Firefighting on Trees Beyond Integrality Gaps

David Adjiashvili* Andrea Baggio† Rico Zenklusen‡

January 5, 2016

Abstract

The Firefighter problem and a variant of it, known as Resource Minimization for Fire Containment (RMFC), are natural models for optimal inhibition of harmful spreading processes. Despite considerable progress on several fronts, the approximability of these problems is still badly understood. This is the case even when the underlying graph is a tree, which is one of the most-studied graph structures in this context and the focus of this paper. In their simplest version, a fire spreads from one fixed vertex step by step from burning to adjacent non-burning vertices, and at each time step B-many non-burning vertices can be protected from catching fire. The Firefighter problem asks, for a given B, to maximize the number of vertices that will not catch fire, whereas RMFC (on a tree) asks to find the smallest B which allows for saving all leaves of the tree. Prior to this work, the best known approximation ratios were an $O(1)$-approximation for the Firefighter problem and an $O(\log^* n)$-approximation for RMFC, both being LP-based and matching the integrality gaps of two natural LP relaxations.

We improve on both approximations by presenting a PTAS for the Firefighter problem and an $O(1)$-approximation for RMFC, both qualitatively matching the known hardness results. Our results are obtained through a combination of the LP with several new techniques, which allow for efficiently enumerating subsets of super-constant size of a good solution to reduce the integrality gap of the LPs.

1 Introduction

The Firefighter problem was introduced by Hartnell [22] as a natural model for optimal inhibition of harmful spreading phenomena on a graph. Despite considerable interest in the problem and progress on several fronts, our understanding of how well this and related problems can be approximated is still very limited. Interestingly, this is even true when the underlying graph is a spanning tree, which is one of the most-studied graph structures in this context and also the focus of this paper.

The Firefighter problem on trees is defined as follows. We are given a graph $G = (V, E)$ which is a spanning tree and a vertex $r \in V$, called *root*. The problem is defined over discretized time steps. At time 0, a fire starts at r and spreads step by step to neighboring vertices. During each time step $1, 2, \ldots$ an arbitrary non-burning vertex u can be *protected*, which prevents u from burning in all future time steps. In its original form, the goal is to find a protection strategy that minimizes the number of vertices that will catch fire. A closely related problem, called *Resource Minimization for Fire Containment (RMFC)* on trees, was introduced by Chalermsook and Chuzhoy [8]. Here the task is to determine the smallest number $B \in \mathbb{Z}_{>0}$ such that if one can protect B vertices at each time step (instead of just 1), then there is a protection strategy

*Department of Mathematics, ETH Zurich, Zurich, Switzerland. Email: david.adjiashvili@ifor.math.ethz.ch. Supported by Seed Project "Risk Protection in Complex Networks" of ETH Zurich Risk Center.

†Department of Mathematics, ETH Zurich, Zurich, Switzerland. Email: andrea.baggio@ifor.math.ethz.ch. Supported by EU grant FP7-PEOPLE-2012-ITN no. 316647, "Mixed-Integer Nonlinear Optimization".

‡Department of Mathematics, ETH Zurich, Zurich, Switzerland. Email: ricoz@math.ethz.ch.

such that none of the leaves of the tree will catch fire. In this context, one typically refers to B as the *number of firefighters*.

Both the Firefighter problem and RMFC—both restricted to trees as defined above—are known to be computationally hard problems. More precisely, Finbow, King, MacGillivray and Rizzi [16] showed NP-hardness for the Firefighter problem on trees with maximum degree three. For RMFC on trees, it is NP-hard to decide whether one firefighter, i.e., $B = 1$, is sufficient [25]; thus, unless P = NP, there is no (efficient) approximation algorithm with an approximation factor strictly better than 2.

On the positive side, several approximation algorithms have been suggested for the Firefighter problem and RMFC. Hartnell and Li [23] showed that a natural greedy algorithm is a $\frac{1}{2}$-approximation for the Firefighter problem. This approximation guarantee was later improved by Cai, Verbin and Yang [6] to $1 - \frac{1}{e}$, using a natural linear programming (LP) relaxation and dependent randomized rounding. It was later observed by Anshelevich, Chakrabarty, Hate and Swamy [1] that the Firefighter problem on trees can be interpreted as a monotone submodular function maximization (SFM) problem subject to a partition matroid constraint. This leads to alternative ways to obtain a $(1 - \frac{1}{e})$-approximation by using a recent $(1 - \frac{1}{e})$-approximation for monotone SFM subject to a matroid constraint [33, 11]. The factor $1 - \frac{1}{e}$ was later only improved for various restricted tree topologies (see [24]) and hence, for arbitrary trees, this is the best known approximation factor to date.

For RMFC on trees, Chalermsook and Chuzhoy [8] presented an $O(\log^* n)$-approximation, where $n = |V|$ is the number of vertices.[1] Their algorithm is based on a natural linear program which is a straightforward adaptation of the one used in [6] to get a $(1 - \frac{1}{e})$-approximation for the Firefighter problem on trees.

Whereas there are still considerable gaps between current hardness results and approximation algorithms for both the Firefighter problem and RMFC on trees, the currently best approximations essentially match the integrality gaps of the underlying LPs. More precisely, the LP used for the Firefighter problem on trees has an integrality gap of $1 - \frac{1}{e} + o(n)$ as shown in [6]. For RMFC on trees, the integrality gap of the underlying LP is $\Theta(\log^* n)$ [8].

It remained open to what extend these integrality gaps may reflect the approximation hardnesses of the problems. This question is motivated by two related problems whose hardnesses of approximation indeed matches the above-mentioned integrality gaps for the Firefighter problem and RMFC. In particular, many versions of monotone SFM subject to a matroid constraint—which we recall was shown in [1] to capture the Firefigther problem on trees as a special case—are hard to approximate up to a factor of $1 - 1/e - \epsilon$ for any constant $\epsilon > 0$. This includes the problem of maximizing an explicitly given coverage function subject to a single cardinality constraint, as shown by Feige [15]. Moreover, as highlighted in [8], the Asymmetric k-center problem is similar in nature to RMFC, and has an approximation hardness of $\Theta(\log^* n)$.

The goal of this paper is to fill the gap between current approximation ratios and hardness results for the Firefighter problem and RMFC on trees. In particular, we present approximation ratios that nearly match the hardness results, thus showing that both problems can be approximated to factors that are substantially better than the integrality gaps of the natural LPs. Our results are based on several new techniques, which may be of independent interest.

1.1 Our results

Our main results show that both the Firefighter problem and RMFC admit strong approximations that essentially match known hardness bounds, thus implying that both problems can be approximated well below the integrality gaps of their natural LPs.

In particular, we obtain the following result for RMFC.

[1] $\log^* n$ denotes the minimum number k of logs of base two that have to be nested such that $\underbrace{\log \log \ldots \log}_{k \text{ logs}} n \leq 1$.

Theorem 1. *There is a* 12*-approximation for RMFC.*

Recalling that RMFC is hard to approximate to a factor better than 2, the above result is optimal up to a constant factor, and improves on the previously best $O(\log^* n)$-approximation of Chalermsook and Chuzhoy [8].

Moreover, our main result for the Firefighter problem is the following, which, in view of NP-hardness of the problem, is essentially best possible in terms of approximation guarantee.

Theorem 2. *There is a PTAS for the Firefighter problem on trees.*[2]

Notice that the Firefighter problem does not admit an FPTAS[3] unless $\mathrm{P} = \mathrm{NP}$ since the optimal value of any Firefighter problem on a tree of n vertices is bounded by $O(n)$.[4] We introduce several new techniques that allow us to obtain approximation factors below the integrality gaps, which have been a barrier for previous approaches. We start by providing an overview of these techniques.

Despite the fact that we obtain approximation factors below the integrality gaps, the natural LPs plays a central role in our approaches. We start by introducing general transformations that allow for transforming the Firefighter problem and RMFC into a more compact and better structured form, only losing small factors in terms of approximability. These transformations by themselves do not decrease the integrality gaps. However, they allow us to identify small substructures, over which we can optimize efficiently, and having an optimal solution to these subproblems we can define a residual LP with small integrality gap.

Similar high-level approaches, like guessing a constant-size but important subset of an optimal solution are well-known in various contexts to decrease integrality gaps of natural LPs; the best-known example may be classic PTASs for the knapsack problem, where the integrality gap of the natural LP can be decreased to an arbitrarily small constant by first guessing a constant number of heaviest elements of an optimal solution. However, our approach differs substantially from this standard enumeration idea. Apart from the above-mentioned transformations which, as we will show later, already lead to new results for both RMFC and the Firefighter problem, we will introduce new combinatorial approaches to gain information about a *super-constant* subset of an optimal solution. In particular, for the RMFC problem we define a recursive enumeration algorithm which, despite being very slow for enumerating all solutions, can be shown to reach a good subsolution within a small recursion depth that can be reached in polynomial time. For the Firefighter problem, we use a well-chosen enumeration procedure to identify a polynomial number of additional constraints to be added to the LP, that decrease its integrality gap down to $1+\epsilon$.

1.2 Further related results

Iwaikawa, Kamiyama and Matsui [24] showed that the approximation guarantee of $1-\frac{1}{e}$ can be improved for some restricted families of trees. The best approximation guarantee the authors prove is 0.7144 for trees with maximum degree 4. Anshelevich, Chakrabarty, Hate and Swamy [1] studied the approximability of the Firefighter problem in general graphs, which they prove admits no $n^{1-\epsilon}$-approximation for any $\epsilon > 0$, unless P$=$NP. In a different model, where the protection also spreads through the graph (the *Spreading Model*), the authors show that the problem admits a polynomial $(1-\frac{1}{e})$-approximation on general graphs. For RMFC the authors prove the existence of a $O(\sqrt{n})$-approximation for general graphs and a $O(\log n)$-approximation for directed layered graphs. The latter result was obtained independently by Chalermsook and Chuzhoy [8]. Klein, Levcopoulos and Lingas [26] introduced a geometric variant of the Firefighter problem, proved its

[2] A polynomial time approximation scheme (PTAS) is an algorithm that, for any constant $\epsilon > 0$, returns in polynomial time a $(1-\epsilon)$-approximate solution.

[3] An FPTAS is a PTAS with running time polynomial in the input size and $\frac{1}{\epsilon}$.

[4] The nonexistence of FPTASs unless $\mathrm{P} = \mathrm{NP}$ can often be derived easily from strong NP-hardness. Notice that the Firefighter problem is indeed strongly NP-hard because its input size is $O(n)$, in which case NP-hardness is equivalent to strong NP-hardness.

NP-hardness and provided a constant-factor approximation algorithm. The Firefighter problem and RMFC are natural special cases of the Maximum Coverage Problem with Group Constraints (MCGC) [9] and the Multiple Set Cover problem (MSC) [13], respectively. The input in MCGC is a set system consisting of a finite set X of elements with nonnegative weights, a collection of subsets $\mathcal{S} = \{S_1, \cdots, S_k\}$ of X and an integer k. The sets in $\mathcal{S}$ are partitioned into groups $G_1, \cdots, G_l$. The goal is to pick a subset $H \subseteq \mathcal{S}$ of k sets from $\mathcal{S}$ whose union covers elements of total weight as large as possible with the additional constraint that $|H \cap G_j| \leq 1$ for all $j \in [l] := \{1, \ldots, l\}$. In MSC, instead of the fixed bounds for groups and the parameter k, the goal is to choose a subset $H \subseteq \mathcal{S}$ that covers X completely, and the goal is to minimize the $\max_{j \in [l]} |H \cap G_j|$. The Firefighter problem and RMFC can naturally be interpreted as special cases of the latter problems with a laminar set system $\mathcal{S}$.

The Firefighter problem admits polynomial time algorithms in some restricted classes of graphs. Finbow, King, MacGillivray and Rizzi [16] showed that, while the problem is NP-hard in trees with maximum degree three, when the fire starts at a vertex with degree two in a subcubic tree, the problem is solvable in polynomial time. Fomin, Heggernes and van Leeuwen [19] presented polynomial algorithms for interval graphs, split graphs, permutation graphs and P_k-free graphs.

Several sub-exponential exact algorithms were developed for the Firefighter problem on trees. Cai, Verbin and Yang [6] presented a $2^{O(\sqrt{n} \log n)}$-time algorithm. Floderus, Lingas and Persson [18] presented a simpler algorithm with a slightly better running time. A sub-exponential algorithm for general graphs in the spreading model and a constant-factor approximation in planar graphs for some sets of parameters were also presented.

Additional directions of research on the Firefighter problem include parameterized complexity (Cai, Verbin and Yang [6], Bazgan, Chopin and Fellows [3], Cygan, Fomin and van Leeuwen [12] and Bazgan, Chopin, Cygan, Fellows, Fomin and van Leeuwen [2]), generalizations to the case of many initial fires and many firefighters (Bazgan, Chopin and Ries [4] and Costa, Dantas, Dourado, Penso and Rautenbach [10]).

Lastly, let us review related work on the closely related problem of computing the *Survivability* of a graph. For a graph G and a parameter $k \in \mathbb{Z}_{\geq 0}$, the k-survivability of G is the average fraction of nodes that one can save with k firefighters in G, when the fire starts at a random node. Cai and Wang [7] first introduced this notion and proved that the 1-survivability of any n-node tree is $1 - o(1)$. The bound for trees was subsequently improved by Cai, Cheng, Verbin and Zhou [5], and the result was generalized to bounded treewidth graphs. Other classes of graphs that were studied include bounded degree graphs (Pralat [29] and Pralat [30]), planar graphs (Esperet, van den Heuvel, Maffray and Sipma [14] and Gordinowicz [20]) and directed graphs (Kong, Zhang and Wang [27]).

For further references we refer the reader to the survey of Finbow and MacGillivray [17].

1.3 Organization of the paper

We start by introducing the classic linear programming relaxations for the Firefighter problem and RMFC in Section 2. In Section 3 we outline our main techniques and algorithms. For brevity, some of the proofs and additional discussion are deferred to later sections, namely Section 4, providing details on a compression technique that is crucial for both our algorithms, Section 5, containing proofs for results related to the Firefighter problem, and Section 6, containing proofs for results related to RMFC.

Finally, Appendix A contains some basic reductions showing how to reduce different variations of the Firefighter problem to each other.

2 Classic LP relaxations and preliminaries

Interestingly, despite the fact that we obtain approximation factors below the integrality gaps, the natural LPs play a central role in our approaches. We thus start by introducing these LPs together with some basic notation and terminology.

Let $L \in \mathbb{Z}_{\geq 0}$ be the *depth* of the tree, i.e., the largest distance—in terms of number of edges—between r and any other vertex in G. Hence, after at most L time steps, the fire spreading process will halt. For $\ell \in [L] := \{1, \ldots, L\}$, let $V_\ell \subseteq V$ be the set of all vertices of distance ℓ from r, which we call the *ℓ-th level* of the instance. For brevity, we use $V_{\leq \ell} = \cup_{k=1}^{\ell} V_k$, and we define in the same spirit $V_{\geq \ell}$, $V_{<\ell}$, and $V_{>\ell}$. Moreover, we denote by $\Gamma \subseteq V$ the set of all leaves of the tree, and for any $u \in V$, the set $P_u \subseteq V \setminus \{r\}$ denotes the set of all vertices on the unique u-r path except for the root r.

