A single-exponential FPT algorithm for Distance-Hereditary Vertex Deletion

O-joung Kwon

Institute for Computer Science and Control,
Hungarian Academy of Sciences in Budapest, Hungary

Joint work with
Eduard Eiben (TU Wien, Austria)
Robert Ganian (TU Wien, Austria)

Seminar at MTA SZTAKI
5th, May, 2015

Distance-hereditary graphs

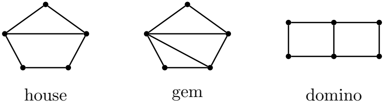
A graph G is **distance-hereditary** if for every connected induced subgraph H of G and $u, v \in V(H)$, the distance between u and v in H is the same as the distance in G.
(First introduced by Howorka, 77)

Distance-hereditary graphs

A graph G is **distance-hereditary** if for every connected induced subgraph H of G and $u, v \in V(H)$, the distance between u and v in H is the same as the distance in G.
(First introduced by Howorka, 77)

Equivalent characterizations (Bandelt, Mulder, 82):

- Every induced path is a shortest path.
- Every cycle of length at least 5 contains a pair of crossing chords.
- Graphs can be constructed from a vertex by a sequence of adding twins or adding leaf vertices.
- Having no obstructions : Induced cycles of length at least 5 +

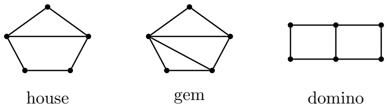


Distance-hereditary graphs

A graph G is **distance-hereditary** if for every connected induced subgraph H of G and $u, v \in V(H)$, the distance between u and v in H is the same as the distance in G.
(First introduced by Howorka, 77)

Equivalent characterizations (Bandelt, Mulder, 82):

- Every induced path is a shortest path.
- Every cycle of length at least 5 contains a pair of crossing chords.
- Graphs can be constructed from a vertex by a sequence of adding twins or adding leaf vertices.
- Having no obstructions : Induced cycles of length at least 5 +



- Rank-width $\leqslant 1$ (Oum, 05).

Distance-Hereditary Vertex Deletion

Input : A graph G, an integer k
Parameter : k
Question : $\exists$? $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G - S$ is distance-hereditary?

This can be seen as a counterpart of Feedback Vertex Set.

Rank-width w Vertex Deletion

Input : A graph G, an integer k
Parameter : k
Question : $\exists$? $S \subseteq V(G)$ with $|S| \leqslant k$ such that $G - S$ has rank-width at most w?

Distance-Hereditary Vertex Deletion

Input : A graph G, an integer k
Parameter : k
Question : $\exists?\ S \subseteq V(G)$ with $|S| \leqslant k$ such that $G - S$ is distance-hereditary?

This can be seen as a counterpart of Feedback Vertex Set.

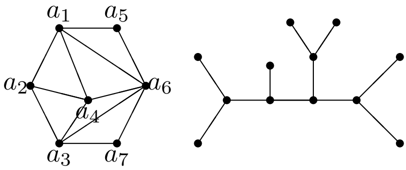
Rank-width w Vertex Deletion

Input : A graph G, an integer k
Parameter : k
Question : $\exists?\ S \subseteq V(G)$ with $|S| \leqslant k$ such that $G - S$ has rank-width at most w?

So, what is rank-width?

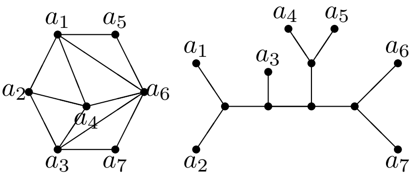
What is rank-width?

- A rank-decomposition (T, L) of G consists of a subcubic tree T, and a bijective function L from $V(G)$ to the leaves of T.



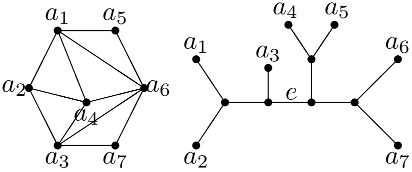
What is rank-width?

