

SURVIVING RATES OF GRAPHS WITH BOUNDED TREEWIDTH FOR THE FIREFIGHTER PROBLEM*

LEIZHEN CAI[†], YONGXI CHENG[‡], ELAD VERBIN[§], AND YUAN ZHOU[¶]

Abstract. The firefighter problem is the following discrete-time game on a graph. Initially, a fire starts at a vertex of the graph. In each round, a firefighter protects one vertex not yet on fire, and then the fire spreads to all unprotected neighbors of the vertices on fire. The objective of the firefighter is to save as many vertices as possible. The *surviving rate* of a graph is the average percentage of vertices that can be saved when a fire starts randomly at one vertex of the graph, which measures the defense ability of a graph as a whole. In this paper, we study the surviving rates of graphs with bounded treewidth. We prove that the surviving rate of every n -vertex outerplanar graph is at least $1 - \Theta(\frac{\log n}{n})$, which is asymptotically tight. We also prove that if k firefighters are available in each round, then the surviving rate of an n -vertex graph with treewidth at most k is at least $1 - O(\frac{k^2 \log n}{n})$. Furthermore, we show that the greedy strategy of Hartnell and Li [*Congr. Numer.*, 145 (2000), pp. 187–192] for trees saves at least $1 - \Theta(\frac{\log n}{n})$ percent of vertices on average for an n -vertex tree. Our results settle a conjecture and two problems of Cai and Wang [*SIAM J. Discrete Math.*, 23 (2009), pp. 1814–1826] in affirmative.

Key words. firefighter problem, surviving rate, tree, treewidth, outerplanar graph

AMS subject classifications. 05C35, 05C90

DOI. 10.1137/100791130

1. Introduction. The *firefighter problem* is a discrete-time game on graphs introduced by Hartnell [8] at a conference in 1995, who attempted to model firefighting or virus control on a network. The game goes as follows. A fire breaks out at a vertex of a graph $G = (V, E)$, and then the fire and a firefighter make alternate moves on the graph. In each round, the firefighter *protects* at most one vertex not yet on fire, and the fire then spreads from all burning vertices (i.e., vertices on fire) to all their unprotected neighbors. Once a vertex is burning or protected, it remains so during the whole process. The process ends when the fire can no longer spread. All vertices that are not burning are *saved*. The main objective of the firefighter is to save as many vertices as possible.

Various aspects of the firefighter problem have been studied in the literature. Finbow et al. [5] showed that it is NP-hard for the firefighter to save the maximum number of vertices, even for trees of maximum degree three. Hartnell and Li [9] proved that a simple greedy method for trees is a 0.5-approximation algorithm, and MacGillivray and Wang [11] gave a 0-1 integer programming formulation of the prob-

*Received by the editors April 5, 2010; accepted for publication July 29, 2010; published electronically October 12, 2010.

<http://www.siam.org/journals/sidma/24-4/79113.html>

[†]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong SAR, China (lcai@cse.cuhk.edu.hk). This author was partially supported by Earmarked Research Grant 410206 of the Research Grants Council of Hong Kong SAR, China.

[‡]Corresponding author. School of Management, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (chengyx@mail.xjtu.edu.cn).

[§]The Institute for Theoretical Computer Science, Tsinghua University, Beijing 100084, China (elad.verbin@gmail.com). This author was supported in part by the National Natural Science Foundation of China grant 60553001 and the National Basic Research Program of China grants 2007CB807900 and 2007CB807901.

[¶]Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891 (yuanzhou@cmu.edu).

lem for trees and solved the problem in polynomial time for some subclasses of trees. Cai, Verbin, and Yang [1] obtained a $(1 - 1/e)$ -approximation algorithm and several FPT (fixed-parameter tractable) algorithms for the problem on trees. Develin and Hartke [3], Fogarty [7], and Wang and Moeller [15] considered the scenario where more than one firefighter is available and examined the number of firefighters required to contain the fire for d -dimensional grids. Scott, Stege, and Zeh [14] and Ng and Raff [12] investigated the situation where the number of firefighters available varies over time. Finbow et al. [4] looked into the expected number of burned vertices when fires start randomly at fixed-size subsets of vertices, and Cai and Wang [2] studied the average percentage of saved vertices when a fire randomly starts at one vertex of a graph. Other variations of the problem have been discussed in the literature as well, and recently Finbow and MacGillivray [6] have written a survey on the problem.

In this paper, we follow the approach of Cai and Wang [2] and Finbow et al. [4] to consider the defending ability of a graph as a whole for the firefighter problem. For a vertex $v \in G$, let $\xi(v)$ denote the maximum number of vertices the firefighter can save when a fire starts at v . Cai and Wang [2] defined the *surviving rate* of G , denoted by $\rho(G)$, to be the average percentage of vertices that can be saved when the fire starts randomly at one vertex of the graph, i.e.,

$$\rho(G) = \frac{1}{n} \sum_{v \in V} \frac{\xi(v)}{n}.$$

We note that the concept of surviving rates is closely related to the notion of *expected damage* introduced by Finbow et al. [4], who investigated graphs of minimum expected damage. To be precise, $\rho(G) = 1 - \text{ed}(G)/n$, where $\text{ed}(G) = \frac{1}{n} \sum_{v \in V} (n - \xi(v))$ is the expected damage of G .

Cai and Wang [2] showed that $\rho(G) \geq 1 - \sqrt{\frac{2}{n}}$ if G is an n -vertex tree, $\rho(G) > 1/6$ if G is an outerplanar graph, and $\rho(G) > 0.3$ if G is a Halin graph with at least five vertices. They also proposed several problems/conjectures including the following three as identified in [2]:

Problem 6.3 in [2]. For n -vertex outerplanar graphs (Halin graphs, respectively) G , determine whether $\lim_{n \rightarrow \infty} \rho(G) = 1$.

Conjecture 6.4 in [2]. For every n -vertex tree T , $\rho(T) \geq 1 - \Theta(\frac{\log n}{n})$.

Problem 6.6 in [2]. Determine whether the greedy algorithm of Hartnell and Li [9] for trees achieves an approximation ratio $1 - \Theta(\frac{\log n}{n})$ for the surviving rate of n -vertex trees.

In this paper, we consider the surviving rates of trees and outerplanar graphs (which are graphs of treewidth 1 and at most 2, respectively), and we settle the above three problems/conjectures (Problem 6.3 in [2] for outerplanar graphs) in affirmative. Furthermore, we study the surviving rates of graphs of bounded treewidth in general when we allow multiple firefighters to protect vertices in each round. Let $\xi_k(v)$ denote the maximum number of vertices that k firefighters can save when a fire starts at vertex v , and call

$$\rho_k(G) = \frac{1}{n} \sum_{v \in V} \frac{\xi_k(v)}{n}$$

the *surviving rate* of G for k firefighters. We prove that for every graph of treewidth k , $\rho_k(G) = 1 - O(\frac{k^2 \log n}{n})$. Note that there are graphs of treewidth k that require at least k firefighters to save any constant portion of vertices (see section 3).

