BURNING A GRAPH IS HARD

STÉPHANE BESSY, ANTHONY BONATO, JEANNETTE JANSSEN, DIETER RAUTENBACH, AND ELHAM ROSHANBIN

Abstract. Graph burning is a model for the spread of social contagion. The burning number is a graph parameter associated with graph burning that measures the speed of the spread of contagion in a graph; the lower the burning number, the faster the contagion spreads. We prove that the corresponding graph decision problem is **NP**-complete when restricted to acyclic graphs with maximum degree three, spider graphs and path-forests. We provide polynomial time algorithms for finding the burning number of spider graphs and path-forests if the number of arms and components, respectively, are fixed. Finally, we describe a polynomial time approximation algorithm with approximation factor 3 for general graphs.

1. Introduction

Suppose you were attempting to spread gossip, a meme, or some other social contagion in an online social network such as Facebook or Twitter. Our assumptions, similar to those in the recent study on the spread of emotional contagion in Facebook [16], are that in-person interaction and nonverbal cues are not necessary for the spread of the contagion. Hence, agents in the network spread the contagion to their friends or followers, and the contagion propagates over time. If the goal was to minimize the time it took for the contagion to reach the entire network, then which users would you target with the contagion, and in which order? Related questions emerge in study of the spread of social influence, which is an active topic in social network analysis; see, for example, [7, 14, 15, 18, 19]. *Graph burning* is a simplified deterministic model for the spread of social contagion in a social network that considers an answer to these questions, and was introduced in [5].

Graph burning is a newly discovered deterministic graph process in which we attempt to burn all the nodes as quickly as possible, and is inspired by contact processes on graphs such as graph bootstrap percolation [1], and graph searching paradigms such as Firefighter [6, 9]. Throughout, we work with simple, undirected, and finite graphs. There are discrete time-steps or rounds. At time $t = 0$ all the nodes are unburned. Then at each time $t \geq 1$, we burn one new unburned node if such a node is available. Once a node is burned in round t, each of its unburned neighbours becomes burned in round $t + 1$. If a node is burned, then it remains in that state until the end of the process. The process ends when all nodes are burned.

We denote the node that we burn in the i-th step by x_i, and we call it a *source of fire*. If we burn a graph G in k steps, then the sequence $(x_1, x_2, \ldots, x_k)$ is called a *burning sequence* for G. The *burning number* of G, written by $b(G)$, is the length of a shortest burning sequence for G; such a burning sequence is referred to as *optimum*. In other words, the burning number of G is the minimum number of steps needed for the burning process to end.

Probabilistic results and random variations on the burning process were presented in [17, 20]. New bounds on the burning number of trees and algorithmic aspects of the burning problem were given in [3]. For further background on graph burning, we refer the reader to [20]. The following graph decision problem is our main focus in this paper.

Supported by grants from NSERC.

Problem: Graph Burning
Instance: A simple graph G of order n and an integer $k \geq 2$.
Question: Is $b(G) \leq k$? In other words, does G contain a burning sequence $(x_1, x_2, \ldots, x_k)$?

As we show in Lemma 14 from [5], the burning number of a graph is tightly bounded in terms of its distance domination numbers. We also proved in Corollary 5 from [5] that the burning number of a connected graph G is the minimum burning number over the set of spanning subtrees of G. On the other hand, the burning problem has some similarities to some other known graph processes such as Firefighter. It is known that the distance domination problems are polynomially solvable for trees (see [11]), however, the Firefighter problem is **NP**-complete even for trees of maximum degree three (see [8]). These are our motivation to investigate the complexity of the burning problem for graphs; in particular for trees.

This paper is organized as follows. We first provide a short review on the basic results that we have found on the burning number in [5, 20]. These results are needed in this paper. In Section 1, we prove that the Graph Burning problem is **NP**-complete when restricted to trees of maximum degree three. As a corollary, this shows the **NP**-completeness of the burning problem for chordal graphs, bipartite graphs, planar graphs, and disconnected graphs. In Section 2, we show that the burning problem remains **NP**-complete even for trees with a structure as simple as spider graphs, and also for disconnected graphs such as path-forests. In Section 3, we provide polynomial algorithms for finding the burning number of path-forests and spider graphs, when the number of arms and components is fixed. In the final section, we describe a polynomial time approximation algorithm with approximation factor 3 for general graphs.

1.1. **Preliminaries.** In this subsection, we give some background and notation from graph theory. For further background, see [21]. Then we review some facts about the burning problem from [5,20] that will be useful in the present study.

If v is a node of a graph G, then the *eccentricity* of v is defined as $\max\{d(v,u) : u \in V(G)\}$. The *center* of G consists of the nodes in G with minimum eccentricity. Every node in the center of G is called a *central* node of G. The *radius* of G is the minimum eccentricity over the set of all nodes in G. The *diameter* of G is the maximum eccentricity over the set of all nodes in G. Given a positive integer k, the *k-th closed neighborhood* of v is defined to be the set $\{u \in V(G) : d(u,v) \leq k\}$ and is denoted by $N_k[v]$; we denote $N_1[v]$ simply by $N[v]$. We call a graph a *path-forest* if it is the disjoint union of a collection of paths. For $s \geq 3$, let $K_{1,s}$ denote a *star*; that is, a complete bipartite graph with parts of order 1 and s.

We call a tree that has only one node c of degree at least three a *spider graph*, and the node c is called the *spider head*. In a spider graph every leaf is connected to the spider head by a path which is called an *arm*. If all the arms of a spider graph with maximum degree s are of the same length r, then we denote such a spider graph by $SP(s,r)$. We denote the *disjoint union* of two graphs G and H by $G \cup H$.

The following facts about graph burning can be found in [5, 20]. In a graph G the sequence $(x_1, x_2, \ldots, x_k)$ forms a burning sequence if and only if, for each pair i and j with $1 \leq i < j \leq k$, we have that $d(x_i, x_j) \geq j - i$, and the following set equation holds:

$$N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k] = V(G). \tag{1}$$

By the following corollary from [5], we can get rid of the condition $d(x_i, x_j) \geq j - i$ for a burning sequence.

Corollary 1 ([5]). *If $(x_1, x_2, \ldots, x_k)$ is a sequence of nodes in a graph G, such that $N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k] = V(G)$, then $b(G) \leq k$.*

We next provide a lemma used for the approximation algorithms in Section 4. It follows from (1) and Corollary 1; however, we include a full proof below as it is used in our discussion of approximation algorithms.

Lemma 2. *The burning number of a graph G is the minimum length of a sequence $(x_1, \dots, x_k)$ of nodes in G that satisfies (1).*

Proof. Suppose that k is the minimum length of a sequence in G satisfying (1). Since a burning sequence satisfies (1), we have that $b(G) \geq k$. We want to show that the equality holds. By contradiction, suppose $b(G) > k$. Suppose that S is the set of sequences that satisfy (1). Let $s = (x_1, \dots, x_k)$ be a sequence in S for which $j(s) = \min\{j \in [k] : \text{for some } i \in [j-1] : d(x_i, x_j) < j-i\}$ is the maximum among all of the sequences in S. Note that since $b(G) > k$, the index $j(s)$ is well defined. Let $i(s) \in [j(s) - 1]$ be such that $d(x_{i(s)}, x_{j(s)}) < j(s) - i(s)$. Note that in such a case,

$$N_{k-j(s)}[x_{j(s)}] \subseteq N_{k-i(s)}[x_{i(s)}].$$

Since $k > j(s) - 1$, there is a node y in $V(G) \smallsetminus \big(N_{(j(s)-1)-1}[x_1] \cup N_{(j(s)-1)-2}[x_2] \cup \dots \cup N_0[x_{j(s)-1}]\big)$. Therefore, the sequence $s' = (x_1, \dots, x_{j(s)-1}, y, x_{j(s)+1}, \dots, x_k)$ satisfies (1) and $j(s') > j(s)$, which is a contradiction. ☐

We proved in Lemma 11 of [5] that the burning number of a graph G of radius r is at most $r + 1$. Namely, by burning a central node of G at the first step, every other node will be burned after at most r more steps.

Theorem 3 ([5])**.** *For a path P_n we have that $b(P_n) = \lceil n^{1/2} \rceil$. More precisely, if $n = k^2$ for some integer k, then burning P_n in k steps is equivalent to decomposing P_n into k subpaths of orders $1, 3, \dots, 2k - 1$. If $\lceil n^{1/2} \rceil = k$, and n is not a square number, then every optimum burning sequence for P_n corresponds to a decomposition of P_n into k smaller subpaths $Q_1, Q_2, \dots, Q_k$, in which the order of each Q_i is a number between one and $2i - 1$.*

Theorem 4 ([20])**.** *If G is a path-forest of order n with $t \geq 1$ components, then*

$$b(G) \leq \lceil n^{1/2} \rceil + t - 1.$$

A subgraph H of graph G is called an *isometric* subgraph of G if the distance between any pair of nodes u and v in H equals the distance between u and v in G. For example, any subtree of a tree T is an isometric subgraph of T. The following corollary is a generalization of Theorem 7 in [5] for disconnected graphs.

Corollary 5 ([20])**.** *If G is a graph and H is an isometric subforest of G, then we have that $b(H) \leq b(T)$.*

Note that the only graph with burning number one is K_1. Moreover, the following theorem characterizes the graphs with burning number 2.

Theorem 6 ([20])**.** *If G is a graph of order n, then $b(G) = 2$ if and only if $n \geq 2$, and G has maximum degree $n - 1$ or $n - 2$.*

Since finding the maximum degree of a graph is solvable in polynomial time, by Theorem 6, we can recognize graphs with burning number 2 in polynomial time. Thus, in the rest of the paper we restrict our attention to the case $k \geq 3$.

2. Burning Trees and Forests with Maximum Degree Three

We now consider Graph Burning in acyclic graphs with maximum degree three. In particular, we show in Theorem 7 that the burning problem is NP-complete for trees of maximum degree three. We show the **NP**-completeness of the burning problem by a reduction from a variant of the 3-Partition problem [10]. Here is the statement of this problem.

Problem: Distinct 3-Partition
Instance: A finite set $X = \{a_1, a_2, \dots, a_{3n}\}$ of positive distinct integers, and a positive integer B where $\sum_{i=1}^{3n} a_i = nB$, and $B/4 < a_i < B/2$, for $1 \le i \le 3n$.
Question: Is there any partition of X into n triples such that the elements in each triple add up to B?

