Better Caching for Better Model Counting

Jeroen Rook, Anna Latour, Holger Hoos and Siegfried Nijssen.

Universiteit
Leiden
The Netherlands

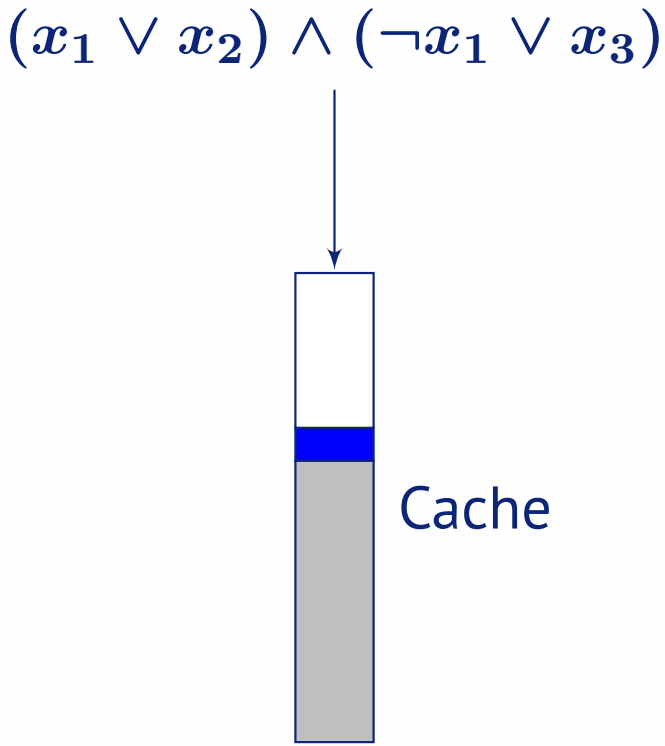
Exact Model Counting

- DPLL
- Knowledge compilation
- Tree decomposition
- **CDCL with component caching**

Contributions

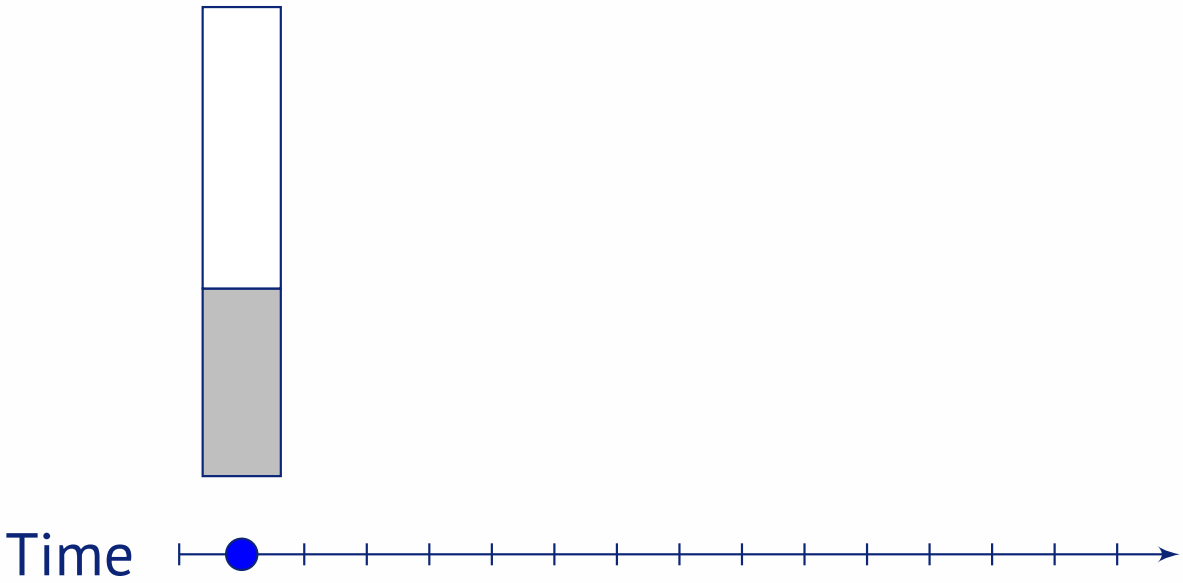
- Parameter configuration on a highly configurable version of Ganak under constrained memory. [Sharma et al.,2019]
- Exploratory analysis of caching behaviour.

