Making life easier for firefighters*

Fedor V. Fomin, Pinar Heggernes, and Erik Jan van Leeuwen

Department of Informatics, University of Bergen, Norway
{fedor.fomin, pinar.heggernes, e.j.van.leeuwen}@ii.uib.no

Abstract. Being a firefighter is a tough job, especially when tight city budgets do not allow enough firefighters to be on duty when a fire starts. This is formalized in the FIREFIGHTER problem, which aims to save as many vertices of a graph as possible from a fire that starts in a vertex and spreads through the graph. In every time step, a single additional firefighter may be placed on a vertex, and the fire advances to each vertex in its neighborhood that is not protected by a firefighter. 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 $W[1]$-hard when parameterized by the number of saved vertices, and it is NP-hard to approximate within $n^{1-\epsilon}$ for any $\epsilon > 0$. We aim to simplify the task of a firefighter by providing algorithms that show him/her how to efficiently fight fires in certain types of networks. We show that FIREFIGHTER can be solved in polynomial time on various well-known graph classes, including interval graphs, split graphs, permutation graphs, and P_k-free graphs for fixed k. On the negative side, we show that the problem remains NP-hard on unit disk graphs.

1 Introduction

Extinguishing a fire is a difficult task; just ask any firefighter. In particular, the task gets harder when there are not enough firefighters on duty when a fire breaks out, and additional firefighter resources are granted only as the fire spreads. Our aim here is to help firefighters extinguish a fire efficiently using the structure of the burning site. In the Firefighting game on a graph, a fire starts in a vertex s. At each step, a firefighter may be placed on a vertex which is not yet touched by the fire, which makes that vertex *protected*, i.e., unburnable, for the rest of the game. Then the fire spreads to every neighbor of the burning vertices that is not protected by a firefighter. After this, a new step starts. If, after some step, the burning vertices are separated from the rest of the graph by the protected vertices, then the fire is *contained* and the unburned vertices are referred to as *saved*. The FIREFIGHTER problem takes as input a graph G on n vertices and a vertex s of G, and the goal is to place firefighters as to maximize the number of saved vertices.

The FIREFIGHTER problem was introduced in 1995 and intended to capture also other important applications, like immunizing a population against a

* This work is supported by the Research Council of Norway.

virus [13]. The problem is notoriously difficult. It is NP-hard even on bipartite graphs [15] and on trees of maximum degree 3 [10]. It is NP-hard to approximate the FIREFIGHTER problem within $n^{1-\epsilon}$ for any $\epsilon > 0$ [1]. From a parameterized point of view, the problem is $W[1]$-hard when parameterized by the natural parameter of the number of saved vertices [2, 8].

Given the difficulty of the problem, it has naturally been studied for obtaining tractability on restricted inputs. However, although the problem and its variants are well studied [7], the only polynomial-time algorithms known for the problem so far are on graphs of maximum degree three when the fire starts at a vertex of degree at most two [10], and on so-called P-trees [15]. Even with respect to approximation and fixed-parameter tractability, the only positive results known so far are on trees and graphs of bounded treewidth. On arbitrary trees, the problem is fixed-parameter tractable [6], and a simple 2-approximation algorithm [14] along with a more involved $(1-1/e)$-approximation algorithm [6] exists. A recent survey of combinatorial and algorithmic results on the FIREFIGHTER problem has been given by Finbow and MacGillivray [11]

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. In particular, some of these constitute large classes of graphs of unbounded treewidth and cliquewidth. 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.

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 $v \in V(G)$ is denoted by $N(v)$. For a subset $U \subseteq V(G)$, $N(U) = \cup_{u \in U} N(u) \setminus U$. 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.

As we study FIREFIGHTER when the input graph belongs to various graph classes, we now we give the definitions of these. Below we list several well-known results without references; all details can be found in one of several excellent books on graph classes, e.g. [12, 4]. 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 intervals of the real line can be assigned to its vertices such that two vertices are adjacent if and only if their intervals overlap. 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 not related to each other, but they are both AT-free. (For convenience, we include a figure in Appendix A showing the inclusion relationship between all mentioned graph classes.)

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 to interval and permutation graphs. *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. Cographs are exactly the class of P_4-free graphs. Cographs form a subclass of permutation graphs, but they are unrelated to split and interval graphs.

For some of our algorithms, we provide a different but equivalent definition of the Firefighter problem (see also [8]). The Firefighter Reserve Deployment problem is defined as follows. Initially, the fire breaks out at a vertex s of G and the firefighter reserve has one firefighter. At each time step, the fire brigade can (permanently) deploy any number of its firefighter reserves to vertices of the graph that are not yet on fire, and the reserve decreases accordingly. Afterwards, the fire spreads to all of its unprotected neighbors, and one firefighter is added to the reserve. The objective is to save the maximum number of vertices.

