Parameterized Complexity of the Firefighter Problem

Cristina Bazgan[1,2], Morgan Chopin[1], and Michael R. Fellows[3,⋆]

[1] Université Paris-Dauphine, LAMSADE, France
{bazgan,chopin}@lamsade.dauphine.fr
[2] Institut Universitaire de France
[3] Charles Darwin University, Australia
michael.fellows@cdu.edu.au

Abstract. In this paper we study the parameterized complexity of the firefighter problem. More precisely, we show that SAVING k-VERTICES and its dual SAVING ALL BUT k-VERTICES are both W[1]-hard for parameter k even for bipartite graphs. We also investigate several cases for which the firefighter problem is tractable. For instance, SAVING k-VERTICES is fixed-parameter tractable on planar graphs for parameter k. Moreover, we prove a lower bound to polynomial kernelization for SAVING ALL BUT k-VERTICES.

1 Introduction

The firefighter problem was introduced in [10] and can be used to model the spread of a fire, a virus, or an idea through a network. It is a dynamic problem defined as follows. Initially, a fire breaks out at some special vertex s of a graph. At each time step, we have to choose one vertex which will be protected by a firefighter. Then the fire spreads to all unprotected neighbors of the vertices on fire. The process ends when the fire can no longer spread, and then all vertices that are not on fire are considered as saved. The objective consists of choosing, at each time step, a vertex which will be protected by a firefighter such that a maximum number of vertices in the graph is saved at the end of the process.

The firefighter problem was proved to be NP-hard even for trees of maximum degree three [8] and cubic graphs [11]. From the approximation point of view, the firefighter problem is $\frac{e}{e-1}$-approximable on trees [3] and it is not $n^{1-\varepsilon}$-approximable on general graphs [1], if P$\neq$ NP. However, very little is known about the fixed parameter tractability of this problem. In [3], the authors give fixed-parameter tractable algorithms and polynomial kernels on trees for each of the following parameters: the number of saved leaves, the number of burned vertices, and the number of protected vertices.

In this paper, we consider the parameterized complexity of the firefighter problem on general graphs where, at each time step, b firefighters can be deployed. This problem, called SAVING k-VERTICES, is defined as follows. Given

⋆ Supported by the Australian Research Council.

T. Asano et al. (Eds.): ISAAC 2011, LNCS 7074, pp. 643–652, 2011.

a graph $G = (V, E)$, an initially burned vertex $s \in V$, and two integers $b \geq 1$ and $k \geq 0$, can we protect b vertices at each time step such that at least k vertices are saved at the end of the process? The dual problem, Saving All But k-Vertices, is also studied and asks whether we can protect b vertices at each time step such that at most k vertices are burned at the end of the process. We show that Saving k-Vertices is W[1]-hard for parameter k when b is fixed. In contrast, Saving All But k-Vertices is fixed-parameter tractable for parameter k when b is fixed and it is W[1]-hard for parameter k when b is part of the input. Saving k-Vertices is proved to be fixed-parameter tractable when parameterized by k and the treewidth of the graph or when restricted to planar graphs and parameterized by k. Both problems, Saving All But k-Vertices and Saving All But k-Vertices, admit a kernel when parameterized by k and the vertex cover number. We also show that Saving All But k-Vertices parameterized by k and b does not admit polynomial kernels unless coNP$\subseteq$NP/poly. Our results are summarized in Table 1.

Our paper is organized as follows. Definitions, terminology and preliminaries are given in Section 2. In Section 3, we give several parameterized tractability results, and polynomial kernelization feasibility is studied in Section 4. Conclusions are given in Section 5. Due to the space limit, some proofs are omitted and could be found in the extended version.

Table 1. Summary of results. The vertex cover number is denoted by τ, and the treewidth by tw. Results in bold font are proved in this paper; results in italic are a direct consequence of these last results. Notice that vertex cover number is larger than treewidth.

