Sequential Metric Dimension *

Julien Bensmail[1], Dorian Mazauric[2], Fionn Mc Inerney[1], Nicolas Nisse[1], and Stéphane Pérennes[1]

[1] Université Côte d'Azur, Inria, CNRS, I3S, France
[2] Université Côte d'Azur, Inria, France

Abstract. In the localization game, introduced by Seager in 2013, an invisible and immobile target is hidden at some vertex of a graph G. At every step, one vertex v of G can be probed which results in the knowledge of the distance between v and the secret location of the target. The objective of the game is to minimize the number of steps needed to locate the target whatever be its location.
We address the generalization of this game where $k \geq 1$ vertices can be probed at every step. Our game also generalizes the notion of the *metric dimension* of a graph. Precisely, given a graph G and two integers $k, \ell \geq 1$, the LOCALIZATION Problem asks whether there exists a strategy to locate a target hidden in G in at most ℓ steps and probing at most k vertices per step. We first show that this problem is NP-complete when k (resp., ℓ) is a fixed parameter.
Our main results concern the study of the LOCALIZATION Problem in the class of trees. We prove that this problem is NP-complete in trees when k and ℓ are part of the input. On the positive side, we design a (+1)-approximation in n-node trees, *i.e.*, an algorithm that computes in time $O(n \log n)$ (independent of k) a strategy to locate the target in at most one more step than an optimal strategy. This algorithm can be used to solve the LOCALIZATION Problem in trees in polynomial-time if k is fixed.

Keywords: Games in graphs, metric dimension, complexity.

1 Introduction

Localization (or *Identification*) problems consist of distinguishing the vertices of a graph $G = (V, E)$ using a smallest subset $R \subseteq V$ of its vertices. Many variants have been studied depending on how a subset of vertices allows to identify other vertices. For instance, *identifying codes* [13] and *locating dominating sets* [18] ask for the vertices to be distinguished by their neighbourhood in R. Another well studied example is the one of a *resolving set* [10, 17] which aims at distinguishing every vertex of a graph by their distances to each vertex of this set. Given a graph

* This work has been partially supported by ANR program "Investments for the Future" under reference ANR-11- LABX-0031-01, the Inria Associated Team AlDyNet.
Due to lack of space, full versions of the omitted or sketched proofs are postponed to the Appendix.

G, the main problem is to compute a resolving set with minimum size, called the *metric dimension* of G [10, 17]. The corresponding decision problem is NP-complete in planar graphs [6] and in graphs of diameter 2 [9], and W[2]-hard (parameterized by the solution's size) [11]. On the positive side, the problem is FPT in the class of graphs with bounded treelength [1]. Bounds on the metric dimension have also been determined for various graph classes [8]. In this paper, we address a *sequential* variant of this problem.

Let us consider a graph $G = (V, E)$ where an unknown vertex $t \in V$ hosts a hidden (invisible) and immobile target. *Probing* one vertex $v \in V$ results in the knowledge of the distance between t and v, denoted by $d_G(v, t)$. Probing a set $R \subseteq V$ of vertices results in the distance vector $(d_G(v, t))_{v \in R}$ and a set is a *resolving set* if the distance vectors are pairwise distinct for every $t \in V$. The *metric dimension* of G, denoted by $MD(G)$, is then the minimum number of vertices that must be probed simultaneously (in one step) to determine the location t of the target wherever it is. For instance, in the case of a path, probing one of its ends is sufficient to locate the target, *i.e.*, $MD(P) = 1$ for every path P. Another example (that we use throughout the paper) is the case of a star (tree with a universal vertex) with n leaves, denoted by S_n, for which it is necessary and sufficient to probe every leaf but one, *i.e.*, $MD(S_n) = n - 1$.

If less than $MD(G)$ vertices can be probed at once, it is natural to allow more than one step. Obviously, if at most $1 \leq k < MD(G)$ vertices can be probed at once, it is always feasible to locate an immobile target in $\lceil MD(G)/k \rceil$ steps, simply by sequentially probing k different vertices of a smallest resolving set at each step. However, there are graphs for which the target can be located much faster (see Claim 8 in Appendix 5.3). In [15], Seager initiated the study of the following sequential locating game: at every step, one vertex of a graph can be probed, and the objective is to minimize the number of steps required to locate the target wherever it is. Seager gave bounds and exact values on this minimum number of steps in particular subclasses of trees (e.g., subdivisions of caterpillars) [15]. In this paper, we study the generalization of this game where $k \geq 1$ vertices can be probed at every step.

Precisely, let $k \geq 1$ be an integer and let $G = (V, E)$ be a graph hosting an invisible and immobile target hidden at $t \in V$. A *k-strategy* is allowed to probe at most k vertices at each step of the game (where the choice of the probed vertices at some step may obviously depend on the results of the probes during previous steps) until the location t of the target is uniquely determined. Let $\lambda_k(G)$ denote the minimum integer h such that there exists a k-strategy that locates the target in G in at most h steps, whatever be the location of the target. Given G, k and $\ell \geq 1$, the LOCALIZATION Problem asks whether $\lambda_k(G) \leq \ell$. We also consider the dual parameter $\kappa_\ell(G)$ defined as the minimum integer h such that there exists an h-strategy that locates the target in G in at most ℓ steps. Note that, for every graph G, $\kappa_1(G)$ is exactly the metric dimension $MD(G)$ of G, and $\lambda_k(G) \leq \ell$ if and only if $\kappa_\ell(G) \leq k$. We are interested in the complexity of the LOCALIZATION Problem in general graphs and particularly in trees.

1.1 Related Work

Moving target. Sequential games related to resolving sets have first been introduced and mainly studied in the case of a mobile target. That is, at every step, some vertices may be probed and, if the target has not been located yet, it may move to one of its neighbours (sometimes, it is required that the target cannot move to a vertex that has been probed during the previous step which is called the "no-backtrack condition") [14]. In this setting, locating the target may not be feasible. For instance, it is not possible to locate a moving target in a triangle when probing one vertex per step if the target may "backtrack". The question of how many times all the edges of a graph must be subdivided to ensure locating a moving target probing 1 vertex (resp., k vertices) per step has been addressed in [5] (resp. [12]). Let a graph be called *locatable* if there exists a 1-strategy for locating the target in a finite number of steps with the "no-backtrack condition". The case of trees with the "no-backtrack condition" has first been studied in [14] where it was shown that all trees T are locatable, and in [4], the upper bound on the number of steps it takes to locate the target in T was improved. In [16], the case of trees where the target may "backtrack" was considered. Let $\zeta(G)$ be the minimum integer k such that there exists a k-strategy for locating a moving target in G. In [3], it was shown that deciding whether $\zeta(G) \leq k$ is NP-hard and that $\zeta(G)$ is not bounded in the class of graphs with treewidth 2. Moreover, $\zeta(G) \leq 3$ for any outerplanar graph G [2].

Relative distance and centroidal dimension. Foucaud *et al.* defined a variant of resolving sets, called *centroidal basis*, where the vertices of a graph must be distinguished by their *relative* distance to the probed vertices [7]. In this setting, given an integer $k \geq 2$, probing a set $B = \{v_1, \cdots, v_k\}$ of vertices results in the vector $(\delta_{i,j}(t))_{1 \leq i < j \leq k}$ where, for every $1 \leq i < j \leq k$, $\delta_{i,j}(t) = 0$ if $d_G(t, v_i) = d_G(t, v_j)$, $\delta_{i,j}(t) = 1$ if $d_G(t, v_i) > d_G(t, v_j)$ and $\delta_{i,j}(t) = -1$ otherwise. The set B is a *centroidal basis* if the vectors of relative distances for every $t \in V$ are pairwise distinct. The *centroidal dimension* of a graph G, denoted by $CD(G) \geq 2$, is the minimum size of a centroidal basis of G [7] (this is well defined since, clearly, V is a centroidal basis of G). The decision problem associated to the centroidal dimension is NP-complete and almost tight bounds on the centroidal dimension of paths have been computed [7].

A sequential variant of the centroidal basis can naturally be defined. This variant has been studied in the case of a moving target in [2].

Here, we also initiate the study of this variant when the target is immobile. Let $k \geq 2$ be an integer and G be a graph. Let $\lambda_k^{rel}(G)$ denote the minimum integer h such that there exists a k-strategy that locates (using relative distances) a hidden immobile target in G in at most h steps, whatever be the location of the target. Given G, k, and ℓ, the Relative-Localization Problem asks whether $\lambda_k^{rel}(G) \leq \ell$. The dual parameter $\kappa_\ell^{rel}(G)$ is defined as the minimum integer h such that there exists an h-strategy (with relative distances) that locates the target in G in at most ℓ steps. Note that, for every graph G, $\kappa_1^{rel}(G)$ is exactly the centroidal dimension $CD(G)$ of G, and $\lambda_k^{rel}(G) \leq \ell$ if and only if $\kappa_\ell^{rel}(G) \leq k$.

1.2 Our results

In the whole paper, G denotes a connected undirected simple graph. We consider the computational complexity of the LOCALIZATION Problem. In Section 2, we show that it is NP-complete when k or ℓ are fixed parameters. Precisely:

- Let $k \geq 1$ be a fixed integer. Given a graph G with a universal vertex and an integer $\ell \geq 1$ as inputs, the problem of deciding whether $\lambda_k(G) \leq \ell$ is NP-complete (Theorem 1).
- Let $\ell \geq 1$ be a fixed integer. Given a graph G with a universal vertex and an integer $k \geq 1$ as inputs, the problem of deciding whether $\kappa_\ell(G) \leq k$ is NP-complete (Theorem 3).

On the way, we also show that the RELATIVE-LOCALIZATION Problem is NP-complete when k or ℓ are fixed parameters (Theorems 2 and 4).

In Section 3, we then focus on the LOCALIZATION Problem in the class of trees. Surprisingly, in trees, the complexity of the LOCALIZATION Problem only comes from the first step. We show that, after the first step, the problem becomes polynomial-time solvable. This allows us to design a polynomial-time approximation algorithm for the problem. More precisely, we show that

- deciding whether $\lambda_k(T) \leq \ell$ is NP-complete in the class of trees T when both k and ℓ are part of the input (Theorem 5);
- there exists an algorithm that computes, in time $O(n \log n)$ (independent of k), a k-strategy for locating a target in at most $\lambda_k(T)+1$ steps in any n-node tree (possibly edge-weighted)(Theorem 8);
- deciding whether $\lambda_k(T) \leq \ell$ can be solved in time $O(n^{k+2} \log n)$ (independent of ℓ) in the class of n-node trees (possibly edge-weighted) (Theorem 9).

2 Complexity of the LOCALIZATION Problem

This section is devoted to prove that the LOCALIZATION Problem is NP-complete when k or ℓ is fixed. The proof when ℓ is fixed is an almost straightforward reduction from the METRIC DIMENSION Problem. In the case when k is fixed, it is a much more involved reduction from the 3-DIMENSIONAL MATCHING Problem.

2.1 When the number k of probed vertices per step is fixed

Let $k \geq 1$ be a fixed integer. The k-PROBE LOCALIZATION Problem takes a graph G and an integer $\ell \geq 1$ as inputs and asks whether $\lambda_k(G) \leq \ell$.

Theorem 1. *Let $k \geq 1$ be a fixed integer. The k-PROBE LOCALIZATION Problem is NP-complete in the class of graphs with a universal vertex.*

Sketch of proof. Since any k-strategy in a graph G has length at most $|V(G)|$, the problem is in NP. Let us prove it is NP-hard by a reduction from the 3-DIMENSIONAL MATCHING Problem (3DMP) which is a well known NP-hard problem. The 3DMP takes a set $\mathcal{X} = I_1 \cup I_2 \cup I_3$ of $3n$ elements ($|I_1| = |I_2| = |I_3| = n$)

and a set $\mathcal{S}$ of triples $(x, y, z) \in I_1 \times I_2 \times I_3$ as inputs and asks whether there are n triples of $\mathcal{S}$ that are pairwise disjoint.

Let $k \geq 1$ be a fixed integer and let $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ be an instance of 3DMP. First, we may assume that $|\mathcal{X}| = 3kn$ since, if not, it is sufficient to take k disjoint copies of $(\mathcal{X}, \mathcal{S})$. Moreover, we may assume that $m = |\mathcal{S}|$ is such that $2m-1 \equiv 0 \mod k$ (for instance by adding dummy triples if needed). Let $\mathcal{X} = \{x_1, \cdots, x_{3kn}\}$ and $\mathcal{S} = \{S_1, \cdots, S_m\}$.

Let us build the graph $G = (V, E)$ as follows. Let the vertex-set $V = X \cup X'' \cup S \cup \{s\} \cup \{q\}$ be such that $X = X^1 \cup \ldots \cup X^{k+2}$ with $X^i = \{x^i_1, \cdots, x^i_{3kn}\}$ for every $1 \leq i \leq k+2$; $X'' = \{x''_1, \ldots, x''_{(k+2)m}\}$; and $S = S^1 \cup \ldots \cup S^{k+2}$ with $S^i = \{s^i_j, 1 \leq j \leq m\}$ for every $i \in [\![1, k+2]\!]$. The vertex s is universal (*i.e.*, adjacent to every other vertex), the vertex q is adjacent to every vertex in $X \cup X''$ and, for every $j \in [\![1, 3kn]\!]$ and every $g \in [\![1, m]\!]$ such that $x_j \in S_g$, there is an edge between x^i_j and s^i_g for every $i \in [\![1, k+2]\!]$. Intuitively, X^i is a "copy" of $\mathcal{X}$ and S^i is a "copy" of $\mathcal{S}$ for every $1 \leq i \leq k+2$.