In [13], it is shown that the Distinct 3-Partition problem is **NP**-complete *in the strong sense* (see [10]); that is, this problem is **NP**-complete, even when restricted to the cases where B is bounded above by a polynomial in n. In the rest of the paper, by O_m, we mean the set of the m first positive odd integers; that is, $O_m = \{1, 3, \dots, 2m-1\}$.

Theorem 7. *The burning problem is **NP**-complete for trees of maximum degree three.*

Proof. Given a graph G of order n and a sequence $(x_1, x_2, \dots, x_k)$ of the nodes in G, we can easily find $N_{k-i}[x_i]$ in polynomial time, for $1 \le i \le k$. Thus, we can check in polynomial time if $V(G) = \bigcup_{i=1}^{k} N_{k-i}[x_i]$. Hence, the burning problem is in **NP**.

Now, we show the **NP**-completeness of the burning problem for trees of maximum degree three by a reduction from the Distinct 3-Partition problem.

Suppose that we have an instance of the Distinct 3-Partition problem; that is, we are given a non-empty finite set $X = \{a_1, a_2, \dots, a_{3n}\}$ of distinct positive integers, and a positive integer B such that $\sum_{i=1}^{3n} a_i = nB$, and $B/4 < a_i < B/2$, for $1 \le i \le 3n$. Since the Distinct 3-Partition problem is **NP**-complete in the strong sense, without loss of generality we can assume that B is bounded above by a polynomial in the length of the input. Assume that the maximum of the set X is m which is by assumption bounded above by $B/2$. We now construct a tree of maximum degree 3 as follows.

Let $Y = \{2a_i - 1 : a_i \in X\}$. Hence, $Y \subseteq O_m$, and $2nB - 3n = \sum_{i=1}^{3n}(2a_i - 1)$ is the sum of the numbers in Y. Let $Z = O_m \smallsetminus Y$. Note that $1 \le |Y| \le m$, and consequently, $|Z| \le m-1$. Let $|Z| = k$, for some $k \le m-1$. For $1 \le i \le k$, let Q'_i be a path of order l_i, where l_i is the i-th largest number in Z. For $1 \le i \le m+1$, we define T_i to be a spider $SP(3, 2m+1-i)$ with centre r_i. We also take Q_i to be a path of order $2B-3$, for $1 \le i \le n$. Then we combine the graphs that we created above from left to right in the following order:

$$Q_1, T_1, Q_2, T_2, \dots, Q_n, T_n, Q'_1, T_{n+1}, Q'_2, T_{n+2}, \dots, Q'_k, T_{n+k}, T_{n+k+1}, \dots, T_{m+1}$$

such that each graph in this order is joined by an edge from one of its leaves to a leaf of the next graph in the presented order. The resulting graph is named $T(X)$; note that it is a tree of maximum degree three.

For example, let $X = \{10, 11, 12, 14, 15, 16\}$, and $B = 39$. Then $n = 2$, and $m = \max\{a_i : a_i \in X\} = 16$. Therefore, $Y = \{19, 21, 23, 27, 29, 31\}$, and $Z = O_{16} \smallsetminus Y = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 25\}$. Thus, $k = |Z| = |O_{16} \smallsetminus Y| = 10$. The graph $T(X)$ is depicted in Figure 1. For simplicity, we do not draw the nodes in the paths Q_i and Q'_j, and the spiders T_i in the figure.

For $1 \le i \le m+1$, let v_i be a leaf of $T(X)$ that is also a leaf of T_i, as a subgraph of $T(X)$. Note that for $1 \le i \le m+1$ the two arms of T_i that do not contain v_i, together with its centre r_i, form a path. We call this path T'_i. The order of T'_i is $2(2m+1-i)+1 \in O_{2m+1}$. Hence, the subgraph of $T(X)$ induced by

$$\left(\bigcup_{i=1}^{m+1} T'_i\right) \bigcup \left(\bigcup_{i=1}^{n} Q_i\right) \bigcup \left(\bigcup_{i=1}^{k} Q'_i\right)$$

forms a path of order

$$\sum_{i=1}^{2m+1}(2i-1) = (2m+1)^2,$$

that we denote it by P. Therefore,

$$T(X) - P = \bigcup_{i=1}^{m+1}(T_i \setminus P),$$

which is a disjoint union of paths of orders $\{2m+1-i\}_{i=1}^{m+1}$. Note that $T_i \setminus P$ is the arm of T_i that contains v_i. Thus, we have that

$$|V(T(X))| = (2m+1)^2 + \sum_{i=1}^{m+1}(2m+1-i) = (2m+1)^2 + \frac{3(m^2+m)}{2}.$$

Since m is bounded by B and by assumption, B is bounded above by a polynomial in terms of n, then $T(X)$ is obtained in polynomial time in terms of the length of the input.

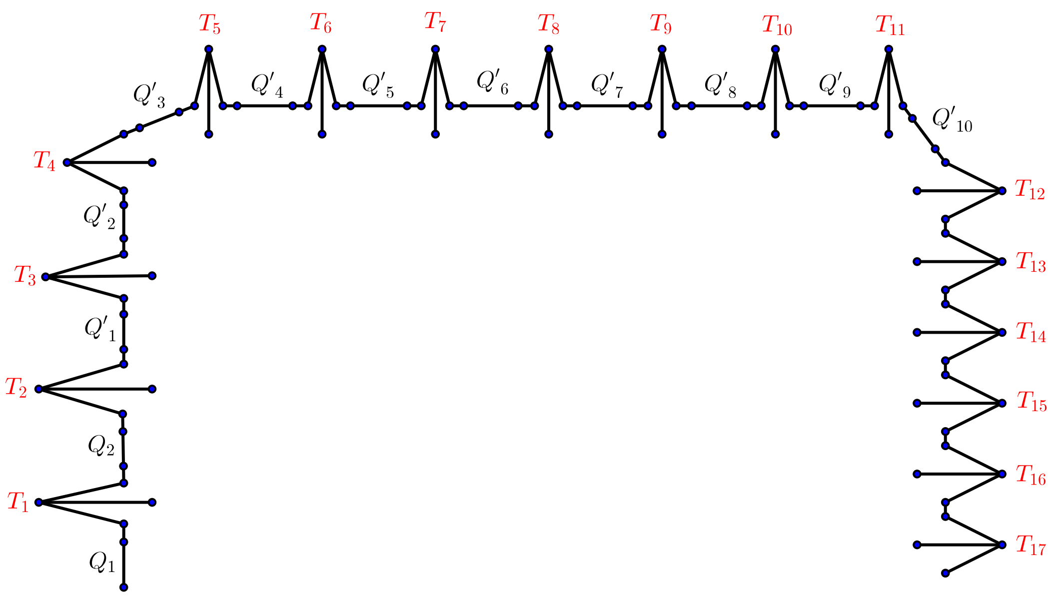


FIGURE 1. A sketch of the tree $T(X)$.

We can easily see that, there is a partition of X into triples such that the elements in each triple add up to B if and only if we can decompose the paths $Q_1, Q_2, \ldots, Q_n$ into subpaths of orders $2a_i - 1 \in Y$. First, assume that there is a partition of X into triples such that the elements in each triple add up to B. Equivalently, we have a partition for the paths $Q_1, Q_2, \ldots, Q_n$ in terms of subpaths $\{P_l : l \in Y\}$. Since $O_m = Y \cup Z$, we conclude that there is a partition for the subgraph $(\bigcup_{i=1}^{n} Q_i) \bigcup (\bigcup_{i=1}^{k} Q'_i)$ in terms of the subpaths $\{P_l : l \in O_m\}$. Now, for $m+2 \le i \le 2m+1$, let x_i be the centre of a path P_l in such a partition, where $l = 2(2m+2-i)-1 \in O_m = Y \cup Z$. For $1 \le i \le m+1$, let $x_i = r_i$ (the centre of T_i). Thus, we have that

$$V(T(X)) = \bigcup_{i=1}^{2m+1} N_{2m+1-i}[x_i].$$

Consequently, by equation (1), we conclude that $(x_1, x_2, \ldots, x_{2m+1})$ forms a burning sequence of length $2m+1$ for $T(X)$. Therefore, $b(T(X)) \le 2m+1$.

Conversely, suppose that $b(T(X)) \leq 2m+1$. Note that the path P of order $(2m+1)^2$ is a subtree of $T(X)$. Therefore, by Theorem 3 and Corollary 5, we have that

$$b(T(X)) \geq b(P) = 2m+1.$$

Thus, we conclude that $b(T(X)) = 2m+1$. Assume that $(x_1, x_2, \ldots, x_{2m+1})$ is an optimum burning sequence for $T(X)$.

We first claim that each x_i must be in P. Since $T(X) = P \cup (\bigcup_{j=1}^{m+1}(T_j \setminus P))$, every x_i is either in P or in $T_j \setminus P$, for some $1 \leq j \leq m+1$. On the other hand, every node in P must receive the fire from one of the x_i's. Note that the only connection of P to $T(X) \setminus P$ is through the nodes r_i. Hence, for $1 \leq i \leq 2m+1$, $N_{2m+1-i}[x_i] \cap P$ must be a path of order at most $2(2m+1-i)+1$. If for some $1 \leq i \leq 2m+1$, a node x_i is out of P, then $N_{2m+1-i}[x_i] \cap P$ is a path of order less than $2(2m+1-i)+1$. Therefore, the total sum of the orders of the subpaths $\{N_{2m+1-i}[x_i] \cap P\}_{i=1}^{2m+1}$ will be less than $(2m+1)^2 = |V(P)|$, which is a contradiction. Thus, every x_i must be selected from P.

Now, we claim that for $1 \leq i \leq m+1$, we must have $x_i = r_i$. We prove this by strong induction on i. Note that each v_i, $1 \leq i \leq m+1$, receives the fire from a fire source $x_j \in P$ (by the above argument) where $1 \leq j \leq 2m+1$. Therefore, $d(x_j, v_i) \leq 2m+1-j$, for some $1 \leq j \leq 2m+1$. For $i=1$, since the only node in P that is within distance $2m+1-i = 2m$ from v_1 is r_1, then we must have $x_1 = r_1$. Suppose that for $1 \leq i \leq m$ and for every $1 \leq j \leq i$, $x_j = r_j$. Since the only node in P within distance $2m+1-(i+1)$ from v_{i+1} is the node r_{i+1}, and by induction hypothesis, we conclude that $x_{i+1} = r_{i+1}$. Therefore, the claim is proved by induction.

