Theoretical Computer Science 613 (2016) 38–50

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

The Firefighter problem on graph classes ☆

Fedor V. Fomin [a], Pinar Heggernes [a], Erik Jan van Leeuwen [b,*]

[a] *Department of Informatics, University of Bergen, PB 7803, N-5020 Bergen, Norway*
[b] *Max-Planck Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany*

ARTICLE INFO

Article history:
Received 5 August 2014
Received in revised form 4 November 2015
Accepted 18 November 2015
Available online 23 November 2015
Communicated by G. Ausiello

Keywords:
Firefighter problem
Algorithms
Interval graphs
Permutation graphs
Split graphs
P_k-free graphs

ABSTRACT

The Firefighter problem aims to save as many vertices of a graph as possible from a fire that starts in a vertex and spreads through the graph. At every time step a new firefighter may be placed on some vertex, and then the fire advances to every vertex that is not protected by a firefighter and has a neighbor on fire. The problem is notoriously hard: it is NP-hard even when the input graph is a bipartite graph or a tree of maximum degree 3, it is NP-hard to approximate within $n^{1-\epsilon}$ for any $\epsilon > 0$, and it is $W[1]$-hard when parameterized by the number of saved vertices. We show that Firefighter can be solved in polynomial time on interval graphs, split graphs, permutation graphs, and P_k-free graphs for fixed k. To complement these results, we show that the problem remains NP-hard on unit disk graphs.

1. Introduction

In the Firefighting game on a graph, a fire starts in a given vertex s at time $t = 0$. At each subsequent time step $t \geq 1$, first, a firefighter may be placed on a vertex which is not yet touched by the fire, which makes that vertex *protected*, meaning that it cannot catch fire for the rest of the game. Second, the fire spreads to every neighbor of the burning vertices that is not protected by a firefighter. After this, a new time step starts. A vertex that has been infected by the fire continues to burn until the end of the game. If, after some time step, the burning vertices are separated from the rest of the graph by the protected vertices, then the fire is *contained* and the game ends. The vertices that have not been touched by the fire at the end of the game are referred to as *saved* (in particular, protected vertices are saved). The Firefighter problem takes as input a graph G on n vertices and a vertex s of G, and the goal is to play the Firefighting game on (G, s) such that the number of saved vertices is maximized.

The Firefighter problem was introduced in 1995 and intended to capture also other important applications, like immunizing a population against a virus [2]. The problem is notoriously difficult. It is NP-hard even on bipartite graphs [3] and on trees of maximum degree 3 [4]. It is NP-hard to approximate the Firefighter problem within $n^{1-\epsilon}$ for any $\epsilon > 0$ [5]. From a parameterized point of view, the problem is $W[1]$-hard when parameterized by the natural parameter of the number of saved vertices [6–8].

The difficulty of the problem raises the question of tractability on restricted inputs. Although the problem and its variants are well studied [9], the only polynomial-time algorithms known for Firefighter so far are on graphs of maximum degree

☆ This work has been supported by the Research Council of Norway and by ERC StG project PAAl no. 259515. A preliminary version appeared at FUN 2012 [1].

* Corresponding author.
E-mail addresses: fedor.fomin@ii.uib.no (F.V. Fomin), pinar.heggernes@ii.uib.no (P. Heggernes), erikjan@mpi-inf.mpg.de (E.J. van Leeuwen).

http://dx.doi.org/10.1016/j.tcs.2015.11.024
0304-3975/© 2015 Elsevier B.V. All rights reserved.

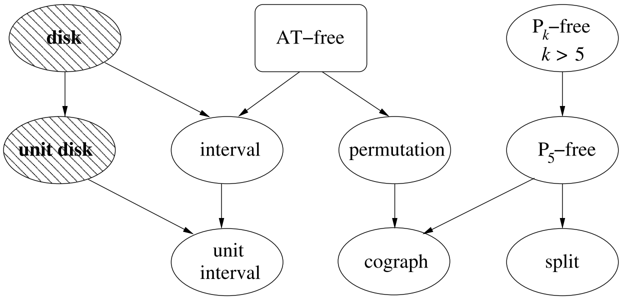

Fig. 1. The graph classes mentioned in this paper, where → represents the ⊃ relation. Due to our results, Firefighter can be solved in polynomial time on graph classes that are represented with white ellipses, and it is NP-hard on graph classes that are represented with filled ellipses. The complexity of the problem is open on AT-free graphs (marked with a rounded rectangle).

3 when the fire starts at a vertex of degree at most 2 [4], and on so-called P-trees [3]. Even with respect to approximation and fixed-parameter tractability, the only positive results known so far are on trees and on graphs of bounded treewidth [10]. On arbitrary trees, the problem is fixed-parameter tractable [11,6–8], and a simple 2-approximation algorithm [12] along with a more involved $(1 - 1/e)$-approximation algorithm [11] exists. A recent survey of combinatorial and algorithmic results on the Firefighter problem has been given by Finbow and MacGillivray [13].

In this paper we show that Firefighter can be solved in polynomial time on several well-known graph classes, giving the first polynomial-time algorithms for a variety of graphs that are not close to trees. Our main results are polynomial-time algorithms for Firefighter on interval graphs and on permutation graphs. We also obtain polynomial-time algorithms on P_k-free graphs for every fixed k, and linear-time algorithms on split graphs and on cographs. We complement these positive results by showing that Firefighter remains NP-hard on unit disk graphs. Note that all of these graph classes have unbounded treewidth, and all of them, except cographs, have unbounded clique-width. Our results are summarized in Fig. 1.

2. Preliminaries

Let (G, s) be an instance of the Firefighter problem. If G is disconnected, then all connected components except the one that contains s are automatically saved. Hence, we can assume G to be connected. Throughout the paper we consider simple, undirected, unweighted, connected input graphs.

Given a graph G, its set of vertices is denoted by $V(G)$ and its set of edges by $E(G)$. We adhere to the convention that $n = |V(G)|$ and $m = |E(G)|$. Given a set $U \subseteq V(G)$, the subgraph of G induced by U is denoted by $G[U]$. The set of neighbors of a vertex v is denoted by $N_G(v)$, and $N_G[v] = N_G(v) \cup \{v\}$. For a subset $U \subseteq V(G)$, $N_G(U) = \cup_{u \in U} N(u) \setminus U$ and $N_G[U] = N_G(U) \cup U$. We drop the subscript from N_G whenever the graph is clear from the context. Given two non-adjacent vertices u and v in G, a set $S \subseteq V(G)$ is a *minimal* u, v*-separator* if u and v appear in different connected components of $G[V(G) \setminus S]$ and no proper subset of S has this property. A *minimal separator* is a set $S \subseteq V(G)$ that is a minimal u, v-separator for some pair u, v in G.

2.1. Graph classes

Since we study Firefighter when the input graph belongs to various graph classes, we now give their definitions. We list several well-known properties of these graph classes without references; all details can be found in one of several excellent books on graph classes, e.g. [14,15]. For convenience, we show in Fig. 1 how all the graph classes that we study in this paper are related to each other with respect to the subset relation. Given an integer k, we denote by P_k a path on k vertices and exactly $k - 1$ edges. A graph is P_k*-free* if it does not contain P_k as an induced subgraph. An *asteroidal triple (AT)* in a graph G is a triple of pairwise non-adjacent vertices, such that there is a path between any two of them that does not contain a neighbor of the third. A graph is *AT-free* if no triple of its vertices forms an AT.

A graph is an *interval graph* if there is a bijection between its vertices and a set of intervals of the real line such that two vertices are adjacent if and only if their intervals overlap. Such a set of intervals is called an *interval representation* of the corresponding interval graph. An interval graph is a *unit interval graph* if it has an interval representation where all the intervals have the same length. A graph is a *permutation graph* if it can be obtained from a permutation π of the integers between 1 and n in the following way: vertex i and vertex j are adjacent if and only if $i < j$ and j appears before i in π. Interval graphs and permutation graphs are unrelated (using the subset relation), but they are both AT-free.

A graph is a *split graph* if its vertices can be partitioned into a clique and an independent set. It is easy to see that split graphs are P_5-free. Split graphs are unrelated (using the subset relation) to interval and permutation graphs. For the definition of cographs, note that the *disjoint union* of two vertex-disjoint graphs is simply the graph that consists of the union of the vertex sets and the union of edge sets of the two graphs, whereas a *complete join* of two vertex-disjoint graphs G_1 and G_2 is the graph that is obtained from the disjoint union of G_1 and G_2 by adding an edge between every vertex of G_1 and every vertex of G_2. *Cographs* are defined recursively as follows. A single vertex is a cograph; the disjoint union of two cographs is a cograph; the complete join of two cographs is a cograph. This defines a tree representation of a cograph called the *cotree*. Cographs are exactly the class of P_4-free graphs. Cographs form a subset of permutation graphs, but they are unrelated (using the subset relation) to split and interval graphs.