Let $p = \frac{m(k+2)-1}{k} \in \mathbb{N}$. We prove the theorem by showing that $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ admits a 3DM if and only if $\lambda_k(G) \leq (k+2)n + p + 1$. Due to lack of space, the proof is postponed to Appendix 5.1. □

The same proof also works for the case with relative distances. Hence,

Theorem 2. *Let $k \geq 2$ be a fixed integer. Given a graph G with a universal vertex and $1 \leq \ell \in \mathbb{N}$, the problem of deciding if $\lambda^{rel}_k(G) \leq \ell$ is NP-complete.*

2.2 When the number ℓ of steps is fixed

Let $\ell \geq 1$ be a fixed integer. The ℓ-Step Localization Problem takes a graph G and an integer $k \geq 1$ as inputs and asks whether $\kappa_\ell(G) \leq k$.

Theorem 3. *Let $\ell \geq 1$ be a fixed parameter. The ℓ-Step Localization Problem is NP-complete in the class of graphs with a universal vertex.*

Sketch of proof. For $\ell = 1$, the result follows from the fact that $\kappa_1(G)$ is exactly the metric dimension and from its NP-completeness [6].

Let $\ell \geq 2$ be fixed. We focus on proving the NP-hardness of the ℓ-Step Localization Problem, as it is clearly in NP. The proof is by reduction from the Metric Dimension Problem restricted to the class of graphs with diameter 2, which is known to be NP-hard [9]. Let thus $< G, k >$ be an instance of Metric Dimension where G has diameter 2. We construct, in polynomial time, an instance $< G', k >$ of the ℓ-Step Localization Problem such that $MD(G) \leq k$ if and only if a target hidden in G' can be located in at most ℓ steps by probing at most k vertices per step, *i.e.*, $\kappa_\ell(G) \leq k$.

The construction of G' is as follows. Start from $k(\ell-1)+1$ disjoint copies $G_1, ..., G_{k(\ell-1)+1}$ of G. Add the vertices $v_1, ..., v_{k(\ell-1)+1}$ and add all the edges from v_i to every vertex of G_i for all integers $1 \leq i \leq k(\ell-1)+1$. Add a vertex u to the graph, and all edges between u and each vertex of the copies of G. Due to lack of space, the proof is postponed to Appendix 5.2. □

A similar proof (but based on a reduction of Centroidal Dimension) works for the case with relative distances. Hence,

Theorem 4. *Let $\ell \geq 1$ be a fixed integer. Given a graph G with a universal vertex and $2 \leq k \in \mathbb{N}$, the problem of deciding if $\kappa_\ell^{rel}(G) \leq k$ is NP-complete.*

3 The Localization Problem in trees

This section is devoted to the study of the Localization Problem in the class of trees. Note that, if the number of steps is $\ell = 1$, the problem is equivalent to the one of Metric dimension which can trivially be solved in polynomial-time in trees [10, 17]. We first show that, if k and ℓ are part of the input, deciding whether $\lambda_k(T) \leq \ell$ is NP-complete in the class of trees T. Our reduction actually shows that the difficulty of the problem comes from the choice of the vertices to be probed during the first step. Surprisingly, we show that the first step is actually the only source of complexity. More precisely, our main result is that, if the first step is given (intuitively, either given by an oracle or imposed by an adversary), then an optimal strategy (according to this first pre-defined step) can be computed in polynomial-time. This allows us to design a (+1)-approximation algorithm for the Localization Problem in trees and to prove that, in contrast with general graphs (Theorem 1), the k-Probe Localization Problem is polynomial-time solvable in the class of trees.

3.1 NP-hardness

Theorem 5. *The* Localization *Problem is NP-complete in the class of trees.*

Sketch of proof. Again, the problem is in NP. To prove the NP-hardness, let us reduce the Hitting-Set Problem. The inputs are an integer $k \geq 1$, a ground-set $B = (b_1, \cdots, b_n)$ and a set $\mathcal{S} = \{S_1, \cdots, S_m\}$ of subsets of B, *i.e.*, $S_i \subseteq B$ for every $i \leq m$. The Hitting-Set Problem aims at deciding if there exists a set $H \subseteq B$, $|H| \leq k$ and $H \cap S_i \neq \emptyset$ for every $1 \leq i \leq m$.

Adding one new element to the ground-set and adding this element to one single subset clearly does not change the solution. Therefore, by adding some dummy elements (each one belonging to a single subset), we may assume that all the subsets are of the same size σ and that $\sigma - 1 \equiv 0 \mod k$.

Let γ be any integer such that $\gamma - 1 \equiv 0 \mod k$ and $\gamma > n - k - 1$.

The instance T of the Localization Problem is built as follows. Let us start with n vertex-disjoint paths $B_1, \cdots, B_n$ (the *branches*) of length $2m$, where $B_i = (b_1^i, \cdots, b_{2m+1}^i)$ for each $1 \leq i \leq n$. Then, let us add one new vertex c adjacent to b_1^i for all $1 \leq i \leq n$. For every $1 \leq j \leq m$ and for every $1 \leq i \leq n$ such that $b_i \in S_j$, let us add γ new vertices adjacent to b_{2j}^i. The subgraph induced by b_{2j}^i and by the γ leaves adjacent to it is referred to as the *star* representing the element i in the set S_j (or representing the set S_j in the branch i). The obtained tree T is depicted in Figure 1.

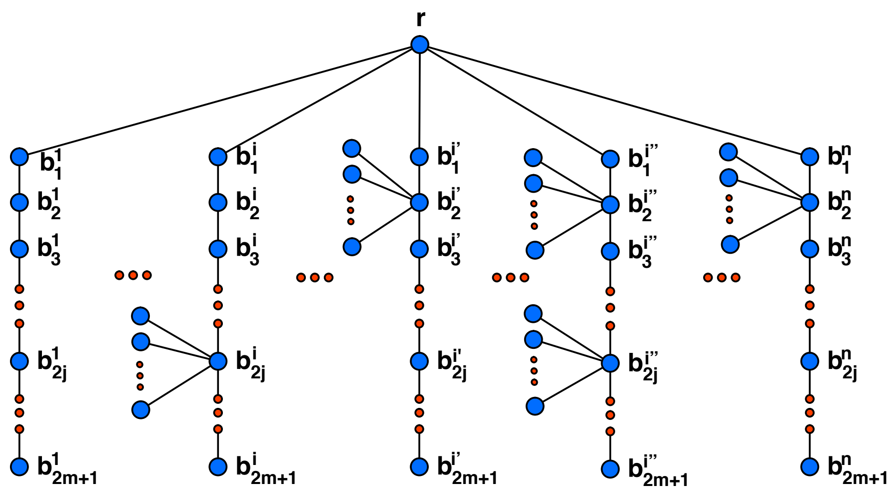

Fig. 1: An example of a tree T built from an instance $(k, B, \mathcal{S})$ of HITTING SET in the proof of Theorem 5. In this example, the elements $b_{i'}, b_{i''}$, and b_n belong to the set S_1 (but not the elements b_1 and b_i) as figured by the three "stars" at level 2. The elements b_i and $b_{i''}$ belong to S_j (stars at level $2j$), but not the elements $b_1, b_{i'}$, and b_n.

Intuitively, it will always be better for the target to be located in a leaf of some star because γ is "huge". During the first turn of any strategy, the *level* (roughly, the distance to the root) of the target can be identified. Each even level $2j$ corresponds to a set S_j. If, during the first turn, one star corresponding to each even level can be eliminated from the possible locations (which corresponds to hit every subset), then the strategy finishes one step earlier than if all subsets cannot be hit (if so, all stars would have to be checked).

More formally, we show that $\lambda_k(T) \leq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$ if and only if there is a hitting set of size at most k. Due to lack of space, the proof is postponed to Appendix 5.3. $\square$

3.2 Algorithm in trees

The proof above actually shows that, in our reduction, choosing the vertices to be probed during the first step to ensure an optimal strategy is equivalent to finding a minimum hitting set. We show below that the first step is actually the only "part" of the problem that is difficult.

The key argument is the following easy remark. Let us consider a tree T where a target is hidden and assume that a single vertex $r \in V(T)$ is probed. After this single probe, the distance $d \in \mathbb{N}$ between the target and r is known.

Therefore, from the second step, the instance becomes equivalent to a tree T' (a subtree of T) rooted in r, with all leaves the same distance d from r, and where the target is known to occupy some leaf of T'. We first present an algorithm that computes in polynomial-time (independent of k and ℓ) an optimal strategy to locate the target in such instances.

Let $\mathcal{T}$ be the set of rooted trees with all leaves the same distance from the root. Given a rooted tree $(T, r) \in \mathcal{T}$ (in what follows, we omit r when it is clear from the context), let $\lambda_k^L(T)$ be the minimum integer h such that there exists a k-strategy that locates a target in at most h steps knowing *a priori* that the target occupies some leaf of T. The next claim is one of the key arguments that makes the problem easier in the class $\mathcal{T}$ when the target is known to occupy a leaf.

Claim **1** *Let $(T, r) \in \mathcal{T}$ rooted in r, let v be a child of r and T_v the subtree rooted in v. If the target is known to occupy a leaf of T, then probing any vertex in T_v allows to learn if the target occupies a leaf of T_v or a leaf of $T \setminus T_v$.*

Proof of claim. Let d be the distance between r and the leaves of T. Let w be any vertex of T_v and let d' be the distance between w and r. The target occupies a leaf of T_v if and only if its distance to w is $< d + d'$. $\diamond$

Let $T \in \mathcal{T}$ rooted in r, let v be a child of r and let us assume that the secret location of the target is some leaf of T_v. Note that $(T_v, v) \in \mathcal{T}$. Let us assume that T_v is not a path and let s be the first step of ϕ that probes some vertex of T_v (such a step s must exist since otherwise the target would never be detected in T_v). By Claim 1, it is sufficient to probe a single vertex of T_v to learn that the target occupies a leaf of T_v. Then, applying an optimal strategy ϕ_v in T_v will locate the target in a total of at most $s + \lambda_k^L(T_v)$ steps. But it may be possible to do better. Indeed, probing several vertices of T_v during the s^{th} step of ϕ may serve not only to detect the target in T_v but also to "play" the first step of ϕ_v. Doing so, the strategy will take only $s + \lambda_k^L(T_v) - 1$ steps. So, elaborating, an optimal strategy will consist of doing a tradeoff between probing one single vertex in a subtree (and detect "quickly" in which subtree the target is hidden since several subtrees are considered simultaneously) and probing more vertices in a subtree in order to get a head start for the strategy in the case the target is in this subtree.

For any tree T, let $\pi(T)$ be the minimum integer q such that there exists a k-strategy that locates a target in at most $\lambda_k^L(T)$ steps, knowing *a priori* that the target occupies some leaf of T, and such that at most q vertices are probed during the first step.

To illustrate the need of a tradeoff, let us consider the following simple example utilizing π. Consider two children v_1 and v_2 of r such that $(\lambda_k^L(T_{v_1}), \pi(T_{v_1})) = (6, 4)$ and $(\lambda_k^L(T_{v_2}), \pi(T_{v_2})) = (6, 3)$. Let $k = 6$. Then, at the first step, we cannot probe $\pi(T_{v_1}) + \pi(T_{v_2}) = 7$ vertices. W.l.o.g., let us assume that at most $3 < \pi(T_{v_1})$ vertices of T_{v_1} have been probed during the first step. Thus, by definition of π, a total of $\lambda_k^L(T_{v_1}) + 1 = 7$ steps are necessary if we learn at the first

Algorithm 1 $\mathcal{A}_1(k,(T,r))$.

Require: An integer k and a tree $T \in \mathcal{T}$ rooted in r with children $v_1, \cdots, v_{d^*}$
Ensure: $(\lambda_k^L(T), \pi(T))$

if (T,r) is a rooted path **then**
  return $(0,0)$
for $i = 1$ to d^* **do**
  Let $(\lambda_i, \pi_i) = \mathcal{A}_1(k,(T[i], v_i)$
Sort the $(\lambda_i, \pi_i)_{1 \le i \le d^*}$ in non-increasing lexicographical order
return $\mathcal{A}_2(k,(T,r),(\lambda_i,\pi_i)_{1\le i\le d^*})$

step that the target occupies some leaf of T_{v_1}. For a more detailed example of a strategy in a tree, see the example in Appendix 5.4.

Let $T \in \mathcal{T}$ rooted in r and let $v_1, \ldots, v_{d^*}$ be the children of r. From previous arguments, the computation of an optimal strategy for T consists of determining, for each subtree T_{v_i} $(1 \le i \le d^*)$, the first step for which a vertex of T_{v_i} will be probed (if the target has not been located in a different subtree at a previous step). If 1 vertex is probed during this step, then $\lambda_k^L(T_{v_i})$ extra steps are needed if the target occupies some leaf of T_{v_i}. Furthermore, if we want to locate the target in at most $\lambda_k^L(T_{v_i}) - 1$ extra steps (if the target occupies some leaf of T_{v_i}), $\pi(T_{v_i})$ vertices of T_{v_i} must be probed during this step.

Description of Algorithm 1. The main algorithm $\mathcal{A}_1(k,(T,r))$ takes an integer $k \ge 1$ and a rooted tree $(T,r) \in \mathcal{T}$ as inputs and computes $(\lambda_k^L(T), \pi(T))$ and a corresponding k-strategy. It proceeds bottom-up by dynamic programming from the leaves to the root. Precisely, let $v_1, \ldots, v_{d^*}$ be the children of r. For any $1 \le i \le j \le d^*$, let $T[i] = T_{v_i}$ be the subtree rooted at v_i, and let $T[i,j] = \{r\} \cup T_{v_i} \cup \ldots \cup T_{v_j}$ $(T[i,j] = \emptyset$ if $i > j)$. To lighten the notations, let us set $\lambda_i = \lambda_k^L(T[i])$ and $\pi_i = \pi(T[i])$ for every $1 \le i \le d^*$. Assume that, $(\Lambda, \Pi) = (\lambda_i, \pi_i)_{1 \le i \le d^*}$ have been computed recursively and sorted in non-increasing lexicographical order. Then, $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$, described in Algorithm 2, takes the integer $k \ge 1$, the rooted tree $(T,r) \in \mathcal{T}$, and the sorted tuple (Λ, Π) as inputs and computes $(\lambda_k^L(T), \pi(T))$ and a corresponding strategy.