Note that $N_{2m+1-i}[r_i] = V(T_i)$. Therefore, the fire started at $x_i = r_i$ will burn all the nodes in T_i. The above argument implies that the nodes in $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ must be burned by receiving the fire started at $x_{m+2}, x_{m+3}, \ldots, x_{2m+1}$ (the last m sources of fire). Since $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ is a disjoint union of paths, then we derive that for $m+2 \leq i \leq 2m+1$, $N_{2m+1-i}[x_i] \bigcap (T(X) \setminus \bigcup_{i=1}^{m+1} T_i)$ is a path of order at most $2(2m+1-i)+1$ $(\leq 2m-1)$. On the other hand, the path-forest $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ is of order

$$\sum_{i=1}^{m}(2i-1) = m^2.$$

Thus, we conclude that for $m+2 \leq i \leq 2m+1$, $N_{2m+1-i}[x_i] \bigcap (T(X) \setminus \bigcup_{i=1}^{m+1} T_i)$ is a path of order equal to $2(2m+1-i)+1$; since otherwise, we can not burn all the nodes in $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ in m steps, which is a contradiction. Therefore, there must be a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ (induced by the burning sequence $(x_{m+2}, x_{m+3}, \ldots, x_{2m+1})$) for $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$) into subpaths $\{P_l : l \in O_m\}$.

Now, considering the partition described in the previous paragraph, we claim that there is a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m in which the paths $Q_1, Q_2, \ldots, Q_n$ are decomposed into paths of orders in Y, and each path Q_i' is covered by itself. Note that by definition, for $1 \leq i \leq k$, each path Q_i' is a component of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$. Hence, it suffices to prove that there is a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m such that each Q_i' is covered by itself. Assume that in a partition of $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ into subpaths of orders in O_m, there is a path Q_i' of order $l \in O_m \setminus Y = Z$ that is partitioned by a union of paths of orders in O_m rather than by P_l itself. We know that P_l must have covered some part of a path Q_j' with $j \neq i$, or must be used in partitioning $Q_1, Q_2, \ldots, Q_n$. Hence, we can easily modify the partition by switching the place of P_l and those paths that have covered P_l (as they have equal lengths). Therefore, we have decreased the number of such displaced paths in our partition for $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$. Since the number of Q_i''s, where $1 \leq i \leq k$, is finite, we will end up after finite number of switching in a partition for $T(X) \setminus \bigcup_{i=1}^{m+1} T_i$ in which every Q_i', $1 \leq i \leq k$, is covered by itself.

Since each Q_i is of order $2B-3$, there must be a partition of Y into triples such that the elements in each triple add up to $2B-3$. Equivalently, there must be a partition of X into triples such that the elements in each triple add up to B. Since $T(X)$ is a tree of maximum degree 3, then, we have a polynomial time reduction from the Distinct 3-Partition problem to the Graph Burning problem for trees with maximum degree 3. □

Since any tree is a chordal graph, and also planar and bipartite, then we conclude the following corollary.

Corollary 8. *The burning problem is* ***NP****-complete for chordal graphs, planar graphs, and bipartite graphs.*

If in the proof of Theorem 7, we keep the graphs Q_i's, Q_i''s, and T_i's disjoint, then we will have exactly the same argument to show a polynomial time reduction from the Distinct 3-Partition problem to the Graph Burning problem. Thus, we have the following immediate corollary as well.

Corollary 9. *The burning problem is* ***NP****-complete for forests of maximum degree three.*

3. Burning Spider Graphs and Path-Forests

In this section, we prove that the Graph Burning problem is **NP**-complete even for spider graphs and path-forests. We first provide some background on the burning number of trees.

A *terminal path* in a tree T is a path P in T such that one of the end points of P is a leaf of T. The other end point of P, that is not necessary a leaf, is called the *non-terminal* end point of P (if P is of order one, then the non-terminal end point of P and the leaf in P coincide). Assume that $\{Q_i\}_{i=1}^t$ is a set of disjoint terminal paths in T, and let v_i denote the non-terminal point of the path Q_i, for $1 \le i \le t$. We call $\{Q_i\}_{i=1}^t$ a *decomposed spider* in T if the path between every pair v_i and v_j does not contain any node of Q_i and Q_j except v_i and v_j.

Theorem 10. *Suppose that $\{Q_i\}_{i=1}^t$, where $t \ge 3$, forms a decomposed spider in a tree T, and let v_i be the non-terminal end point of Q_i, for $1 \le i \le t$. If $d(v_i, v_j) \ge 2k$ for all $1 \le i, j \le t$, and $t \ge k$, then $b(T) \ge k+1$.*

Proof. Let T' be the smallest connected subgraph of T that contains $\bigcup_{i=1}^t Q_i$. Since T' is an isometric subtree of T, to prove that $b(T) \ge k+1$ it suffices to show that $b(T') \ge k+1$, as follows.

First, we show that the burning number of T' is at least k. Let w_i denote the leaf of T' in Q_i. Note that we may have $w_i = v_i$ (in the case that Q_i is of order one). We claim that there is no fire source x_j that spreads the fire to two distinct leaves. By contradiction, suppose that there are two distinct leaves w_i and w_r, and a fire source x_j for which we have that $d(x_j, w_i) \le k-j$ and $d(x_j, w_r) \le k-j$ (that is, w_i and w_r both receive the fire started at x_j). By triangle inequality, we conclude that

$$2k \le d(v_i, v_r) \le d(w_i, w_r) \le d(w_i, x_j) + d(x_j, w_r) \le 2k - 2j < 2k,$$

which is a contradiction. Therefore, it implies that corresponding to every leaf w_i there is a unique fire source x_j such that the fire spread from x_j only burns one leaf of T, that is w_i. Thus, the number of fire sources must be at least as large as the number of the leaves in T' that is $t \ge k$. Hence, we must have $b(T') \ge k$.

Now, we claim that $b(T') \ne k$. By contradiction, suppose that $b(T') = k$, and $(x_1, x_2, \ldots, x_k)$ is an optimum burning sequence for T'. If $t > k$, then the above argument leads to the same contradiction, as the number of the fire sources has to be as large as the number of the leaves. If $t = k$, then let w_i be the leaf that receives the fire from x_k. Since $b(T') = k$, then it implies that $x_k = w_i$. We claim that there is no fire source $x_j \ne x_k$ with $d(v_i, x_j) \le k$. By contradiction, suppose that there is a fire source $x_j \ne x_k$ with $d(v_i, x_j) \le k$, and let w_r be the leaf of T' that receives the fire spread from x_j. Thus, we have that

$$2k \le d(v_r, v_i) \le d(w_r, v_i) \le d(w_r, x_j) + d(x_j, v_i) \le k - j + k < 2k,$$

which is a contradiction.

Let s be a neighbour of v_i that is not in the path between v_i and $w_i = x_k$. Since $t \geq 3$, we are sure that such a node s does exist. By assumption, we know that $x_k = w_i$, and therefore s can not receive the fire spread from x_k. On the other hand, the distance between s and any other fire source must be at least k. Thus, s can not be burned by the end of the k-th step, which is a contradiction. Hence, we have that $b(T') \geq k + 1$. □

Assume that we want to find the burning number and an optimum burning sequence for a given tree T. If there is an optimum burning sequence $(x_1, x_2, \dots, x_k)$ for T such that $V(T) \subseteq N_{k-1}[x_1]$, then we can see that $b(T) = \text{radius}(T) + 1 = k$. If $T \setminus N_{k-1}[x_1]$ is non-empty, then it implies that

$$T \setminus N_{k-1}[x_1] \subseteq N_{k-2}[x_2] \cup N_{k-3}[x_3] \cup \dots \cup N_0[x_k].$$

This observation suggests the following conjecture.

Conjecture 11. *Suppose that $\{Q_i\}_{i=1}^t$, where $t \geq 3$, forms a decomposed spider in a tree T, and let v_i be the non-terminal end point of Q_i, for $1 \leq i \leq t$. If $b(\cup_{i=1}^t Q_i) \geq k$, and $d(v_i, v_j) \geq 2k$ for all $1 \leq i, j \leq t$, then $b(T) \geq k + 1$.*

Conjecture 11 may be helpful in finding a lower bound on the burning number of a tree T (as we can see the truth of it for paths by Theorem 3 [5], and also we will see the truth of it later on for some specific spider graphs). In particular, if the burning number of a tree T is strictly less than $\text{radius}(T) + 1$, and the conjecture was true, then by starting from the leaves of T we could find a good lower bound on $b(T)$. Note that by Theorem 10, when $t \geq k$, the above conjecture is true. Also, we can prove the following lemma, since the leaves in any spider graph $SP(s, r)$, with $s \geq r$, form a decomposed spider that satisfies the conditions in Theorem 10.

Lemma 12. *For a spider graph $SP(s, r)$, with $s \geq r$, we have that $b(SP(s, r)) = r + 1$. Moreover, for $s \geq r + 2$, every optimal burning sequence of $SP(s, r)$ must start by burning the central node.*

Proof. We know that $b(SP(s, r)) \leq r + 1$, as $SP(s, r)$ has radius r. Since $SP(r, r)$ is an isometric subgraph of $SP(s, r)$ where $s \geq r$, then it suffices to show that $b(SP(r, r)) = r + 1$.

First, we prove that $b(SP(r, r)) \geq r+1$. We index the leaves of $SP(r, r)$ with $w_1, w_2, \dots, w_r$. For $1 \leq i \leq r$, let Q_i be the graph induced by w_i; that is, Q_i is a path of order one. Hence, every Q_i is a terminal path in $SP(r, r)$ with the non-terminal end w_i, and for every distinct pair $1 \leq i, j \leq r$, we have that $d(w_i, w_j) = 2r$. Therefore, by Theorem 10, we conclude that $b(SP(r, r)) \geq r + 1$. Hence, the proof follows.

Now, suppose that $s \geq r + 2$ and there exists an optimal burning sequence $(x_1, x_2, \dots, x_{r+1})$ for $SP(s, r)$ in which x_1 is not the central node. Since $s \geq r + 2$ and $b(SP(s, r)) = r + 1$, then by Pigeonhole Principle, one of the arms does not include any source of fire, unless we choose the central node as a fire source. Note that by assumption, x_1 is not the central node. Since the only connection between the nodes in that arm to the rest of the nodes in $SP(s, r)$ goes through the central node, then in both cases, we need at least $1 + (r + 1)$ steps for burning the leaf on that arm, which is a contradiction. Thus, every optimal burning sequence of $SP(s, r)$ starts by burning the central node where $s \geq r + 2$. □

Using the above lemma, we now prove that the burning problem is **NP**-complete even for spider graphs. We note that Theorem 13 and Corollary 14 were proven independently in [3], and in a preprint of this paper on arXiv [4].