A graph is a *disk graph* if there is a bijection between its vertices and a set of circles drawn on the plane such that two vertices are adjacent if and only if their circles overlap. Such a set of circles, or disks, is called a *disk representation* of the corresponding disk graph. A disk graph is a *unit disk graph* if it has a disk representation where all disks have the same diameter, which we assume to be 1. Disk graphs form a superset of interval graphs, and unit disk graphs form a superset of unit interval graphs.

2.2. Firefighter strategies

Let (G, s) be an instance of the Firefighter problem. A *strategy* for (G, s) is an ordered set of vertices, representing the placement of the firefighters in each time step of the Firefighting game on (G, s). A strategy is said to be *minimal* if no subset of the vertices protected in the strategy yields a strategy that saves the same set of vertices, and *optimal* if it saves the maximum possible number of vertices and is minimal among all such strategies. An ordered set of vertices $S = \{v_1, \ldots, v_k\}$ is a *valid strategy* for (G, s) if during the Firefighting game on (G, s), vertex v_i is not burning at the start of time step i, and the game ends at time step k.

Lemma 1. *Given a graph G on n vertices and m edges, a vertex s of G, and an ordered set S of vertices of G, we can test whether S is a valid strategy for the Firefighting game on (G, s) and count the number of vertices saved by S in $O(n + m)$ time.*

Proof. Suppose that $S = \{v_1, \ldots, v_k\}$, in this order. For any integer $i = 1, \ldots, k$, define $S_i = \{v_1, \ldots, v_i\}$. To check whether S is a valid strategy, we play the Firefighting game starting at s. We employ a (slightly modified) breadth-first search starting in s to verify, for each $i = 1, \ldots, k$, that v_i is not in the set $R_i(s)$ of vertices that are burning at the start of time step i, where $R_1(s) = \{s\}$ and $R_i(s) = N_{G-S_{i-1}}[R_{i-1}(s)]$ when $i > 1$. During the same procedure, we can verify that the game ends at time step k. Assuming this can indeed be verified, we can compute the number of vertices saved by S as n minus the number of vertices reachable from s in $G - S$, which can be computed using a breadth-first search from s in $G - S$. □

2.3. The firefighter reserve

For some of our algorithms, we provide a different but equivalent definition of the Firefighter problem (see also [7,8]). In the Firefighting-with-Reserve game on a graph, a fire starts in a given vertex s at time $t = 0$ and the firefighters have a reserve of one firefighter. At each subsequent time step $t \geq 1$, first, the firefighters can (permanently) deploy any number of its firefighter reserves to vertices of the graph that are not yet on fire in order to protect them, and the reserve decreases accordingly (however, it cannot get below zero). Second, the fire spreads to every neighbor of the burning vertices that is not protected by a firefighter. Third, a firefighter is added to the reserve and a new time step starts. We define the end of the game and the notion of protected and saved vertices as in the Firefighting game. Then the Firefighter Reserve Deployment problem takes as input a graph G on n vertices and a vertex s of G, and the goal is to play the Firefighting-with-Reserve game on (G, s) such that the number of saved vertices is maximized.

Let (G, s) be an instance of the Firefighter Reserve Deployment problem. A *strategy* is an ordered collection $F_1, \ldots, F_k$ of vertex subsets, representing the placement of the firefighters in each time step of the Firefighting-with-Reserve game on (G, s) such that firefighters are deployed on the vertices of F_i in step i, where F_i might be empty for several i. Again, a strategy is said to be *minimal* if no subset of the vertices protected in the strategy yields a strategy that saves the same set of vertices, and *optimal* if it saves the maximum possible number of vertices and is minimal among all such strategies.

Lemma 2. *The* Firefighter *and the* Firefighter Reserve Deployment *problem are equivalent.*

Proof. Consider a strategy $v_1, \ldots, v_k$ for the Firefighting game and look at the Firefighting-with-Reserve. At time step t, if the fire reaches vertices $F_t \subseteq \{v_1, \ldots, v_k\}$ at time step $t + 1$ in $G - \bigcup_{i=1}^{t-1} F_i$, deploy the firefighters in F_t at time t. Because $v_1, \ldots, v_k$ is a valid strategy, this must also be a valid strategy. Moreover, it saves exactly the same set of vertices.

Consider a strategy $F_1, \ldots, F_k$ for the Firefighting-with-Reserve game. Consider any ordering $v_1, \ldots, v_{k'}$ of the vertices in $F_1, \ldots, F_k$ such that $v_a \in F_i$, $v_b \in F_j$ for $i < j$ implies $a < b$. Any such ordering is a valid strategy for the Firefighting game, saving exactly the same set of vertices. □

Consider the two transformations in the above proof. If we apply the second transformation on a strategy for the Firefighting-with-Reserve game, and then the first, we obtain a strategy for the Firefighting-with-Reserve game with the following property.

Corollary 3. *Let (G, s) be an instance of* Firefighter Reserve Deployment. *There exists an optimal strategy for the Firefighting-with-Reserve game on (G, s) that only protects vertices that neighbor a vertex that is on fire.*

When it is more convenient algorithmically, we will solve Firefighter Reserve Deployment instead of Firefighter; Lemma 2 shows that this is an equivalent problem. Moreover, Corollary 3 gives a useful property when solving Firefighter Reserve Deployment.

If a vertex is ever infected by the fire, then we refer to it as *burning* or *burned*. The vertices that are not touched by the fire are referred to as *unburned* vertices. Recall that *saved* vertices are all the unburned vertices when the Firefighting (with-Reserve) game is over, including the protected vertices. We refer to the saved vertices that are not protected as *rescued*. The *last line of defense* of a strategy is the set $N(R)$, where R is the set of vertices rescued by the strategy.

3. P_k-free graphs

In this section we show that Firefighter can be solved in time $O(n^k)$ on P_k-free graphs. We also observe that it is not likely that Firefighter can be solved in time $f(k)\,n^{O(1)}$ on P_k-free graphs, due to Theorem 6 below.

To prove the algorithmic results of this section, we rely on the following auxiliary lemma.

Lemma 4. *Let (G, s) be an instance of* Firefighter*, and let ℓ be the number of vertices on a longest induced path in G starting in s. Then no optimal strategy can protect more than $\ell - 1$ vertices.*

Proof. Suppose that vertices $v_1, \ldots, v_t$ are protected by some optimal strategy in that order, and that t is maximum among all optimal strategies. Since the strategy is optimal, there is an induced path P between s and v_t such that all vertices on P, except v_t, burn. Let P be a shortest path with this property. Then P contains at least $t + 1$ vertices, or v_t would burn before we could protect it. It follows from the premises of the lemma that $t \leq \ell - 1$. □

Theorem 5. Firefighter *can be solved in time $O(n^{k-2}(n + m)) = O(n^k)$ on P_k-free graphs.*

Proof. Let (G, s) be an instance of Firefighter such that G is a P_k-free graph on n vertices and m edges. The longest induced path in G has at most $k-1$ vertices, as G is P_k-free. Consequently, by Lemma 4, any optimal strategy on G protects at most $k - 2$ vertices. This suggests the following algorithm. We enumerate all ordered subsets $S \subseteq V(G)$ of size at most $k - 2$ in $O(n^{k-2})$ time. For each such S, we use Lemma 1 to verify that S is a valid strategy and to count the number of saved vertices in $O(n + m)$ time. Finally, we return the valid strategy S that saves the largest number of vertices. □

In terms of complexity classes FPT and XP (see e.g. [16] for their definitions), Theorem 5 shows that the Firefighter problem is in XP when parameterized by the length of the longest induced path in the graph. The result of Theorem 5 is in fact tight in the sense that we cannot expect to solve Firefighter in time $f(k)\,n^{O(1)}$ on P_k-free graphs, as stated in the next theorem, which was proved by Cygan et al. [7,8]. The statement of the theorem is different in [7,8]; however, the statement below is implicit. The reduction of [7,8] is from k-Clique, and yields a bipartite graph. Upon inspection, the length of the longest induced path in this construction is $\max\{k + 1, 3\}$.