Description of Algorithm 2. We now informally describe $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$. First, Line 2 to Line 5 deals with the subtrees $T_{v_{d+1}}, \cdots, T_{v_d^*}$ that are rooted paths (path rooted at one of its vertices of degree one, the other vertex is the leaf). In other words, it concerns all the subtrees T_{v_i} such that $(\lambda_i, \pi_i) = (0,0)$. Indeed, this case is somehow pathologic. Claim 2 proves that Line 2 to Line 5 computes $(\lambda_k^L(T[v_{d+1}, d^*]), \pi(T[v_{d+1}, d^*]))$. Let us define $\mathcal{S} \subset \mathcal{T}$ as the set of subdivided stars S (*i.e.*, trees with at most one vertex of degree at least 3) with all leaves the same distance from the root, where the root of S is the (unique) vertex with degree > 2 or one of the two ends if S is a path. Due to lack of space, the proof of Claim 2 is postponed to Appendix 5.4.

Claim **2** *Let* $S \in \mathcal{S}$ *with* $\delta = |V(S)| - 1$ *(i.e.,* δ *is the degree of the root* r*). Then,* $\lambda_k^L(S) = \lceil \frac{\delta-1}{k} \rceil$ *and* $\pi(S) = -k(\lceil \frac{\delta-1}{k} \rceil - \lceil \frac{\delta-1}{\delta} \rceil) + (\delta - 1)$.

Algorithm 2 $\mathcal{A}_2(k, (T, r), (\Lambda, \Pi))$.

Require: $k \in \mathbb{N}^*$, a rooted tree (T, r) with $v_1, \cdots, v_{d^*}$ the children of r such that $(\Lambda, \Pi) = (\lambda_i, \pi_i)_{1 \leq i \leq d^*}$ is sorted in non-increasing lexicographical order.

1: $l \leftarrow 1$, $p \leftarrow k$, $d \leftarrow d^*$
2: **if** $T[d^*]$ is a rooted path **then**
3: $\quad d \leftarrow z$ with $0 \leq z < d^*$ the smallest integer such that $T[z+1]$ is a rooted path
4: $\quad l \leftarrow 1 + \lceil \frac{d^*-d-1}{k} \rceil$
5: $\quad p \leftarrow k + k(\lceil \frac{d^*-d-1}{k} \rceil - \lceil \frac{d^*-d-1}{d^*-d} \rceil) - (d^* - d - 1)$
6: **for** $i = d$ down to 1 **do**
7: $\quad$ **if** $p = 0$ or $l < \lambda_i + 1$ **then**
8: $\quad\quad p \leftarrow k$, $l \leftarrow \max(l+1, \lambda_i + 1)$
9: $\quad \alpha \leftarrow \pi_i - (\pi_i - 1)\lceil (l - (\lambda_i + 1))/l \rceil$
10: $\quad$ **if** $\alpha \leq p$ **then**
11: $\quad\quad p \leftarrow p - \alpha$
12: $\quad$ **else**
13: $\quad\quad p \leftarrow k - 1$, $l \leftarrow l + 1$
14: **return** $(l - 1, k - p)$

We are now able to detail the second part of the algorithm (from Line 6). Informally, $\mathcal{A}_2(k, (T, r), (\Lambda, \Pi))$ recursively builds, for $i = d$ down to 1, an optimal k-strategy ϕ for $T[i, d^*]$ from an optimal k-strategy ϕ' of $T[i+1, d^*]$ and from an optimal k-strategy ϕ'' of $T[i]$ (the latter one being given as input through (λ_i, π_i)). In other words, $(\lambda_k^L(T[i, d^*]), \pi(T[i, d^*]))$ is computed from $(\lambda_k^L(T[i+1, d^*]), \pi(T[i+1, d^*]))$ and (λ_i, π_i). For every $1 \leq i \leq d+1$, let l_i (resp., p_i) denote the value of l (resp. of p) just before the $(d+2-i)^{th}$ iteration of the for-loop (so, l_1 and p_1 are the final values of l and p). Intuitively, let us assume that an optimal strategy for $T[i+1, d^*]$ has been computed, takes at most $l_{i+1} - 1$ steps and requires $k - p_{i+1} = \pi(T[i+1, d^*])$ vertices to be probed during its first step. Roughly, there are five cases to be considered.

- If $\pi_i \leq p_{i+1}$ and $\lambda_i = l_{i+1} - 1$, the strategy ϕ follows ϕ' but, in addition, probes π_i vertices of $T[i]$ during its first step. If the target is in $T[i]$, then ϕ follows ϕ'' (and takes a total of at most λ_i steps), otherwise, it proceeds as ϕ' (and takes a total of at most $l_{i+1} - 1$ steps). We get $l_i = l_{i+1}$ and $p_i = p_{i+1} - \pi_i$.
- Else if $\pi_i > p_{i+1} > 0$ and $\lambda_i = l_{i+1} - 1$, the first step of ϕ probes a unique vertex in $T[i]$. If the target is in $T[i]$, then ϕ follows ϕ'' (and takes a total of at most $\lambda_i + 1$ steps). Otherwise, it proceeds as ϕ' (and takes a total of at most l_{i+1} steps). We get $l_i = l_{i+1} + 1$ and $p_i = k - 1$.
- Else, if $p_{i+1} = 0$ and $\lambda_i \leq l_{i+1} - 1$, the first step of ϕ probes a unique vertex in $T[i]$. If the target is in $T[i]$, then ϕ follows ϕ'' (and takes a total of at most $\lambda_i + 1$ steps). Otherwise, it proceeds as ϕ' (and takes a total of at most l_{i+1} steps). We get $l_i = l_{i+1} + 1$ and $p_i = k - 1$.
- Else, if $\lambda_i < l_{i+1} - 1$ and $p_{i+1} > 0$, the strategy ϕ follows ϕ' but, in addition, probes one vertex of $T[i]$ during its first step. If the target is in $T[i]$, then ϕ follows ϕ'' (and takes a total of at most $\lambda_i + 1$ steps), otherwise, it proceeds

as ϕ' (and takes a total of at most $l_{i+1}-1$ steps). We get $l_i = l_{i+1}$ and $p_i = p_{i+1}-1$.
- Else ($\lambda_i > l_{i+1}-1$), the strategy ϕ probes π_i vertices in $T[i]$ during the first step. If the target is in $T[i]$, then ϕ follows ϕ'' (and takes a total of at most λ_i steps), otherwise, it proceeds as ϕ' (and takes a total of at most l_{i+1} steps). We get $l_i = \lambda_i + 1$ and $p_i = k - \pi_i$.

As the subtrees are sorted in non-increasing lexicographical order (of (λ_i, π_i)), we prove in Lemma 1 that the strategy ϕ described before is optimal for $T[i, d^*]$, that is, it computes $(\lambda_k^L(T[i,d^*]), \pi(T[i,d^*]))$. Due to lack of space, the proof of this main technical lemma is postponed to Appendix 5.4.

Lemma 1. *For every* $1 \leq i \leq d+1$, $\lambda_k^L(T[i,d^*]) = l_i - 1$ *and* $\pi(T[i,d^*]) = k - p_i$.

Correctness and complexity of Algorithm 1 and Algorithm 2. We prove in Theorem 7 that $\mathcal{A}_1(k,(T,r))$ computes $(\lambda_k^L(T), \pi(T))$ and a corresponding k-strategy in time $O(n \log n)$, where n is the number of vertices. To do that, Theorem 6 proves the correctness and the linear (in the number of children of r) time complexity of $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$.

Theorem 6. *Let* $k \geq 1$, *let* $(T,r) \in \mathcal{T}$ *be a rooted tree, and let* $v_1, \cdots, v_{d^*}$ *be the children of* r *such that the tuples* $(\Lambda, \Pi) = (\lambda_i, \pi_i)_{1 \leq i \leq d^*}$ *are sorted in non-increasing lexicographical ordering. Then,* $\mathcal{A}_2(k,(T,r),(\Lambda,\Pi))$ *returns* $(\lambda_k^L(T), \pi(T))$ *and a corresponding strategy. Furthermore, the time-complexity of* $\mathcal{A}_2$ *is* $O(d^*)$ *(independent of* k*).*

Proof. The time-complexity is obvious and the correctness follows from Lemma 1 for $i = 1$. The fact that the strategy is also returned is not explicitly described in Algorithm 2 but directly follows from the proof of Lemma 1. □

Theorem 7. *Let* $k \geq 1$, *and let* $(T,r) \in \mathcal{T}$ *be an* n*-node rooted tree. Then,* $\mathcal{A}_1(k,(T,r))$ *returns* $(\lambda_k^L(T), \pi(T))$ *and a corresponding strategy. Furthermore, the time-complexity of* $\mathcal{A}_1$ *is* $O(n \log n)$ *(independent of* k*).*

Proof. The correctness is simply proved by induction and by Theorem 6. For the time-complexity, at every recursive call on a subtree T_v rooted at v (with d_v children), the additional number of operations is $O(d_v \log d_v)$ (sorting) plus $O(d_v)$ (Algorithm $\mathcal{A}_2$, by Theorem 6). Since in a tree, $\sum_{v \in V(T)} d_v = 2(n-1)$, this gives a total complexity of $O(\sum_{v \in V(T)} d_v \log d_v) = O(n \log n)$. Again, the strategy is not explicit in our presentation but can be easily computed. □

Main results. From $\mathcal{A}_1(k,(T,r))$ presented before, it is easy to get an efficient approximation algorithm when k and ℓ are part of the input and a polynomial-time algorithm when k is fixed.

Theorem 8. *There exists an algorithm that, given any integer* $k \geq 1$ *and any* n*-node tree* T, *computes a* k*-strategy that locates a target in* T *in at most* $\lambda_k(T)+1$ *steps. Furthermore, the time-complexity of the algorithm is* $O(n \log n)$, *Proof.* The strategy proceeds as follows. The first step probes any arbitrary vertex r of T. Let d be the distance between r and the target, let $L \subseteq V(T)$ be the set of vertices at distance exactly d from r, and let T^d be the subtree induced by r and every vertex on a path between r and the vertices in L. Note that $(T^d, r) \in \mathcal{T}$ and that the target is occupying a leaf of T^d. Hence, it is sufficient to apply $\mathcal{A}_1(k, (T^d, r))$.

By Theorem 7, the above strategy will locate the target in at most $1 + \max_d \lambda_k^L(T^d) \leq 1 + \lambda_k(T)$ steps (by Claim 10). □

Theorem 9. *There exists an algorithm that, given any integer $k \geq 1$ and any n-node tree T, computes an optimal k-strategy for locating a target in T in at most $\lambda_k(T)$ steps. Furthermore, the time-complexity of the algorithm is $O(n^{k+2} \log n)$.*

Proof. The proof is similar to the one of the previous theorem, but instead of probing a single vertex during the first step, we enumerate all the $O(n^k)$ possibilities for the first step and enumerate each answer among the n possible. □

To conclude this section, it is important to mention that both Theorems 8 and 9 also hold in the case of edge-weighted trees. Indeed, distances are only used in Claim 1 which clearly holds for edge-weighted trees.

4 Further Work

Our results in trees leave the open question of whether $\lambda_k(T)$ is Fixed Parameter Tractable (in k) in the class of n-node trees T. Moreover, it would be interesting to study the LOCALIZATION Problem in other graph classes such as interval graphs and planar graphs. Also, what is the complexity of the ℓ-STEP LOCALIZATION Problem in trees?

The RELATIVE-LOCALIZATION Problem seems to be much more intricate even for simple topologies. A first step towards a better understanding of this problem would be to fully solve it in the case of paths (*i.e.*, to determine $\kappa_1^{rel}(P)$ for every path P), which has been partially solved in [7], before studying it in the class of trees.

References

1. R. Belmonte, F. V. Fomin, P. A. Golovach, and M. S. Ramanujan. Metric dimension of bounded tree-length graphs. *SIAM J. Discrete Math.*, 31(2):1217–1243, 2017.
2. B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokól, and M. Sleszynska-Nowak. Centroidal localization game. *CoRR*, abs/1711.08836, 2017.
3. B. Bosek, P. Gordinowicz, J. Grytczuk, N. Nisse, J. Sokól, and M. Sleszynska-Nowak. Localization game on geometric and planar graphs. *CoRR*, abs/1709.05904, 2017.
4. A. Brandt, J. Diemunsch, C. Erbes, J. LeGrand, and C. Moffatt. A robber locating strategy for trees. *Discrete Applied Mathematics*, 232:99 – 106, 2017.
5. J. M. Carraher, I. Choi, M. Delcourt, L. H. Erickson, and D. B. West. Locating a robber on a graph via distance queries. *Theor. Comput. Sci.*, 463:54–61, 2012.
6. J. Díaz, O. Pottonen, M. J. Serna, and E. J. van Leeuwen. Complexity of metric dimension on planar graphs. *J. Comput. Syst. Sci.*, 83(1):132–158, 2017.
7. F. Foucaud, R. Klasing, and P. J. Slater. Centroidal bases in graphs. *Networks*, 64(2):96–108, 2014.
8. F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. i. bounds. *Theor. Comput. Sci.*, 668:43–58, 2017.
9. F. Foucaud, G. B. Mertzios, R. Naserasr, A. Parreau, and P. Valicov. Identification, location-domination and metric dimension on interval and permutation graphs. II. algorithms and complexity. *Algorithmica*, 78(3):914–944, 2017.
10. F. Harary and R. A. Melter. On the metric dimension of a graph. *Ars Combinatoria*, 2:191–195, 1976.
11. S. Hartung and A. Nichterlein. On the parameterized and approximation hardness of metric dimension. In *Proceedings of the 28th Conference on Computational Complexity, CCC*, pages 266–276. IEEE Computer Society, 2013.
12. J. Haslegrave, R. A. B. Johnson, and S. Koch. Locating a robber with multiple probes. *Discrete Mathematics*, 341(1):184–193, 2018.
13. M. G. Karpovsky, K. Chakrabarty, and L. B. Levitin. On a new class of codes for identifying vertices in graphs. *IEEE Trans. Information Theory*, 44(2):599–611, 1998.
14. S. M. Seager. Locating a robber on a graph. *Discrete Mathematics*, 312(22):3265–3269, 2012.
15. S. M. Seager. A sequential locating game on graphs. *Ars Comb.*, 110:45–54, 2013.
16. S. M. Seager. Locating a backtracking robber on a tree. *Theor. Comput. Sci.*, 539:28–37, 2014.
17. P. J. Slater. Leaves of trees. pages 549–559. Congressus Numerantium, No. XIV, 1975.
18. P. J. Slater. Domination and location in acyclic graphs. *Networks*, 17(1):55–64, 1987.