	Parameter(s)	**k**	**k + b**	**k + tw**	**k + τ**
Saving k-Vertices	Tractability	*W[1]-hard*, **XP** **FPT for planar graphs**	**W[1]-hard**, *XP*	**FPT**	**FPT**
	Poly Kernel?	*no*	*no*	open	open
Saving All But k-Vertices	Tractability	**W[1]-hard, XP**	**FPT**	open	**FPT**
	Poly Kernel?	*no*	**no**	no [5]	open

2 Preliminaries

Graph terminology. All graphs in this paper are undirected, connected, finite and simple. Let $G = (V, E)$ be a graph. An edge in E between vertices $u, v \in V$ will be denoted by uv. The *degree* of a vertex $u \in V$, denoted by $deg(v)$, is the number of edges incident to u. The *open* (resp. *close*) *neighborhood* of a vertex $v \in V$ is the set $N(v) = \{u \in V : uv \in E\}$ (resp. $N[v] = N(v) \cup \{v\}$). Given a subset $S \subseteq V$, the open (resp. close) neighborhood of S is the set $N(S) = \bigcup_{u \in S} N(u)$ (resp. $N[S] = N(S) \cup S$). We denote by $d(u, v)$ the minimum length of a path with endpoints $u, v \in V$. Throughout this paper, the vertex cover number will be denoted by $\tau(G)$ or τ and the treewidth by $tw(G)$ or tw.

Parameterized complexity. Here we only give the basics notions on parameterized complexity, for more background the reader is referred to [6,9]. The parameterized complexity is a framework which provides a new way to express the computational complexity of problems. A problem parameterized by k is called *fixed-parameter tractable* (fpt) if there exists an algorithm (called an fpt algorithm) that solves it in time $f(k).n^{O(1)}$ (fpt-time). The function f is typically super-polynomial and only depends on k. In other words, the combinatorial explosion is confine into f. A parameterized problem P with parameter k will be denoted by (P, k). The XP class is the set of parameterized problems (P, k) that can be solved in time $n^{g(k)}$ for a given computable function g.

One of the main tool to design such algorithms is the *kernelization*. A kernelization algorithm transforms in polynomial time an instance I of a given problem parameterized by k into an equivalent instance I' of the same problem parameterized by $k' \leq k$ such that $|I'| \leq g(k)$ for some computable function g. The instance I' is called a *kernel* of size $g(k)$ (if g is a polynomial then I' is a *polynomial kernel*). By applying any (even exponential) algorithm to a kernel of a given problem, we can derive an fpt algorithm for that problem.

Conversely to the previous approach, we can prove the parameterized intractability of a problem. To this end, we need to introduce the notion of *parameterized reduction*. An fpt-reduction is an algorithm that reduces any instance I of a problem with parameter k to an equivalent instance I' with parameter $k' = g(k)$ in fpt-time for some function g. The basic class of parameterized intractability is $W[1]$ and there is a good reason to believe that $W[1]$-hard problems (according to the fpt-reduction) are unlikely to be FPT. We have the following inclusions FPT $\subseteq$ W[1] $\subseteq$ XP.

Very recently a new result [2] has been introduced for proving the non existence of a polynomial kernel under a reasonable complexity hypothesis. This result is based on the notion of *OR-composition* of parameterized problems. A problem P parameterized by k is OR-compositional if there exists a polynomial algorithm that receives as inputs a finite sequence $(I_1, k_1), \ldots, (I_N, k_N)$ of instances of (P, k) such that $k_1 = \ldots = k_N$. The algorithm is required to output an instance (I, k) of (P, k) such that $k = k_1^{O(1)}$ and (I, k) is a yes-instance if and only if there is some $i \in \{1, \ldots, N\}$ such that (I_i, k_i) is a yes-instance. We have the following theorem.

Theorem 1. [2] *Let (P, k) be a parameterized problem. If P is* NP*-complete and (P, k) is OR-compositional then (P, k) has no polynomial kernel unless* coNP$\subseteq$NP/poly.

Problems definition. In order to define the firefighter problem, we use an undirected graph $G = (V, E)$ and notations of [1]. Each vertex in the graph can be in exactly one of the following states: *burned*, *saved* or *vulnerable*. A vertex is said to be burned if it is on fire. We call a vertex saved if it is either protected by a firefighter — that is the vertex cannot be burned in subsequent time steps — or if all paths from any burned vertex to it contains at least one protected vertex. Any vertex which is neither saved nor burned is called vulnerable. At time step $t = 0$, all vertices are vulnerable, except vertex s, which is burned. At each time $t > 0$, at most b vertices can be protected by firefighters and any vulnerable vertex v which is adjacent to a burned vertex u becomes burned at time $t + 1$, unless it is protected at time step t. Burned and saved vertices remain burned and saved, respectively.