A *strategy* for the Firefighter problem is simply an ordered set of vertices, representing the placement of the firefighters at each step. A strategy for the Firefighter Reserve Deployment is then an ordered collection $F_1, \ldots, F_k$ of vertex subsets, such that firefighters are deployed on the vertices of F_i at step i. In particular, this means that F_i might be empty for several i.

Lemma 1. *The* Firefighter *and the* Firefighter Reserve Deployment *problems are equivalent.*

Proof. Consider a strategy $v_1, \ldots, v_k$ for the Firefighter problem and look at the Firefighter Reserve Deployment problem. 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 Firefighter Reserve Deployment problem. 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 $a < b$. Clearly, any such ordering is a valid strategy for Firefighter, saving exactly the same set of vertices. □

When it is more convenient algorithmically, we will solve Firefighter Reserve Deployment instead of Firefighter. Recall that *saved* vertices are all unburned vertices when the Firefighting 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

If a building on fire does not have long corridors then we will show the firefighters how to find an optimal strategy efficiently. More formally, in this section we show that Firefighter can be solved in time $O(n^k)$ on P_k-free graphs. This result can be considered tight, as it is not likely that Firefighter can be solved in time $f(k)\, n^{O(1)}$ on P_k-free graphs, due to Theorem 2 below.

Lemma 2. *Let (G, s) be an instance of* Firefighter*, and let ℓ be the number of vertices on a longest induced path in G starting from 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. 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 1. Firefighter *can be solved in $O(n^{k-2}(n + m)) = O(n^k)$ time on P_k-free graphs.*

Proof. The longest induced path in a P_k-free graph G has at most $k-1$ vertices. Consequently, by Lemma 2, any optimal strategy on G protects at most $k - 2$ vertices. Hence we can enumerate all subsets $S \subseteq V(G)$ of size at most $k - 2$, check using a breadth-first search whether we can protect S and contain the fire, and then count the number of saved vertices. □

In terms of complexity classes FPT and XP (see e.g. [9] for definitions), Theorem 1 shows that the Firefighter problem is in XP when parameterized by the length of the longest induced path in the graph. This result 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. [8]. The statement of the theorem is different in [8], however the statement below is implicit. The reduction of [8] is from k-Clique, and yields a bipartite graph. Upon inspection, it is easy to see that the length of the longest induced path in this construction is $\max\{k + 1, 3\}$.

Theorem 2 ([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 1 immediately implies algorithms for FIREFIGHTER on these graph classes with running times $O(n^4)$ and $O(n^5)$, respectively. However, we next show that the problem can be solved in linear time on these graph classes.

Theorem 3. FIREFIGHTER *can be solved in* $O(n)$ *time on cographs.*

Proof. Let G be a connected cograph and let s be the vertex where the fire starts. 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. We can protect at most one vertex of G_2 since s is adjacent to all vertices in G_2. In the next step, we can protect at most one vertex of G_1 since all vertices in G_1 will at that point have burning neighbors. In particular, we can protect a vertex of G_1 if is not adjacent to s, regardless of the choice of protected vertex in the first step. Hence if s has a non-neighbor then we can protect and save two vertices: an arbitrary neighbor and an arbitrary non-neighbor of s. Otherwise we can protect and save one arbitrary vertex. □

Theorem 4. FIREFIGHTER *can be solved in linear time on split graphs.*

Proof. Let (G, s) be an instance of FIREFIGHTER such that G is a split graph with $V(G) = I \cup C$ for an independent set I and a clique C. Observe first that there is an optimal strategy that protects at most one vertex of I. To see this, consider an optimal strategy that protects at least two vertices of I: u and v, such that u is protected before v. Let w be a neighbor of v. At the time that u is protected, w is not burning, otherwise it would not be valid to protected v. The strategy that protects w instead of u saves at least as many vertices as the optimal strategy. Hence in the following, consider optimal strategies that protect at most one vertex from I. To avoid trivial cases, assume that both C and I contain at least two vertices each. Observe also that at most two vertices of C can be protected regardless of where the fire starts, since C is a clique. Since split graphs are P_5-free, the longest induced path contains at most four vertices. By Lemma 2, at most three vertices can be protected in total.