5 Appendix

Notation. In the whole paper, $G = (V, E)$ denotes a connected undirected simple graph. For any $v, u \in V$, $N_G(v)$ denotes the set of neighbours of v and $d_G(u, v)$ denotes the distance between u and v in G. A *k-strategy* is a localization strategy that probes at most k vertices per step. After the s^{th} step of a localization strategy, $H_s \subseteq V$ is the set of vertices that may still host the target. Unless stated otherwise, $H_0 = V$.

5.1 Proofs of Theorem 1 and Theorem 2

Theorem 1. *Let $k \geq 1$ be a fixed integer. The k-*PROBE LOCALIZATION *Problem is NP-complete in the class of graphs with a universal vertex.*

Proof. Since any k-strategy in a graph G has length at most $|V(G)|$, the problem is in NP. Let us prove it is NP-hard by a reduction from the 3-DIMENSIONAL MATCHING Problem (3DMP) which is a well known NP-hard problem. The 3DMP takes a set $\mathcal{X} = I_1 \cup I_2 \cup I_3$ of $3n$ elements ($|I_1| = |I_2| = |I_3| = n$) and a set $\mathcal{S}$ of triples $(x, y, z) \in I_1 \times I_2 \times I_3$ as inputs and asks whether there are n triples of $\mathcal{S}$ that are pairwise disjoint.

Let $k \geq 1$ be a fixed integer and let $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ be an instance of 3DMP. First, we may assume that $|\mathcal{X}| = 3kn$ since, if not, it is sufficient to take k disjoint copies of $(\mathcal{X}, \mathcal{S})$. Moreover, we may assume that $m = |\mathcal{S}|$ is such that $2m-1 \equiv 0 \mod k$ (for instance by adding dummy triples if needed). Let $\mathcal{X} = \{x_1, \cdots, x_{3kn}\}$ and $\mathcal{S} = \{S_1, \cdots, S_m\}$.

Let us build the graph $G = (V, E)$ as follows.

- $X = X^1 \cup \ldots \cup X^{k+2}$ with $X^i = \{x^i_1, \cdots, x^i_{3kn}\}$ for every $i \leq k+2$. Each of the vertices x^i_j, for $i \in [\![1, k+2]\!]$, represents the element x_j, for $j \leq 3kn$;
- $X'' = \{x''_1, \ldots, x''_{(k+2)m}\}$;
- $S = S^1 \cup \ldots \cup S^{k+2}$ with $S^i = \{s^i_j, 1 \leq j \leq m\}$ for every $i \in [\![1, k+2]\!]$. Each of the vertices s^i_j, for $i \in [\![1, k+2]\!]$, represents the element S_j, for $j \leq 3kn$;

The set of edges is defined as follows.

- There is an edge between s and every vertex of $V \setminus \{s\}$.
- There is an edge between q and every vertex of $X \cup X''$.
- For every $j \in [\![1, 3kn]\!]$ and every $g \in [\![1, m]\!]$ such that $x_j \in S_g$, there is an edge between x^i_j and s^i_g for every $i \in [\![1, k+2]\!]$.

The graph G is depicted in Figure 2.

Let $p = \frac{m(k+2)-1}{k} \in \mathbb{N}$. We prove the theorem by showing that $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ admits a 3DM if and only if $\lambda_k(G) \leq (k+2)n + p + 1$.

Claim **3** *If $\mathcal{I}$ admits a 3DM, then $\lambda_k(G) \leq (k+2)n + p + 1$.*

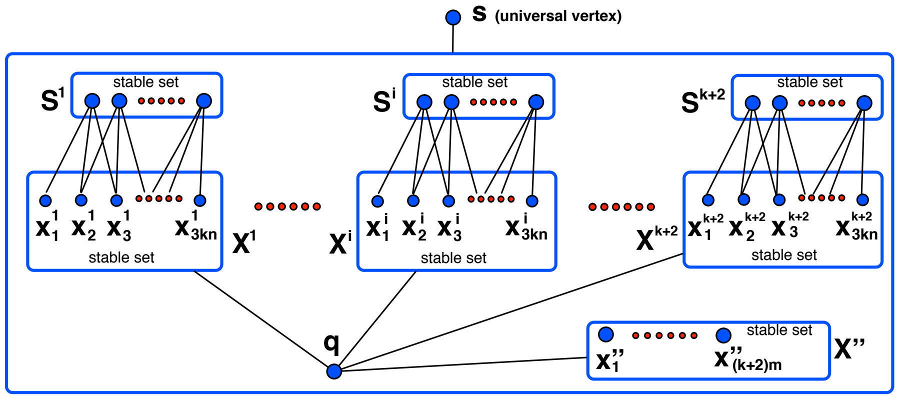

Fig. 2: Graph G constructed from an instance of 3DMP for proof of Theorem 1. A thin line between one vertex (blue circle) and one rectangle represents all edges between this vertex and every vertex in the rectangle. The instance of the 3DMP is encoded in the edges between the vertices in S^i (representing the sets) and the vertices in X^i, for every $i \leq k+1$.

Proof of claim. Let $Y \subseteq \mathcal{S}$ be a 3DM of $\mathcal{I} = (\mathcal{X}, \mathcal{S})$ (of size $|Y| = kn$). Up to renumbering the sets and the elements, let us assume that $Y = \{S_1, S_2, \dots, S_{kn}\}$ and assume that $S_i = \{x_{3(i-1)+1}, x_{3(i-1)+2}, x_{3(i-1)+3}\}$ for every $i \in [\![1, kn]\!]$. Note that, because Y is a 3DM of size kn, $\bigcup_{1 \leq i \leq kn} S_i = \mathcal{X}$ (*i.e.*, all elements are covered).

In the following, we describe a k-strategy Φ that locates the target in G in at most $(k+2)n + p + 1$ steps.

The first step of Φ consists of probing only the vertex q. Three cases may occur. Either $H_1 = \{q\}$ in which case the target is located. Or the target is at distance 2 from q, *i.e.*, $H_1 = S$, in which case, Φ sequentially probes every vertex of S but one until the target is located, which takes at most p extra steps. Or the target is at distance 1 from q and $H_1 = X \cup X'' \cup \{s\}$.

Hence, we may assume that $H_1 = X \cup X'' \cup \{s\}$. In this case, Φ proceeds by phases of at most n steps each. There will be at most $k+2$ phases. Intuitively, during Phase $i \leq k+2$, the strategy Φ probes vertices in S^i in such a way that either the target is detected at one of the vertices of X^i, or at the end of the phase, the target is known not to be in X^i.

Let us assume by induction on $1 \leq i \leq k+2$ and $1 \leq j \leq n$ that, before the j^{th} step of Phase i, if the target has not been located yet, the set of possible locations for the target is

$$H_{1+(i-1)n+j-1} = \sigma \cup X'' \cup \{x^i_{3k(j-1)+1}, \cdots, x^i_{3kn}\} \cup (\bigcup_{i<y\leq k+2} X^y),$$

where $\sigma = \{s\}$ if $i = j = 1$ (or possibly, in the case $k = 1$, if $i = 1$ and $j = 2$), and $\sigma = \emptyset$ otherwise.

This holds for $i = j = 1$. Then, the strategy Φ consists of probing the vertices $\mathcal{P}_{i,j} = \{s^i_{k(j-1)+1}, \cdots, s^i_{kj}\}$. There are three cases to be considered. Before going into the details of the cases, recall that the sets $S_{k(j-1)+1}, \cdots, S_{kj}$ belong to the 3DM Y and so are pairwise disjoint. Hence, by construction of G, for every $a, b \in \mathcal{P}_{i,j}$, $(N_G(a) \cap X^i) \cap (N_G(b) \cap X^i) = \emptyset$.

- Either, all vertices of $\mathcal{P}_{i,j}$ are at distance one from the target. In this case, the target is located at s (this case may only happen for $i = j = 1$ or, possibly, $i = 1$ and $j = 2$ in the case $k = 1$).
- Or exactly one vertex, say $s^i_{k(j-1)+x}$ for $1 \leq x \leq k$, of $\mathcal{P}_{i,j}$ is at distance one from the target. Let $y = k(j-1) + x$. In this case, the target must be occupying a vertex of $x^i_{3(y-1)+1}, x^i_{3(y-1)+2}, x^i_{3(y-1)+3}$ (the vertices corresponding to the elements that are contained in S_y).
  The strategy Φ probes two of these vertices, until the target is located in at most 2 extra steps. Therefore, in this case, the target is located in at most $1 + (i-1)n + j + 2 \leq (k+2)n + p + 1$ steps (since $i \leq k+2$ and $j \leq n$).
- The last case is when all the vertices of $\mathcal{P}_{i,j}$ are at distance 2 from the target. In this latter case, it means that the target cannot occupy a vertex in $U = \{s\} \cup \{x^i_{3k(j-1)+1}, \cdots, x^i_{3kj}\}$. And so, if $j < n$, then $H_{1+(i-1)n+j} = H_{1+(i-1)n+j-1} \setminus U = X'' \cup \{x^i_{3kj+1}, \cdots, x^i_{3kn}\} \cup (\bigcup_{i<y\leq k+2} X^y)$, hence the induction hypothesis holds for $j+1$.
  To conclude, if $j = n$, $H_{1+in} = H_{1+(i-1)n+n-1} \setminus U = X'' \cup (\bigcup_{i+1<y\leq k+2} X^y)$ and the induction hypothesis holds for $i+1$ and $j = 1$. In this case, the Phase $i+1$ starts if $i+1 \leq k+2$.

After the n^{th} step of Phase $k+2$, we get that $H_{1+(k+2)n} = X''$. The strategy Φ ends by sequentially probing every vertex of X'' but one. So the target can be located in at most p extra steps. Therefore, $\lambda_k(G) \leq (k+2)n + p + 1$. ⋄

Claim **4** *If every 3DM of $\mathcal{I}$ has size $< kn$, then $\lambda_k(G) > (k+2)n + p + 1$.*

Proof of claim. Let us assume that every 3DM of $\mathcal{I}$ has size $< kn$. We show that every k-strategy needs at least $(k+2)n+p+2$ steps to guarantee the localization of the target.

To avoid technicality, let us assume that $H_0 = X \cup X''$, *i.e.*, the target is known *a priori* to occupy a vertex in $X \cup X''$. We show that even with this extra assumption (that is not favourable for the target), every k-strategy needs at least $(k+2)n + p + 2$ steps to guarantee the localization of the target.

Let Φ be any k-strategy. First, let us note that, since $H_0 = X \cup X''$ and both q and s are universal for $X \cup X''$, then probing q or s does not bring further information. Therefore, we may assume that Φ never probes q nor s.

Let us precisely describe the effect of probing one vertex depending on whether it is in X, X'' or S.

1. Let $u \in X''$. Note that, for every $z \in X \cup X'' \setminus \{u\}$, $d_G(u,z) = 2$. Therefore, probing u only determines if the target is on u or not, and gives no further information. We say that probing u only allows to remove u from the set of possible locations.
2. Let $u \in X^i$ for any $i \leq k+2$. Note that, for every $z \in X \cup X'' \setminus \{u\}$, $d_G(u,z) = 2$. Therefore, similarly, probing u only allows to remove u from the set of possible locations.
3. Let $u \in S^i$ for any $i \leq k+2$. Let $\{x,y,z\} = N_G(u) \cap X^i$, *i.e.*, x, y, and z are the vertices corresponding to the elements contained in the set that corresponds to u. For every $z \in X \cup X'' \setminus \{x,y,z\}$, $d_G(u,z) = 2$. Therefore, probing u only allows to remove at most 3 vertices, namely x, y, z, from the set of possible locations.
4. More generally, for any $Z \subseteq S^i$ with $|Z| < kn$. Probing all vertices of Z allows to remove $N_G(Z) \cap X^i$, *i.e.*, at most $3|Z|$ vertices from the set of possible locations of the target.
5. Finally, let $Z \subseteq S^i$ with $|Z| = kn$. Because $\mathcal{I}$ has no 3DM of size kn, there must be at least two vertices of Z whose neighbourhoods intersect in X^i. That is, $|N_G(Z) \cap X^i| \leq 3kn - 1$. Probing all vertices of Z allows to remove at most $3kn-1$ vertices from the set of possible locations of the target.

Let $P \subseteq X \cup X'' \cup S$ be the set of all vertices that have been probed during the $(k+2)n + p + 1$ first steps of Φ. We will show that, at this point, the set of possible locations for the target still contains at least two vertices and so an extra step is required.