The relaxation for RMFC used in [8] is the following:

$$\begin{array}{rrcll} \min & B & & & \\ & x(P_u) & \geq & 1 & \forall u \in \Gamma \\ & x(V_{\leq \ell}) & \leq & B \cdot \ell & \forall \ell \in [L] \\ & x & \in & \mathbb{R}_{\geq 0}^{V \setminus \{r\}}, & \end{array} \qquad (\text{LP}_{\text{RMFC}})$$

where $x(U) := \sum_{u \in U} x(u)$ for any $U \subseteq V \setminus \{r\}$. Indeed, if one enforces $x \in \{0,1\}^{V \setminus \{r\}}$ and $B \in \mathbb{Z}$ in the above relaxation, an exact description of RMFC is obtained where x is the characteristic vector of the vertices to be protected and B is the number of Firefighters: The constraints $x(P_u) \geq 1$ for $u \in \Gamma$ enforce that for each leaf u, a vertex between u and r will be protected, which makes sure that u will not be reached by the fire; moreover, the constraints $x(V_{\leq \ell}) \leq B \cdot \ell$ for $\ell \in [L]$ describe the vertex sets that can be protected given B firefighters per time step (see [8] for more details). Also, as already highlighted in [8], there is an optimal solution to RMFC (and also the Firefighter problem), that protects with the firefighters available at time step ℓ only vertices in V_ℓ. Hence, the above relaxation can be transformed into one with same optimal objective value by replacing the constraints $x(V_{\leq \ell}) \leq B \cdot \ell \;\; \forall \ell \in [L]$ by the constraints $x(V_\ell) \leq B \;\; \forall \ell \in [L]$.

The natural LP relaxation for the Firefighter problem, which leads to the currently best $(1 - 1/e)$-approximation presented in [6], is obtained analogously. Due to higher generality, and even more importantly to obtain more flexibility in reductions to be defined later, we work on a slight generalization of the Firefighter problem on trees, extending it in two ways:

(i) Weighted version: vertices $u \in V \setminus \{r\}$ have weights $w(u) \in \mathbb{Z}_{\geq 0}$, and the goal is to maximize the total weight of vertices not catching fire. In the classical Firefighter problem all weights are one.

(ii) General budgets/firefighters: We allow for having a different number of Firefighters at each time step, say $B_\ell \in \mathbb{Z}_{>0}$ Firefighters for time step $\ell \in [L]$.[5]

Indeed, the above generalizations are mostly for convenience of presentation, since general budgets can be reduced to unit budgets (see Appendix A for a proof):

Lemma 3. *Any weighted Firefighter problem on trees with n vertices and general budgets can be transformed efficiently into an equivalent weighted Firefighter problem with unit-budgets and $O(n^2)$ vertices.*

We also show in Appendix A that up to an arbitrarily small error in terms of objective, any weighted Firefighter instance can be reduced to a unit-weighted one. In what follows, we always assume to deal with a weighted Firefighter instance if not specified otherwise. Regarding the budgets, we will be explicit about whether we work with unit or general budgets, since some techniques are easier to explain in the unit-budget case, even though it is equivalent to general budgets by Lemma 3.

[5]Without loss of generality we exclude $B_\ell = 0$, since a level with zero budget can be eliminated through a simple contraction operation. For more details we refer to the proof of Theorem 4 which, as a sub-step, eliminates zero-budget levels.

An immediate extension of the LP relaxation for the unit-weighted unit-budget Firefighter problem used in [6]—which in turn is based on an IP formulation presented in [28]—leads to the following LP relaxation for the weighted Firefighter problem with general budgets. For $u \in V$, we denote by $T_u \subseteq V$ the set of all vertices in the subtree starting at u and including u, i.e., all vertices v such that the unique r-v path in G contains u.

$$\begin{aligned} \max \quad \sum_{u \in V \setminus \{r\}} x_u w(T_u) & & & \\ x(P_u) &\leq 1 & & \forall u \in \Gamma \\ x(V_{\leq \ell}) &\leq \sum_{i=1}^{\ell} B_i & & \forall \ell \in [L] \\ x &\in \mathbb{R}_{\geq 0}^{V \setminus \{r\}}. & & \end{aligned} \tag{$\mathrm{LP_{FF}}$}$$

The budget constraints are identical to RMFC, with the difference that the budget B_ℓ depends on the level ℓ. The constraints $x(P_u) \leq 1$ exclude redundancies, i.e., a vertex u is forbidden of being protected if another vertex above it, on the r-u path, is already protected. This elimination of redundancies allows for writing the objective function as shown above.

We recall that the integrality gap of $\mathrm{LP_{RMFC}}$ was shown to be $\Theta(\log^* n)$ [8], and the integrality gap of $\mathrm{LP_{FF}}$ is asymptotically $1 - 1/e$ (when $n \to \infty$) [6].

Throughout the paper, all logarithms are base 2 if not indicated otherwise. When using big-O and related notations (like $\Omega, \Theta, \ldots$), we will always be explicit about the dependence on small error terms ϵ—as used when talking about $(1-\epsilon)$-approximations—and not consider it to be part of the hidden constant. To make statements where ϵ is part of the hidden constant, we will use the notation O_ϵ and likewise $\Omega_\epsilon, \Theta_\epsilon, \ldots$.

3 Overview of techniques and algorithms

In this section, we present our main technical contributions and outline our algorithms. We start by introducing a compression technique in Section 3.1 that works for both RMFC and the Firefighter problem and allows for transforming any instance to one on a tree with only logarithmic depth. One key property we achieve with compression, is that we can later use (partial) enumeration techniques with exponential running time in the depth of the tree. However, compression in its own already leads to interesting results. In particular, it allows us to obtain a QPTAS for the Firefighter problem, and a quasipolynomial time 2-approximation for RMFC.[6] However, it seems highly non-trivial to transform these quasipolynomial time procedures to efficient ones.

To obtain the claimed results, we develop two (partial) enumeration methods to reduce the integrality gap of the LP together with further techniques. In Section 3.2, we provide an overview of our PTAS for the Firefighter problem, and Section 3.3 presents our $O(1)$-approximation for RMFC.

3.1 Compression

Compression is a technique that is applicable to both the Firefighter problem and RMFC. It allows for reducing the depth of the input tree at a very small loss in the objective. We start by discussing compression in the context of the Firefighter problem.

To reduce the depth of the tree, we will first do a sequence of what we call *down-pushes*. Each down-push acts on two levels $\ell_1, \ell_2 \in [L]$ with $\ell_1 < \ell_2$ of the tree, and moves the budget B_{ℓ_1} of level ℓ_1 down

[6]The running time of an algorithm is *quasipolynomial* if it is of the form $2^{\mathrm{polylog}(\langle \mathrm{input} \rangle)}$, where $\langle \mathrm{input} \rangle$ is the input size of the problem. A QPTAS is an algorithm that, for any constant $\epsilon > 0$, returns a $(1-\epsilon)$-approximation in quasipolynomial time.

to ℓ_2, i.e., the new budget of level ℓ_2 will be $B_{\ell_1} + B_{\ell_2}$, and the new budget of level ℓ_1 will be 0. Clearly, down-pushes only restrict our options for protecting vertices. However, we can show that one can do a sequence of down-pushes such that first, the optimal objective value of the new instance is very close to the one of the original instance, and second, only $O(\log L)$ levels have non-zero budgets. Finally, levels with 0-budget can easily be removed through a simple contraction operation, thus leading to a new instance with only $O(\log L)$ depth.

Theorem 4 below formalizes our main compression result for the Firefighter problem, which we state for unit-budget Firefighter instances for simplicity. Since Lemma 3 implies that every general-budget Firefighter instance with n vertices can be transformed into a unit-budget Firefighter instance with $O(n^2)$ vertices—and thus $O(n^2)$ levels—Theorem 4 can also be used to reduce any Firefighter instance on n vertices to one with $O(\frac{\log n}{\delta})$ levels, by losing a factor of $1-\delta$ in terms of objective.

Theorem 4. *Let $\mathcal{I}$ be a unit-budget Firefighter instance on a tree with depth L, and let $\delta \in (0,1)$. Then one can efficiently construct a general budget Firefighter instance $\overline{\mathcal{I}}$ with depth $L' = O(\frac{\log L}{\delta})$, and such that the following holds, where $\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}}))$ and $\mathrm{val}(\mathsf{OPT}(\mathcal{I}))$ are the optimal values of $\overline{\mathcal{I}}$ and $\mathcal{I}$, respectively.*

(i) $\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}})) \geq (1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I}))$, *and*

(ii) any solution to $\overline{\mathcal{I}}$ can be transformed efficiently into a solution of $\mathcal{I}$ with same objective value.

For RMFC we can use a very similar compression technique leading to the following.

Theorem 5. *Let $G = (V,E)$ be a rooted tree of depth L. Then one can construct efficiently a rooted tree $G' = (V', E')$ with $|V'| \leq |V|$ and depth $L' = O(\log L)$, such that:*

(i) If the RMFC problem on G has a solution with budget $B \in \mathbb{Z}_{>0}$ at each level, then the RMFC problem on G' with non-uniform budgets, where level $\ell \geq 1$ has a budget of $B_\ell = 2^\ell \cdot B$, has a solution.

(ii) Any solution to the RMFC problem on G', where level ℓ has budget $B_\ell = 2^\ell \cdot B$, can be transformed efficiently into an RMFC solution for G with budget $2B$.

Interestingly, the above compression results already allow us to obtain strong quasipolynomial approximation algorithms for the Firefighter problem and RMFC, using dynamic programming. Consider for example the RMFC problem. We can first guess the optimal budget B, which can be done efficiently since $B \in \{1,\dots,n\}$. Consider now the instance G' claimed by Theorem 5 with budgets $B_\ell = 2^\ell B$. By Theorem 5 this RMFC instance is feasible and any solution to it can be converted to one of the original RMFC problem with budget $2B$. It is not hard to see that, for the fixed budgets B_ℓ, one can solve the RMFC problem on G' in quasipolynomial time using a bottom-up dynamic programming approach. More precisely, starting with the leaves and moving up to the root, we compute for each vertex $u \in V$ the following table. Consider a subset of the available budgets, which can be represented as a vector $q \in [B_1] \times \cdots \times [B_{L'}]$. For each such vector q we want to know whether or not using the sub-budget described by q allows for disconnecting u from all leaves below it. Since $L' = O(\log L)$ and the size of each budget B_ℓ is at most the number of vertices, the table size is quasipolynomial. Moreover, one can check that these tables can be constructed bottom-up in quasipolynomial time. Hence, this approach leads to a quasipolynomial time 2-approximation for RMFC, which is best possible in terms of approximation ratio unless $\mathrm{P} = \mathrm{NP}$ as mentioned previously. A similar dynamic programming approach for the Firefighter problem on a compressed instance leads to a QPTAS.

However, our focus is on efficient algorithms, and it seems non-trivial to transform the above quasipolynomial time dynamic programming approaches into efficient algorithms. To obtain our results, we therefore combine the above compression techniques with different approaches to be discussed next.

3.2 Overview of PTAS for Firefighter problem

Despite the fact that $\mathrm{LP}_{\mathrm{FF}}$ has a large integrality gap—which can be shown to be the case even after compression—it is a crucial tool in our PTAS.

We start by observing that for any vertex solution solution $x \in \mathbb{R}^{V\setminus\{r\}}$ to $\mathrm{LP}_{\mathrm{FF}}$, there is a small subset U of vertices of size at most the depth of the tree, such that one can easily get an integral solution whose objective value differs from the LP-value of x by at most the LP-contribution of U, which is $\sum_{u\in U} w(T_u)$. We will then introduce approaches to limit the LP-contribution of vertices in U.

Consider a general-budget Firefighter instance, and let x be a vertex solution to $\mathrm{LP}_{\mathrm{FF}}$. We say that a vertex $u \in V \setminus \{r\}$ is x-*loose*, or simply *loose*, if $u \in \operatorname{supp}(x) := \{v \in V \setminus \{r\} \mid x(v) > 0\}$ and $x(P_u) < 1$. Analogously, we call a vertex $u \in V \setminus \{r\}$ x-*tight*, or simply *tight*, if $u \in \operatorname{supp}(x)$ and $x(P_u) = 1$. Hence, $\operatorname{supp}(x)$ can be partitioned into $\operatorname{supp}(x) = V^{\mathcal{L}} \cup V^{\mathcal{T}}$, where $V^{\mathcal{L}}$ and $V^{\mathcal{T}}$ are the set of all loose and tight vertices, respectively.

Lemma 6. *Let x be a vertex solution to $\mathrm{LP}_{\mathrm{FF}}$ for a Firefighter problem with general budgets. Then the number of x-loose vertices is at most L, the depth of the tree.*

We observe next that to obtain a set U with the above-claimed properties, one can choose $U = V^{\mathcal{L}}$. Having a vertex solution x to $\mathrm{LP}_{\mathrm{FF}}$, we can consider a simplified LP obtained from $\mathrm{LP}_{\mathrm{FF}}$ by only allowing to protect vertices that are x-tight. A simple yet useful property of x-tight vertices is that for any $u, v \in V^{\mathcal{T}}$ with $u \neq v$ we have $u \notin P_v$. Indeed, if $u \in P_v$, then $x(P_u) \leq x(P_v) - x(v) < x(P_v) = 1$ because $x(v) > 0$. Hence, no two tight vertices lie on the same leaf-root path. Thus, when restricting $\mathrm{LP}_{\mathrm{FF}}$ to $V^{\mathcal{T}}$, the path constraints $x(P_u) \leq 1$ for $u \in \Gamma$ transform into trivial constraints requiring $x(v) \leq 1$ for $v \in V^{\mathcal{T}}$, and one can easily observe that the resulting constraint system is totally unimodular because it describes a laminar matroid constraint given by the budget constraints (see [32, Volume B] for more details on matroid optimization). Re-optimizing over this LP we get an integral solution of objective value at least $\sum_{u\in V\setminus\{r\}} x_u w(T_u) - \sum_{u\in V^{\mathcal{L}}} x_u w(T_u)$, because the restriction of x to $V^{\mathcal{T}}$ is still feasible for the new LP.

In particular, if $\sum_{u\in V^{\mathcal{L}}} x_u w(T_u)$ was at most $\epsilon \cdot \mathrm{val}(\mathsf{OPT})$, where $\mathrm{val}(\mathsf{OPT})$ is the optimal value of the instance, then this would lead to a PTAS. Clearly, this is not true in general, since it would contradict the $(1-\frac{1}{e})$-integrality gap of $\mathrm{LP}_{\mathrm{FF}}$. Thus, in the following, we will present techniques to limit the impact of the term $\sum_{u\in V^{\mathcal{L}}} x_u w(T_u)$. Notice that when we work with a compressed instance, by first invoking Theorem 4, we have $|V^{\mathcal{L}}| = O(\frac{\log N}{\epsilon})$, where N is the number of vertices in the original instance. Hence, a PTAS would be achieved if for all $u \in V^{\mathcal{L}}$, we had $w(T_u) = \Theta(\frac{\epsilon^2}{\log N}) \cdot \mathrm{val}(\mathsf{OPT})$. One way to achieve this in quasipolynomial time is to first guess a subset of $\Theta(\frac{\log N}{\epsilon^2})$ many vertices of an optimal solution with highest impact, i.e., among all vertices $u \in \mathsf{OPT}$ we guess those with largest $w(T_u)$. This techniques has been used in various other settings (see for example [31, 21] for further details) and leads to another QPTAS for the Firefighter problem. Again, it is unclear how this QPTAS could be turned into an efficient procedure.

The above discussion motivates to investigate vertices $u \in V \setminus \{r\}$ with $w(T_u) \geq \eta$ for some $\eta = \Theta(\frac{\epsilon^2}{\log N})\,\mathrm{val}(\mathsf{OPT})$. We call such vertices *heavy*; later, we will provide an explicit definition of η that does not depend on the unknown $\mathrm{val}(\mathsf{OPT})$ and is explicit about the hidden constant. Let $H = \{u \in V \setminus \{r\} \mid w(u) \geq \eta\}$ be the set of all heavy vertices. Observe that $G[H \cup \{r\}]$—i.e., the induced subgraph of G over the vertices $H \cup \{r\}$—is a subtree of G, which we call the *heavy tree*.

