Sequential Metric Dimension (in trees)

Julien Bensmail, Dorian Mazauric, Fionn Mc Inerney,
Nicolas Nisse, Stéphane Pérennes

Université Nice Côte d'Azur, France

Indian Statistical Institute, Kolkata, India
February 6, 2019

Introduction to the problem

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

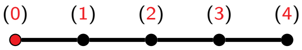
This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.



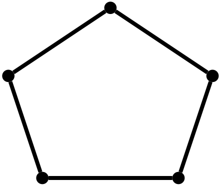
$\mathrm{MD}(P_n) = 1$

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.

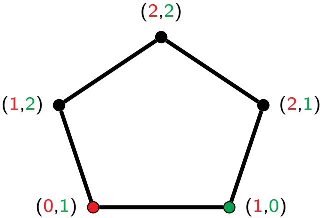


Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.



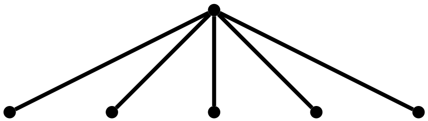
$\mathrm{MD}(C_n) = 2$

Rules:

- Graph $G = (V, E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v, t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.

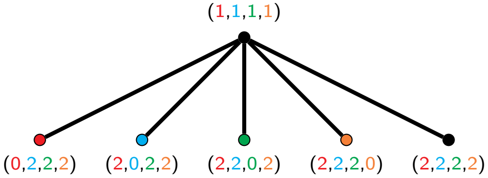


Rules:

- Graph $G=(V,E)$;
- "Secret" vertex $t \in V$;
- Probing a vertex $v \Rightarrow \mathrm{dist}_G(v,t)$.

Locate t (*at once*) by probing a minimum set of vertices?

This minimum = Metric Dimension $\mathrm{MD}(G)$ of G.



$$\mathrm{MD}(S_n) = n-1$$

How faster can we locate the target through *multiple* probing steps of *one* vertex?

Min. # of steps = Sequential Location Number $\mathrm{SL}(G)$ of G.

How faster can we locate the target through *multiple* probing steps of *one* vertex?

Min. # of steps = Sequential Location Number $\mathrm{SL}(G)$ of G.

$\sim$ related to a well-known game ☺ ...

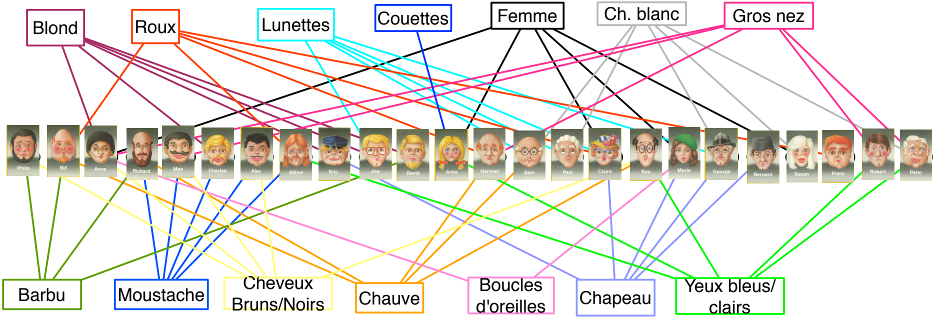
Sequential Locating Game (Seager, 2013)

How faster can we locate the target through *multiple* probing steps of *one* vertex?

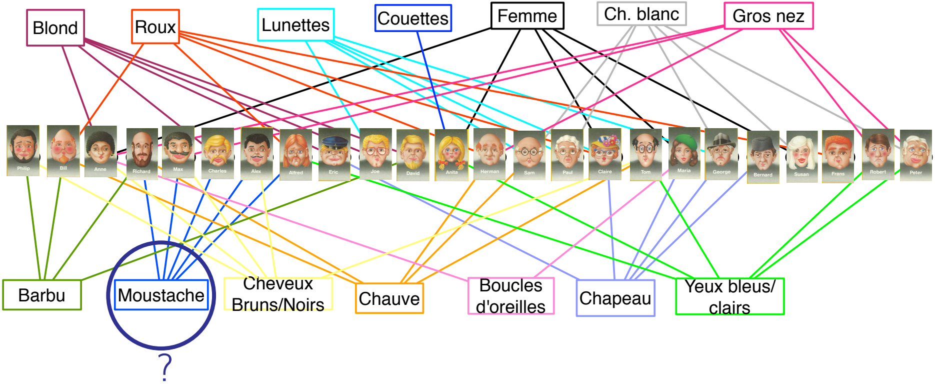
Min. # of steps = Sequential Location Number $\mathrm{SL}(G)$ of G.

$\sim$ related to a well-known game ☺ ...

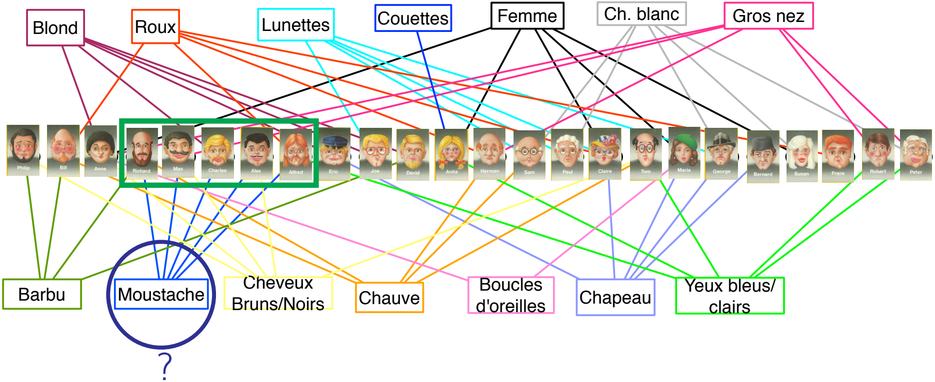
Sequential Locating Game and Guess Who?



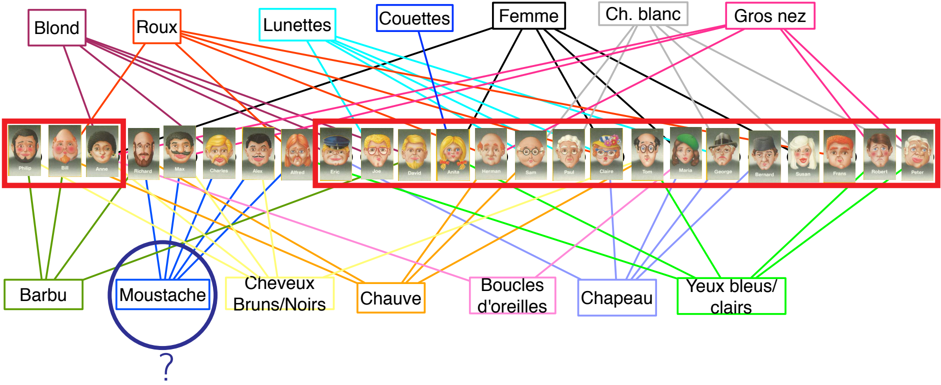
Sequential Locating Game and Guess Who?