For every $0 \leq j \leq kn$, let α_j be the number of sets S^i that contain exactly $kn - j$ vertices of P. Formally, $\alpha_j = |\{i \mid 1 \leq i \leq k+2, |S^i \cap P| = kn - j\}|$. For every $kn < j \leq m$, let α_j be the number of sets S^i whose exactly j vertices have been probed, *i.e.*, $\alpha_j = |\{i \mid 1 \leq i \leq k+2, |S^i \cap P| = j\}|$. By definition, since $|S^i| = m$ for every $i \leq k+2$:

$$\sum_{0 \leq j \leq m} \alpha_j = k+2. \tag{1}$$

Let $y = |X \cap P|$ be the total number of vertices probed in X and let $x'' = |X'' \cap P|$ be the total number of vertices probed in X''. By definition of y, x'', and the α's, the total number ρ of vertices that have been probed after $(k+2)n+p+1$ steps satisfies:

$$\rho = y + x'' + \sum_{kn < j \leq m} j\alpha_j + \sum_{0 \leq j \leq kn} (kn - j)\alpha_j. \tag{2}$$

Moreover, since at most k vertices can be probed each step:

$$\rho \leq k[(k+2)n + p + 1] \tag{3}$$

Note that, by Item (1) above, if $x'' \leq (k+2)m - 2$, then at least two vertices have not been probed and, therefore, are still potential locations for the target (as noticed above, a vertex of X'' can be removed from potential locations only by being probed). In such a case, another step would be needed to ensure the localization. Therefore, we may assume that $x'' \in \{(k+2)m-1; (k+2)m\}$.

Let us assume that $x'' = (k+2)m$ (below, we point out the few differences in the case $x'' = (k+2)m-1$). In that case, all vertices in X'' are removed from the possible locations of the target that must be in X.

Let $0 < j \leq kn$ and let $i \leq k+2$ such that $kn-j$ vertices have been probed in S^i. By item (4) above, the probes of the vertices in S^i remove at most $3(kn-j)$ vertices of X^i (and no other vertices) from the possible locations of the target. In other words, it leaves at least $3j$ vertices of X^i as possible locations for the target. Let $i \leq k+2$ such that kn vertices have been probed in S^i. By item (5) above, the probes of the vertices in S^i remove at most $3kn-1$ vertices of X^i (and no other vertices) from the possible locations of the target. In other words, it leaves at least 1 vertex of X^i as a possible location for the target.

Summing over all $j \in [\![0, kn]\!]$, the vertices probed in S leave at least $\alpha_0 + \sum_{1\leq j\leq kn} 3j\alpha_j$ vertices of X as possible locations for the target. To ensure the detection of the target without more steps, only one vertex of X must remain as a possible location (in the case when $x'' = (k+2)m-1$, *i.e.*, one vertex of X'' is still a possible location, then no vertex of X must remain possible). Since, by item (2) above, only the vertices y probed in X may allow removing further vertices from the set of possible locations, it follows that:

$$y + 1 \geq \alpha_0 + \sum_{1\leq j\leq kn} 3j\alpha_j, \tag{4}$$

(in the case $x'' = (k+2)m-1$, it becomes $y \geq \alpha_0 + \sum_{1\leq j\leq kn} 3j\alpha_j$).

We are now ready to show that the above equations lead to a contradiction, proving that an extra step is required. For this purpose, let us consider again the total number ρ of vertices that have been probed during the first $(k+2)n+p+1$ steps.

$$\begin{aligned}
\rho &= y + x'' + \sum_{kn<j\leq m} j\alpha_j + \sum_{0\leq j\leq kn} (kn-j)\alpha_j && (\textit{Equation 2})\\
&= y + x'' + \sum_{kn+1\leq j\leq m} (j-kn)\alpha_j + kn\sum_{0\leq j\leq m}\alpha_j - \sum_{0\leq j\leq kn} j\alpha_j\\
&= y + x'' + \sum_{kn+1\leq j\leq m} (j-kn)\alpha_j + kn(k+2) - \sum_{0\leq j\leq kn} j\alpha_j && (\textit{Equation 1})\\
&= y + (k+2)m + \sum_{kn+1\leq j\leq m} (j-kn)\alpha_j + kn(k+2) - \sum_{0\leq j\leq kn} j\alpha_j\\
&&& (\textit{if } x'' = (k+2)m)\\
&\geq \alpha_0 + \sum_{1\leq j\leq kn} 3j\alpha_j - 1 + (k+2)m + \sum_{kn+1\leq j\leq m} (j-kn)\alpha_j + kn(k+2) - \sum_{0\leq j\leq kn} j\alpha_j\\
&&& (\text{Equation 4})\ (\textit{if } x'' = (k+2)m)\\
&= \alpha_0 + \sum_{1\leq j\leq kn} 3j\alpha_j + pk + \sum_{kn+1\leq j\leq m} (j-kn)\alpha_j + kn(k+2) - \sum_{0\leq j\leq kn} j\alpha_j\\
&&& (\textit{by definition of } p)
\end{aligned}$$

$= k[n(k+2)+p+1] + \sum_{kn+1 \le j \le m} (j-kn)\alpha_j + \alpha_0 + \sum_{1 \le j \le kn} 2j\alpha_j - k$

$=$

$k[n(k+2)+p+1]+2(k+2)-2\sum_{0 \le j \le m} \alpha_j + \sum_{kn+1 \le j \le m} (j-kn)\alpha_j + \alpha_0 + \sum_{1 \le j \le kn} 2j\alpha_j - k$ (*Equation 1*)

$= k[n(k+2)+p+1] + 4 + \sum_{kn+1 \le j \le m} (j-kn)\alpha_j - 2\sum_{kn+1 \le j \le m} \alpha_j - \alpha_0 + \sum_{1 \le j \le kn} 2(j-1)\alpha_j + k$

$= k[n(k+2)+p+1] + 4 + \sum_{kn+2 \le j \le m} (j-kn-1)\alpha_j - \sum_{kn+1 \le j \le m} \alpha_j - \alpha_0 + \sum_{1 \le j \le kn} 2(j-1)\alpha_j + k$

$\ge k[n(k+2)+p+1] + 4 + k - \alpha_0 - \sum_{kn+1 \le j \le m} \alpha_j$

$\rho \ge k[n(k+2)+p+1] + 2$ (*Equation 1*)

This contradicts Eq. 3 and concludes the proof. ◇ □

Theorem 2. *Let $k \ge 2$ be a fixed integer. Given a graph G with a universal vertex and an integer $\ell \ge 1$, the problem of deciding whether $\lambda_k^{rel}(G) \le \ell$ is NP-complete.*

Proof. The proof is exactly the same as for Theorem 1, but the strategy designed in Claim 3 starts by probing both s and q (instead of only q). Moreover, in this strategy, if the target is in $S \cup \{s\}$, the localization may require one more step than for the case with exact distances (but the claim still holds). □

5.2 Proofs of Theorem 3 and Theorem 4

Theorem 3. *Let $\ell \ge 1$ be a fixed parameter. The ℓ-Step Localization Problem is NP-complete in the class of graphs with a universal vertex.*

Proof. For $\ell = 1$, the result follows from the fact that $\kappa_1(G)$ is exactly the metric dimension and from its NP-completeness [6].

Let $\ell \ge 2$ be fixed. We focus on proving the NP-hardness of the ℓ-Step Localization Problem, as it is clearly in NP. The proof is by reduction from the Metric Dimension Problem restricted to the class of graphs with diameter 2, which is known to be NP-hard [9]. Let thus $< G, k >$ be an instance of Metric Dimension where G has diameter 2. We construct, in polynomial time, an instance $< G', k >$ of the ℓ-Step Localization Problem such that $MD(G) \le k$ if and only if a target hidden in G' can be located in at most ℓ steps, by probing at most k vertices per step, *i.e.*, $\kappa_\ell(G) \le k$.

The construction of G' is as follows. Start from $k(\ell-1)+1$ disjoint copies $G_1, ..., G_{k(\ell-1)+1}$ of G. Add the vertices $v_1, ..., v_{k(\ell-1)+1}$ and add all the edges from v_i to every vertex of G_i for all integers $1 \le i \le k(\ell-1)+1$. Add a vertex

u to the graph, and all edges between u and each vertex of the copies of G. The resulting graph is G'. We will need the following claim.

Claim **5** *For some* $1 \leq a \leq k(\ell-1)+1$*, if the target is known to occupy a vertex of* G_a*, then probing a vertex* $w \in V(G' \setminus G_a)$ *does not remove any vertices in* G_a *as possible locations for the target.*

Proof of claim. u and v_a are universal to G_a and therefore, all vertices of G_a are the same distance from u and v_a, and every shortest path from w to a vertex of G_a includes either v_a or u and thus, any vertices of G_a cannot be distinguished by their distance to w. ◇

Let us show that $MD(G) \leq k$ if and only if $\kappa_\ell(G) \leq k$.

- First let us assume that $MD(G) \leq k$, we show that $\kappa_\ell(G') \leq k$. On step s for $1 \leq s \leq \ell-1$, let us probe the vertices $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$. If the target is at some vertex, say v_i, probed at this step, then it is located immediately. If the target is at distance 1 from such a probed vertex v_i, then it occupies a vertex in the corresponding G_i. Note that, because G has diameter 2, then each of its copies G_i is an isometric subgraph of G'. Hence, any resolving set of size k of G (it exists since $MD(G) \leq k$) is also a resolving set for the vertices of G_i in G'. Probing such a resolving set in G_i allows to locate the target during the next step $s+1$.
  If the target is at distance 2 of all v_is, then it is located at u.
  If on turn $\ell-1$, the target is at distance 4 from the probed vertices, then it is located at $v_{k(\ell-1)+1}$. Otherwise, it is in $G_{k(\ell-1)+1}$ and can be located next turn since we have assumed that $MD(G) \leq k$ and each G_i is isometric in G'.
- Now we prove the other direction, that is, we show that $MD(G) > k$ implies that $\kappa_\ell(G') > k$.
  Since there are $k(\ell-1)+1$ copies of G_i and only $k(\ell-1)$ probes may occur in the first $\ell-1$ steps, then on the last step, regardless of strategy, there will always exist a copy, say G_a for some $1 \leq a \leq k(\ell-1)+1$, for which no vertices in G_a have been probed. If the target is hidden in G_a, then by Claim 5, all the vertices of G_a are still potential locations for the target. The last step is not sufficient to locate a target hidden in G_a since probing a vertex $w \in V(G' \setminus G_a)$ is useless by Claim 5, G_a is isometric in G', and $MD(G_a) > k$. Hence, $\kappa_\ell(G') > k$. □

Theorem 4. *Let* $\ell \geq 1$ *be a fixed integer. Given a graph* G *with a universal vertex and* $2 \leq k \in \mathbb{N}$*, the problem of deciding if* $\kappa_\ell^{rel}(G) \leq k$ *is NP-complete.*

Proof. For $\ell = 1$, the result follows from the fact that $\kappa_1^{rel}(G)$ is exactly the centroidal dimension and from its NP-completeness [7].

Let $\ell \geq 2$ be fixed. The problem is clearly in NP. To prove the NP-hardness, let us reduce the Centroidal Dimension Problem restricted to the class of graphs with diameter 2 that contain a universal vertex, which is known to be NP-hard [7]. Let thus $< G, k >$ be an instance of Centroidal Dimension where

G has diameter 2 and contains a universal vertex. We construct, in polynomial time, an instance $< G', k >$ of our problem such that $CD(G) \leq k$ if and only if a target hidden in G' can be located in at most ℓ steps, by probing at most k vertices per step, *i.e.*, $\kappa_\ell^{rel}(G) \leq k$.

The construction of G' is as follows. Start from $k(\ell-1)+1$ disjoint copies $G_1, ..., G_{k(\ell-1)+1}$ of G. Let v be a universal vertex of G, and for $1 \leq i \leq k(\ell-1)+1$, let v_i denote the copy of v in G_i. Add all the edges so that $v_{k(\ell-1)+1}$ is a universal vertex in G'. The resulting graph is G'.

