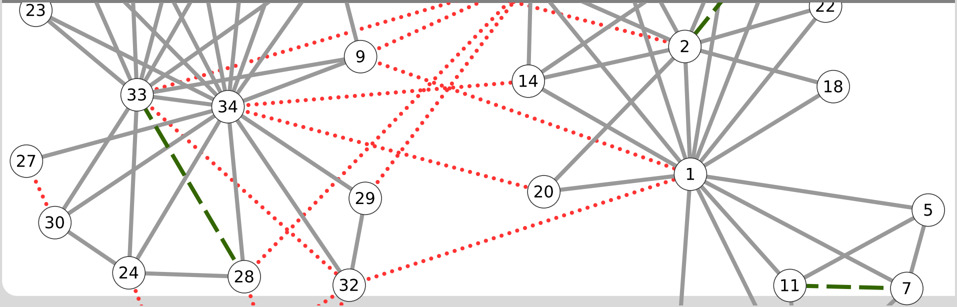
Karlsruhe Institute of Technology

Engineering Exact Quasi-Threshold Editing

Lars Gottesbüren, **Michael Hamann**, Philipp Schoch, Ben Strasser, Dorothea Wagner and Sven Zühlsdorf | June 2020

KARLSRUHE INSTITUTE OF TECHNOLOGY – INSTITUTE OF THEORETICAL INFORMATICS – GROUP ALGORITHMICS I



KIT – The Research University in the Helmholtz Association

www.kit.edu

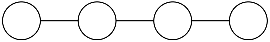
Quasi-Threshold Graphs

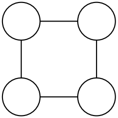
- Trivially perfect graphs

Institute of Theoretical Informatics
Group Algorithmics

Quasi-Threshold Graphs

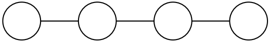
- Trivially perfect graphs
- No P_4 or C_4 as node-induced subgraph

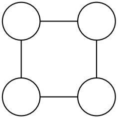




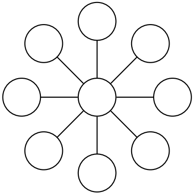
Quasi-Threshold Graphs

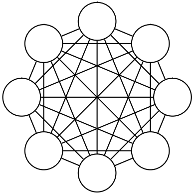
- Trivially perfect graphs
- No P_4 or C_4 as node-induced subgraph





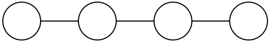
- Dense? Sparse? – Both!

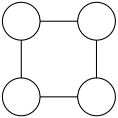




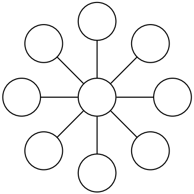
Quasi-Threshold Graphs

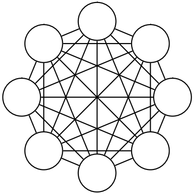
- Trivially perfect graphs
- No P_4 or C_4 as node-induced subgraph





- Dense? Sparse? – Both!





- Certifying recognition in linear time. [Chu08, BHSW15]

Why Quasi-Threshold Graphs?

Karlsruhe Institute of Technology

- Interesting graph class with natural definition

Institute of Theoretical Informatics
Group Algorithmics

Why Quasi-Threshold Graphs?

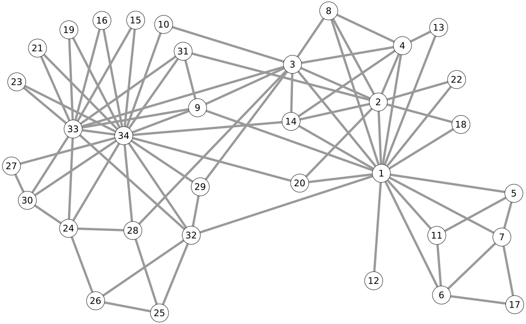
- Interesting graph class with natural definition
- Components of quasi-threshold graphs are communities [NG13]

Why Quasi-Threshold Graphs?

- Interesting graph class with natural definition
- Components of quasi-threshold graphs are communities [NG13]
- Real world graphs are not quasi-threshold graphs
  $\rightsquigarrow$ Find quasi-threshold graph with small edge edit distance

Why Quasi-Threshold Graphs?

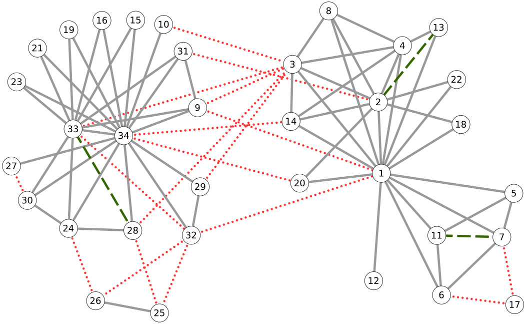
- Interesting graph class with natural definition
- Components of quasi-threshold graphs are communities [NG13]
- Real world graphs are not quasi-threshold graphs
  ⇝ Find quasi-threshold graph with small edge edit distance



Why Quasi-Threshold Graphs?

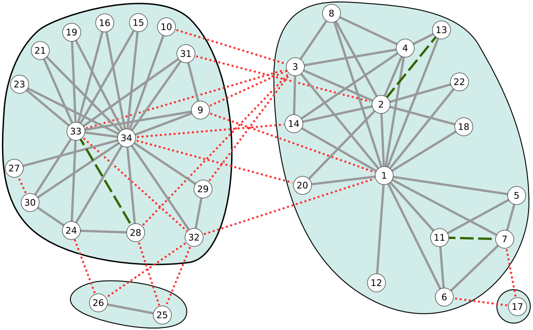
Karlsruhe Institute of Technology

- Interesting graph class with natural definition
- Components of quasi-threshold graphs are communities [NG13]
- Real world graphs are not quasi-threshold graphs
  ⇝ Find quasi-threshold graph with small edge edit distance



Why Quasi-Threshold Graphs?