Component caching



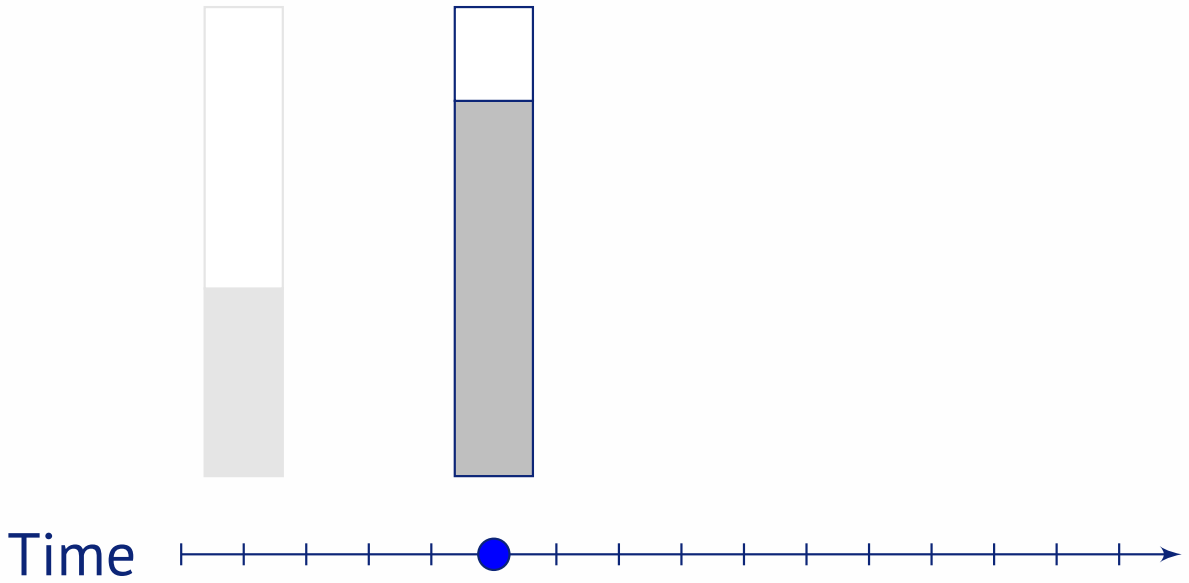
Component caching

- Cachet [Sang et al.,2004]
- sharpSAT 2013 [Thurley,2006]
- GANAK [Sharma et al.,2019]



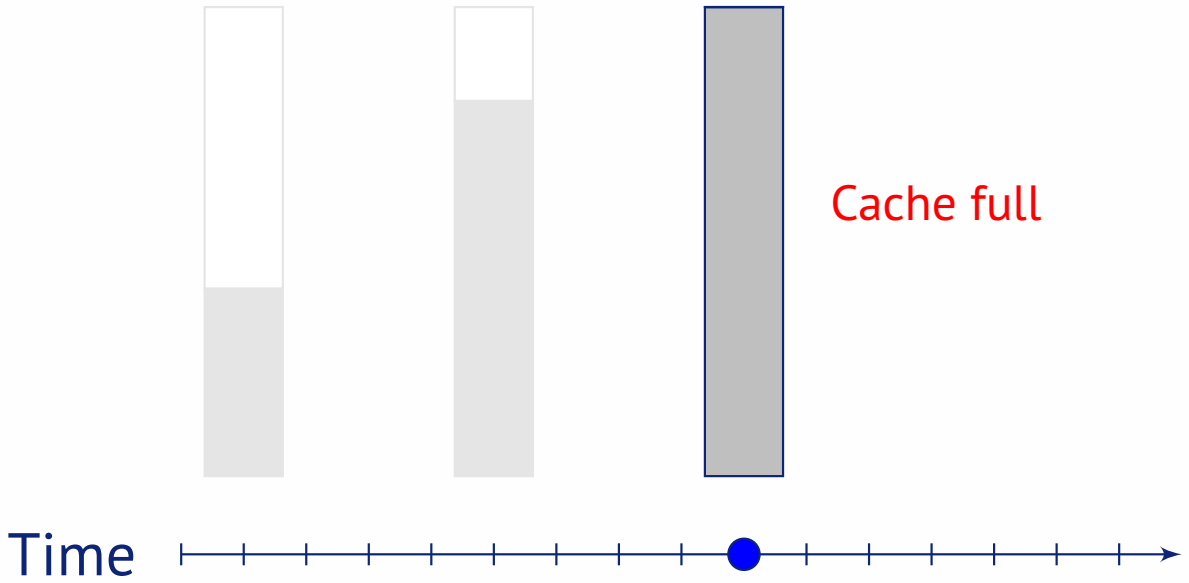
Component caching

- Cachet [Sang et al.,2004]
- sharpSAT 2013 [Thurley,2006]
- GANAK [Sharma et al.,2019]



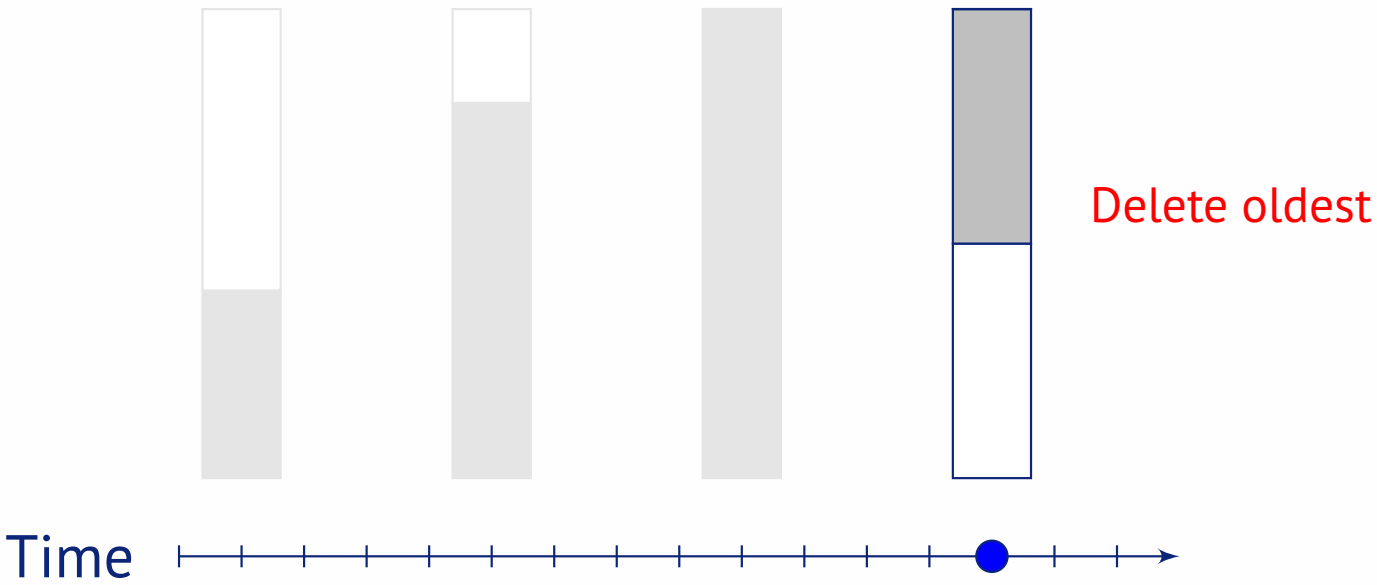
Component caching

- Cachet [Sang et al.,2004]
- sharpSAT 2013 [Thurley,2006]
- GANAK [Sharma et al.,2019]



