and $x_v = 0$ for all vertices $v \neq m$ at level i gives a solution which is integer valued at levels $1, 2, \ldots, i$ and has the same value of the objective function. Repeating this process if necessary produces an integral optimum solution. In particular, the maximum value of the objective function is always integer valued.

4.4 P-trees and Perfect Graphs

We now describe a subclass of trees for which the firefighter problem can be solved in polynomial time by linear programming via a translation to perfect graphs. Recall that a graph G is called *perfect* if every induced subgraph of G has the property that its chromatic number χ and clique number ω are equal. Many NP-hard graph-theoretic optimization problems can be solved in polynomial time (often by linear programming) when restricted to the class of perfect graphs (see [8]). Maximum weight independent set is such a problem.

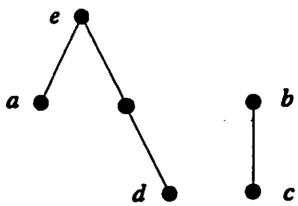

Figure 2: The forbidden configuration.

Let (T, r) be a rooted tree and $P(T, r)$ the graph obtained from T by adding edges joining each vertex to all of its descendants and to all vertices at the same level. If $P(T, r)$ is made into a weighted graph by defining $w(r) = 0$ and, for $v \neq r$, $w(v) = 1 + desc(v)$, then a maximum weight independent set in $P(T, r)$ corresponds to an optimum solution to firefighter in (T, r). The strategy is to defend the vertices in the independent set in order of increasing distance from r.

A rooted tree (T, r) will be called a P-tree if it does not contain the configuration shown in Figure 2. Note that there is no requirement that this be an induced subgraph. An *octopus* is a rooted tree in which all descendants of each stem are leaves. Every octopus is a P-tree. After stating a few properties of P-trees and octopii that follow immediately from the definitions, we prove that (T, r) is a P-tree if and only if the graph $P(T, r)$ defined above is perfect. Thus, if (T, r) is a P-tree, MVS(T, r) can be found in polynomial time by constructing the graph $P(T, r)$, weighting it as described above, and finding a maximum weight independent set in this perfect graph.

Proposition 4.4. *Let (T, r) be a P-tree. Then:*
(i) A vertex $v \neq r$ has at most two children which are not leaves.
(ii) A stem $v \neq r$ of degree greater than 2 has at most one child which is not a leaf.
(iii) All vertices $v \neq r$ with at least two children which are not both leaves occur on the same branch B and at level at least $\ell - 1$, where ℓ is the largest level of a leaf of (T, r) that does not belong to B.

Proof. If any of the statements fails, there is a forbidden configuration. □

The branch in Proposition 4.4 (iii) will be called the P-branch of (T, r). All other branches of (T, r), if any, will be called O-branches.

Theorem 4.5. *A rooted tree (T, r) is a P-tree if and only if $P(T, r)$ is a perfect graph.*

Proof. ($\Leftarrow$) Suppose (T, r) is not a P-tree. Then it contains the configuration in

Figure 4.4, and $P(T,r)$ contains the induced 5-cycle *abcdea*. The corresponding induced subgraph has $\chi > \omega$. Thus, $P(T,r)$ is not a perfect graph.

($\Rightarrow$) Suppose (T,r) is a P-tree. We show that every induced subgraph of $P(T,r)$ has the property that $\chi = \omega$ by induction on $|V(T)|$. This is clearly true if $|V(T)| = 1$. Suppose it is true for all rooted trees with k vertices and let (T,r) be a rooted tree with $k+1$ vertices. There are two cases to consider.

Case 1. There are no O-branches.

In this case, (T,r) has only one branch – a P-branch – and r has a unique child r'.

Let $T' = T - r$. Since (T,r) does not contain the configuration in Figure 4.4, neither does the rooted tree (T',r'). Hence, (T',r') is a P-tree. Since any induced subgraph of $P(T,r)$ that does not contain r is an induced subgraph of $P(T',r')$, the induction hypothesis asserts that any such subgraph has $\chi = \omega$. Thus, it remains to show that $\chi = \omega$ for any induced subgraph of $P(T,r)$ that contains r. Let G be such an induced subgraph. The graph $G-r$ is an induced subgraph of $P(T',r')$, so $\chi(G-r) = \omega(G-r)$. Let c be an $\omega(G-r)$-colouring of $G-r$. Since r belongs to every maximal clique of $P(T,r)$, $\omega(G) = \omega(G-r)+1$. Thus, c can be extended to an $\omega(G)$-colouring of G by introducing a new colour and assigning it to r.

Case 2. There is an O-branch.

Let v be a leaf of maximum level over all leaves belonging to O-branches.

Similarly to Case 1, $\chi = \omega$ for any induced subgraph of $P(T,r)$ that does not contain v, and it must be shown that the same is true of any induced subgraph that does contain v. Let G be such a subgraph.

