Complexity and Exact Algorithms for MULTICUT

J. Guo[1], F. Hüffner[1], **E. Kenar**[2], R. Niedermeier[1], J. Uhlmann[2]

[1]Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany

[2]Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany

SOFSEM 2006

Multicut

Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$

Task: Find a minimum set of edges or vertices whose removal disconnects each pair of H.

edge deletion	**vertex deletion**
Multiterminal Cut (MTC) Edge Multicut (EMC)	Unrestricted Vertex Multicut (UVMC) Restricted Vertex Multicut (RVMC)

Multicut

Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$

Task: Find a minimum set of edges or vertices whose removal disconnects each pair of H.

edge deletion	**vertex deletion**
Multiterminal Cut (MTC)	Unrestricted Vertex Multicut (UVMC)
Edge Multicut (EMC)	Restricted Vertex Multicut (RVMC)

Restricted Vertex Multicut (RVMC)

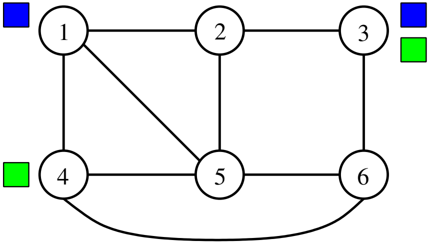
Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$
- Integer $k \geq 0$

Task: Find a subset V' of V with $|V'| \leq k$ that contains **no terminal** and whose removal disconnects each pair of H.

Figure: RVMC instance $G = (V, E)$ with $H = \{(1, 3), (3, 4)\}$, $k = 2$



Restricted Vertex Multicut (RVMC)

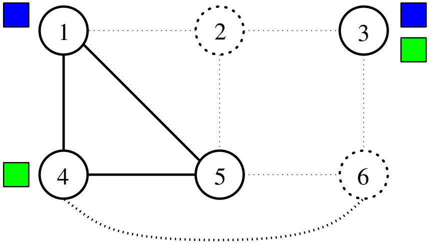
Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$
- Integer $k \geq 0$

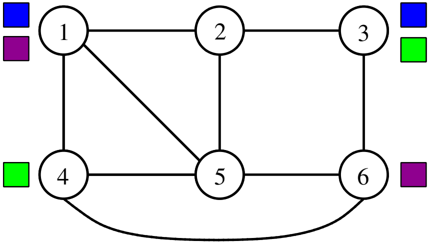
Task: Find a subset V' of V with $|V'| \leq k$ that contains **no terminal** and whose removal disconnects each pair of H.

Figure: RVMC instance $G = (V, E)$ with $H = \{(1, 3), (3, 4)\}$, $k = 2$



Restricted Vertex Multicut (RVMC)

Figure: RVMC instance $G = (V, E)$ with $H = \{(1,3), (3,4)\}$, $k = 2$



Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$
- Integer $k \geq 0$

Task: Find a subset V' of V with $|V'| \leq k$ that contains **no terminal** and whose removal disconnects each pair of H.

Restricted Vertex Multicut (RVMC)

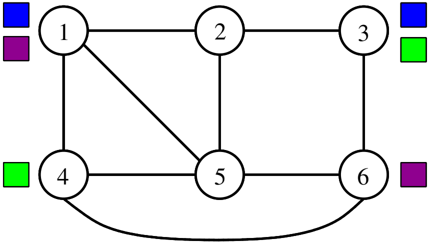
Definition

Input:

- Undirected graph $G = (V, E)$
- Set of terminals $S \subseteq V$
- Set of pairs of terminals $H \subseteq S \times S$
- Integer $k \geq 0$

Task: Find a subset V' of V with $|V'| \leq k$ that contains **no terminal** and whose removal disconnects each pair of H.

Figure: RVMC instance $G = (V, E)$ with $H = \{(1, 3), (3, 4)\}$, $k = 2$

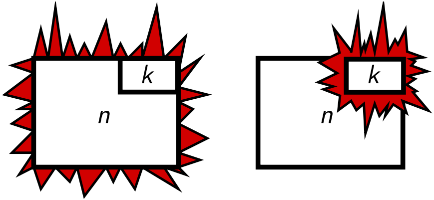


Note: No solution for $H = \{(1, 3), (1, 6), (3, 4)\}$!