Sequential Locating Game and Guess Who?



Sequential Locating Game and Guess Who?



Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

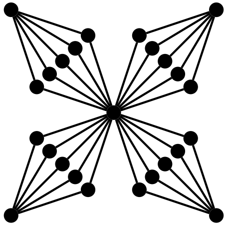
Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:



Question

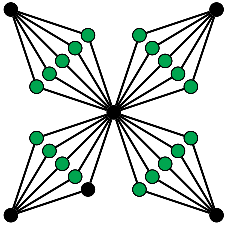
Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

All vertices in green probed



$\mathrm{MD}(G) = 19$

Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

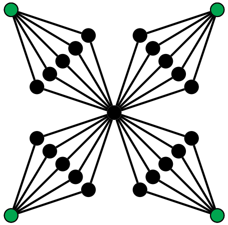
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; FIRST step:



Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

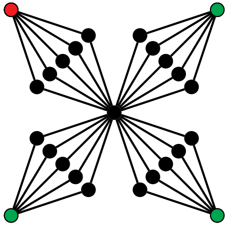
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; FIRST step:



Answer: (0,4,4,4)

Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

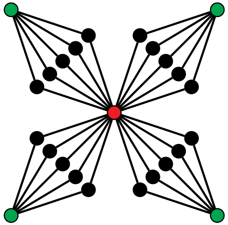
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; FIRST step:



Answer: (2,2,2,2)

Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

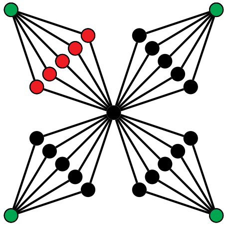
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; FIRST step:



Answer: (1,3,3,3)

Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

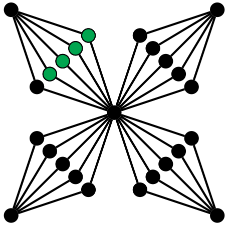
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; SECOND step:



Question

Given G, k, ℓ, is it possible to locate an immobile invisible target in G in at most ℓ steps, by probing at most k vertices each step?

Related:

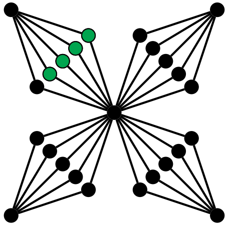
- $\ell = 1$ (Metric Dimension);
- $k = 1$ (Sequential Locating Game);
- Moving target (Bosek et al., 2017).

Note: $\lceil \mathrm{MD}(G)/k \rceil$ steps suffice. But can be far from optimal:

Now sequential probing ($k = 4$)

All vertices in green probed

Location in 2 steps; SECOND step:



$\Rightarrow \mathrm{MD}(G) = 19$. And $\lceil \mathrm{MD}(G)/4 \rceil = 5$, while 2 steps suffice.

Sequential Metric Dimension in trees

$\lambda_k(T)$: min. # of steps to locate t in T (probing at most k vertices each step).

Localisation Problem

Given a tree T, can we locate an immobile invisible target in at most ℓ steps, provided we can probe at most k vertices each step?

$\lambda_k(T)$: min. # of steps to locate t in T (probing at most k vertices each step).

Localisation Problem

Given a tree T, can we locate an immobile invisible target in at most ℓ steps, provided we can probe at most k vertices each step?

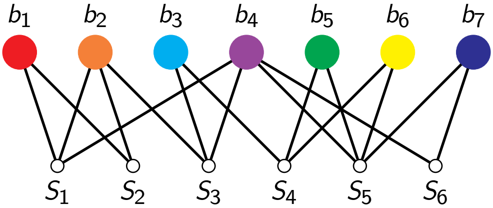
Tree case (k fixed, minimize ℓ):

- Making the appropriate first probing step is NP-complete ☹ ...
- ... but deciding how to probe optimally afterwards is polytime doable ☺ .
- ⇒ Polytime (+1)-approximation algorithm, yielding $\lambda_k(T)$ or $\lambda_k(T)+1$.

Theorem

Determining $\lambda_k(T)$ is NP-hard in trees T.

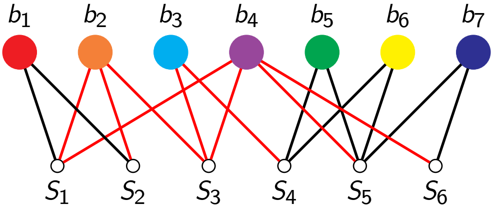
Proof (sketch). Reduction from Hitting Set (given a set $B := \{b_1, \ldots, b_n\}$ and a set $\mathscr{S} := \{S_1, \ldots, S_m\}$ of subsets of B, find a smallest subset of B hitting all S_i's).



Theorem

Determining $\lambda_k(T)$ is NP-hard in trees T.

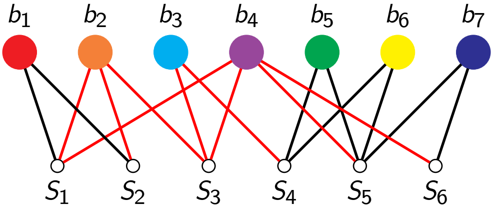
Proof (sketch). Reduction from Hitting Set (given a set $B := \{b_1, \ldots, b_n\}$ and a set $\mathscr{S} := \{S_1, \ldots, S_m\}$ of subsets of B, find a smallest subset of B hitting all S_i's).



Theorem

Determining $\lambda_k(T)$ is NP-hard in trees T.

Proof (sketch). Reduction from Hitting Set (given a set $B := \{b_1, \ldots, b_n\}$ and a set $\mathscr{S} := \{S_1, \ldots, S_m\}$ of subsets of B, find a smallest subset of B hitting all S_i's).



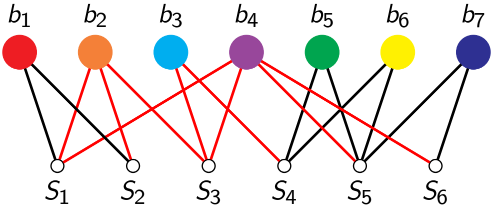
Main ideas:

- Have many big stars in the tree, so that the target has to hide in one such.
- Spend a few steps identifying the hosting big star, and then "peel" its leaves.

