UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

HELSINKI INSTITUTE FOR INFORMATION TECHNOLOGY

Solving Graph Problems via Potential Maximal Cliques: Experimental Evaluation of the Bouchitté–Todinca Algorithm

Tuukka Korhonen, Jeremias Berg, and Matti Järvisalo

Challenge: The BT algorithm of Bouchitté and Todinca offers the theoretically best known approach to several important graph problems related to computing optimal graph triangulations. Due to its complexity, *less attention has been put on empirical studies of its efficiency*.

Contributions: Experimental evaluation of our implementation of the BT algorithm on five different graph problems.
We compare the BT algorithm to available implementations of other previously proposed exact approaches to each problem.

Results: The BT algorithm yields an empirically competitive approach to each of the considered problems.
Implementation: Available at https://github.com/Laakeri/Triangulator.

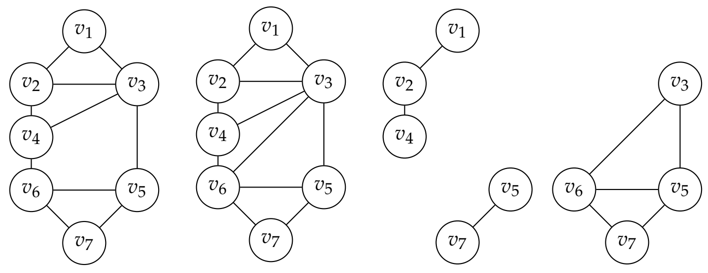
Potential Maximal Cliques and Graph Parameters

Consider the undirected simple graph G.

- $\omega = \{v_2, v_3, v_4\}$ is a *maximal clique* of G.
- G is not *chordal* due to chordless cycle (v_3, v_4, v_6, v_5).
- H is a *minimal triangulation* of G.
- $\Omega = \{v_3, v_4, v_6\}$ is a maximal clique of H.
  $\Rightarrow \Omega$ is a **potential maximal clique** (PMC) of G.
- $C_1 = \{v_5, v_7\}$ and $C_2 = \{v_1, v_2, v_4\}$ are connected components of $G \setminus \{v_3, v_6\}$
- The *neighborhoods* of C_1 and C_2 in G:
  $\mathrm{N}(C_1) = \mathrm{N}(C_2) = \{v_3, v_6\}$.
  $\Rightarrow C_1$ and C_2 are *full components* of $\{v_3, v_6\}$.
  $\Rightarrow \{v_3, v_6\}$ is a *minimal separator* of G.
  $\Rightarrow (S, C) = (\{v_3, v_6\}, \{v_5, v_7\})$ is a full block.
- (S, C) is associated with the PMC $\Omega = \{v_3, v_4, v_6\}$
- The realization $R(S, C)$ is the subgraph $G[S \cup C]$ of G with S as a clique.

Graph parameters $f(G)$
Many known graph problems can be defined in terms of minimal triangulations of G.

Example: Treewidth $f(G) = \min_H \max_{\omega \in H} |\omega| - 1$



Graph G | A minimal triangulation H | $G \setminus \{v_3, v_6\}$ | $R(S, C)$

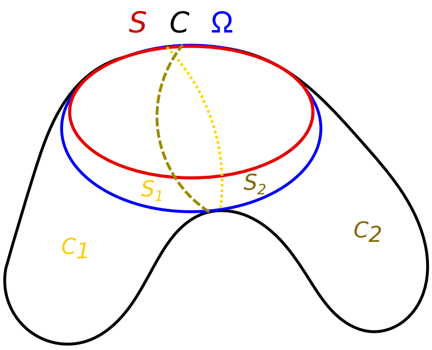
The BT Algorithm

Main ideas:

- Reduce the problem to finding an optimal triangulation of given G w.r.t. given f.
- Enumerate all PMCs and full blocks of G.
- Compute $f(G)$ via dynamic programming:
  states correspond to the full blocks and transitions to the PMCs.

More details:

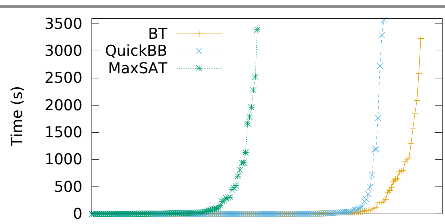
- Decompose the computation of $f(G)$ into those of $f(R(S, C))$ for full blocks.
- Computation of $f(R(S, C))$ decomposed into computing $f(R(S_i, C_i))$ for full blocks $|(S_i, C_i)| < |(S, C)|$ by considering filling PMCs with $S \subsetneq \Omega \subset (S, C)$.
- Child states of $R(S, C)$ w.r.t. Ω: full blocks associated with Ω contained in (S, C).
- Time complexity: $O(\#\text{PMCs} \cdot T_f(n)nm)$, where $T_f(n)$ depends on the parameter f.
  Example: $T_f(n) = O(1)$ for treewidth.

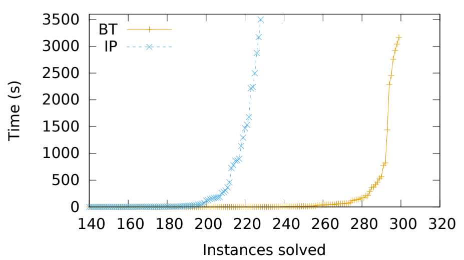


RESULTS

Graph Problems:

1. Treewidth (top)
   Minimize size of largest clique over triangulations
2. Minimum fill-in (bottom)
   Minimize number of extra edges over triangulations
3. Total table size (table)
   Minimize total size of CPTs of a Bayesian network
4. Generalized hypertreewidth
5. Fractional hypertreewidth





Instance	$\lvert V\rvert$	$\lvert E\rvert$	TTS	Time (seconds) BT	EDFS
alarm	37	65	996	**0.01**	0.35
andes	223	626	-	TO	TO
barley	48	126	17140796	**2.25**	4458.14
child	20	30	642	**0.01**	0.31
diabetes	413	819	-	TO	TO
hailfinder	56	99	9406	**0.08**	3.28
hepar2	70	158	2617	**0.01**	0.42
insurance	27	70	23880	**0.03**	0.83
mildew	35	80	3400464	**0.51**	2.95
munin1	186	354	-	TO	TO
pathfinder	109	208	182641	**0.08**	3.04
pigs	441	806	-	TO	TO
water	32	123	3028305	**0.15**	2.80
win95pts	76	225	2684	**0.22**	10.88