Fixed-parameter tractability

Idea

Restrict the seemingly inherent combinatorial explosion of hard problems to some problem-specific parameters.



Definition (fixed-parameter tractable)

Problem P is fixed-parameter tractable

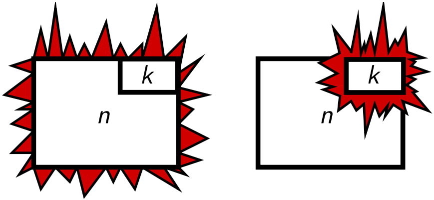
$$\Longleftrightarrow$$

P is solvable in $O(f(k) \cdot n^c)$ time

Fixed-parameter tractability

Idea

Restrict the seemingly inherent combinatorial explosion of hard problems to some problem-specific parameters.



Definition (fixed-parameter tractable)

Problem P is fixed-parameter tractable

$$\Longleftrightarrow$$

P is solvable in $O(f(k) \cdot n^c)$ time

Outline

1. Introduction
2. RVMC in Trees
   - Complexity Results
   - FPT-algorithm: Search Tree
3. RVMC in Interval Graphs
   - Complexity Results
4. RVMC in General Graphs
   - Complexity Results
   - FPT-algorithm: Coloring Problem
5. Conclusion

RVMC in trees

RVMC hardness results for trees

- Paths: RVMC in trees with maximum **vertex degree two** is solvable in polynomial time: $O(|V| \cdot |H|)$.

- Trees: NP-completeness has been shown for RVMC in trees with maximum **vertex degree four**. [Călinescu et al., Journal of Algorithms, 2003]

- In comparison: UNRESTRICTED VERTEX MULTICUT in trees can easily be solved in $O(|V| \cdot |H|)$ time (least common ancestor).

RVMC hardness results for trees

- Paths: RVMC in trees with maximum **vertex degree two** is solvable in polynomial time: $O(|V| \cdot |H|)$.

Theorem

RESTRICTED VERTEX MULTICUT in trees with maximum **vertex degree three** is NP-complete.

- Trees: NP-completeness has been shown for RVMC in trees with maximum **vertex degree four**. [Călinescu et al., Journal of Algorithms, 2003]

- In comparison: UNRESTRICTED VERTEX MULTICUT in trees can easily be solved in $O(|V| \cdot |H|)$ time (least common ancestor).

RVMC hardness results for trees

- Paths: RVMC in trees with maximum **vertex degree two** is solvable in polynomial time: $O(|V| \cdot |H|)$.

Theorem

RESTRICTED VERTEX MULTICUT in trees with maximum **vertex degree three** is NP-complete.

- Trees: NP-completeness has been shown for RVMC in trees with maximum **vertex degree four**. [Călinescu et al., Journal of Algorithms, 2003]
- In comparison: UNRESTRICTED VERTEX MULTICUT in trees can easily be solved in $O(|V| \cdot |H|)$ time (least common ancestor).

FPT-algorithm

Theorem

RVMC in trees can be solved in $O(2^k \cdot |V| \cdot |H|)$ time, where k is the number of allowed vertex removals.

- FPT-algorithm is based on a depth-bounded search tree

FPT-algorithm

Theorem

RVMC in trees can be solved in $O(2^k \cdot |V| \cdot |H|)$ time, where k is the number of allowed vertex removals.

- FPT-algorithm is based on a depth-bounded search tree
- Preprocessing of the instance $T = (V, E)$ with S and H:

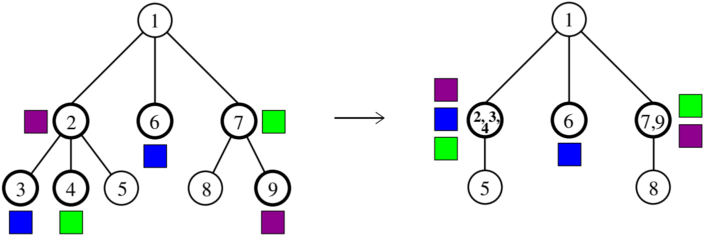
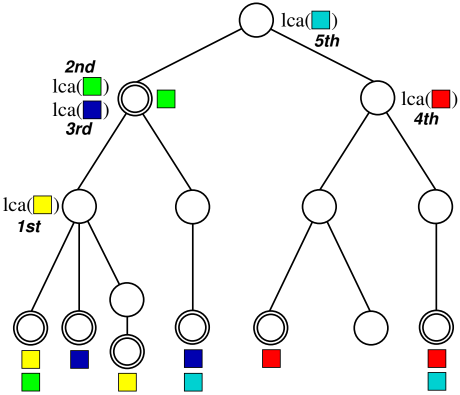


Figure: $H = \{(2,9), (3,6), (4,7)\}$. Edges with both endpoints being terminals are contracted.