Our paper is organized as follows (log is of base 2 in the paper). In section 2, we show that the greedy strategy of Hartnell and Li [9] for trees on average saves

at least $1 - \Theta(\frac{\log n}{n})$ percent of vertices, which answers Problem 6.6 in [2] and also settles Conjecture 6.4 in [2] in affirmative. We also construct a tree to show that the bound $1 - \Theta(\frac{\log n}{n})$ is tight, and we give another proof for Conjecture 6.4 in [2]. In section 3, we use the main idea in the second proof for Conjecture 6.4 in [2] to show that $\rho_k(G) = 1 - O(\frac{k^2 \log n}{n})$ for graphs G of treewidth k . In section 4, we prove that $\rho(G) \geq 1 - \Theta(\frac{\log n}{n})$ for n -vertex outerplanar graphs G , which settles Problem 6.3 in [2] for outerplanar graphs.

2. Firefighting on trees. For the firefighter problem on trees, the following greedy method of Hartnell and Li [9] achieves an approximation ratio $1/2$ for the number of saved vertices: the firefighter always protects a vertex that cuts off the maximum number of nonburning vertices from the fire. In this section, we prove that their greedy method on average saves $1 - \Theta(\frac{\log n}{n})$ percent of vertices, which thus settles Problem 6.6 in Cai and Wang [2]. This also confirms Conjecture 6.4 in [2] that the surviving rate of a tree is at least $1 - \Theta(\frac{\log n}{n})$. Furthermore, we construct a class of trees to show that this lower bound is optimal, and we give another proof for the surviving rates of trees. The main idea of the second proof will be used in the next section to deal with graphs of bounded treewidth.

2.1. Hartnell and Li's method (see [9]). Let T be a tree. The greedy method of Hartnell and Li [9] produces a strategy for the firefighter, which will be called an *HL-strategy* for T . Note that an HL-strategy for T is not unique since, in each round, there may be more than one vertex that the firefighter can choose to protect. A vertex u is a *fire source* for a vertex v if, when the fire starts at u , the greedy method of Hartnell and Li cannot always save v , i.e., there is an HL-strategy that will not save v .

THEOREM 2.1. *For a tree T , the greedy method of Hartnell and Li saves on average at least $1 - \Theta(\frac{\log n}{n})$ percent of vertices when the fire starts randomly at one vertex of T .*

Proof. To prove the theorem, we need prove only that no vertex $v \in T$ has more than $3 + 2 \log_2 n$ fire sources. Let r be a fire source for v that is farthest away from v in T and $P = v_0, v_1, \dots, v_k$ the (r, v) -path in T , where $v_0 = r$ and $v_k = v$. Regard T as a rooted tree with root r , and denote the subtree rooted at vertex x by $T(x)$.

We first show that all fire sources for v are on the (r, v) -path P . Suppose that there is a fire source $u \notin P$ for v . Then u is in $T(v_1)$ as any vertex not in $T(v_1)$ is farther away from v than r . Since r is a fire source for v , some HL-strategy will not protect v_1 when a fire starts at r . Therefore $|T(v_1)| < n/2$, and hence $|T - T(v_1)| > n/2$, which implies that $|T - T(u)| > n/2$ as $T - T(u)$ contains $T - T(v_1)$. This indicates that, when a fire starts at u , any HL-strategy would have saved the parent of u and hence v , a contradiction to u being a fire source for v (see Figure 1(a)).

Next we show that, if v_i is a fire source for v , then $|T(v_i)| > 2|T(v_{i+2})|$, where $1 \leq i \leq k - 2$. Consider the situation when the fire starts at vertex v_i . Since $|T - T(v_i)| > n/2$ (note that $T - T(v_i)$ contains $T - T(v_1)$), any HL-strategy will protect v_{i-1} at time 1. By the assumption that v_i is a fire source for v , we see that there is an HL-strategy that does not protect v_{i+2} but another grandchild v_i'' of v_i at time 2. This implies that $|T(v_i'')| \geq |T(v_{i+2})|$ and hence that $|T(v_i)| > 2|T(v_{i+2})|$ (see Figure 1(b)).

Now let $v_{s(0)}, v_{s(1)}, \dots, v_{s(t)} \in P$ be fire sources of v ordered from r to v . Then $|T(v_{s(i)})| > 2|T(v_{s(i+2)})|$ as $T(v_{s(i+2)})$ is a subtree of $T(v_{s(i+2)})$. Therefore

$$|T(v_{s(1)})| > 2|T(v_{s(3)})| > \dots > 2^i |T(v_{s(2i+1)})| > \dots,$$

which implies $t \leq 2 + 2 \log_2 n$ as $|T(v_{s(1)})| < n$ and hence the lemma. \square

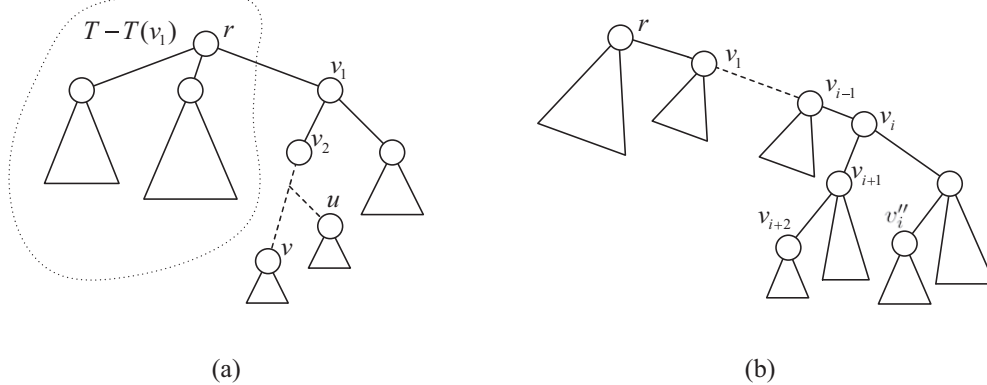


FIG. 1. (a) All fire sources of v are on the (r, v) -path; (b) $|T(v_i)| \geq 2|T(v_{i+2})|$.

The above theorem answers Problem 6.6 in Cai and Wang [2] in affirmative that the greedy method of Hartnell and Li [9] achieves an approximation ratio $1 - \Theta(\frac{\log n}{n})$ for the surviving rate of a tree and also settles Conjecture 6.4 in Cai and Wang [2] regarding the surviving rates of trees.