- Interesting graph class with natural definition
- Components of quasi-threshold graphs are communities [NG13]
- Real world graphs are not quasi-threshold graphs
  ⇝ Find quasi-threshold graph with small edge edit distance



Quasi-Threshold Editing

Karlsruhe Institute of Technology

Quasi-Threshold Editing Problem

Given a graph G find a quasi-threshold graph with minimum edge editing (insertion + deletion) distance to G.

Quasi-Threshold Editing

Quasi-Threshold Editing Problem

Given a graph G find a quasi-threshold graph with minimum edge editing (insertion + deletion) distance to G.

- Is NP-hard [NG13]
- Is FPT $O\left(6^k\,(|V| + |E|)\right)$ [Cai96]
- Polynomial kernel exists ($O\left(k^7\right)$ vertices) [DP17]
- Heuristics exist [NG13, BHSW15]

Institute of Theoretical Informatics
Group Algorithmics

Quasi-Threshold Editing

Quasi-Threshold Editing Problem

Given a graph G find a quasi-threshold graph with minimum edge editing (insertion + deletion) distance to G.

- Is NP-hard [NG13]
- Is FPT $O(6^k (|V| + |E|))$ [Cai96]
- Polynomial kernel exists ($O(k^7)$ vertices) [DP17]
- Heuristics exist [NG13, BHSW15]

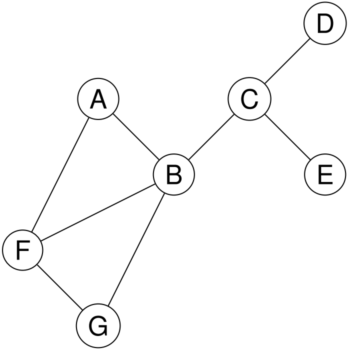
Our contribution:

- Exact algorithms – evaluation of heuristics and exact solutions
- Improved branch-and-bound FPT algorithm and ILP
- For forbidden subgraphs $\mathcal{F}$
- Experimental evaluation for $\{P_4, C_4\}$

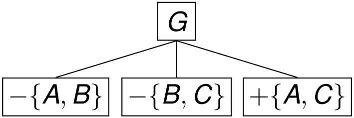
Example: P_3-free, $k = 3$

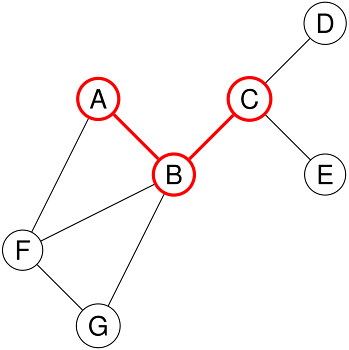
Karlsruhe Institute of Technology

G

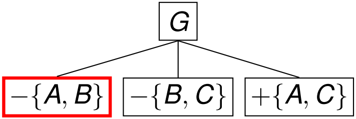


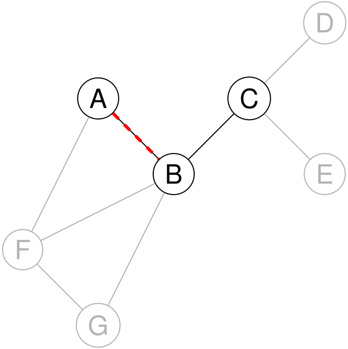
Example: P_3-free, $k = 3$



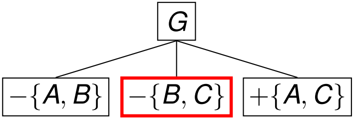


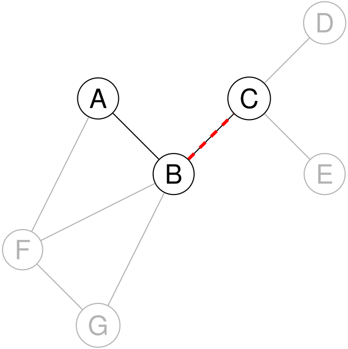
Example: P_3-free, $k = 3$



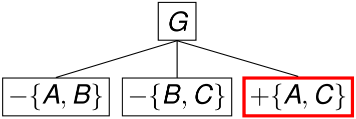


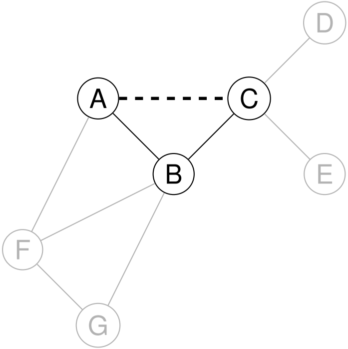
Example: P_3-free, $k = 3$





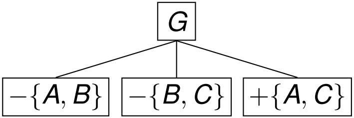
Example: P_3-free, $k = 3$

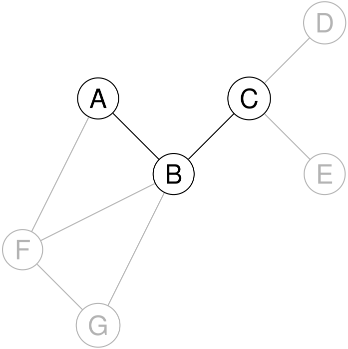




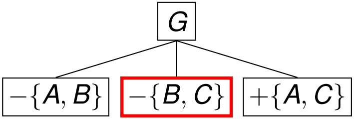
Institute of Theoretical Informatics
Group Algorithmics