- A rank-decomposition (T, L) of G consists of a subcubic tree T, and a bijective function L from $V(G)$ to the leaves of T.



What is rank-width?

- A rank-decomposition (T, L) of G consists of a subcubic tree T, and a bijective function L from $V(G)$ to the leaves of T.

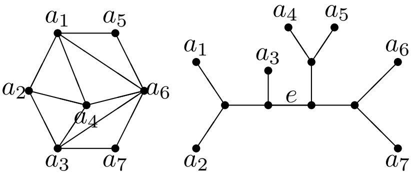


- Width of an edge of T: the rank of the matrix with the partition induced by the edge.

$$\text{Width of } e = \text{rank}\left(\begin{array}{c} \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{array}\begin{array}{c} \begin{array}{ccc} a_1 & a_2 & a_3 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}\right) = 3$$

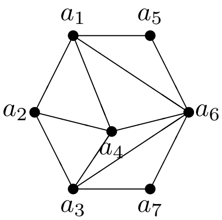
What is rank-width?

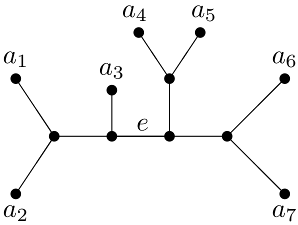
- A rank-decomposition (T, L) of G consists of a subcubic tree T, and a bijective function L from $V(G)$ to the leaves of T.



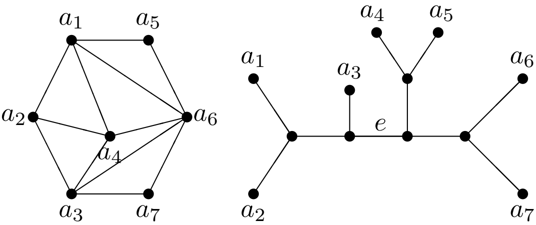
- Width of an edge of T: the rank of the matrix with the partition induced by the edge.

$$\text{Width of } e = \text{rank}\left(\begin{array}{cccc} & a_1 & a_2 & a_3 \\ a_4 & 1 & 1 & 1 \\ a_5 & 1 & 0 & 0 \\ a_6 & 1 & 0 & 1 \\ a_7 & 0 & 0 & 1 \end{array}\right) = 3$$





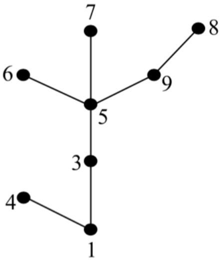
- Width of (T, L): max width over all edges in T
- **Rank-width** of G: min width over all rank-decompositions of G (Oum and Seymour 04; introduced for approximating clique-width)



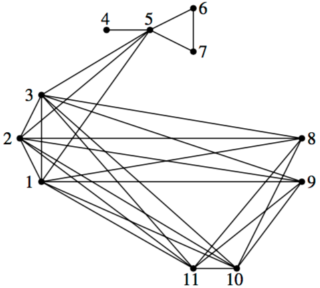
- Width of (T, L): max width over all edges in T
- **Rank-width** of G: min width over all rank-decompositions of G (Oum and Seymour 04; introduced for approximating clique-width)

compare with clique-width : $\mathrm{rw} \leqslant \mathrm{cw} \leqslant 2^{\mathrm{rw}-1} + 1$.

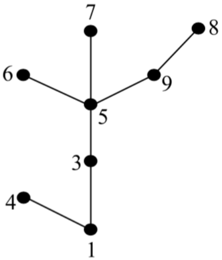
- Clique-width : easier to design a dynamic programming algorithm, no direct FPT approximation algorithm, decomposition is always rooted.
- Rank-width : (3k+1)-FPT approximation algorithm, decomposition is not rooted, fit with 'vertex-minor' (every bipartite graph with large rank-width contain a grid-like vertex-minor...)