Theorem 6. *(See [7,8].)* Firefighter *is $W[1]$-hard when parameterized by the length of a longest induced path in the input graph, even if the graph is bipartite.*

Since cographs are P_4-free and split graphs are P_5-free, Theorem 5 immediately implies algorithms for Firefighter on these graph classes with running times $O(n^4)$ and $O(n^5)$, respectively. We next show that the problem can in fact be solved much faster on these graph classes.

Theorem 7. Firefighter *can be solved in time $O(n + m)$ on cographs.*

Proof. Let (G, s) be an instance of Firefighter such that G is a connected cograph on n vertices and m edges. Find a cotree representation of G; this takes $O(n + m)$ time [17]. Since G is connected, it cannot be the disjoint union of two cographs. Therefore, let G_1 and G_2 be the cographs which G is the complete join of. Assume, without loss of generality, that s is in G_1. If s has no non-neighbors in G, then we can save exactly one vertex, namely the one that we protect in the first (and last) time step. If s has non-neighbors, then as cographs are P_4-free, $N[N[s]] = V(G)$ and $N[s] \neq V(G)$. Moreover, using Lemma 4, any optimal strategy thus protects exactly two vertices. A strategy that protects two non-neighbors of s saves exactly two vertices, namely the protected vertices, as all vertices of G_2 are a neighbor of s and all vertices of G_1 are a neighbor of all vertices of G_2. However, a strategy that protects a neighbor and a non-neighbor of s saves at least two vertices, and is thus preferred. If $|V(G_2)| > 1$, then every non-protected vertex is burned after the second time step. However, if $|V(G_2)| = 1$, then protecting the single vertex of G_2 could save some vertices in G_1, as G_1 might be a disjoint union of two cographs G'_1, G''_1, in which case (assuming without loss of generality that G'_1 contains s) we save (at least) the vertices in G''_1. Hence, a strategy that chooses an arbitrary neighbor of s in G_2 and an arbitrary non-neighbor of s (in G_1) is always optimal. The total running time of the algorithm is $O(n + m)$. □

Theorem 8. Firefighter *can be solved in time $O(n+m)$ on split graphs.*

Proof. Let (G, s) be an instance of Firefighter such that G is a connected split graph on n vertices and m edges, with $V(G) = I \cup C$ for an independent set I and a clique C. We can find I and C in $O(n+m)$ time [18]. We may assume that every vertex of C has a neighbor in I; otherwise, we take a vertex of C that has no neighbors in I, remove it from C, and add it to I.

We consider two cases, depending on whether $s \in C$ or $s \in I$. We first address the case that $s \in C$. Consider any optimal strategy. If the first firefighter is placed on a vertex of I, then the fire spreads to (at least) every vertex of C. In the next time step, we can maybe protect one more vertex of I before the fire spreads to all other vertices of I, meaning that we can save at most two vertices. Hence, if there is an optimal strategy that places its first firefighter on a vertex of I, then there is one that first protects a neighbor of s in I, and then a vertex of $I \setminus N(s)$ if $I \setminus N(s) \neq \emptyset$. We can find such a strategy in $O(n+m)$ time.

If the first firefighter is placed on a vertex v of C, then the fire spreads to (at least) every vertex of $C \setminus \{v\}$. If v has neighbors of degree 1, then these are saved. In the next time step, we can maybe protect one more vertex w of I that is not adjacent to s, before the fire spreads to all other vertices of I. Hence, an optimal strategy that places its first firefighter on a vertex of C finds a vertex $v \neq s$ of C with the highest number of neighbors of degree 1 and protects v. It then protects a vertex w of I that is not a neighbor of s nor a neighbor of v of degree 1 if w exists. Vertices v and w, and thus an optimal strategy of this kind, can be found in $O(n+m)$ time.

Now consider the case that $s \in I$. If s has degree 1, then it is optimal to protect the single neighbor of s in C, and we save all vertices in the graph except s. So assume that s has degree at least 2. Since split graphs are P_5-free, it follows from Lemma 4 that we can protect at most three vertices.

Suppose there is an optimal strategy that only protects vertices from I. Then the fire first spreads from s to every vertex of $N(s) \subseteq C$, then to (at least) every vertex of the rest of C, and then to all unprotected vertices of I. Hence, any such strategy can save at most three vertices. Hence, if there is an optimal strategy that only protects vertices from I, then in the first two time steps it protects a neighbor in $I \setminus \{s\}$ of a vertex in $N(s)$ or any other vertex of $I \setminus \{s\}$ otherwise, and then a vertex of I that is not a neighbor of a vertex in $N(s)$ if it exists. We can find such an optimal strategy in $O(n+m)$ time.

Suppose there is an optimal strategy that protects two or more vertices from C. Since s has degree at least 2, the fire first spreads to at least one vertex of C, and then to all unprotected vertices of C. This means that such an optimal strategy protects exactly two vertices u, v from C, and that these are the first two vertices that are protected by the strategy. Moreover, no vertex of $I \setminus \{s\}$ catches fire before u and v are protected, because at the first time step only the vertices of $N(s) \setminus \{u\} \subseteq C$ catch fire. So for $X \subseteq C$, let $U(X)$ denote the set of vertices in I whose neighborhoods are subsets of X. Following the above reasoning, all vertices of $U(\{u, v\})$ are saved, plus at most one vertex of I. Hence, the optimal strategy has chosen u, v such that $|U(\{u, v\})|$ is maximum over all $X \subseteq C$ with $|X| = 2$ and $|X \cap N(s)| \leq 1$. We first compute $|U(c)|$ for each $c \in C$, which takes $O(n+m)$ time. Note that $|U(\{c_1, c_2\})| \geq |U(c_1)| + |U(c_2)|$. However, as $|I| \leq n$, there are at most n pairs (c_1, c_2) for which $|U(\{c_1, c_2\})| > |U(c_1)| + |U(c_2)|$, namely those for which there is an $i \in I$ with $N(i) = \{c_1, c_2\}$. Call $|U(\{c_1, c_2\})| - |U(c_1)| - |U(c_2)|$ the *pair-bonus* of (c_1, c_2). We can find all pairs of vertices with a nonzero pair-bonus, as well as the pair-bonus of these pairs, in $O(n+m)$ time as follows. Create a bucket for each vertex of C. For each degree-2 vertex $i \in I$, adjacent to say c_1, c_2, add c_1 to the bucket of c_2, and vice versa. Then for any fixed $c \in C$, we count how often each c' in c's bucket occurs in the bucket, which gives the pair-bonuses in $O(n+m)$ time. Now find the pair (c_1, c_2) with a nonzero pair-bonus for which $a = |U(\{c_1, c_2\})|$ is maximal, and the pair (c'_1, c'_2) for which $b = |U(c'_1)| + |U(c'_2)|$ is maximal. Suppose that (u, v) is the pair attaining $\max\{a, b\}$. Then there is an optimal strategy that chooses u, v, and possibly one more vertex of I. Such an optimal strategy can be found in $O(n+m)$ time.

Finally, suppose that each optimal strategy protects exactly one vertex of C. Suppose that such a strategy first protects two vertices of I. Then the fire first spreads from s to every vertex of $N(s) \subseteq C$ and then to every vertex of the rest of C, contradicting that a vertex from C is protected. Hence, any optimal strategy picks at least one vertex of C among its first two vertices.

Suppose that any optimal strategy first protects a vertex w of I and then a vertex v of C. Note that all vertices of $N(s) \subseteq C$ are burning at the end of time step 1 and that all vertices of $C \setminus \{v\}$ are burning at the end of time step 2. If the strategy protects another vertex of I after v, then it is not a neighbor of a vertex in $N(s)$. Hence, the strategy that first protects v and then w (and then possibly another vertex of I) saves at least as many vertices, a contradiction. Therefore, there is an optimal strategy that first protects a vertex $v \in C$.

Suppose that every strategy that first protects a vertex $v \in C$ also protects at least two vertices of I, say u and w, such that u is protected before w. Let $x \in C$ be a neighbor of w. At the first time step, all vertices of $N(s) \setminus \{v\} \subseteq C$ catch fire, then (at least) every vertex of the rest of $C \setminus \{v\}$, and finally every vertex of the rest of $I \setminus \{u, w\}$ except degree-1 neighbors of v in $I \setminus \{s\}$. The vertices that are saved are u, v, w, and the degree-1 neighbors of v in $I \setminus \{s\}$. Note that $x \notin N(s) \setminus \{v\}$, or w would catch fire in the second time step, contradicting that it can be protected in the third time step. Hence, we can replace u by x and obtain another valid strategy. Moreover, this strategy would save (at least) v, w, x, and all degree-1 neighbors of v in $I \setminus \{s\}$, which contradicts our assumptions. Hence, there is an optimal strategy that protects two vertices: (first) a vertex v of C and (then) a vertex w of I. Moreover, this strategy saves exactly v, w, and all degree-1 neighbors of v in $I \setminus \{s\}$. But then the vertex of C that we want to protect is one with the highest number of degree-1 neighbors in $I \setminus \{s\}$. Therefore, such an optimal strategy can be found in $O(n+m)$ time.