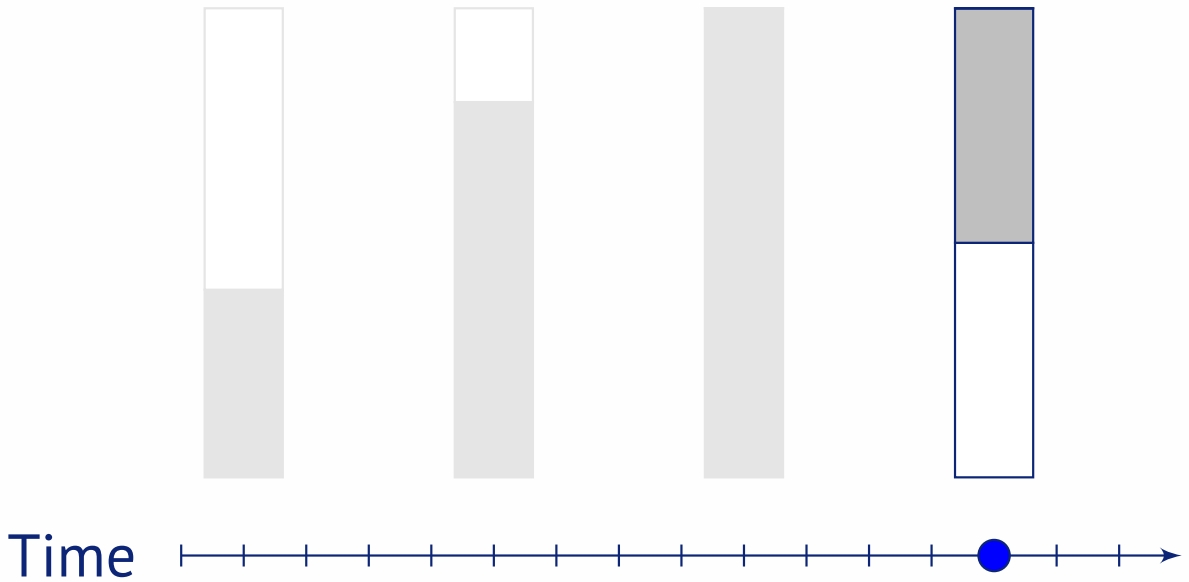
Component caching

- Cachet [Sang et al.,2004]
- sharpSAT 2013 [Thurley,2006]
- GANAK [Sharma et al.,2019]

