Polynomial Kernel for Block Graph Deletion

Abraham Hinteregger

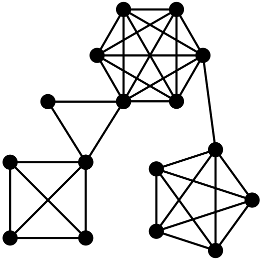
Vienna University of Technology

25.1.2018

Chapter

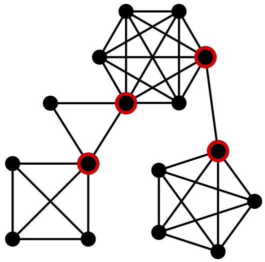
Block Graphs

- Block graphs consist of cliques



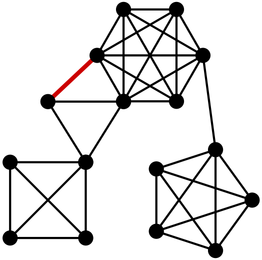
Block Graphs

- Block graphs consist of cliques
- Multiple cliques can share a vertice (articulation points)



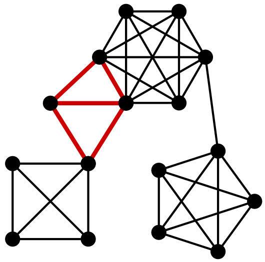
Block Graphs

- Block graphs consist of cliques
- Multiple cliques can share a vertice (articulation points)
- No diamond graphs  and cycles of length ≥ 4 as induced subgraphs (necessary & sufficient condition)



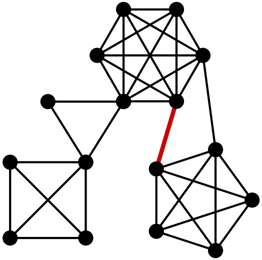
Block Graphs

- Block graphs consist of cliques
- Multiple cliques can share a vertice (articulation points)
- No diamond graphs  and cycles of length ≥ 4 as induced subgraphs (necessary & sufficient condition)



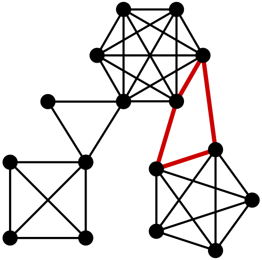
Block Graphs

- Block graphs consist of cliques
- Multiple cliques can share a vertice (articulation points)
- No diamond graphs  and cycles of length ≥ 4 as induced subgraphs (necessary & sufficient condition)



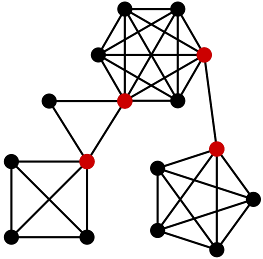
Block Graphs

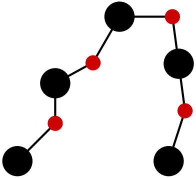
- Block graphs consist of cliques
- Multiple cliques can share a vertice (articulation points)
- No diamond graphs  and cycles of length ≥ 4 as induced subgraphs (necessary & sufficient condition)



Block Tree

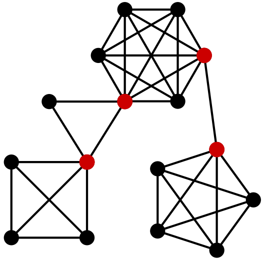
A block tree $\mathcal{T}_G$ of a graph G is a graph with the vertex set $\mathcal{B} \cup \mathcal{C}$.
$\mathcal{B} =$ set of blocks of G and $\mathcal{C} =$ set of articulation points of G.
Vertices $\mathcal{T}_G$ are connected with edge $\{c, B\} \in E(\mathcal{T}_G)$ iff the vertex c is in the block B in G.