Given a graph $G = (V, E)$ and a vertex s initially on fire, a *protection strategy* is a set $\Phi \subseteq V \times T$ where $T = \{1, 2, \ldots, |V|\}$. We say that a vertex v is protected at time $t \in T$ according to the protection strategy Φ if $(v, t) \in \Phi$. A protection strategy is *valid* with respect to a budget b, if the following two conditions are satisfied:

1. if $(v, t) \in \Phi$ then v is not burned at time t;
2. let $\Phi_t = \{(v, t) \in \Phi\}$; then $|\Phi_t| \leq b$ for $t = 1, \ldots, |V|$.

Thus at each time $t > 0$, if a vulnerable vertex v is adjacent to at least one burned vertex and $(v, t) \notin \Phi$, then v gets burned at time $t + 1$. We now define in the following the problems we study.

Saving k-Vertices
Input: A graph $G = (V, E)$, a burned vertex $s \in V$ and two integers k and b.
Question: Is there a valid strategy Φ with respect to budget b that saves at least k vertices?

Saving All But k-Vertices
Input: A graph $G = (V, E)$, a burned vertex $s \in V$ and two integers k and b.
Question: Is there a valid strategy Φ with respect to budget b where at most k vertices are burned?

Remark 1. For the Saving k-Vertices problem, we may only consider instances for which any valid strategy $\Phi \subseteq V \times T$ is such that $|\Phi| < k$, otherwise the answer is clearly *yes*.

Remark 2. For the Saving All But k-Vertices problem, we may assume that for any valid strategy $\Phi \subseteq V \times T$, if $(v, t) \in \Phi$ then $t < k$, otherwise the answer is necessarily *no*. Indeed, there is at least one newly burned vertex at each time step, then if we protect a vertex at time step $t \geq k$ there will be at least k burned vertices.

3 Parameterized Tractability

We first show that Saving k-Vertices and its dual Saving All But k-Vertices are both in XP but are fixed-parameter intractable even for bipartite graphs.

Theorem 2. Saving k-Vertices *is solvable in time* $n^{O(k)}$.

Proof. It follows from Remark 1 that the algorithm only have to try each of the $\binom{n}{k}$ possible sets of protected vertices. Notice that all of these configurations does not necessarily correspond to a valid strategy. However, one can check if a set $D \subseteq V$ corresponds to a valid strategy with the following procedure. Let r_i be the number of firefighters we did not use from time step 1 to $i - 1$ ($r_1 = 0$), and $L_i = \{v \in V : d(s,v) = i\}$. For each time step $i = 1, \ldots, k$, protect vertices in $D \cap L_i$. If $|D \cap L_i| > b + r_i$ then this strategy is not valid. Otherwise, set $r_{i+1} = r_i + (b - |D \cap L_i|)$. It follows that the running time is $n^{O(k)}$. □

Theorem 3. Saving All But k-Vertices *is solvable in time* $n^{O(k)}$.

Proof. First of all, we need to introduce the notion of *valid burning set*. Given a graph $G = (V, E)$ and an initially burned vertex $s \in V$, a valid burning set is a subset $B \subseteq V$ with $s \in B$ such that there exists a valid strategy for which, at the end of the process, the burned vertices are exactly those in B. We have the following lemma.

Lemma 1. *Let* $G = (V, E)$ *be a graph with an initially burned vertex* $s \in V$*. Verifying if a subset* $B \subseteq V$ *is a valid burning set can be done in linear time.*

Proof. It follows from Remark 2 that we only have to considers the first k time steps. Notice that the set of protected vertices must be exactly $N(B)$. Moreover, given a vertex $v \in N(B)$ this vertex has a *due date*: it has to be protected before or at time step $d(s,v)$. Hence, at each time step $t = 1, \ldots, k$, it suffices to protect all the vertices in $N(B)$ with due dates equal to t. If there are more firefighters than vertices with due date equal to t then protect vertices with due date equal to $t+1$, then vertices with due date equal to $t+2$ and so on until there are no more firefighters. Clearly, if all vertices in $N(B)$ are protected using the previous procedure then the answer is *yes*. However, if, at a given time step t, there are less firefighters than vertices with due date equal to t the answer is *no*. □

The algorithm then proceeds as follows. For each subset $B \subseteq V$ with $|B| = k+1$ if B is a valid burning set then the answer is *yes*. If no valid burning set were found then the answer is *no*. It follows from lemma 1 that the running time is $n^{O(k)}$. □

The following result was also proved independently by Cygan *et al.* [5].