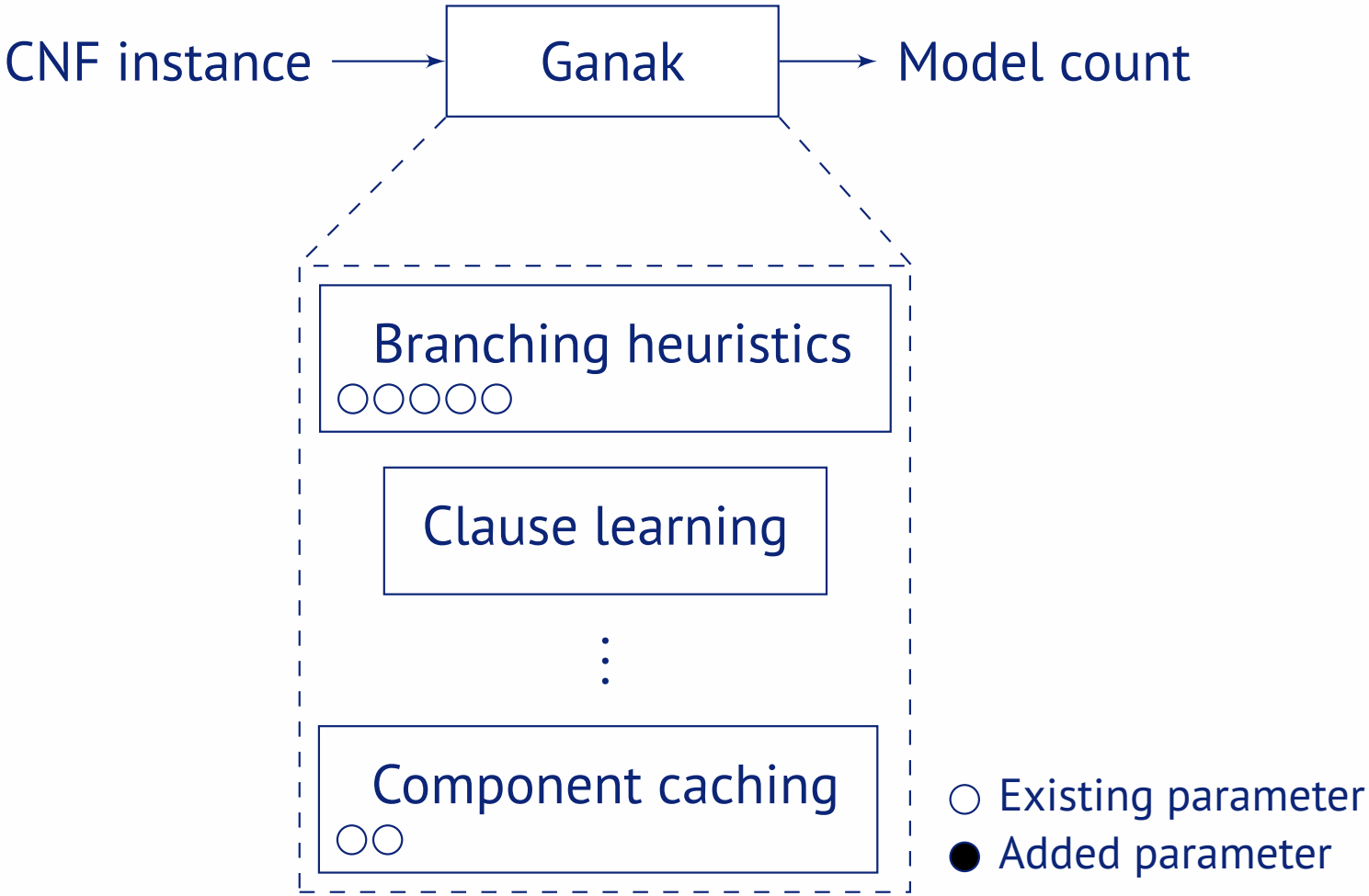


Component caching

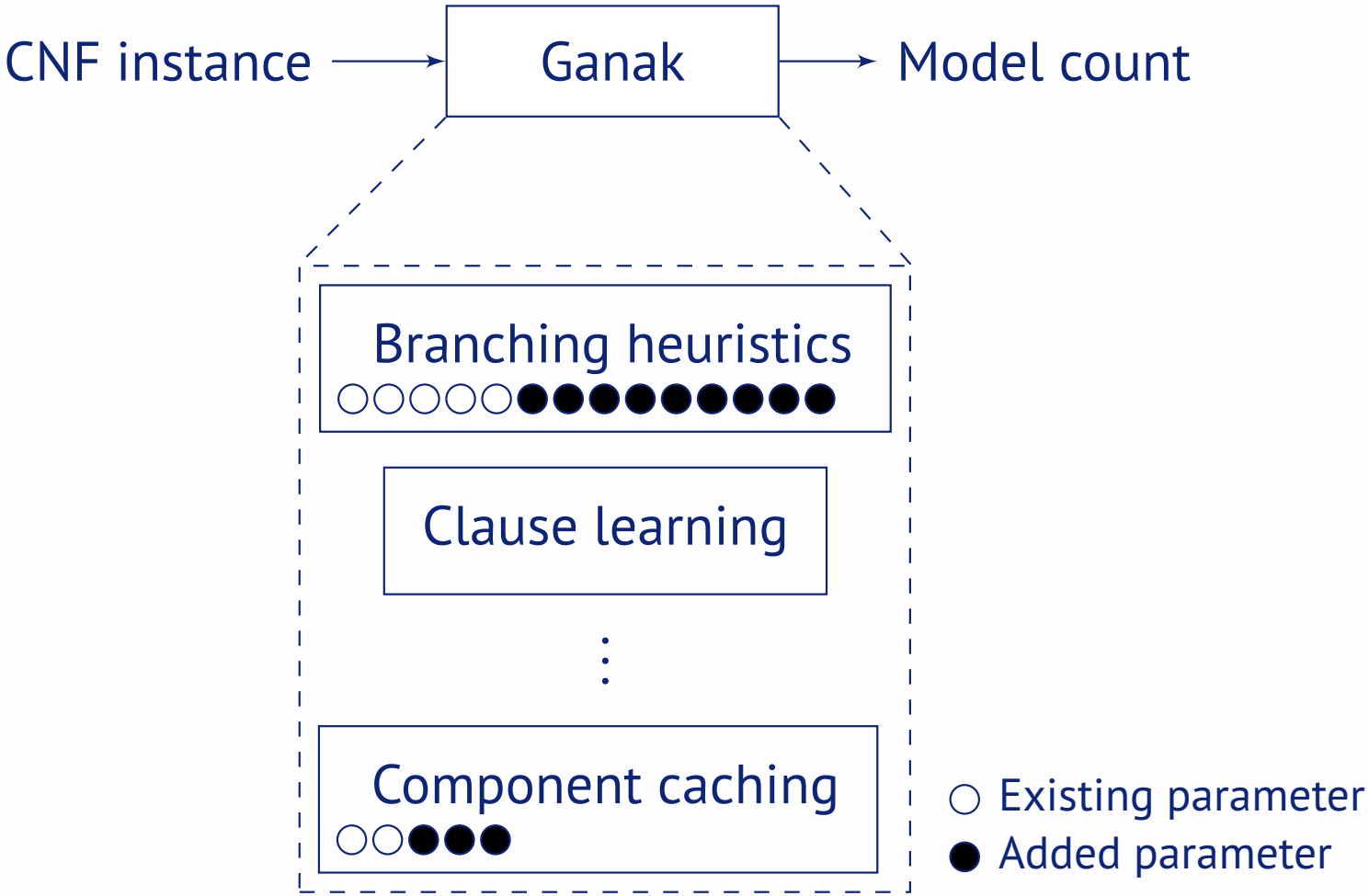
- Removed components need to be solved again when they are seen again.
- Every found component is cached, regardless of their reusability.