COROLLARY 2.2. *The surviving rate of every tree is at least $1 - \Theta(\frac{\log n}{n})$.*

The above lower bound is the asymptotically best possible, which is established by the following theorem.

THEOREM 2.3. *Let T_h be a balanced complete ternary tree (i.e., each nonleaf vertex has three children) of height h and with n vertices. Then $\rho(T_h) \leq 1 - \Theta(\frac{\log n}{n})$.*

Proof. We will prove the following: If the fire starts at a vertex v of height k ($0 \leq k \leq h$), let T_k denote the subtree with v as its root, and then the number of burnt leaves of T_k in the end is at least $\frac{1}{2}(3^k + 1)$ under any protecting strategy. Since n , the number of vertices of T_h , is $\sum_{i=0}^h 3^i = \frac{3^{h+1}-1}{2} = \Theta(3^h)$, this implies that, no matter what protecting strategy is adopted, when the fire starts randomly at one vertex of T_h , the minimum average percentage of vertices that will get burnt in the end is at least

$$\frac{\sum_{k=0}^h \frac{1}{2}(3^k + 1) \times 3^{h-k}}{n^2} \geq \frac{\sum_{k=0}^h \frac{3^h}{2}}{n^2} = \frac{(h+1)3^h}{2n^2} = \Theta\left(\frac{h}{n}\right) = \Theta\left(\frac{\log n}{n}\right)$$

which implies the theorem.

In what follows we consider the subtree T_k with root v and assume that the fire starts at v . Then, within T_k , the fire stops to propagate at time k . Thus, we can assume without loss of generality that the number of protected vertices in T_k is at most k . Furthermore, at time i ($0 \leq i \leq k$), the fire stops to propagate among the vertices in T_k having distance at most i from the root v ; therefore, we can assume that there are at most i protected vertices in T_k which are within distance i from v . Let a_j denote the number of protected vertices in T_k that have distance j from v , $j = 0, 1, \dots, k$. Then $\sum_{j=0}^i a_j \leq i$ for $i = 0, 1, \dots, k$.

If the fire starts at v , for each leaf u of T_k which is saved in the end, there must exist an ancestor w of u such that w is in T_k and w is protected at some time t , where $1 \leq t \leq k$. Therefore, the total number of leaves of T_k that are saved in the end cannot exceed $\sum_{j=0}^k a_j \times 3^{k-j}$. Define $b_i = \sum_{j=0}^i a_j$, then $b_i \leq i$ for $i = 0, 1, \dots, k$,

and thus

$$\begin{aligned} \sum_{j=0}^k a_j \times 3^{k-j} &= \sum_{j=1}^k (b_j - b_{j-1}) \times 3^{k-j} = b_k + \sum_{j=1}^{k-1} b_j \times (3^{k-j} - 3^{k-j-1}) \\ &\leq k + \sum_{j=1}^{k-1} j \times (3^{k-j} - 3^{k-j-1}) = \frac{3^k - 1}{2}. \end{aligned}$$

Therefore, the total number of burnt leaves of T_k is at least $3^k - \frac{3^k - 1}{2} = \frac{3^k + 1}{2}$, which completes the proof of the theorem. \square

2.2. A different proof. We will give a protecting strategy different from Hartnell and Li's strategy [9], to give a different proof for the surviving rate of trees. This proof will also serve as a basis for coping with graphs of bounded treewidth in section 3.

The idea is that if T is balanced, the strategy that protects only the parent of the vertex where the fire starts is good enough. For more general cases where the trees may not be balanced, we will "reduce" such cases to the case of balanced trees by identifying a "large" subtree to protect in the second round. Before describing our strategy in detail, we need to define a few terms. In a rooted tree, a vertex v (other than the root) is a *heavy vertex* if the subtree rooted at v has more than half vertices of the subtree rooted at the parent vertex of v ; otherwise, v is a *light vertex*. Note that, for any vertex, at most one of its children is heavy. If such a child exists, we call it the *heavy child*. Clearly, in any path from the root to a leaf, there are at most $\log_2 n$ light vertices.

THEOREM 2.4. *For every n -vertex tree T , there exists a strategy where the firefighter protects vertices in at most two rounds, such that the average number of burnt vertices is at most $2 \log_2 n + 2$.*

Proof. Our strategy is as follows. We first root the tree at an arbitrary vertex r . Suppose the fire breaks out at v . In the first round, protect the parent vertex of v . (If v is the root, then do not protect any vertex in the first round.) In the second round, protect the heavy grandchild (that is, the heavy child of the heavy child) of v . Note that, if such a heavy grandchild exists, it is also unique. If such a heavy grandchild does not exist (that is, v does not have a heavy child, or the heavy child of v does not have a heavy child), then do not protect any vertex in the second round. For the remaining rounds, do not protect any vertex.

It is easy to see that the above strategy is valid, that is, we do not protect any vertex that is already burnt. A vertex v is called a *fire source* of a vertex u under the above protecting strategy if, when the fire starts at v and the above strategy is adopted, u will eventually catch fire. We will show that, for every vertex u , the number of fire sources of u is at most $2 \log_2 n + 2$. Clearly, this establishes Theorem 2.4.

From the description of the above strategy, it is not hard to verify that if v is a fire source of u , then the following hold:

1. v is an ancestor of u in the rooted tree T (with root r);
2. Let v, v_1, v_2, \dots, u be the path from v to u . Then either v is the parent of u or at least one from $\{v_1, v_2\}$ is a light vertex.

Therefore, for any vertex $u \in T$, all the possible fire sources of u are on the unique tree path from r to u and fall into one of the following three categories:

1. the vertices whose child on this path is a light vertex,
2. the vertices whose grandchild on this path is a light vertex,
3. the vertex which is the parent of u or is u itself.

This implies that the number of fire sources of u is at most twice the number of light vertices on the path from r to u , plus two, which is no more than $2 \log_2 n + 2$. This concludes Theorem 2.4. \square

3. Firefighting on graphs with bounded treewidth. We now consider the firefighter problem for graphs of bounded treewidth by allowing more than one firefighter in each round. We will show that k firefighters per round are necessary and sufficient for graphs with treewidth k to save a constant portion of vertices. For the formal definition of treewidth, please see section 3.1.

THEOREM 3.1. *For any n -vertex graph G of treewidth k , there exists a strategy that places k firefighters at each round so that the average number of burnt vertices is $O(k^2 \log n)$, i.e., $\rho_k(G) = 1 - O(\frac{k^2 \log n}{n})$.*

First we note that, for graphs with treewidth k in general, k firefighters are indeed necessary to save a constant proportion of vertices. Consider $K_{k,n-k}$, the complete bipartite graph with k vertices on one side and $n - k$ vertices on the other. It is easy to see that this graph has treewidth k . Also, if we are given only $k - 1$ firefighters at each round, then at most $2k - 2$ vertices can be saved in the end, no matter at which vertex the fire starts. Before proving our main theorem, we first give a brief introduction to the notions of tree decomposition and treewidth.