Theorem

Determining $\lambda_k(T)$ is NP-hard in trees T.

Proof (sketch). Reduction from Hitting Set (given a set $B := \{b_1, ..., b_n\}$ and a set $\mathscr{S} := \{S_1, ..., S_m\}$ of subsets of B, find a smallest subset of B hitting all S_i's).

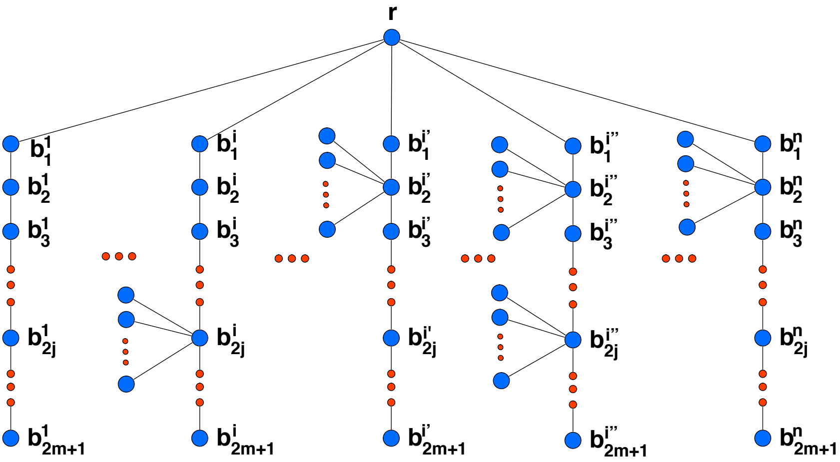


Main ideas:

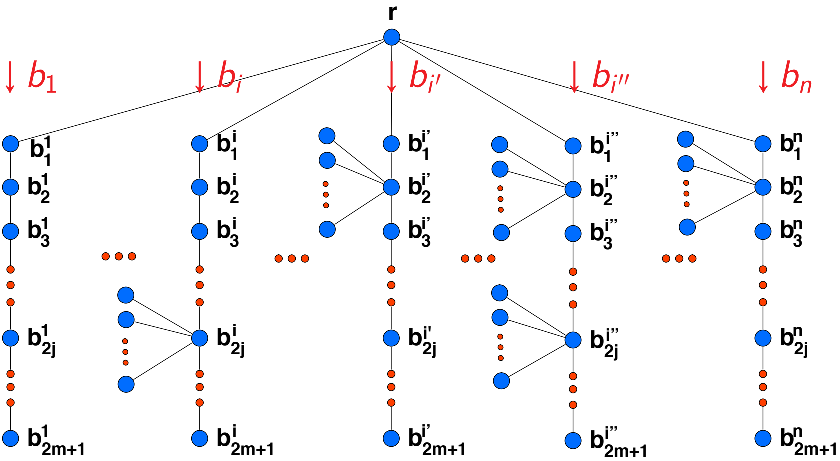
- Have many big stars in the tree, so that the target has to hide in one such.
- Spend a few steps identifying the hosting big star, and then "peel" its leaves.