Recall that by the above discussion, if we work on a compressed instance with $L = O(\frac{\log N}{\epsilon})$ levels, and if an optimal vertex solution to $\mathrm{LP}_{\mathrm{FF}}$ has no loose vertices that are heavy, then an integral solution can be obtained being at most a factor of $1-\epsilon$ off the LP value. Hence, if we were able to guess the heavy vertices contained in an optimal solution, the integrality gap of the reduced problem would be small since no heavy vertices are left in the LP, and can thus not be loose anymore.

Whereas there are too many options to enumerate over all possible subsets of heavy vertices that an optimal solution may contain, we will do a coarser enumeration. More precisely, we will partition the heavy vertices into $O_\epsilon(\log N)$ subpaths and guess for each subpath whether it contains a vertex of OPT. For this to work out we need that the heavy tree has a very simple topology; in particular, it should only have $O_\epsilon(\log N)$ leaves. Whereas this does not hold in general, we can enforce it by a further transformation making sure that OPT saves a constant-fraction of $w(V)$ which—as we will observe next—indeed limits the number of leaves of the heavy tree to $O_\epsilon(\log N)$. Furthermore, this transformation is useful to complete our definition of heavy vertices by explicitly defining the threshold η.

Lemma 7. *Let $\mathcal{I}$ be a general-budget Firefighter instance on a tree $G = (V, E)$ with weights w. Then for any $\lambda \in \mathbb{Z}_{\geq 1}$, one can efficiently construct a new Firefighter instance $\overline{\mathcal{I}}$ on a subtree $G' = (V', E')$ of G with same budgets, by starting from $\mathcal{I}$ and applying node deletions and weight reductions, such that*

(i) $\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}})) \geq \left(1 - \frac{1}{\lambda}\right) \mathrm{val}(\mathsf{OPT}(\mathcal{I}))$, *and*

(ii) $\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}})) \geq \frac{1}{\lambda} w'(V')$, *where $w' \leq w$ are the vertex weights in instance $\overline{\mathcal{I}}$.*

The deletion of $u \in V$ corresponds to removing the whole subtree below u from G, i.e., all vertices in T_u.

Since Lemma 7 constructs a new instance using only node deletions and weight reductions, any solution to the new instance is also a solution to the original instance of at least the same objective value.

Our PTAS for the Firefighter problem first applies the compression Theorem 4 with $\delta = \epsilon/3$ and then Lemma 7 with $\lambda = \lceil \frac{3}{\epsilon} \rceil$ to obtain a general budget Firefighter instance on a tree $G = (V, E)$. We summarize the properties of this new instance $G = (V, E)$ below. As before, to avoid confusion, we denote by N the number of vertices of the original instance.

Property 8.

(i) The depth L of G satisfies $L = O(\frac{\log N}{\epsilon})$.

(ii) $\mathrm{val}(\mathsf{OPT}) \geq \lceil \frac{3}{\epsilon} \rceil^{-1} w(V) \geq \frac{1}{4} \epsilon w(V)$.

(iii) The optimal value $\mathrm{val}(\mathsf{OPT})$ of the new instance is at least a $(1 - \frac{2}{3}\epsilon)$-fraction of the optimal value of the original instance.

(iv) Any solution to the new instance can be transformed efficiently into a solution of the original instance of at least the same value.

Hence, to obtain a PTAS for the original instance, it suffices to obtain, for any $\epsilon > 0$, a $(1 - \frac{\epsilon}{3})$-approximation for an instance satisfying Property 8. In what follows, we assume to work with an instance satisfying Property 8 and show that this is possible.

Due to the lower bound on $\mathrm{val}(\mathsf{OPT})$ provided by Property 8, we now define the threshold $\eta = \Theta(\frac{\epsilon}{\log N}) \mathrm{val}(\mathsf{OPT})$ in terms of $w(V)$ by

$$\eta = \frac{1}{12} \frac{\epsilon^2}{L} w(V),$$

which implies that we can afford losing L times a weight of η, which will sum up to a total loss of at most $\frac{1}{12}\epsilon^2 w(V) \leq \frac{1}{3}\epsilon \, \mathrm{val}(\mathsf{OPT})$, where the inequality is due to Property 8.

Consider again the heavy tree $G[H \cup \{r\}]$. Due to Property 8 its topology is quite simple. More precisely, the heavy tree has only $O(\frac{\log N}{\epsilon^3})$ leaves. Indeed, each leaf $u \in H$ of the heavy tree fulfills $w(T_u) \geq \eta$, and two different leaves $u_1, u_2 \in H$ satisfy $T_{u_1} \cap T_{u_2} = \emptyset$; since the total weight of the tree is $w(V)$, the heavy tree has at most $12L/\epsilon^2 = O(\frac{\log N}{\epsilon^3})$ many leaves.

In the next step, we define a well-chosen small subset Q of heavy vertices whose removal (together with r) from G will break G into components of weight at most η. Simultaneously, we choose Q such that removing it together with r from the heavy tree breaks it into paths, over which we will do an enumeration later.

Lemma 9. *One can efficiently determine a set $Q \subseteq H$ satisfying the following.*

(i) $|Q| = O(\frac{\log N}{\epsilon^3})$.
(ii) Q *contains all leaves and all vertices of degree at least* 3 *of the heavy tree, except for the root* r.
(iii) *Removing* $Q \cup \{r\}$ *from* G *leads to a graph* $G[V \setminus (Q \cup \{r\})]$ *where each connected component has vertices whose weight sums up to at most* η.

For each vertex $q \in Q$, let $H_q \subseteq H$ be all vertices that are visited when traversing the path P_q from q to r until (but not including) the next vertex in $Q \cup \{r\}$. Hence, H_q is a subpath of the heavy tree such that $H_q \cap Q = \{q\}$, which we call for brevity a *Q-path*. Moreover the set of all Q-paths partitions H.

We use an enumeration procedure to determine on which Q-paths to protect a vertex. Since Q-paths are subpaths of leaf-root paths, we can assume that at most one vertex is protected in each Q-path. Our algorithm enumerates over all $2^{|Q|}$ possible subsets $Z \subseteq Q$, where Z represents the Q-paths on which we will protect a vertex. Incorporating this guess into $\mathrm{LP}_{\mathrm{FF}}$, we get the following linear program $\mathrm{LP}_{\mathrm{FF}}(Z)$:

$$\begin{array}{rlcll} \max & \sum_{u \in V\setminus\{r\}} x_u w(T_u) & & & \\ & x(P_u) & \leq & 1 & \forall u \in \Gamma \\ & x(V_{\leq \ell}) & \leq & \sum_{i=1}^{\ell} B_i & \forall \ell \in [L] \\ & x(H_q) & = & 1 & \forall q \in Z \\ & x(H_q) & = & 0 & \forall q \in Q \setminus Z \\ & x & \in & \mathbb{R}_{\geq 0}^{V\setminus\{r\}}. & \end{array} \qquad (\mathrm{LP}_{\mathrm{FF}}(Z))$$

We start with a simple observation regarding $\mathrm{LP}_{\mathrm{FF}}(Z)$.

Lemma 10. *The polytope over which* $\mathrm{LP}_{\mathrm{FF}}(Z)$ *optimizes is a face of the polytope describing the feasible region of* $\mathrm{LP}_{\mathrm{FF}}$. *Consequently, any vertex solution of* $\mathrm{LP}_{\mathrm{FF}}(Z)$ *is a vertex solution of* $\mathrm{LP}_{\mathrm{FF}}$.

Proof. The statement immediately follows by observing that for any $q \in Q$, the inequalities $x(H_q) \leq 1$ and $x(H_q) \geq 0$ are valid inequalities for $\mathrm{LP}_{\mathrm{FF}}$. Notice that $x(H_q) \leq 1$ is a valid inequality for $\mathrm{LP}_{\mathrm{FF}}$ because H_q is a subpath of a leaf-root path, and the load on any leaf-root path is limited to 1 in $\mathrm{LP}_{\mathrm{FF}}$. □

Analogously to $\mathrm{LP}_{\mathrm{FF}}$ we define loose and tight vertices for a solution to $\mathrm{LP}_{\mathrm{FF}}(Z)$. A crucial implication of Lemma 10 is that Lemma 6 also applies to any vertex solution of $\mathrm{LP}_{\mathrm{FF}}(Z)$.

We will show in the following that for the right set $Z \subseteq Q$, $\mathrm{LP}_{\mathrm{FF}}(Z)$ has a small integrality gap and we can easily get a nearly optimal integral solution. A key observation in the analysis of our algorithm is that we can now limit the impact of loose vertices. More precisely, any loose vertex outside of the heavy tree has LP contribution at most η by definition of the heavy tree. Furthermore, for each loose vertex u on the heavy tree, which lies on some Q-path H_q, its load $x(u)$ can be redistributed on the tight vertex on H_q. Such a redistribution will have low impact due to our choice of Q.

We are now ready to state our $(1 - \frac{\epsilon}{3})$-approximation for an instance satisfying Property 8, which, as discussed, implies a PTAS for the Firefighter problem. Algorithm 1 describes our $(1 - \frac{\epsilon}{3})$-approximation.

The following statement completes the proof of Theorem 2.

Theorem 11. *For any general-budget Firefighter instance satisfying Property 8, Algorithm 1 computes efficiently a feasible set of vertices* $U \subseteq V \setminus \{r\}$ *to protect that is a* $(1 - \frac{\epsilon}{3})$*-approximation.*

Proof. First observe that the linear program solved in step 4 will indeed lead to a characteristic vector with only $\{0, 1\}$-components. This is the case since no two x-tight vertices can lie on the same leaf-root path. Hence, as discussed previously, the linear program $\mathrm{LP}_{\mathrm{FF}}$ restricted to variables corresponding to $V^{\mathcal{T}}$ is totally unimodular; indeed, the leaf-root path constraints $x(P_u) \leq 1$ for $u \in \Gamma$ reduce to $x(v) \leq 1$ for

Algorithm 1: A $(1-\frac{\epsilon}{3})$-approximation for a general-budget Firefighter instance satisfying Property 8.

1. Determine heavy vertices $H = \{u \in V \mid w(T_u) \geq \eta\}$, where $\eta = \frac{1}{12}\frac{\epsilon^2}{L}w(V)$.
2. Compute $Q \subseteq H$ using Lemma 9.
3. For each $Z \subseteq Q$, obtain an optimal vertex solution to $\mathrm{LP}_{\mathrm{FF}}(Z)$. Let $Z^* \subseteq Q$ be a set for which the optimal value of $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$ is largest among all subsets of Q, and let x be an optimal vertex solution to $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$.
4. Let $V^{\mathcal{T}}$ be the x-tight vertices. Obtain an optimal vertex solution to $\mathrm{LP}_{\mathrm{FF}}$ restricted to variables corresponding to vertices in $V^{\mathcal{T}}$. The solution will be a $\{0,1\}$-vector, being the characteristic vector of a set $U \subseteq V^{\mathcal{T}}$ which we return.

$v \in V^{\mathcal{T}}$, and the remaining LP corresponds to a linear program over a laminar matroid, reflecting the budget constraints. Moreover, the set U is clearly budget-feasible since the budget constraints are enforced by $\mathrm{LP}_{\mathrm{FF}}$. Also, Algorithm 1 runs in polynomial time because $|Q| = O(\frac{\log N}{\epsilon^3})$ by Lemma 9 and hence, the number of subsets of Q is bounded by $N^{O(\frac{1}{\epsilon^3})}$.

It remains to show that U is a $(1-\frac{\epsilon}{3})$-approximation. Let OPT be an optimal solution to the considered Firefighter instance with value $\mathrm{val}(\mathsf{OPT})$. Observe first that the value ν^* of $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$ satisfies $\nu^* \geq \mathrm{val}(\mathsf{OPT})$, because one of the sets $Z \subseteq Q$ corresponds to OPT, namely $Z = \{q \in Q \mid H_q \cap \mathsf{OPT} \neq \emptyset\}$, and for this Z the characteristic vector $\chi^{\mathsf{OPT}} \in \{0,1\}^{V\setminus\{r\}}$ of OPT is feasible for $\mathrm{LP}_{\mathrm{FF}}(Z)$. We complete the proof of Theorem 11 by showing that the value $\mathrm{val}(U)$ of U satisfies $\mathrm{val}(U) \geq (1-\frac{\epsilon}{3})\nu^*$. For this we show how to transform an optimal solution x of $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$ into a solution y to $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$ with $\mathrm{supp}(y) \subseteq V^{\mathcal{T}}$ and such that the objective value $\mathrm{val}(y)$ of y satisfies $\mathrm{val}(y) \geq (1-\frac{\epsilon}{3})\nu^*$.

Let $V^{\mathcal{L}} \subseteq \mathrm{supp}(x)$ be the set of x-loose vertices, and let H be all heavy vertices, as usual. To obtain y, we start with $y = x$ and first set $y(u) = 0$ for each $u \in V^{\mathcal{L}} \setminus H$. Moreover, for each $u \in V^{\mathcal{L}} \cap H$ we do the following. Being part of the heavy vertices and fulfilling $x(u) > 0$, the vertex u lies on some Q-path H_{q_u} for some $q_u \in Z^*$. Because $x(H_{q_u}) = 1$, there is a tight vertex $v \in H_{q_u}$. We move the y-value from vertex u to vertex v, i.e., $y(v) = y(v) + y(u)$ and $y(u) = 0$. This finishes the construction of y. Notice that y is feasible for $\mathrm{LP}_{\mathrm{FF}}(\mathrm{Z}^*)$, because it was obtained from x by reducing values and moving values to lower levels.

To upper bound the reduction of the LP-value when transforming x into y, we show that the modification done for each loose vertex $u \in V^{\mathcal{L}}$ decreased the LP-value by at most η. Clearly, for each $u \in V^{\mathcal{L}} \setminus H$, since u is not heavy we have $w(T_u) \leq \eta$; thus setting $y(u) = 0$ will have an impact of at most η on the LP value. Similarly, for $u \in V^{\mathcal{L}} \cap H$, moving the y-value of u to q_u decreases the LP objective value by

$$y(u) \cdot (w(T_u) - w(T_{q_u})) \leq w(T_u) - w(T_{q_u}) = w(T_u \setminus T_{q_u}) \leq \eta,$$

where the last inequality follows by observing that $T_u \setminus T_{q_u}$ are vertices in the same connected component of $G[V \setminus (Q \cup \{r\})]$, and thus have a total weight of at most η by Lemma 9.

Hence, $\mathrm{val}(x) - \mathrm{val}(y) \leq |V^{\mathcal{L}}| \cdot \eta \leq L \cdot \eta$, where the second inequality follows by Property 8. This completes the proof by observing that $|V^{\mathcal{L}}| \leq L$ by Lemma 6, and thus

$$\begin{aligned}\mathrm{val}(y) &= \mathrm{val}(x) + (\mathrm{val}(y) - \mathrm{val}(x)) \geq \mathrm{val}(\mathsf{OPT}) + \mathrm{val}(y) - \mathrm{val}(x) \geq \mathrm{val}(\mathsf{OPT}) - L \cdot \eta \\ &= \mathrm{val}(\mathsf{OPT}) - \frac{1}{12}\epsilon^2 w(V) \geq \left(1 - \frac{1}{3}\epsilon\right) \mathrm{val}(\mathsf{OPT}),\end{aligned}$$

where the last inequality is due to Property 8.