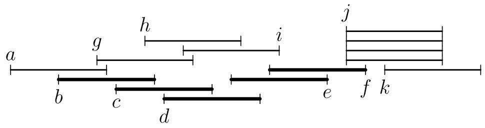

Fig. 2. In this (unit) interval graph, the thicker lines represent ten intervals with the same endpoints. The fire starts in a. The four vertices of j and vertex k both are minimal separators (imagine that the graph continues after k) that can be protected before the fire reaches them. However, if we choose to protect the vertices of j, we can protect at most one of the vertices that come before, namely one of $\{g, i\}$. If we choose to protect k, then we can also protect g, h, i, and three vertices of j. The latter strategy saves more vertices (7) than the former (6).

In each of the cases that are distinguished above, we can find an optimal strategy for the case in $O(n+m)$ time. By returning the best strategy among these, we can indeed find an optimal strategy in linear time. □

4. Interval graphs

We have seen that the absence of long induced paths is helpful in achieving tractability for Firefighter. Now we will see that long induced paths can be handled if the graph has some linear structure. In particular, we will show that Firefighter can be solved in polynomial time on interval graphs.

Interval graphs can be recognized in linear time and an interval representation of an interval graph can be computed in linear time [19]. We will speak about vertices and intervals interchangeably. For an interval u, we use $l(u)$ to denote its left endpoint and $r(u)$ to denote its right endpoint. By slightly moving and resizing the intervals of an interval representation, we may assume that all endpoints are unique. Among a set of intervals, we say that the interval u is *leftmost/rightmost* if $l(u)/r(u)$ is smallest/largest.

We need the following lemma, which applies to AT-free graphs, and thus also to interval graphs.

Lemma 9. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is an AT-free graph. For any minimal strategy for the Firefighting-with-Reserve game on (G, s), the last line of defense is the union of at most two minimal separators of G.*

Proof. Let R denote the set of vertices rescued by the strategy, let $R_1, \dots, R_p$ denote the connected components of $G[R]$, and let BP denote the set of vertices that either burn or are protected. Observe that BP is connected, or the strategy would not be minimal. Moreover, $R_1, \dots, R_p$ are connected components of $G[V(G) \setminus \text{BP}]$; hence, $N(R_i)$ is a minimal separator of G for each $i \in \{1, \dots, p\}$. Now consider the partial order that is induced on $\{N(R_i)\}_{i=1}^p$ by the inclusion-relation, and let $I \subseteq \{1, \dots, p\}$ be the set of indices corresponding to the maximal elements of this relation. Then the elements of $\{N(R_i)\}_{i \in I}$ are minimal separators that are incomparable with respect to set inclusion. Moreover, for each $j \in I$, the component of $G - N(R_j)$ that is not equal to R_j contains BP, and thus in particular it contains $(\bigcup_{i \in I} N(R_i)) \setminus N(R_j)$. Combined with the assumption that G is AT-free, it then follows from Broersma et al. [20, Definition 1, 18; Lemma 20] that $|I| \le 2$. Hence, as $N(R) = \bigcup_{i=1}^p N(R_i) = \bigcup_{i \in I} N(R_i)$, the lemma follows. □

Using Lemma 9 and the fact that interval graphs have at most $n-1$ minimal separators [21], one could think that it is sufficient to just protect the vertices of the pair of minimal separators that are closest to s and for which this yields a valid strategy. However, the example of Fig. 2 shows that protecting vertices between s and the separators allows for strictly better solutions, even for unit interval graphs. We thus need to get insight into which vertices to choose on the way, which we do in the following lemmas.

Lemma 10. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is an interval graph. Consider any strategy for the Firefighting-with-Reserve game on (G, s). At the start of time step $t \ge 1$, let u denote the leftmost burning vertex and let v denote the rightmost burning vertex. Then every vertex w with $r(u) < l(w) < r(w) < l(v)$ is either burning or protected. Moreover, any vertex that neighbors a burning vertex and that is neither burning nor protected at the start of time step t is in $N(u) \cup N(v)$.*

Proof. We apply induction. The statement is true at time step $t = 1$, because $u = v = s$ and thus there is no vertex w satisfying $r(s) < l(w) < r(w) < l(s)$. Now assume inductively that the statement is true at the start of time step $t-1$. Let u' and v' denote the leftmost and rightmost vertices respectively that are burning at the start of time step $t-1$, and let u and v denote the leftmost and rightmost vertices respectively that are burning at the start of time step t. By the induction hypothesis, all vertices that start burning in time step $t-1$ neighbor u' or v'. Therefore, u is the leftmost unprotected neighbor of u' (or $u = u'$) and v is the rightmost unprotected neighbor of v' (or $v = v'$). Moreover, note that any vertex w that satisfies $r(u) < l(w) < r(w) < l(v)$ also satisfies either $r(u') < l(w) < r(w) < l(v')$, or $r(u) < l(w) < r(u')$, or $l(v') < r(w) < l(v)$. In the first case, by induction, w is still either burning or protected at the start of time step t. In the second and third case, since u is a neighbor of u' (or $u = u'$) and v is a neighbor of v' (or $v = v'$), w is a neighbor of u' or of v' respectively, and thus w is burning or protected at the start of time step t. Therefore, any vertex x that neighbors a burning vertex and that is neither burning nor protected at time step t satisfies $l(x) < r(u)$ or $l(v) < r(x)$. Since u and v are the leftmost and rightmost vertex respectively that burns at time step t, x neighbors u or v. □

Lemma 11. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is an interval graph. Let $F_1, \dots, F_k$ be an optimal strategy for the Firefighting-with-Reserve game on (G, s). At the start of time step $t \geq 1$, let v denote the rightmost burning vertex. Let Y denote the set of vertices in $\bigcup_{i=1}^{t} F_i$ that neighbor v and do not neighbor any burning vertex in time step $t-1, \dots, 0$. Then $F'_1, \dots, F'_k$ is also an optimal strategy for the Firefighting-with-Reserve game on (G, s), where $F'_i = F_i \setminus Y$ for all $i \neq t$, and F'_t consists of $F_t \setminus Y$ and the $|Y|$ rightmost intervals that are unburned at the start of time step t and that intersect v.*

Proof. First, observe that $F_1 \setminus Y, \dots, F_{t-1} \setminus Y, F_t \cup Y, F_{t+1}, \dots, F_k$ is also an optimal strategy, because we can keep the firefighters to be placed on the vertices of Y in the reserve until time step t. Let Y' be the $|Y|$ rightmost neighbors of v that are unburned at time step t. Suppose that $Y \neq Y'$. Let I' be any interval of Y' that is not in Y and let I be any interval of Y that is not in Y'. Using Lemma 10, note that the set of unburned neighbors of I at time step $t+1$ is a subset of the set of unburned neighbors of I' at time step $t+1$, as both I and I' intersect v, but I' ends further to the right than I. Hence, $F'_1, \dots, F'_k$ saves at least as many vertices as $F_1, \dots, F_k$, and thus is also optimal. □

An analogous lemma can be proved for the leftmost interval that is on fire.

By Lemma 10, any unburned vertex that neighbors a burning vertex in fact neighbors u or v, where u and v denote the leftmost and rightmost burning vertex respectively at the start of a time step $t \geq 1$. By Lemma 11, we only need to protect the leftmost or rightmost neighbors of u and v respectively. Hence, we can strengthen Corollary 3 on interval graphs by combining it with the previous two lemmas, as follows.

Corollary 12. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is an interval graph. There is an optimal strategy for the Firefighting-with-Reserve game on (G, s) that, in each time step $t \geq 1$, protects the f_l leftmost vertices that neighbor u and the f_r rightmost vertices that neighbor v, for an appropriate choice of f_l and f_r, where u and v are the leftmost and rightmost vertex respectively that burn at the start of time step t.*

Corollary 12 is not only helpful to identify which vertices to choose before the last line of defense, but also to identify the last line of defense itself. In an interval graph, every minimal separator is a clique, which in turn corresponds to a point on the real line. Hence, the rightmost minimal separator of the two that we need to choose consists of all intervals containing the right endpoint of some other interval. Consequently, we can avoid guessing the minimal separators that make up the last line of defense, and use a unified approach instead.