$\Rightarrow$ Identifying the big star early $\Leftrightarrow$ Hitting set. ■

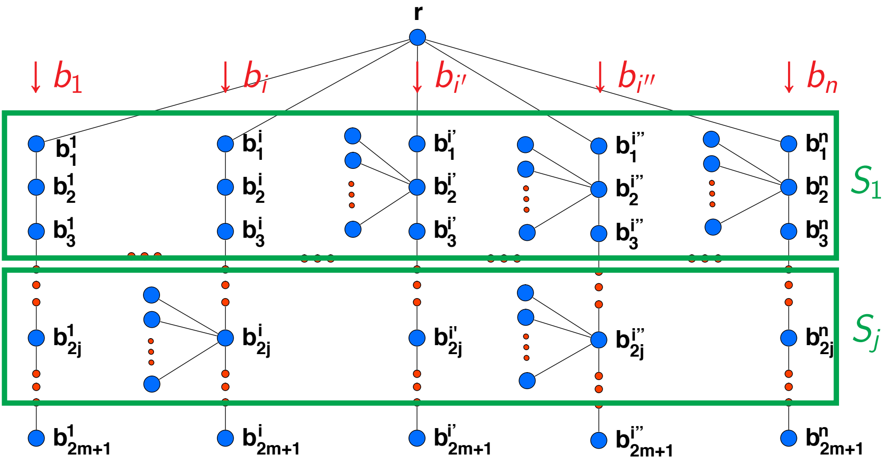
Reduction, illustrated



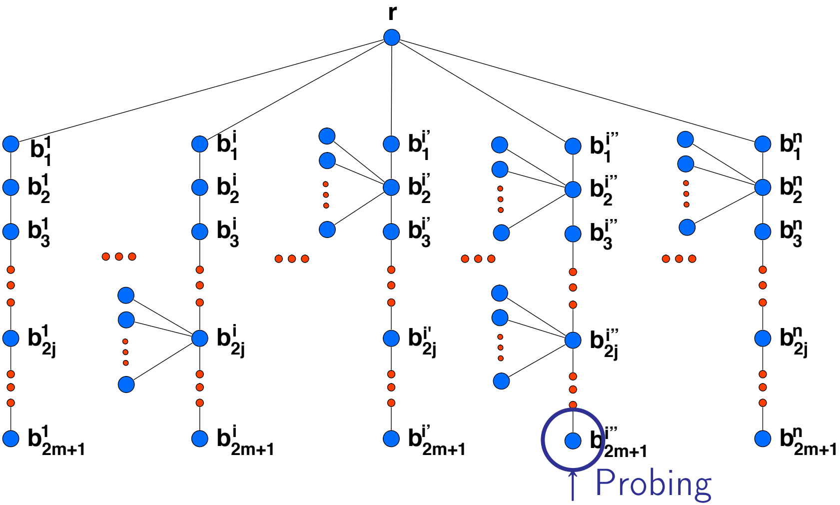
Reduction, illustrated



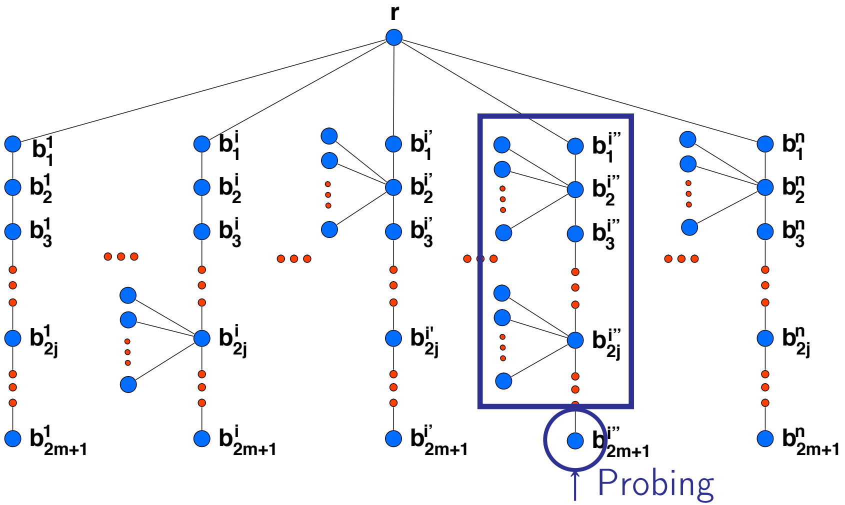
Reduction, illustrated



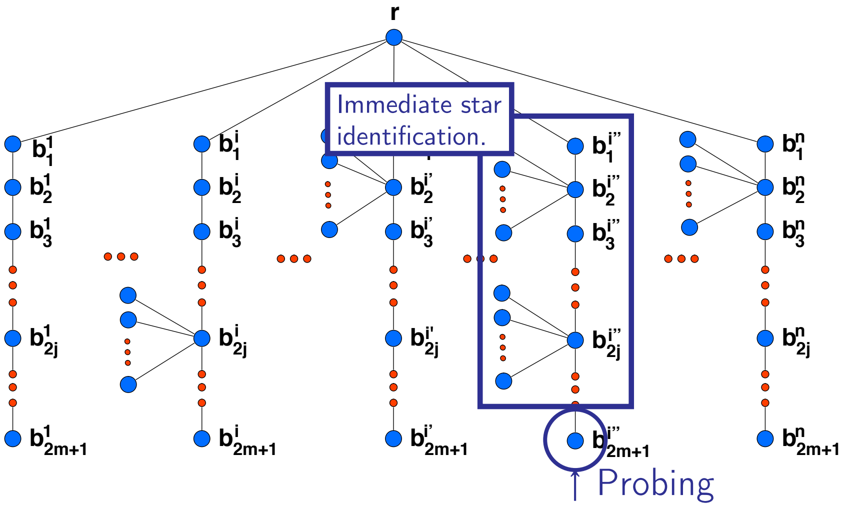
Reduction, illustrated



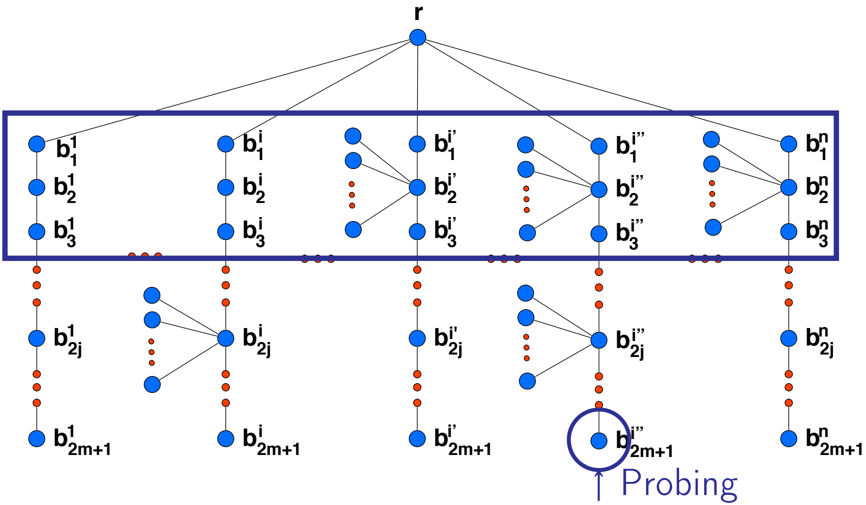
Reduction, illustrated



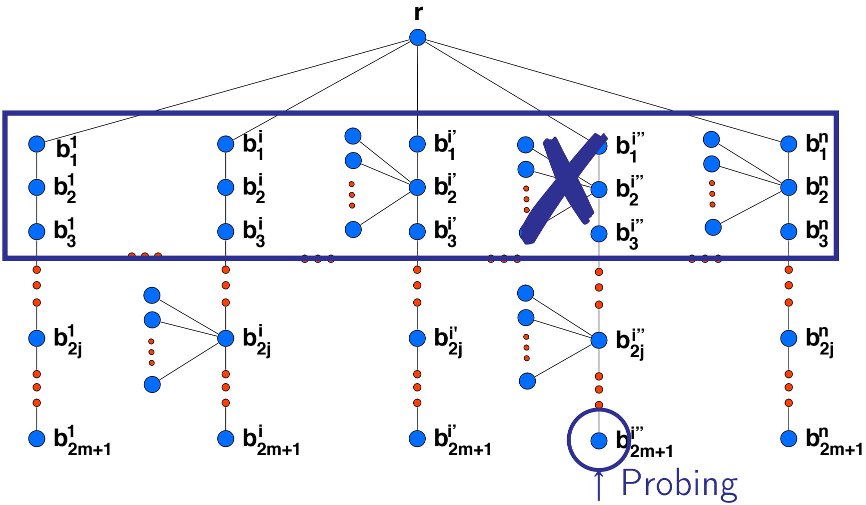
Reduction, illustrated



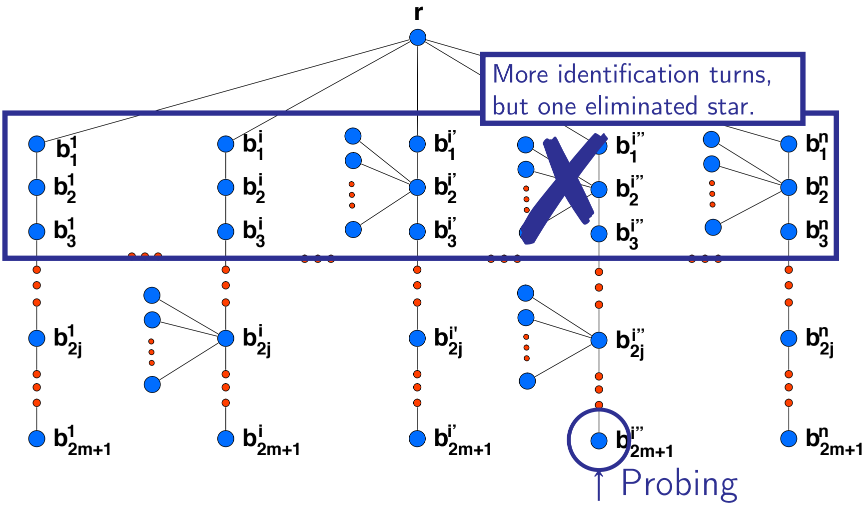
Reduction, illustrated

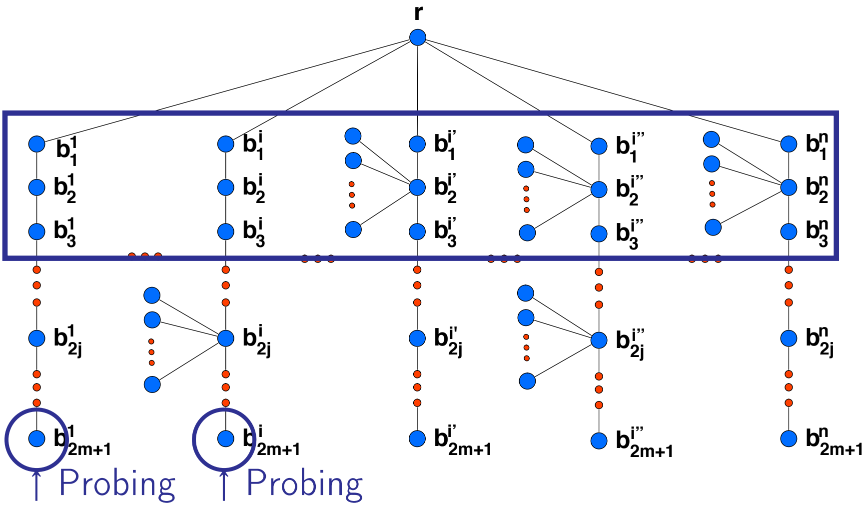


Reduction, illustrated

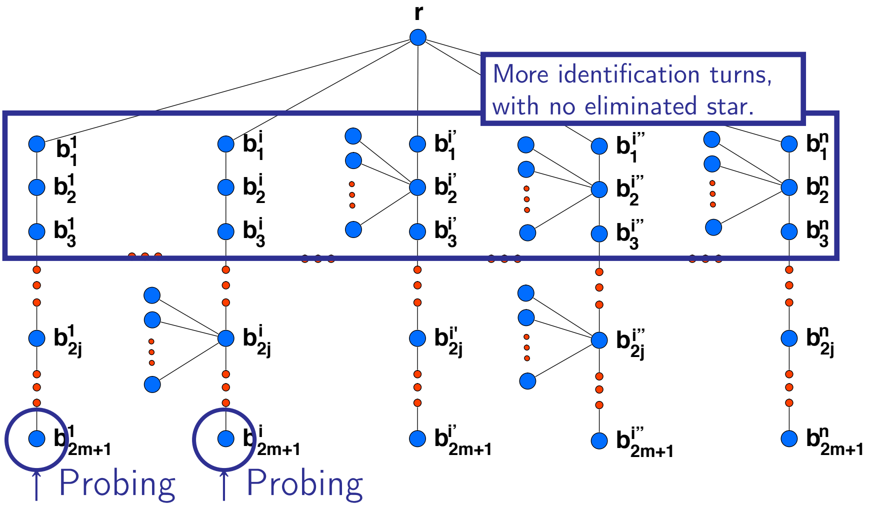


Reduction, illustrated