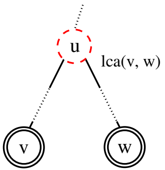
FPT-algorithm — search tree

1) Compute the least common ancestor (lca) for each terminal pair and sort them by decreasing depth in a list *L*.



FPT-algorithm — search tree

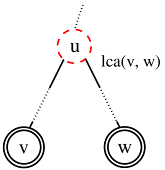
2) While $L \neq \emptyset$, consider the first element u of L (least common ancestor of pair (v, w)):



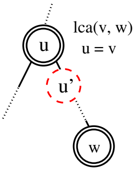
Case 1:
$u \notin S$ (nonterminal)
$\rightarrow$ remove u

FPT-algorithm — search tree

2) While $L \neq \emptyset$, consider the first element u of L (least common ancestor of pair (v, w)):



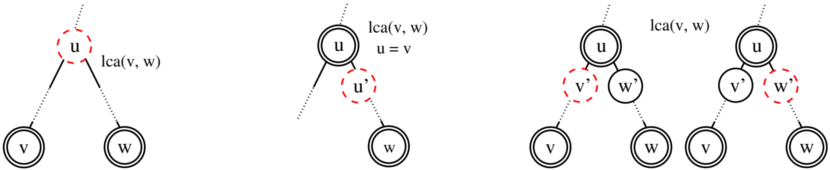
Case 1:
$u \notin S$ (nonterminal)
$\longrightarrow$ remove u



Case 2a:
$u \in S$, $u = v$ or $u = w$
$\longrightarrow$ remove $u' \in P$

FPT-algorithm — search tree

2) While $L \neq \emptyset$, consider the first element u of L (least common ancestor of pair (v, w)):



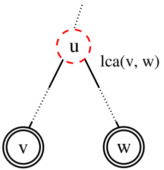
Case 1:
$u \notin S$ (nonterminal)
$\rightarrow$ remove u

Case 2a:
$u \in S$, $u = v$ or $u = w$
$\rightarrow$ remove $u' \in P$

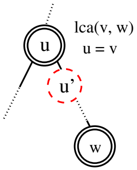
Case 2b: (branching)
$u \in S$, $u \neq v$ and $u \neq w$
$\rightarrow$ remove u' **or** v'

FPT-algorithm — search tree

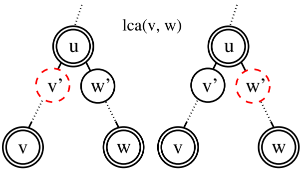
2) While $L \neq \emptyset$, consider the first element u of L (least common ancestor of pair (v, w)):



Case 1:
$u \notin S$ (nonterminal)
$\longrightarrow$ remove u



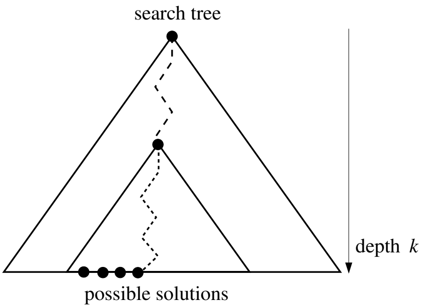
Case 2a:
$u \in S$, $u = v$ or $u = w$
$\longrightarrow$ remove $u' \in P$



Case 2b: (branching)
$u \in S$, $u \neq v$ and $u \neq w$
$\longrightarrow$ remove u' **or** v'

Termination: If $L = \emptyset$ or k nodes have been removed.

FPT-algorithm — run time



Depth: parameter k
Size: $O(2^k)$
Update step: $O(|V| \cdot |H|)$ time
$\rightarrow$ Run time: $O(2^k \cdot |V| \cdot |H|)$

RVMC in interval graphs

Interval graph

- A graph is an *interval graph* if we can label its vertices by intervals of the real line such that there is an edge between two vertices if and only if their intervals intersect.