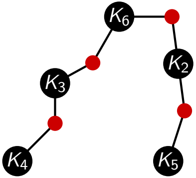




Block Tree

A block tree $\mathcal{T}_G$ of a graph G is a graph with the vertex set $\mathcal{B} \cup \mathcal{C}$. $\mathcal{B}$ = set of blocks of G and $\mathcal{C}$ = set of articulation points of G. Vertices $\mathcal{T}_G$ are connected with edge $\{c, B\} \in E(\mathcal{T}_G)$ iff the vertex c is in the block B in G.





Block Graph Deletion

BLOCK GRAPH DELETION

Input: A simple undirected graph G, an integer k

Parameter: k

Question: Is there a subset S of V with $|S| \leq k$ such that $G - S$ is a block graph?

Paper overview

The main results of [Kim and Kwon, 2017] are:

- Block Graph Deletion admits a kernel with size $\mathcal{O}(k^6)$
- Block Graph Deletion can be solved in time $10^k \cdot n^{\mathcal{O}(1)}$

Paper overview

The main results of [Kim and Kwon, 2017] are:

- Block Graph Deletion admits a kernel with size $\mathcal{O}(k^6)$
- Block Graph Deletion can be solved in time $10^k \cdot n^{\mathcal{O}(1)}$

Paper overview

The main results of [Kim and Kwon, 2017] are:

- Block Graph Deletion admits a kernel with size $\mathcal{O}(k^6)$
- Block Graph Deletion can be solved in time $10^k \cdot n^{\mathcal{O}(1)}$

Some details and most proofs will be omitted. For details look at the published paper (not the conference paper or the arxiv-preprint)

Chapter

Reduction Rules 1-3

Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

Reduction Rules 1-3

Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G.
- **RR2 (Cut vertex rule):** $\exists v \in V(G)$ s.t. $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph, we remove H from G.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

Reduction Rules 1-3

Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G.
- **RR2 (Cut vertex rule):** $\exists v \in V(G)$ s.t. $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph, we remove H from G.
- **RR3 (Twin rule):** If S is a set of vertices that are pairwise true twins[1] in G and $|S| \geq k + 2$ remove vertices until $|S| = k + 1$.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

Reduction Rules 1-3

Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G. $\implies$ graph only has obstructed components
- **RR2 (Cut vertex rule):** $\exists v \in V(G)$ s.t. $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph, we remove H from G.
- **RR3 (Twin rule):** If S is a set of vertices that are pairwise true twins[1] in G and $|S| \geq k + 2$ remove vertices until $|S| = k + 1$.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

Reduction Rules 1-3

Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G. $\Longrightarrow$ graph only has obstructed components
- **RR2 (Cut vertex rule):** $\exists v \in V(G)$ s.t. $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph, we remove H from G. $\Longrightarrow$ no “unobstructed” blocks
- **RR3 (Twin rule):** If S is a set of vertices that are pairwise true twins[1] in G and $|S| \geq k + 2$ remove vertices until $|S| = k + 1$.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

Reduction Rules 1-3

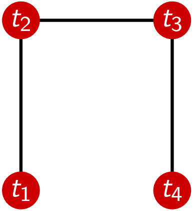
Testing if a graph is a block graph can be decided in quadratic time [Hopcroft and Tarjan, 1973]

- **RR1 (Block component rule):** If G has a component H that is a block graph then we remove H from G. $\Longrightarrow$ graph only has obstructed components
- **RR2 (Cut vertex rule):** $\exists v \in V(G)$ s.t. $G - v$ contains a component H where $G[V(H) \cup \{v\}]$ is a connected block graph, we remove H from G. $\Longrightarrow$ no "unobstructed" blocks
- **RR3 (Twin rule):** If S is a set of vertices that are pairwise true twins[1] in G and $|S| \geq k + 2$ remove vertices until $|S| = k + 1$. $\Longrightarrow$ If one of the vertices is part of an obstruction, all of them are.

