Structural parameterizations of Geodetic Set on directed (acyclic) graphs

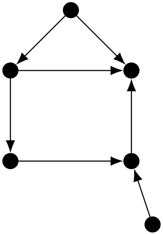
MFCS 2026, Paris, France

Laurent Beaudou[1] Florent Foucaud[1]
Lucas Lorieau[1,2] Prafullkumar Tale[3]

1 LIMOS, Université Clermont Auvergne, France
2 CNRS
3 IISER Pune, India

August 28, 2026

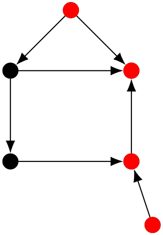
The Geodetic Set Problem



Definition: Geodetic set

A set S of vertices is a **geodetic set** in digraph $D = (V, A)$ if all vertices of $V \setminus S$ are lying on some shortest directed path whose endpoints are vertices of S.

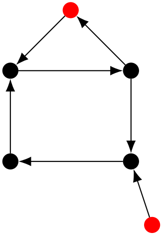
The Geodetic Set Problem



Definition: Geodetic set

A set S of vertices is a **geodetic set** in digraph $D = (V, A)$ if all vertices of $V \setminus S$ are lying on some shortest directed path whose endpoints are vertices of S.

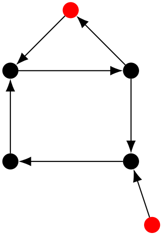
The Geodetic Set Problem



Definition: Geodetic set

A set S of vertices is a **geodetic set** in digraph $D = (V, A)$ if all vertices of $V \setminus S$ are lying on some shortest directed path whose endpoints are vertices of S.

The Geodetic Set Problem



Definition: Geodetic set

A set S of vertices is a **geodetic set** in digraph $D = (V, A)$ if all vertices of $V \setminus S$ are lying on some shortest directed path whose endpoints are vertices of S.

(Directed) Geodetic Set is the problem of finding a geodetic set of minimum size in a given (directed) graph.

Parameterized complexity

Definition: Parameterized tractability

For a problem Π parameterized by k, we say that Π is **Fixed Parameter Tractable** (FPT) with respect to k if there exists an algorithm solving this problem on any instance (x, k) in time $f(k)\mathrm{poly}(|x|)$.

We considered two parameterization for the problem, using:

- the **vertex cover number** (vcn) of the underlying undirected graph
- the combination of:
  - **reachability diameter** (rdiam), an analogue of diameter in digraphs
  - **maximum degree** (Δ) of the underlying undirected graph
  - **solution size** (k)

Previous results : overview

GEODETIC SET has been studied both from the structural and, more recently, algorithmic points of view. Notable results:

- Algorithmic complexity **almost extensively determined** on undirected graphs, even in the parameterized setting
- GEODETIC SET is **NP-hard** on DAGs whose underlying graphs is bipartite, co-bipartite or split and polytime solvable on oriented cactii (Araújo, Arraes 2022)
- GEODETIC SET was proven to be **polytime solvable** on directed trees, **FPT** on oriented graphs with bounded fen and **NP-hard** on oriented DAGs with fvn $= 12$ (Foucaud et al. 2026)

Previous results : parameters on undirected graphs

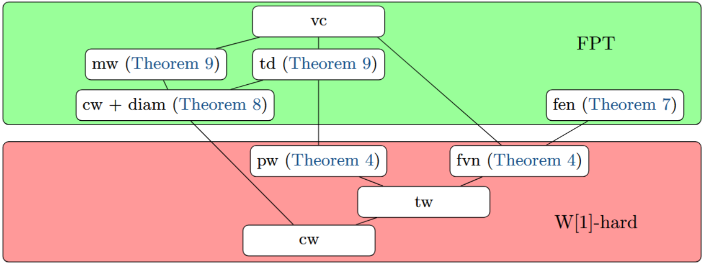


Figure 2: An overview of our results for GEODETIC SET, containing the parameters vertex cover number (vc), modular-width (mw), tree-depth (td), clique-width (cw), diameter (diam), feedback edge number (fen), path-width (pw), feedback vertex number (fvn) and tree-width (tw). An edge between two parameters indicates that the one below is smaller than some function of the other.

Kellerhals and Koana (2022)

Fine-Grained complexity and ETH

The **Exponential Time Hypothesis** (ETH) states that 3-SAT cannot be solved by an algorithm whose running time is $2^{o(n)}$.