Ganak


CNF instance
Ganak
Model count
Branching heuristics
Clause learning
Component caching
Existing parameter
Added parameter

Ganak


CNF instance
Ganak
Model count
Branching heuristics
Clause learning
Component caching
Existing parameter
Added parameter

Automated Algorithm Configuration

Given an **algorithm** with **parameter space** find a configuration that minimises a **target measure** on a given **instance set**.

[Hoos,2012]

Benchmarks

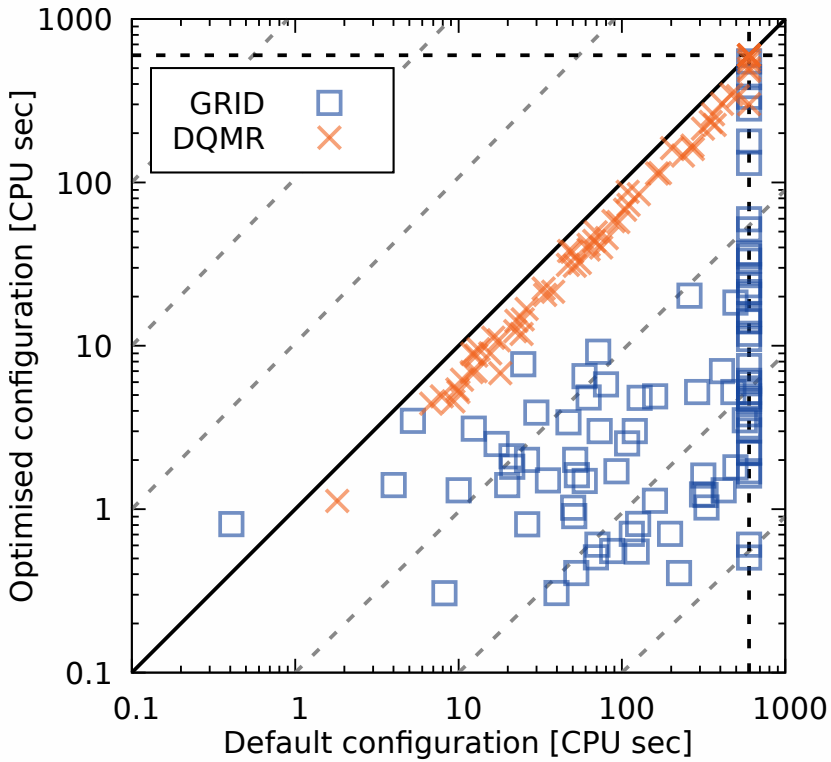
- Unweighted DQMR and GRID benchmarks. [Sang et al.,2005]
- Solving time between $\sim 5s$ and $3600s$ with default parameters.
- Cache size of 100MB to ensure limited cache behaviour.