[1] Two vertices are true twins if they share all their neighbors (except each other) and are connected

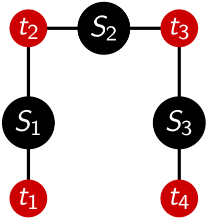
Reduction Rule 4

- **RR4 (Reduce block-cut vertex paths):**
  Let $t_1 t_2 t_3 t_4$ be an induced path



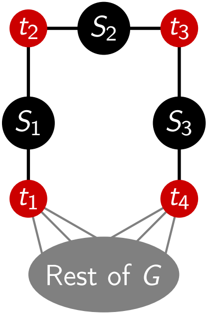
Reduction Rule 4

- **RR4 (Reduce block-cut vertex paths):** Let $t_1 t_2 t_3 t_4$ be an induced path and for $1 \leq i \leq 3$, let $S_i \subseteq V(G)\{t_1 \ldots t_4\}$ be a clique of G s.t.:
  - For each $1 \leq i \leq 3$ and $v \in S_i$:
    $$N_G(v) \setminus S_i = \{t_i, t_{i+1}\}$$



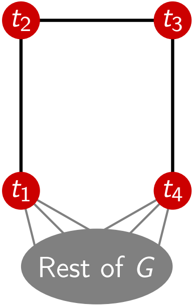
Reduction Rule 4

- **RR4 (Reduce block-cut vertex paths):** Let $t_1 t_2 t_3 t_4$ be an induced path and for $1 \leq i \leq 3$, let $S_i \subseteq V(G)\{t_1 \dots t_4\}$ be a clique of G s.t.:
  - For each $1 \leq i \leq 3$ and $v \in S_i$:
    $$N_G(v) \setminus S_i = \{t_i, t_{i+1}\}$$
  - For each $2 \leq i \leq 3$:
    $$N_G(t_i) = \{t_{i-1}, t_{i+1}\} \cup S_{i-1} \cup S_i$$



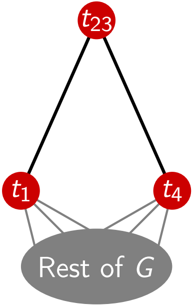
Reduction Rule 4

- **RR4 (Reduce block-cut vertex paths):** Let $t_1 t_2 t_3 t_4$ be an induced path and for $1 \leq i \leq 3$, let $S_i \subseteq V(G)\{t_1 \dots t_4\}$ be a clique of G s.t.:
  - For each $1 \leq i \leq 3$ and $v \in S_i$:
    $$N_G(v) \setminus S_i = \{t_i, t_{i+1}\}$$
  - For each $2 \leq i \leq 3$:
    $$N_G(t_i) = \{t_{i-1}, t_{i+1}\} \cup S_{i-1} \cup S_i$$
  - Then: Remove $S = S_1 \cup S_2 \cup S_3$



Reduction Rule 4

- **RR4 (Reduce block-cut vertex paths):** Let $t_1 t_2 t_3 t_4$ be an induced path and for $1 \leq i \leq 3$, let $S_i \subseteq V(G)\{t_1 \ldots t_4\}$ be a clique of G s.t.:
  - For each $1 \leq i \leq 3$ and $v \in S_i$: $$N_G(v) \setminus S_i = \{t_i, t_{i+1}\}$$
  - For each $2 \leq i \leq 3$: $$N_G(t_i) = \{t_{i-1}, t_{i+1}\} \cup S_{i-1} \cup S_i$$
  - Then: Remove $S = S_1 \cup S_2 \cup S_3$ & contract edge between t_2 and t_3



Proof: Soundness of RR4

- Assume there is an obstruction in $S \cup \{t_2, t_3\}$ (the vertices affected by RR4)

Proof: Soundness of RR4

- Assume there is an obstruction in $S \cup \{t_2, t_3\}$ (the vertices affected by RR4)
- A vertex in S cannot be part of an obstruction (a cycle of length ≥ 4 or a diamond graph 

Proof: Soundness of RR4

- Assume there is an obstruction in $S \cup \{t_2, t_3\}$ (the vertices affected by RR4)
- A vertex in S cannot be part of an obstruction (a cycle of length ≥ 4 or a diamond graph 
- Vertices in obstructions are 2-connected, thus the obstruction is a superset of $\{t_2, t_3\}$ that contains no vertex of S.