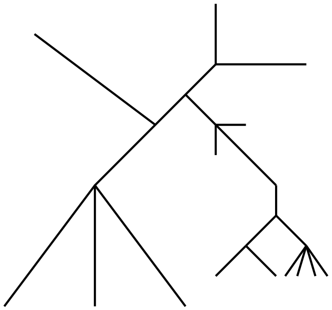

r
b11
b12
b13
b12j
b12m+1
bi1
bi2
bi3
bi2j
bi2m+1
bi'1
bi'2
bi'3
bi'2j
bi'2m+1
bi''1
bi''2
bi''3
bi''2j
bi''2m+1
bn1
bn2
bn3
bn2j
bn2m+1
Probing
Probing

Reduction, illustrated

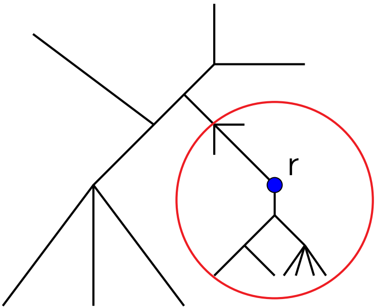


Want: First turn such that all S_i's are hit... ⇒ Hitting set.

First step: Probe any one vertex r...



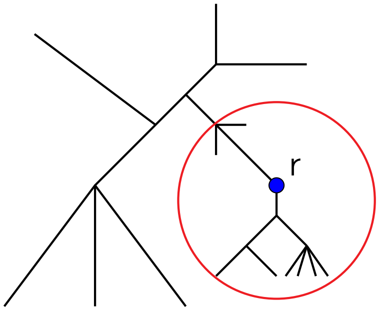
First step: Probe any one vertex r...



First step: Probe any one vertex r...

$\Rightarrow$ for next steps, reduces instance to:

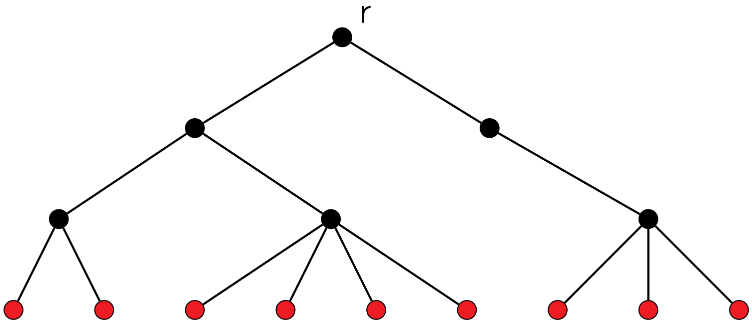
- a tree T' rooted in r;
- all leaves are at same distance from r;
- target is on a leaf.



First step: Probe any one vertex r...