Theorem 13. *The burning problem is **NP**-complete for spider graphs.*

Proof. Clearly, by Theorem 7, the burning problem is in **NP**. As in the proof of Theorem 7, we give a reduction from the Distinct 3-Partition problem into the burning problem, in which the gadget graph that we construct is a spider graph.

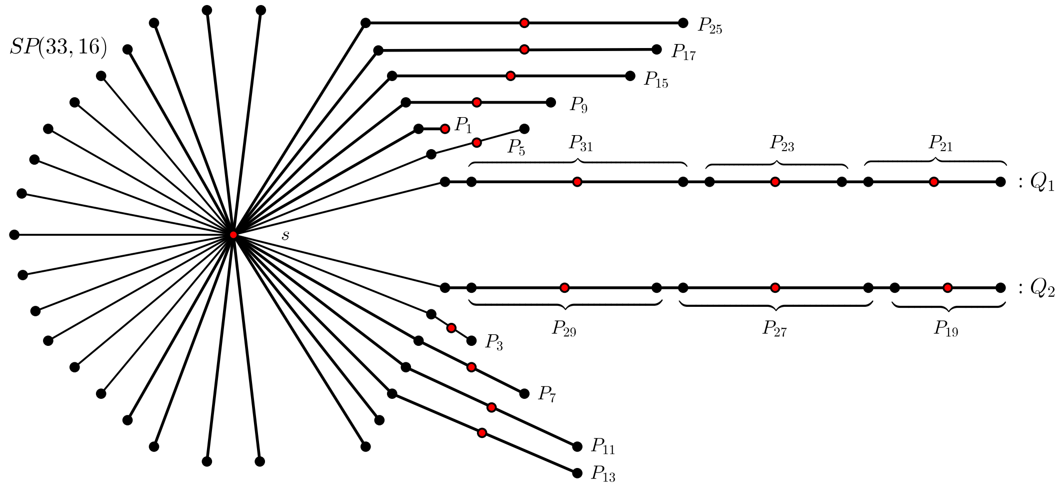


FIGURE 2. A sketch of the tree G.

Given an instance of the Distinct 3-Partition problem, that is, a set $X = \{a_1, a_2, \ldots, a_{3n}\}$ of positive distinct integers and a positive integer B such that each $B/4 < a_i < B/2$, we construct a graph G as follows. Since the Distinct 3-Partition problem is strongly **NP**-complete (as in the proof of Theorem 7), without loss of generality we assume that B is bounded above by a polynomial in the length of the input.

Suppose that $\max X = m+1$, and let $Y = \{2a_i - 1 : a_i \in X\}$. Clearly, $Y \subseteq O_{m+1}$. Then we make a copy of the spider graph $SP(2m+5, m+1)$ with centre s, called G_s. Now, for any positive odd integer $l \in O_{m+1} \smallsetminus Y$, we connect by an edge a leaf of a copy of P_l (a path on l nodes) to a distinct leaf of $SP(2m+5, m+1)$. We connect (by an edge) n copies of P_{2B-3}, called $Q_1, Q_2, \ldots, Q_n$ to distinct leaves of $SP(2m+5, m+1)$ that we have not used for attaching any other P_l, with $l \in O_{m+1} \smallsetminus Y$. We call the resulting graph G. Clearly, G is a spider tree with spider head s. Since $V(G)$ is the disjoint union of the spider graph $SP(2m+5, m+1)$ and the paths $Q_1, Q_2, \ldots, Q_n$, and the paths P_l, with $l \in O_{m+1} \smallsetminus Y$, we have that

$$|V(G)| = \sum_{i=1}^{m+1} (2i-1) + (2m+5)(m+1) + 1 = (m+1)^2 + (2m+5)(m+1) + 1,$$

which is of order $O(B^2)$ in terms of B. Since B is bounded above by a polynomial in terms of n, it implies that we construct graph G in polynomial time in terms of n. We want to show that, there is a partition of X into n triples such that the numbers in each triple add up to B if and only if $b(G) \leq m+2$.

First, assume that there is a partition of X into n triples such that the numbers in each triple add up to B. Consequently, paths $Q_1, Q_2, \ldots, Q_n$ can be partitioned into smaller paths of orders $\{2a_i - 1 : a_i \in X\}$. For $l \in Y$, we set $x_{m+2-(\frac{l-1}{2})}$ to be the middle node of the paths P_l, applied in such a partition of $Q_1, Q_2, \ldots, Q_n$. Then we take $x_1 = s$, and for any $l \in O_{m+1} \smallsetminus Y$, we set the middle node of P_l as $x_{m+2-(\frac{l-1}{2})}$. The sequence $(x_1, x_2, \ldots, x_{m+2})$ is a burning sequence for G. Thus, $b(G) \leq m+2$.

For example, let $X = \{10, 11, 12, 14, 15, 16\}$, and $B = 39$. Then the graph G is shown in Figure 2. Here, we have that $n = 2$, and $m = \max\{a_i : a_i \in X\} - 1 = 15$. Therefore, $Y = \{19, 21, 23, 27, 29, 31\}$, and $O_{16} \smallsetminus Y = \{1, 3, 5, 7, 9, 11, 13, 15, 17, 25\}$. The red nodes in Figure 2 denote a burning sequence of length 17 for tree G.

Conversely, suppose that $b(G) \le m+2$. Since G_s is an isometric subtree of G, then, Theorem 5 and Lemma 12 imply that $b(G) = m+2$. Thus, G has a burning sequence $(x_1, x_2, \ldots, x_{m+2})$. We have to show that there is a partition of X into n triples such that the numbers in each triple add up to B. First, note that we use at most $m+1$ leaves of G_s for attaching the paths P_l, with $l \in O_{m+1}$, and the paths $Q_1, Q_2, \ldots, Q_n$. Thus, there is a copy of $SP(m+4, m+1)$ that is an isometric subtree of G and the only connection of its leaves to the rest of G is through node s. Therefore, by Lemma 12, we conclude that $x_1 = s$. On the other hand, by burning node s at the first step, all the nodes in G_s will be burned by the end of the $(m+2)$-th step. Thus, without loss of generality we can assume that for $2 \le i \le m+2$, all x_i's are selected from $G \setminus G_s$.

Now, by equation (1), we know that $G \setminus G_s = \bigcup_{i=2}^{m+2} N_{m+2-i}[x_i]$. Since $G \setminus G_s$ is a path-forest, then $N_{m+2-i}[x_i]$ must be a path of order at most $l = 2(m+2-i)+1$, for $2 \le i \le m+2$. Besides, we have that

$$
\begin{aligned}
|V(G \setminus G_s)| &= 2nB - 3n + \sum_{l \in O_{m+1} \setminus Y} l \\
&= \sum_{i=2}^{m+2} \left(2(m+2-i)+1\right).
\end{aligned}
$$

Therefore, it implies that $N_{m+2-i}[x_i]$ must be a path of order exactly equal to $l = 2(m+2-i)+1$, for $2 \le i \le m+2$. Hence, there must be a partition of $G \setminus G_s$ by the set of paths of orders in O_{m+1}, in which the center of each path in the partition is a fire source.

We claim that there is a burning sequence for G in which the central node of each P_l, $l \in O_{m+1} \setminus Y$ (that we attached to a leaf of G_s), is selected as a fire source. We can easily prove this claim by switching the paths that are possibly displaced in the current partition for $G \setminus G_s$. Thus, the closed neighbourhoods of the rest of the fire sources form a partition for $Q_1, Q_2, \ldots, Q_n$ in terms of paths of orders $2a_i - 1 \in Y$. Since each Q_i is of order $2B-3$, then it implies that there is partition for X into triples such that the elements in each triple add up to B. □

If we delete the spider graph $SP(2m+5, m+1)$ in the proof of Theorem 13, and keep the rest of the parts of the gadget graph G the same, then we will have the analogous argument for the disjoint union of the paths $Q_1, Q_2, \ldots, Q_n$, and the paths P_l with $l \in O_{m+1} \setminus Y$. Thus, we can have a reduction from the Distinct 3-Partition problem to the burning problem for the path-forests. Therefore, we conclude the following corollary.

Corollary 14. *The Burning problem is* ***NP****-complete for path-forests.*

Note that in Theorem 13 and the above corollary, we do not have any restriction on the number of the arms in $SP(2m+5, m+1)$ and on the length of the paths in constructing the gadget graphs. In other words, the parameter m is unbounded.

4. Algorithms for Burning Path-Forests and Spider Graphs

In this section, we present a polynomial time algorithm that finds the burning number of path-forests when the number of components and their orders are restricted, and then we find another polynomial time algorithm that finds the burning number of spider trees with fixed number of arms and with restrictions on the length of the arms. We first provide some terminology.

Let G be a path-forest with components $Q_1, Q_2, \ldots, Q_k$, where $k \ge 1$, and the order of each path Q_i is l_i such that $l_1 \ge l_2 \ge \cdots \ge l_k$. In other words, we assume that the paths are indexed according to the decreasing order of their lengths. We say that G is a *maximal path-forest* if it can be decomposed into paths of orders $1, 3, \ldots, 2t-1$ for some positive integer t. It is clear that such a graph G is of order t^2. Let MPF_t denote the set of all maximal path-forests of order t^2. If G is a path-forest with burning number t, then G corresponds to a sequence of positive integers such as

$(l_1, l_2, \ldots, l_s)$, where $s \leq t$, and $l_1 \geq l_2 \geq \cdots \geq l_s$, in which l_i denotes the order of the i-th component of G. From now on, we represent a path-forest with burning number t by a sequence of integers as defined above.

We denote the set of maximal path-forests with t components and with burning number k by MPF_k^t. For example, $\mathrm{MPF}_1^1 = \{P_1\} = \{(1)\}$. In general, we can see that for any $k \geq 1$, $\mathrm{MPF}_k^k = \{(2k-1, 2k-3, \ldots, 1)\}$. Also, note that for any $k \geq 1$, $\mathrm{MPF}_k^1 = \{P_{k^2}\}$.