Theorem 4. Saving k-Vertices *parameterized by* k *is* W[1]-hard *even for bipartite graphs with a fixed budget.*

Proof. We construct an fpt-reduction from the W[1]-hard problem Multi-Colored Clique [12] to Saving k-Vertices. We first recall the definition of the former problem.

Multi-Colored Clique
Input: A graph $G = (V, E)$, an integer k, and a proper k-coloring of G (*i.e.*, every two adjacent vertices have different colors).
Parameter: k
Question: Is there a k-clique (*i.e.*, a complete subgraph on k vertices) in G?

Let I be an instance of Multi-Colored Clique consisting of a graph $G = (V, E)$, an integer k, and a proper k-coloring. We construct an instance I' of Saving k-Vertices consisting of a graph $G' = (V', E')$, a burned vertex $s \in V'$, and two integers k' and b as follows. Set $k' = \binom{k}{2} + k$ and $b = 1$. We construct G' from G as follows: add a new vertex s; add k copies of V denoted as $V_1, \ldots, V_k$; join s to every vertex of V_1; for $i = 1, \ldots, k$, join every vertex of V_i to every vertex of V_{i+1}; join every vertex of V_k to every vertex of V; remove every edge $uv \in E$ and add an edge-vertex x_{uv} adjacent to u and v where $u, v \in V$.

Suppose that there is a clique $C \subseteq V$ of size k in G. Let Φ be the following valid strategy for I': at each time step $i = 1, \ldots, k$, protect a vertex $v \in V$ of G' such that $v \in C$. At time step $k + 1$, Φ protects any non-burned vertex. Clearly, Φ saves at least k vertices and $\binom{k}{2}$ edge-vertices.

Conversely, suppose that a valid strategy Φ saves at least k' vertices in G'. We may assume that Φ protects no vertex in $V_c = V_1 \cup \ldots \cup V_k$. Indeed, suppose that Φ protects vertices in V_c. Notice that, at time step k, all vertices in V_c are burned except the protected ones. Consider now the strategy Φ' obtained from Φ that protects vertices in $V' \setminus V_c$ instead of those in V_c. Thus, Φ' saves at least the same number of vertices. Moreover, we may assume that Φ' protects no edge-vertex x_{uv}, otherwise we could protect either u or v instead of x_{uv} in order to save at least the same number of vertices. It follows that Φ' protects exactly k vertices of V in G'. Since we save at least $\binom{k}{2} + k$, the corresponding set C is a clique in G of size k. □

Theorem 5. Saving All But k-Vertices *parameterized by* k *is* W[1]-hard *even for bipartite graphs.*

Proof. We construct an fpt-reduction from the W[1]-hard problem Regular Clique [12] to Saving All But k-Vertices. We first give the definition of the former problem.

Regular Clique
Input: A regular graph $G = (V, E)$ and an integer k.
Parameter: k
Question: Is there a k-clique in G?

In this reduction, a *busy* gadget denotes a $(b+k)$-star with center c (*i.e.*, a tree with one internal vertex c and $b+k$ leaves). Attaching a busy gadget to a vertex v means to create a copy of a $(b+k)$-star and makes c adjacent to v. Thus, if v is burning at a given time step then c has to be protected, otherwise more than k vertices would burn.