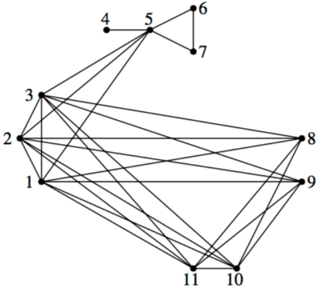
Tree (tree-width $\leqslant 1$)



Distance-hereditary graph (rank-width $\leqslant 1$)



Tree (tree-width $\leqslant 1$)



Distance-hereditary graph (rank-width $\leqslant 1$)

Graphs of bounded rank-width contain dense graphs but also extend useful properties of graphs of bounded tree-width :

Courcelle, Makowsky, Rotics, 00

Every MSO_1 properties can be checked in cubic time on graphs of bounded rank-width.

Distance to graphs of bounded tree-width

graph class $\mathcal{C}$	problem
Edgeless graphs	VERTEX COVER FPT $\mathcal{O}^*(2^k)$ Kernel $2k$ vertices
Forests (graphs of tree-width $\leqslant 1$)	FEEDBACK VERTEX SET FPT $\mathcal{O}^*(3.618^k)$ [Kociumaka et al. 14] Kernel $4k^2$ vertices [Thomassé 10]
Series-parallel graphs (graphs of tree-width $\leqslant 2$)	K_4-MINOR COVER FPT $\mathcal{O}^*(c^k)$ [Kim et al. 15]
Graphs of tree-width $\leqslant w$	TREE-WIDTH w-VERTEX DELETION FPT $\mathcal{O}^*(c^k)$ (Non-uniform) Polynomial kernel [Fomin et al.12 /Kim et al. 13]

Distance to graphs of bounded rank-width?

graph class $\mathcal{C}$	problem
Disjoint union of Complete graphs	Cluster Vertex Deletion easy FPT $\mathcal{O}^*(3^k)$ (obstruction is P_3)
Block graphs	Block Graph Vertex Deletion
Thread graphs (linear rank-width $\leqslant 1$)	Linear rank-width-1 Vertex Deletion
Distance-hereditary graphs (graphs of rank-width $\leqslant 1$)	Distance-hereditary Vertex Deletion
Graphs of rank-width $\leqslant w$	Rank-width w-Vertex Deletion

Distance to graphs of bounded rank-width?

graph class $\mathcal{C}$	problem
Disjoint union of Complete graphs	CLUSTER VERTEX DELETION easy FPT $\mathcal{O}^*(3^k)$ (obstruction is P_3)
Block graphs	BLOCK GRAPH VERTEX DELETION FPT $\mathcal{O}^*(4^k)$ Kernel $\mathcal{O}(k^4)$ vertices [Kim, K 15 / improved by Agrawal et al. 16]
Thread graphs (linear rank-width $\leqslant 1$)	LINEAR RANK-WIDTH-1 VERTEX DELETION
Distance-hereditary graphs (graphs of rank-width $\leqslant 1$)	DISTANCE-HEREDITARY VERTEX DELETION
Graphs of rank-width $\leqslant w$	RANK-WIDTH w-VERTEX DELETION

Distance to graphs of bounded rank-width?

graph class $\mathcal{C}$	problem
Disjoint union of Complete graphs	CLUSTER VERTEX DELETION easy FPT $\mathcal{O}^*(3^k)$ (obstruction is P_3)
Block graphs	BLOCK GRAPH VERTEX DELETION FPT $\mathcal{O}^*(4^k)$ Kernel $\mathcal{O}(k^4)$ vertices [Kim, K 15 / improved by Agrawal et al. 16]
Thread graphs (linear rank-width $\leqslant 1$)	LINEAR RANK-WIDTH-1 VERTEX DELETION FPT $\mathcal{O}^*(8^k)$ Kernel $\mathcal{O}(k^{33})$ vertices [Kanté, Kim, K, Paul 15]
Distance-hereditary graphs (graphs of rank-width $\leqslant 1$)	DISTANCE-HEREDITARY VERTEX DELETION
Graphs of rank-width $\leqslant w$	RANK-WIDTH w-VERTEX DELETION