Algorithm 15. *Suppose that $G = (s_1, s_2, \ldots, s_t)$, for a constant $t \geq 1$, represents a path-forest in which s_i denotes the order of the i-th component of G, and $s = s_1 \geq s_2 \geq \cdots \geq s_t$. Then we perform the following steps.*

Stage 1. *First, for each $1 \leq r \leq t-1$, we perform Stages 1.1 and 1.2:*
Stage 1.1. *We set $\mathrm{MPF}_r^r = \{(2r-1, 2r-3, \ldots, 1)\}$.*
If $(s_1, s_2, \ldots, s_r) \notin \mathrm{MPF}_r^r$, then go to the next step.

Stage 1.2. *For $k \geq r+1$, we perform the following steps:*

Stage 1.2.1. *For each $H = (l_1, l_2, \ldots, l_{r-1}) \in \mathrm{MPF}_{k-1}^{r-1}$, we form the sequence $H' = (2k-1, l_1, \ldots, l_{r-1})$. We rearrange the numbers in the sequence H' if they do not appear in a decreasing order, and we add it to the set MPF_k^r.*
If $(s_1, s_2, \ldots, s_r) \subseteq H'$, then finish Stage 1.2, and go to the next stage.

Stage 1.2.2. *For each $H = (l_1, l_2, \ldots, l_r) \in \mathrm{MPF}_{k-1}^r$, and each $1 \leq i \leq r$, we form the sequence $H_i = (l_1, \ldots, l_{i-1}, l_i + 2k - 1, l_{i+1}, \ldots, l_r)$. We rearrange the numbers in the sequences H_i if they do not appear in a decreasing order, and we add them to the set MPF_k^r.*
If $(s_1, s_2, \ldots, s_r) \subseteq H_i$, then finish Stage 1.2, and go to the next stage.

Stage 2. *For $r = t$, we perform the following steps:*

Stage 2.1. *We set $\mathrm{MPF}_t^t = \{(2t-1, 2t-3, \ldots, 1)\}$.*
If $G \in \mathrm{MPF}_t^t$, then stop and return $b(G) = t$.

Stage 2.2. *For $k \geq t+1$, we perform the following steps:*

Stage 2.2.1. *For each $H = (l_1, l_2, \ldots, l_{t-1}) \in \mathrm{MPF}_{k-1}^{t-1}$, we form the sequence $H' = (2k-1, l_1, \ldots, l_{t-1})$. We rearrange the numbers in the sequence H' if they do not appear in a decreasing order, and we add it to the set MPF_k^t.*
If $G \subseteq H'$, then stop and return $b(G) = k$.

Stage 2.2.2. *For each $H = (l_1, l_2, \ldots, l_t) \in \mathrm{MPF}_{k-1}^t$, and for each $1 \leq i \leq t$, we form the sequence $H_i = (l_1, \ldots, l_{i-1}, l_i + 2k - 1, l_{i+1}, \ldots, l_t)$. We rearrange the numbers in the sequence H_i if they do not appear in a decreasing order, and we add it to the set MPF_k^t.*
If $G \subseteq H_i$, then stop and return $b(G) = k$.

The algorithm works since every graph G that is not a subgraph of a graph in MPF_i^t, for all $1 \leq i < k$, but G is a subgraph of a graph in MPF_k^t, has burning number k. Besides, we have the following fact about Algorithm 15.

Theorem 16. *Algorithm 15 finds the burning number of G in time $O(s^t)$, that is polynomial for fixed t.*

Proof. Given the graph G, suppose that for some $k \geq t$, Algorithm 15 stops by recognizing G as a subgraph of a graph in MPF_k^t. Note that the order of the components of G is bounded above by s, and t is a fixed constant in terms of s. Thus, by Theorem 4, we derive that if $H = (l_1, l_2, \ldots, l_r)$

is a graph in MPF_i^r (generated by Algorithm 15), with $1 \le r \le t$ and $i \ge r$, then

$$b(H) \le \left\lceil \sqrt{\sum_{j=1}^{r} l_j} \right\rceil + i - 1 \le \sqrt{st} + t - 1 = O(\sqrt{s}).$$

On the other hand, since $b(H) = i$, then there is a partition of the set O_i into subsets $\{A_j\}_{j=1}^r$ such that $l_j = \sum_{a \in A_j} a$, for $1 \le j \le r$. It implies that $l_j \le \sum_{a \in O_i} a = i^2 = O(s)$, for $1 \le j \le r$. Hence, the length of the longest l_j that appears in the representation of such a graph H is of order s. Let $l = O(s)$ be the length of the longest component in a graph H generated by Algorithm 15. Thus, any graph H generated by Algorithm 15 is a subgraph of the graph $G_0 = (l, l, \dots, l)$ with t components. Since these graphs are distinct, then the total number of graphs generated by Algorithm 15 is of order $O(s^t)$.

Moreover, note that for $r = t$ and $k \ge t$, each time that we add a new graph $H = (l_1, l_2, \dots, l_t)$ to MPF_k^t, we check to see if G is a subgraph of H or not. We simply can do this comparison by checking if $s_i \le l_i$, for $1 \le i \le t$. Thus, the total number of steps that we perform in Algorithm 15 is bounded above by $O(ts^t)$. Since t is a fixed constant in terms of s, then Algorithm 15 is polynomial time in the length of the input. □

In the following, we try to find the burning number of spider graphs, again using a dynamical programming approach. First we need some facts to use for this algorithm. We state the following lemma since we use it for proving the next theorem. Assume that G and H are two disjoint graphs, and $u \in G$ and $v \in H$ are two nodes. We can make a new graph $G + uv + H$ by adding edge uv to $G \cup H$.

Lemma 17. *If G and H are two disjoint non-empty graphs then we have that*

$$b(G + uv + H) \le b(G \cup H),$$

where $u \in V(G)$ and $v \in V(H)$.

Proof. Since $V(G + uv + H) = V(G \cup H)$, then every burning sequence for $G \cup H$ induces a covering for $G + uv + H$; in particular, any minimum burning sequence of $G \cup H$ induces a covering for $G + uv + H$. Therefore, $b(G + uv + H) \le b(G \cup H)$. □

The following theorem plays a key role in the algorithm that we will present, and shows that for a spider tree we always can have an optimum burning sequence in which the first source of fire is close to the spider head.

Theorem 18. *If G is a spider graph with $s \ge 3$ arms and the spider head c, then there is an optimum burning sequence $(x_1, x_2, \dots, x_k)$ for G such that $d(x_1, c) \le k - 1$.*

Proof. We prove this by strong induction on the number of nodes in G. The smallest order spider graph is a star with three leaves. By Theorem 6, we know that the burning number of such a star equals 2 and in every optimum burning sequence for this graph the first fire must be the centre that is the spider head. Hence, the theorem statement is true for this spider.

Now, suppose that the theorem statement is true for every spider graph of order at most $n-1$, and G is a spider graph of order n with $s \ge 3$ arms and spider head c. Also, assume that $L_1, L_2, \dots, L_s$ are the arms of G, and $v_1, v_2, \dots, v_s$ are their corresponding leaves. Finally, suppose that the order of each arm L_i is denoted by l_i. Let $(x_1, x_2, \dots, x_k)$ be an optimum burning sequence for G. By equation (1), we know that

$$V(G) = N_{k-1}[x_1] \cup N_{k-2}[x_2] \cup \dots \cup N_0[x_k].$$

If $d(x_1, c) \le k - 1$, then we are done. Hence, let $d(x_1, c) \ge k$, and $x_1 \in L_i$ where $1 \le i \le s$. We consider two possibilities for l_i: either $l_i \le 2k - 2$ or $l_i \ge 2k - 1$.

Case 1. If $l_i \le 2k-2$, then it implies that $d(c, v_i) \le 2k-2$. Let x be the node in L_i for which $d(x, v_i) = k-1$. Therefore, we have that $d(c, x) \le k-1$. Note that we can cover all the nodes in $L_i \cup \{c\}$ with $N_{k-1}[x]$. Hence, $G - N_{k-1}[x]$ is a subforest of $G - N_{k-1}[x_1]$. Thus, we still have that

$$V(G) = N_{k-1}[x] \cup N_{k-2}[x_2] \cup \ldots \cup N_0[x_k].$$

Note that some of the fire sources x_j's, with $j \ge 2$, might be in $N_{k-1}[x] \cap L_i$. Therefore, by Corollary 5, we have that $b(G \setminus N_{k-1}[x]) = t \le k-1$. Hence, we can find a burning sequence of length t such as $(x_2', x_3', \ldots, x_t')$ for $G \setminus N_{k-1}[x]$. Also, for $t+1 \le j \le k$, we define x_j' to be a node of distance $j-1$ from x. Thus, for $t+1 \le j \le k$, $d(x_j', x) \ge j-1$, and $d(x_j', x_r) \ge r-1+j-1 \ge j-r$, for any $2 \le r \le t$. Therefore, the sequence $(x_1' = x, x_2', \ldots, x_k')$ forms a desired optimum burning sequence for G.

Case 2. If $l_i \ge 2k-1$, then either $d(v_i, x_1) \le k-1$ or $d(v_i, x_1) \ge k$. We claim that there is a burning sequence for G such as $(x_1', x_2', \ldots, x_k')$ such that $x_1' \in L_i$ and $d(x_1', v_i) \le k-1$, or equivalently, $G \setminus N_{k-1}[x_1']$ is connected. If $d(x_1, v_i) \le k-1$, then we are done. If $d(v_i, x_1) \ge k$, then $G \setminus N_{k-1}[x_1]$ is the disjoint union of a spider graph G' and a path P, such that P is a subpath of L_i containing v_i. Let u be the leaf of G' that is in L_i, and v be the other end point of P that probably is different from v_i. We know that $b(G \setminus N_{k-1}[x_1]) \le k-1$. Hence, by Lemma 17, we have that

$$t = b(G' + uv + P) \le b(G' \cup P) = b(G \setminus N_{k-1}[x_1]) \le k-1.$$

Note that $G' + uv + P$ is a subtree of G that is (isomorphic to) a spider of the same number of arms as G. In fact, the i-th arm of $G' + uv + P$ is (isomorphic to) a subpath of L_i with exactly $2k-1$ less nodes than L_i. Also, note that some of the fire sources x_j's, with $j \ge 2$, might be in $N_{k-1}[x_1] \cap L_i$. Let $(x_2', x_3', \ldots, x_t')$ be an optimum burning sequence for $G' + uv + P$, and x_1' be the node in L_i with $d(x_1', v_i) = k-1$. Also, for $t+1 \le j \le k$, we take x_j' to be a node of distance j from x_1' that is on the path connecting x_1' and v_i. Thus, the sequence $(x_1', x_2', \ldots, x_k')$ forms a burning sequence for G, such that $x_1' \in L_i$, and $G \setminus N_{k-1}[x_1']$ is connected.