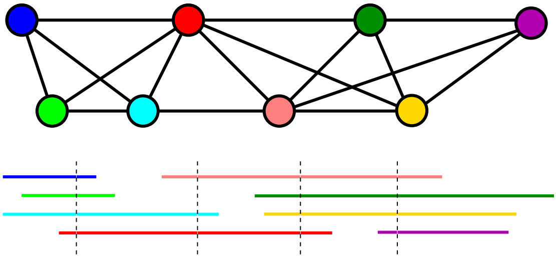


Figure: Example for an interval graph and its corresponding intervals on the real line.

Complexity results

Theorem

RESTRICTED VERTEX MULTICUT in interval graphs can be solved in **polynomial time**.

$\rightarrow$ Dynamic programming algorithm with run time $O(|V|^2 \cdot |H|^2)$

	UVMC	RVMC
Trees	P	NP-c
Interval graphs	NP-c	P

RVMC in general graphs

Hardness results for general graphs

- Restricted Vertex Multicut is NP-complete if there are at least six terminals. [Dahlhaus et al., SIAM Journal on Computing, 1994]
- There is **no** FPT-algorithm with respect to the parameter "number of terminals".
- RVMC is NP-complete for trees with bounded vertex degree and **bounded pathwidth** → **no** FPT-algorithm with *path-* or *treewidth* as parameter.

Idea

A combination of both parameters "number of terminals" and treewidth leads to an FPT-algorithm.

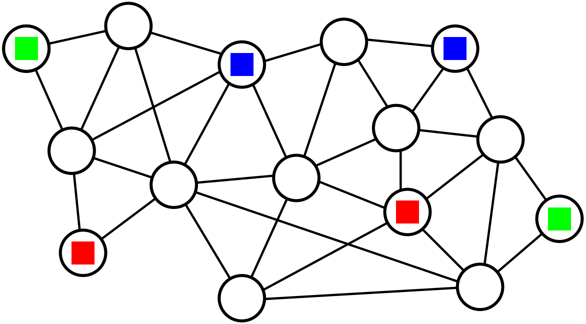
Hardness results for general graphs

- RESTRICTED VERTEX MULTICUT is NP-complete if there are at least six terminals. [Dahlhaus et al., SIAM Journal on Computing, 1994]
- There is **no** FPT-algorithm with respect to the parameter "number of terminals".
- RVMC is NP-complete for trees with bounded vertex degree and **bounded pathwidth** → **no** FPT-algorithm with *path-* or *treewidth* as parameter.

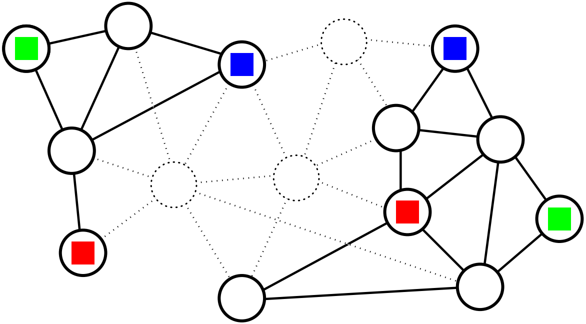
Idea

A combination of both parameters "number of terminals" and treewidth leads to an FPT-algorithm.

FPT-algorithm — basic idea



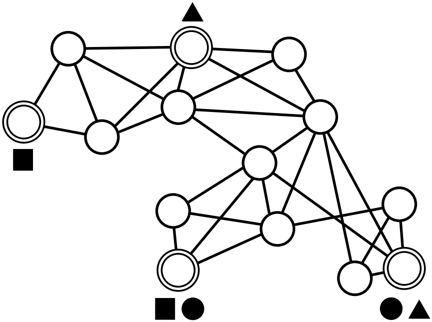
FPT-algorithm — basic idea



Key observation: Any solution of RVMC divides the input graph into at least two connected components → the two terminals of a terminal pair are in distinct connected components!