3.1. Tree decomposition and treewidth. The notions of *tree decomposition* and *treewidth* play important roles in graph theory and graph algorithms. They were originally introduced by Robertson and Seymour [13] in their graph minor theory and have received great attention since then.

Intuitively, a tree decomposition represents the vertices of a given original graph as a connected portion (in fact, an induced subtree) of a tree, and there is an edge between two vertices in the original graph only if the two corresponding portions of the tree intersect.

DEFINITION 3.2 (tree decomposition). *Given an undirected graph $G = (V, E)$, a tree decomposition is a pair (\mathcal{X}, T) , where $\mathcal{X} = \{X_1, X_2, \dots, X_m\}$ is a family of subsets of V with $m = O(|V|)^1$ and T is a tree whose nodes are the subsets X_i . In addition, the following three properties hold:*

1. *The union of all subsets X_i equals V .*
2. *For each edge (u, v) in graph G , there is a subset X_i containing both u and v . That is, vertices are adjacent in G only if the corresponding induced subtrees intersect.*
3. *If X_i and X_j both contain a vertex $v \in G$, then all nodes X_k of the tree T on the unique path between X_i and X_j contain v as well. That is, for any $v \in G$, all the tree nodes containing v form a connected subset in T .*

For instance, one trivial tree decomposition of G is with $\mathcal{X} = \{V\}$, that is, tree T has only one node which is the vertex set V of G . The *width* of a tree decomposition (\mathcal{X}, T) is defined as $\max_{X_i \in \mathcal{X}} \{|X_i| - 1\}$. The *treewidth* of a graph G is the minimum width among all possible tree decompositions of G .

¹In the general definition of tree decomposition, there is no requirement of $m = O(|V|)$. In our paper this requirement is necessary for later analysis. In fact, for any tree decomposition T , it can be transformed into a *nonredundant tree decomposition* T' without increasing the width. Any nonredundant tree decomposition has the extra property that, for any two tree nodes connected by an edge, no one is a subset of the other. Moreover, the number of tree nodes of any nonredundant tree decomposition is no more than the number of vertices of the original graph G . For proofs and more details, the reader is referred to Chapter 10 of [10].

It is easy to see that the tree decomposition of a graph is not unique. The treewidth of every tree is one. In what follows, we always use *tree node* (or *node*, for short) to refer to a node X_i in a tree decomposition which is a subset of vertices of the original graph G , and we use *vertex* to refer to a vertex of the original graph G . Also, we simply use T to denote the tree decomposition of a graph G when there is no confusion. Next we describe the protection strategy, and then prove that, by using this strategy, the average number of burnt vertices is limited to $O(k^2 \log n)$.

3.2. The strategy of placing firefighters. Our strategy is similar in spirit to the strategy we developed for trees in section 2.2 and consists of two rounds of protection. We use T , the tree decomposition of G , to design our strategy. First we root T at an arbitrary node $X_r \in \mathcal{X}$.

We need to define a few more terms and notations before we can describe our strategy. Similarly, for each node $X_i \neq X_r$ in T , X_i is a *heavy node* if the subtree rooted at X_i has more than half of the nodes of the subtree rooted at the parent node of X_i ; otherwise, X_i is a *light node*. Clearly every node has at most one heavy node as its child. A path $\mathcal{P}(X_0) = X_0, X_1, \dots, X_t$ in T is a *heavy path* if each X_i , $1 \leq i \leq t$, is a heavy child node of X_{i-1} (node X_0 can be either a heavy or a light node). For any two nodes X and Y in T , let $\mathcal{T}(X, Y)$ denote the unique path in T connecting X and Y .

For a vertex v in G , define T_v to be the *induced subtree* of T consisting of the nodes containing v . (By the definition of treewidth, these nodes form a connected portion of T , thus forming an induced subtree). Let $r(T_v)$ denote the root of T_v , which is the unique highest node of T_v in the rooted tree T . For any node X in T , we use $st(X)$ to denote the *subtree* of T with root X . Notice that $st(r(T_v))$, the subtree of T with root $r(T_v)$, is not necessarily the induced subtree T_v .

Now we are ready to describe our strategy. Suppose the fire starts at vertex v in G . In the first round, we use up to k firefighters to protect the vertices in $P_1(v) = r(T_v) \setminus \{v\}$. We will prove later that this step saves all the vertices appearing in a node outside $st(r(T_v))$, and so, for the next round, we have to consider only vertices of G that are within $st(r(T_v))$.

To describe which vertices will be protected in the second round, we need a few more notations. Let $I(v) = \{u \mid T_v \cap T_u \neq \emptyset\}$ be the set of vertices that coappear with v in some node of T . Note that $v \in I(v)$. By the definition of treewidth, any neighbor of v in G is in $I(v)$. Consider $\mathcal{P}(r(T_v))$. The heavy path in T starts at $r(T_v)$. (Recall that a heavy path always goes downward in T .) We walk along the heavy path until we come to a node $l(v)$, which is the lowest node in the heavy path that still contains at least one vertex from $I(v) \setminus P_1(v)$. In other words,

$$l(v) = \arg \max_X \{depth(X) \mid X \in \mathcal{P}(r(T_v)), \text{ and } X \cap (I(v) \setminus P_1(v)) \neq \emptyset\},$$

where $depth(X)$ is the distance from X to the root node X_r . Notice that, by definition, $l(v)$ can be any (including the first and the last) node of $\mathcal{P}(r(T_v))$. In the second round, we protect the vertices in $P_2(v) = l(v) \setminus I(v)$. Note that $l(v)$ shares at least one vertex with $I(v)$; thus, $P_2(v)$ contains at most k vertices, and so k firefighters suffice. For the remaining rounds we do not protect any vertex.

It is easy to see that, in the above strategy, we never protect a vertex that has been burnt since, when the fire starts at vertex v , after the first round all the vertices that might be on fire are in $I(v)$, and no vertex of $I(v)$ is in $P_2(v)$.