Now, by above claim, without loss of generality, we assume that $N_{k-1}[x_1]$ contains v_i. That is, we have a burning sequence $(x_1, x_2, \ldots, x_k)$ for G such that $G' = G \setminus N_{k-1}[x_1]$ is a spider graph with smaller number of nodes than G, and with the same number of arms and the same spider head c. In fact, for $j \ne i$, and $1 \le j \le s$, L_j is the j-th arm of G' too, and the i-th arm of G' is a subset of L_i that contains exactly $2k-1$ nodes less than L_i. Hence, we have that $b(G') = t \le k-1$, and by induction hypothesis, G' must have a burning sequence $(x_2', x_3', \ldots, x_t')$ such that $d(x_2', c) \le t-1 \le k-2$. We have two possibilities: either $x_2' \in L_i$, or $x_2' \in L_j$ for some $j \ne i$.

If $j = i$, then let x be the neighbour of x_2' that is on the path which connects x_2' to v_i. Also, let x' be the neighbour of x_1 that is on the path connecting x_1 to v_i. Hence, we have that

$$G \setminus (N_{k-1}[x] \cup N_{k-2}[x']) = G \setminus (N_{k-1}[x_1] \cup N_{k-2}[x_2']).$$

Now, for $t+1 \le r \le k$, we take x_r' to be the node in L_i on the path connecting v_i to x' that is of distance $r-2$ from x'. Finally, we take $x_1' = x$, and we redefine $x_2' = x'$. Thus, the sequence $(x_1', x_2', \ldots, x_k')$ forms a burning sequence for G in which $d(x_1', c) \le k-1$.

If $j \ne i$, then let x be the neighbour of x_2' that is on the path connecting x_2' to c. Also, let x' be the neighbour of x_1 that is closer to v_i. Hence, we have that

$$L_i \setminus (N_{k-1}[x] \cup N_{k-2}[x']) = L_i \setminus (N_{k-1}[x_1] \cup N_{k-2}[x_2']),$$

(by isomorphism). Also,

$$L_j \setminus (N_{k-1}[x] \cup N_{k-2}[x']) = L_j \setminus (N_{k-1}[x_1] \cup N_{k-2}[x_2']).$$

But,

$$G \setminus (N_{k-1}[x] \cup N_{k-2}[x']) \subseteq G \setminus (N_{k-1}[x_1] \cup N_{k-2}[x_2']),$$

and we know that $N_{t-2}[x'_3] \cup N_{t-3}[x'_4] \cup \ldots \cup N_0[x'_t]$ forms a covering for $G \setminus (N_{k-1}[x] \cup N_{k-2}[x'])$. In fact, $G \setminus (N_{k-1}[x] \cup N_{k-2}[x'])$ is an isometric subforest of $G \setminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2])$. Thus, by Corollary 5, we have that

$$\begin{aligned} b(G \setminus (N_{k-1}[x] \cup N_{k-2}[x'])) &\leq b(G \setminus (N_{k-1}[x_1] \cup N_{k-2}[x'_2])) \\ &\leq t-1 \leq k-2. \end{aligned}$$

Hence, there must be an optimum burning sequence $(x''_3, x''_4, \ldots, x''_{t'})$, where $t' \leq t$ for $G \setminus (N_{k-1}[x] \cup N_{k-2}[x'])$. Now, for $t'+1 \leq r \leq k$, we take x''_r to be the node in L_i on the path connecting v_i to x' that is of distance $r-2$ from x'. Finally, we take $x''_1 = x$, and we define $x''_2 = x'$. Thus, the sequence $(x''_1, x''_2, \ldots, x''_k)$ forms a burning sequence for G in which $d(x''_1, c) \leq k-1$. □

The following lemma provides us with another key tool for finding the burning number of spider graphs.

Lemma 19. *Let G be a spider graph with spider head c. Also, suppose that for a positive integer k and a node $x \neq c$ in G, $G \setminus N_{k-1}[x]$ is a path-forest (that is, $d(x,c) \leq k-1$) with at least two components, and $b(G \setminus N_{k-1}[x]) \leq k-1$. If the path-forest $G \setminus N_{k-1}[x]$ is not of order $(k-1)^2$, and the neighbour of x on the path connecting x to c is x', then we have that $b(G \setminus N_{k-1}[x']) \leq k-1$.*

Proof. Assume that a spider graph G with the above conditions is given, and we have the nodes x and x' as mentioned in the lemma's statement. Let x be in an arm of G called L_s. We have two possibilities for L_s: either $L_s \setminus N_{k-1}[x]$ is empty or not.

First, suppose that $L_s \setminus N_{k-1}[x]$ is not empty. Hence, since in this case one of the components of $G \setminus N_{k-1}[x]$ is contained in L_s, then it implies that L_s is of order at least $k + d(x,c)$. By assumption, we know that each component of $G_1 = G \setminus N_{k-1}[x]$ is a subset of one of the arms in G. Let G_2 be the path-forest $G \setminus N_{k-1}[x']$. We know that G_2 is a path-forest since by assumption, $d(x',c) \leq d(x,c) \leq k-1$. Hence, each component of G_2 is also a subpath of an arm in G. We call the components of G_1 and G_2 that are subpaths of L_s by P and P', respectively. In fact, P' is a superset of P with exactly one more node. Also, each non-empty component of G_2 such as $Q' \neq P'$ is a subset of the corresponding component $Q \neq P$ of G_1, and has exactly one node less than Q.

Since by assumption, $b(G_1) = t \leq k-1$, then there must be a burning sequence $(x_1, x_2, \ldots, x_t)$ for G_1. Note that each $N_{t-j}[x_j]$ is a path of order at most $2(t-j)+1$. Therefore, the closed neighbourhoods of the x_i's cover all the nodes in G_2, except for probably the extra node in P' that is a superset of P. We have two possibilities: either $t = k-1$ or $t < k-1$. If $b(G_1) = t < k-1$, then let $x'_1 = x'$, and $x'_i = x_{i-1}$, for $2 \leq i \leq t+1$. We can easily see that $\{N_{t+1-i}[x'_i]\}_{i=1} t+1$ forms a covering for the node set of G_2. Hence, by Corollary 1, we conclude that $b(G_2) \leq t+1 \leq k-1$. If $b(G_1) = t = k-1$, then we have two possibilities: either there is a component $Q \neq P$ in G_1 that is of order one, or the order of each component $Q \neq P$ of G_1 is of order at least two.

If there is a component $Q \neq P$ of G_1 that is of order one and is burned by x_i, then let x'_i be the extra node in $P' \setminus P$. If $(x_1, x_2, \ldots, x_{k-1})$ does not burn x'_i, then $(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_{k-1})$ is a burning sequence for G_2. Thus, $b(G_2) \leq k-1$.

If every component $Q \neq P$ of G_1 is of order at least two, then we have again two possibilities: either x_{k-1} is in P, or $x_{k-1} \notin P$.

If x_{k-1} is in P, then let i be the smallest index for which $x_i \in P$, but x_{i-1} is not in P. We know that such an index i does exist, since otherwise, it means that all the x_i's must be in P, and consequently, it implies that P is the only component of G_1, which is a contradiction. Thus, there must an index i such that $x_i \in P$, but x_{i-1} is in a component of G_1 that we call it Q, with $Q \neq P$. Since, each $N_{k-1-j}[x_j]$ is a path of order at most $2(k-1-j)+1$, then we can assume that $N_{k-1-i}[x_i]$ is potentially able to cover at least two nodes less than $N_{k-1-(i-1)}[x_{i-1}]$.

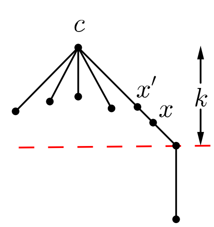
We have two possibilities: the path P either consists of only the single node x_{k-1} or not. If P contains at least two nodes, then let $x'_i = x_{i-1}$ and $x'_{i-1} = x_i$. Therefore, we have a new covering for G_2 induced by $(x_1, \dots, x_{i-2}, x'_{i-1}, x'_i, \dots, x_{k-1})$ in which all the nodes of P' plus one extra node of $L_s \setminus P'$ is covered, while we may have lost covering one node in Q. Now, by moving x_{k-1} to cover such a uncovered node in Q, and shifting the place of the fire sources used for covering P without changing their order (if it is necessary), we find a covering for G_2 with $k-1$ closed neighbourhoods of restricted radii. Hence, by Corollary 1, $b(G_2) \leq k-1$. If x_{k-1} is the only node in P, then note that by assumption G_1 is not of order $(k-1)^2$ and $b(G_1) = k-1$. Thus, G_1 must be of order less than $(k-1)^2$. Therefore, by Theorem 3, there is a partition for G_1 induced by $\{N_{k-1-j}[x_j]\}_{j=1}^{k-1}$. Since G_1 is of order less than $(k-1)^2$, there must be an $1 \leq i < k-1$ (that is, $i \neq k-1$) for which $|N_{k-1-i}[x_i]| < 2(k-1-i)+1$. If we define $x'_i = x$ $(= x_{k-1})$ and $x'_{k-1} = x_i$, then the sequence $(x_1, \dots, x_{i-1}, x'_i, x_{i+1}, \dots, x_{k-1}, x'_{k-1})$ induces a covering for the node set of G_2. Therefore, $b(G_2) \leq k-1$.

If $x_t \notin P$, then there must be a component Q of G_1 for which $x_t \in Q$. Let Q' be the corresponding component of G_2 that has exactly one node less than Q. By moving x_t to cover the extra node in P' (and shifting the place of the fire sources used for covering Q without changing their order, if it is necessary), we find a covering for G_2 by t closed neighbourhoods with restricted radii. Hence, again in this case, $b(G_2) \leq k-1$.

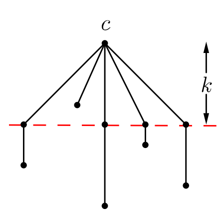
Now, assume that $L_s \setminus N_{k-1}[x]$ is empty, and $G_1 = G \setminus N_{k-1}[x]$, and $G_2 = G \setminus N_{k-1}[x']$. If $L_s \setminus N_{k-1}[x']$ is empty, then G_2 is an isometric subforest of G_1, and therefore $b(G_2) \leq b(G_1) \leq k-1$.