Theorem 13. Firefighter *can be solved in time $O(n^7)$ on interval graphs.*

Proof. Let (G, s) be an instance of Firefighter such that G is an interval graph on n vertices. Instead of solving Firefighter on (G, s), we solve Firefighter Reserve Deployment on (G, s), which is equivalent according to Lemma 2.

Consider the following table: $A(s_l, s_r, u_l, u_r, f)$ is the maximum number of vertices that can be protected if s_l is the leftmost interval that is on fire, s_r is the rightmost interval that is on fire, u_l is the rightmost interval not ending to the right of the right endpoint of s_l that is unburned and unprotected, u_r is the leftmost interval not ending to the left of the left endpoint of s_r that is unburned and unprotected, and f is the size of the reserve. We also allow u_l and u_r to be the special symbol $\perp$ to signify that the fire is contained on the left respectively the right side of the graph. If $u_l \neq \perp \neq u_r$, then we set

$$A(s_l, s_r, u_l, u_r, f) = \max_{\substack{f_l, f_r \geq 0 \\ f_l + f_r \leq f}} \{f_l + f_r + A(s'_l, s'_r, u'_l, u'_r, f - f_l - f_r + 1)\}.$$

In the formula, s'_l is the (f_l+1)-th leftmost unburned interval intersecting the left endpoint of s_l. This interval can be computed from u_l. Similarly, s'_r is the (f_r+1)-th rightmost unburned interval intersecting the right endpoint of s_r, which can be computed from u_r. If s'_l does not exist, we set u'_l to $\perp$ and s'_l to s_l. Otherwise, we set u'_l to the rightmost nonneighbor of s_l ending to left of s_l. If s'_r does not exist, we set u'_r to $\perp$ and s'_r to s_r. Otherwise, we set u'_r to the leftmost non-neighbor of s_r starting to the right of s_r.

If say $u_l = \perp \neq u_r$, the formula simplifies to

$$A(s_l, s_r, u_l, u_r, f) = \max_{0 \leq f_r \leq f} \{f_r + A(s_l, s'_r, u_l, u'_r, f - f_r + 1)\},$$

where the meaning of s'_r and u'_r is the same as before. A similar formula can be given in case $u_l \neq \perp = u_r$. Finally, for any s_l, s_r, f, we set $A(s_l, s_r, \perp, \perp, f)$ to the number of vertices in the connected components of $G \setminus (X_l \cup X_r)$ that do not contain s, where X_l is the set of vertices intersecting the left endpoint of s_l and X_r is the set of vertices intersecting the right endpoint of s_r.

We now compute $p^* = A(s, s, u_l, u_r, 1)$, where u_l is the leftmost neighbor of s and u_r is the rightmost neighbor of s. Then there is a strategy that saves p^* vertices of G.

The correctness of the algorithm follows from Lemma 9 and Corollary 12. Since the table A has five indices that each take at most n different values and since an entry of A is a maximum over at most two numbers that each take at most n different values, the algorithm takes $O(n^7)$ time. □

5. Permutation graphs

In this section we show that Firefighter Reserve Deployment, and thus Firefighter, can be solved in polynomial time on permutation graphs. Let (G, s) be an instance of Firefighter Reserve Deployment such that G is a permutation graph. Since permutation graphs are AT-free, the last line of defense of any optimal strategy can be expressed as the union of at most two minimal separators by Lemma 9. Permutation graphs have $O(n^2)$ minimal separators [22].

A permutation graph with respect to a permutation π can be represented by a *permutation diagram* as follows: The diagram has two horizontal parallel rows drawn in the plane. The upper row has n specific points, labeled from left to right with integers 1 to n in their natural order, and the lower row has n specific points labeled from left to right with these integers in the order given by π. For each integer i between 1 and n, draw a straight line segment between the occurrence of i in the upper row and the occurrence of i in the lower row. Now it is easy to see that two vertices are adjacent if and only if their line segments cross each other. Permutation graphs can be recognized in linear time and a permutation diagram of a permutation graph can be computed in linear time as well [23]. It will be helpful to keep the idea of a permutation diagram in mind in the following proofs (even though we do not use it explicitly).

We need the following notions. Let G be a permutation graph on n vertices for permutation π. Then vertex j *lies to the left* of vertex i if $j < i$ and j appears before i in π. Given a set $S \subseteq V(G)$, a vertex is *to the left* of S if it is to the left of every vertex in S. We use $F_l(i)$ to denote the set of vertices that lie to the left of vertex i, and $F_l(S)$ to denote the set of vertices that lie to the left of $S \subseteq V(G)$. Similarly, a vertex j is said to be *to the right* of vertex i if $i < j$ and j appears after i in π; given a set $S \subseteq V(G)$, a vertex is *to the right* of S if it is to the right of every vertex in S. We use $F_r(i)$ to denote the set of vertices that lie to the right of vertex i, and $F_r(S)$ to denote the set of vertices that lie to the right of $S \subseteq V(G)$. We use $F(S)$ to denote $F_l(S) \cup F_r(S)$ for any $S \subseteq V(G)$. Note that by definition, for any $S \subseteq V(G)$, the vertices of $F(S)$ are neither in S nor do they neighbor any vertex of S. Given a set S of vertices, we call the set of *leftmost* vertices of S the set consisting of the two vertices $\arg\min_{i\in S} i$ and $\arg\min_{i\in S} \pi(i)$. Similarly, we call the set of *rightmost* vertices of S the set consisting of the two vertices $\arg\max_{i\in S} i$ and $\arg\max_{i\in S} \pi(i)$.

For the sake of the lemma, we define $F(\emptyset) = V(G)$. Note also that no vertex is burning at the start of time step 0.

Lemma 14. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is a permutation graph. Consider any strategy for the Firefighting-with-Reserve game on (G, s). At the start of time step $t \geq 1$, let L_t denote the set of leftmost vertices of the set of vertices that are burning and let R_t denote the set of rightmost vertices of the set of vertices that are burning. Then any vertex that is not in $F(L_{t-1} \cup R_{t-1})$ is either burning or protected. Moreover, any vertex that neighbors a burning vertex and that is neither burning nor protected at the start of time step t is in $N(L_t \cup R_t) \cap F(L_{t-1} \cup R_{t-1})$.*

Proof. We apply induction. The statement is true at time step $t = 1$, because $L_{t-1} \cup R_{t-1} = \emptyset$ and thus $F(L_{t-1} \cup R_{t-1}) = V(G)$, and because $L_t \cup R_t = \{s\}$. Now assume inductively that the statement is true at the start of time step $t - 1$. By the induction hypothesis, all vertices that start burning in time step $t - 1$ are in $N(L_{t-1} \cup R_{t-1}) \cap F(L_{t-2} \cup R_{t-2})$. Note that any vertex that is not in $F(L_{t-1} \cup R_{t-1})$ is either not in $F(L_{t-2} \cup R_{t-2})$, or in $N(L_{t-2})$, or in $N(R_{t-2})$. In the first case, it follows by induction that the vertex is either burning or protected; in the last two cases, it follows from the fact that the vertices in L_{t-2} and R_{t-2} respectively burn that the vertex is either burning or protected. To see the second part of the statement, observe that any vertex neither burning nor protected thus is in $F(L_{t-1} \cup R_{t-1})$. At the same time, any vertex that neighbors a burning vertex cannot be in $F(L_t \cup R_t)$. Hence, any vertex that neighbors a burning vertex and that is neither burning nor protected at the start of time step t is in $N(L_t \cup R_t) \cap F(L_{t-1} \cup R_{t-1})$. The lemma follows. □

Lemma 15. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is a permutation graph. Let $S_1, \dots, S_k$ be an optimal strategy for the Firefighting-with-Reserve game on (G, s). At the start of time step $t \geq 1$, let R_t denote the set of rightmost vertices of the set of burning vertices. Let X denote the set of vertices in $N(R_t) \cap F(R_{t-1}) \cap \bigcup_{i=1}^{t} S_i$ that do not neighbor any burning vertex in time step $t - 1, \dots, 0$. Then $S'_1, \dots, S'_k$ is also an optimal strategy for the Firefighting-with-Reserve game on (G, s), where $S'_i = S_i \setminus X$ for all $i \neq t$ and S'_t consists of $S_t \setminus X$ and, for some integer ℓ with $0 \leq \ell \leq |X|$:*

- *the set Y' of ℓ vertices i in $N(R_t) \cap F(R_{t-1})$ for which i is largest, and*
- *the set Z' of $|X| - \ell$ vertices i in $(N(R_t) \cap F(R_{t-1})) \setminus Y'$ for which $\pi(i)$ is largest.*