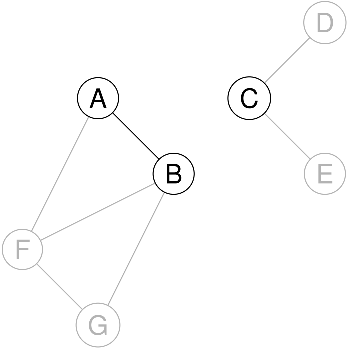
Example: P_3-free, $k = 3$



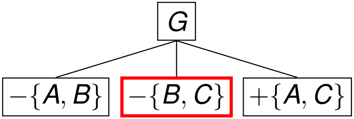


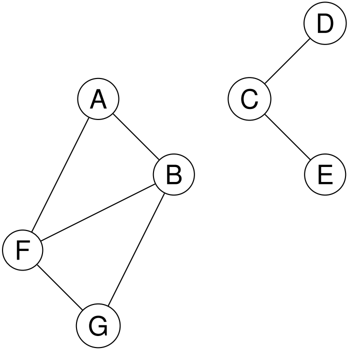
Example: P_3-free, $k = 3$



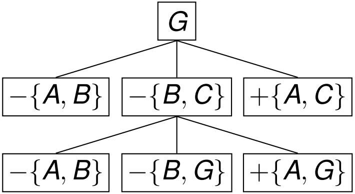


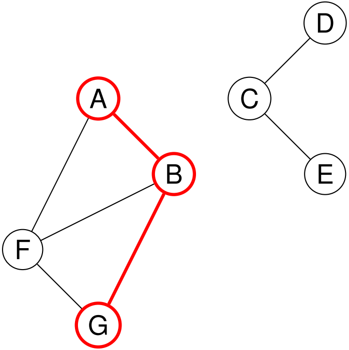
Example: P_3-free, $k = 3$



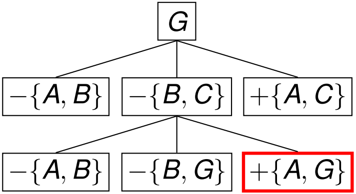


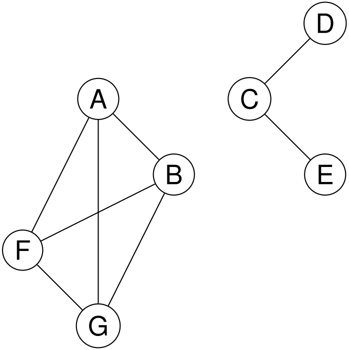
Example: P_3-free, $k = 3$



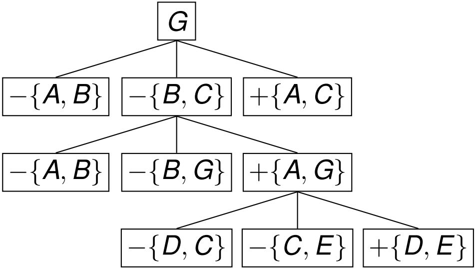


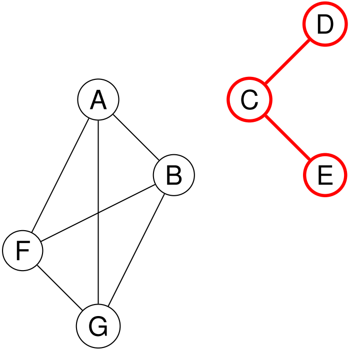
Example: P_3-free, $k = 3$



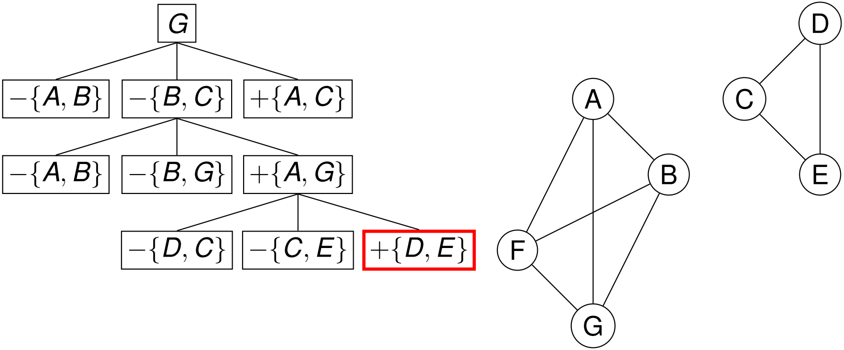


Example: P_3-free, $k = 3$

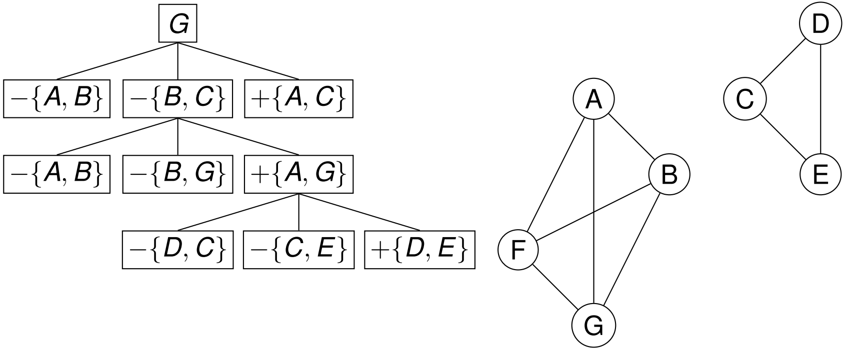




Example: P_3-free, $k = 3$

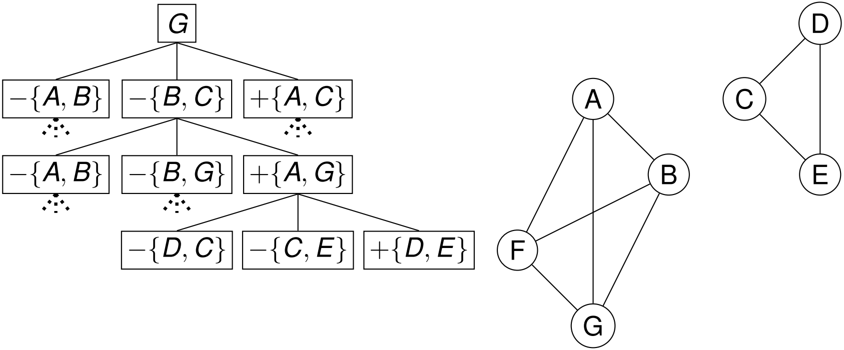


Example: P_3-free, $k = 3$



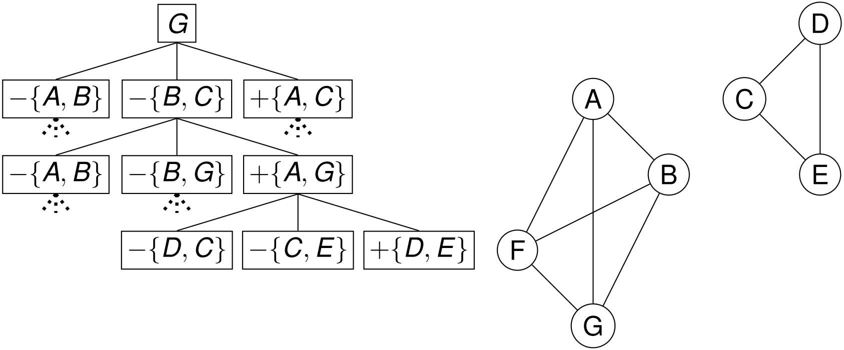
$\Rightarrow$ Found solution.

Example: P_3-free, $k = 3$