Claim **6** *Let $1 \leq a \leq k(l-1)+1$ and assume that the target is known to be in G_a, and all the vertices of G_a remain as possible positions for the target. if $CD(G) > k$, then probing k vertices is insufficient to locate the target in one step.*

Proof of claim. Assume k vertices are probed in G_a. Since $CD(G_a) > k$ and G_a is an isometric subgraph of G', there exist at least two vertices $y_1, y_2 \in G_a$ that cannot be distinguished based on the information received. That is, the distance orderings of the k probed vertices to y_1 and y_2 are identical. If any number of the k vertices probed in G_a had instead been replaced by vertices in $G' \setminus G_a$, then the distance orderings of the k probed vertices to y_1 and y_2 may change but they would still be identical to one another since $v_{k(\ell-1)+1}$ is a universal vertex (and thus, distance 1 from both y_1 and y_2) and a cut vertex which separates all the copies of G_i for all $1 \leq i \leq k(\ell-1)+1$. $\diamond$

- First let us assume that $CD(G) \leq k$, we show that $\kappa_\ell^{rel}(G') \leq k$. On step s for $1 \leq s \leq \ell-1$, let us probe the vertices $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$.
  If the target is closest to one of the vertices $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$ probed at step s, say $v_{(s-1)k+x}$ for some integer $1 \leq x \leq k$, then the target is at a vertex in $G_{(s-1)k+x}$. Indeed, all the G_is are separated by a cut vertex $v_{k(\ell-1)+1}$ and since $v_{k(\ell-1)+1}$ is universal, it is equidistant from all the vertices of $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$. Note that each G_i is an isometric subgraph of G'. Hence, any centroidal locating set of size k of G (it exists since $CD(G) \leq k$) is also a centroidal locating set for the vertices of G_i in G'. Probing such a centroidal locating set in G_i allows to locate the target during the next step $s+1 \leq \ell$.
  If the target is equidistant from each of the vertices $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$ probed at step s, then the target may not be at the vertices $\{v_{(s-1)k+1}, \cdots, v_{sk}\}$ nor at the vertices of $G_{(s-1)k+1}, \cdots, G_{sk}$. Therefore, if $s < \ell-1$, then $H_s = \{v_{sk+1}, \cdots, v_{(s+1)k+1}\} \cup \bigcup_{0 \leq i \leq k} V(G_{sk+1+i})$. Hence, after $s = \ell-1$ steps, then $H_s = V(G_{k(\ell-1)+1})$. Then, since each G_i is an isometric subgraph of G' and $CD(G) \leq k$, probing a centroidal locating set in $G_{k(\ell-1)+1}$ allows to locate the target during the next step $s+1 = \ell$.
- Now we prove the other direction, that is, we show that $CD(G) > k$ implies that $\kappa_\ell^{rel}(G') > k$.
  Whatever be the probing strategy, if on the last turn, there exists a copy, say G_a for some $1 \leq a \leq k(\ell-1)+1$, for which no vertices in G_a have been

probed, then there is no way to know at which vertex of G_a, the target is located. Indeed, all G_is are separated by a cut vertex, so probing a vertex in some G_i provides no information on any other G_j, $j \neq i$. Since there are $k(\ell-1)+1$ copies of G_i and only $k(\ell-1)$ probes may occur in the first $\ell-1$ steps, then on the last step, regardless of strategy, there will always exist a copy, say G_a for some $1 \leq a \leq k(\ell-1)+1$, for which no vertices in G_a have been probed. The last step is not sufficient to locate the target by Claim 6. Hence, $\kappa_\ell^{rel}(G') > k$. □

5.3 Proof of Theorem 5

Let us start with simple results that will be used below.

Claim **7** *Let S_n be the star with n leaves. Then $\lambda_k(S_n) = \lceil \frac{n-1}{k} \rceil$.*

Proof of claim. It comes from the fact that, in any star, the optimal strategy consists of probing all leaves but one. ◇

Claim **8** *Let $1 < r \in \mathbb{N}$ be such that $r-1 \equiv 0 \mod k$. For $1 < n \in \mathbb{N}$, let S_n^r be the tree obtained from r copies of S_n by adding one new vertex c adjacent to the center of each of the r stars.*

$$\lambda_k(S_n^r) = \frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil.$$

Furthermore, $MD(S_n^r) = r(n-1)$.

Proof of claim. For $1 \leq i \leq r$ and $1 \leq j \leq n$, let c^i be the center of the i^{th} copy of S_n, denoted by S^i, and let c_j^i be the j^{th} leaf of the i^{th} copy of S_n. First, we prove that $\lambda_k(S_n^r) \leq \frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil$. Consider a strategy ϕ that at step $1 \leq s \leq \frac{r-1}{k}$, probes the vertices $c_1^{(s-1)k+1}, \cdots, c_1^{sk}$. If at step s, one of the probed vertices, say $c_1^{(s-1)k+x}$ for some $1 \leq x \leq k$, is:

- distance 0 from the target, then the target is located at $c_1^{(s-1)k+x}$.
- distance 1 from the target, then the target is located at $c^{(s-1)k+x}$.
- distance 2 from the target and $k = 1$, then the target is located at c or $c_y^{(s-1)k+x}$ for some $2 \leq y \leq n$. The target is then located in a total of at most $s + \lceil \frac{n-1}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by $c_y^{(s-1)k+x}$ and its neighbours which happens to be a star S_n that is also an isometric subgraph of S_n^r.
- distance 2 from the target and $k > 1$, then the target is located at c if it is also distance 2 from the other probed vertices. Otherwise, it is at $c_y^{(s-1)k+x}$ for some $2 \leq y \leq n$. The target is then located in a total of at most $s + \lceil \frac{n-2}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by $c_y^{(s-1)k+x}$ and all its neighbours except for c, which happens to be a star S_{n-1} that is also an isometric subgraph of S_n^r.

If at step $s < \frac{r-1}{k}$, all of the probed vertices are at distance 3 from the target, then the target is located at one of the vertices $c^{sk+1}, \cdots, c^{(s+1)k}$. If at step $s < \frac{r-1}{k}$, all of the probed vertices are at distance 4 from the target, then the target is located at one of the vertices $c_j^{sk+1}, \cdots, c^{(s+1)k_j}$.

If at step $\frac{r-1}{k}$, all of the probed vertices are at distance 3 from the target, then the target is located at c^r. If at step $\frac{r-1}{k}$, all of the probed vertices are at distance 4 from the target, then the target is located at one of the vertices c_j^r. The target is then located in a total of at most $\frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil$ steps since it occupies a leaf of the subgraph induced by c^r and all its neighbours except for c which happens to be a star S_n that is also an isometric subgraph of S_n^r.

Now, we prove that $\lambda_k(S_n^r) > \frac{r-1}{k} + \lceil \frac{n-1}{k} \rceil - 1$. We may assume that the target is on a leaf as this is not favourable for the target.

Whatever be the probing strategy, if after some step, there exists a copy of S_n, say S^a for some $1 \leq a \leq r$, for which no vertices in S^a have been probed, then there is no way to know at which vertex of c_j^a, the target is located if it is there. Indeed, all c_j^as are separated by a cut vertex that is universal to them, so probing a vertex in $S_n^r \setminus S^a$ provides no information on the location of the target at a vertex c_j^a. Since there are r copies of S_n and only $k\frac{r-1}{k}$ probes may occur in the first $\frac{r-1}{k}$ steps, then after step $\frac{r-1}{k}$, there will always exist a copy of S_n, say S^a for some $1 \leq a \leq r$, for which no vertices in S^a have been probed. The next $\lceil \frac{n-1}{k} \rceil - 1$ steps of any strategy are not sufficient to locate the target if it occupies a leaf of S^a as the subgraph induced by S^a is an isometric subgraph of S_n^r and, as stated above, probing vertices in $S_n^r \setminus S^a$ does not provide any information. Therefore, $\lceil \frac{n-1}{k} \rceil$ steps are still required since S^a is a star with n leaves.

By [10,17], $MD(S_n^r) = r(n-1)$. ⋄

Theorem 5. *The* Localization *Problem is NP-complete in the class of trees.*

Proof. Since there always exists a winning strategy with length at most $O(n)$, it is a certificate with polynomial size and so the problem is in NP.

To prove the NP-hardness, let us reduce the Hitting-Set Problem. The inputs are an integer $k \geq 1$, a ground-set $B = \{b_1, \cdots, b_n\}$ and a set $\mathcal{S} = \{S_1, \cdots, S_m\}$ of subsets of B, *i.e.*, $S_i \subseteq B$ for every $i \leq m$. The Hitting-Set Problem aims at deciding if there exists a set $H \subseteq B$ such that $|H| \leq k$ and $H \cap S_i \neq \emptyset$ for every $i \leq m$.

Adding one new element to the ground-set and adding this element to one single subset clearly does not change the solution. Therefore, by adding some dummy elements (each one belonging to a single subset), we may assume that all subsets are of the same size σ and that $\sigma - 1 \equiv 0 \mod k$.

Let γ be any integer such that $\gamma - 1 \equiv 0 \mod k$ and $\gamma > n - k - 1$.

The instance T of the Localization Problem is built as follows. Let us start with n vertex-disjoint paths $B_1, \cdots, B_n$ (the *branches*) of length $2m$, where $B_i = (b_1^i, \cdots, b_{2m+1}^i)$ for each $i \leq n$. Then, let us add one new vertex r adjacent to b_1^i for all $i \leq n$. For every $1 \leq j \leq m$ and for every $1 \leq i \leq n$ such that

$b_i \in S_j$, let us add γ new vertices adjacent to b^i_{2j}. The subgraph induced by b^i_{2j} and by the γ leaves adjacent to it is referred to as the *star* representing the element i in the set S_j (or representing the set S_j in the branch i).

Intuitively, it will always be better for the target to be located in a leaf of some star because γ is "huge". During the first turn of any strategy, the *level* (roughly, the distance to the root) of the target can be identified. Each even level $2j$ corresponds to a set S_j. If, during the first turn, one star corresponding to each even level can be eliminated from the possible locations (which corresponds to hit every subset) then the strategy finishes one step earlier than if all subsets cannot be hit (if so, all stars would have to be checked).

More formally, we show below that $\lambda_k(T) \leq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$ if and only if there is a hitting set of size at most k.

Let us first show that, if there is a hitting set H of size at most k for $(B, \mathcal{S})$, then $\lambda_k(T) \leq \ell$ for any $\ell \geq 1 + \frac{\sigma-1}{k} + \frac{\gamma-1}{k}$. W.l.o.g. (up to renumbering the elements), let us assume that $H = \{b_1, \cdots, b_k\}$ and let us present the corresponding winning strategy. During the first turn, the vertices $b^1_{2m+1}, \cdots, b^k_{2m+1}$ are probed. There are two cases to be considered.

- First, if the target is at distance exactly $2m+1$ from one of (actually from all) the probed vertices, it is at r.
- Then, let us assume that the target is at distance $< 2m+1$ from one of the probed vertices, w.l.o.g., the target occupies a vertex in the branch B_1 (including the leaves of the stars in this branch). If the target is at an odd distance from b^1_{2m+1}, then the target is identified since there is a unique vertex at distance $2h+1$ from b^1_{2m+1} for each $0 \leq h \leq m$. Otherwise, the target is at distance $d = 2(m-h)$ from d_{2m+1} for some $0 \leq h < m$ (if $h = m$, the target is trivially located). If $b_1 \notin S_{m-h}$, then b^1_{2m+1-d} does not belong to a star and b^1_{2m-d} is the unique vertex at distance d from b^1_{2m+1} and the target is located. Otherwise, the target may occupy b^1_{2m+1-d} or any leaf adjacent to b^1_{2m+1-d}. By Claim 7, this can be checked in $\lceil \frac{\gamma}{k} \rceil$ steps by sequentially checking each of these vertices but one.
  Overall, in this case, the target is located in at most $1 + \lceil \frac{\gamma}{k} \rceil$ steps (including the first one).
- Hence, we may assume that the target is at distance at least $2m+2$ from each of $b^1_{2m+1}, \cdots, b^k_{2m+1}$. Note that, in this case, the target is the same distance from every probed vertex. Said differently, the information brought by the first turn is that the target is at some distance $d \geq 1$ from the root c and not in branches $B_1, \cdots, B_k$.
  If d is even, then the target can be at $b^{k+1}_d, \cdots, b^n_d$. Indeed, for every $i \leq n$, and any even distance d', there is a unique vertex at distance d' from r in the branch B_i. By Claim 7, the target can be located in $\lceil \frac{n-k-1}{k} \rceil$ steps by sequentially checking each of these vertices but one. Overall, it took $1 + \lceil \frac{n-k-1}{k} \rceil$ steps to locate the target.
  Otherwise, $d = 2j+1$ for some $j \leq m$. Recall that H is a hitting set. In particular, $|S_j \setminus H| < |S_j| = \sigma$. In the worst case, $|S_j \setminus H| = \sigma - 1$ and, w.l.o.g. (up to renumbering), $S_j \setminus H = \{b_{k+1}, \cdots, b_{k+\sigma-1}\}$. In this case, the

target can be located at $b_d^{k+1}, \cdots, b_d^n$ or at any leaf adjacent to one of the vertices $b_{2j}^{k+1}, \cdots, b_{2j}^{k+\sigma-1}$ (*i.e.*, the leaves of the stars corresponding to the set S_j in the branches that have not been hit). Then, the strategy continues by sequentially probing the vertices $b_d^{k+1}, \cdots, b_d^{n-1}$ (Note that we start by the branches containing the stars that remain to be checked). There are two cases to be considered.

- Either after checking $b_d^{k+1}, \cdots, b_d^{k+\sigma-1}$ in $\frac{\sigma-1}{k}$ steps (recall that $\sigma-1 \equiv 0 \mod k$), the target is located to be in some star (this is the case if it is at distance 2 from one probed vertex). Then, it remains to identify which leaf of the star is the location of the target. This can be done in $\frac{\gamma-1}{k}$ steps by sequentially checking each of these leaves but one (Claim 7). Overall, in this case, the target has been located in $1+\frac{\sigma-1}{k}+\frac{\gamma-1}{k}$ steps.
- Or the target does not occupy a leaf of a star and is located after a total of $1+\lceil\frac{n-k-1}{k}\rceil$ steps (including the first step).

To conclude, if the minimum size of a hitting set is at most k, then $\lambda_k(T) \leq \ell$ for any $\ell \geq 1+\max\{\lceil\frac{\gamma}{k}\rceil, \lceil\frac{n-k-1}{k}\rceil, \frac{\sigma-1}{k}+\frac{\gamma-1}{k}\} = 1+\frac{\sigma-1}{k}+\frac{\gamma-1}{k}$ (the last equality holds since $\gamma > n-k-1$ and, since $\sigma-1 \equiv 0 \mod k$ and $\sigma > 1$, $\frac{\sigma-1}{k} \geq 1$).