This is used to determine **fine-grained complexity** of FPT problems :
Assuming ETH, we can prove lower bounds on the running time of algorithms solving a parameterized problem.

We present two **relatively simple algorithms** (Brute force on a kernel) and prove they are **optimal** under the ETH.

Positive results

Theorem 1: Algorithm for vcn

Directed Geodetic Set admits an algorithm running in time $2^{O(\mathsf{vcn}^2)} \cdot n^{O(1)}$. Moreover, assuming the ETH, it admits no algorithm running in time $2^{o(\mathsf{vcn}^2)} \cdot n^{O(1)}$, even on DAGs.

Theorem 2: Algorithm for rdiam $+ \Delta + k$

Directed Geodetic Set admits a kernel of size $(k\Delta)^{O(\mathsf{rdiam})}$, and an algorithm running in time $(k\Delta)^{O(\mathsf{rdiam} \cdot k)} \cdot n^{O(1)}$. Moreover, assuming the ETH, it admits no algorithm running in time $(k\Delta)^{o(\mathsf{rdiam} \cdot k)} \cdot n^{O(1)}$, even on DAGs with $\mathsf{rdiam} = 2$.

Hardness result

Theorem 3: Hardness for $k + \Delta$

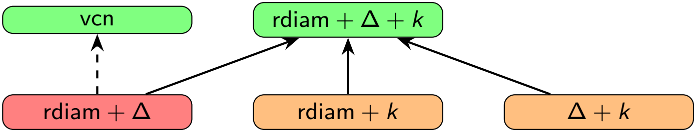
Directed Geodetic Set remains W[2]-hard for solution size k, even on DAGs with maximum degree 3.

Theorem 4: Hardness for $\Delta + \mathrm{rdiam}$

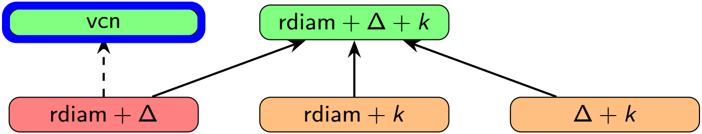
Directed Geodetic Set remains NP-complete on DAGs with maximum degree 3 and reachability diameter 14.

N.B.: Directed Geodetic Set is known to be W[2]-hard parameterized by the solution size on DAGs with $\mathrm{rdiam} = 3$ (Araújo, Arraes 2022).

Our results: a summary



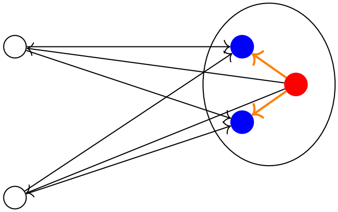
Our results: a summary



A kernel based on twins and shortcuts

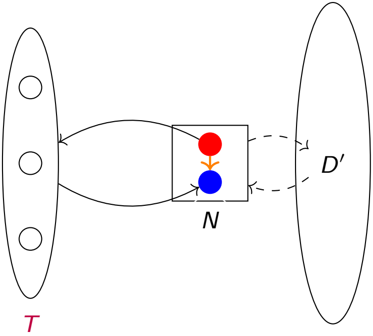
Two vertices are twins if they share both their (open) incoming and outgoing neighborhood.

The neighborhood S of a vertex v is shortcutting if v is **transitive** with respect to S.



A kernel based on twins and shortcuts

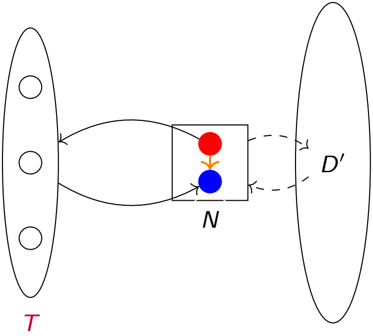
Let T be a set of twins whose neighborhood N is shortcutting.



A kernel based on twins and shortcuts

Let T be a set of twins whose neighborhood N is shortcutting.

Vertices of T cannot be covered by vertices of N or D' $\rightarrow$ they belong to any geodetic set.