Automated Algorithm Configuration

Cache size **100MB**, cutoff time **600s**

	GRID	90
	PAR10	timeout
Default	2483	36
Optimised	28	0

	DQMR	62
	PAR10	timeout
Default	949	9
Optimised	481	4

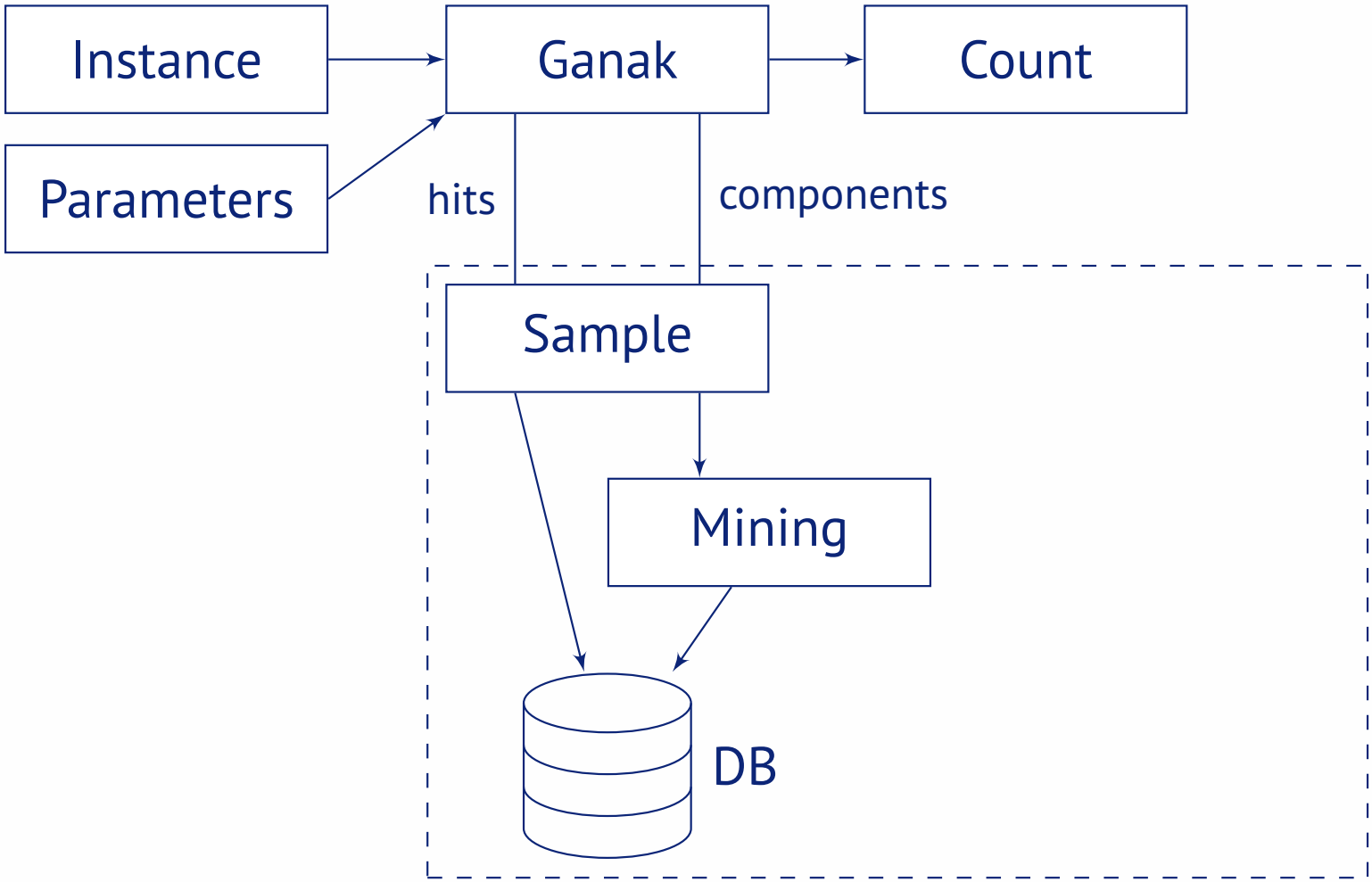


AAC Observations

- Different instances are affected by different parameters.
- Branching heuristics contribute most to the decrease of PAR10.

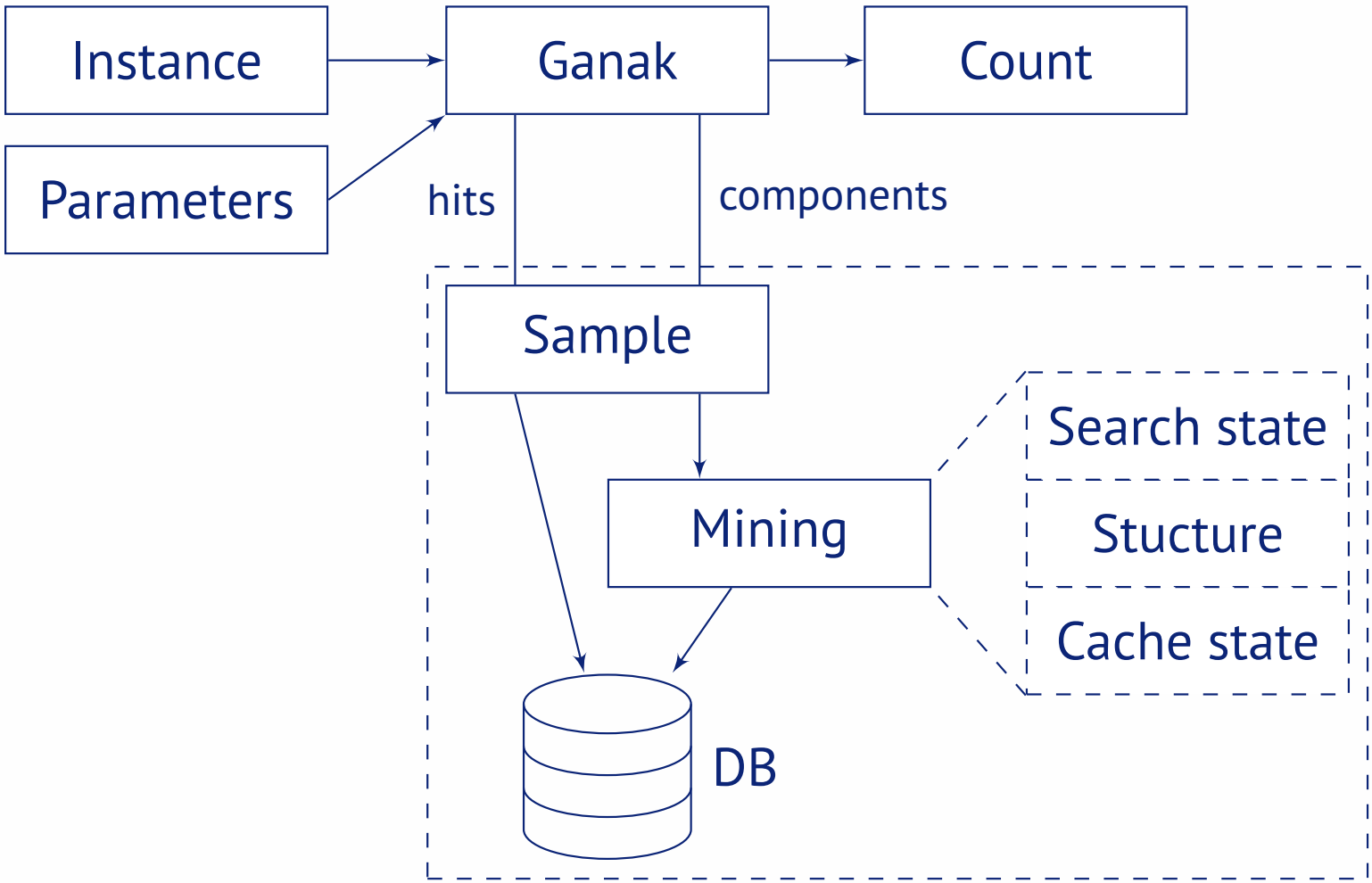
Mining the cache

Inspired by CrystalBall [Soos et al.,2019]