⇒ for next steps, reduces instance to:

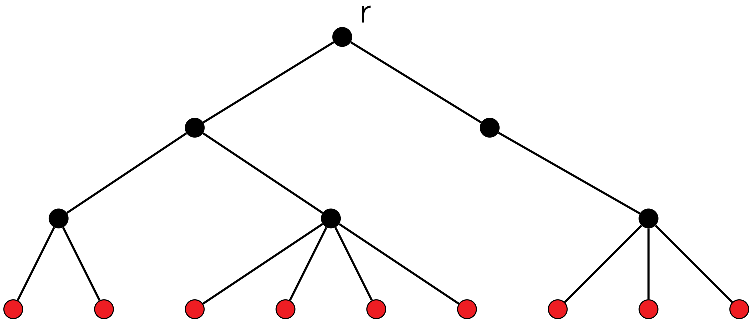
- a tree T' rooted in r;
- all leaves are at same distance from r;
- target is on a leaf.



First step: Probe any one vertex r...

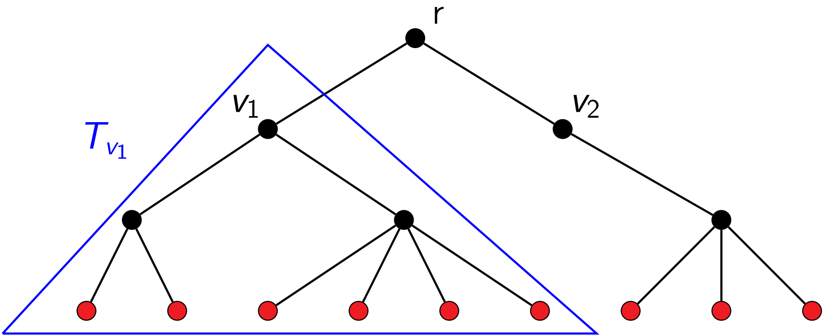
⇒ for next steps, reduces instance to:

- a tree T' rooted in r;
- all leaves are at same distance from r;
- target is on a leaf.

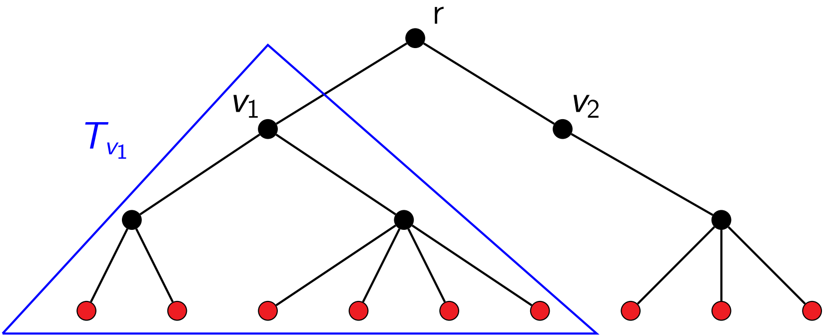


(because playing outside T' is pointless)

T_v: subtree rooted in v of T' rooted in r (v is a child of r).

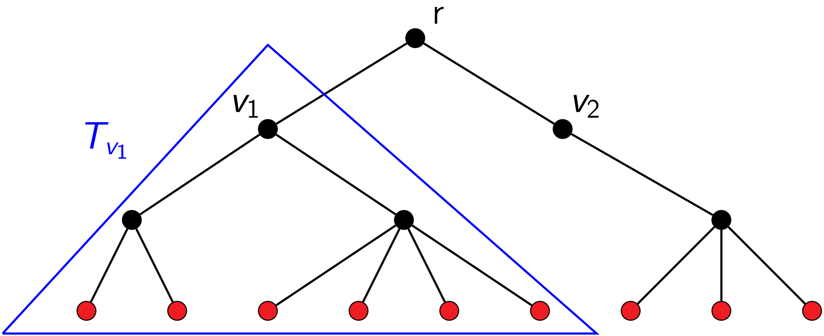


T_v: subtree rooted in v of T' rooted in r (v is a child of r).



Key fact: Probing any vertex of T_v $\Rightarrow$ Know whether T_v hosts the target!

T_v: subtree rooted in v of T' rooted in r (v is a child of r).



Key fact: Probing any vertex of T_v ⇒ Know whether T_v hosts the target!

Crucial question

When playing in T_v for the first time, how many vertices should be probed?

Example: What if T_{v_1} is a big star, while $T_{v_2}, ..., T_{v_{1000}}$ are much smaller stars?

$\Rightarrow$ Trade-off between inspecting **many** of the T_v's, and inspecting **efficiently**.

⇒ Trade-off between inspecting **many** of the T_v's, and inspecting **efficiently**.

Two main parameters for each T_v (assuming target on a leaf):

1. $\lambda_k(T_i)$: min. # of steps needed, probing at most k vertex each step;
2. $\pi_k(T_i)$: min. # of vertices that can be probed **during the first step** in a strategy taking $\lambda_k(T_i)$ steps.

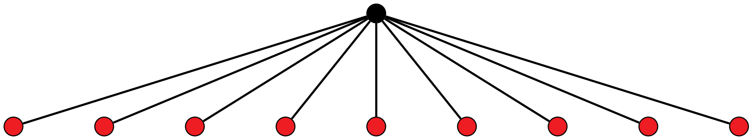
$\sim$ what initial delay does not compromise a quick localisation?

⇒ Trade-off between inspecting **many** of the T_v's, and inspecting **efficiently**.

Two main parameters for each T_v (assuming target on a leaf):

1. $\lambda_k(T_i)$: min. # of steps needed, probing at most k vertex each step;
2. $\pi_k(T_i)$: min. # of vertices that can be probed **during the first step** in a strategy taking $\lambda_k(T_i)$ steps.

$\sim$ what initial delay does not compromise a quick localisation?

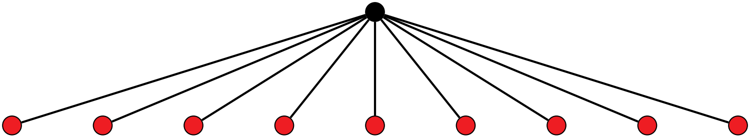


⇒ Trade-off between inspecting **many** of the T_v's, and inspecting **efficiently**.

Two main parameters for each T_v (assuming target on a leaf):

1. $\lambda_k(T_i)$: min. # of steps needed, probing at most k vertex each step;
2. $\pi_k(T_i)$: min. # of vertices that can be probed **during the first step** in a strategy taking $\lambda_k(T_i)$ steps.

$\sim$ what initial delay does not compromise a quick localisation?



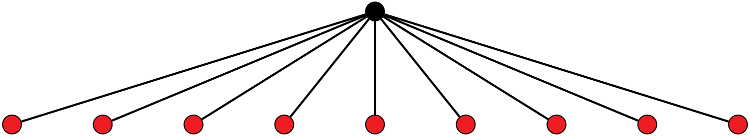
$\lambda_3(T) = 3$

⇒ Trade-off between inspecting **many** of the T_v's, and inspecting **efficiently**.