□

3.3 Overview of $O(1)$-approximation for RMFC

Also our $O(1)$-approximation for RMFC uses the natural LP, i.e, $\mathrm{LP}_{\mathrm{RMFC}}$, as a crucial tool to guide the algorithm. Throughout this section we will work on a compressed instance $G=(V,E)$ of RMFC, obtained through Theorem 5. Hence, the number of levels is $L=O(\log N)$, where N is the number of vertices of the original instance. Furthermore, the budget on level $\ell \in [L]$ is given by $B_\ell = 2^\ell B$. The advantage of working with a compressed instance for RMFC is twofold. First, we will again apply sparsity reasonings to limit in certain settings the number of loose (badly structured) vertices by the number of levels of the instance. Second, the fact that low levels—i.e., levels far away from the root—have high budget, will allow us to protect a large number of loose vertices by only increasing B by a constant.

For simplicity, we work with a slight variation of $\mathrm{LP}_{\mathrm{RMFC}}$, where we replace, for $\ell \in [L]$, the budget constraints $x(V_{\leq \ell}) \leq \sum_{i=1}^{\ell} B_i$ by $x(V_\ell) \leq B_\ell$. For brevity, we define

$$P_B = \left\{ x \in \mathbb{R}_{\geq 0}^{V \setminus \{r\}} \;\middle|\; x(V_\ell) \leq B \cdot 2^\ell \;\; \forall \ell \in [L] \right\}.$$

As previously mentioned (and shown in [8]), the resulting LP is equivalent to $\mathrm{LP}_{\mathrm{RMFC}}$. Furthermore, since the budget B for a feasible RMFC solution has to be chosen integral, we require $B \geq 1$. Hence, the resulting linear relaxation asks to find the minimum $B \geq 1$ such that the following polytope is non-empty:

$$\bar{P}_B = P_B \cap \left\{ x \in \mathbb{R}_{\geq 0}^{V \setminus \{r\}} \;\middle|\; x(P_u) \geq 1 \;\; \forall u \in \Gamma \right\}.$$

We start by discussing approaches to partially round a fractional point $x \in \bar{P}_B$, for some fixed budget $B \geq 1$. First, we can assume that $x(P_u) = 1$ for $u \in \Gamma$. Indeed, whenever $x(P_u) > 1$, then one can reduce the x-values toward the bottom of the path P_u to obtain $x(P_u) = 1$ and maintaining $x \in \bar{P}_B$. Hence, any leaf $u \in \Gamma$ is fractionally cut off from the root through the x-values on P_u. A crucial property we derive and exploit is that leaves that are cut off from r on mostly low levels, i.e., most of the x-value on P_u comes from vertices far away from the root, can be cut off from the root via a set of vertices to be protected that are budget-feasible when increasing B only by a constant.

To exemplify the above statement, consider the level $h = \lfloor \log L \rfloor$ as a threshold to define top levels V_ℓ as those with indices $\ell \leq h$ and bottom levels when $\ell > h$. For any leaf $u \in \Gamma$, we partition the path P_u into its top part $P_u \cap V_{\leq h}$ and its bottom part $P_u \cap V_{>h}$. Consider all leaves that are mostly cut off in bottom levels: $W = \{u \in \Gamma \mid x(P_u \cap V_{>h}) \geq 0.5\}$. We will show that there is a subset of vertices $R \subseteq V_{>h}$ on bottom levels to be protected that is feasible for budget $\bar{B} = 2B + 1 \leq 3B$ and cuts off all leaves in W from the root. We provide a brief sketch why this result holds, and present a formal proof later. If we set all entries of x on top levels $V_{\leq h}$ to zero, we get a vector y with $\mathrm{supp}(y) \subseteq V_{>h}$ such that $y(P_u) \geq 0.5$ for $u \in W$. Hence, $2y$ fractionally cuts off all vertices in W from the root and is feasible for budget $2B$. To increase sparsity, we can replace $2y$ by a vertex $\bar{z}$ of the polytope

$$Q = \left\{ z \in \mathbb{R}_{\geq 0}^{V \setminus \{r\}} \;\middle|\; z(V_\ell) \leq 2B \cdot 2^\ell \;\; \forall \ell \in [L], z(V_{<h}) = 0, z(P_u) \geq 1 \;\; \forall u \in W \right\},$$

which describes possible ways to cut off W from r only using levels $V_{\geq h}$, and Q is non-empty since $2y \in Q$. Exhibiting a sparsity reasoning analogous to the one used for the Firefighter problem, we can show that z has no more than L many z-loose vertices. Thus, we can first include all z-loose vertices in the set R of vertices to be protected by increasing the budget of each level $\ell > h$ by at most $L \leq 2^{h+1} \leq 2^\ell$. The remaining vertices in $\mathrm{supp}(z)$ are well structures (no two of them lie on the same leaf-root path), and an integral solution can be obtained easily. The new budget value is $\bar{B} = 2B + 1$, where the "$+1$" term pays for the loose vertices.

The following theorem formalizes the above reasoning and generalizes it in two ways. First, for a leaf $u \in \Gamma$ to be part of W, we required it to have a total x-value of at least 0.5 within the bottom levels; we will allow for replacing 0.5 by an arbitrary threshold $\mu \in (0, 1]$. Second, the level h defining what is top and bottom can be chosen to be of the form $h = \lfloor \log^{(q)} L \rfloor$ for $q \in \mathbb{Z}_{\geq 0}$, where $\log^{(q)} L := \log\log \ldots \log L$ is the value obtained by taking q many logs of L, and by convention we set $\log^{(0)} L := L$. The generalization in terms of h can be thought of as iterating the above procedure on the RMFC instance restricted to $V_{\leq h}$.

Theorem 12. *Let $B \in \mathbb{R}_{\geq 0}$, $\mu \in (0, 1]$, $q \in \mathbb{Z}_{\geq 0}$, and $h = \lfloor \log^{(q)} L \rfloor$. Let $x \in P_B$ with $\operatorname{supp}(x) \subseteq V_{>h}$, and we define $W = \{u \in \Gamma \mid x(P_u) \geq \mu\}$. Then one can efficiently compute a set $R \subseteq V_{>h}$ such that*

(i) $R \cap P_u \neq \emptyset \quad \forall u \in W$, *and*

(ii) $\chi^R \in P_{B'}$, *where* $B' = \frac{q}{\mu} B + 1$ *and* $\chi^R \in \{0, 1\}^{V \setminus \{r\}}$ *is the characteristic vector of* R.

Theorem 12 has several interesting consequences. It immediately implies an LP-based $O(\log^* N)$-approximation for RMFC, thus matching the currently best approximation result by Chalermsook and Chuzhoy [8]: It suffices to start with an optimal LP solution $B \geq 1$ and $x \in \bar{P}_B$ and invoke the above theorem with $\mu = 1$, $q = 1 + \log^* L$. Notice that by definition of $\log^*$ we have $\log^* L = \min\{\alpha \in \mathbb{Z}_{\geq 0} \mid \log^{\alpha} L \leq 1\}$; hence $h = \lfloor \log^{1+\log^* L} L \rfloor = 0$, implying that all levels are bottom levels. Since the integrality gap of the LP is $\Omega(\log^* N) = \Omega(\log^* L)$, Theorem 12 captures the limits of what can be achieved by techniques based on the standard LP.

Interestingly, Theorem 12 also implies that the $\Omega(\log^* L)$ integrality gap is only due to the top levels of the instance. More precisely, if, for any $q = O(1)$ and $h = \lfloor \log^{(q)} L \rfloor$, one would know what vertices an optimal solution R^* protects within the levels $V_{\leq h}$, then a constant-factor approximation for RMFC follows easily by solving an LP on the bottom levels $V_{>h}$ and using Theorem 12 with $\mu = 1$ to round the obtained solution.

Also, using Theorem 12 it is not hard to find constant-factor approximation algorithms for RMFC if the optimal budget B_{OPT} is large enough, say $B \geq \log L$.[7] The main idea is to solve the LP and define $h = \lfloor \log L \rfloor$. Leaves that are primarily cut off by x on bottom levels can be handled using Theorem 12. For the remaining leaves, which are cut-off mostly on top levels, we can resolve an LP only on the top levels $V_{\leq h}$ to cut them off. This LP solution is sparse and contains at most $h \leq B$ loose nodes. Hence, all loose vertices can be selected by increasing the budget by at most $h \leq B$, leading to a well-structured residual problem for which one can easily find an integral solution. The following theorem summarizes this discussion. A formal proof for Theorem 13 can be found in Section 6.

Theorem 13. *There is an efficient algorithm that computes a feasible solution to a (compressed) instance of RMFC with budget $B \leq 3 \cdot \max\{\log L, B_{\mathsf{OPT}}\}$.*

In what follows, we therefore assume $B_{\mathsf{OPT}} < \log L$ and present an efficient way to partially enumerate vertices to be protected on top levels, leading to the claimed $O(1)$-approximation.

Partial enumeration algorithm

Throughout our algorithm, we set

$$h = \lfloor \log^{(2)} L \rfloor$$

to be the threshold level defining top vertices $V_{\leq h}$ and bottom vertices $V_{>h}$. Within our enumeration procedure we will solve LPs where we explicitly include some vertex set $A \subseteq V_{\leq h}$ to be part of the protected

[7] Actually, the argument we present in the following works for any $B = \log^{(\Omega(1))} L$. However, we later only need it for $B \geq \log L$ and thus focus on this case.

vertices, and also exclude some set $D \subseteq V_{\leq h}$ from being protected. Our enumeration works by modifying the sets A and D throughout the algorithm. We thus define the following LP for two disjoint sets $A, D \subseteq V_{\leq h}$:

$$\begin{array}{rrcll} \min & B & & & \\ & x & \in & \bar{P}_B & \\ & B & \geq & 1 & \\ & x(u) & = & 1 & \forall u \in A \\ & x(u) & = & 0 & \forall u \in D\ . \end{array} \tag{LP(A, D)}$$

Notice that LP(A, D) is indeed an LP even though the definition of $\bar{P}_B$ depends on B (but it does so linearly). Before formally stating our enumeration procedure, we briefly discuss the main idea behind it. Let $\mathsf{OPT} \subseteq V \setminus \{r\}$ be an optimal solution to our (compressed) RMFC instance corresponding to some budget $B_{\mathsf{OPT}} \in \mathbb{Z}_{\geq 1}$. We assume without loss of generality that OPT does not contain redundancies, i.e., there is precisely one vertex of OPT on each leaf-root path. Assume that we already guessed some vertex set $A \subseteq V_{\leq h}$ to be protected and a vertex set $D \subseteq V_{\leq h}$ not to be protected, and that these guesses are compatible with OPT, i.e., $A \subseteq \mathsf{OPT}$ and $D \cap \mathsf{OPT} = \emptyset$.

Let (B, x) be an optimal solution to LP(A, D). Because we assume that the sets A and D do not conflict with OPT, we have $B \leq B_{\mathsf{OPT}}$ because $(B_{\mathsf{OPT}}, \chi^{\mathsf{OPT}})$ is feasible for LP(A, D). We define

$$W_x = \left\{ u \in \Gamma \;\middle|\; x(P_u \cap V_{>h}) \geq \frac{2}{3} \right\}$$

to be the set of leaves primarily, i.e., with x-load at least $\mu = \frac{2}{3}$, cut off from the root within bottom levels. For each $u \in \Gamma \setminus W_x$, let $f_u \in V_{\leq h}$ be the vertex closest to the root among all vertices in $P_u \cap V_{\leq h} \cap \mathrm{supp}(x)$, and we define

$$F_x = \{f_u \mid u \in \Gamma \setminus W_x\} \setminus A. \tag{1}$$

Notice that by definition, no two vertices of F_x lie on the same leaf-root path. Furthermore, every leaf $u \in W_x$ is part of the subtree T_f for precisely one $f \in F_x$. The main motivation for considering F_x is that to guess vertices in top levels, we only need to focus on vertices lying below some vertex in F_x, i.e., vertices in the set $Q_x = V_{\leq h} \cap (\cup_{f \in F_x} T_f)$.

To exemplify this, we first consider the special case $\mathsf{OPT} \cap Q_x = \emptyset$, which will also play a central role later in the analysis of our algorithm. We show that for this case we can get an $O(1)$-approximation to RMFC, even though we may only have guessed a proper subset $A \subsetneq \mathsf{OPT} \cap V_{\leq h}$ of the OPT-vertices within the top levels.

Lemma 14. *Let $A \subseteq \mathsf{OPT} \cap V_{\leq h}$, $D \subseteq V_{\leq h} \setminus \mathsf{OPT}$ be two disjoint sets, and x be an optimal solution to LP(A, D), and assume that $\mathsf{OPT} \cap Q_x = \emptyset$. Moreover, let $(y, \bar{B})$ be an optimal solution to LP(A, $V_{\leq h} \setminus$ A). Then $\bar{B} \leq \frac{5}{2} B_{\mathsf{OPT}}$.*

Furthermore, by applying Theorem 12 to $y \wedge \chi^{V_{>h}}$ with $\mu = 1$, a set of vertices $R \subseteq V_{>h}$ is obtained such that $R \cup A$ is a feasible solution to RMFC with respect to the budget $6 \cdot B_{\mathsf{OPT}}$.[8]

Proof. Notice that $\mathsf{OPT} \cap Q_x = \emptyset$ implies that for each $u \in \Gamma \setminus W_x$, the set OPT contains a vertex on $P_u \cap V_{>h}$. Hence, $z = \frac{3}{2}(x \wedge \chi^{V_{>h}}) + (\chi^{\mathsf{OPT}} \wedge \chi^{V_{>h}})$ satisfies $z(P_u) \geq 1$ for $u \in \Gamma$ and $z \in P_{\frac{3}{2}B + B_{\mathsf{OPT}}}$ because $x \wedge \chi^{V_{>h}} \in P_B$ and $\chi^{\mathsf{OPT}} \in P_{B_{\mathsf{OPT}}}$. This implies that $(z, \frac{3}{2}B + B_{\mathsf{OPT}})$ is feasible for LP(A, $V_{\leq h} \setminus$ A), and thus $\bar{B} \leq \frac{3}{2}B + B_{\mathsf{OPT}} \leq \frac{5}{2}B_{\mathsf{OPT}}$, as claimed.

The second part of the lemma follows in a straightforward way from Theorem 12. Observe first that for each leaf $u \in \Gamma$, the solution y either cuts off u from the root only using top levels or only using bottom

[8] For two vectors $a, b \in \mathbb{R}^n$ we denote by $a \wedge b \in \mathbb{R}^n$ the component-wise minimum of a and b.

levels because y is a $\{0,1\}$-solution on the top levels $V_{\leq h}$, since on top levels it was fixed to χ^A. Hence, Theorem 12 can indeed be applied to y with $\mu = 1$ leading to a set $R \subseteq V_{>h}$ that is feasible with respect to the budget $5B_{\mathsf{OPT}} + 1 \leq 6B_{\mathsf{OPT}}$. Furthermore, A is feasible for budget B_{OPT} because it is a subset of OPT. Since $A \subseteq V_{\leq h}$ and $R \subseteq V_{>h}$ are on disjoint levels, the set $R \cup A$ is feasible for the budget $6B_{\mathsf{OPT}}$. ☐

Our final algorithm is based on a recursive enumeration procedure that computes a polynomial collection of pairs of disjoint sets (A, D) with $A \subseteq V_{\leq h}$ and $D \subseteq V_{\leq h}$, such that there is one pair (A, D) in the collection with a corresponding LP solution x of $\mathrm{LP(A, D)}$ satisfying that the triple (A, D, x) fulfills the conditions of Lemma 14, and thus leading to a constant-factor approximation. Our enumeration algorithm $\mathrm{Enum}(A, D, \gamma)$ is described below. It contains a parameter $\gamma \in \mathbb{Z}_{\geq 0}$ that bounds the recursion depth of the enumerations.