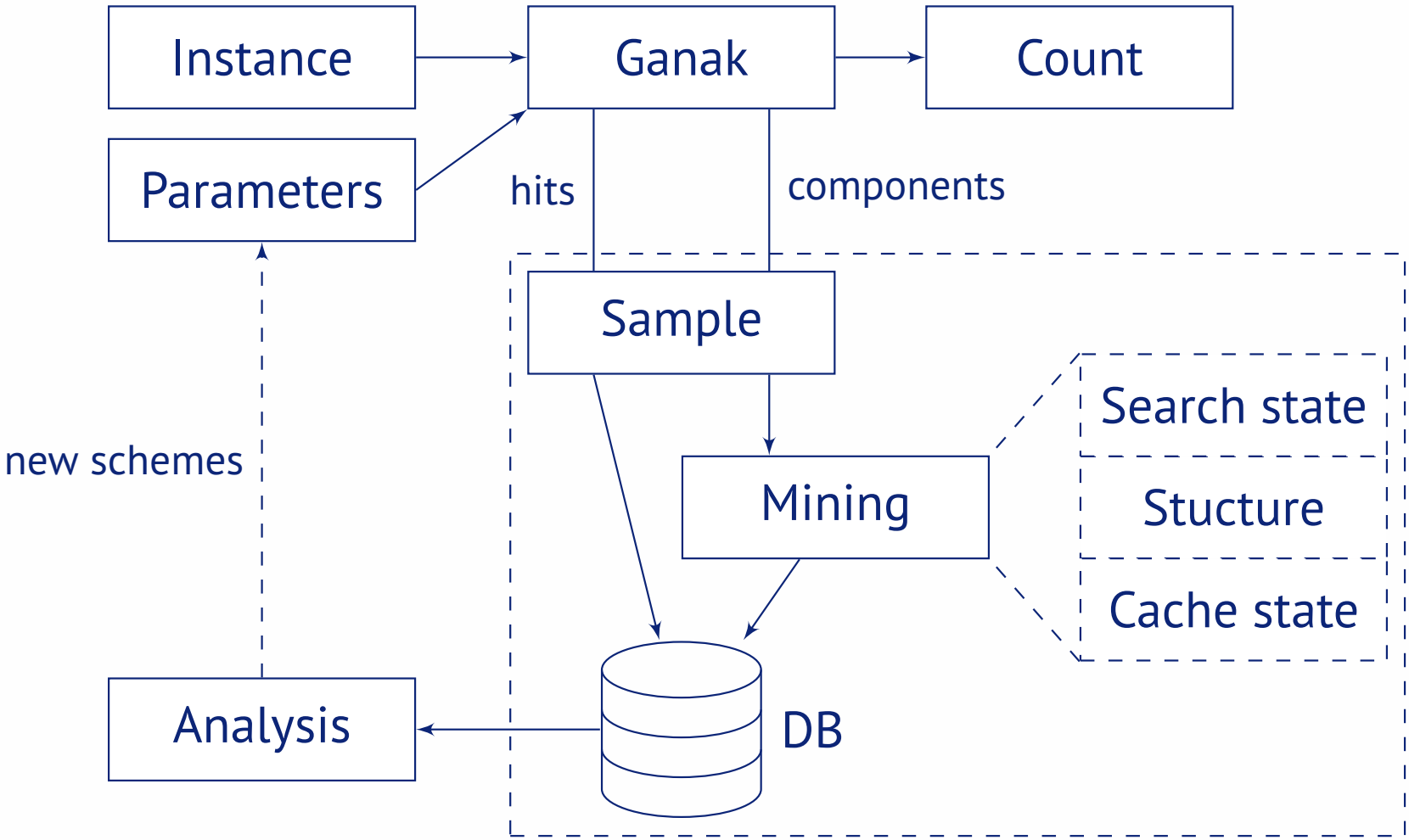
Mining the cache

Inspired by CrystalBall [Soos et al.,2019]



Mining the cache

Inspired by CrystalBall [Soos et al.,2019]



Initial analysis

For `or-60-20-3.cnf`

How much of all cached components are actually reused?

Initial analysis

For `or-60-20-3.cnf`

How much of all cached components are actually reused?

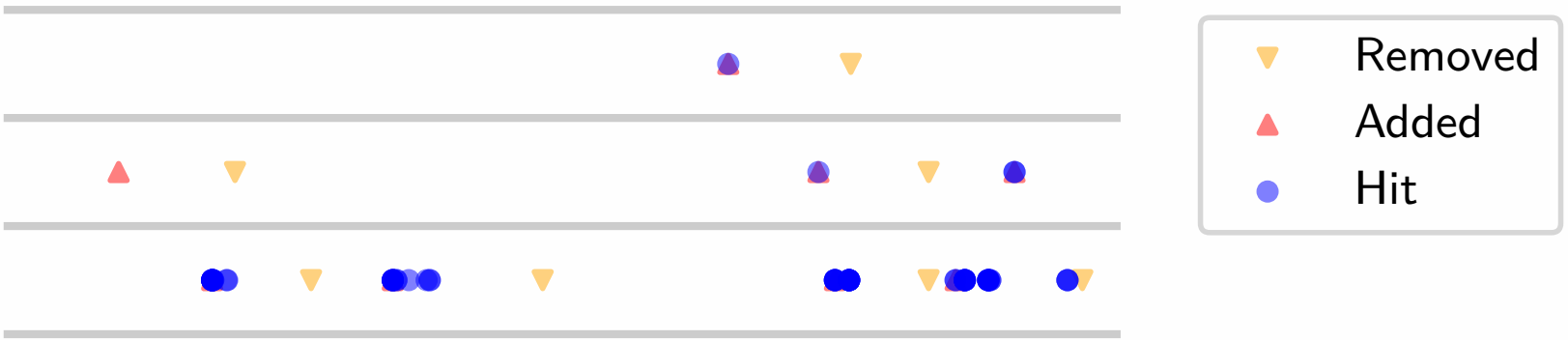
What is the impact of cache removal on the number of decisions?