3.3. The average number of burnt vertices. We prove that, if the above strategy is adopted, the number of fire sources for every vertex is at most $O(k^2 \log n)$. Define $l'(v)$ to be the child of $l(v)$ on the heavy path $\mathcal{P}(r(T_v))$. If $l(v)$ is the last node

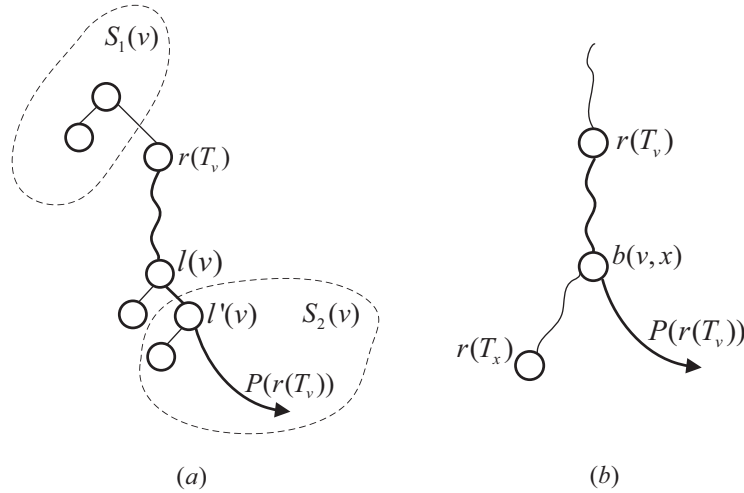


FIG. 2. (a) $S_1(v)$ and $S_2(v)$ are saved. (b) Branching node of v and x , $b(v, x)$.

of the heavy path $\mathcal{P}(r(T_v))$, then $l'(v)$ does not exist. Note that, by definition, $v \notin l'(v)$ since $v \in I(v) \setminus P_1(v)$.

We show that the first round protection limits the fire to the subtree $st(r(T_v))$, and the second round protection keeps the fire away from a “heavy” subtree, $st(l'(v))$. More accurately, let $S_1(v)$ be the set of vertices that do not appear in any node in $st(r(T_v))$, and let $S_2(v)$ be the set of vertices that appear only in nodes of $st(l'(v))$. (If $l'(v)$ does not exist, then $S_2(v)$ is empty.) Then we have the following lemma (please see Figure 2(a)).

LEMMA 3.3. *Protecting $P_1(v)$ in the first round saves all vertices in $S_1(v)$, and protecting $P_2(v)$ in the second round saves all vertices in $S_2(v)$.*

Proof. For the first part of the lemma, let $u \in S_1(v)$. Suppose u catches fire in the end. Then there exists a path $P = (v_0 = v, v_1, \dots, v_q = u)$ from v to u in G with all vertices catching fire in the end. Consider the edge in T connecting $r(T_v)$ and its parent node. The removal of this edge breaks T into two parts: (a) $st(r(T_v))$, the subtree of T with root $r(T_v)$, and (b) the rest. Note that T_v is within part (a) and that T_u is within part (b). Consider the sequence of induced subtrees $T_{v_0}, T_{v_1}, \dots, T_{v_q}$. Each of them is a connected portion of T , and any two consecutive terms in the sequence share at least one node in T . Therefore, there exists some T_{v_i} , $1 \leq i < q$ such that T_{v_i} contains nodes from both parts (a) and (b). Thus T_{v_i} contains node $r(T_v)$ and $v_i \neq v$, and so $v_i \in r(T_v) \setminus \{v\} = P_1(v)$, which implies that v_i is protected in the first round. However, this contradicts the assumption that v_i catches fire in the end.

The proof for the second part of the lemma is similar. Let $w \in S_2(v)$. Suppose w catches fire in the end. Then there exists a path $P' = (v'_0 = v, v'_1, \dots, v'_q = w)$ from v to w in G such that all vertices of P' catch fire in the end. Consider the edge in T connecting $l(v)$ and $l'(v)$. The removal of this edge breaks T into two parts: (a) $st(l'(v))$, the subtree of T with root $l'(v)$, and (b) the rest. By the definition of $S_2(v)$, T_w is within part (a). Also, T_v is within part (b) since otherwise T_v must contain $l'(v)$ (note T_v is connected) which contradicts that $v \notin l'(v)$. Consider the sequence $T_{v'_0}, T_{v'_1}, \dots, T_{v'_q}$. By similar arguments there must exist some $T_{v'_i}$, $1 \leq i < q$, such that $T_{v'_i}$ contains nodes from both parts (a) and (b). Thus $T_{v'_i}$ contains nodes

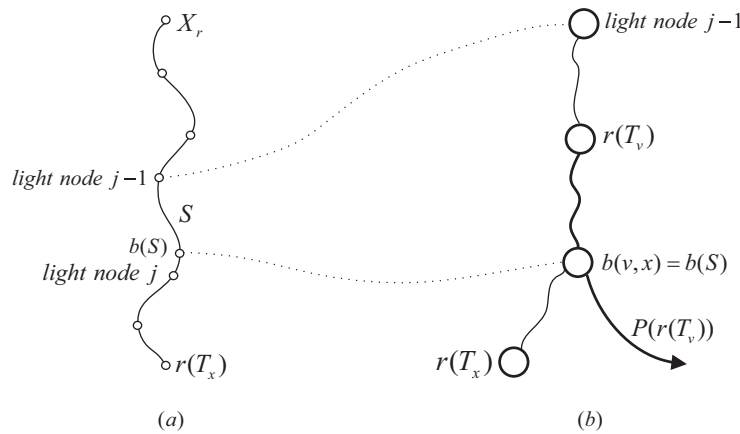


FIG. 3. (a) Decomposition of $\mathcal{T}(X_r, r(T_x))$ into $O(\log n)$ sections. (b) $b(v, x) = b(S)$ for all the vertices v such that $r(T_v) \in S$.

$l(v)$ and $l'(v)$. It follows that $v'_i \in l(v)$ and $v'_i \in l'(v)$. By the assumption that v'_i will catch fire in the end, thus $v'_i \in I(v)$ since otherwise $v'_i \in P_2(v) = l(v) \setminus I(v)$ will be protected. Also, that v'_i will catch fire implies that $v'_i \notin P_1(v)$, and thus $v'_i \in I(v) \setminus P_1(v)$. However, this contradicts $v'_i \in l'(v)$ since, by definition, $l'(v)$ cannot contain any vertex from $I(v) \setminus P_1(v)$. \square

For any $v \in G$ and for any vertex $x \in G$ such that $r(T_x)$ is in the subtree $st(r(T_v))$, define $b(v, x)$, the *branching node* of v and x , as the lowest common node of $\mathcal{P}(r(T_v))$ and $\mathcal{T}(r(T_v), r(T_x))$. Please see Figure 2(b).