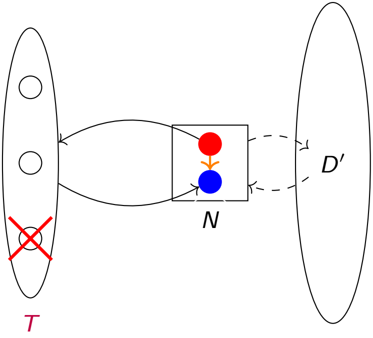


A kernel based on twins and shortcuts

Let T be a set of twins whose neighborhood N is shortcutting.

Vertices of T cannot be covered by vertices of N or D' $\rightarrow$ they belong to any geodetic set.

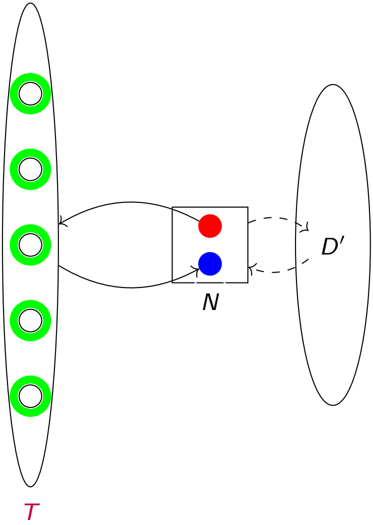
Rule 1: If $|T| \geq 3$ and the neighborhood of T is shortcutting, delete one vertex from T and decrease k by one.



A kernel based on twins and shortcuts

Let T be a set of twins whose neighborhood N is **not** shortcutting.

Claim: Any optimal geodetic set contains at most 4 vertices of T.

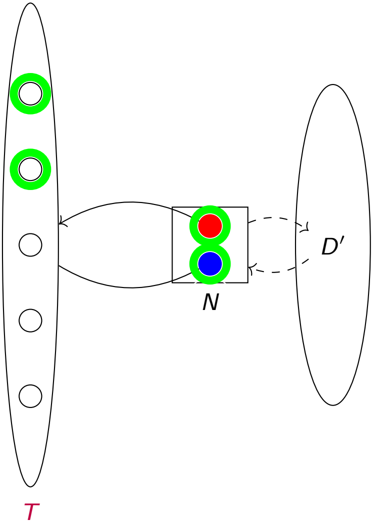


A kernel based on twins and shortcuts

Let T be a set of twins whose neighborhood N is **not** shortcutting.

Claim: Any optimal geodetic set contains at most 4 vertices of T.

Rule 2: If $|T| \geq 6$, and the neighborhood of T is not shortcutting, then delete one vertex from T.



The algorithm

Applying Rules 1 and 2 on the input (D, k) yields an equivalent instance (D', k'), such that the size of D' is $2^{O(\mathsf{vcn})}$.

Furthermore, we can compute easily an optimal geodetic set in time $n^{O(\mathsf{vcn})}$:

- Compute a **minimal vertex cover** X in FPT time.
- Compute **mandatory vertices** S
- $X + S$ is a solution: **brute force** over all sets of size $\leq |X| = \mathsf{vcn}$ in $V \setminus S$

Apply this algorithm on D', whose size is $2^{O(\mathsf{vcn})}$. thus, the total running time is $2^{O(\mathsf{vcn}^2)} \cdot n^{O(1)}$.

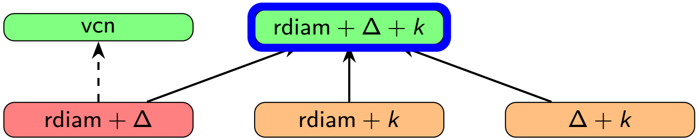
Main ingredients of the ETH reduction

3-Partition 3-SAT: Variables are partitioned into 3 sets of same size N, and a clause contains at most one variable from each of those sets. Under the ETH, no algorithm solves it in time $2^{o(N)}$ [Lampis et al., 2025].

Encoding subsets of variables: Buckets of $\sqrt{N}$ variables are defined, and subsets of a given bucket are encoded using $2^{\sqrt{N}}$ vertices.

Encoding adjacency through distances: Shortcuts are setup so that no clause vertex is covered by an irrelevant bucket assignment.

Recall of results



A brute force on the whole graph

- Decompose D into maximal strongly connected components. The **component graph** obtained is a DAG. The number of sources in this DAG is at most k, since **each source** component contains a **source vertex** of D.
- The number of vertices **reachable** from a given vertex v is $\sum_{i=o}^{\text{rdiam}} \Delta^i \leq \Delta^{\text{rdiam}+1}$.
- Thus, the **total number** of vertices in D is $k \cdot \Delta^{\text{rdiam}+1}$.
- **Brute force** on the subsets of size at most k gives a running time of $(k\Delta)^{O(k\text{rdiam})}$.