Since $(T-v,r)$ does not contain the configuration in Figure 4.4, it is a P-tree. Since $G-v$ is an induced subgraph of $P(T-v,r)$, $\chi(G-v) = \omega(G-v)$, by the induction hypothesis. Let c be an $\omega(G-v)$-colouring of $G-v$. If $\omega(G) > \omega(G-v)$, then c can be extended to an $\omega(G)$-colouring of G as in Case 1. Suppose, then, that $\omega(G) = \omega(G-v)$. If some colour is not used on a neighbour of v, then c can be extended to an $\omega(G)$-colouring of G by assigning that colour to v. Hence every colour is assigned to some neighbour of v.

Every vertex adjacent to v in G is either at the same level as v in (T,r), or is an ancestor of v in (T,r). Since $\omega(G) = \omega(G-v)$ some colour α_1 is not used on any neighbour of v in G at the same level as v in (T,r). Thus α_1 must be used on some ancestor x_1 of v in (T,r). Since the vertices of G that are ancestors of v in (T,r) form a clique, x_1 is the only neighbour of v in G coloured α_1. Similarly, some colour $\alpha_2 \neq \alpha_1$ is not used on any neighbour of v in G which is an ancestor of v in (T,r), but is used on some vertex x_2 at the same level as v in (T,r). As above, x_2 is the only neighbour of v in G coloured α_2.

Consider the subgraph H of $G-v$ consisting of the vertices coloured α_1 or α_2.

If x_1 and x_2 do not belong to the same component of H, then switching colours in the component of H containing x_1 results in a colouring of $G - v$ in which α_1 is not used on a neighbour of v. This can be extended to an $\omega(G)$-colouring of G, as above.

Suppose, then, that x_1 and x_2 belong to the same component of H. We will obtain a contradiction. Since H is connected, there is an (x_1, x_2)-path in H, and since the vertices of H are 2-coloured, any such path is of odd length. Let X be a shortest (x_1, x_2)-path in H.

Let (T'', r) be the P-tree obtained from (T, r) by deleting all vertices whose level is greater than that of v. By Proposition 4.4 (iii) and the choice of v, (T'', r) is an octopus and, further, the path X belongs to $P(T'', r)$.

Since (T'', r) is an octopus and X is a path in $P(T'', r)$, the edges of X alternately join two vertices at the same level in (T'', r), and a vertex to an ancestor or descendant in (T'', r). By the argument above, the edge of X incident with the vertex x_1 joins it to an ancestor or descendant in (T'', r), and the edge of X incident with x_2 joins it to a vertex at the same level in (T'', r). Hence, since the edges of X alternate "types", the length of X is even, a contradiction. This completes the proof of Case 2.

Both cases have now been considered, and the result follows by induction. □

5 Square Grids

In this section we consider the firefighter problem in an $n \times n$ grid G_n whose rows and columns are indexed $1, 2, \ldots, n$, with $(1,1)$ being the top left corner. When the fire breaks out in the two outermost rows and columns, we are able to establish the value of the optimum solution and give a strategy for achieving it. In the remaining cases, we give bounds for the maximum number of vertices that can be saved. The results in this section improve those in [11].

A vertex is said to be at *level k* if it is graph distance k from the vertex where the fire breaks out. For $t \geq 1$, we define $d(t)$ to be the number of vertices at level t defended by time t, and $s(t)$ to be the number of vertices at level t which are not burning at time t. We use $p(t)$ to denote the number of vertices at levels greater than t defended by time t. Note that it is possible for a vertex at a level less than t to be defended at time t, if it is not burning. Also, $d(t), s(t)$ and $p(t)$ are non-negative for all $t \geq 0$.

For any t we have $p(t) + \sum_{i=1}^{t} d(i) \leq t$, as the vertices defended at time i need not be on level i. It follows from the definitions that for any $t \geq 1$, $p(t+1) + d(t+1) \leq p(t) + 1$.

Lemma 5.1. *Suppose the fire starts at $f = (r, c)$, where $1 \leq r \leq c \leq \lceil n/2 \rceil$.*

Then, for $1 \le t \le n-r$, $s(t) \le t - p(t)$.

Proof. The statement is clearly true when $t = 1$. Suppose it is true when $t = k$, and consider time $k+1$. Since $k - p(k) \le k$, it follows from the structure of G_n that the union of the neighbourhoods of any $k - p(k) + 1$ vertices at level $k+1$ contains at least $k - p(k) + 1$ vertices at level k. Since $s(k) \le k - p(k)$, one of these vertices at level k is burning. Thus, at most $k - p(k)$ vertices at level $k+1$ have no neighbour burning. Therefore, $s(k+1) \le k - p(k) + d(k+1) = (k+1) - (p(k) + 1 - d(k+1)) \le (k+1) - p(k+1)$. This completes the proof. □

Corollary 5.2. *Suppose the fire starts at* $f = (r, c)$, *where* $1 \le r \le c \le \lceil n/2 \rceil$. *Then, for* $1 \le t \le n-r$, $s(t) \le t$.

Corollary 5.3. *Suppose the fire starts at* $f = (r, c)$, *where* $1 \le r \le c \le \lceil n/2 \rceil$. *Then,*

$$\sum_{t=1}^{n-r} s(t) \le \binom{n-r+1}{2}.$$

Strategy 5.4. *When the fire breaks out at* (r, c), $1 \le r \le c \le \lceil n/2 \rceil$, *defend vertices in the following order:* $(r+1, c), (r+1, c+1), (r+2, c-1), (r+2, c+2), (r+3, c-2), (r+3, c+3), \ldots, (r+c, 1), (r+c, 2c), (r+c, 2c+1), \ldots, (r+c, n)$.