Suppose that $s \in C$. Then only one vertex of C can be protected, and all vertices of C, except the protected vertex v, will burn. If v has neighbors of degree 1, then these are saved. In the next step, the best we can do is to protect an unsaved vertex w in I which is not adjacent to s. After this, all vertices that are not protected or saved so far will be on fire. Hence an optimal strategy simply finds a vertex $v \neq s$ of C with the highest number of degree 1 neighbors and protects it. It then protects vertex w if it exists. Vertices v and w can clearly be found in $O(n)$ time.

Suppose that $s \in I$. It then follows from the above arguments that any optimal strategy protects either one or two vertices of C and exactly one vertex of I. Moreover, the first vertex that is protected is in C.

1. Suppose that any optimal strategy protects exactly one vertex of C. If s has degree one, then the optimal strategy is trivial. So assume otherwise. By the

above observation, at most one vertex of I is protected, and the strategy protects at most two vertices in total. Then s must be adjacent to every vertex of C, since we could have protected two vertices in C otherwise, as this saves at least as many vertices as protecting one vertex of C and at most one vertex of I. But then the vertex of C that we want to protect is one with the highest number of degree 1 neighbors in $I \backslash \{s\}$.

2. Suppose that there is an optimal strategy protecting two vertices of C. Let $U(X)$ denote the set of vertices in I that only have neighbors in the set $X \subseteq C$. Using a similar line of reasoning as above, we can conclude that the two vertices $u, v \in C$ that the optimal strategy uses must have maximal $|U(\{u, v\})|$ over all $X \subseteq C$ with $|X| \leq 2$ and $|X \cap N(s)| \leq 1$. The crux is to find these two vertices in linear time. We first compute $|U(c)|$ for each $c \in C$. This takes linear 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 exact pair-bonus, in linear time as follows. Create a bucket for each vertex of C. For each degree-two 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 linear 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 it follows that there is an optimal strategy that chooses u, v, and possibly one more vertex of I.

From the above description, it is clear that an optimal strategy can be found in linear time. □

4 Interval Graphs

We have seen that buildings without long corridors are helpful with respect to deciding an optimal firefighting strategy when a fire breaks out. Now we will see that even if there are long corridors, if all the long corridors go in the same direction and have enough connections to each other, then the firefighters may also be able to figure out an optimal strategy efficiently. In particular, we will now show that Firefighter can be solved in polynomial time on interval graphs.

An interval model of an interval graph can be computed in linear time. We will speak about vertices and intervals interchangeably. We say that an interval u is *to the left (right) of* an interval v if the left (right) endpoint of the interval of u is to the left (right) of the left (right) endpoint of the interval of v. We will use *leftmost* and *rightmost* analogously.

In an arbitrary graph G, let $C \subseteq V(G)$ be such that $G[C]$ is connected, and let $C_1, \ldots, C_t$ be the connected components of $G[V(G) \setminus C]$. It is well known that $N(C_i)$ is a minimal separator of G, for $1 \leq i \leq t$. Furthermore, if G is an

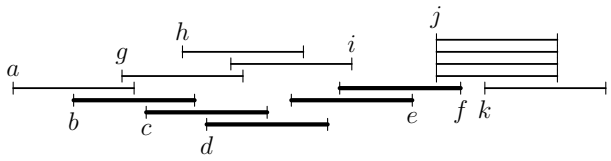

Fig. 1. In this (unit) interval graph, the thicker lines represent ten intervals with the same endpoints. The fire starts at a. The four vertices of j and vertex k both are minimal separators (imagine that the graph continues after k) that can be protected before the fire reaches them. However, if we 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).

AT-free graph and S is the union of $N(C_1), \ldots, N(C_t)$, then there is a collection of at most two minimal separators whose union equals S [5, 3]. Although AT-free graphs can have an exponential number of minimal separators (consider e.g., the complement of a bipartite graph), in an interval graph every minimal separator is a clique, and there are at most $n - 1$ minimal separators.

Let (G, s) be an instance of Firefighter Reserve Deployment, where G is an interval graph, and let R be any maximal connected set of rescued vertices. Then, by the above, $N(R)$ is a minimal separator of G. Furthermore, since interval graphs form a subclass of AT-free graphs, we can immediately conclude that the last line of defense in an interval graph is the union of at most two minimal separators. One could think that it sufficient to just protect the vertices of these minimal separators, and then find the pair of minimal separators that are closest to the root of the fire for which this works. However, the example of Figure 1 shows that protecting vertices between the root 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.