Proof: Soundness of RR4

- Assume there is an obstruction in $S \cup \{t_2, t_3\}$ (the vertices affected by RR4)
- A vertex in S cannot be part of an obstruction (a cycle of length ≥ 4 or a diamond graph 
- Vertices in obstructions are 2-connected, thus the obstruction is a superset of $\{t_2, t_3\}$ that contains no vertex of S.
- As $t_1 \ldots t_4$ is an induced path, the obstruction cannot be a diamond graph or a cycle of length 4 $\implies$ the obstruction has to be a cycle of length ≥ 5

Proof: Soundness of RR4

- Assume there is an obstruction in $S \cup \{t_2, t_3\}$ (the vertices affected by RR4)
- A vertex in S cannot be part of an obstruction (a cycle of length ≥ 4 or a diamond graph 
- Vertices in obstructions are 2-connected, thus the obstruction is a superset of $\{t_2, t_3\}$ that contains no vertex of S.
- As $t_1 \dots t_4$ is an induced path, the obstruction cannot be a diamond graph or a cycle of length 4 $\implies$ the obstruction has to be a cycle of length ≥ 5
- Contracting an edge in a cycle of length ≥ 5 results in a cycle of length ≥ 4. Therefore, RR4 does not remove an obstruction.

Reduction Rule 5

Proposition: Given a graph G and $v \in V(G)$ and k a positive integer. In $\mathcal{O}(kn^3)$ time we can find either:

1. $k+1$ pairwise vertex disjoint obstructions
2. $k+1$ obstructions containing vertex v
3. $S_v \subseteq V(G) \setminus \{v\}$ with $|S_v| \leq 7k$ s.t. $G - S_v$ has no obstruction containing v.

The complete degree of v is defined as the minimum number of compoments of $G[N_G(v) \setminus S_v]$ among all S_v considered above.

Reduction Rule 5

Proposition: Given a graph G and $v \in V(G)$ and k a positive integer. In $\mathcal{O}(kn^3)$ time we can find either:

1. $k+1$ pairwise vertex disjoint obstructions $\Longrightarrow$ No-instance
2. $k+1$ obstructions containing vertex v $\Longrightarrow$ Remove v, $k=k-1$
3. $S_v \subseteq V(G) \setminus \{v\}$ with $|S_v| \leq 7k$ s.t. $G - S_v$ has no obstruction containing v.

The complete degree of v is defined as the minimum number of compoments of $G[N_G(v) \setminus S_v]$ among all S_v considered above.

- **RR5 (($k+1$)-distinct obstructions rule):**
  Apply the proposition above.

Reduction Rule 5

Proposition: Given a graph G and $v \in V(G)$ and k a positive integer. In $\mathcal{O}(kn^3)$ time we can find either:

1. $k+1$ pairwise vertex disjoint obstructions $\Longrightarrow$ No-instance
2. $k+1$ obstructions containing vertex v $\Longrightarrow$ Remove v, $k = k-1$
3. $S_v \subseteq V(G) \setminus \{v\}$ with $|S_v| \leq 7k$ s.t. $G - S_v$ has no obstruction containing v.

The complete degree of v is defined as the minimum number of compoments of $G[N_G(v) \setminus S_v]$ among all S_v considered above.