⇒ Found solution. If not: need to search the full tree.
If nothing found at level k: impossible with k edits.

Example: P_3-free, $k = 3$

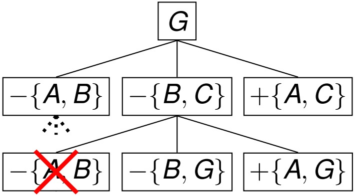


$\Rightarrow$ Found solution. If not: need to search the full tree.
If nothing found at level k: impossible with k edits.

- Time $O(3^k \cdot \text{poly}(n))$
- Best known: $O(1.62^k + m + n)$ [Böc12]

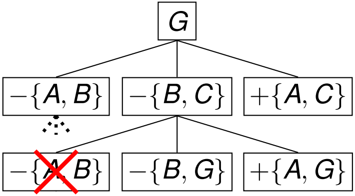
Branch-and-Bound Algorithm

- Increasing values of k to find exact k
  - show impossibility with $k - 1$
  - show solution with k
- Blocking: avoid duplicate enumeration [Dam08]
- Bounding: limit explored branches.



Branch-and-Bound Algorithm

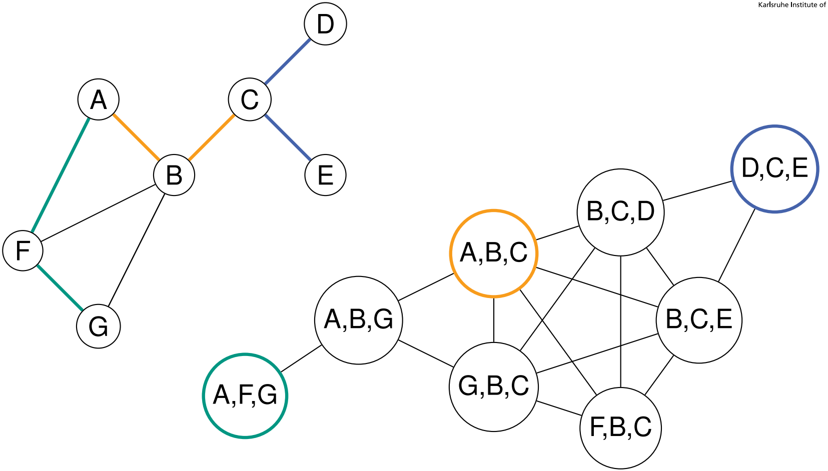
- Increasing values of k to find exact k
  - show impossibility with $k-1$
  - show solution with k
- Blocking: avoid duplicate enumeration [Dam08]
- Bounding: limit explored branches.