Lemma 3. *Let G be an interval graph and let $F_1, \ldots, F_k$ be an optimal strategy for* Firefighter Reserve Deployment *from a given start vertex. For a time step $t > 1$, let u denote the rightmost interval that is on fire. Then there is an optimal strategy $F'_1, \ldots, F'_{k'}$ such that $k = k'$, $F_i = F'_i$ for all $i \neq t$, and F'_t consists of X and the $|F_t| - |X|$ unburned intervals having the rightmost endpoint and intersecting u, where X is the set of intervals in F_t intersecting the leftmost interval that is on fire.*

Proof. Let Y be the set of vertices of F_t intersecting u and let Y' be the $|Y|$ rightmost intervals intersecting u (i.e. the ones whose endpoint is rightmost). Assume that $Y \neq Y'$ and let I' be any interval of Y' that is not in Y. Let I be any interval of Y that is not in Y'. Clearly, the set of unburned neighbors of I in time step $t + 1$ is a subset of the set of unburned neighbors of I' in time step $t + 1$, as both I and I' intersect u, but I' ends further to the right than I. Hence $F'_1, \ldots, F'_k$ is also an optimal strategy. □

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

Lemma 3 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 rather use a unified approach.

Theorem 5. FIREFIGHTER *can be solved in time* $O(n^7)$ *on interval graphs.*

Proof. Consider the following table: $A(s_1, s_2, u_1, u_2, f)$ is the maximum number of vertices that can be protected if s_1 is the leftmost interval that is on fire, s_2 is the rightmost interval that is on fire, u_1 is the rightmost interval not ending to the right of the right endpoint of s_1 that is unburned and unprotected, u_2 is the leftmost interval not ending to the left of the left endpoint of s_2 that is unburned and unprotected, and f is the size of the reserve. We also allow u_1 and u_2 to be the special symbol $\perp$ to signify that the fire is contained on the left respectively the right side of the graph. If $u_1 \neq \perp \neq u_2$, we then set

$$A(s_1, s_2, u_1, u_2, f) = \max_{0 \leq f_1 + f_2 \leq f} \{f_1 + f_2 + A(s_1', s_2', u_1', u_2', f - f_1 - f_2 + 1)\}.$$

In the formula, s_1' is the $(f_1 + 1)$-th leftmost unburned interval intersecting the left endpoint of s_1. This can easily be computed from u_1. Similarly, s_2' is the $(f_2 + 1)$-th rightmost unburned interval intersecting the right endpoint of s_2, which can be computed from u_2. If s_1' does not exist, we set u_1' to $\perp$ and s_1' to s_1. Otherwise, we set u_1' to the rightmost nonneighbor of s_1 ending to left of s_1. If s_2' does not exist, we set u_2' to $\perp$ and s_2' to s_2. Otherwise, we set u_2' to the leftmost non-neighbor of s_2 starting to the right of s_2.

If say $u_1 = \perp \neq u_2$, the formula simplifies to

$$A(s_1, s_2, u_1, u_2, f) = \max_{0 \leq f_2 \leq f} \{f_2 + A(s_1, s_2', u_1, u_2', f - f_2 + 1)\},$$

where the meaning of s_2' and u_2' is the same as before. A similar formula can be given in can $u_1 \neq \perp = u_2$. Finally, we set $A(s_1, s_2, \perp, \perp, f)$ to the number of vertices in the connected components of $G \backslash (X_1 \cup X_2)$ that do not contain s, where X_1 is the set of vertices intersecting the left endpoint of s_1 and X_2 is the set of vertices intersecting the right endpoint of s_2.

We now compute $p^* = A(s, s, u_1, u_2, 1)$, where u_1 is the leftmost neighbor of s and u_2 is the rightmost neighbor of s. Then there is a strategy that saves p^* vertices of G.

The correctness of the algorithm follows immediately from Lemmas 1 and 3, and the mentioned properties of minimal separators of interval graphs. It is immediate from the description that computing the table A and the solution takes $O(n^7)$ time. □

5 Permutation Graphs

We continue our quest against fires in burning sites where all the long corridors go in the same direction. Let (G, s) be an instance of Firefighter Reserve Deployment such that G is a permutation graph. Since permutation graphs are AT-free, exactly as for interval graphs, the last line of defense of any optimal strategy can be expressed as the union of at most two minimal separators. Permutation graphs have $O(n^2)$ minimal separators.

A permutation graph can be represented by a *permutation diagram* as follows: The diagram has two rows, one containing the integers 1 to n in their natural order and one containing 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 one row and the occurrence of i in the other row. Now it is easy to see that two vertices are adjacent if and only if their line segments cross each other. A permutation diagram of a permutation graph can be computed in linear time.