$\mathrm{Enum}(A, D, \gamma)$: Enumerating triples (A, D, x) to find one satisfying the conditions of Lemma 14.

1. Compute optimal solution (x, B) to $\mathrm{LP(A, D)}$.
2. **If** $B > \log L$**: stop**. Otherwise, continue with step 3.
3. Add (A, D, x) to the family of triples to be considered.
4. **If** $\gamma \neq 0$ **:** //recursion depth not yet reached
   - **For** $u \in F_x$**:** //F_x is defined as in (1)
     - Recursive call to $\mathrm{Enum}(A \cup \{u\}, D, \gamma - 1)$.
     - Recursive call to $\mathrm{Enum}(A, D \cup P_u, \gamma - 1)$.

We can show that only a small recursion depth γ is needed for the enumeration algorithm to explore a good triple (A, D, x), which satisfies the conditions of Lemma 14.

Lemma 15. *Let $\bar{\gamma} = 2(\log L)^2 \log^{(2)} L$. The enumeration procedure $\mathrm{Enum}(\emptyset, \emptyset, \bar{\gamma})$ runs in polynomial time. Furthermore, if $B_{\mathsf{OPT}} \leq \log L$, then $\mathrm{Enum}(\emptyset, \emptyset, \bar{\gamma})$ will encounter a triple (A, D, x) satisfying the conditions of Lemma 14, i.e.,*

(i) $A \subseteq \mathsf{OPT} \cap V_{\leq h}$,
(ii) $D \subseteq V_{\leq h} \setminus \mathsf{OPT}$, *and*
(iii) $\mathsf{OPT} \cap Q_x = \emptyset$.

Hence, combining Lemma 15 and Lemma 14 completes our enumeration procedure and implies the following result.

Corollary 16. *Let $\mathcal{I}$ be an RMFC instance on L levels on a graph $G = (V, E)$ with budgets $B_\ell = 2^\ell \cdot B$. Then there is a procedure with running time polynomial in 2^L, returning a solution (Q, B) for $\mathcal{I}$, where $Q \subseteq V \setminus \{r\}$ is a set of vertices to protect that is feasible for budget B, satisfying the following: If the optimal budget B_{OPT} for $\mathcal{I}$ satisfies $B_{\mathsf{OPT}} \leq \log L$ then $B \leq 6B_{\mathsf{OPT}}$.*

Proof. It suffices to run $\mathrm{Enum}(\emptyset, \emptyset, \bar{\gamma})$ to first efficiently obtain a family of triples $(A_i, D_i, x_i)_i$, where A_i, D_i are disjoint subsets of $V_{\leq h}$, and x is an optimal solution to $\mathrm{LP(A_i, D_i)}$. By Lemma 15, one of these triples satisfies the conditions of Lemma 14. (Notice that these conditions cannot be checked since it would require knowledge of OPT.) For each triple (A_i, D_i, x_i) we obtain a corresponding solution for $\mathcal{I}$ following the construction described in Lemma 14. More precisely, we first compute an optimal solution $(y_i, \bar{B}_i)$ to $\mathrm{LP(A_i, V_{\leq h} \setminus A_i)}$. Then, by applying Theorem 12 to $y_i \wedge \chi^{V_{>h}}$ with $\mu = 1$, a set of vertices $R_i \subseteq V_{>h}$ is obtained such that $R_i \cup A_i$ is feasible for $\mathcal{I}$ for some budget B_i. Among all such sets $R_i \cup A_i$, we return the one with minimum B_i. Because Lemma 15 guarantees that one of the triples (A_i, D_i, x_i) satisfies the conditions of Lemma 14, we have by Lemma 14 that the best protection set $Q = R_j \cup A_j$ among all $R_i \cup A_i$ has a budget B_j satisfying $B_j \leq 6B_{\mathsf{OPT}}$. ◻

Summary of our $O(1)$-approximation for RMFC

Starting with an RMFC instance $\mathcal{I}^{\text{orig}}$ on a tree with N vertices, we first apply our compression result, Theorem 5, to obtain an RMFC instance $\mathcal{I}$ on a graph $G = (V, E)$ with depth $L = O(\log N)$, and non-uniform budgets $B_\ell = 2^\ell B$ for $\ell \in [L]$. Let $B_{\mathsf{OPT}} \in \mathbb{Z}_{\geq 1}$ be the optimal budget—i.e., value of B—for the instance $\mathcal{I}$, and let $B^{\text{orig}}_{\mathsf{OPT}}$ be the optimal budget for $\mathcal{I}^{\text{orig}}$. By Theorem 5, we have $B_{\mathsf{OPT}} \leq B^{\text{orig}}_{\mathsf{OPT}}$, and any solution to $\mathcal{I}$ using budget B can efficiently be transformed into one of $\mathcal{I}^{\text{orig}}_{\mathsf{OPT}}$ of budget $2B$.

We now invoke Theorem 13 and Corollary 16. Both guarantee that a solution to $\mathcal{I}$ with certain properties can be computed efficiently. Among the two solutions derived from Theorem 13 and Corollary 16, we consider the one (Q, B) with lower budget B, where $Q \subseteq V \setminus \{r\}$ is a set of vertices to protect, feasible for budget B. If $B \leq \log L$, then Theorem 13 implies $B \leq 3B_{\mathsf{OPT}}$, otherwise Corollary 16 implies $B \leq 6B_{\mathsf{OPT}}$. Hence, in any case we have a 6-approximation for $\mathcal{I}$. As mentioned before, Theorem 5 implies that the solution Q can efficiently be transformed into a solution for the original instance $\mathcal{I}^{\text{orig}}$ that is feasible with respect to the budget $2B \leq 12B_{\mathsf{OPT}} \leq 12B^{\text{orig}}_{\mathsf{OPT}}$, thus implying Theorem 1.

4 Details on compression results

In this section, we present the proofs for our compression results, Theorem 4 and Theorem 5. We start by proving Theorem 4. The same ideas are used with a slight adaptation in the proof of Theorem 5.

We call an instance $\overline{\mathcal{I}}$ obtained from an instance $\mathcal{I}$ by a sequence of down-push operations a *push-down of* $\mathcal{I}$. We prove Theorem 4 by proving the following result, of which Theorem 4 is an immediate consequence, as we will soon show. Informally, the following theorem states that one can efficiently construct a push-down $\overline{\mathcal{I}}$ with almost no loss in the objective and with only $O(\frac{\log L}{\delta})$ levels with non-zero budgets.

Theorem 17. *Let $\mathcal{I}$ be a unit-budget Firefighter instance with depth L, and let $\delta \in (0,1)$. Then one can efficiently construct a push-down $\overline{\mathcal{I}}$ of $\mathcal{I}$ such that*

(i) $\operatorname{val}(\mathsf{OPT}(\overline{\mathcal{I}})) \geq (1-\delta)\operatorname{val}(\mathsf{OPT}(\mathcal{I}))$, *and*
(ii) $\overline{\mathcal{I}}$ *has nonzero budget on only* $O(\frac{\log L}{\delta})$ *levels.*

Before we prove Theorem 17 let us explain how it implies Theorem 4. Concretely, we will show how levels of zero budget can be removed through the following *contraction operation*. Let $\ell \in [L]$ be a level whose budget is zero. For each vertex $u \in V_{\ell-1}$ we contract all edges from u to its children and increase the weight $w(u)$ of u by the sum of the weights of all of its children. Formally, if u has children $v_1, \ldots, v_k \in V_\ell$, the vertices $u, v_1, \ldots, u_k$ are replaced by a single vertex z with weight $w(z) = w(u) + \sum_{i=1}^k w(v_i)$, and z is adjacent to the parent of u and to all children of $v_1, \ldots, v_k$. One can easily observe that this is an "exact" transformation in the sense that any solution before the contraction remains one after contraction and vice versa (when identifying the vertex z in the contracted version with v); moreover, solutions before and after contraction have the same value.

Now, by first applying Theorem 17 and then repeating the latter contraction operations for all levels with zero budget, we obtain an equivalent instance with the desired depth, thus satisfying the conditions of Theorem 4. It remains to prove Theorem 17.

Proof of Theorem 17. Consider a unit-budget Firefighter instance on a tree $G = (V, E)$ with depth L. The push-down $\overline{\mathcal{I}}$ that we construct will have nonzero budgets precisely on the following levels $\mathcal{L} \subseteq [L]$:

$$\mathcal{L} = \left\{ \lceil (1+\delta)^j \rceil \;\middle|\; j \in \left\{0, \ldots, \left\lfloor \frac{\log L}{\log(1+\delta)} \right\rfloor \right\} \right\} \cup \{L\}.$$

For simplicity, let $\mathcal{L} = \{\ell_1, \dots, \ell_k\}$ with $\ell_1 < \ell_2 < \cdots < \ell_k$. Hence, $k = O(\frac{\log L}{\log(1+\delta)}) = O(\frac{\log L}{\delta})$. The push-down $\overline{\mathcal{I}}$ is obtained by pushing any budget on a level not in $\mathcal{L}$ down to the next level in $\mathcal{L}$. Formally, for $i \in [k]$, the budget B_{ℓ_i} at level ℓ_i is given by $B_{\ell_i} = \ell_i - \ell_{i-1}$, where we set $\ell_0 = 0$. Moreover, $B_\ell = 0$ for $\ell \in [L] \setminus \mathcal{L}$. Clearly, the instance $\overline{\mathcal{I}}$ can be constructed efficiently. Furthermore, the number of levels with nonzero budget is equal to $k = O(\frac{\log L}{\delta})$ as desired. It remains to show point (i) of Theorem 17.

To show (i), consider an optimal redundancy-free solution $S^* \subseteq V$ of $\mathcal{I}$; hence, $\mathrm{val}(\mathsf{OPT}(\mathcal{I})) = \sum_{u \in S^*} w(T_u)$ and no two vertices of S^* lie on the same leaf-root path. We will show that there is a feasible solution $\overline{S}$ to $\overline{\mathcal{I}}$ such that $\overline{S} \subseteq S^*$ and the value of $\overline{S}$ is at least $(1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I}))$. Notice that since S^* is redundancy-free, any subset of S^* is also redundancy-free. Hence, the value of the set $\overline{S}$ to construct will be equal to $\sum_{u \in \overline{S}} w(T_u)$. The set S^* being (budget-)feasible for $\mathcal{I}$ implies

$$|S^* \cap V_{\leq \ell}| \leq \ell \quad \forall \ell \in [L]. \tag{2}$$

Analogously, a set $S \subseteq V$ is feasible for $\overline{\mathcal{I}}$ if and only if

$$|S \cap V_{\leq \ell}| \leq \sum_{i=1}^{\ell} B_i \quad \forall \ell \in [L]. \tag{3}$$

Hence, we want to show that there is a set $\overline{S}$ satisfying the above system and such that $\sum_{u \in \overline{S}} w(T_u) \geq (1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I}))$. Notice that in (3), the constraint for any $\ell \in [L-1]$ such that $B_{l+1} = 0$ is redundant due to the constraint for level $\ell + 1$ which has the same right-hand side but a larger left-hand side. Thus, system (3) is equivalent to the following system

$$\begin{aligned} |S \cap V_{\leq \ell_{i+1}-1}| &\leq \ell_i \quad \forall i \in [k-1], \\ |S \cap V| &\leq L. \end{aligned} \tag{4}$$

To show that there is a good subset $\overline{S} \subseteq S^*$ that satisfies (4) we use a polyhedral approach. Observe that (3) is the constraint system of a laminar matroid (see [32, Volume B] for more information on matroids). Hence, the convex hull of all characteristic vectors $\chi^S \in \{0,1\}^V$ of sets $S \subseteq S^*$ satisfying (4) is given by the following polytope

$$P = \left\{ x \in [0,1]^V \;\middle|\; \begin{array}{c} x(V_{\leq \ell_{i+1}-1}) \leq \ell_i \;\; \forall i \in [k-1], \\ x(V) \leq L, \\ x(V \setminus S^*) = 0 \end{array} \right\}.$$

Alternatively, to see that P indeed describes the correct polytope, without relying on matroids, one can observe that its constraint matrix is totally unimodular because it has the consecutive-ones property with respect to the columns.

Thus there exists a set $\overline{S} \subseteq S^*$ with $\sum_{u \in \overline{S}} w(T_u) \geq (1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I}))$ if and only if

$$\max \left\{ \sum_{u \in S^*} x(u) \cdot w(T_u) \;\middle|\; x \in P \right\} \geq (1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I})). \tag{5}$$

To show (5), and thus complete the proof, we show that $y = \frac{1}{1+\delta}\chi^{S^*} \in P$. This will indeed imply (5) since the objective value of y satisfies

$$\sum_{u \in S^*} y(u) \cdot w(T_u) = \frac{1}{1+\delta}\,\mathrm{val}(\mathsf{OPT}(\mathcal{I})) \geq (1-\delta)\,\mathrm{val}(\mathsf{OPT}(\mathcal{I})).$$

To see that $y \in P$, notice that $y(V \setminus S^*) = 0$ and $y(V) = \frac{1}{1+\delta}|S^*| \leq \frac{1}{1+\delta}L \leq L$, where the first inequality follows by S^* satisfying (2) for $\ell = L$. Finally, for $i \in [k-1]$, we have

$$y(V_{\leq \ell_{i+1}-1}) = \frac{1}{1+\delta}|S^* \cap V_{\leq \ell_{i+1}-1}| \leq \frac{1}{1+\delta}(\ell_{i+1}-1),$$

where the first inequality follows from S^* satisfying (2) for $\ell = \ell_{i+1}-1$. It remains to show $\ell_{i+1}-1 \leq (1+\delta)\ell_i$ to prove $y \in P$. This clearly holds if $\ell_{i+1} = \ell_i + 1$. Thus assume $\ell_{i+1} \geq \ell_i + 2$. By our definition of the levels in $\mathcal{L}$, we have $\ell_{i+1} = \lceil (1+\delta)^\alpha \rceil$ for some $\alpha \in \mathbb{Z}_{\geq 0}$. Hence $\ell_i = \lceil (1+\delta)^{\alpha-1} \rceil$, because for otherwise (if $\lceil (1+\delta)^{\alpha-1} \rceil = \lceil (1+\delta)^{\alpha} \rceil$) we would have $\ell_i = \ell_{i+1}-1$, which contradicts $\ell_{i+1} \geq \ell_i + 2$. We thus obtain

$$\ell_{i+1} - 1 \leq (1+\delta)^\alpha \leq (1+\delta)\left\lceil (1+\delta)^{\alpha-1} \right\rceil = (1+\delta)\ell_i,$$

as desired.

□

We conclude with the proof of Theorem 5.

Proof of Theorem 5. We start by describing the construction of $G' = (V', E')$. As is the case in the proof of Theorem 4, we first change the budget assignment of the instance and then contract all levels with zero budgets. Notice that, for a given budget B per layer, we can consider an RMFC instance as a Firefighter instance, where each leaf $u \in \Gamma$ has weight $w(u) = 1$, and all other weights are zero. Since our goal is to save all leaves, we want to save vertices of total weight $|\Gamma|$.

For simplicity of presentation we assume that L is a power of 2. This assumption does not compromise generality, as one can always augment the original tree with one path starting from the root and going down to level $2^{\lceil \log L \rceil}$.