Reduction from $\ell \times \ell$-Independent Set

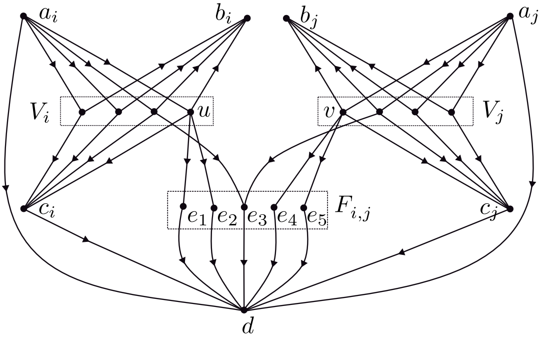
Instance: A graph H partitioned into ℓ independent sets $V_1, \ldots, V_\ell$, each of size ℓ.

Goal: Determine whether there exist an independent set X such that $|X \cap V_i| = 1$ for $1 \leq i \leq \ell$

Theorem: [Cygan et al., *Parameterized Algorithms*]

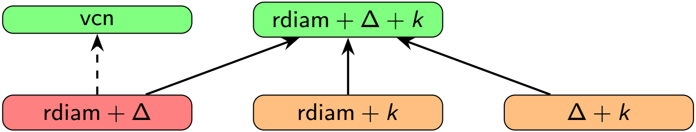
Unless the ETH fails, no algorithm solving exactly $\ell \times \ell$-Independent Set run in time $\ell^{o(\ell)}$.

Reduction overview

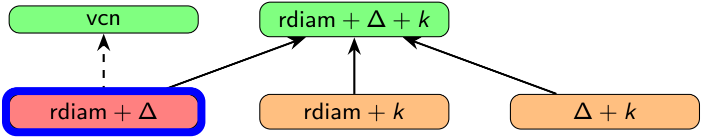


In this construction, we get k and Δ are **bounded** (by some $f(\ell)$ and rdiam $= 2$.
$\rightarrow$ An algorithm running in time $(k\Delta)^{o(k\mathrm{rdiam})}$ would yield an algorithm running in time $\ell^{o(\ell)}$ for $\ell \times \ell$-Independent Set.

Recall of results

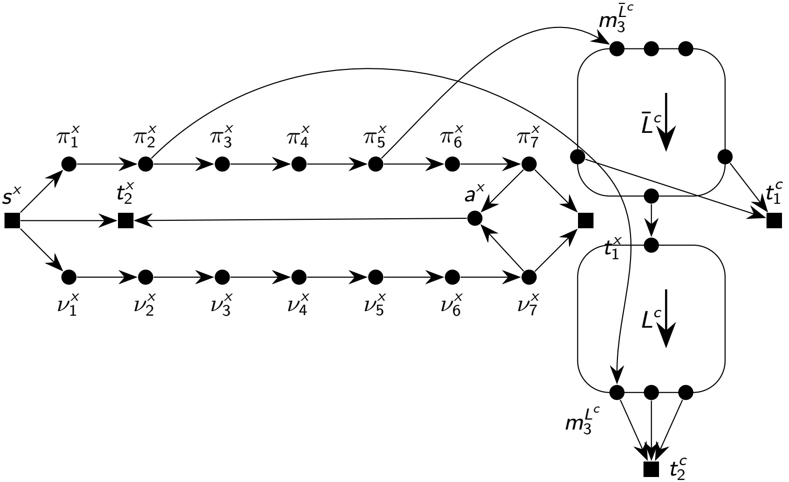


Recall of results



How to encode adjacency with a small degree?

Main idea : Use (subdivised) binary trees (represented by the rounded square gadgets)



Conclusion and open work

- We gave algorithms based on kernels for two parameterized setups, and proved that they were optimal under the ETH.
- These parameters captures a type of **sparsity** of the instance. Previous work also studied such sparse structure on "tree-like" digraphs.
- One could study **other parameters** of the same kind, or look for **other kernels**, especially when the problem is known to be FPT (no polynomial-size kernel known).

Thanks for your attention!