We use the following definitions. Given a set of functions $F = \{f_1, \ldots, f_\ell\}$, where $f_i : \mathbb{R} \to \mathbb{R}$, the *left envelope* of F is the set of points (x, y) such that $f_i(x) = y$ for some $1 \leq i \leq \ell$ and there is no $x' < x$ such that $f_{i'}(x') = y$ for some $1 \leq i' \leq \ell$. The *right envelope* of F is similarly defined. To get an algorithm, we use the permutation diagram of the permutation graph and do not distinguish between the line segments of the diagram and the vertices that they represent. We can then talk about the left and right envelopes of a set of vertices.

Lemma 4. *Let G be a permutation graph and let $F_1, \ldots, F_k$ be an optimal strategy for the* Firefighter Reserve Deployment *problem from a given start vertex. For a time step $t > 1$, let U denote the set of vertices on the right envelope of the set of vertices that are burned at time step t. Then there is an optimal strategy $F'_1, \ldots, F'_{k'}$ and an integer $\ell \geq 0$ such that $k = k'$, $F_i = F'_i$ for all $i \neq t$, and F'_t consists of X, Y', and Z', where*

- *X is the set of vertices in F_t intersecting the left envelope of the set of burned vertices,*
- *Y' is the set of ℓ vertices intersecting a vertex of U whose top endpoint is rightmost, and*
- *Z' is the set of $|F_t| - |X| - \ell$ vertices intersecting a vertex of U whose bottom endpoint is rightmost.*

Proof. First observe that any vertex that lies strictly between the left and the right envelope and does not intersect a vertex on any of the envelopes must be burned at time step t. This can easily be shown by induction. Hence any protected vertex must intersect a vertex of U. Let Y be the set of vertices in F_t whose top endpoint lies to the right of the top endpoint of any vertex in U, and let Z be the vertices of F_t not in X or Y. Choose ℓ to be $|Y|$, which gives a proper determination of Y' and Z'.

Suppose that $Y \neq Y'$ and let v be any vertex of $Y \backslash Y'$. Let u be the vertex of $Y' \backslash Y$ having the rightmost top endpoint. Clearly, the set of unburned neighbors

of v in time step $t+1$ is a subset of the set of unburned neighbors of u in time step $t+1$. Hence we may replace v by u without compromising optimality.

A similar argument can be employed in the case when $Z \neq Z'$. It follows that $F'_1, \ldots, F'_k$ is also an optimal strategy. □

A similar lemma may be proven with respect to the left envelope.

If we want to be able to use a similar dynamic programming approach as in the previous section, it seems to follow from the above lemma that we need to maintain the left and right envelope of the set of burning vertices. Unfortunately, these envelopes can contain up to $O(n)$ vertices, which is not feasible with this kind of dynamic programming approach. Upon closer inspection, however, we only need to be able to discern vertices intersecting these envelopes. To do this efficiently, we use the following observation.

Observation 1 *Let $X \subset V(G)$, and let R be the set of vertices forming the right envelope of X. Let r_1 denote the vertex in R with the rightmost top endpoint, and let r_2 be the vertex in R with the rightmost bottom endpoint. Then any vertex of G intersecting a vertex of R must intersect r_1 or r_2.*

Proof. Since r_1 has the rightmost top endpoint and is in R, any other vertex that is in R must have its bottom endpoint to the right of the bottom endpoint of r_1. We can make a similar observation about r_2. The result follows. □

It follows that the vertices intersecting a vertex of an envelope can be found by maintaining two vertices of the envelope. These are the *representing vertices* of the envelope.

It only remains to figure out which vertices are unburned and unprotected at the current time step. Those are the vertices that we can protect. But to that end it suffices to observe that any such vertices must lie fully to the right of the right envelope of the set of burned vertices in the previous time step, or fully to the left of its left envelope. It is easy to verify this property from the representing vertices of the envelope at the previous time step.

Theorem 6. FIREFIGHTER *can be solved in polynomial time on permutation graphs.*

Proof. The above discussion yields the following dynamic programming algorithm. We first guess two minimal separators X_1, X_2. Then we find a table $A(L, L_{-1}, R, R_{-1}, f)$, which is the maximum number of vertices (including $X_1 \cup X_2$) that can be protected if L is the set of representing vertices of the left envelope of the set of burned vertices, L_{-1} is the set of representing vertices of the left envelope of the set of burned vertices in a previous time step, R and R_{-1} are defined similarly with respect to the right envelope, and f is the size of the reserve. From there the idea is mainly the same as for interval graphs, although the details are more tedious. We leave out these details in this extended abstract.