Now, let us show that, if there are no hitting sets of size at most k, then $\lambda_k(T) > \ell$ for any $\ell \leq 1+\frac{\sigma-1}{k}+\frac{\gamma-1}{k}$. Consider any strategy probing at most k vertices per step. After the first turn, at most k branches have some vertex that has been probed. These at most k branches correspond to at most k elements of the ground-set B and, since all hitting sets have size at least $k+1$, there must be a set that does not contain any of these k elements. W.l.o.g., let $S_1 = \{b_1, \cdots, b_\sigma\}$ be this set. After the first step, let us assume that the target is located at distance 3 from the root (it is possible to decide this *a posteriori* since we are considering a worst case). Therefore, the target may be located at any leaf of some star corresponding to S_1. More precisely, the target may be at any vertex in $\{b_3^1, \cdots, b_3^\sigma\}$ or at any leaf adjacent to one of the vertices $\{b_2^1, \cdots, b_2^\sigma\}$. Actually, the target may also be at other vertices (the third vertex of other branches), but we can ignore these choices. Indeed, even removing these choices (so making the target less powerful), we show that the strategy will last long enough.

Indeed, after the first step, the instance becomes equivalent to an instance that consists of a rooted tree whose root has degree σ and each child of the root is adjacent to $\gamma+1$ leaves, and the target is only known to occupy a leaf. By a direct adaptation of Claim 8, locating the target takes another $\frac{\sigma-1}{k}+\lceil\frac{\gamma}{k}\rceil$ steps.

Overall, locating the target requires at least $1+\frac{\sigma-1}{k}+\lceil\frac{\gamma}{k}\rceil$ steps. Since $\gamma-1 \equiv 0 \mod k$, $\lceil\frac{\gamma}{k}\rceil > \frac{\gamma-1}{k}$ and $\lambda_k(T) > 1+\frac{\sigma-1}{k}+\frac{\gamma-1}{k}$. □

5.4 Algorithm in trees

Example. Figure 3 describes a simple example. The root r of T has 8 children $v_1, \ldots, v_8$ with the pairs $(\lambda_k^L(T_{v_i}), \pi(T_{v_i}))$ being $(4,2)$, $(4,1)$, $(3,3)$, $(3,3)$, $(2,2)$, $(2,2)$, $(1,1)$, and $(0,0)$, respectively. Let $k=4$. There is a strategy ϕ which identifies the target in at most 4 steps.

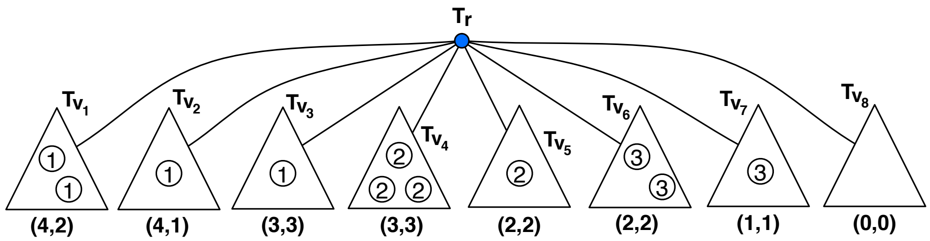

Fig. 3: A tree T rooted at r. The eight children of r are $v_1, \ldots, v_8$. The pairs $(\lambda_k^L(T_{v_i}), \pi(T_{v_i}))$ are written below each subtree. In the figure, 1 (2, 3, resp.) ⓙ in a subtree corresponds to 1 (2, 3, resp.) vertex (vertices) of this subtree probed during step j.

- Step 1. The probed vertices are those labeled 1 in Figure 3, that is, 2 vertices of T_{v_1}, 1 vertex of T_{v_2}, and 1 vertex of T_{v_3}. If the target occupies some leaf of T_{v_1} or T_{v_2}, then there is a strategy which will locate the target in at most $\lambda_k^L(T_{v_1}) - 1 = \lambda_k^L(T_{v_2}) - 1 = 3$ extra steps because $\pi(T_{v_1})$ ($\pi(T_{v_2})$, resp.) vertices of T_{v_1} (T_{v_2}, resp.) have been probed. If the target occupies some leaf of T_{v_3}, then there is a strategy which will locate the target in at most $\lambda_k^L(T_{v_3}) = 3$ extra steps (that is a total of 4 steps). Thus, assume that the target occupies a leaf of some subtree T_{v_i}, $4 \leq i \leq 8$.
- Step 2. The probed vertices are those labeled 2 in Figure 3, that is, 3 vertices of T_{v_4} and 1 vertex of T_{v_5}. If the target occupies some leaf of T_{v_4} or T_{v_5}, then using similar arguments to those above, we can show there is a strategy which will locate the target in at most 2 extra steps (that is a total of 4 steps). Thus, assume that the target occupies a leaf of T_{v_6}, T_{v_7} or T_{v_8}.
- Step 3. The probed vertices are those labeled 3 in Figure 3, that is, 2 vertices of T_{v_6} and 1 vertex of T_{v_7}. Again, if the target occupies some leaf of T_{v_6} or T_{v_7}, then with at most 1 extra step, the target is located. Otherwise, the target is on T_{v_8} and there is no need for an extra step.

Proof of Lemma 1 Let us first mention important claims.

Claim **9** *$\lambda_k^L(T) = 0$ if and only if T is a rooted path, and $\pi(T) = 0$ if and only if T is a rooted path.*

Claim **10** *For any tree T and any subtree T' of T, $\lambda_k^L(T') \leq \lambda_k(T') \leq \lambda_k(T)$ and $\lambda_k^L(T') \leq \lambda_k^L(T)$.*

Claim **11** *For any tree T, there exists a k-strategy that locates the target in at most $\lambda_k(T) + 1$ steps (resp., in at most $\lambda_k^L(T) + 1$ steps if the target is known to occupy a leaf) and that probes a single arbitrary vertex during its first step.*

Claim **2** *Let $S \in \mathcal{S}$ with $d = |V(S)| - 1$ (i.e., d is degree of the root r). Then, $\lambda_k^L(S) = \lceil \frac{d-1}{k} \rceil$ and $\pi(S) = -k(\lceil \frac{d-1}{k} \rceil - \lceil \frac{d-1}{d} \rceil) + (d-1)$.*

Proof of claim. The strategy consists of sequentially probing each leaf of S but one. Either the target will be probed at some step, or it must be in the unique leaf that has not been probed. During the first step, $\pi(S)$ leaves are probed, and exactly k leaves are probed during every other step. Such a strategy takes $\lceil \frac{d-1}{k} \rceil$ steps.

For any strategy using less than $\lceil \frac{d-1}{k} \rceil$ steps, the vertices of at most $k(\lceil \frac{d-1}{k} \rceil - 1) \leq d-2$ branches have been probed. Hence, there are at least two branches of S for which no vertices have been probed and so it is not possible to decide which one of these branches is occupied by the target.

Similarly, it can be checked that, for any strategy using at most $\lceil \frac{d-1}{k} \rceil$ steps and that would probe less than $\pi(S)$ vertices during the first step, there are at least two branches of S for which no vertices have been probed. It makes the proof easier to notice that $\pi(S) = -k(\lceil \frac{d-1}{k} \rceil - \lceil \frac{d-1}{d} \rceil) + (d-1)$, is equivalent to:

- $\pi(S) = 0$ if $d-1 = 0$;
- $\pi(S) = k$ if $d-1 > 0$ and $(d-1) \mod k = 0$;
- $\pi(S) = (d-1) \mod k$ otherwise.

$\diamond$

Lemma 1. *For every* $1 \leq i \leq d+1$, $\lambda_k^L(T[i,d^*]) = l_i - 1$ *and* $\pi(T[i,d^*]) = k - p_i$.

Proof. The proof is by induction on $d+1-i \leq d+1$. For $i = d+1$, there are two cases to be considered.

- If $d = d^*$ (*i.e.*, the condition on Line 2 is not satisfied), then, before the first iteration, $l_{d+1} = 1, p_{d+1} = k$ and $T[d+1,d^*] = \emptyset$, and so $\lambda_k^L(\emptyset) = l_{d+1} - 1 = 0$ and $\pi(\emptyset) = k - p_{d+1} = 0$. So the induction hypothesis is satisfied for $i = d+1$.
- Otherwise, $d < d^*$ and $T_{v_{d+1}}, \cdots, T_{v_{d^*}}$ are rooted paths. That is, $T[d+1,d^*] \in \mathcal{S}$. Then, the induction hypothesis for $i = d+1$ is satisfied by Claim 2 and Lines 2 to 5 of Algorithm 2.

Let us assume that the induction hypothesis holds for $1 < i+1 \leq d+1$. That is, at the end of the $(d-i)^{th}$ iteration of the for-loop, $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1$ and $\pi(T[i+1,d^*]) = k - p_{i+1}$. We will prove that it is also true after the next iteration of the for-loop, *i.e.*, $\lambda_k^L(T[i,d^*]) = l_i - 1$ and $\pi(T[i,d^*]) = k - p_i$.

It is very important to note that, Lines 2 and 3 imply that $\lambda_i > 0$ and $\pi_i > 0$, for every $1 \leq i \leq d$.

There are five different cases to be considered depending on the values of $p_{i+1}, \pi_i, \lambda_i$, and l_{i+1}.

- Case $0 < \pi_i \leq p_{i+1}$, $l_{i+1} = \lambda_i + 1$.
  By induction hypothesis, $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1,d^*]) = k - p_{i+1}$.
  Because the value of l at the beginning of this iteration of the for-loop is $l_{i+1} = \lambda_i + 1$, then $\alpha = \pi_i$. Then, since $\pi_i \leq p_{i+1}$, p becomes $p - \alpha = p_{i+1} - \pi_i$ and l is not modified. Hence, $l_i = l_{i+1}$, and $p_i = p_{i+1} - \pi_i$.
  We now prove that $\lambda_k^L(T[i,d^*]) = l_{i+1} - 1$ and $\pi(T[i,d^*]) = k - p_{i+1} + \pi_i$.

By Claim 10, $\lambda_k^L(T[i,d^*]) \geq \lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1$.
To prove that $\lambda_k^L(T[i,d^*]) \leq \lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1$, it is sufficient to describe a strategy ϕ for $\lambda_k^L(T[i,d^*])$ with a total of at most $l_{i+1} - 1$ steps. Let ϕ' be an optimal strategy for $T[i+1,d^*]$ probing at most $\pi(T[i+1,d^*])$ vertices during the first step. Let also ϕ'' be an optimal strategy for $T[i]$ probing at most π_i vertices during the first step.
The first step of ϕ consists of probing π_i vertices of $T[i]$ (as ϕ'') and $\pi(T[i+1,d^*]) = k - p_{i+1}$ vertices of $T[i+1,d^*]$ (as ϕ'). By assumption, $\pi_i \leq p_{i+1}$, and, by induction hypothesis, $\pi(T[i+1,d^*]) = k - p_{i+1}$, so $\pi_i + \pi(T[i+1,d^*]) \leq k$ and at most k vertices are probed. By Claim 1, this first step allows to decide if the target is in $T[i]$ or not (in the latter case, it is in $T[i+1,d^*]$). If the target is in $T[i]$, then continue the strategy ϕ'' in $T[i]$ which will locate the target in at most $\lambda_i - 1 = l_{i+1} - 2$ extra steps. Otherwise (the target is in $T[i+1,d^*]$), continue the optimal strategy ϕ' for $T[i+1,d^*]$ which will locate the target in at most $\lambda_k^L(T[i+1,d^*]) - 1 = l_{i+1} - 2$ extra steps. In all cases, ϕ locates the target in at most $l_{i+1} - 1$ steps.
We prove that $\pi(T[i,d^*]) = k - p_{i+1} + \pi_i$. For purpose of contradiction, let us assume that there is a strategy locating the target in $T[i,d^*]$ in at most $\lambda_i = l_{i+1} - 1$ steps and probing $< k - p_{i+1} + \pi_i$ vertices during the first step. By definition, at least π_i vertices of $T[i]$ must be probed during the first step to locate the target in at most $\lambda_i = l_{i+1} - 1$ steps. Thus, it means that $< k - p_{i+1}$ vertices of $T[i+1,d^*]$ can be probed during the first step. This contradicts that the strategy performs in at most $\lambda_i = l_{i+1} - 1$ steps since $\pi(T[i+1,d^*]) = k - p_{i+1}$.

– Case $\pi_i > p_{i+1} > 0$, $l_{i+1} = \lambda_i + 1$.
In this case, it can be checked that $\alpha = \pi_i$ and that the "else" (Line 12) is executed and so $l_i = l_{i+1} + 1$ and $p_i = k - 1$. We will prove that $\lambda_k^L(T[i,d^*]) = l_{i+1}$ and $\pi(T[i,d^*]) = 1$.
By induction hypothesis, $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1,d^*]) = k - p_{i+1}$.
We prove that $\lambda_k^L(T[i,d^*]) \geq \lambda_k^L(T[i+1,d^*]) + 1 = l_{i+1}$. For purpose of contradiction, let us assume that $\lambda_k^L(T[i,d^*]) < l_{i+1}$ and let ϕ' be a strategy for $T[i,d^*]$ locating the target in at most $l_{i+1} - 1$ steps. Since $l_{i+1} - 1 = \lambda_i$, then at least π_i vertices of $T[i]$ must be probed during the first step. Since $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1 = \lambda_i$ and $\pi(T[i+1,d^*]) = k - p_{i+1}$, at least $k - p_{i+1}$ vertices of $T[i+1,d^*]$ must be probed during the first step. This means that at least $\pi_i + k - p_{i+1} > k$ vertices must be probed during the first step, a contradiction.
We now prove that $\lambda_k^L(T[i,d^*]) = l_{i+1}$. It is sufficient to design a strategy ϕ for $T[i,d^*]$ locating the target in at most l_{i+1} steps. By Claim 11, there is a strategy ϕ' for $T[i]$ that locates the target in at most $\lambda_i + 1$ steps and probes a single vertex during the first step. Let also ϕ'' be an optimal strategy for $T[i+1,d^*]$.
The first step of ϕ consists of probing one vertex of $T[i]$. If the target is in $T[i]$, the strategy continues with ϕ' (in at most $\lambda_i = l_{i+1} - 1$ steps), otherwise, the strategy continues with ϕ'' (in at most $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1$ steps).