- **RR5 ($(k+1)$-distinct obstructions rule):** Apply the proposition above.
  - If G is reduced with RR1-RR5 and has more than $\mathcal{O}(k^6)$ vertices, G has a vertex v s.t. RR6 can be applied

Reduction Rule 6

- **RR6 (Large complete degree rule):**
  $v \in V(G)$ and $X \subseteq V(G) \setminus \{v\}$ with $|X| \leq 7k|$.
  Let $\mathcal{C}$ be a set of connected components of $G - X \cup \{c\}$.
  Let $\varphi : X \to C_3$ where C_3 is the set of subsets of $\mathcal{C}$ with cardinality 3, s.t.:

Reduction Rule 6

- **RR6 (Large complete degree rule):**
  $v \in V(G)$ and $X \subseteq V(G) \setminus \{v\}$ with $|X| \leq 7k|$.
  Let $\mathcal{C}$ be a set of connected components of $G - X \cup \{c\}$.
  Let $\varphi : X \to C_3$ where C_3 is the set of subsets of $\mathcal{C}$ with cardinality 3, s.t.:
  - v together with any component in $\mathcal{C}$ induces a subgraph of G that is a blockgraph.

Reduction Rule 6

- **RR6 (Large complete degree rule):**
  $v \in V(G)$ and $X \subseteq V(G) \setminus \{v\}$ with $|X| \leq 7k|$.
  Let $\mathcal{C}$ be a set of connected components of $G - X \cup \{c\}$.
  Let $\varphi : X \to C_3$ where C_3 is the set of subsets of $\mathcal{C}$ with cardinality 3, s.t.:
  - v together with any component in $\mathcal{C}$ induces a subgraph of G that is a blockgraph.
  - v has a neighbor in $\forall C \in \mathcal{C}$ and there is a $x \in X$ that has a neighbor in C.

Reduction Rule 6

- **RR6 (Large complete degree rule):**
  $v \in V(G)$ and $X \subseteq V(G) \setminus \{v\}$ with $|X| \leq 7k|$.
  Let $\mathcal{C}$ be a set of connected components of $G - X \cup \{c\}$.
  Let $\varphi : X \to C_3$ where C_3 is the set of subsets of $\mathcal{C}$ with cardinality 3, s.t.:
  - v together with any component in $\mathcal{C}$ induces a subgraph of G that is a blockgraph.
  - v has a neighbor in $\forall C \in \mathcal{C}$ and there is a $x \in X$ that has a neighbor in C.
  - $\varphi(x)$ maps x to components that are neighbors of x.

Reduction Rule 6

- **RR6 (Large complete degree rule):**
  $v \in V(G)$ and $X \subseteq V(G) \setminus \{v\}$ with $|X| \leq 7k|$.
  Let $\mathcal{C}$ be a set of connected components of $G - X \cup \{c\}$.
  Let $\varphi : X \to C_3$ where C_3 is the set of subsets of $\mathcal{C}$ with cardinality 3, s.t.:
  - v together with any component in $\mathcal{C}$ induces a subgraph of G that is a blockgraph.
  - v has a neighbor in $\forall C \in \mathcal{C}$ and there is a $x \in X$ that has a neighbor in C.
  - $\varphi(x)$ maps x to components that are neighbors of x.
  - $\varphi(x) \cap \varphi(y) \neq \emptyset$ iff $x \neq y$

Illustration of RR6

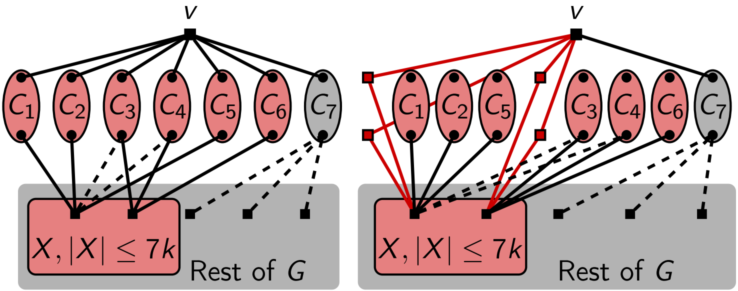


Figure: Application of RR6: $\bigcup_{x \in X} \varphi(x)$ gets disconnected from v and for each $x \in X$ two vertex disjoint paths ( ) from x to v are added