The correctness of the algorithm follows immediately from Lemma 4 and the fact that the last line of defense in an optimal strategy can be covered by at

most two minimal separators. It follows immediately from the description of the algorithm that its running time is polynomial. By Lemma 1, the result follows. □

6 Concluding Discussion and Unit Disk Graphs

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 superclasses of these graph classes, for example co-comparability graphs, or their superclass AT-free 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. Hence we find the computational complexity of FIREFIGHTER on co-comparability graphs an intriguing open question.

We conclude our paper by giving a highly related NP-hardness result. A *unit disk graph* is the intersection graph of disks of unit diameter on the plane, and hence a superclass of unit interval graphs and a subclass of disk graphs.

Theorem 7. FIREFIGHTER *is NP-hard on unit disk graphs.*

Proof. We give a sketch of the proof. The full proof, together with several auxiliary lemmas and definitions is given in Appendix B.

We reduce from the FIREFIGHTER problem on trees of maximum degree three, which is known to be NP-hard [10]. Let (T, s, k) be an instance of this problem. 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 a particular embedding of T to show that the constructed graph is in fact a unit disk graph.

Root T at s, and let $n := |V(T)|$. Then each vertex (except s) has a unique parent. For each vertex $u \neq s$, we call the edge between u and its parent the *edge of* u. Note that each edge of the tree is uniquely assigned to a vertex in this manner. Each edge of T is now $2n-1$-subdivided, and the resulting tree is called T'. For each $u \in V(T)\backslash\{s\}$, let $w_1^u, \ldots, w_{2n-1}^u$ denote the newly created vertices for the edge of u, where w_1^u is the vertex adjacent to the parent of u. For each $u \in V(T)\backslash\{s\}$, we $(2n-1)$-split w_1^u and $4n$-split $w_2^u, \ldots, w_{2n-1}^u$ and u. Call the resulting graph G. Let $k' := 4kn(2n-1) + k(2n-1)$ and let (G, s, k') be the resulting instance of the FIREFIGHTER problem. The proof is completed by proving that one can save at least k' vertices in (G, s) if and only if one can save at least k vertices in (T, s). □

A more direct proof does not work, as not even all binary trees are unit disk graphs: a unit disk graph of diameter ℓ can contain at most $O(\ell^2)$ independent

vertices, whereas a binary tree of diameter ℓ can contain $\Omega(2^\ell)$ independent vertices.

References

1. Anshelevich, E., Chakrabarty, D., Hate, A., Swamy, C.: Approximation algorithms for the firefighter problem: Cuts over time and submodularity. In: Dong, Y., Du, D.Z., Ibarra, O.H. (eds.) ISAAC. Lecture Notes in Computer Science, vol. 5878, pp. 974–983. Springer-Verlag, Berlin (2009)
2. Bazgan, C., Chopin, M., Fellows, M.R.: Parameterized complexity of the firefighter problem. In: ISAAC (2011)
3. Bouchitté, V., Todinca, I.: Approximating the treewidth of at-free graphs. In: Brandes, U., Wagner, D. (eds.) Proceedings WG 2000, LNCS, vol. 1928, pp. 59–70. Springer (2000)
4. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. SIAM (1999)
5. Broersma, H., Kloks, T., Kratsch, D., Müller, H.: A generalization of at-free graphs and a generic algorithm for solving treewidth, minimum fill-in and vertex ranking. In: Hromkovic, J., Sykora, O. (eds.) Proceedings WG 1998, LNCS, vol. 1517, pp. 293–319. Springer (1998)
6. Cai, L., Verbin, E., Yang, L.: Firefighting on trees: (1-1/e)-approximation, fixed parameter tractability and a subexponential algorithm. In: ISAAC. pp. 258–269 (2008)
7. Chalermsook, P., Chuzhoy, J.: Resource minimization for fire containment. In: SODA (2010)
8. Cygan, M., Fomin, F.V., van Leeuwen, E.J.: Parameterized complexity of firefighting revisited. In: IPEC (2011)
9. Downey, R., Fellows, M.: Parameterized Complexity). Springer (1999)
10. Finbow, S., King, A., MacGillivray, G., Rizzi, R.: The firefighter problem for graphs of maximum degree three. Discrete Math. 307(16), 2094–2105 (2007)
11. Finbow, S., MacGillivray, G.: The firefighter problem: A survey of results, directions and questions (2007), manuscript
12. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57). North-Holland Publishing Co. (2004)
13. Hartnell, B.L.: Firefighter! an application of domination. In: 25th Manitoba Conference on Combinatorial Mathematics and Computing (1995)
14. Hartnell, B.L., Li, Q.: 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). Congr. Numer., vol. 145, pp. 187–192 (2000)
15. MacGillivray, G., Wang, P.: On the firefighter problem. J. Combin. Math. Combin. Comput. 47, 83–96 (2003)