We deduce that $\pi(T[i,d^*]) \leq 1$ and by definition of π, we get that $\pi(T[i,d^*]) = 1$.

- Case $p_{i+1} = 0$, $l_{i+1} \geq \lambda_i + 1$.

  In that case, because of the "if" (Line 7), p is set to k and $l_i = l_{i+1} + 1$. Then, $\alpha = 1$ and so ("if" on Line 10) $p_i = k - 1$.

  We will prove that $\lambda_k^L(T[i,d^*]) = l_{i+1}$ and $\pi(T[i,d^*]) = 1$.

  By induction hypothesis, $\lambda_k^L(T[i+1,d^*]) = l_{i+1} - 1 \geq \lambda_i$ and $\pi(T[i+1,d^*]) = k$.

  - We prove that $\lambda_k^L(T[i,d^*]) \geq \lambda_k^L(T[i+1,d^*]) + 1 = l_{i+1}$. For purpose of contradiction, let us assume that $\lambda_k^L(T[i,d^*]) < l_{i+1}$ and let ϕ be a k-strategy for $T[i,d^*]$ locating the target in at most $l_{i+1} - 1$ steps. First, if a vertex of $T[i]$ is probed during the first step of ϕ, it means that at most $k - 1 < k - p_{i+1} = k$ vertices of $T[i+1,d^*]$ are probed during the first step of ϕ, contradicting that $k - p_{i+1} = \pi(T[i+1,d^*])$ is the minimum number of vertices of $T[i+1,d^*]$ that must be probed during the first step of an optimal k-strategy for $T[i+1,d^*]$.

    Hence, neither ϕ nor any k-strategy locating the target in $T[i,d^*]$ in at most $l_{i+1} - 1$ steps can probe some vertex of $T[i]$ during its first step. Below, we will build such a strategy ϕ' (that probes some vertex of $T[i]$ during its first step) from ϕ, which leads to a contradiction.

    Since ϕ does not probe any vertex of $T[i]$ during its first step, then $l_{i+1} - 1 > \lambda_i$ (otherwise, a target hidden in $T[i]$ will not be located in at most $l_{i+1} - 1$ steps, by definition of $\lambda_i > 0$).

    Let $x > 1$ be the first step of ϕ that probes a vertex of $T[i]$ if the target is in $T[i]$ (such a step exists since $T[i]$ is not a rooted path by definition of d, *i.e.*, since $\lambda_i > 0$). Then, $l_{i+1} - x \geq \lambda_i$ since, otherwise, a target hidden in $T[i]$ could not be located by ϕ in at most $l_{i+1} - 1$ steps. If $l_{i+1} - x = \lambda_i$, then the x^{th} step of ϕ must probe π_i vertices of $T[i]$. Otherwise, if $l_{i+1} - x > \lambda_i$, we may assume that the x^{th} step of ϕ probes a single vertex of $T[i]$ (by Claim 11).

    Let $i + 1 \leq j \leq d^*$ be such that the first step of ϕ probes some vertex of $T[j]$. Because the subtrees have been sorted, $\lambda_j \leq \lambda_i < l_{i+1} - 1$ and we may assume that the first step of ϕ probes one vertex in $T[j]$ (by Claim 11).

    Let us define the k-strategy ϕ' as follows. The strategy ϕ' follows ϕ but, during its first step, it probes one vertex of $T[i]$ instead of probing some vertices of $T[j]$. If the target is detected in $T[i]$, ϕ' applies an optimal strategy in $T[i]$ and locates the target in at most $\lambda_i < l_{i+1} - 1$ extra steps. Otherwise, ϕ' continues to mimic the strategy ϕ until its x^{th} step. If the target has been detected in some subtree before the x^{th} step, Strategy ϕ' continues to act as ϕ. Otherwise, the x^{th} step of ϕ' mimics the x^{th} step of ϕ but, instead of probing one vertex of $T[i]$ (resp. π_i vertices of $T[i]$ if $l_{i+1} - x = \lambda_i$), Strategy ϕ' probes one vertex of $T[j]$ (resp. $\pi_j \leq \pi_i$ vertices of $T[j]$ if $l_{i+1} - x = \lambda_i$). Then, ϕ' proceeds as ϕ.

It is easy to show that ϕ' is a k-strategy for $T[i,d^*]$ locating the target in at most $l_{i+1}-1$ steps, and probing some vertex of $T[i]$ during its first step, a contradiction.

- We now prove that $\lambda_k^L(T[i,d^*]) = l_{i+1}$ and that $\pi(T[i,d^*]) = 1$. It is sufficient to design a strategy ϕ for $T[i,d^*]$ locating the target in at most l_{i+1} steps. By Claim 11, there is a strategy ϕ' for $T[i]$ that locates the target in at most λ_i+1 steps and probes a single vertex during the first step. Let also ϕ'' be an optimal strategy for $T[i+1,d^*]$.
  The first step of ϕ consists of probing one vertex of $T[i]$. If the target is in $T[i]$, the strategy continues with ϕ' (in at most $\lambda_i \le l_{i+1}-1$ extra steps), otherwise, the strategy continues with ϕ'' (in at most $\lambda_k^L(T[i+1,d^*]) = l_{i+1}-1$ extra steps).
  We deduce that $\pi(T[i,d^*]) \le 1$ and by definition of π, we get that $\pi(T[i,d^*]) = 1$.

– Case $p_{i+1} > 0$, $l_{i+1} > \lambda_i + 1$.
In this case, the condition of the "if" (Line 7) is not satisfied, $\alpha = 1$ and so the condition of the "if" (Line 10) is satisfied. Hence, $l_i = l_{i+1}$ and $p_i = p_{i+1}-1$. We will prove that $\lambda_k^L(T[i,d^*]) = l_{i+1}-1$ and $\pi(T[i,d^*]) = k - p_{i+1} + 1$.
By induction hypothesis, $\lambda_k^L(T[i+1,d^*]) = l_{i+1}-1 > \lambda_i$ and $\pi(T[i+1,d^*]) = k - p_{i+1}$.
By Claim 10, $\lambda_k^L(T[i,d^*]) \ge \lambda_k^L(T[i+1,d^*]) = l_{i+1}-1$.
To prove that $\lambda_k^L(T[i,d^*]) \le \lambda_k^L(T[i+1,d^*]) = l_{i+1}-1$, it is sufficient to describe a strategy ϕ for $\lambda_k^L(T[i,d^*])$ with a total of at most $l_{i+1}-1$ steps. By Claim 11, there is a strategy ϕ' for $T[i]$ that locates the target in at most λ_i+1 steps and probes a single vertex during the first step. Let ϕ'' be an optimal strategy for $T[i+1,d^*]$ probing at most $\pi(T[i+1,d^*]) = k - p_{i+1} < k$ vertices during the first step.
The first step of ϕ consists of probing one vertex in $T[i]$ (as ϕ') and $\pi(T[i+1,d^*]) = k-p_{i+1}$ vertices of $T[i+1,d^*]$ (as ϕ''). By assumption, $0 < p_{i+1}$, so $1+\pi(T[i+1,d^*]) \le k$ and at most k vertices are probed. By Claim 1, this first step allows to decide if the target is in $T[i]$ or not (in which case, it is in $T[i+1,d^*]$). If the target is in $T[i]$, then continue the strategy ϕ' in $T[i]$ which will locate the target in at most $\lambda_i < l_{i+1}-1$ extra steps. Otherwise (the target is in $T[i+1,d^*]$), continue the optimal strategy ϕ'' for $T[i+1,d^*]$ which will locate the target in at most $\lambda_k^L(T[i+1,d^*]) - 1 = l_{i+1}-2$ extra steps. In all cases, ϕ locates the target in at most $l_{i+1}-1$ steps.
Let us prove that $\pi(T[i,d^*]) = k - p_{i+1} + 1$. For purpose of contradiction, let us assume that there is a strategy ϕ locating the target in $T[i,d^*]$ in at most $l_{i+1}-1$ steps and probing $< k - p_{i+1} + 1$ vertices during the first step. We will show that we can construct a strategy ϕ' in $T[i+1,d^*]$ that locates the target in at most $\ell_{i+1}-1$ steps and probes at most $k-p_{i+1}-1$ vertices during the first step, a contradiction. If the first step of ϕ probes at least one vertex of $T[i]$, then it probes at most $k-p_{i+1}-1$ vertices of $T[i+1,d^*]$ contradicting the fact that $\lambda_k^L(T[i+1,d^*]) = l_{i+1}-1$ and $\pi(T[i+1,d^*]) = k-p_{i+1}$. Hence, we may assume that the first step of ϕ probes $k-p_{i+1}$ vertices of $T[i+1,d^*]$ and no vertices in $T[i]$.

Let $t > 1$ be the minimum integer such that at least one vertex of $T[i]$ is probed during the t^{th} step of ϕ. After step t, at most $l_{i+1}-t-1$ steps remain and so $l_{i+1}-t-1 \geq \lambda_i - 1$. Let $j \in [\![i+1, d^*]\!]$ be such that at least one vertex of $T[j]$ is probed during the first step of ϕ. Note that $j > i$ and, because the subtrees are ordered in non-increasing lexicographical order, either $\lambda_j < \lambda_i$ or ($\lambda_j = \lambda_i$ and $\pi_j \leq \pi_i$).
Let us consider the following strategy ϕ' for $T[i+1, d^*]$. The first $t-1$ steps of the strategy ϕ' follow the ones of ϕ but do not probe any vertex of $T[j]$. That is, for every $j' \in [\![i+1, d^*]\!] \setminus \{j\}$ and for every $t' < t$, the step t' of ϕ' probes the same vertices of $T[j']$ as the step t' of ϕ. In particular, the first step of ϕ' probes at most $k - p_{i+1} - 1$ vertices.
If the target has been detected in a subtree different from $T[j]$ during the first $t-1$ steps, then ϕ' continues as ϕ (but without probing the vertices of $T[i]$ since ϕ' is a strategy for $T[i+1, d^*]$). Otherwise, the t^{th} step of ϕ' proceeds as follows. For every $j' \in [\![i+1, d^*]\!] \setminus \{j\}$, the step t of ϕ' probes the same vertices of $T[j']$ as the step t of ϕ. Again, the strategy ϕ' does not probe any vertex of $T[i]$. Note that, during its step t, the strategy ϕ probes at least one vertex of $T[i]$, and it probes at least π_i vertices of $T[i]$ if $l_{i+1} - t = \lambda_i$. Therefore, there are two cases to be considered.

- If $l_{i+1} - t > \lambda_j$, then ϕ' probes one vertex of $T[j]$ during step t. If the target is detected in $T[j]$ then the next steps of ϕ' follow an optimal strategy in $T[j]$ and will locate the target in at most λ_j extra steps. Otherwise, the next steps of ϕ' follow the ones of ϕ.
- If $l_{i+1}-t = \lambda_j$, then it implies that $l_{i+1}-t = \lambda_i$ (since $\ell_{i+1}-t \geq \lambda_i \geq \lambda_j$) and that the step t of ϕ was probing π_i vertices in $T[i]$. The strategy ϕ' replaces these π_i probes by probing $\pi_j \leq \pi_i$ vertices of $T[j]$. If the target is detected in $T[j]$ then the next steps of ϕ' follow an optimal strategy in $T[j]$ and will locate the target in at most $\lambda_j - 1$ extra steps. Otherwise, the next steps of ϕ' follow the ones of ϕ.

Overall, ϕ' is a strategy that locates a target in $T[i+1, d^*]$, in at most $l_{i+1}-1$ steps, and probing at most $k - p_{i+1} - 1$ vertices during the first step. This contradicts the fact that $\pi(T[i+1, d^*]) = k - p_{i+1}$. Hence, $\pi(T[i, d^*]) = k - p_{i+1} + 1$.

– Case $l_{i+1} < \lambda_i + 1$.
In this case, because of the "if" (Line 7), p is set to k and $l_i = \lambda_i + 1$. Then, $\alpha = \pi_i$ and so ("if" on Line 10) $p_i = k - \pi_i$.
We will prove that $\lambda_k^L(T[i, d^*]) = \lambda_i$ and $\pi(T[i, d^*]) = \pi_i$.
By induction hypothesis, $\lambda_k^L(T[i+1, d^*]) = l_{i+1}-1 < \lambda_i$ and $\pi(T[i+1, d^*]) = k - p_{i+1}$.
By Claim 10, $\lambda_k^L(T[i, d^*]) \geq \lambda_k^L(T[i]) = \lambda_i$.
To prove that $\lambda_k^L(T[i, d^*]) \leq \lambda_i$, it is sufficient to describe a strategy ϕ for $\lambda_k^L(T[i, d^*])$ with a total of at most λ_i steps. Let ϕ' be an optimal strategy for $T[i]$ probing at most π_i vertices during the first step. Let ϕ'' be an optimal strategy for $T[i+1, d^*]$ that locates the target in at most $l_{i+1} - 1 < \lambda_i$ steps and probing at most $\pi(T[i+1, d^*]) = k - p_{i+1}$ vertices during the first step.

The first step of ϕ probes π_i vertices of $T[i]$ (as ϕ'). By Claim 1, this first step allows to decide if the target is in $T[i]$ or not. If it is in $T[i]$ then ϕ follows ϕ'. Otherwise, ϕ executes ϕ'' in $T[i+1, d^*]$.
To conclude, let us prove that $\pi(T[i, d^*]) = \pi_i$. The previous strategy ϕ shows that $\pi(T[i, d^*]) \leq \pi_i$. Since $\lambda_k^L(T[i, d^*]) = \lambda_i$, any strategy for $T[i, d^*]$ must probe at least π_i vertices of $T[i]$ during the first step by definition of π_i. This concludes the proof. □