The set of levels in which the transformed instance will have nonzero budget is

$$\mathcal{L} = \left\{ 2^j - 1 \,\middle|\, j \in \{1, \dots, \log L\} \right\}.$$

However, instead of down-pushes we will do *up-pushes* were budget is moved upwards. More precisely, the budget of any level $\ell \in [L] \setminus \mathcal{L}$ will be assigned to the first level in $\mathcal{L}$ that is above ℓ, i.e., has a smaller index than ℓ. As for the Firefighter case, we now remove all 0-budget levels using contraction, which will lead to a new weight function w' on the vertices. Since our goal is to save the weight of the whole tree, we can remove for each vertex u with $w'(u) > 0$, the subtree below u. This does not change the problem since we have to save u, and thus will anyway also save its subtree. This finishes our construction of $G' = (V', E')$, and the task is again to remove all leaves of G'. Notice that G' has $L' \leq |\mathcal{L}| = \log L$ many levels, and level $\ell \in [L']$ has a budget of $B2^\ell$ as desired. Analogous to the discussion for compression in the context of the Firefighter problem we have that if the original problem is feasible, then so is the RMFC problem on G' with budgets $B2^\ell$. Indeed, before performing the contraction operations (which do not change the problem), the original RMFC problem was a push-down of the one we constructed.

Similarly, one can observe that before contraction, the instance we obtained is itself a push-down of the original instance with budgets $2B$ on each level. Hence, analogously to the compression result for the Firefighter case, any solution to the RMFC problem on G' can efficiently be transformed into a solution to the original RMFC problem on G with budgets $2B$ on each level.

□

5 Missing details for Firefighter PTAS

In this section we present the missing proofs for our PTAS for the Firefighter problem.

We start by proving Lemma 6, showing that any vertex solution x to LP_{FF} has few x-loose vertices. More precisely, the proof below shows that the number of x-loose vertices is upper bounded by the number of tight budget constraints. The precise same reasoning used in the proof of Lemma 6 can also be applied in further contexts, in particular for the RMFC problem.

Proof of Lemma 6

Let x be a vertex of the polytope defining the feasible set of LP_{FF}. Hence, x is uniquely defined by $|V \setminus \{r\}|$-many linearly independent and tight constraints of this polytope. Notice that the tight constraints can be partitioned into three groups:

(i) Tight nonnegativity constraints, one for each vertex in $\mathcal{F}_1 = \{u \in V \setminus \{r\} \mid x(u) = 0\}$.

(ii) Tight budget constraints, one for each level in $\mathcal{F}_2 = \{\ell \in [L] \mid x(V_{\leq \ell}) = \sum_{i=1}^{\ell} B_i\}$.

(iii) Tight leaf constraints, one for each vertex in $\mathcal{F}_3 = \{u \in \Gamma \mid x(P_u) = 1\}$.

Due to potential degeneracies of the polytope describing the feasible set of LP_{FF} there may be several options to describe x as the unique solution to a full-rank linear subsystem of the constraints described by $\mathcal{F}_1 \cup \mathcal{F}_2 \cup \mathcal{F}_3$. We consider a system that contains all tight nonnegativity constraints, i.e., constraints corresponding to $\mathcal{F}_1$, and complement these constraints with arbitrary subsets $\mathcal{F}_2' \subseteq \mathcal{F}_2$ and $\mathcal{F}_3' \subseteq \mathcal{F}_3$ of budget and leaf constraints that lead to a full rank linear system corresponding to the constraints $\mathcal{F}_1 \cup \mathcal{F}_2' \cup \mathcal{F}_3'$. Hence

$$|\mathcal{F}_1| + |\mathcal{F}_2'| + |\mathcal{F}_3'| = |V| - 1. \tag{6}$$

Let $V^{\mathcal{L}} \subseteq \text{supp}(x)$ and $V^{\mathcal{T}} \subseteq \text{supp}(x)$ be the x-loose and x-tight vertices, respectively. We first show $|\mathcal{F}_3'| \leq |V^{\mathcal{T}}|$. For each leaf $u \in \mathcal{F}_3'$, let $f_u \in V^{\mathcal{T}}$ be the first vertex on the unique u-root path that is part of $\text{supp}(x)$. In particular, if $u \in \text{supp}(x)$ then $f_u = u$. Clearly, f_u must be an x-tight vertex because the path constraint with respect to u is tight. Notice that for any distinct vertices $u_1, u_2 \in \mathcal{F}_3'$, we must have $f_{u_1} \neq f_{u_2}$. Assume by sake of contradiction that $f_{u_1} = f_{u_2}$. However, this implies $\chi^{P_{u_1}} - \chi^{P_{u_2}} \in \text{span}(\{\chi^v \mid v \in \mathcal{F}_1\})$, since $P_{u_1} \Delta P_{u_2} := (P_{u_1} \setminus P_{u_2}) \cup (P_{u_2} \setminus P_{u_1}) \subseteq \mathcal{F}_1$, and leads to a contradiction because we exhibited a linear dependence among the constraints corresponding to $\mathcal{F}_3'$ and $\mathcal{F}_1$. Hence, $f_{u_1} \neq f_{u_2}$ which implies that the map $u \mapsto f_u$ from $\mathcal{F}_3'$ to $V^{\mathcal{T}}$ is injective and thus

$$|\mathcal{F}_3'| \leq |V^{\mathcal{T}}|. \tag{7}$$

We thus obtain

$$\begin{aligned}
|\text{supp}(x)| &= |V| - 1 - |\mathcal{F}_1| && (\text{supp}(x) \text{ consists of all } u \in V \setminus \{r\} \text{ with } x(u) \neq 0\text{, i.e., } u \notin \mathcal{F}_1) \\
&= |\mathcal{F}_2'| + |\mathcal{F}_3'| && (\text{by (6)}) \\
&\leq |\mathcal{F}_2'| + |V^{\mathcal{T}}| && (\text{by (7)}),
\end{aligned}$$

which leads to the desired result since

$$|V^{\mathcal{L}}| = |\text{supp}(x)| - |V^{\mathcal{T}}| \leq |\mathcal{F}_2'| \leq L.$$

□

Proof of Lemma 7

Within this proof we focus on protection sets where the budget available for any level is spent on the same level (and not a later one). As discussed, there is always an optimal protection set with this property.

Let $B_\ell \in \mathbb{Z}_{\geq 0}$ be the budget available at level $\ell \in [L]$ and let $\lambda_\ell = \lambda B_\ell$. We construct the tree G' using the following greedy procedure. Process the levels of G from the first one to the last one. At every level

$\ell \in [L]$, pick λ_ℓ vertices $u_1^\ell, \cdots, u_{\lambda_\ell}^\ell$ at the ℓ-th level of G greedily, i.e., pick each next vertex such that the subtree corresponding to that vertex has largest weight among all remaining vertices in the level. After each selection of a vertex the greedy procedure can no longer select any vertex in the corresponding subtree in subsequent iterations.[9]

Now, the tree G' is constructed by deleting from G any vertex that is both not contained in any subtree $T_{u_i^\ell}$, and not contained in any path $P_{u_i^\ell}$ for $\ell \in [L]$ and $i \in [\lambda_\ell]$. In other words, if $U \subseteq V$ is the set of all leaves of G that were disconnected from the root by the greedy algorithm, then we consider the subtree of G induced by the vertices $\cup_{u \in U} P_u$. Finally, the weights of vertices on the paths $P_{u_i^\ell} \setminus \{u_i^\ell\}$ for $\ell \in [L]$ and $i \in [\lambda_\ell]$ are reduced to zero. This concludes the construction of $G' = (V', E')$ and the new weight function w'. Denote by $D_\ell = \{u_1^\ell, \cdots, u_{\lambda_\ell}^\ell\}$ the set of vertices chosen by the greedy procedure in level ℓ. Observe that by construction we have

$$w'(V') = \sum_{\ell \in [L]} \sum_{u \in D_\ell} w'(T'_u).$$

The latter immediately implies the second claim, as

$$\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}})) \geq \sum_{\ell \in [L]} \max_{\substack{S \subseteq D_\ell \\ |S| \leq B_\ell}} \sum_{u \in S} w'(T'_u) \geq \frac{1}{\lambda} w'(V'),$$

and since no two vertices selected by the greedy procedure lie on the same path to the root. In other words, the vertices with non-zero weight in the new tree G' can be partitioned into λ disjoint Firefighter solutions by construction, hence an optimal solution to the Firefighter problem on G' covers at least a $\frac{1}{\lambda}$-fraction of the total weight of G'.

It remains to prove that the first claim holds. Let $S^* = S_1^* \cup \cdots \cup S_L^*$ be the vertices protected in some optimal solution in G, where $S_\ell^* \subseteq V_\ell$ are the vertices protected in level ℓ (and hence $|S_\ell^*| \leq B_\ell$). For distinct vertices $u, v \in V$ we say that u *covers* v if $v \in T_u \setminus \{u\}$.

For $\ell \in [L]$ let $I_\ell = S_l^* \cap D_\ell$ be the set of vertices protected by the optimal solution that are also chosen by the greedy algorithm in level ℓ. Furthermore, let $J_\ell \subseteq S_\ell^*$ be the set of vertices of the optimal solution that are covered by vertices chosen by the greedy algorithm in earlier iterations, i.e., $J_\ell = S_\ell^* \cap \bigcup_{u \in D_1 \cup \cdots \cup D_{\ell-1}} T_u$. Finally, let $K_\ell = S_\ell^* \setminus (I_\ell \cup J_\ell)$ be all other optimal vertices in level ℓ. Clearly, $S_\ell^* = I_\ell \cup J_\ell \cup K_\ell$ is a partition of S_ℓ^*.

Consider a vertex $u \in K_\ell$ for some $\ell \in [L]$. From the guarantee of the greedy algorithm it holds that for every vertex $v \in D_\ell$ we have $w'(T_v) = w(T_v) \geq w(T_u)$. The same does not necessarily hold for covered vertices. On the other hand, covered vertices are contained in G' with their original weights. We exploit these two properties to prove the existence of a solution in G' of almost the same weight as S^*.

To prove the existence of a good solution we construct a solution $A = A_1 \cup \cdots \cup A_L$ with $A_\ell \subseteq V_\ell$ and $|A_\ell| \leq B_\ell$ randomly, and prove a bound on its expected quality. We process the levels of the tree G' top-down to construct A step by step. This clearly does not compromise generality. Recall that we only need to prove the existence of a good solution, and not compute it efficiently. We can hence assume the knowledge of S^* in the construction of A. To this end assume that all levels $\ell' < \ell$ were already processed, and the corresponding sets $A_{\ell'}$ were constructed. The set A_ℓ is constructed as follows:

1. Include in A_ℓ all vertices in I_ℓ.
2. Include in A_ℓ all vertices in J_ℓ that are not covered by vertices in $A_1 \cup \cdots \cup A_{\ell-1}$ (vertices selected so far).

[9] For $\lambda = 1$ this procedure produces a set of vertices, which comprise a $\frac{1}{2}$-approximation for the Firefighter problem, as it coincides with the greedy algorithm of Hartnell and Li [23].

3. Include in A_ℓ a *uniformly random subset* of $|K_\ell|$ vertices from $D_\ell \setminus I_\ell$.

It is easy to verify that the latter algorithm returns a redundancy-free solution, as no two chosen vertices in A lie on the same path to the root. Next, we show that the expected weight of vertices saved by A is at least $(1-\frac{1}{\lambda})\,\mathrm{val}(\mathsf{OPT}(\overline{\mathcal{I}}))$, which will prove our claim, since then at least one solution has the desired quality.

Since we only need a bound on the expectation we can focus on a single level $\ell \in [L]$ and show that the contribution of vertices in A_ℓ is in expectation at least $1-\frac{1}{\lambda}$ times the contribution of the vertices in S^*_ℓ. Observe that the vertices in I_ℓ are contained both in S^*_ℓ and in A_ℓ, hence it suffices to show that the contribution of $A_\ell \setminus I_\ell$ is at least $1-\frac{1}{\lambda}$ times the contribution of $S^*_\ell \setminus I_\ell$, in expectation. Also, recall that every vertex in D_ℓ contributes at least as much as any vertex in K_ℓ, by the greedy selection rule. It follows that the $|K_\ell|$ randomly selected vertices in A_ℓ have at least as much contribution as the vertices in K_ℓ. Consequently, to prove the claim is suffices to bound the expected contribution of vertices in $A_\ell \cap J_\ell$ with respect to the contribution of J_ℓ. Since $A_\ell \cap J_\ell \subseteq J_\ell$ it suffices to show that every vertex $u \in J_\ell$ is also present in A_ℓ with probability at least $1-\frac{1}{\lambda}$.

To bound the latter probability we make use of the random choices in the construction of A as follows. Let $\ell' < \ell$ be the level at which for some $w \in D_{\ell'}$ it holds that $u \in T_w$. In other words, ℓ' is the level that contains the ancestor of u that was chosen by the greedy construction of G'. Now, since S^* is an efficient solution, and by the way that A is constructed it holds that if $u \notin A_\ell$ then $w \in A_{\ell'}$, namely if u is covered, it can only be covered by the unique ancestor w of u that was chosen in the greedy construction of G'. Furthermore, in such a case the vertex w was selected randomly in the third step of the ℓ'-th iteration. Put differently, the probability that the vertex u is covered is exactly the probability that its ancestor w is chosen randomly to be part of $A_{\ell'}$. Since these vertices are chosen to be a random subset of $|K_{\ell'}|$ vertices from the set $D_{\ell'} \setminus I_{\ell'}$, this probability is at most

$$\frac{|K_{\ell'}|}{|D_{\ell'}|-|I_{\ell'}|} = \frac{|K_{\ell'}|}{\lambda B_{\ell'}-|I_{\ell'}|} \leq \frac{1}{\lambda},$$

where the last inequality follows from $|K_{\ell'}|+|I_{\ell'}| \leq B_{\ell'}$. This implies that $u \in A_\ell$ with probability of at least $1-\frac{1}{\lambda}$, as required and concludes the proof of the lemma.

□

Proof of Lemma 9

We construct the set Q in two phases as follows. First we construct a set $\overline{Q} \subseteq H$ of vertices fulfilling the first and the third properties, i.e., it will satisfy $|\overline{Q}| = O(\frac{\log N}{\epsilon^3})$, as well as the property that $G[V \setminus \overline{Q} \cup \{r\}]$ has connected components each of weight at most η. Then, we add to $\overline{Q}$ all vertices of H of degree at least three to arrive at the final set Q.

It will be convenient to define heavy vertices and heavy tree with respect to any subtree $G' = (V', E')$ of G which contains the root r. Concretely, we define $H_{G'} = \{u \in V' \setminus \{r\} \mid w(T'_u) \geq \eta\}$ to be the set of G'-heavy vertices. The G'-heavy tree is the subtree $G'[H_{G'} \cup \{r\}]$ of G'. Observe that $H = H_G$ and that $H_{G'} \subseteq H$ for every subtree G' of G.

To construct $\overline{Q}$ we process the tree G in a bottom-up fashion starting with $\overline{Q} = \emptyset$. We will also remove parts of the tree in the end of every iteration. The first iteration starts with $G' = G$. In every iteration that starts with tree G', include in $\overline{Q}$ an arbitrary leaf $u \in H_{G'}$ of the heavy tree and remove u and all vertices in its subtree from G'. The procedure ends when r is the only remaining vertex in the heavy tree.

Let us verify that the claimed properties indeed hold. The fact that $|\overline{Q}| = O(\frac{\log N}{\epsilon^3})$ follows from the fact that at each iteration we remove a G'-heavy vertex including all its subtree from the current tree G'. This implies that the total weight of the tree G' decreases by at least η in every iteration. Since we only include one vertex in every iteration we have $|\overline{Q}| \le \frac{w(V)}{\eta} = O(\frac{\log N}{\epsilon^3})$.