Distance to graphs of bounded rank-width?

graph class $\mathcal{C}$	problem
Disjoint union of Complete graphs	Cluster Vertex Deletion easy FPT $\mathcal{O}^*(3^k)$ (obstruction is P_3)
Block graphs	Block Graph Vertex Deletion FPT $\mathcal{O}^*(4^k)$ Kernel $\mathcal{O}(k^4)$ vertices [Kim, K 15 / improved by Agrawal et al. 16]
Thread graphs (linear rank-width $\leqslant 1$)	Linear rank-width-1 Vertex Deletion FPT $\mathcal{O}^*(8^k)$ Kernel $\mathcal{O}(k^{33})$ vertices [Kanté, Kim, K, Paul 15]
Distance-hereditary graphs (graphs of rank-width $\leqslant 1$)	Distance-hereditary Vertex Deletion FPT $\mathcal{O}^*(2^{\mathcal{O}(k \log k)})$ [Kim, K, manuscript]
Graphs of rank-width $\leqslant w$	Rank-width w-Vertex Deletion FPT by meta-theorem

OPEN: Rank-width w-Vertex Deletion can be solved in $\mathcal{O}^*(c^k)$-time for some constant c?

Eiben, Ganian, K, 16

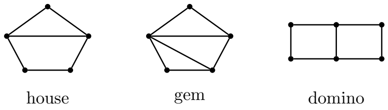
DISTANCE-HEREDITARY VERTEX DELETION can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

Eiben, Ganian, K, 16

DISTANCE-HEREDITARY VERTEX DELETION can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

Recall :

- having no obstructions : Induced cycles of length at least 5 $+$

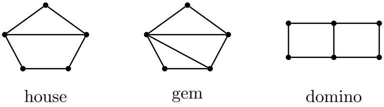


Eiben, Ganian, K, 16

Distance-Hereditary Vertex Deletion can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

Recall :

- having no obstructions : Induced cycles of length at least 5 +



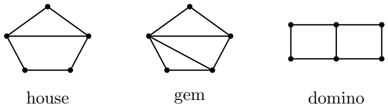
– Hitting induced cycles of length at least 5 is W[2]-hard. (Heggernes et al. 13)

Eiben, Ganian, K, 16

Distance-Hereditary Vertex Deletion can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

Recall :

- having no obstructions : Induced cycles of length at least 5 +



– Hitting induced cycles of length at least 5 is W[2]-hard. (Heggernes et al. 13)

Important property

Let G be a graph obtained from an induced path of length at least 3 by adding a vertex v adjacent to its end vertices. Then G has a DH obstruction containing v.

Eiben, Ganian, K, 16

DISTANCE-HEREDITARY VERTEX DELETION can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

Eiben, Ganian, K, 16

DISTANCE-HEREDITARY VERTEX DELETION can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

(1) **Iterative Compression**. Reduce to DISJOINT DISTANCE-HEREDITARY VERTEX DELETION.

DISJOINT DISTANCE-HEREDITARY VERTEX DELETION

Input : A graph G, an integer k, and $S \subseteq V(G)$ with $|S| \leqslant k+1$ such that $G - S$ is distance-hereditary.

Parameter : k

Question : Is there $Q \subseteq V(G) \setminus S$ with $|Q| \leqslant k$ such that $G - Q$ is distance-hereditary?

Eiben, Ganian, K, 16

DISTANCE-HEREDITARY VERTEX DELETION can be solved in time $37^k \cdot n^{\mathcal{O}(1)}$.

(1) **Iterative Compression**. Reduce to DISJOINT DISTANCE-HEREDITARY VERTEX DELETION.