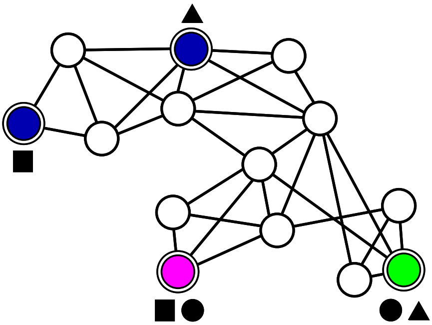
FPT-algorithm — the two stages

Stage one: Enumerate all possible colorings of the terminal set S with colors C such that for each terminal pair the two terminals are differently colored (*pre-coloring* of S).



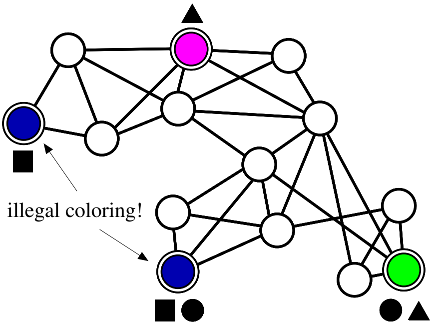
FPT-algorithm — the two stages

Stage one: Enumerate all possible colorings of the terminal set S with colors C such that for each terminal pair the two terminals are differently colored (*pre-coloring* of S).



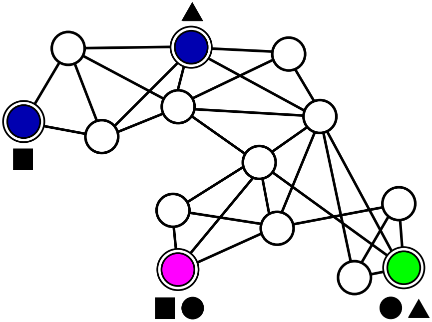
FPT-algorithm — the two stages

Stage one: Enumerate all possible colorings of the terminal set S with colors C such that for each terminal pair the two terminals are differently colored (*pre-coloring* of S).



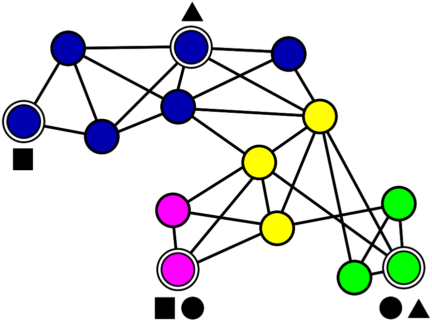
FPT-algorithm — the two stages

Stage two: Extend the legal pre-colorings of stage one with the previous colors of C **plus** an additional color (e.g. *yellow*).



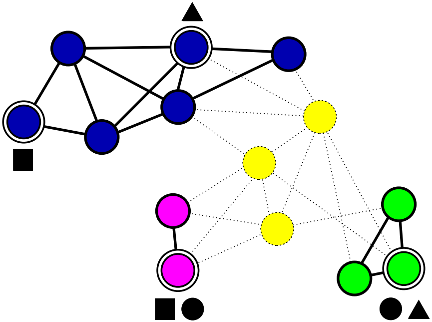
FPT-algorithm — the two stages

Stage two: Extend the legal pre-colorings of stage one with the previous colors of C **plus** an additional color (e.g. *yellow*).



FPT-algorithm — the two stages

Stage two: Extend the legal pre-colorings of stage one with the previous colors of C **plus** an additional color (e.g. *yellow*).



FPT-algorithm — run time

- The color extension can be computed by dynamic programming on the tree decomposition of the input graph with *treewidth* ω.

Theorem

Given an undirected graph $G = (V, E)$ with a tree decomposition of width ω, Restricted Vertex Multicut can be solved in $O(|S|^{|S|+\omega+1} \cdot (|V| + |E|))$ time, where S is the terminal set.

- Note: This FPT-algorithm also works for Unrestricted Vertex Multicut in general graphs.

Conclusion

Summary

Graph class	Parameter	EMC	UVMC	RVMC
Interval graphs		NP-c	NP-c	**P**
Trees		NP-c	P	NP-c
	k	FPT	P	**FPT**
General graphs		NP-c	NP-c	NP-c
	k	*open*	*open*	*open*
	$\|S\|$	NP-c	FPT	NP-c
	ω	NP-c	NP-c	NP-c
	$\|S\|$ and ω	FPT	FPT	**FPT**

Table: Complexity of MULTICUT problems for several graph classes. $|S|$: number of terminals, k: number of deletions, ω: treewidth of the input graph

Thank you for listening!