Two main parameters for each T_v (assuming target on a leaf):

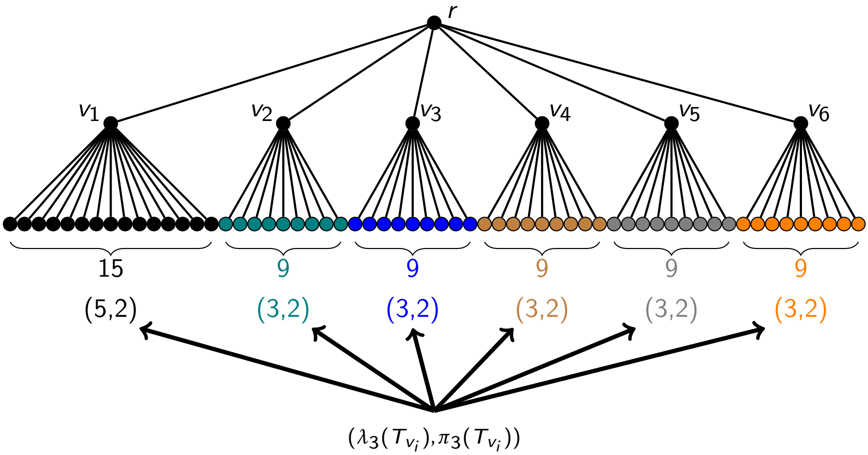
1. $\lambda_k(T_i)$: min. # of steps needed, probing at most k vertex each step;
2. $\pi_k(T_i)$: min. # of vertices that can be probed **during the first step** in a strategy taking $\lambda_k(T_i)$ steps.

$\sim$ what initial delay does not compromise a quick localisation?



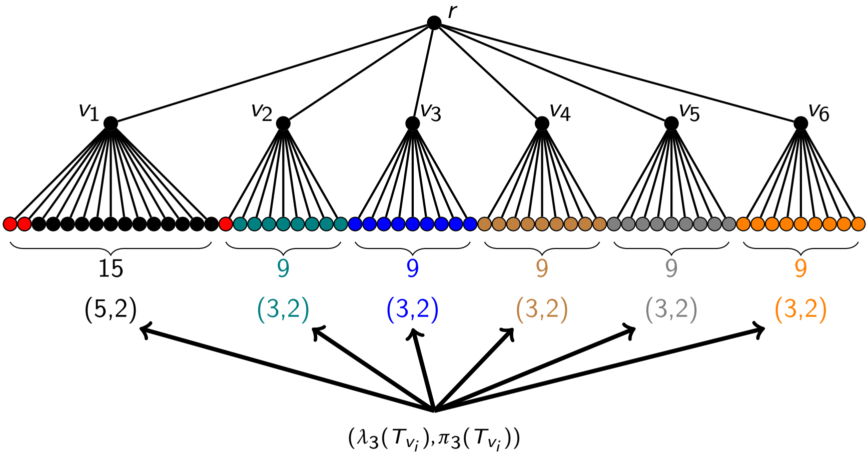
$\lambda_3(T) = 3$ $\pi_3(T) = 2$

In next example, $\lambda_3(T') = 5$.

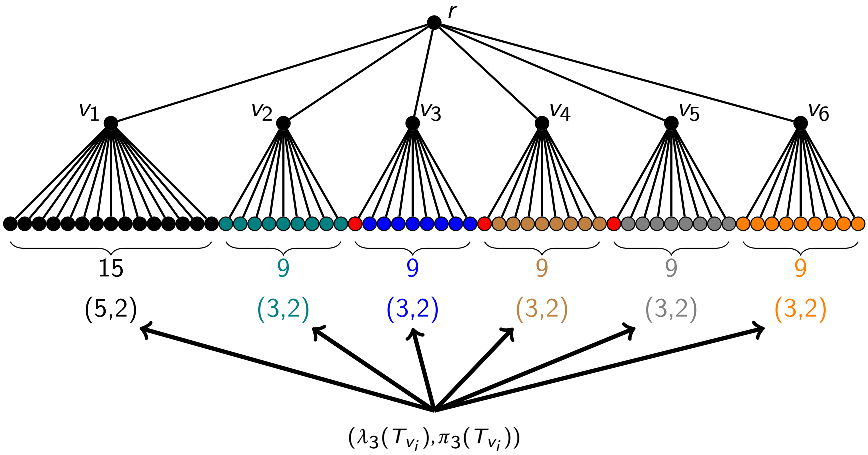


Full example

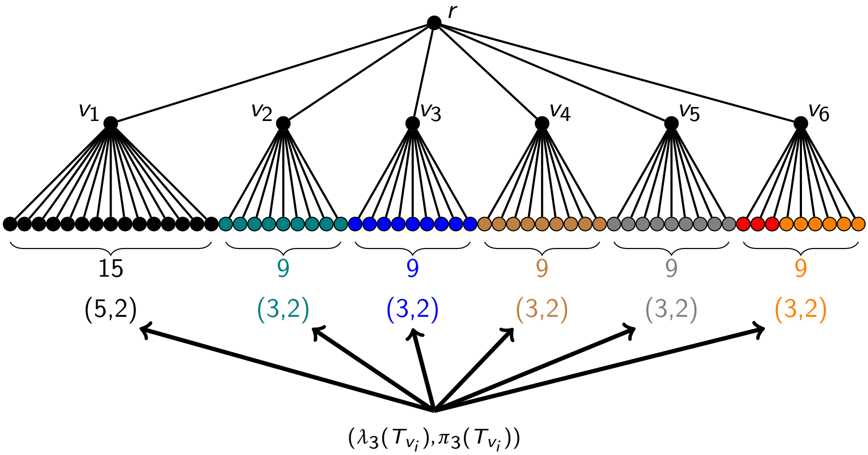
In next example, $\lambda_3(T') = 5$.



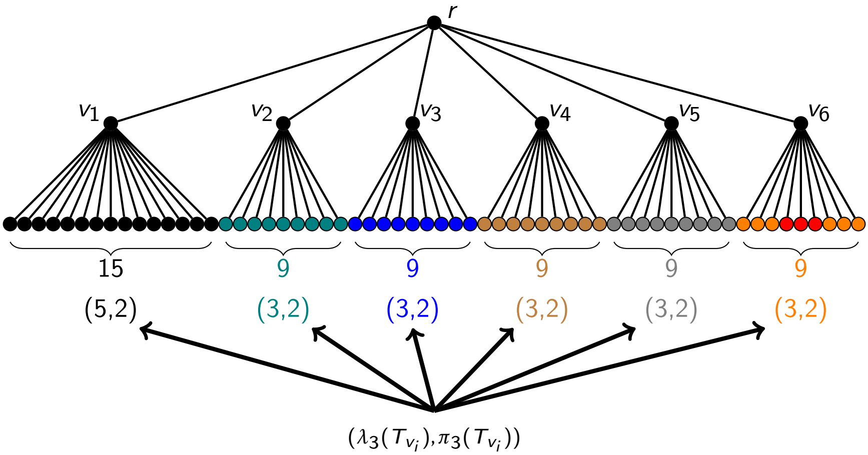
In next example, $\lambda_3(T') = 5$.