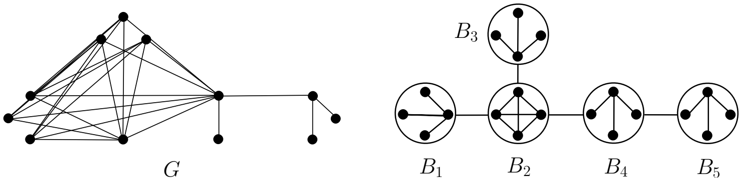
DISJOINT DISTANCE-HEREDITARY VERTEX DELETION

Input : A graph G, an integer k, and $S \subseteq V(G)$ with $|S| \leqslant k+1$ such that $G - S$ is distance-hereditary.
Parameter : k
Question : Is there $Q \subseteq V(G) \setminus S$ with $|Q| \leqslant k$ such that $G - Q$ is distance-hereditary?

(2) **Branching Rules**. Apply two branching rules $\rightarrow$ G has no small obstructions and furthermore that certain connectivity conditions hold on $G[S]$.

(3) **Simplification of Split Decomposition**. We compute the split decomposition of $G - S$ and exploit the properties of our instance G guaranteed by branching to prune the decomposition.

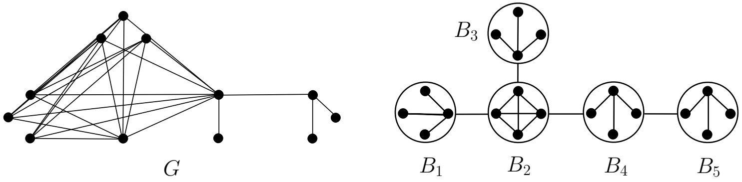
Split Decompositions



Two types of vertices : original vertices, marked vertices.
Two types of edges : marked edges, unmarked edges.

Two original vertices are adjacent if and only if they are linked by an alternating path.
(unmarked–marked–unmarked– $\cdots$ –unmarked)

Split Decompositions



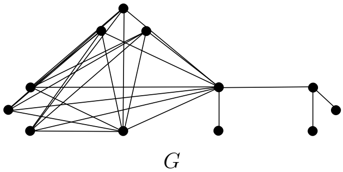
Cunningham, 82

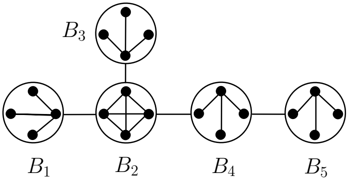
This decomposition is unique if

- every bag is a complete graph or a star or a prime graph,
- we cannot obtain a decomposition with same property by reversing a marked edge.

It is called the canonical split decomposition of a graph.

Split Decompositions





Cunningham, 82

This decomposition is unique if

- every bag is a complete graph or a star or a prime graph,
- we cannot obtain a decomposition with same property by reversing a marked edge.

It is called the canonical split decomposition of a graph.

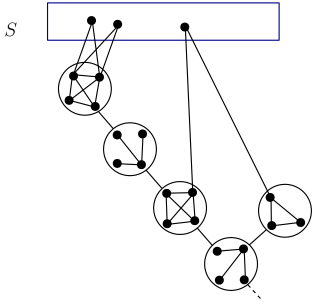
Bouchet, 88

A graph is distance-hereditary if and only if every bag of its canonical split decomposition is either a star or a complete graph.

Now, we are ready ..

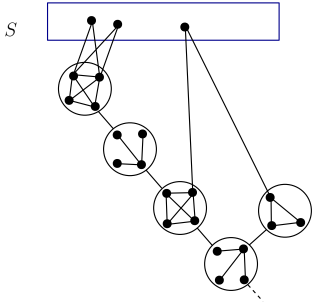
Now, we are ready ..

There is a given solution S and the canonical split decomposition of $G - S$.



Now, we are ready ..

There is a given solution S and the canonical split decomposition of $G - S$.