LEMMA 3.4. *If v is a fire source of x , then the following hold: (1) $r(T_x)$ is in the subtree $st(r(T_v))$; and (2) there exists some vertex $u \in b(v, x)$ such that $v \in r(T_u)$.*

Proof. The first statement is easy to obtain from the first part of Lemma 3.3. For the second statement, notice that $b(v, x)$ is not lower than $l(v)$ in the heavy path $\mathcal{P}(r(T_v))$. Since otherwise $x \in S_2(v)$, and by the second part of Lemma 3.3 when the fire starts at v , x will be saved in the end which contradicts v being a fire source of x . Also, by the definition of $l(v)$, each node on the segment in $\mathcal{P}(r(T_v))$ from $r(T_v)$ to $l(v)$ (inclusively) contains at least one vertex in $I(v) \setminus P_1(v)$. Actually, assume that $w \in I(v) \setminus P_1(v)$ is in $l(v)$. Then T_w and T_v share at least one node, and it is not hard to see that any node on the path from $r(T_v)$ to $l(v)$ contains at least one vertex from $\{w, v\}$, which is a subset of $I(v) \setminus P_1(v)$. Therefore, $b(v, x)$ also contains at least one vertex in $I(v) \setminus P_1(v)$. Let u denote this vertex. Then $v \in r(T_u)$ since T_u and T_v share at least one node and $r(T_u)$ cannot be an ancestor of $r(T_v)$; otherwise, $u \in P_1(v)$. \square

LEMMA 3.5. *For each vertex $x \in G$, let S^x denote the set of all fire sources of x . Then $|S^x| = O(k^2 \log n)$.*

Proof. Recall that X_r is the root of T , the tree decomposition of G . By the first statement of Lemma 3.4, $S^x \subseteq \{v \mid r(T_v) \in \mathcal{T}(X_r, r(T_x))\}$, where $\mathcal{T}(X_r, r(T_x))$ is the path connecting X_r and $r(T_x)$ in T . Thus, we need to examine only the vertices v such that $r(T_v)$ is in $\mathcal{T}(X_r, r(T_x))$.

Clearly there are at most $\lceil \log_2 m \rceil = O(\log n)$ light nodes in $\mathcal{T}(X_r, r(T_x))$. (Note that $m = O(n)$.) The deletion of these light nodes divides $\mathcal{T}(X_r, r(T_x))$ into $O(\log n)$ sections. We add each light node to the section that immediately follows it. For each section S , let $b(S)$ denote the lowest tree node of S , and let $v(S)$ denote the set of vertices $\{v \mid r(T_v) \in S\}$ (please see Figure 3(a)).

By definition, any heavy path in T starting at a node in S will reach node $b(S)$ but cannot go any farther along $\mathcal{T}(X_r, r(T_x))$, that is, it will branch away from $\mathcal{T}(X_r, r(T_x))$ at $b(S)$. Thus, $b(S)$ is the branching node for all the vertices in $v(S)$ and x . Note that this is still true for the case where S is the last section of $\mathcal{T}(X_r, r(T_x))$. In other words, for any $v \in v(S)$, we have $b(v, x) = b(S)$ (please see Figure 3(b)).

Now we count the number of fire sources of x that are in $v(S)$. By the second statement of Lemma 3.4, for each $v \in v(S)$ that is a fire source of x , there exists at least one vertex $u \in b(v, x) = b(S)$ such that $v \in r(T_u)$. Since the tree node $b(S)$ contains at most $k + 1$ vertices of G , there are at most $k + 1$ such u 's. Moreover, for any such u , there are at most $k + 1$ vertices v such that $v \in r(T_u)$. Thus, there are at most $(k + 1)^2$ vertices in $v(S)$ that are fire sources of x . Since S^x , the set of all the fire sources of x , is the union of all the subsets of fire sources of x belonging to $v(S)$ for the at most $O(\log n)$ sections S of path $\mathcal{T}(X_r, r(T_x))$. Therefore, $|S^x| = O(k^2 \log n)$. \square

By the above lemma, our strategy ensures that the average number of burnt vertices of any graph G with treewidth k is $O(k^2 \log n)$. This concludes Theorem 3.1. We remark that the bound $O(k^2 \log n)$ seems not to be optimal, and it may be possible to reduce it to $O(k \log n)$.

4. Firefighting on outerplanar graphs. A graph is an *outerplanar graph* if it has a planar embedding with all vertices on the boundary of the outer face. Since outerplanar graphs have treewidth at most 2, the result from the last section immediately implies that two firefighters can save all but $O(\log n)$ vertices on average. However, in this section we will show that one firefighter is enough to achieve this, i.e., the surviving rate of outerplanar graphs is also at least $1 - \Theta(\frac{\log n}{n})$, which settles Problem 6.3 of Cai and Wang [2] for outerplanar graphs. Notice that this lower bound is asymptotically tight as it is asymptotically tight for trees (Theorem 2.3), and outerplanar graphs form a superset of trees.

To establish a lower bound for the surviving rate of outerplanar graphs, we need to consider only *maximal outerplanar graphs*, i.e., outerplanar graphs where the addition of any edge will destroy the outerplanarity. Let $G = (V, E)$ be a *maximal outerplanar graph*, i.e., a planar embedding of a maximal outerplanar graph with all vertices on the boundary of the exterior face. We will establish our result for G by considering the *dual graph* $G^* = (V^*, E^*)$ of G constructed as follows: place a vertex inside each face of G , and, if two faces have an edge e in common, join their corresponding vertices by an edge e' crossing only e .

The firefighting problem on vertices of G can be transformed into that on faces of the dual graph G^* : A fire starts at a face of G^* and spreads from a burning face f to each unprotected face sharing a common edge with f in one unit of time. In each unit of time, a firefighter can protect one face not yet on fire. See Figure 4 for an example.

Let x denote the vertex in G^* corresponding to the exterior face of G . It is well known that $G^* - x$ is a tree of maximum degree 3 as every face of G (except the exterior face) is a triangle. We turn $G^* - x$ into a rooted binary tree $T = (V_T, E_T)$ by picking up a leaf r as the root (see Figure 4(b) for an example). Note that each edge in $G^* - T$ connects a vertex v of T with vertex x , and, for convenience, we also regard vertex x as a child of v . For each vertex $v \in V_T \setminus \{r\}$, all vertices on the (r, v) -path are *ancestors* of v . For each vertex $v \in V_T \setminus \{r\}$, we use $p(v)$ to denote its parent in T , T_v the subtree rooted at v , and $depth(v)$ the depth of v which is the distance from the root to v . We also designate left and right children of each vertex $v \in V_T \setminus \{r\}$ in