The third property follows from the fact that we always remove a leaf of the G'-heavy tree. Observe that the connected components of $G[V \setminus (\overline{Q} \cup \{r\})]$ are contained in the subtrees we disconnect in every iteration in the construction of $\overline{Q}$. By definition of G'-heavy leaves, in any such iteration where a G'-heavy leaf u is removed from the tree, these parts have weight at least η, but any subtree rooted at any descendant of u has weight strictly smaller than η (otherwise this descendant would be G'-heavy as well, contradicting the assumption that it has a G'-heavy leaf u as an ancestor). Now, since u is included in $\overline{Q}$, the connected components are exactly these subtrees, so the property indeed holds.

To construct Q and conclude the proof it remains to include in $\overline{Q}$ all remaining nodes of degree at least three in the heavy tree. The fact that also all leaves of the heavy tree are included in Q is readily implied by the construction of $\overline{Q}$, so the second property holds for Q. Clearly, by removing more vertices from the heavy tree, the sizes of connected components only get smaller, so Q also satisfies the third condition, since $\overline{Q}$ already did. Finally, the number of vertices of degree at least three in the heavy tree is strictly less than the number of its leaves, which is $O(\frac{\log N}{\epsilon^3})$; for otherwise a contradiction would occur since the tree would have an average degree of at least 2. This implies that, in total, $|Q| = O(\log N)$, so the first property also holds.

To conclude the proof of the lemma it remains to note that the latter construction can be easily implemented in polynomial time.

□

6 Missing details for $O(1)$-approximation for RMFC

This section contains the missing proofs for our 12-approximation for RMFC.

Proof of Theorem 12

To prove Theorem 12 we first show the following result, based on which Theorem 12 follows quite directly.

Lemma 18. *Let $B \in \mathbb{R}_{\ge 0}$, $\eta \in (0,1]$, $k \in \mathbb{Z}_{\ge 1}$, and $\ell_1 = \lfloor \log^{(k)} L \rfloor$, $\ell_2 = \lfloor \log^{(k-1)} L \rfloor$. Let $x \in P_B$ with $\text{supp}(x) \subseteq V_{(\ell_1,\ell_2]} := V_{>\ell_1} \cap V_{\le \ell_2}$, and we define $Y = \{u \in \Gamma \mid x(P_u) \ge \eta\}$. Then one can efficiently compute a set $R \subseteq V_{(\ell_1,\ell_2]}$ such that*

(i) $R \cap P_u \neq \emptyset \quad \forall u \in Y$, *and*
(ii) $\chi^R \in P_{B'}$, *where* $\bar{B} = \frac{1}{\eta} B + 1$.

We first observe that Lemma 18 indeed implies Theorem 12.

Proof of Theorem 12. For $k = 1, \dots, q$, let $\ell_1^k = \lfloor \log^{(k)} L \rfloor$ and $\ell_2^k = \lfloor \log^{(k-1)} L \rfloor$, and we define $x^k \in P_B$ by $x^k = x \wedge \chi^{V_{(\ell_1^k,\ell_2^k]}}$. Hence, $x = \sum_{k=1}^{k} x^k$. For each $k \in [q]$, we apply Lemma 18 to x^k with $\eta = \frac{\mu}{q}$ to obtain a set $R^k \subseteq V_{(\ell_1^k,\ell_2^k]}$ satisfying

(i) $R^k \cap P_u \neq \emptyset \quad \forall u \in Y^k = \{u \in \Gamma \mid x^k(P_u) \ge \eta\}$, and
(ii) $\chi^{R^k} \in P_{B'}$, where $B' := \frac{q}{\mu} B + 1 = \frac{1}{\eta} B + 1 =: \bar{B}$.

We claim that $R = \cup_{k=1}^{q} R^k$ is a set satisfying the conditions of Theorem 12. The set R clearly satisfies $\chi^R \in P_{B'}$ since $\chi^{R^k} \in P_{B'}$ for $k \in [q]$ and the sets R^k are on disjoint levels. Furthermore, for each $u \in W = \{v \in \Gamma \mid x(P_v) \ge \mu\}$ we indeed have $P_u \cap R \neq \emptyset$ due to the following. Since $x = \sum_{k=1}^{q} x^k$ and $x(P_u) \ge \mu$ there exists an index $j \in [q]$ such that $x^j(P_u) \ge \eta = \frac{\mu}{q}$, and hence $P_u \cap R \supseteq P_u \cap R^j \neq \emptyset$.

□

Thus, it remains to prove Lemma 18.

Proof of Lemma 18.

Let $\tilde{B} = \frac{1}{\eta}B$. We start by determining an optimal vertex solution y to the linear program $\min\{z(V \setminus \{r\}) \mid z \in Q\}$, where

$$Q = \{z \in P_{\tilde{B}} \mid z(u) = 0 \ \forall u \in V \setminus (V_{(\ell_1,\ell_2]} \cup \{r\}), \ \ z(P_u) \geq 1 \ \forall u \in Y\}.$$

Notice that $Q \neq \emptyset$ since $\frac{1}{\eta}x \in Q$; hence, the above LP is feasible. Furthermore, notice that $y(P_u) \leq 1$ for $u \in \Gamma$; for otherwise, there is a vertex $v \in \mathrm{supp}(y)$ such that $y(P_v) > 1$, and hence $y - \epsilon\chi^{\{v\}} \in Q$ for a small enough $\epsilon > 0$, violating that y is an *optimal* vertex solution.

Let $V^{\mathcal{L}}$ be all y-loose vertices. We will show that the set

$$R = V^{\mathcal{L}} \cup \{u \in V \setminus \{r\} \mid y(u) = 1\}$$

fulfills the properties claimed by the lemma. Clearly, $R \subseteq V_{(\ell_1,\ell_2]}$ since $\mathrm{supp}(y) \subseteq V_{(\ell_1,\ell_2]}$.

To see that condition (i) holds, let $u \in Y$, and notice that we have $y(P_u) = 1$. Either $|P_u \cap \mathrm{supp}(y)| = 1$, in which case the single vertex v in $P_u \cap \mathrm{supp}(y)$ satisfies $y(u) = 1$ and is thus contained in R; or $|P_u \cap \mathrm{supp}(y)| > 1$, in which case $P_u \cap V^{\mathcal{L}} \neq \emptyset$ which again implies $R \cap P_u \neq \emptyset$.

To show that R satisfies (ii), we have to show that R does not exceed the budget $B' \cdot 2^\ell = (\frac{1}{\eta}B + 1)2^\ell$ of any level $\ell \in \{\ell_1 + 1, \ldots, \ell_2\}$. We have

$$|R \cap V_\ell| \leq y(V_\ell) + |V^{\mathcal{L}}| \leq \tilde{B}2^\ell + |V^{\mathcal{L}}| = \frac{1}{\eta}B2^\ell + |V^{\mathcal{L}}|,$$

where the second inequality follows from $y \in Q$. To complete the proof it suffices to show $|V^{\mathcal{L}}| \leq 2^\ell$. This follows by a sparsity reasoning analogous to Lemma 6 implying that the number of y-loose vertices is bounded by the number of tight budget constraints, and thus

$$|V^{\mathcal{L}}| \leq \ell_2 - \ell_1 \leq \ell_2 = \lfloor \log^{(k-1)} L \rfloor. \tag{8}$$

Furthermore,

$$2^\ell \geq 2^{\ell_1+1} = 2^{\lfloor \log^{(k)} L \rfloor + 1} \geq 2^{\log^{(k)} L} = \log^{(k-1)} L,$$

which, together with (8), implies $|V^{\mathcal{L}}| \leq 2^\ell$ and thus completes the proof.

□

Proof of Theorem 13

Let (y, B) be an optimal solution to the RMFC relaxation $\min\{B \mid x \in \bar{P}_B\}$ and let $h = \lfloor \log L \rfloor$. Hence, $B \leq B_{\mathsf{OPT}}$. We invoke Theorem 12 with respect to the vector $y \wedge \chi^{V_{>h}}$ and $\mu = 0.5$ to obtain a set $R_1 \subseteq V_{>h}$ satisfying

(i) $R_1 \cap P_u \neq \emptyset \quad \forall u \in W$, and

(ii) $\chi^{R_1} \in P_{2B+1}$,

where $W = \{u \in \Gamma \mid y(P_u \cap V_{>h}) \geq 0.5\}$. Hence, R_1 cuts off all leaves in W from the root by only protecting vertices on levels $V_{>h}$ and using budget bounded by $2B + 1 \leq 3B \leq 3\max\{\log L, B_{\mathsf{OPT}}\}$.

We now focus on the leaves $\Gamma \setminus W$, which we will cut off from the root by protecting a vertex set $R_2 \subseteq V_{\leq h}$ feasible for budget $3\max\{\log L, B_{\mathsf{OPT}}\}$. Let $(z, \bar{B})$ be an optimal vertex solution to the following linear program

$$\min\left\{\bar{B} \;\middle|\; x \in P_{\bar{B}},\ x(P_u) = 1 \ \forall u \in \Gamma \setminus W\right\}. \tag{9}$$

First, notice that (9) is feasible for $\bar{B} \leq 2B$. This follows by observing that the vector $q = 2(y \cap \chi^{V_{\leq h}})$ satisfies $q \in P_{2B}$ since $y \in P_B$. Moreover, for $u \in \Gamma \setminus W$, we have

$$q(P_u) = 2y(P_u \cap V_{\leq h}) = 2(1 - y(P_u \cap V_{>h})) > 1,$$

where the last inequality follows from $y(P_u \cap V_{>h}) < 0.5$ because $u \in \Gamma \setminus W$. Finally, there exists a vector $q' < q$ such that $q'(P_u) = 1$ for $u \in \Gamma \setminus W$. The vector q' can be obtained from q by successively reducing values on vertices $v \in \mathrm{supp}(q)$ satisfying $q(P_v) > 1$. This shows that $(q', 2B)$ is a feasible solution to (9) and hence $\bar{B} \leq 2B$.

Consider the set of all z-loose vertices $V^{\mathcal{L}} = \{u \in \mathrm{supp}(z) \mid z(P_u) < 1\}$. We define

$$R_2 = V^{\mathcal{L}} \cup \{u \in \mathrm{supp}(z) \mid z(u) = 1\}.$$

Notice that for each $u \in \Gamma \setminus W$, the set R_2 contains a vertex on the path from u to the root. Indeed, either $|\mathrm{supp}(z) \cap P_u| = 1$ in which case there is a vertex $v \in P_u$ with $z(v) = 1$, which is thus contained in R_2, or $|\mathrm{supp}(z) \cap P_u| > 1$ in which case the vertex $v \in \mathrm{supp}(z) \cap P_u$ that is closest to the root among all vertices in $\mathrm{supp}(z) \cap P_u$ is a z-loose vertex. Hence, the set $R = R_1 \cup R_2$ cuts off all leaves from the root. It remains to show that it is feasible for budget $3\max\{\log L, B_{\mathsf{OPT}}\}$.

Using an analogous sparsity reasoning as in Lemma 6, we obtain that $|V^{\mathcal{L}}|$ is bounded by the number of tight budget constraints, which is at most $h = \lfloor \log L \rfloor \leq \log L$. Hence, for any level $\ell \in [h]$, we have

$$\begin{aligned}
|R_2 \cap V_\ell| &\leq |V^{\mathcal{L}}| + z(V_\ell) \\
&\leq \log L + 2^\ell \bar{B} && ((z, \bar{B}) \text{ feasible for } (9)) \\
&\leq \log L + 2^\ell \cdot (2B) && (\bar{B} \leq 2B) \\
&\leq 2^\ell \cdot (3\max\{\log L, B_{\mathsf{OPT}}\}). && (B \leq B_{\mathsf{OPT}})
\end{aligned}$$

Thus, both R_1 and R_2 are budget-feasible for budget $3\max\{\log L, B_{\mathsf{OPT}}\}$, and since they contain vertices on disjoint levels, $R = R_1 \cup R_2$ is feasible for the same budget.

□

Proof of Lemma 15

To show that the running time of $\mathrm{Enum}(\emptyset, \emptyset, \bar{\gamma})$ is polynomial, we show that there is only a polynomial number of recursive calls to $\mathrm{Enum}(A, D, \gamma)$. Notice that the number of recursive calls done in one execution of step 4 of the algorithm is equal to $2|F_x|$. We thus start by upper bounding $|F_x|$ for any solution (x, B) to $\mathrm{LP(A,D)}$ with $B < \log L$. Consider a vertex $f_u \in F_x$, where $u \in \Gamma \setminus W_x$. Since u is a leaf not in W_x, we have $x(P_u \cap V_{\leq h}) > \frac{1}{3}$, and thus

$$x(T_{f_u} \cap V_{\leq h}) > \frac{1}{3} \quad \forall f_u \in F_x.$$

Because no two vertices of F_x lie on the same leaf-root path the sets $T_{f_u} \cap V_{\leq h}$ are all disjoint for different $f_u \in F_x$ and hence

$$\begin{aligned}
\frac{1}{3}|F_x| &< \sum_{f \in F_x} x(T_f \cap V_{\leq h}) \\
&\leq x(V_{\leq h}) && (\text{disjointness of sets } T_f \cap V_{\leq h} \text{ for different } f \in F_x) \\
&\leq \sum_{\ell=1}^{h} 2^\ell B && (x \text{ satisfies budget constraints of } \mathrm{LP(A,D)}\ ) \\
&< 2^{h+1} B \\
&< 2(\log L)^2. && (h = \lfloor \log^{(2)} L \rfloor \text{ and } B < \log L)
\end{aligned}$$

Since the recursion depth is $\bar{\gamma} = 2(\log L)^2 \log^{(2)} L$, the number of recursive calls is bounded by

$$O\left((2|F_x|)^{\bar{\gamma}}\right) = (\log L)^{O((\log L)^2 \log^{(2)} L)} = 2^{o(L)} = o(N),$$

thus showing that $\text{Enum}(\emptyset, \emptyset, \bar{\gamma})$ runs in polynomial time.

It remains to show that $\text{Enum}(\emptyset, \emptyset, \bar{\gamma})$ finds a triple satisfying the conditions of Lemma 14. For this we identify a particular execution path of the recursive procedure $\text{Enum}(\emptyset, \emptyset, \bar{\gamma})$ that, at any point in the algorithm, will maintain two disjoint sets $A, D \subseteq V_{\leq h}$ that are compatible with OPT, i.e., $A \subseteq \mathsf{OPT}$ and $D \cap \mathsf{OPT} = \emptyset$. At the beginning of the algorithm we clearly have compatibility with OPT since $A = D = \emptyset$. To identify the execution path we are interested in, we highlight which recursive call we want to follow given that we are on the execution path. Hence, consider two disjoint sets $A, D \subseteq V_{\leq h}$ that are compatible with OPT and assume we are within the execution of $\text{Enum}(A, D, \gamma)$. Let (x, B) be an optimal solution to LP(A, D). Notice that $B \leq B_{\mathsf{OPT}} \leq \log L$, because (A, D) is compatible with OPT. If $\mathsf{OPT} \cap Q_x = \emptyset$, then (A, D, x) fulfills the conditions of Lemma 14 and we are done. Hence, assume $\mathsf{OPT} \cap Q_x \neq \emptyset$, and let $f \in F_x$ be such that $\mathsf{OPT} \cap T_f \cap V_{\leq h} \neq \emptyset$. If $f \in \mathsf{OPT}$, then consider the execution path continuing with the call of $\text{Enum}(A \cup \{f\}, D, \gamma - 1)$; otherwise, if $f \notin \mathsf{OPT}$, we focus on the call of $\text{Enum}(A, D \cup P_f, \gamma - 1)$. Notice that compatibility with OPT is maintained in both cases.