If $L_s \setminus N_{k-1}[x']$ is non-empty, then it means that $P' = L_s \setminus N_{k-1}[x']$ contains exactly one node. Also, we know that all the non-empty components of G_2 are subsets of the corresponding components of G_1, with exactly one less node. Assume that $(x_1, x_2, \dots, x_t)$ is an optimum burning sequence for G_1. Since, $L_s \setminus N_{k-1}[x]$ is empty and G_1 is not of order $(k-1)^2$, then there must be a non-empty component Q of G_1 for which $x_t \in Q$. By moving x_t to cover the extra node in P' (and shifting the place of the fire sources used for covering Q without changing their order, if it is necessary), we find a covering for G_2 by t closed neighbourhoods with restricted radii. Hence, again in this case we conclude that $b(G_2) \leq k-1$. □

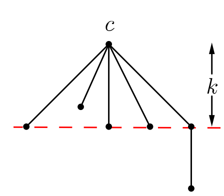
As a consequence of the above lemma we have the following result.



(A) Part (*ii*)



(B) Part (*iii*)



(C) Part (*iv*)

FIGURE 3

Lemma 20. *Let G be a spider graph with spider head c. Also, suppose that for a positive integer k and a node x in G, $G \smallsetminus N_{k-1}[x]$ is a non-empty path-forest (that is, $d(x,c) \le k-1$) with at least one component, and $b(G \smallsetminus N_{k-1}[x]) \le k-1$. If $x \neq c$, then we have one of the following possibilities:*

(i) There is a node $x' \neq c$ on the path connecting x to c for which $G \smallsetminus N_{k-1}[x']$ is of order $(k-1)^2$, and $b(G \smallsetminus N_{k-1}[x']) = k-1$.

(ii) There is a node $x' \notin \{x,c\}$ on the path connecting x to c for which $G \smallsetminus N_{k-1}[x']$ has only one component (that is a subset of L_s), and $b(G \smallsetminus N_{k-1}[x']) \le k-1$, and there is a leaf of G that is of distance $k-1$ from x'.

(iii) The graph $G \smallsetminus N_{k-1}[c]$ has at least two components, and $b(G \smallsetminus N_{k-1}[c]) \le k-1$.

(iv) There is a leaf in G that is of distance $k-1$ from c, and $G \smallsetminus N_{k-1}[c]$ has only one component, and $b(G \smallsetminus N_{k-1}[c]) \le k-1$.

Proof. If $G \smallsetminus N_{k-1}[x]$ is of order $(k-1)^2$, then we have part (i); otherwise, since the spider graph G satisfies the conditions in Lemma 19, by applying Lemma 19 for a finite number of times we derive the desired result. In Figure 3 we see a layout of the cases (ii), (iii), and (iv) stated in the lemma. □

A *perfect spider of radius r* is a spider graph G with a unique centre node c such that $d(v,c) = r$ for every leaf $v \in G$. We denote the set of all perfect spider trees of radius k with PS_k.

A *k-burning maximal spider graph*, is a spider graph with spider head c that its node set can be decomposed into a perfect spider graph $F = N_{k-1}[s] \in \mathrm{PS}_{k-1}$, where s is a node with $d(s,c) \le k-1$, and a graph $H \in \mathrm{MPF}_{k-1}$. We denote the set of all k-burning maximal spider graphs by k-BMS. By above Lemma, we can see that there are two different types of the graphs in k-BMS like G: either G is a graph for which the centre node s of the perfect spider graph in the decomposition of G is the spider head, or G is a graph such that the centre node s is not the spider head. If the latter holds, then by Lemma 20 part (ii), we conclude that the path-forest $G \smallsetminus N_{k-1}[s]$ must be a single path of order $(k-1)^2$.

Note that the path-forest that appears in decomposing a k-BMS forms a decomposed spider as described in Conjecture 11. Now, we have the following useful theorem that also shows the truth of Conjecture 11 for k-BMS trees.

Theorem 21. *If G is a k-*BMS *with spider head c, then $b(G) = k$.*

Proof. Let G be a k-BMS with spider head c. If the centre of the perfect spider in decomposing G equals c, then it implies that $b(G \smallsetminus N_{k-1}[c]) = k-1$. In this case, $G_1 = G \smallsetminus N_{k-1}[c]$ is in MPF_{k-1}. By contradiction suppose that $b(G) = t \le k-1$. Thus, by Theorem 18, there must be an optimum burning sequence for G like $(x_1, x_2, \dots, x_t)$ such that $d(x_1, c) \le t-1 \le k-2$.

If $x_1 = c$, then $G \smallsetminus N_{k-1}[c]$ is an isometric subforest of $G \smallsetminus N_{t-1}[c]$, and therefore, $b(G \smallsetminus N_{t-1}[c]) \le t-1 \le k-2$, which is a contradiction.

If $x_1 \neq c$, then note that all the arms of G are of length at least $k-1$, and since $b(G \smallsetminus N_{k-1}[c]) = k-1$, there must be at least one arm of G that is of length at least $k-1+2k-3 = 3k-4$. Thus, $G \smallsetminus N_{t-1}[c]$ must have at least two non-empty components, and therefore, by Lemma 19, we have that $b(G \smallsetminus N_{t-1}[c]) \le t-1 \le k-2$, which is a contradiction, as $G \smallsetminus N_{k-1}[c]$ is an isometric subforest of $G \smallsetminus N_{t-1}[c]$. Hence, in both cases we find a contradiction, and therefore, $b(G) \ge k$.

If the centre of the perfect spider in decomposing G is a node $s \neq c$, then as we discussed before the theorem's statement, the graph $G_1 = G \smallsetminus N_{k-1}[s]$ is a single path of order $(k-1)^2$. Let L_s be the arm of G with $s \in L_s$, and assume that v_s is the node in L_s such that $d(s, v_s) = k-1$. Also, assume that P is the path between v_s and s, and P' is the path connecting s to c excluding s.

By contradiction suppose that $b(G) = t \le k-1$. Thus, by Theorem 18, there must be an optimum burning sequence for G like $(x_1, x_2, \dots, x_t)$ such that $d(x_1, c) \le t-1 \le k-2$. We consider different possibilities for x_1 as follows:

If x_1 is in $G \setminus (L_s \setminus P')$, then clearly $G \setminus N_{k-1}[s]$ is an isometric subforest of $G \setminus N_{t-1}[c]$, and therefore, we must have $b(G_1) \leq t-1 \leq k-2$, which is a contradiction.

If x_1 is in P, then let x' be the neighbour of s on the path connecting s to c. Note that all the leaves of G, except for the leaf in L_s, are of distance $k-1$ from s. Thus, $G \setminus N_{t-1}[x_1]$ must have at least two non-empty components, and therefore, by applying Lemma 19 for a finite number of times, we have that $b(G \setminus N_{t-1}[x']) \leq t-1 \leq k-2$, which is a contradiction, as $G \setminus N_{k-1}[s]$ is an isometric subforest of $G \setminus N_{t-1}[x']$. Hence, in both cases we find a contradiction, and therefore, $b(G) \geq k$. □

By the arguments in proof of Theorem 21, we can conclude that the spider graphs in 2-MBS can be decomposed into a perfect spider $SP(s,1)$ and a single node P_1, with $s \geq 3$; that is, a spider with $s-1$ arms of length one and an arm of length two. Now, we can present an algorithm for finding the burning number of a spider tree, as follows. Note that the burning number of every spider graph is at least two. We denote the set of all perfect spider trees of radius k with t arms by PS_k^t. We denote the set of all k-burning maximal spider graphs with t arms by k-BMS^t.

Algorithm 22. *Suppose that G is a spider graph with arms $L_1, L_2, \ldots, L_t$, for a constant $t \geq 1$, such that the length of each arm L_i is denoted by l_i, and $l = l_1 \geq l_2 \geq \cdots \geq l_t$. Then we perform the following steps until $G \subseteq H$, for some $H \in k$-BMS^t where $k \geq 2$.*

Stage 1. *For the initial case $k = 2$, we put the graph $SP(t,1)$ in PS_1^t. Then we add a single node to one of the arms in $SP(t,1) \in \mathrm{PS}_1^t$, and we put the resulting graph H in 2-MBS^t.*
If $G \subseteq H$, then return $b(G) = 2$; otherwise, go to Stage 2.

Stage 2. *For $k \geq 3$, we perform the following steps:*

Stage 2.1. *For $0 \leq i \leq k-2$, we make a spider graph with $t-1$ arms of length $k-1-i$, and then we add an additional arm of length $i+k-1$ to it. We call the resulting spider (with t arms) by H_i and we put it in PS_{k-1}^t.*

Stage 2.2. *For $1 \leq s \leq k-1$, and each $F \in \mathrm{MPF}_{k-1}^s$ (generated by Algorithm 15 for the graph $G' = (l_1, l_2, \ldots, l_t)$), we join an end point of each component of F to a distinct leaf of $H_0 \in \mathrm{PS}_{k-1}^t$, and we call the resulting graph by F'. Then we add F' to k-MBS^t.*
If $G \subseteq F'$, then stop and return $b(G) = k$.

Stage 2.3. *For $1 \leq i \leq k-2$, we join the end point of longest arm of H_i to a path of order $(k-1)^2$ in MPF_{k-1}^1, and we call the resulting graph by H_i'. Then we add H_i' to k-MBS^t.*
If $G \subseteq H_i'$, then stop and return $b(G) = k$.

If Algorithm 22 stops at $i = k$, then it means that G is a subgraph of a graph in k-MBS^t. By Theorem 21, we know that the burning number of a graph in i-MBS^t equals i. Hence, by Corollary 5 from [20], we conclude that $b(G) = k$. We have the following theorem about the complexity of Algorithm 22.

Theorem 23. *Algorithm 22 finds the burning number of G in time $O(t^2 l^{t+2})$, that is polynomial for fixed t.*

Proof. Given the graph G, suppose that for some $k \geq t$, Algorithm 22 stops by recognizing G as a subgraph of a graph in MBS_k^t; that is, $b(G) = k$. Note that the length of each arm in G is bounded above by l. In Algorithm 22, we first generate all the perfect spider graphs of radius i with t arms, for $1 \leq i \leq k$. Then at Stage 2.2, we need to perform Algorithm 15 for the graph $(l_1, l_2, \ldots, l_t)$ which satisfies all the conditions in Theorem 16. Hence, we perform at most $O(tl^t)$ steps to find all the maximal path-forests generated by Algorithm 15 at Stage 2.2.