- Branching Rule 1.
  $\forall X \subseteq V(G - S)$ with $|X| \leqslant 5$, if $G[S \cup X]$ is not distance-hereditary, then we remove one of the vertices in X.
- Branching Rule 2.
  $\forall X \subseteq V(G - S)$ with $|X| \leqslant 5$ such that $G[X]$ is connected and adding X to S decreases the number of components in $G[S]$, then we either remove one of the vertices in X, or put all of them into S.

And nine reduction rules.. what?

And nine reduction rules.. what?

Main ingredient : bad vertex for an induced cycle of length at least 5.

- For every induced cycle C of length at least 5 and $v \in V(C)$, a vertex $w \in S$ is a bad vertex for C and v if it is adjacent to the two neighbors of v in C.

And nine reduction rules.. what?

Main ingredient : bad vertex for an induced cycle of length at least 5.

- For every induced cycle C of length at least 5 and $v \in V(C)$, a vertex $w \in S$ is a bad vertex for C and v if it is adjacent to the two neighbors of v in C.
- It is easy to see that $v \in V(G - S)$ can be safely removed if for every induced cycle of length at least 5 containing v, there is a bad vertex for C and v.

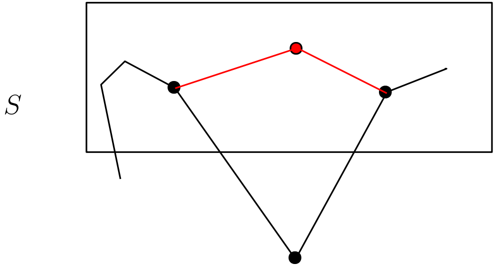
Is it easy to find a bad vertex?

And nine reduction rules.. what?

Main ingredient : bad vertex for an induced cycle of length at least 5.

- For every induced cycle C of length at least 5 and $v \in V(C)$, a vertex $w \in S$ is a bad vertex for C and v if it is adjacent to the two neighbors of v in C.
- It is easy to see that $v \in V(G - S)$ can be safely removed if for every induced cycle of length at least 5 containing v, there is a bad vertex for C and v.

Is it easy to find a bad vertex?



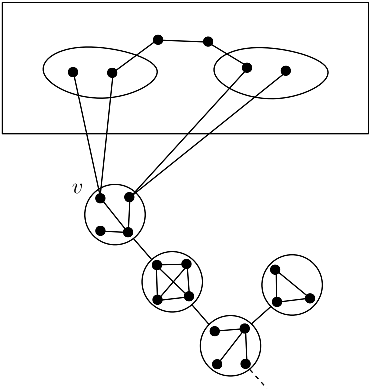
Twin Reduction Rule

If there exist two non-adjacent twins v, w in $G - S$ having at least one common neighbor in $G - S$ such that $(N_G(v) \cap S) \setminus (N_G(w) \cap S) \neq \emptyset$, then we can safely remove v.

Twin Reduction Rule

If there exist two non-adjacent twins v, w in $G - S$ having at least one common neighbor in $G - S$ such that $(N_G(v) \cap S) \backslash (N_G(w) \cap S) \neq \emptyset$, then we can safely remove v.

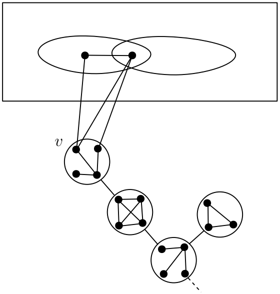
- Claim 1. $(N_G(v) \cap S) \cap (N_G(w) \cap S) \neq \emptyset$.

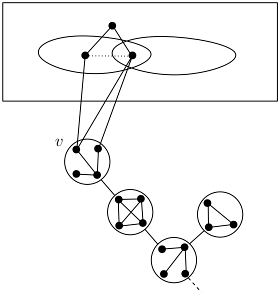


Twin Reduction Rule

If there exist two non-adjacent twins v, w in $G - S$ having at least one common neighbor in $G - S$ such that $(N_G(v) \cap S) \setminus (N_G(w) \cap S) \neq \emptyset$, then we can safely remove v.