Our contribution:

- Good lower bounds
- Heuristic for selecting subgraphs to branch on

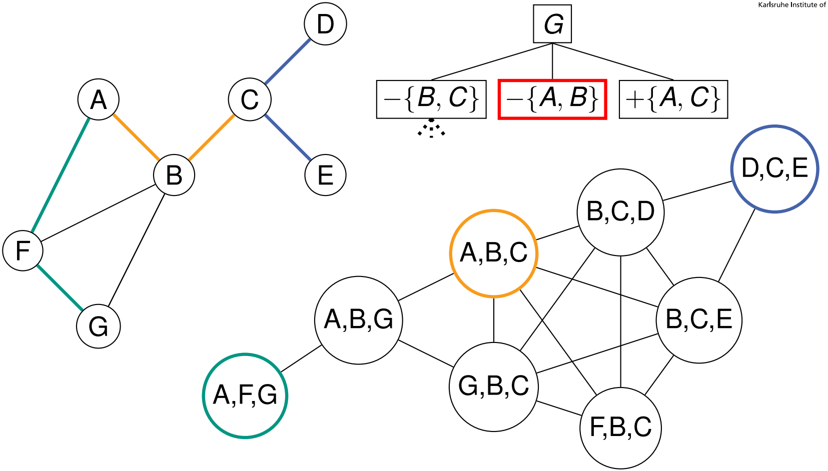
Lower Bound



- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

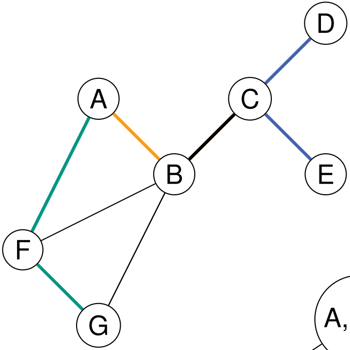
Lower Bound

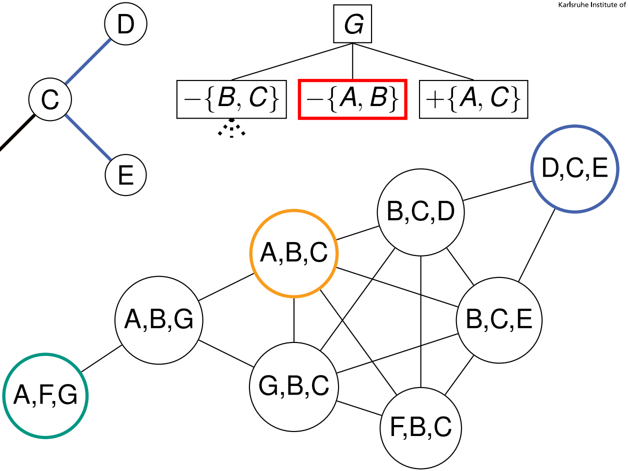


- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

Lower Bound

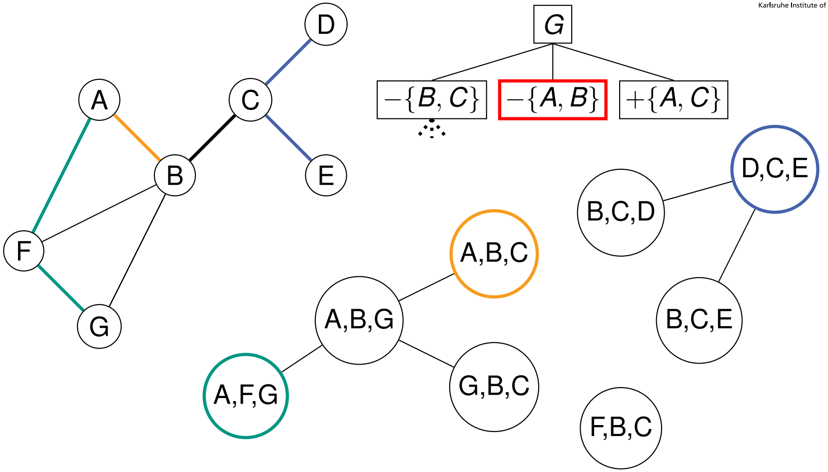




- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

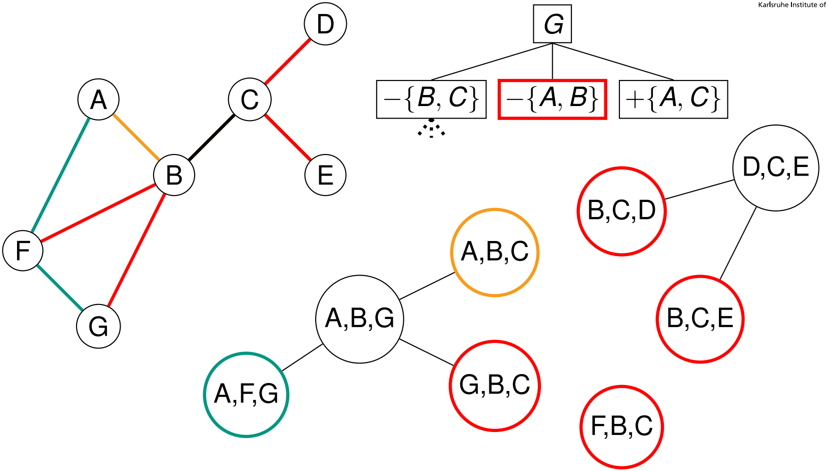
Lower Bound



- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

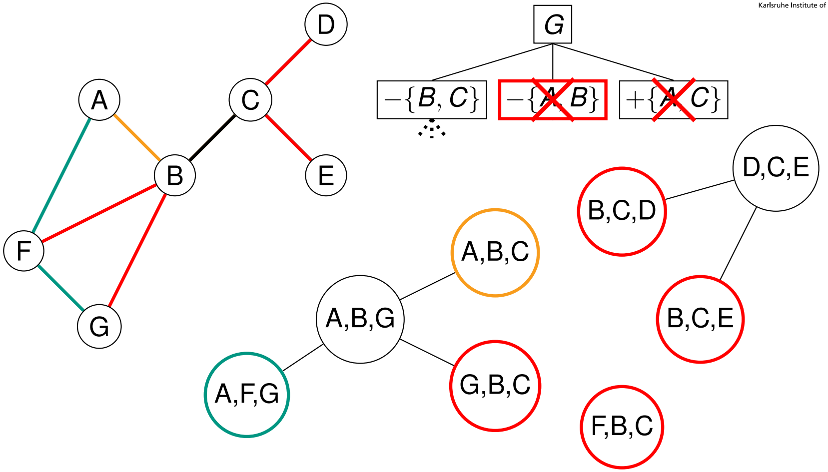
Lower Bound



- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

Lower Bound

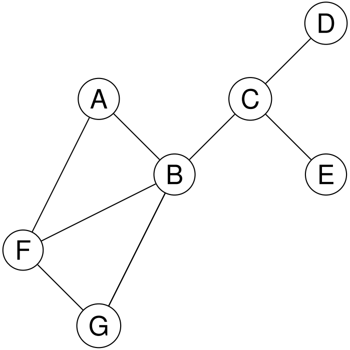


- Implicit conflict graph representation
- Use two-improvements and plateau search
- Incremental updates, exploit blocked edges

[ARW12]

Optimizing Branching Decisions

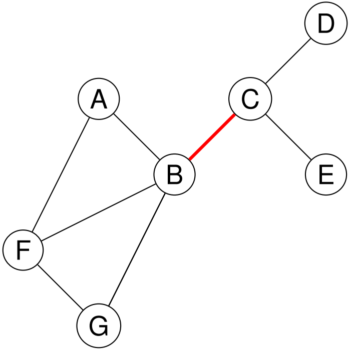
- Prune early: even before recursion
- First node pair = most likely edited node pair in **all** solutions



Optimizing Branching Decisions

- Prune early: even before recursion