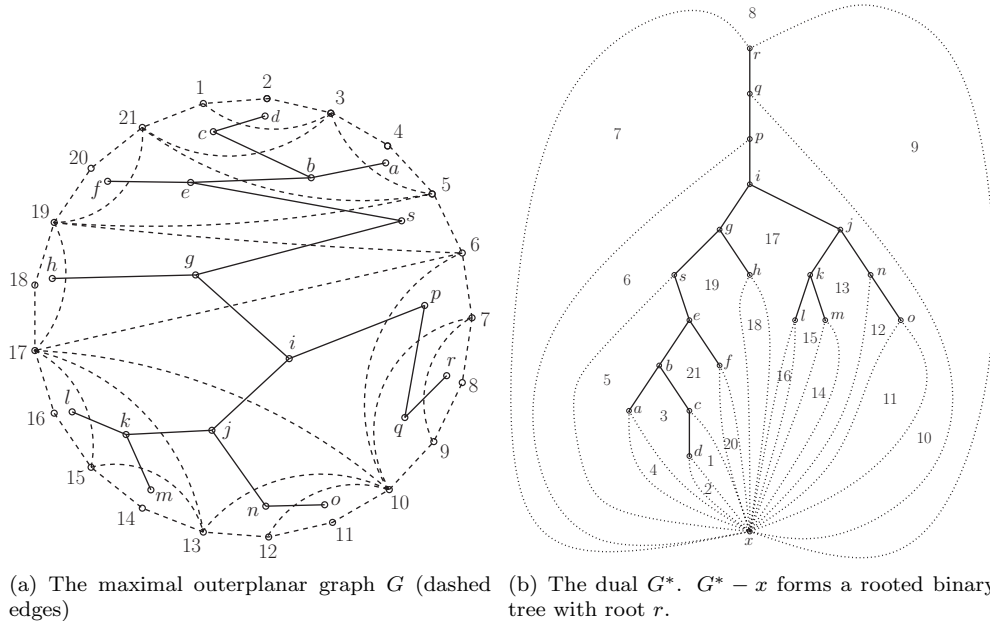


FIG. 4. A maximal outerplanar graph G and its dual G^* . For example, a fire starting at vertex 17 in G is equivalent to a fire starting at face 17 in G^* . After one time unit the fire will spread to the face neighbors 16, 15, 13, 10, 6, 19, and 18 of face 17 in G^* , and the firefighter can protect one of these neighbors (or any face other than 17).

a natural way: We start with edge $vp(v)$ and turn around v clockwise in a very small circle; the first edge we cross connects v with its *right child*, and the other child of v is its *left child*.²

We will use T to design our strategy for the dual graph G^* , and we will define a few more terms before we can describe our strategy. A vertex $v \in V_T \setminus \{r\}$ is a *heavy vertex* if T_v contains more than half of the vertices of $T_{p(v)}$ and is a *light vertex* otherwise. Clearly each vertex has at most one heavy vertex as its child, and every path from the root has at most $\lceil \log n \rceil$ light vertices. A path $P(v_0) = v_0, v_1, \dots, v_t$ is a *heavy path* if each $v_i, 1 \leq i \leq t$, is a heavy vertex (vertex v_0 can be either a heavy or a light vertex), and a vertex $v_i, 1 \leq i \leq t - 1$, is a *turning vertex* if v_i is the left (right, respectively) child of v_{i-1} and v_{i+1} is the right (left, respectively) child of v_i .

For a subtree T_v of T , we use T_v^x to denote the induced subgraph $G^*[V(T_v) \cup \{x\}]$. For each face f in G^* , its *top vertex*, denoted by v_f , is the vertex on the boundary of f with minimum depth in T . Note that, except for the root r , each vertex in T is a top vertex of exactly one face. Finally we define two important faces for f : for the two faces enclosing $T_{p(v_f)}^x$ in G^* , the one that contains both v_f and $p(v_f)$ in its boundary is the *critical face* of f , and the other one is the *nearly critical face* of f (see Figure 5(a)). Note that, if we protect the critical and near-critical faces of f in sequence, the whole subgraph $T_{p(v_f)}^x$ will be cut off from the rest of T .

We are now ready to describe our strategy for the firefighting in the dual graph G^* . Let f be the face where the fire breaks out. Our strategy consists of at most four

²When designating the left and right children of a vertex $v \in V_T \setminus \{r\}$, we also take the exterior vertex x into account. That is, if v has one (zero, respectively) child in T , then there must be one (two) edge(s) in G^* connecting v and x ; we also regard x as a child of v . For example, in Figure 4(b), e is the right child of s , d is the left child of c , and x is both the left and right children of f .

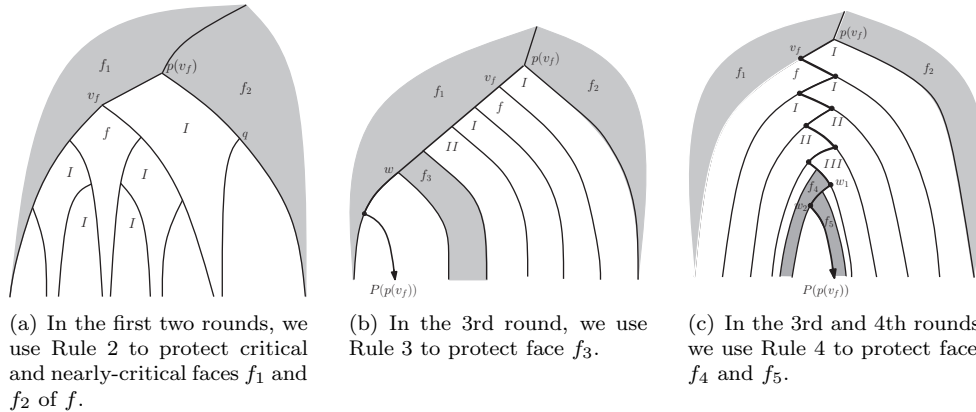


FIG. 5. Our strategy when the fire starts at f . Grey faces are protected. The Roman numeral in a face indicates the earliest time for the face to catch fire.

rounds of protection: the first two rounds will contain the fire in $T_{p(v_f)}^x$, and the next two rounds will go along the heavy path $P(p(v_f))$ to save all faces inside a “heavy subgraph” of $T_{p(v_f)}^x$. The strategy is given by the following four rules carried out in order.

- Rule 1.* We do nothing if $depth(v_f) \leq 1$; otherwise, we use Rules 2–4.
- Rule 2.* For the first two rounds, we protect the critical and nearly critical faces of f in turn to contain the fire inside subgraph $T_{p(v_f)}^x$ (see Figure 5(a)).
- Rule 3.* If the heavy path $P(p(v_f))$ has at least six vertices and no turning vertex among the first five vertices, let w be the its sixth vertex. In the third round, we protect the face whose top vertex is $p(w)$ to save all faces inside subgraph T_w^x (see Figure 5(b)). We do nothing in the fourth round.
- Rule 4.* If the condition in Rule 3 doesn’t hold (thus the strategy in Rule 3 hasn’t been carried out), and if the heavy path $P(p(v_f))$ has at least nine turning vertices, then let w_1 and w_2 be the eighth and ninth turning vertices. Let f_4 and f_5 be the faces whose top vertices are $p(w_1)$ and $p(w_2)$. We protect f_4 and f_5 in the third and fourth rounds to save all faces inside subgraph $T_{w_2}^x$ (see Figure 5(c)).