If the fire starts at vertex (r, c), $1 \le r \le c \le \lceil n/2 \rceil$, then Strategy 5.4 results in $2((n-r)+(n-r-1)+\cdots+(n-r-c-1))+(n-2c)(c-1) = n(n-r)-(c-1)(n-c)$ vertices being saved.

Theorem 5.5. *If the fire breaks out at* $f = (1, c)$, *then Strategy 5.4 produces an optimal solution.*

Proof. Without loss of generality, $1 \le r \le c \le \lceil n/2 \rceil$. Since $s(t)$ counts be the number of vertices at level t which are not burning at time t, and some of these may subsequently burn or be defended later, Corollary 5.3 implies that the maximum number of vertices among those at levels less than or equal to $n-1$ that can be saved under any strategy is $\binom{n}{2}$. Since Strategy 5.4 results in this many vertices at at levels less than or equal to $n-1$ and all other vertices being saved, it produces the optimum solution. □

By symmetry Strategy 5.4 produces an optimum solution when the fire breaks out in the last row, or in the first or last column.

Theorem 5.6. *For* $2 \le r \le c \le \lceil n/2 \rceil$,
$MVS(G_n, (r, c)) \le -(r-1) + \binom{n-r+1}{2} + \binom{n-c+1}{2} + \binom{c}{2} + \binom{2r-c-1}{2} + \binom{2r+c-n-2}{2}$,
where $\binom{a}{b} = 0$ *if* $a < b$.

Proof. By Corollary 5.3 no strategy can save more than $\binom{n-r+1}{2}$ vertices at level less than or equal to $n-r$. Since there are at least n vertices at level $n-r$, Lemma 5.1 asserts that under any strategy at least $n-(n-r)+p(n-r)=r+p(n-r)$ vertices at level $n-r$ are burning at the end of time $n-r$. Since there are at least $n-1$ vertices at level $n-r+1$, each of which has two neighbours at level $n-r$, at least $r+p(n-r)$ vertices at level $n-r+1$ are adjacent to a burning vertex at the end of time $n-r$. Since at most $1+p(n-r)$ vertices at level $n-r+1$ are defended by time $n-r+1$, at least $r-1$ vertices at this level must burn. The bound is obtained by assuming that no other vertices of G_n are burned. All such vertices lie in the corners of the grid and at levels at least $n-r+2$. Let S be the set of all vertices of G_n whose level is at least $n-r+2$. The lower left and right corners, respectively, contain $\binom{c}{2}$ and $\binom{n-c-1}{2}$ vertices in S. The upper right corner contains vertices in S only if $(r-1)+(n-c)\geq n-r+2$ (that is, $2r-c-3\geq 0$, or $r\geq(c-3)/2$), and then it contains $\binom{2r-c-1}{2}$ of them. The upper left corner contains vertices of S only if $(r-1)+(c-1)\geq n-r+2$ (that is, $2r+c-(n-4)\geq 0$, or $r\geq(n+4-c)/2$), and then it contains $\binom{2r+c-n-2}{2}$ of them. This completes the proof. □

Corollary 5.7. *If the fire breaks out at $f=(2,c)$, then Strategy 5.4 produces an optimal solution.*

Proof. Without loss of generality $2\leq c\leq\lceil n/2\rceil$. Then $n\geq 3$. Thus, so $\binom{2r-c-1}{2}=\binom{2r+c-n-2}{2}=0$, so the difference between the bound in Theorem 5.6 and the value of the solution obtained under Strategy 5.4 is $r(r-3)/2+1=0$. □

By symmetry, Strategy 5.4 produces an optimum solution when the fire breaks out anywhere in the row two, row $n-1$, column two, or column $n-1$.

We conclude by remarking that the difference between the bound in Theorem 5.6 and the value of the solution obtained under Strategy 5.4 grows as the start of the fire approaches the middle of the grid. If the fire breaks out at $(3,3)$, then the difference is two if $n>5$ and three if $n=5$. In general, if the fire breaks out at $(\lceil n/2\rceil,\lceil n/2\rceil)$, then the difference is $(3/8)n^2-(11/4)n+5$ if n is even, and $(3/8)n^2-(3/2)n+9/8$ if n is odd.

References

[1] J.A Bondy and U.S.R. Murty, *Graph Theory with Applications*, North-Holland, 1976.

[2] A.S. Finbow and B.L. Hartnell, On Designing a Network to Defend 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] G. Gunther and B. L. Hartnell, On M-connected and k-neighbour-connected graphs, *Graph Theory, Combinatorics, and Applications*, 585-596, 1991.

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

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

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

[8] M. Grötschel, L. Lovász,and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, New York, 1988.

[9] B. Hartnell and Q. Li, Firefighting on trees: How bad is the greedy algorithm?, manuscript, 2000.

[10] 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.

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