Proof. First, observe that $S_1 \setminus X, \dots, S_{t-1} \setminus X, S_t \cup X, S_{t+1}, \dots, S_k$ is also an optimal strategy, because we can keep the firefighters to be placed on the vertices of X in the reserve until time step t. We now describe an iterative procedure to improve X. Assume that $X \neq Y' \cup Z'$ for every choice of ℓ. Consider vertices $i = \arg\max_{i' \in (N(R_t) \cap F(R_{t-1})) \setminus X} i'$ and $j = \arg\max_{j' \in (N(R_t) \cap F(R_{t-1})) \setminus X} \pi(j')$. Note that i and j are properly defined unless $X = N(R_t) \cap F(R_{t-1})$, in which case $Y' \cup Z' = X$ for some choice of ℓ, a contradiction. If there is no vertex $i' \in X$ such that $i' < i$ and $\pi(i') < \pi(j)$, then there is a choice of ℓ such that $X = Y' \cup Z'$, a contradiction. So suppose that there is indeed a vertex $i' \in X$ such that $i' < i$ and $\pi(i') < \pi(j)$. Using Lemma 14, note that the set of unburned neighbors of i' at time step $t + 1$ is a subset of the set of unburned neighbors of i (j) at time step $t + 1$, because both i (j) and i' are in $N(R_t) \cap F(R_{t-1})$ but $i' < i$ $(\pi(i') < \pi(j))$. Hence, we can replace i' by i or j and save at least as many vertices. By performing this procedure iteratively, we eventually end up in the situation that $X = Y' \cup Z'$ for a suitable choice of ℓ. Hence, $S'_1, \dots, S'_k$ saves at least as many vertices as $S_1, \dots, S_k$, and thus is also optimal. □

An analogous lemma can be proved for the set of leftmost vertices of the set of burning vertices.

By Lemma 10, any unburned vertex that neighbors a burning vertex in fact neighbors L_t or R_t. By Lemma 11, we only need to protect the 'leftmost' or 'rightmost' neighbors of L_t and R_t respectively. Hence, we can strengthen Corollary 3 on permutation graphs by combining it with the previous two lemmas, as follows.

Corollary 16. *Let (G, s) be an instance of* Firefighter Reserve Deployment *such that G is a permutation graph. There is an optimal strategy for the Firefighting-with-Reserve game on (G, s) that, in each time step $t \geq 1$, protects:*

- *the ℓ_l vertices i in $N(L_t) \cap F(L_{t-1} \cup R_{t-1})$ for which i is smallest,*
- *the $f_l - \ell_l$ vertices i in $N(L_t) \cap F(L_{t-1} \cup R_{t-1})$ for which $\pi(i)$ is smallest and that were not part of the previous set,*
- *the ℓ_r vertices i in $N(R_t) \cap F(L_{t-1} \cup R_{t-1})$ for which i is largest, and*
- *the $f_r - \ell_r$ vertices i in $N(R_t) \cap F(L_{t-1} \cup R_{t-1})$ for which $\pi(i)$ is largest and that were not part of the previous set,*

for an appropriate choice of f_l, f_r, ℓ_l, ℓ_r, where L_t denotes the set of leftmost vertices of the set of vertices that are burning and R_t denotes the set of rightmost vertices of the set of vertices that are burning at the start of time step t.

Using this corollary, we can prove the following theorem.

Theorem 17. Firefighter *can be solved in polynomial time on permutation graphs.*

Proof. Let (G, s) be an instance of Firefighter such that G is a permutation graph on n vertices. Instead of solving Firefighter on (G, s), we solve Firefighter Reserve Deployment on (G, s), which is equivalent according to Lemma 2.

Compute all minimal separators of G in polynomial time [22], and add two dummy separators: one that lies to the left of all vertices and one that lies to the right of all vertices. For every pair of minimal separators X_l, X_r of G we do the following. We fill a table A, where $A(L, L_{-1}, R, R_{-1}, f)$ is the maximum number of vertices (including $X_l \cup X_r$) that can be protected if L is the set of leftmost vertices of the set of burned vertices, L_{-1} is the set of leftmost vertices of the set of burned vertices at the previous time step, R and R_{-1} are defined similarly with respect to the rightmost vertices, and f is the size of the reserve. We also allow L and R to be the special symbol $\perp$ to signify that the fire is contained on the 'left side' respectively the 'right side' of the graph. If $L \neq \perp \neq R$, then we set

$$A(L, L_{-1}, R, R_{-1}, f) = \max_{\substack{0 \leq f_l + f_r \leq f - |X'_l| - |X'_r| \\ 0 \leq \ell_l \leq f_l, 0 \leq \ell_r \leq f_r}} \{f_l + f_r + A(L', L'_{-1}, R', R'_{-1}, f - f_l - f_r + 1)\},$$

where:

- $X'_l = X_l \cap F_l(L_{-1}) \cap N(L)$ — these are the vertices of X_l that must be protected at the current time step;
- Y'_l is the set of ℓ_l vertices in $(F_l(L_{-1}) \cap N(L)) \setminus X'_l$ whose top endpoint is leftmost;
- Z'_l is the set of $f_l - \ell_l$ vertices in $(F_l(L_{-1}) \cap N(L)) \setminus X'_l$ whose bottom endpoint is leftmost;
- L' is the set of leftmost vertices of $N(L) - X'_l - Y'_l - Z'_l$, unless no vertices of X_l lie strictly to the left of L, in which case $L' = \perp$;
- $L'_{-1} = L$;
- $X'_r = X_r \cap F_r(R_{-1}) \cap N(R)$ — these are the vertices of X_r that must be protected at the current time step;
- Y'_r is the set of ℓ_r vertices in $(F_r(R_{-1}) \cap N(R)) \setminus X'_r$ whose top endpoint is rightmost;
- Z'_r is the set of $f_r - \ell_r$ vertices in $(F_r(R_{-1}) \cap N(R)) \setminus X'_r$ whose bottom endpoint is rightmost;
- R' is the set of rightmost vertices of $N(R) - X'_r - Y'_r - Z'_r$, unless no vertices of X_r lie strictly to the right of R, in which case $R' = \perp$;
- $R'_{-1} = R$.

We assume that $f - |X'_l| - |X'_r| \geq 0$, or we set the table entry to $-\infty$.

As in the algorithm for interval graphs, the formula simplifies if $L = \perp$ or $R = \perp$. In the former case ($L = \perp$), it is:

$$A(L, L_{-1}, R, R_{-1}, f) = \max_{\substack{0 \leq f_r \leq f - |X'_r| \\ 0 \leq \ell_r \leq f_r}} \{f_r + A(L, L_{-1}, R', R'_{-1}, f - f_r + 1)\},$$

where the meaning of X'_r, R', and R'_{-1} is the same as before, and we assume that $f - |X'_r| \geq 0$ or we set the table entry to $-\infty$. In the latter case ($R = \perp$), the formula is similar. Finally, we set $A(\perp, L_{-1}, \perp, R_{-1}, f) = 0$ for all L_{-1}, R_{-1}, f.

We now compute $p(X_l, X_r) = A(\{s\}, \emptyset, \{s\}, \emptyset, 1)$. Then there is a strategy that saves $p(X_l, X_r) + r(X_l, X_r)$ vertices, where $r(X_l, X_r) = |F_l(X_l)| + |F_r(X_r)|$ is the number of rescued vertices when $X_l \cup X_r$ constitutes the last line of defense. Finally, compute

$$\max_{X_l, X_r}\{p(X_l, X_r) + r(X_l, X_r)\},$$

where the maximum is over all minimal separators of G (including the two dummy ones).

The correctness of the algorithm follows from Lemma 9 and Corollary 16. Since the number of minimal separators of a permutation graph is $O(n^2)$ and they can be enumerated in polynomial time [22], the algorithm runs in polynomial time. □

6. Unit disk graphs

As mentioned in Section 1, Firefighter is NP-hard on trees (even of maximum degree 3). This immediately implies that the problem is NP-hard on circle graphs and disk graphs, which both form a superset of trees. On the positive side, we showed in Section 4 that Firefighter can be solved in polynomial time on interval graphs, and hence, also on unit interval graphs. In this section, we show that Firefighter is NP-hard on unit disk graphs, which form a superset of unit interval graphs and a subset of disk graphs.