It is easy to see that the strategy is valid as it does not protect any burning faces. A face f is a *fire source* for g if, when the fire starts at face f , our strategy may not save face g . A vertex u is a *bad ancestor* of a face g if there is a fire source f for g with $depth(v_f) > 1$ such that v_f is a child of u . We now show that our strategy ensures that the average number of burnt faces in G^* is $O(\log n)$. For this purpose, we first put an upper bound on the number of bad ancestors.

LEMMA 4.1. *Every face g has at most $O(\log n)$ bad ancestors.*

Proof. By Rule 2, we know that when the fire starts at a face f with $depth(v_f) > 1$, all faces outside $T_{p(v_f)}^x$ are saved because of the protection of the critical and nearly critical faces of f in the first two rounds. Therefore, any bad ancestor of g is an ancestor of v_g and hence is on the (r, v_g) -path P of T .

To put an upper bound on the number of bad ancestors of g in P , we first decompose P into sections. Because P contains at most $\lfloor \log n \rfloor$ light vertices, the deletion of these light vertices divides P into at most $\lfloor \log n \rfloor + 1$ sections. We add each light vertex to the section that immediately follows it. Let S be an arbitrary such section. The turning vertices on S further divide S into sections, which we refer to as

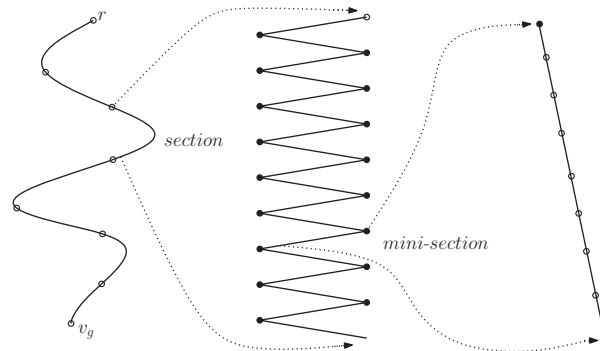


FIG. 6. Decomposition of the (r, v_g) -path into sections and further into minisections.

minisections (see Figure 6). We add each turning vertex to the end of the minisection that immediately follows it. Suppose $v \in P$ is a bad ancestor of g . By Rule 4, we see that v can reside only in the last nine minisections for each section. For each minisection, we see from Rule 3 that v can be only one of the last four vertices in the minisection. It follows that the number of possible bad ancestors of g in each section is at most 36, implying that the total number of possible bad ancestors on P is at most $36(\lfloor \log n \rfloor + 1)$, which completes the lemma. \square

With Lemma 4.1 at hand, we can now establish the surviving rate of outerplanar graphs.

THEOREM 4.2. *The surviving rate of every n -vertex outerplanar graph is $1 - \Theta(\frac{\log n}{n})$.*

Proof. As mentioned earlier, the firefighting problem on vertices of an outerplanar graph G is equivalent to that on faces of its dual graph G^* . From Rules 1 and 2, we see that every fire source f for a face g satisfies either (a) $\text{depth}(v_f) \leq 1$ or (b) $p(v_f)$ is an ancestor of v_g . There are at most four faces satisfying (a), and the number of fire sources satisfying (b) is at most twice the number of bad ancestors of g as each vertex in T has at most two children.

Since the number of bad ancestors of g is at most $O(\log n)$ (by Lemma 4.1), the number of fire sources for g is no more than $4 + 2O(\log n) = O(\log n)$. Therefore, our strategy saves at least $n - O(\log n)$ vertices on average, and the theorem easily follows from this and Theorem 2.3. \square

Acknowledgments. The authors would like to thank the anonymous referees for their valuable comments and suggestions. The authors also would like to thank Jaikumar Radhakrishnan for many discussions and useful suggestions, particularly on the case of outerplanar graphs.

REFERENCES

- [1] L. CAI, E. VERBIN, AND L. YANG, *Firefighting on trees: $(1-1/e)$ -approximation, fixed parameter tractability and a subexponential algorithm*, in Algorithms and Computation, Lecture Notes in Comput. Sci. 5369, 2008, pp. 258–269.
- [2] L. CAI AND W. WANG, *The surviving rate of a graph for the firefighter problem*, SIAM J. Discrete Math., 23 (2009), pp. 1814–1826.
- [3] M. DEVELIN AND S. G. HARTKE, *Fire containment in grids of dimension three and higher*, Discrete Appl. Math., 155 (2007), pp. 2257–2268.
- [4] S. FINBOW, B. HARTNELL, Q. LI, AND K. SCHMEISSER, *On minimizing the effects of fire or a virus on a network*, J. Combin. Math. Combin. Comput., 33 (2000), pp. 311–322.

- [5] S. FINBOW, A. KING, G. MACGILLIVRAY, AND R. RIZZI, *The firefighter problem for graphs of maximum degree three*, Discrete Math., 307 (2007), pp. 2094–2105.
- [6] S. FINBOW AND G. MACGILLIVRAY, *The firefighter problem: A survey of results, directions and questions*, Austral. J. Combin., 43 (2009), pp. 57–78.
- [7] P. FOGARTY, *Catching the fire on grids*, M.Sc. thesis, University of Vermont, Burlington, VT, 2003.
- [8] B. HARTNELL, *Firefighter! An application of domination*, presentation in the 25th Manitoba Conference on Combinatorial Mathematics and Computing, University of Manitoba in Winnipeg, Canada, 1995.
- [9] B. HARTNELL AND Q. LI, *Firefighting on trees: How bad is the greedy algorithm?* Congr. Numer., 145 (2000), pp. 187–192.
- [10] J. KLEINBERG AND E. TARDOS, *Algorithm Design*, Addison–Wesley, Reading, MA, 2005.
- [11] G. MACGILLIVRAY AND P. WANG, *On the firefighter problem*, J. Combin. Math. Combin. Comput., 47 (2003), pp. 83–96.
- [12] K. L. NG AND P. RAFF, *A generalization of the firefighter problem on $z \times z$* , Discrete Appl. Math., 156 (2008), pp. 730–745.
- [13] N. ROBERTSON AND P. D. SEYMOUR, *Graph minors. I. Excluding a forest*, J. Combin. Theory Ser. B, 35 (1983), pp. 39–61.
- [14] A. SCOTT, U. STEGE, AND N. ZEH, *Politician’s firefighting*, in Algorithms and Computation, Lecture Notes in Comput. Sci. 4288, 2006, pp. 608–617.
- [15] P. WANG AND S. A. MOELLER, *Fire control on graphs*, J. Combin. Math. Combin. Comput., 41 (2002), pp. 19–34.