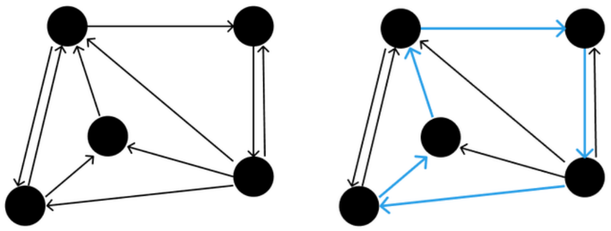
Ágoston Szabó (agostonszabo@tudelft.nl)
Responsible Professor: Emir Demirović
Supervisor: Imko Marijnissen

Articulation-Based Propagation for the Circuit Constraint

1. Motivation

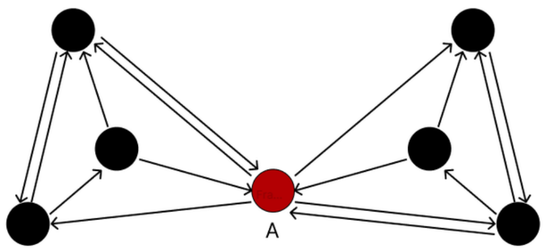
Constraint Programming: a paradigm for solving combinatorial problems using constraints and propagation



Circuit Constraint: Find one Hamiltonian cycle through all vertices

Existing propagators

- Prevent premature cycles
- Local reasoning



(Strong) Articulation points can detect global infeasibility early

2. Main Idea

Extend existing propagator

- Detect (strong) articulation points
- Force edges
- Prune edges
- Detect conflicts earlier

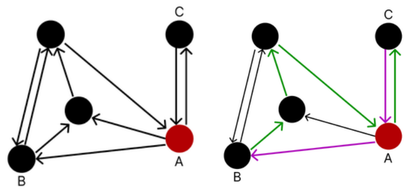
LCG Explanations

- Record all removed edges
- Re-verify connectivity property

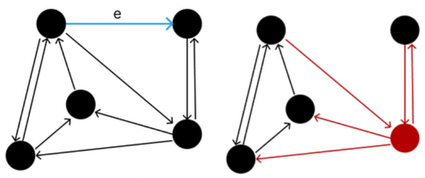
3. Propagation Workflow

Support Graph → Find Articulation Points → Conflict Detection → Edge Forcing → Find Strong Articulation Points → Conflict Detection → Edge Pruning → Smaller Search Tree

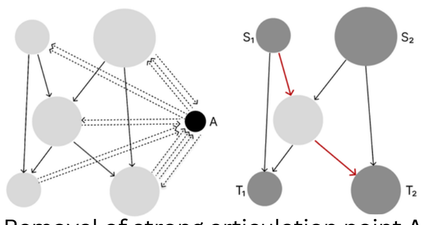
4. Reasoning



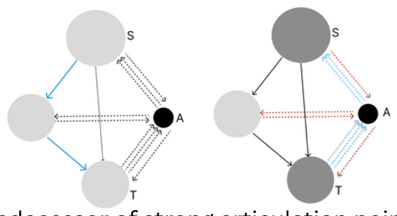
Articulation point A causes conflict



Edge e is forced otherwise conflict



Removal of strong articulation point A introduces 2 source (S_1, S_2), 2 sink(T_1, T_2) SCCs, longest path of 2 (< 4) => conflict



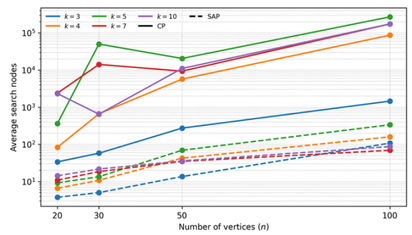
Predecessor of strong articulation point A must be in T, successor in S => Other edges from/to A can be pruned

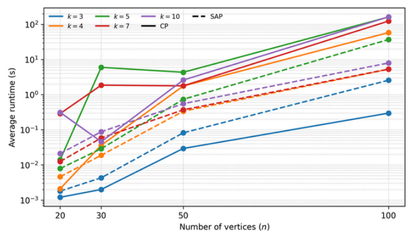
5. Experiments

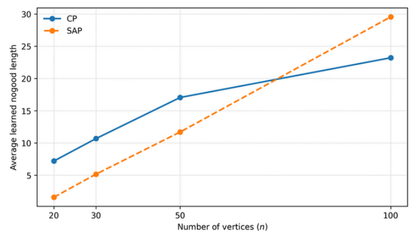
- Pumpkin LCG solver
- n = 20–100 vertices
- varying k for different densities
- 400 benchmark instances
- Compared to cycle prevention

Up to:

- **2510×** fewer search nodes
- **7420×** fewer failures
- **23×** faster runtime
- **77%** fewer timeouts







Search Nodes

- Orders-of-magnitude search reduction
- Benefits increase with graph size
- Up to 2500× fewer nodes

Runtime

- Less search outweighs analysis overhead
- Faster on most benchmark classes
- Up to 23× speedup

Learned Nogoods

- Slightly larger explanations on large graphs
- Expected due to global graph reasoning

6. Conclusion

- Connectivity bottlenecks provide powerful propagation
- SAP reasoning substantially reduces search effort
- Up to 2500× fewer search nodes
- Up to 7400× fewer failures

7. Future Work

- Incremental graph analysis
- Explanation minimization
- Adaptive activation based on graph density
- Evaluation on real-world routing benchmarks