Before we formally prove the main result of this section, we briefly observe that a direct reduction from Firefighter on trees (as for the proof of NP-hardness on disk graphs) does not work, because not every tree (even of maximum degree 3) is a unit disk graph. Indeed, there are even binary trees that are not unit disk graphs; for example, a full binary tree of diameter ℓ (for ℓ sufficiently large) is not a unit disk graph, because it has a set of $\Omega(2^\ell)$ pairwise non-adjacent vertices whereas an area bound shows that any set of pairwise non-adjacent vertices of any unit disk graph of diameter ℓ has size at most $O(\ell^2)$. Therefore, we need a more involved reduction.

To describe the reduction, we require some additional definitions and results. First, we need a special embedding of a low-degree tree in the plane. A *planar embedding* of a graph G is an assignment of closed curves in the plane to each edge of G such that a) no two curves intersect except possibly at their ends; b) for each $v \in V(G)$ there is a unique point in the plane such that the set of all curves meeting at this point corresponds precisely to the set of edges incident to v. A *rectilinear embedding* of a graph is a planar embedding where all curves consist of horizontal and/or vertical line segments. The number of *bends* of an edge in a rectilinear embedding is the number of times the curve switches from a horizontal to a vertical segment or vice versa.

Lemma 18. *Every tree T of maximum degree 4 on n vertices has a rectilinear embedding such that each edge has length exactly n and at most one bend. Moreover, such an embedding can be found in linear time.*

Proof. Root T at an arbitrary leaf r. Given a vertex $v \in V(T)$, let T_v denote the subtree of T rooted at v (note that $v \in V(T_v)$) and let C_v denote the set of children of v. Let L denote the set of non-root leafs of T, i.e. L contains all leafs of T except r. Use a pre-order (or depth-first) traversal of T starting at r to number the vertices of L from 1 up to $|L|$ in order of appearance in this pre-order traversal; let $b(u)$ denote the number assigned to a leaf u. For each $v \in V(T) \setminus L$, set $b(v)$ as the median of the b-values of the children of v if v has three children, and set $b(v)$ as $\max_{c \in C_v}\{b(c)\}$ otherwise. We can compute $b(v)$ for each vertex $v \in V(T) \setminus L$ in bottom-up fashion in linear time.

We now embed T as follows. First, place r at position $(b(r), 0)$. Then, perform the following recursive procedure on a vertex $v \in V(T)$, starting at $v = r$. If $v \in L$, then do nothing. So suppose that $v \notin L$. For each child $c \in C_v$, draw an edge from the position of v to that goes left/right by $|b(v) - b(c)|$ when $b(v) - b(c)$ is positive/negative respectively and goes down by $n - |b(v) - b(c)|$, and place c at the end of it.

From the description of the algorithm, it is immediate that each edge has length exactly n and at most one bend. Moreover, the algorithm only relies on pre-order and bottom-up tree traversals where each 'vertex visit' takes constant time, and thus the algorithm runs in linear time.

It remains to show that we indeed obtain a rectilinear embedding. To this end, we claim that for any $v \in V(T)$, $\min_{w \in V(T_v) \cap L}\{b(w)\} \leq b(v) \leq \max_{w \in V(T_v) \cap L}\{b(w)\}$. We prove the claim by induction. The claim is immediate if $v \in L$. So let $v \notin L$, and assume inductively that the statement is true for any $c \in C_v$. Then the claim follows from the definition of b. By the claim and the fact that each vertex v is embedded at the x-coordinate specified by $b(v)$, it follows that for any vertex v the vertices of $V(T_v)$ are embedded in the 'column' between x-coordinates $\min_{w \in V(T_v) \cap L}\{b(w)\}$ and $\max_{w \in V(T_v) \cap L}\{b(w)\}$. Since we used a pre-order traversal to number the vertices of L, this implies that for any $v \in V(T) \setminus L$, the columns in which its children were embedded are pairwise disjoint. Moreover, for each $v \in V(T) \setminus L$ and each $c \in C_v$, it follows from the claim that $0 \leq |b(v) - b(c)| \leq |L| < n$, and thus the y-coordinate of c is less than the y-coordinate of v. Therefore, the algorithm indeed produces a rectilinear embedding. □

The operation of *splitting a vertex* v is to create a new vertex v' and add edges such that $N[v] = N[v']$ (note that, in particular, v and v' become adjacent). We then call v and v' *(true) twins*. The *k-split* of a vertex v is obtained by splitting v iteratively k times; the resulting clique that contains v and the k new vertices is called the *split of* v. The operation of *k-subdividing* an edge (u, v) is to remove (u, v) and to add k new vertices $w_1, \dots, w_k$ such that w_i is adjacent to w_{i+1} for $i = 1, \dots, k-1$, w_1 is adjacent to u, and w_k is adjacent to v.

We are now ready to prove the NP-hardness result.

Theorem 19. Firefighter *is NP-hard on unit disk graphs.*

Proof. To prove that Firefighter is NP-hard, we show that the corresponding decision version is NP-complete. The decision version is, given a graph G, a vertex s of G, and an integer k, to decide whether there is a strategy for the Firefighting game on (G, s) such that at least k vertices are saved. The decision version of Firefighter is indeed in NP, even if G is a unit disk graph, because we can use the strategy as a witness and apply Lemma 1 to verify that the strategy saves at least k vertices.

To prove that the decision version of Firefighter is NP-hard, we reduce from the decision version of Firefighter on trees of maximum degree 3, which is known to be NP-hard [4]. Let (T, s, k) be an instance of this problem, where T is a tree, s a vertex of T, and k an integer. The idea is to subdivide each edge a suitable number of times, and then adapt the resulting tree such that the nature of the optimal solution to the problem is unchanged. We then use the embedding given by Lemma 18 to show that the constructed graph is in fact a unit disk graph.

Let $n = |V(T)| + 1$. Create a new vertex r and make it adjacent to s. Let T' denote the resulting tree; note that T' has maximum degree 4. Root T' at r. Then each vertex of T has a unique parent in T'. For each vertex $u \in V(T)$, we call the edge in T' between u and its parent the *parental edge of u*. Note that each edge of T' is uniquely assigned to a vertex in this manner. We $(2n-1)$-subdivide each edge of T'. Call the resulting tree T''. For each $u \in V(T)$, let $w^u_1, \dots, w^u_{2n-1}$ denote the newly created vertices for the parental edge of u, where w^u_1 is the vertex adjacent in T'' to the parent of u in T'. For each $u \in V(T)$, we $(2n-1)$-split w^u_1 and $(4n-1)$-split $w^u_2, \dots, w^u_{2n-1}$ and u. Call the resulting graph G.

Let $k' = 4kn(2n-1) + 2kn$ and let (G, r, k') be the resulting instance of the decision version of the Firefighter problem. We now prove a series of claims to show that one can save at least k' vertices in (G, r) if and only if one can save at least k vertices in (T, s).

Claim 1. *There exists an optimal strategy for the Firefighting game on (G, r) that protects no vertices of the split of $w^u_2, \dots, w^u_{2n-1}$ for any $u \in V(T)$.*

Proof. Consider any optimal strategy for the Firefighting game on (G, r) and suppose that it protects at least one vertex of the split of w^u_i for some $u \in V(T)$ and some $i \in \{2, \dots, 2n-1\}$. If the strategy protects at least $4n$ vertices of the splits of $u, w^u_2, \dots, w^u_{2n-1}$ in total, then we can instead protect all vertices of the split of w^u_1 and save more vertices. Indeed, since we $(2n-1)$-subdivided each edge, at least $2n$ of the protected vertices are protected before the fire reaches the vertices of the split of w^u_1, and thus protecting all vertices of the split of w^u_1 is possible. If the strategy protects $\ell < 4n$ vertices of the splits of $u, w^u_2, \dots, w^u_{2n-1}$ in total, then the fire will reach the vertices of the split of u anyway. Hence, one could just as well protect ℓ vertices of the split of u. The claim follows. $\diamond$

We now strengthen the above claim as follows.

Claim 2. *There exists an optimal strategy for the Firefighting game on (G, r) that protects no vertices of the split of $w^u_2, \dots, w^u_{2n-1}$ for any $u \in V(T)$, and protects vertices of the split of at most one vertex $x \in V(T)$.*

Proof. We further modify the strategy that we obtained in the previous claim. First, we observe that no optimal strategy protects all $4n$ vertices of the split of u for any $u \in V(T)$. Otherwise, as argued in the previous claim, it would be possible to protect all vertices of the split of w^u_1, and thus save more vertices. But then at least one vertex burns of the split of each $u \in V(T)$ that contains a protected vertex. Let v be a deepest vertex of T' such that a vertex of the split of v burns. Then instead of protecting a vertex in the split of some other $u \in V(T)$, we can protect a vertex of the split of v. Since v is at least as deep in the tree as u, the distance in G from r to the split of v is at least as large as the distance to the split of u. Hence, we can indeed perform this switch without decreasing the number of saved vertices. By iterating this operation, the claim follows. $\diamond$

This claim allows us to prove the first crucial claim to this proof.