Appendix A : Inclusion relationship between graph classes

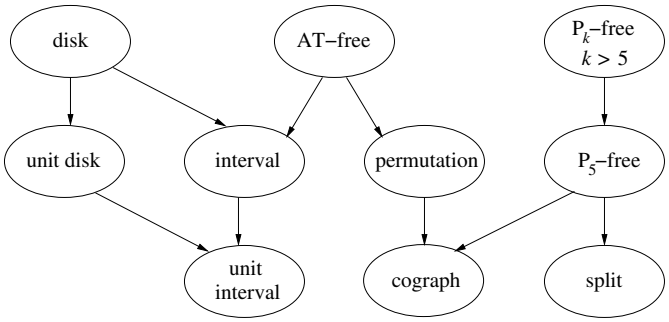

Fig. 2. The graph classes mentioned in this paper, where $\rightarrow$ represents the $\supset$ relation.

Appendix B: NP-hardness on Unit Disk Graphs

In this section we give the full proof of Theorem 7, along with the necessary additional definitions and lemmas.

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 vertical segment or vice versa.

Lemma 5. *Every tree T of maximum degree three has a rectilinear embedding such that each edge has length exactly $|V(T)|$ and at most one bend. Moreover, such an embedding can be found in linear time.*

Proof. Root T at an arbitrary leaf r. Use a pre-order (or depth-first) traversal of T to obtain the list of leaves L of T, and number them 1 up to $|L|$ in the order in which they were found. Let $x(u)$ denote the number assigned to a leaf u. We can then compute $x(v) = \max_{w \in C_v}\{x(w)\}$ in a bottom-up fashion, where C_v is the set of children of v. We will use x to determine the x-coordinates of the vertices of the tree in the embedding.

We first place r at position $(x(r), 0)$. For each internal vertex v of the tree (we count the root among the internal vertices), do the following. Let c_1 denote the child of v that has the smallest value of x, and (if it exists) let c_2 denote the other child. Draw an edge from v that goes to the left by $x(v) - x(c_1) \geq 0$ and down by $|V(T)| - x(v) + x(c_1) \geq 1$. Place c_1 at the end and recurse on c_1. Similarly, if c_2 exists, draw an edge from v that goes to the right by $x(c_2) - x(v) \geq 0$ and down by $|V(T)| - x(c_2) + x(v) \geq 1$. Place c_2 at the end and recurse on c_2.

It is immediate from the description of the algorithm that each edge has length $|V(T)|$ and at most one bend. The choice of x ensures that this is indeed a rectilinear embedding (note that $\max\{x(v) - x(c_1), x(c_2) - x(v)\} \geq 1$). The algorithm clearly has linear running time. □

Observe that the lemma extends to trees of maximum degree four, but we do not need this.

Before we prove Theorem 7, we need a few more definitions. The operation of *splitting a vertex* v is to create a new vertex v' and add edges such that $N[v] = N[v']$. We then call v and v' *(true) twins*. A *k-split* is obtained by splitting a vertex k times (this adds k new vertices). The operation of *k-subdividing* an edge (u, v) is to remove (u, v) and to add k new vertices $w_1, \ldots, w_k$ such that w_i is adjacent to w_{i+1} for $i = 1, \ldots, k-1$, w_1 is adjacent to u and w_k is adjacent to v.

Proof of Theorem 7. We reduce from the Firefighter problem on trees of maximum degree three, which is known to be NP-hard [10]. Let (T, s, k) be an instance of this problem. 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 5 to show that the constructed graph is in fact a unit disk graph.

Root T at s, and let $n := |V(T)|$. Then each vertex (except r) has a unique parent. For each vertex $u \neq s$, we call the edge between u and its parent the *edge of* u. Note that each edge of the tree is uniquely assigned to a vertex in this manner. Each edge of T is now $2n-1$-subdivided, and the resulting tree is called T'. For each $u \in V(T)\backslash\{s\}$, let $w_1^u, \ldots, w_{2n-1}^u$ denote the newly created vertices for the edge of u, where w_1^u is the vertex adjacent to the parent of u. For each $u \in V(T)\backslash\{s\}$, we $(2n-1)$-split w_1^u and $4n$-split $w_2^u, \ldots, w_{2n-1}^u$ and u. Call the resulting graph G.