Let I be an instance of Regular Clique consisting of a d-regular graph $G = (V, E)$ and an integer k. We construct an instance I' of Saving All But k-Vertices consisting of a graph $G' = (V', E')$, a burned vertex $s \in V'$, and two integers k' and b as follows. Set $k' = k$ and $b = b_1 + b_2$ where $b_1 = k(n-k)$ and $b_2 = kd - \binom{k}{2}$. We construct G' from G as follows: add a new vertex s adjacent to all vertices of G; attach $b_1 + b_2 - (n-k)$ busy gadgets to s; attach $n-k$ busy gadgets to every vertex in V; remove every edge $uv \in E$ and add an edge-vertex x_{uv} adjacent to u and v (see Figure 1). Notice that, at time step 1, there are only $n-k$ firefighters that can be placed freely because of the busy gadgets.

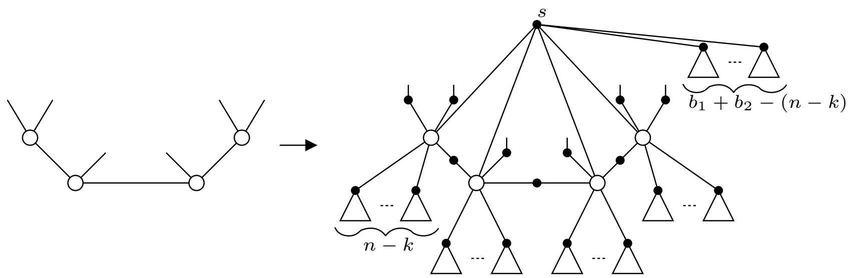

Fig. 1. The construction of G'. Added vertices are black and a triangle represents a busy gadget.

Suppose that we have a k-clique in G. Let Φ be the following valid strategy for I': at time step 1, the strategy uses the $n - k$ remaining firefighters to protect all the original vertices V in G' except those in the k-clique. At time step 2, all the k vertices in the clique are burned. Since there are $n - k$ busy gadgets attached to each vertex in the k-clique we need to protect $b_1 = k(n-k)$ vertices. Moreover, there are $kd - \binom{k}{2}$ edge-vertices adjacent to the vertices in the clique, since it remains $b_2 = b - b_1$ firefighters we can protect them all. Hence, no more than k vertices are burned at the end of the process.

Conversely, suppose that there is no k-clique in G. A time step 1, any valid strategy has to place the $n - k$ remaining firefighters on vertices that are not edge-vertices otherwise at least $k' + 1$ vertices will burn. At time step 2, since there is no k-clique, there will be at least $kd - \binom{k}{2} + 1$ edge-vertices adjacent to the k burned vertices. For the same reason as before, it remains $b_2 = b - b_1$ firefighters which is not enough to protect these edge-vertices. Therefore, given any valid strategy there will be at least k' burned vertices. □

We note that the parameters in the reduction used in Theorem 5 are linearly related. Since Regular Clique cannot be solved in time $n^{o(k)}$ unless FPT = M[1] [12], we obtain the following lower bound that shows that the algorithm given in Theorem 3 is optimal.

Corollary 1. Saving All But k-Vertices *cannot be solved in time* $n^{o(k)}$ *unless* FPT = M[1].

The following results show that Saving k-Vertices and Saving All But k-Vertices are fixed-parameter tractable in several cases.

Theorem 6. Saving All But k-Vertices *parameterized by* k *and the budget* b *is* FPT.

Proof. We describe a recursive algorithm that solves Saving All But k-Vertices in time $O^*(2^{k^2(b+1)+kb})$. Let $G = (V, E)$ be a graph with an initially burned vertex $s_0 \in V$. We may assume that $|N(s_0)| < b + k$ otherwise the answer is *no*. Since a solution of the problem will protect none or up to b vertices of $N(s_0)$, the algorithm must decide which of the $\sum_{i=0}^{b} \binom{b+k}{i}$ possible subsets of $N(s_0)$ to protect at time step 1. At time step 2, the newly burned vertices are exactly those in $N(s_0)$ that were not protected. Notice that we can merge every burned vertex into a single burned vertex s_1 without changing the answer of the instance. We may also assume that $|N(s_1)| < b + k + r$ where r is the number of firefighters we did not use in the previous time step. The algorithm has now $\sum_{i=0}^{b+r} \binom{b+k+r}{i}$ possible subsets to protect at time step 2. Clearly, we may apply the previous procedure recursively. Moreover, it follows from Remark 2 that there are at most k recursive calls. The value of r is then at most bk, and the running time of the algorithm is $O^*(2^{k(b+k+r)}) = O^*(2^{k^2(b+1)+kb})$ □

We use the Monadic Second Order Logic formulation of Saving k-Vertices and theorems from [4] and [7] to prove the following theorems. These were also proved independently in [5].