	Decisions	
Total	4,078,392	
Recomputation	910,030	*22.3%*

Initial analysis

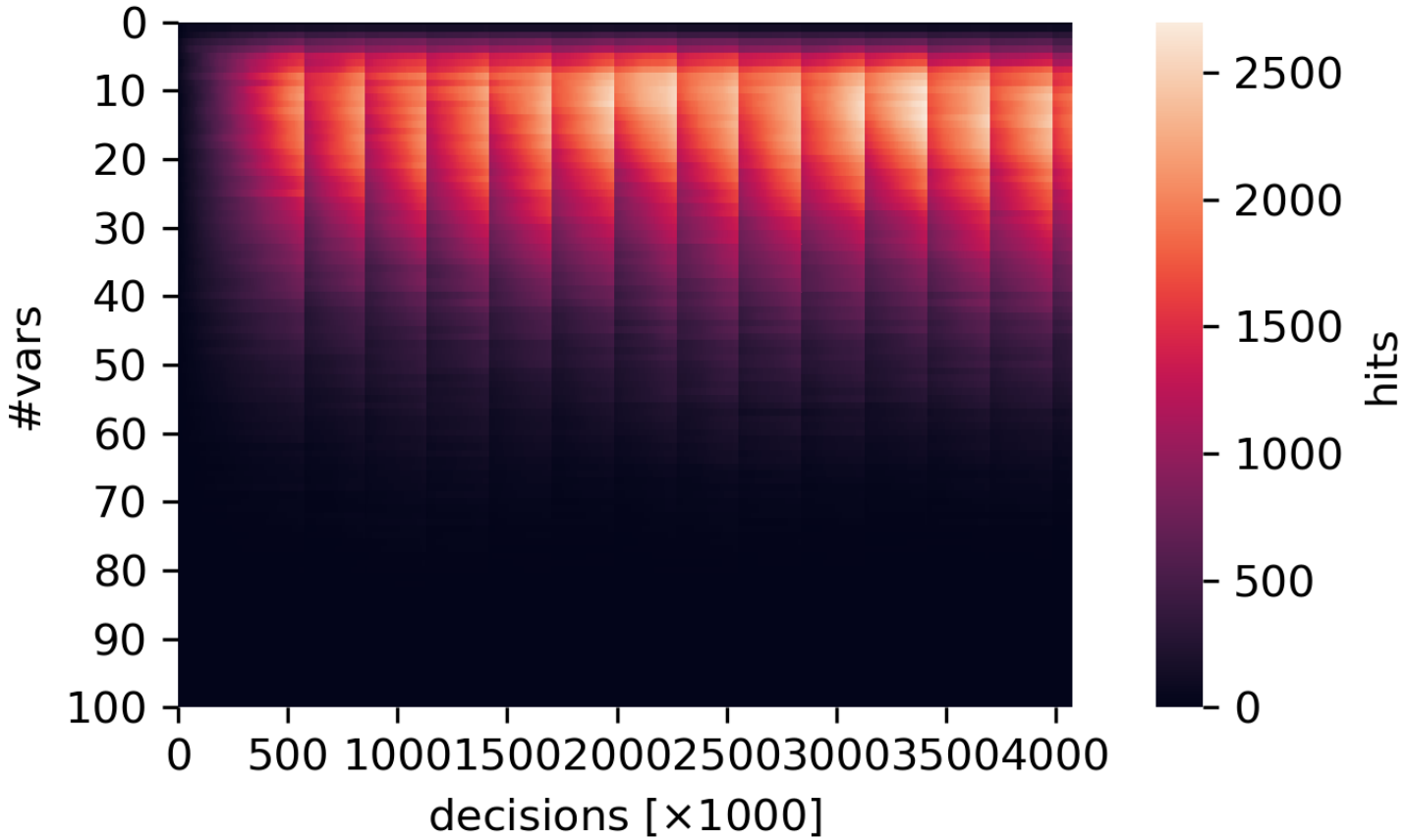
For `or-60-20-3.cnf`

Can we characterize different component profiles?



Initial analysis

For `or-60-20-3.cnf`



Future work

- Include component dependencies.
- Leverage machine learning to predict component reusability.
- Scale to harder and more diverse benchmark sets.

Take aways

- Current caching schemes can negatively impact solver running time performance.
- Only a small fraction of cached components is frequently reused.
- Cache behaviour is influenced by instances and solver parameters.

Contact:
Jeroen Rook
j.g.rook@umail.leidenuniv.nl

References I


Shubham Sharma, Subhajit Roy, Mate Soos, and Kuldeep S. Meel.
GANAK: A Scalable Probabilistic Exact Model Counter.
IJCAI 2019


Tian Sang, Paul Beame, and Henry A. Kautz.
Performing Bayesian Inference by Weighted Model Counting.
AAAI 2004


Marc Thurley.
sharpSAT - Counting Models with Advanced Component Caching and ImplicitBCP.
SAT 2006


Holger H. Hoos.
Programming by optimization.
Communications of the ACM 2012


Mate Soos, Raghav Kulkarni, and Kuldeep S. Meel.
CrystalBall: Gazing in the Black Box of SAT Solving.
SAT 2019

Theme by Joost Schalken. Updated by Pepijn van Heiningen & Anna Louise Latour.