- Claim 1. $(N_G(v) \cap S) \cap (N_G(w) \cap S) \neq \emptyset$.
- Claim 2. Let C be an induced cycle of length at least 5 containing v.
  3 types: 1) $S - v - S$ 2) $S - v - V(G - S)$ 3) $V(G - S) - v - V(G - S)$
  There is a bad vertex for C and v.

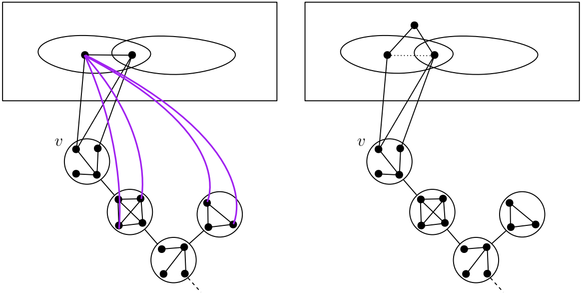




Twin Reduction Rule

If there exist two non-adjacent twins v, w in $G - S$ having at least one common neighbor in $G - S$ such that $(N_G(v) \cap S) \backslash (N_G(w) \cap S) \neq \emptyset$, then we can safely remove v.

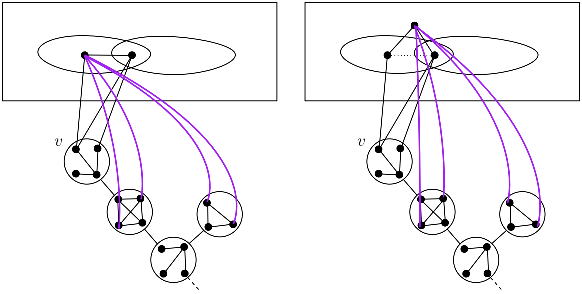
- Claim 1. $(N_G(v) \cap S) \cap (N_G(w) \cap S) \neq \emptyset$.
- Claim 2. Let C be an induced cycle of length at least 5 containing v.
  3 types: 1) $S - v - S$ 2) $S - v - V(G-S)$ 3) $V(G-S) - v - V(G-S)$
  There is a bad vertex for C and v.



Twin Reduction Rule

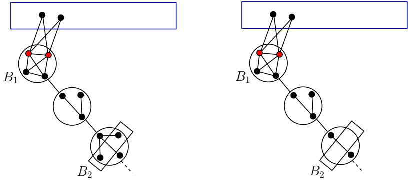
If there exist two non-adjacent twins v, w in $G - S$ having at least one common neighbor in $G - S$ such that $(N_G(v) \cap S) \setminus (N_G(w) \cap S) \neq \emptyset$, then we can safely remove v.

- Claim 1. $(N_G(v) \cap S) \cap (N_G(w) \cap S) \neq \emptyset$.
- Claim 2. Let C be an induced cycle of length at least 5 containing v.
  3 types: 1) $S - v - S$ 2) $S - v - V(G - S)$ 3) $V(G - S) - v - V(G - S)$
  There is a bad vertex for C and v.

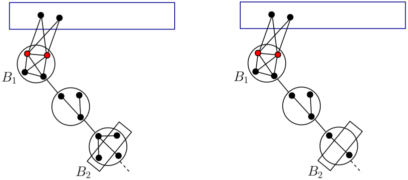


- Rule 1. D contains at most one class having neighbors on S, then we remove it.
- Rule 2. Deleting a vertex of degree 1.
- Rule 3. Twin Reduction Rule.

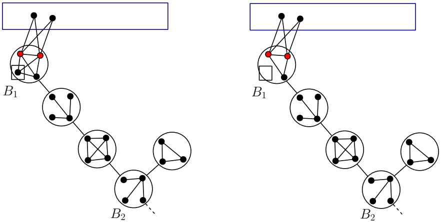
- Rule 1. D contains at most one class having neighbors on S, then we remove it.
- Rule 2. Deleting a vertex of degree 1.
- Rule 3. Twin Reduction Rule.
- Rule 4.