Claim 3. *One can save at least k' vertices in (G, r) if and only if one can save at least k vertices in (T, s).*

Proof. Let $P = \{p_1, \dots, p_\ell\}$ be a set of vertices that forms a strategy for (T, s) that saves at least k vertices. Then protecting the vertices of the split of w^u_1 for each $u \in P$ is a strategy for (G, r) that saves at least k' vertices. Indeed, since each edge of T' was $(2n-1)$-subdivided, it is possible to protect this set of vertices. Moreover, for each vertex $u \in V(T)$ that is saved by P, the constructed strategy for (G, r) saves all vertices of the split of $u, w^u_1, \dots, w^u_{2n-1}$, which are $4n(2n-1) + 2n$ vertices. Hence, this strategy saves at least k' vertices.

For the converse, consider an optimal strategy for (G, r) and suppose that it saves at least k' vertices. Let P be the set of vertices u for which the strategy protects all vertices of w_1^u. By the construction of G, P yields a valid strategy for (T, s). Suppose that this strategy saves at most $k-1$ vertices on (T, s). By the preceding claim, we may assume that there is an optimal strategy for the Firefighting game on (G, r) that protects no vertices of the split of $w_2^u, \dots, w_{2n-1}^u$ for any $u \in V(T)$, and protects vertices of the split of at most one vertex $x \in V(T)$. Moreover, in the best case, the strategy is such that if the strategy protects at least one vertex of the split of w_u^1 for some $u \in V(T)$, then it protects all vertices of the split of w_u^1. But then the strategy can save at most $(k-1)4n(2n-1) + (k-1)2n + 4n$ vertices, which is less than k'. This is a contradiction. ⋄

It remains to prove the second crucial claim, namely that G is a unit disk graph.

Claim 4. *G is a unit disk graph.*

Proof. In order to prove this, we apply Lemma 18 to T'. Note that each vertex in this embedding has integer coordinates and that the bends occur at integer coordinates, and thus the embedding is a subset of suitably large grid. Now multiply all coordinates by 2. This means that each edge of the embedding has length exactly $2n$, and that the embedding still is a subset of a grid. We can then embed T'' by placing its vertices on the grid points touched by the embedding of T'. Observe that each edge in this embedding of T'' has length exactly 1, and each nonedge has length at least $\sqrt{2} > 1$ due to the multiplication by 2 we did before (the L_1 distance became at least 2, and thus the Euclidean distance is at least $\sqrt{2}$). But then T'' is a unit disk graph, as we can just place unit disks at the points of the grid where vertices of T'' are placed in the constructed embedding. Splitting a vertex v of a unit disk graph can be done by duplicating the unit disk corresponding to v (i.e. we center the new disk at the same point as where the disk corresponding to v is centered). It then follows from the construction of G that G is a unit disk graph. ⋄

This completes the proof of Theorem 19. □

Note that the proof naturally extends to other unit geometric objects, such as unit squares.

7. Concluding remarks

Although the Firefighter problem is NP-hard on even very restricted trees, our positive results in this paper show that we should seek to determine where its tractability border lies. A natural question, following the results on interval and permutation graphs, is whether Firefighter is polynomial-time solvable on common supersets of these graph classes, like AT-free graphs, or a graph class that is a superset of interval graphs and a subset of AT-free graphs, like co-comparability graphs.

The NP-hardness result on trees immediately implies that Firefighter is NP-hard on chordal graphs, circle graphs, polygon-circle graphs, interval filament graphs, and disk graphs, since these are superclasses of trees. This list contains several superclasses of co-comparability graphs. This makes the computational complexity of Firefighter on co-comparability graphs an intriguing open question.

References

[1] F.V. Fomin, P. Heggernes, E.J. van Leeuwen, Making life easier for firefighters, in: Proceedings of 6th International Conference Fun with Algorithms, FUN, in: Lecture Notes in Computer Science, vol. 7288, Springer Berlin Heidelberg, 2012, pp. 177–188.

[2] B.L. Hartnell, Firefighter! An application of domination, in: 25th Manitoba Conference on Combinatorial Mathematics and Computing, 1995.

[3] G. MacGillivray, P. Wang, On the firefighter problem, J. Combin. Math. Combin. Comput. 47 (2003) 83–96.

[4] S. Finbow, A. King, G. MacGillivray, R. Rizzi, The firefighter problem for graphs of maximum degree three, Discrete Math. 307 (16) (2007) 2094–2105.

[5] E. Anshelevich, D. Chakrabarty, A. Hate, C. Swamy, Approximation algorithms for the firefighter problem: cuts over time and submodularity, in: Proceedings of 20th International Symposium on Algorithms and Computation, ISAAC, in: Lecture Notes in Computer Science, vol. 5878, Springer, 2009, pp. 974–983.

[6] C. Bazgan, M. Chopin, M.R. Fellows, Parameterized complexity of the firefighter problem, in: Proceedings of 22nd International Symposium on Algorithms and Computation, ISAAC, in: Lecture Notes in Computer Science, vol. 7074, Springer, 2011, pp. 643–652.

[7] M. Cygan, F.V. Fomin, E.J. van Leeuwen, Parameterized complexity of firefighting revisited, in: Proceedings of 6th International Symposium on Parameterized and Exact Computation, IPEC, in: Lecture Notes in Computer Science, vol. 7112, Springer, 2011, pp. 13–26.

[8] C. Bazgan, M. Chopin, M. Cygan, M.R. Fellows, F.V. Fomin, E.J. van Leeuwen, Parameterized complexity of firefighting, J. Comput. System Sci. 80 (2014) 1285–1297.

[9] P. Chalermsook, J. Chuzhoy, Resource minimization for fire containment, in: Proceedings of 21st Annual ACM–SIAM Symposium on Discrete Algorithms, SODA, SIAM, 2010, pp. 1334–1349.

[10] L. Cai, Y. Cheng, E. Verbin, Y. Zhou, Surviving rates of graphs with bounded treewidth for the firefighter problem, SIAM J. Discrete Math. 24 (4) (2010) 1322–1335.

[11] L. Cai, E. Verbin, L. Yang, Firefighting on trees: $(1-1/e)$-approximation, fixed parameter tractability and a subexponential algorithm, in: Proceedings of 19th International Symposium on Algorithms and Computation, ISAAC, in: Lecture Notes in Computer Science, vol. 5369, Springer, 2008, pp. 258–269.

[12] B.L. Hartnell, Q. Li, Firefighting on trees: how bad is the greedy algorithm?, in: Proceedings of the Thirty-First Southeastern International Conference on Combinatorics, Graph Theory and Computing, Boca Raton, FL, 2000, in: Congr. Numer., vol. 145, 2000, pp. 187–192.

[13] S. Finbow, G. MacGillivray, The firefighter problem: a survey of results, directions and questions, Australas. J. Combin. 43 (2009) 57–77.

[14] A. Brandstädt, V.B. Le, J.P. Spinrad, Graph Classes: A Survey, SIAM, 1999.

[15] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Annals of Discrete Mathematics, vol. 57, North-Holland Publishing Co., 2004.
[16] R. Downey, M. Fellows, Parameterized Complexity, Springer, 1999.
[17] D.G. Corneil, Y. Perl, L. Stewart, A linear recognition algorithm for cographs, SIAM J. Comput. 14 (1985) 926–934.
[18] P.L. Hammer, B. Simeone, The splittance of a graph, Combinatorica 1 (1981) 275–284.
[19] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. System Sci. 13 (3) (1976) 335–379.
[20] H. Broersma, T. Kloks, D. Kratsch, H. Müller, A generalization of AT-free graphs and a generic algorithm for solving triangulation problems, Algorithmica 32 (4) (2002) 594–610.
[21] G.A. Dirac, On rigid circuit graphs, Abh. Math. Semin. Univ. Hambg. 25 (1961) 71–76.
[22] H. Bodlaender, T. Kloks, D. Kratsch, Treewidth and pathwidth of permutation graphs, SIAM J. Discrete Math. 8 (4) (1995) 606–616.
[23] R.M. McConnell, J.P. Spinrad, Modular decomposition and transitive orientation, Discrete Math. 201 (1–3) (1999) 189–241.