Theorem 7. Saving k-Vertices *parameterized by k is* FPT *for planar graphs.*

Theorem 8. Saving k-Vertices *parameterized by the treewidth and k is* FPT.

4 Kernelization Feasibility

In this section, we provide a kernelization for Saving k-Vertices (resp. Saving All k-Vertices) when parameterized by τ and k. Moreover, we show that Saving All But k-Vertices parameterized by k and b does not admit a polynomial kernel unless coNP⊆NP/poly.

Theorem 9. Saving k-Vertices *admits a kernel of size at most* $O(2^\tau k)$.

Proof. Let $G = (V, E)$ be a graph where the fire breaks out at vertex $s \in V$ and there are b firefighters available at each time step. A set $S \subseteq V$ is called a *twins set* if for every $v, u \in S$, $v \neq u$, we have $N(u) = N(v)$ and $uv \notin E$. Consider the following reduction rule.

Rule: If there exists a twins set S such that $|S| \geq k + 1$ then delete $|S| - k$ vertices of S.

Let $G' = (V', E')$ be the graph obtained by iteratively applying the above rule to every twins set in G. Notice that the procedure runs in polynomial time. Let $C \subseteq V'$ be a minimum vertex cover and let $D = V' \setminus C$ be an independent set. The number of distinct twins set in D is at most 2^τ (one for each subset in C). Moreover, each twins set in G' has at most k vertices. Therefore, the size of the reduced instance is at most $O(2^\tau k)$.

Correctness of the rule: Suppose that there exists a valid strategy Φ that saves at least k vertices in G. We have the following observation.

Observation 1. *Let $G = (V, E)$ be a graph, $S \subseteq V$ be a twins set, and Φ be a valid strategy with respect to budget b that saves at least k vertices. If Φ protects a subset $S_1 \subseteq S$ then protecting any subset $S_2 \subseteq S$ instead of S_1 such that $|S_2| = |S_1|$ leads to a valid strategy Φ' that saves exactly the same number of vertices.*

It follows from Observation 1 that if Φ protects a vertex in a twins set that has been deleted by the reduction rule then we can protect instead any other non-deleted vertex in the same twins set. Moreover, it follows from Remark 1 that Φ protects no more than k vertices in G. Since there are k vertices in any twins set in G', we can always apply Observation 1. Hence there is a strategy Φ' for the reduced instance that saves at least k vertices in G'. Conversely, if a strategy saves at least k vertices in G' then this strategy clearly saves at least k vertices in G. □

Theorem 10. SAVING ALL BUT k-VERTICES *admits a kernel of size at most* $O(2^\tau k\tau)$.

Proof. First we may assume that $b < \tau$, otherwise it suffices to protect all the vertices in the vertex cover at time step 1 to stop the fire. The reduction is the same as the one describes in Theorem 9 but we use the following slightly different reduction rule.

Rule: Let $S \subseteq V$ be a twins set. If $|S| \geq kb$ then delete $|S| - kb$ vertices of S.

Similarly to the proof of Theorem 9, the size of the kernel is $O(2^\tau kb) = O(2^\tau k\tau)$.

Correctness of the rule: It follows from Remark 2 that there are at most kb protected vertices in any twins set at the end of the process. Using the same argument as in Theorem 9, the result follows. □

Theorem 11. SAVING ALL BUT k-VERTICES *parameterized by k and the budget b has no polynomial kernel unless* coNP⊆NP/poly.

Proof. We show that SAVING ALL BUT k-VERTICES is OR-compositional. Let $I_1 = (G_1, s_1, k_1, b_1), ..., I_N = (G_N, s_N, k_N, b_N)$ be a sequence of SAVING ALL BUT k-VERTICES instances with $k_1 = \ldots = k_N$ and $b_1 = \ldots = b_N$. First we may assume that $N < 2^{p(b_1,k_1)}$ where $p(b_1, k_1) = k_1^2(b_1 + 1) + k_1 b_1$, otherwise we could apply the fpt algorithm of Theorem 6 to each input instance. The output instance is the first input instance for which the fpt algorithm return *yes* if such instance exists, and the last instance otherwise. Clearly, the output instance is *yes* if and only if there exists a *yes* instance in the input sequence. Moreover, the procedure runs in time $O(N.2^{p(b_1,k_1)}n) = O(N^2 n)$.