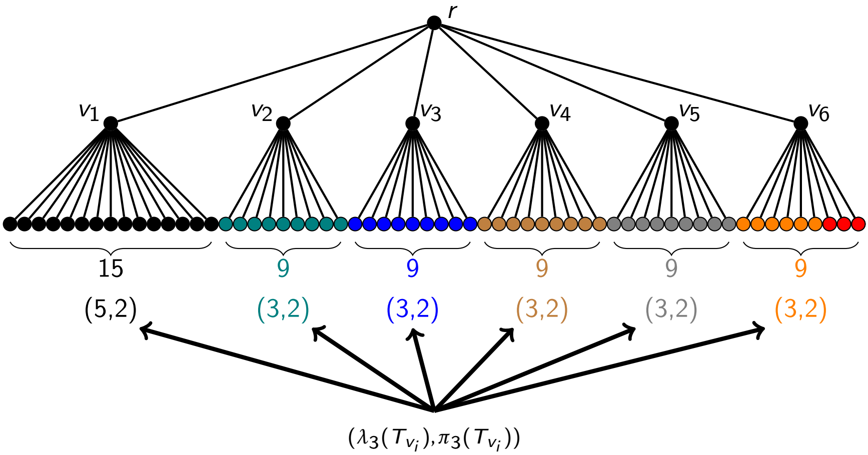
In next example, $\lambda_3(T') = 5$.



In next example, $\lambda_3(T') = 5$.



In next example, $\lambda_3(T') = 5$.



1. Probe *any* vertex r during first step.

1. Probe *any* vertex r during first step.
   - Full "power" of k not used, but...
   - ... the other $k-1$ information might be useless anyway.

1. Probe *any* vertex r during first step.
   - Full "power" of k not used, but...
   - ... the other $k-1$ information might be useless anyway.
2. Next, in the "convenient" context (when having T'):
   - Compute the pair $(\lambda_k(T_v), \pi_k(T_v))$ for each T_v inductively.
   - Deduce that of T'.

$\Rightarrow$ Optimal from here + Polytime algorithm.

1. Probe *any* vertex r during first step.
   - Full "power" of k not used, but...
   - ... the other $k-1$ information might be useless anyway.
2. Next, in the "convenient" context (when having T'):
   - Compute the pair $(\lambda_k(T_v), \pi_k(T_v))$ for each T_v inductively.
   - Deduce that of T'.

   $\Rightarrow$ Optimal from here + Polytime algorithm.

Also:

- For each pair $(\lambda_k(T_v), \pi_k(T_v))$, can retrieve corresponding strategies.
- $\binom{n}{k}$ possible first steps; polynomial when k is a constant.

Conclusion and perspectives

Future work and questions

- Sequential metric dimension of more classes of graphs?

- Sequential metric dimension of more classes of graphs?
- Centroidal dimension of paths?

Future work and questions

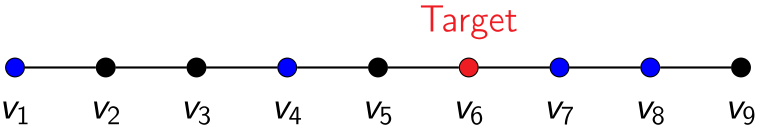
- Sequential metric dimension of more classes of graphs?
- Centroidal dimension of paths?
- All questions for sequential centroidal dimension.

- Sequential metric dimension of more classes of graphs?
- Centroidal dimension of paths?
- All questions for sequential centroidal dimension.

Thank you for your attention!

Locating via relative distances

From probed vertices, get relative distances to the target instead.



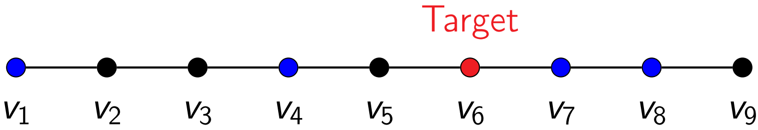
Each $v_i \rightarrow$ Vector of relative distances to the probed vertices:

$$v_1 \rightarrow (\{v_1\},\{v_4\},\{v_7\},\{v_8\}) \quad v_2 \rightarrow (\{v_1\},\{v_4\},\{v_7\},\{v_8\}) \quad v_3 \rightarrow (\{v_4\},\{v_1\},\{v_7\},\{v_8\})$$

$$v_4 \rightarrow (\{v_4\},\{v_1,v_7\},\{v_8\}) \quad v_5 \rightarrow (\{v_5\},\{v_7\},\{v_8\},\{v_1\}) \quad v_6 \rightarrow (\{v_7\},\{v_4,v_8\},\{v_1\})$$

$$v_7 \rightarrow (\{v_7\},\{v_8\},\{v_4\},\{v_1\}) \quad v_8 \rightarrow (\{v_8\},\{v_7\},\{v_4\},\{v_1\})$$

From probed vertices, get relative distances to the target instead.



Each v_i → Vector of relative distances to the probed vertices:

$$v_1 \rightarrow (\{v_1\},\{v_4\},\{v_7\},\{v_8\}) \quad v_2 \rightarrow (\{v_1\},\{v_4\},\{v_7\},\{v_8\}) \quad v_3 \rightarrow (\{v_4\},\{v_1\},\{v_7\},\{v_8\})$$
$$v_4 \rightarrow (\{v_4\},\{v_1,v_7\},\{v_8\}) \quad v_5 \rightarrow (\{v_5\},\{v_7\},\{v_8\},\{v_1\}) \quad v_6 \rightarrow (\{v_7\},\{v_4,v_8\},\{v_1\})$$
$$v_7 \rightarrow (\{v_7\},\{v_8\},\{v_4\},\{v_1\}) \quad v_8 \rightarrow (\{v_8\},\{v_7\},\{v_4\},\{v_1\})$$

Arising notions of:

- Centroidal set (Foucaud, Klasing, Slater, 2014);
- Centroidal dimension (Foucaud, Klasing, Slater, 2014);
- Sequential centroidal dimension (us, 2018+).

Decision problems related to the last notion are NP-complete...