Illustration of RR6

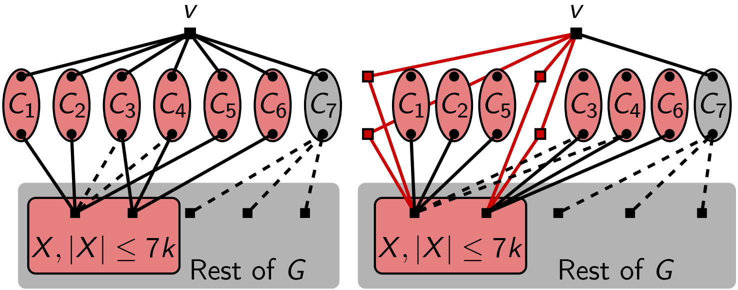


Figure: Application of RR6: $\bigcup_{x \in X} \varphi(x)$ gets disconnected from v and for each $x \in X$ two vertex disjoint paths ( ) from x to v are added

For $\geq 3|X|$ removed edges $2|X|$ edges are introduced that are connected to a vertex with degree 2

Illustration of RR6

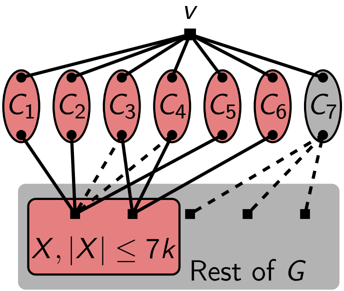


Figure: Application of RR6: $\bigcup_{x \in X} \varphi(x)$ gets disconnected from v and for each $x \in X$ two vertex disjoint paths ( ) from x to v are added

For $\geq 3|X|$ removed edges $2|X|$ edges are introduced that are connected to a vertex with degree 2

RR6 reduces edges that connect two vertices with degree ≥ 3 by at least 1.

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions ($\# > k \implies$ No-instance)

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions ($\# > k \implies$ No-instance)
- Application of RR6 yields components that are connected to the rest of the graph via a cut vertex $x \in X$.

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions ($\# > k \implies$ No-instance)
- Application of RR6 yields components that are connected to the rest of the graph via a cut vertex $x \in X$.
- These components can be removed from G with RR2 (Cut vertex rule), then start again from top.

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions ($\# > k \implies$ No-instance)
- Application of RR6 yields components that are connected to the rest of the graph via a cut vertex $x \in X$.
- These components can be removed from G with RR2 (Cut vertex rule), then start again from top.
- Each application of RR6 reduces the number of edges which connect two vertices with a vertex degree ≥ 3 by at least 1.

Outline of algorithm for polykernel

- Apply rules 1-5 exhaustively. Then one of the following is true:
  1. The given instance is a No-instance
  2. The reduced graph is a polykernel of size $\mathcal{O}(k^6)$
  3. There exists a vertex v s.t. RR6 can be applied
- Count the number of components (from RR6) with obstructions ($\# > k \implies$ No-instance)
- Application of RR6 yields components that are connected to the rest of the graph via a cut vertex $x \in X$.
- These components can be removed from G with RR2 (Cut vertex rule), then start again from top.
- Each application of RR6 reduces the number of edges which connect two vertices with a vertex degree ≥ 3 by at least 1.
- Therefore at some point either (1) or (3) is true

References

Hopcroft, J. and Tarjan, R. (1973).
Algorithm 447: Efficient Algorithms for Graph Manipulation.
Commun. ACM, 16(6):372–378.

Kim, E. J. and Kwon, O.-J. (2017).
A Polynomial Kernel for Block Graph Deletion.
Algorithmica, 79(1):251–270.

Thomassé, S. (2010).
A $4k^2$ kernel for feedback vertex set.
ACM Transactions on Algorithms (TALG), 6(2):32.