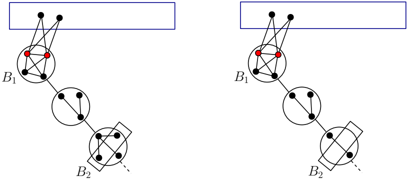
- Rule 1. D contains at most one class having neighbors on S, then we remove it.
- Rule 2. Deleting a vertex of degree 1.
- Rule 3. Twin Reduction Rule.
- Rule 4.



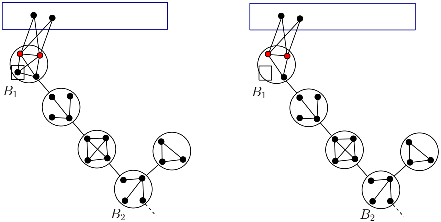
- Rule 5.



- Rule 1. D contains at most one class having neighbors on S, then we remove it.
- Rule 2. Deleting a vertex of degree 1.
- Rule 3. Twin Reduction Rule.
- Rule 4.

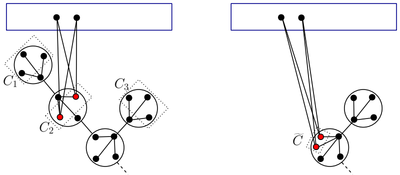


- Rule 5.



- Rule 6. Swapping the type of bags.

- Rule 7.

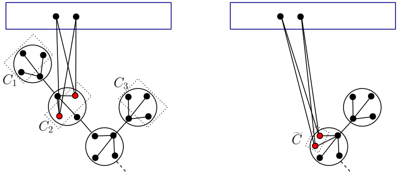


We obtain a property in a reduced instance:

(1) Every non-trivial branch should have a neighbor in S, and

(2) if there is a bag branching two branches, then two classes having neighbors in S in these branches have distance at most 4 in $G - S$.

- Rule 7.



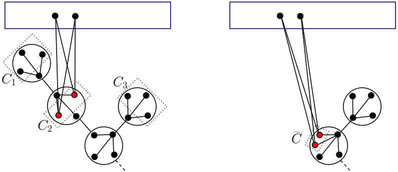
We obtain a property in a reduced instance:

(1) Every non-trivial branch should have a neighbor in S, and

(2) if there is a bag branching two branches, then two classes having neighbors in S in these branches have distance at most 4 in $G - S$.

We can apply some special rule to remove one of them (an extension of Twin Reduction Rule).

- Rule 7.



We obtain a property in a reduced instance:

(1) Every non-trivial branch should have a neighbor in S, and

(2) if there is a bag branching two branches, then two classes having neighbors in S in these branches have distance at most 4 in $G - S$.

We can apply some special rule to remove one of them (an extension of Twin Reduction Rule).

Eventually, we will arrive at a trivial instance.

Conclusion

- Distance-Hereditary Vertex Deletion can be solved in time $\mathcal{O}^*(37^k)$. Can we improve the running time?
- Does Distance-Hereditary Vertex Deletion admit a polynomial kernel?
  - Twin Reduction Rule is not possible, because we cannot kill the small obstructions by branching.
- For $w \geqslant 2$, Rank-width w Vertex Deletion can be solved in time $\mathcal{O}^*(c^k)$ for some constant c?

Conclusion

- Distance-Hereditary Vertex Deletion can be solved in time $\mathcal{O}^*(37^k)$. Can we improve the running time?
- Does Distance-Hereditary Vertex Deletion admit a polynomial kernel?
  - Twin Reduction Rule is not possible, because we cannot kill the small obstructions by branching.
- For $w \geqslant 2$, Rank-width w Vertex Deletion can be solved in time $\mathcal{O}^*(c^k)$ for some constant c?

Thank for listening!