On the other hand, we know that $k = b(G) \leq \mathrm{radius}(G) + 1 \leq l + 1 = O(l)$. Thus, $\frac{k(k+1)}{2} = O(l^2)$. Note that the number of the perfect spider graphs that we generate in Algorithm 22 for each

$1 \le i \le k$ equals i. Therefore, the total number of the graphs that we create and consider by Algorithm 22 is asymptotically of order

$$\sum_{i=1}^{k} kO(tl^t) = \frac{k(k+1)}{2} O(tl^t) = O(tl^{t+2}).$$

Finally, note that each time that we add a new spider graph F to MBS_k^t, for $k \ge 2$, we compare G with F. We can simply do this comparison by comparing the lengths of the arms between G and F. Since, G and F both have t arms, then the total number of the steps that we perform in Algorithm 22 is bounded above by $O(t^2 l^{t+2})$. Since t is a fixed constant in terms of l, then Algorithm 22 is polynomial time in the length of the input. □

5. Approximation

We proceed to the description of our approximation algorithm, which is inspired by the approximation algorithm for the k-center problem due to Hochbaum and Shmoys [12]. The following procedure is a central ingredient in the algorithm. For a pair (G, k), where G is a graph and k is a positive integer, this procedure returns a sequence $(x_1, \ldots, x_k)$ of nodes of G.

Procedure 24. *Given a pair (G, k), where G is a graph and k is a positive integer, perform the following steps.*

Stage 1. *Choose an arbitrary node x_1 in G.*

Stage 2. *For $i = 2$ to k, select x_i as a node in G that maximizes* $\min\left\{\frac{d(u,x_j)}{k-j+1} : j \in [i-1]\right\}$.

Return $(x_1, \ldots, x_k)$.

For a given graph G, the following algorithm now applies Procedure 24 to pairs (G, k) starting with $k = 1$ and repeatedly increasing k until Procedure 24 returns a sequence $(x_1, \ldots, x_k)$ that satisfies (1).

Algorithm 25. *Given a graph G we perform the following steps.*

Stage 1. *For $k \ge 1$, perform Procedure 24 and set $S = (x_1, \ldots, x_k)$, where $(x_1, \ldots, x_k)$ is the output of Procedure 24.*

Stage 2. *If S satisfies (1), then stop and return $S = (x_1, \ldots, x_k)$; otherwise go to Stage 1 for $k + 1$.*

Note that if G has order k, then Procedure 24 applied to (G, k) returns a sequence containing all k nodes of G, which clearly satisfies (1). Therefore, Algorithm 25 terminates after at most $|V(G)|$ applications of Procedure 24. The correctness of Algorithm 25 is obvious. Note that the output of Algorithm 25 can easily be transformed in polynomial time ($O(n^3)$) into a burning sequence by applying the construction used in the proof of Lemma 2.

Lemma 26. *Let G be a graph and let k be a positive integer. Let Procedure 24 return $(x_1, \ldots, x_k)$ when applied to (G, k). If $(x_1, \ldots, x_k)$ does not satisfy (1), then $b(G) \ge \left\lfloor \frac{k}{3} \right\rfloor + 1$.*

Proof. Since $(x_1, \ldots, x_k)$ does not satisfy (1), there is some node u^* of G such that $d(u^*, x_j) \ge k-j+1$ for every $j \in [k]$, which implies $\min\left\{\frac{d(u^*,x_j)}{k-j+1} : j \in [k]\right\} \ge 1$. By the selection rule within Procedure 24, we obtain that

$$\begin{aligned}\min\left\{\frac{d(x_i,x_j)}{k-j+1}:j\in[i-1]\right\} &\geq \min\left\{\frac{d(u^*,x_j)}{k-j+1}:j\in[i-1]\right\}\\ &\geq \min\left\{\frac{d(u^*,x_j)}{k-j+1}:j\in[k]\right\}\\ &\geq 1,\end{aligned}$$

for every i in $[k]\setminus[1]$. For every two indices i and j in $[k]$ with $j<i$, this implies $d(x_i,x_j)\geq k-j+1>k-j$. Since $j<i$ implies $k-j\geq\left\lceil\frac{k-j}{2}\right\rceil+\left\lceil\frac{k-i}{2}\right\rceil$, we obtain that the k sets

$$N_{\lceil\frac{k-1}{2}\rceil}[x_1],N_{\lceil\frac{k-2}{2}\rceil}[x_2],N_{\lceil\frac{k-3}{2}\rceil}[x_3],\ldots,N_0[x_k]$$

are pairwise disjoint.

Now by contradiction, suppose that $b(G)=k'\leq\left\lfloor\frac{k}{3}\right\rfloor$. Let $(y_1,\ldots,y_{k'})$ be a burning sequence of length k'. Since

$$\left\lceil\frac{k-\left\lfloor\frac{k}{3}\right\rfloor-1}{2}\right\rceil\geq\left\lfloor\frac{k}{3}\right\rfloor-1\geq k'-1,$$

each of the $\left\lfloor\frac{k}{3}\right\rfloor+1$ sets

$$N_{\lceil\frac{k-1}{2}\rceil}[x_1],N_{\lceil\frac{k-2}{2}\rceil}[x_2],N_{\lceil\frac{k-3}{2}\rceil}[x_3],\ldots,N_{\left\lceil\frac{k-\lfloor\frac{k}{3}\rfloor-1}{2}\right\rceil}\left[x_{\lfloor\frac{k}{3}\rfloor+1}\right]$$

contains an element of the sequence $(y_1,\ldots,y_{k'})$. Since these sets are all disjoint, we obtain the contradiction that $(y_1,\ldots,y_{k'})$ contains more than k' distinct elements. ☐

We finish with the following theorem.

Theorem 27. *Let G be a graph. If Algorithm 25 returns a sequence of length k, then $b(G)\geq\frac{k}{3}$, that is, Algorithm 25 is a polynomial time approximation algorithm with approximation factor* 3.

Proof. If Algorithm 25 returns a sequence of length k, then Procedure 24 applied to $(G,k-1)$ returned a sequence of length $k-1$ that did not satisfy (1). By Lemma 26, this implies $b(G)\geq\left\lfloor\frac{k-1}{3}\right\rfloor+1\geq\frac{k}{3}$. Obviously, Algorithm 25 can be implemented to run in polynomial time. ☐

As you can see in Theorem 15 from [2], there is a polynomial time characterization for binary trees with radius r that has burning number $r+1$. Also, by Lemma 12, we can easily see that for a spider graph G of radius r, $b(G)=r+1$ if and only if $SP(r,r)$ is a subtree of G. Thus, we are motivated to make the following conjecture.

Conjecture 28. *For a tree T of radius r, we can recognize in polynomial time weather or not $b(T)=r+1$.*

Acknowledgment This paper comes out of a collaborative work based on [3] and [4].

References

[1] J. Balogh, B. Bollobás, R. Morris, Graph bootstrap percolation, *Random Structures & Algorithms* **41** (2012) 413-440.
[2] S. Bessy, A. Bonato, J. Janssen, D. Rautenbach, E. Roshanbin, Bounds on the Burning Number, submitted 2016.
[3] S. Bessy, D. Rautenbach, Bounds, Approximation, and Hardness for the Burning Number, arXiv:1511.06023.
[4] A. Bonato, J. Janssen, E. Roshanbin, Burning a Graph is Hard, arXiv:1511.06774.
[5] A. Bonato, J. Janssen, E. Roshanbin, How to burn a graph, Internet Mathematics 1-2 (2016) 85-100.

[6] A. Bonato, R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, Providence, Rhode Island, 2011.

[7] P. Domingos, M. Richardson, Mining the network value of customers, In: *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD)*, 2001.

[8] S. Finbow, A. King, G. MacGillivray, R. Rizzi, The firefighter problem for graphs of maximum degree three, *Discrete Mathematics* **307** (2007) 2094-2105.

[9] S. Finbow, G. MacGillivray, The Firefighter problem: a survey of results, directions and questions, *Australasian Journal of Combinatorics* **43** (2009) 57–77.

[10] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of* ***NP****-Completeness*, W.H. Freeman, 1979.

[11] T.W. Haynes, S.T. Hedetniemi, P.J. Slater, *Fundamentals of Domination in Graphs* Marcel Dekker, New York, 1998.

[12] D.S. Hochbaum, D.B. Shmoys, A best possible heuristic for the k-center problem, Math. Oper. Res. 10 (1985) 180-184.

[13] H. Hulett, T.G. Will, G.J. Woeginger, Multigraph realizations of degree sequences: Maximization is easy, minimization is hard, *Operations Research Letters* **36** (2008) 594–596.

[14] D. Kempe, J. Kleinberg, E. Tardos, Maximizing the spread of influence through a social network, In: *Proceedings of the 9th International Conference on Knowledge scovery and Data Mining (KDD)*, 2003.

[15] D. Kempe, J. Kleinberg, E. Tardos, Influential nodes in a diffusion model for social networks, In: *Proceedings 32nd International Colloquium on Automata, Languages and Programming(ICALP)*, 2005.

[16] A.D.I. Kramer, J.E. Guillory, J.T. Hancock, Experimental evidence of massive-scale emotional contagion through social networks, *Proceedings of the National Academy of Sciences* **111** (2014) 8788-8790.

[17] D. Mitsche, P. Pralat, E. Roshanbin, Burning graphs - a probabilistic perspective, arXiv:1505.03052.

[18] E. Mossel, S. Roch, On the submodularity of influence in social networks, In: *Proceedings of 39th Annual ACM Symposium on Theory of Computing(STOC)*, 2007.

[19] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, In *Proceedings of the 8th International Conference on Knowledge scovery and Data Mining (KDD)*, 2002.

[20] E. Roshanbin, Burning a graph as a model of social contagion, *PhD Thesis*, Dalhousie University, 2016.

[21] D.B. West, *Introduction to Graph Theory, 2nd edition*, Prentice Hall, 2001.

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM),, Montpellier, France
E-mail address: stephane.bessy@lirmm.fr

Department of Mathematics, Ryerson University, Toronto, ON, Canada, M5B 2K3
E-mail address: abonato@ryerson.ca

Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada, B3H 3J5
E-mail address: jeannette.janssen@dal.ca

Institute of Optimization and Operations Research, Ulm University,, Ulm, Germany
E-mail address: dieter.rautenbach@uni-ulm.de

Department of Mathematics and Statistics, Dalhousie University, Halifax, NS, Canada, B3H 3J5
E-mail address: e.roshanbin@dal.ca