To show that the thus identified execution path of $\text{Enum}(\emptyset, \emptyset, \bar{\gamma})$ indeed leads to a triple satisfying the conditions of Lemma 14, we measure progress as follows. For any pair $A, D \subseteq V_{\leq h}$ of disjoint sets compatible with OPT, we define a potential function $\Phi(A, D) \in \mathbb{Z}_{\geq 0}$ as follows. For each $u \in \mathsf{OPT} \cap V_{\leq h}$, let $d_u \in \mathbb{Z}_{\geq 0}$ be the distance of u to the first vertex in $A \cup D \cup \{r\}$ when following the unique u-r path. We define $\Phi(A, D) = \sum_{u \in \mathsf{OPT} \cap V_{\leq h}} d_u$. Notice that as long as we have a triple (A, D, x) on our execution path that does not satisfy the conditions of Lemma 14, then the next triple (A', D', x') on our execution path satisfies $\Phi(A', D') < \Phi(A, D)$. Clearly, latest when having a triple (A, D, x) compatible with OPT and $\Phi(A, D) = 0$, then $\mathsf{OPT} \cap V_{\leq h} = A$ and we thus correctly guessed all vertices of $\mathsf{OPT} \cap V_{\leq h}$ implying that the conditions of Lemma 14 are satisfied for the triple (A, D, x). Since $\Phi(A, D) \geq 0$ for any compatible sets A and D, this implies that a triple satisfying the conditions of Lemma 14 will be encountered if the recursion depth $\bar{\gamma}$ is at least $\Phi(\emptyset, \emptyset)$. To evaluate $\Phi(\emptyset, \emptyset)$ we have to compute the sum of the distances of all vertices $u \in \mathsf{OPT} \cap V_{\leq h}$ to the root. The distance of u to the root is at most h since $u \in V_{\leq h}$. Moreover, $|\mathsf{OPT} \cap V_{\leq h}| < 2^{h+1} B_{\mathsf{OPT}}$ due to the budget constraints. Hence,

$$\begin{aligned}
\Phi(\emptyset, \emptyset) &< h \cdot 2^{h+1} \cdot B_{\mathsf{OPT}} \\
&\leq 2 \log^{(2)} L \cdot (\log L)^2 && (h = \lfloor \log^{(2)} L \rfloor \text{ and } B_{\mathsf{OPT}} \leq \log L) \\
&= \bar{\gamma},
\end{aligned}$$

implying that a triple fulfilling the conditions of Lemma 14 is encountered by $\text{Enum}(\emptyset, \emptyset, \bar{\gamma})$.

□

Acknowledgements

We are grateful to Noy Rotbart for many stimulating discussions and for bringing several relevant references to our attention.

References

[1] E. Anshelevich, D. Chakrabarty, A. Hate, and C. Swamy. Approximability of the firefighter problem. *Algorithmica*, 62(1-2):520–536, 2012.

[2] C. Bazgan, M. Chopin, M. Cygan, M. R. Fellows, F. Fomin, and E. J. van Leeuwen. Parameterized complexity of firefighting. *Journal of Computer and System Sciences*, 80(7):1285–1297, 2014.

[3] C. Bazgan, M. Chopin, and M. R. Fellows. Parameterized complexity of the firefighter problem. In *Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC)*, pages 643–652. Springer-Verlag, 2011.

[4] C. Bazgan, M. Chopin, and B. Ries. The firefighter problem with more than one firefighter on trees. *Discrete Applied Mathematics*, 161(7):899–908, 2013.

[5] L. Cai, Y. Cheng, E. Verbin, and Y. Zhou. Surviving rates of graphs with bounded treewidth for the firefighter problem. *SIAM Journal on Discrete Mathematics*, 24(4):1322–1335, 2010.

[6] L. Cai, E. Verbin, and L. Yang. Firefighting on trees:$(1-1/e)$–approximation, fixed parameter tractability and a subexponential algorithm. In *Proceedings of the 19th International Symposium on Algorithms and Computation (ISAAC)*, pages 258–269. Springer-Verlag, 2008.

[7] L. Cai and W. Wang. The surviving rate of a graph for the firefighter problem. *SIAM Journal on Discrete Mathematics*, 23(4):1814–1826, 2009.

[8] P. Chalermsook and J. Chuzhoy. Resource minimization for fire containment. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1334–1349, 2010.

[9] C. Chekuri and A. Kumar. Maximum coverage problem with group budget constraints and applications. In *International Workshop an Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 72–83. Springer, 2004.

[10] V. Costa, S. Dantas, M. C. Dourado, L. Penso, and D. Rautenbach. More fires and more fighters. *Discrete Applied Mathematics*, 161(16):2410–2419, 2013.

[11] G. Călinescu, C. Chekuri, M. Pál, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[12] M. Cygan, F. Fomin, and E. van Leeuwen. Parameterized complexity of firefighting revisited. In *Parameterized and Exact Computation*, pages 13–26. Springer, 2012.

[13] M. Elkin and G. Kortsarz. An approximation algorithm for the directed telephone multicast problem. *Algorithmica*, 45(4):569–583, 2006.

[14] L. Esperet, J. van den Heuvel, F. Maffray, and F. Sipma. Fire containment in planar graphs. *Journal of Graph Theory*, 73(3):267–279, 2013.

[15] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45:634–652, 1998.

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

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

[18] P. Floderus, A. Lingas, and M. Persson. Towards more efficient infection and fire fighting. *International Journal of Foundations of Computer Science*, 24(01):3–14, 2013.

[19] F. V. Fomin, P. Heggernes, and E. J. van Leeuwen. Making life easier for firefighters. In *Fun with Algorithms*, pages 177–188. Springer, 2012.

[20] P. Gordinowicz. Planar graph is on fire. *arXiv preprint arXiv:1311.1158*, 2013.

[21] F. Grandoni, R. Ravi, M. Singh, and R. Zenklusen. New approaches to multi-objective optimization. *Mathematical Programming, Series A*, 146(1):525–554, 2014.

[22] B. Hartnell. Firefighter! an application of domination. In *24th Manitoba Conference on Combinatorial Mathematics and Computing*, 1995.

[23] B. Hartnell and Q. Li. Firefighting on trees: how bad is the greedy algorithm? In *Proceedings of Congressus Numerantium*, volume 145, pages 187–192, 2000.

[24] Y. Iwaikawa, N. Kamiyama, and T. Matsui. Improved approximation algorithms for firefighter problem on trees. *IEICE Transactions on Information and Systems*, 94(2):196–199, 2011.

[25] A. King and G. MacGillivray. The firefighter problem for cubic graphs. *Discrete Mathematics*, 310(3):614–621, 2010.

[26] R. Klein, C. Levcopoulos, and A. Lingas. Approximation algorithms for the geometric firefighter and budget fence problems. In *11th Latin American Symposium on Theoretical Informatics (LATIN)*, pages 261–272. Springer, 2014.

[27] J. Kong, L. Zhang, and W. Wang. The surviving rate of digraphs. *Discrete Mathematics*, 334:13–19, 2014.

[28] G. MacGillivray and P. Wang. On the firefighter problem. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 47:83–96, 2003.

[29] P. Pralat. Sparse graphs are not flammable. *SIAM Journal on Discrete Mathematics*, 27(4):2157–2166, 2013.

[30] P. Pralat. Graphs with average degree smaller than $\frac{30}{11}$ burn slowly. *Graphs and Combinatorics*, 30(2):455–470, 2014.

[31] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem. In *Proceedings of 5th Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 66–75, 1996.

[32] A. Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24. Springer, 2003.

[33] J. Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 67–74, 2008.

A Basic transformations for the Firefighter problem

In this section we provide some basic transformations showing how different natural variations of the Firefighter problem can be reduced to each other. We start by proving Lemma 3.

Proof of Lemma 3. Consider an instance of the weighted Firefighter problem with general budgets consisting of a tree $G = (V, E)$ of depth L rooted at the vertex $r \in V$, weights $w(u) \in \mathbb{Z}_{\geq 0}$ for all $u \in V \setminus \{r\}$ and budgets $B_\ell \in \mathbb{Z}_{>0}$ for all $\ell \in [L]$. We transform the instance into an equivalent instance with unit budgets by performing the following simple steps for all levels V_ℓ for $\ell \in [L]$:

- For every $u \in V_\ell$, subdivide the edge connecting u to its ancestor in G into a path with B_ℓ edges, by introducing $B_\ell - 1$ new vertices. Denote the nodes on this path, excluding the ancestor of u in G, by Y_u.
- Set the weight of all new vertices to zero, while maintaining the weight $w(u)$ for the original vertex u.

Denote the resulting tree by $G' = (V', E')$. To conclude the construction it remains to allow one unit of budget in every level of the transformed tree. It is easy to verify that feasible solutions to the Firefighter problem for the two instances are in correspondence. A feasible solution for G is transformed to a solution in G' by replacing the B_ℓ vertices S_ℓ protected in any level V_ℓ of G with any B_ℓ vertices on the corresponding paths $\{Y_u \mid u \in S_\ell\}$ in G', one in each of the B_ℓ distinct levels of G' that are in correspondence with V_ℓ. The opposite transformation selects for every protected vertex $u \in V'$ in a feasible solution for G' the vertex $u \in V$ such that $u' \in Y_u$. It is straightforward to verify that in both transformations the obtained solutions are feasible and that they have weights identical to the original solutions.

Finally, since $B_\ell \leq n$ can be assumed for every $\ell \in [L]$, each one of the $n-1$ edges in G is subdivided into a path of length at most n, thus the number of vertices in G' is at most $O(n^2)$.

□

We remark that a construction analogous to the one used in the proof of Lemma 3 can be used to show that RMFC with non-uniform budgets can be reduced to the uniform budget case. In an RMFC instance with non-uniform budgets, the budget on level ℓ is equal to $B \cdot \lambda_\ell$, where $\lambda_\ell \in \mathbb{Z}_{>0}$ for $\ell \in [L]$ are given as input, and the goal is still to find the minimum B to protect vertices that cut off all leaves from the root and fulfill the budget constraints.

Next, we show how a weighted instance of the Firefighter problem can be transformed into a unit-weight one with only an arbitrarily small loss in term of the objective function.

Lemma 19. *Let $\delta > 0$ and $\alpha \in (0, 1]$. Any weighted unit-budget Firefighter problem on a tree $G = (V, E)$ and weights $w(u) \in \mathbb{Z}_{\geq 0}$ for $u \in V \setminus \{r\}$ can be transformed efficiently into a polynomial-size unit-weight unit-budget Firefighter problem on a tree $G' = (V', E')$ such that any α-approximate feasible solution for G' can be efficiently transformed into a $(1-\delta)\alpha$-approximate solution for G.*

Proof. For simplicity we present the transformation in two steps, each losing an arbitrarily small constant in the objective. First we use a standard scaling and rounding technique to obtain a new weight function that is bounded by a polynomial in the size of the tree. Concretely, we construct weights $w'(u) \in \mathbb{Z}_{\geq 0}$ for $u \in V \setminus \{r\}$ and an integer $D \in \mathbb{Z}_{>0}$ such that $w'(u) = O(\frac{n}{\delta})$ for every $u \in V$ and such that for every $S \subseteq V \setminus \{r\}$:

$$Dw'(S) \leq w(S) \leq Dw'(S) + \delta \, \mathrm{val}(\mathsf{OPT}),$$

where $\mathrm{val}(\mathsf{OPT})$ is the optimal solution value in G. We then we use the obtained instance to construct a unit-weight instance with the desired property.

Let $w_{\max} = \max_{u \in V \setminus \{r\}} w(u)$ be the maximum weight of any vertex in G. Define $D = \delta w_{\max}/n$ and for every $u \in V \setminus \{r\}$ set $w'(u) = \lfloor w(u)/D \rfloor$. Observe that $\mathrm{val}(\mathsf{OPT}) \geq w_{\max}$, and hence $nD = \delta w_{\max} \leq \delta \, \mathrm{val}(\mathsf{OPT})$. The latter scaling indeed fulfills the desired properties, as $w'(u) \leq n/\delta$, and for every $S \subseteq V \setminus \{r\}$ we have

$$Dw'(S) \leq w(S) \leq Dw'(S) + D|S| \leq Dw'(S) + \delta \, \mathrm{val}(\mathsf{OPT}).$$

We show next that the latter transformation looses at most a δ-fraction in the objective function. Let $S \subseteq V \setminus \{r\}$ be the vertices saved from the fire in an α-approximate solution for G'. We show that S is a

$(1-\delta)\alpha$-approximate solution for G. Let $S^* \subseteq V \setminus \{r\}$ be the vertices saved in an optimal solution for G. Then $Dw'(S^*) + \delta \operatorname{val}(\mathsf{OPT}) \geq w(S^*) = \operatorname{val}(\mathsf{OPT})$, implying that $Dw'(S^*) \geq (1-\delta) \operatorname{val}(\mathsf{OPT})$. We conclude:

$$(1-\delta) \operatorname{val}(\mathsf{OPT}) \leq Dw'(S^*) \leq \frac{1}{\alpha} Dw'(S) \leq \frac{1}{\alpha} w(S),$$

which yields $w(S) \geq (1-\delta)\alpha \operatorname{val}(\mathsf{OPT})$, as desired.

Next we present the second transformation, which, given a weighted Firefighter problem with tree $G = (V, E)$ and integer weights $w(u) \in \mathbb{Z}_{\geq 0}$ bounded by $O(n)$, transforms it into a unit-weight instance on a new tree $G' = (V', E')$ by losing an arbitrarily small factor $\delta > 0$ in terms of the weight.

The tree G' is obtained from G by taking a copy of G and attaching $\frac{2n}{\alpha\delta} w(u)$ new leaves to every vertex $u \in V \setminus \{r\}$. For a set of vertices $S \subseteq V \setminus \{r\}$ let $\operatorname{sv}(S) \in \mathbb{Z}_{\geq 0}$ and $\operatorname{sv}'(S) \in \mathbb{Z}_{\geq 0}$ denote the total weight of vertices saved by protecting the vertices of S in G and G', respectively. Recall that the weight of vertices in G is measured with respect to the function w, and hence $\operatorname{sv}(S) = w(\cup_{u \in S} T_u)$, whereas the weight in G' corresponds to the number of saved vertices, and hence $\operatorname{sv}'(S) = |\cup_{u \in S} T'_u|$.

Consider a solution that protects a set $S \subseteq V' \setminus \{r\}$ of vertices in G'. Observe that $V \cap S$ is a feasible set of vertices to protect in G. We can now write

$$\frac{\alpha\delta}{2n} \cdot \operatorname{sv}'(S) - \alpha\delta \leq \operatorname{sv}(S \cap V) \leq \frac{\alpha\delta}{2n} \cdot \operatorname{sv}'(S),$$

where the first inequality follows from

$$\operatorname{sv}'(S) = |S \setminus V| + \sum_{u \in V \cap S} \frac{2n}{\alpha\delta} w(T_u) + \sum_{u \in V \cap S} |T_u| \leq 2n + \sum_{u \in V \cap S} \frac{2n}{\alpha\delta} w(T_u) = 2n + \frac{2n}{\alpha\delta} \operatorname{sv}(S \cap V).$$

Using the trivial lower bound $\mathsf{OPT} \geq 1$ on the optimal solution for G one can now conclude the proof analogously to the first transformation.

Finally, both transformations can be implemented in polynomial time. For the first transformation this is trivial, while for the second transformation one uses the fact that the input weights are polynomially bounded, and hence G' has polynomial size.

□