- First node pair = most likely edited node pair in **all** solutions

Idea: Prefer node pairs that are part of many forbidden subgraphs



Integer Linear Programming

- Binary variables $X_{uv} \forall u, v \in V_G$
- $X_{uv} = 1 \iff$ edge $\{u, v\}$ exists

$$\text{minimize} \sum_{\{u,v\} \in E_G} (1 - x_{uv}) + \sum_{\{u,v\} \in \overline{E_G}} x_{uv}$$

$$\text{subject to} \sum_{\{u,v\} \in E_H} (1 - x_{\pi(u)\pi(v)}) + \sum_{\{u,v\} \in \overline{E_H}} x_{\pi(u)\pi(v)} \geq 1$$

$$\forall H \in \mathcal{F}, \forall \pi : V_H \hookrightarrow V_G$$

- Row generation
- LP relaxation is upper bound for lower bound

Experiments

Implementation:

- $\{P_4, C_4\}$-free editing
- Implemented in C++, parallelization using work stealing
- Gurobi for ILP
- `https://github.com/kit-algo/fpt-editing`

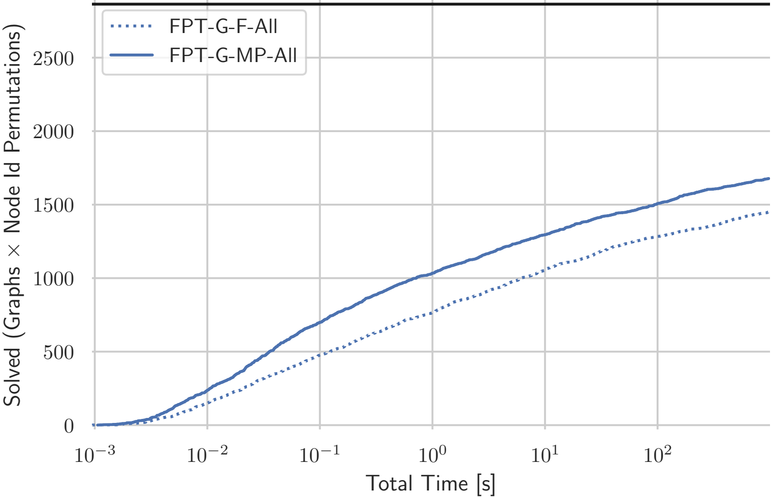
Data:

- 716 of 3964 connected components of COG protein similarity data – remaining need < 20 edits
- 4 node id permutations

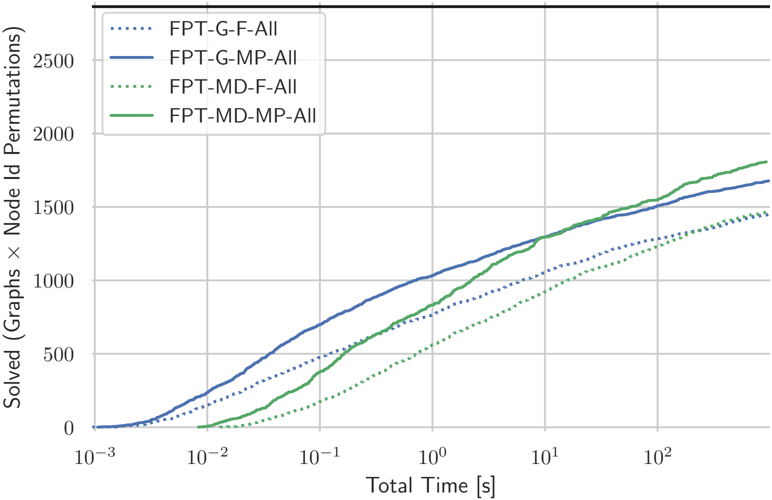
Setup:

- $2 \cdot 8$ core Intel Xeon E5-2670 (Sandy Bridge), 64 GB RAM
- 1000 seconds time limit