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

Claim. There exists an optimal strategy for the Firefighter problem on (G, s) that protects no vertices of the split of $w_2^u, \ldots, w_{2n-1}^u$ for any $u \in V(T)\backslash\{s\}$.

Proof. Consider any optimal strategy for the Firefighter problem on (G, s) and suppose that it protects at least one vertex of the split of w_i^u for some $u \in V(T)\backslash\{s\}$ and some $2 \leq i \leq 2n-1$. If the strategy protects at least $4n$

vertices of the splits of $w_2^u, \ldots, w_{2n-1}^u$, then we can instead protect all vertices of the split of w_1^u and save more vertices. Note that 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_1^u, and thus protecting all vertices of the split of w_1^u is indeed possible. If the strategy protects $\ell < 4n$ vertices of the splits of $w_2^u, \ldots, w_{2n-1}^u$, then the fire will reach all vertices of the split of u anyway. Hence one could just as well protect ℓ vertices of the split of u. The claim follows. □

We now strengthen this claim as follows.

Claim. There exists an optimal strategy for the FIREFIGHTER problem on (G, s) that protects no vertices of the split of $w_2^u, \ldots, w_{2n-1}^u$ for any $u \in V(T)\backslash\{s\}$, and protects vertices of the split of at most one such u. Moreover, u is a leaf of 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 $u \in V(T)\backslash\{s\}$. Otherwise, using a similar argument as in the previous claim, it would be possible to protect all vertices of the split of w_1^u, and thus save more vertices. But then at least one vertex burns of the split of each $u \in V(T)\backslash\{s\}$ that contains a protected vertex. Let v be the deepest leaf of the T such that a vertex of the split of v is burned. Then instead of protecting a vertex in the split of some other $u \in V(T)\backslash\{s\}$, we can protect a vertex of the split of v. Since v is at least as deep in the tree as u, the distance from s in G to the split of v is at least as large as the distance to the split of u. Hence this poses no problem. The claim follows. □

We strengthen this claim as well.

Claim. There exists an optimal strategy for the FIREFIGHTER problem on (G, s) such that if the strategy protects at least one vertex of the split of w_u^1 for some $u \in V(T)\backslash\{s\}$, then it protects all vertices of the split of w_u^1. Moreover, all protected vertices that are not in the split of w_u^1 for some $u \in V(T)\backslash\{s\}$ are in the split of u for some leaf u of T.

Proof. It follows from the previous claim that there exists an optimal strategy that protects no vertices of the split of $w_2^u, \ldots, w_{2n-1}^u$ for any $u \in V(T)\backslash\{s\}$, and protects vertices of the split of at most one such u. Moreover, u is a leaf of T. Suppose there is a $v \in V(T)\backslash\{s\}$ for which this strategy protects at least one, but not all of the vertices of w_1^v. But then we could just as well protect vertices in the split of v. Following the arguments of the previous claim, we can instead protect vertices in the split of the leaf u. If necessary, we re-apply the arguments of the previous claim. The claim follows. □

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

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

Proof. Let $P = \{p_1, \ldots, p_\ell\}$ be a set of vertices that form a strategy for (T, s) that save at least k vertices. Then protecting the vertices of the split of w_1^u for each $u \in P$ is a strategy for (G, s) that saves at least k' vertices. Note that since each edge of T was $(2n-1)$-subdivided, it is indeed possible to save this set of vertices. Moreover, from the construction and the value of k', it is immediate that this strategy saves at least k' vertices.

For the converse, consider an optimal strategy for (G, s) 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 construction of G, P yields a feasible strategy for (T, s). Suppose that this strategy saves at most $k-1$ vertices in (T, s). By the preceding claim, we may assume that the optimal strategy for (G, s) is such that if the strategy protects at least one vertex of the split of w_u^1 for some $u \in V(T)\backslash\{s\}$, then it protects all vertices of the split of w_u^1. Moreover, all protected vertices that are not in the split of w_u^1 for some $u \in V(T)\backslash\{s\}$ are in the split of u for some leaf u of T. But then the strategy for (G, s) that we have can save at most $4(k-1)n(2n-1) + (k-1)(2n-1) + 4n$ 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. G is a unit disk graph.

Proof. In order to prove this, we apply Lemma 5 to T and multiply all coordinates by 2. This means that each edge of the embedding has length exactly $2n$. But then we can embed T' in the plane such that each vertex of T' is placed at a point of the grid. Moreover, each edge in this embedding has length one, and each nonedge has length at least two due to the multiplication by 2 we did before. 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 may 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 would be centered). But then it follows from the construction of G that G is a unit disk graph. □

This proves the theorem. □

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