When $N < 2^{p(b_1,k_1)}$ we construct the output instance $I = (G, s, k, b)$ as follows. Set $k = k_1 + \lceil \log_2 N \rceil$ and $b = b_1$. Build a perfect binary tree (*i.e.,* a tree in which every vertex other than the leaves has two children and all leaves are at the same distance from the root) of height $\lceil \log_2 N \rceil$. For each $i = 1, \ldots, N$, identify a leave of the tree with vertex s_i. Identify remaining leaves with a copy of s_N. Attach $b - 1$ busy gadgets to every non-leaf vertex of the tree. By construction, the only burned vertices from time step 1 to $\lceil \log_2 N \rceil$ are on a path from s to some s_i. It follows that there exists a yes-instance I_i for some i if and only if there exists a valid strategy for I such that no more than $k = k_i + \lceil \log_2 N \rceil$ vertices are burned at then end of the process. Since $k = O(k_1 + p(b_1, k_1))$ and $b = b_1$ it follows that SAVING ALL BUT k-VERTICES is OR-compositional. Using Theorem 1 and the NP-completeness of SAVING ALL BUT k-VERTICES the result follows. □

5 Conclusion

In this paper, we study the parameterized complexity of the firefighter problem on general graphs when more than one firefighter is available at each time step. We establish some tractable and intractable cases and study the existence of a polynomial kernel. Several interesting questions remain open. Does Saving k-Vertices or Saving All But k-Vertices admit a polynomial kernel for parameters τ and k? Is Saving All But k-Vertices fixed-parameter tractable when parameterized by tw and k?

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. (eds.) ISAAC 2009. LNCS, vol. 5878, pp. 974–983. Springer, Heidelberg (2009)
2. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On Problems without Polynomial Kernels. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 563–574. Springer, Heidelberg (2008)
3. Cai, L., Verbin, E., Yang, L.: Firefighting on Trees $(1 - 1/e)$–Approximation, Fixed Parameter Tractability and a Subexponential Algorithm. In: Hong, S.-H., Nagamochi, H., Fukunaga, T. (eds.) ISAAC 2008. LNCS, vol. 5369, pp. 258–269. Springer, Heidelberg (2008)
4. Courcelle, B.: The monadic second-order logic of graphs. III. tree-decompositions, minors and complexity issues. RAIRO Informatique Théorique Appliquée 26(3), 257–286 (1992)
5. Cygan, M., Fomin, F., Van Leeuwen, E.J.: Parameterized complexity of firefighting revisited. In: Proceedings of the 6th International Symposium on Parameterized and Exact Computation (IPEC 2011) (to appear, 2011)
6. Downey, R.G., Fellows, M.R.: Parameterized complexity. Springer, Heidelberg (1999)
7. Fellows, M.R., Szeider, S., Wrightson, G.: On finding short resolution refutations and small unsatisfiable subsets. Theoretical Computer Science 351, 351–359 (2006)
8. Finbow, S., King, A., MacGillivray, G., Rizzi, R.: The firefighter problem for graphs of maximum degree three. Discrete Mathematics 307(16), 2094–2105 (2007)
9. Flum, J., Grohe, M.: Parameterized complexity theory. Springer, Heidelberg (2006)
10. Hartnell, B.: Firefighter! an application of domination, Presentation. In: 10th Conference on Numerical Mathematics and Computing. University of Manitoba in Winnipeg, Canada (1995)
11. King, A., MacGillivray, G.: The firefighter problem for cubic graphs. Discrete Mathematics 310(3), 614–621 (2010)
12. Mathieson, L., Szeider, S.: Parameterized Graph Editing with Chosen Vertex Degrees. In: Yang, B., Du, D.-Z., Wang, C.A. (eds.) COCOA 2008. LNCS, vol. 5165, pp. 13–22. Springer, Heidelberg (2008)