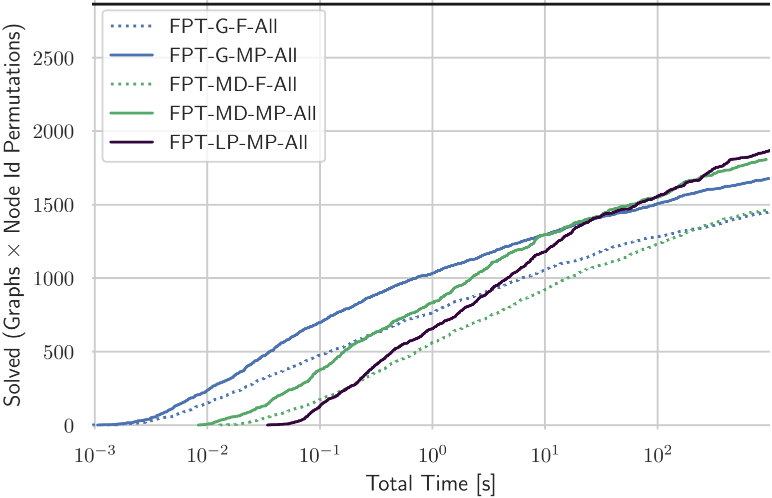
FPT Running Time



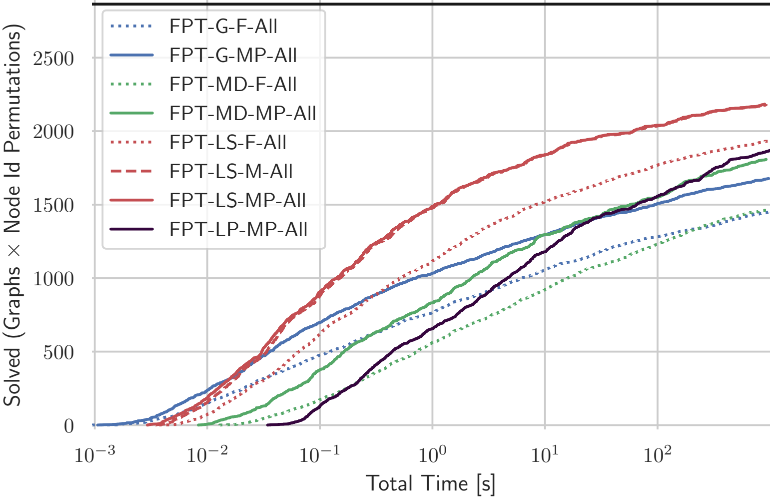
FPT Running Time



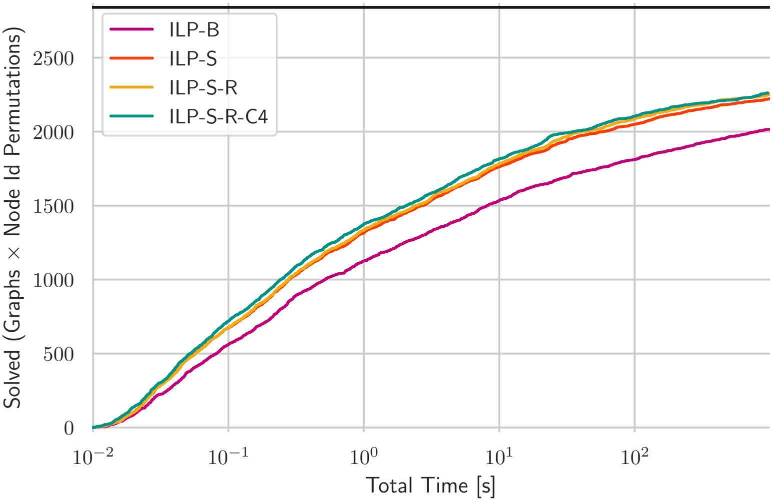
FPT Running Time



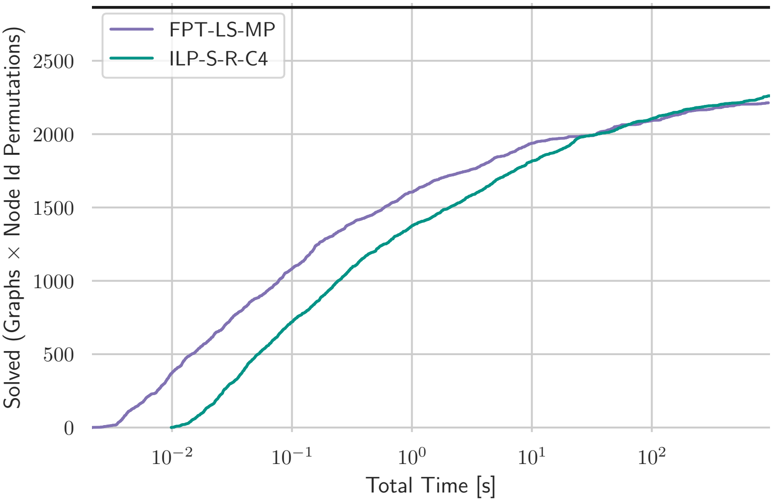
FPT Running Time



ILP Variants

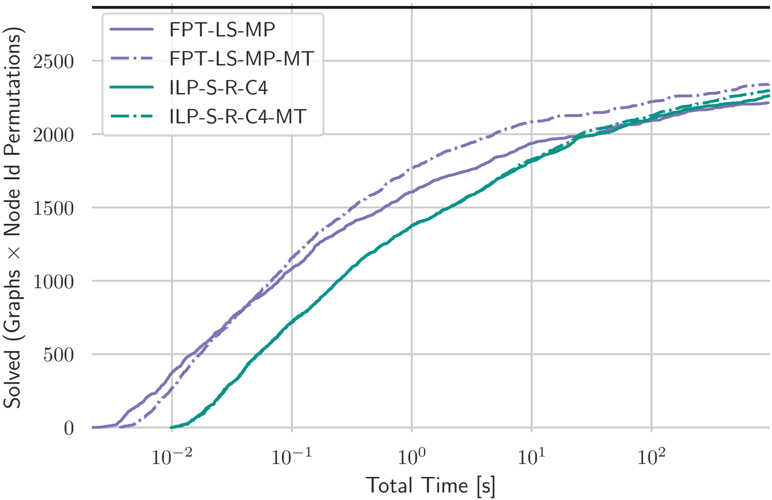


FPT vs. ILP

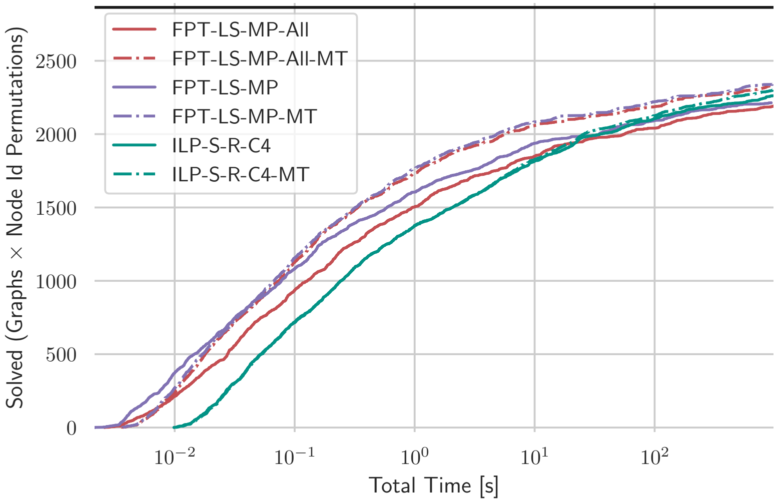


FPT-LS-MP
ILP-S-R-C4
Solved (Graphs × Node Id Permutations)
2500
2000
1500
1000
500
0
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Total Time [s]

FPT vs. ILP

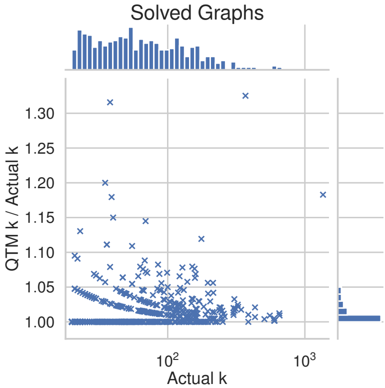

FPT-LS-MP
FPT-LS-MP-MT
ILP-S-R-C4
ILP-S-R-C4-MT
Solved (Graphs × Node Id Permutations)
0
500
1000
1500
2000
2500
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Total Time [s]

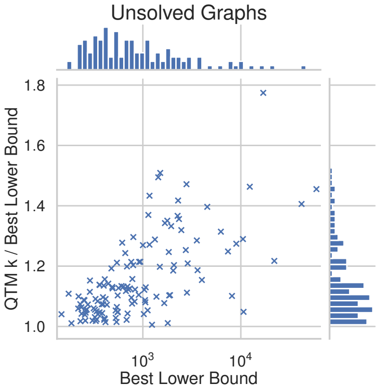
FPT vs. ILP



FPT-LS-MP-All
FPT-LS-MP-All-MT
FPT-LS-MP
FPT-LS-MP-MT
ILP-S-R-C4
ILP-S-R-C4-MT
Solved (Graphs × Node Id Permutations)
2500
2000
1500
1000
500
0
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
Total Time [s]

Comparison with QTM Heuristic





Not shown: Outlier at $k = 64$ where QTM needs 202 edits

Summary

- Carefully engineered branch and bound algorithm beats Gurobi

Future work:

- Adapt for edit costs
- Evaluate for other forbidden subgraphs

Bibliography I

Diogo V. Andrade, Mauricio G. C. Resende, and Renato F. Werneck.
Fast local search for the maximum independent set problem.
Journal of Heuristics, 18(4):525–547, 2012.

Ulrik Brandes, Michael Hamann, Ben Strasser, and Dorothea Wagner.
Fast Quasi-Threshold Editing.
In *Proceedings of the 23rd Annual European Symposium on Algorithms (ESA'15)*, Lecture Notes in Computer Science. Springer, 2015.

Sebastian Böcker.
A golden ratio parameterized algorithm for Cluster Editing.
Journal of Discrete Algorithms, 16:79–89, October 2012.

Leizhen Cai.
Fixed-parameter tractability of graph modification problems for hereditary properties.
Information Processing Letters, 58(4):171–176, May 1996.

Frank Pok Man Chu.
A simple linear time certifying LBFS-based algorithm for recognizing trivially perfect graphs and their complements.
Information Processing Letters, 107(1):7–12, June 2008.

Peter Damaschke.
Fixed-parameter enumerability of cluster editing and related problems.
Theory of Computing Systems, 46(2):261–283, 2008.

Pål Grønås Drange and Michał Pilipczuk.
A Polynomial Kernel for Trivially Perfect Editing.
Algorithmica, 80(12):3481–3524, December 2017.

Bibliography II

James Nastos and Yong Gao.
Familial groups in social networks.
Social Networks, 35(3):439–450, July 2013.

Number of Solutions