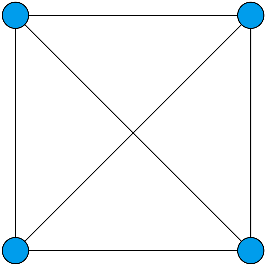
Understanding the Empirical Hardness of Random Optimisation Problems

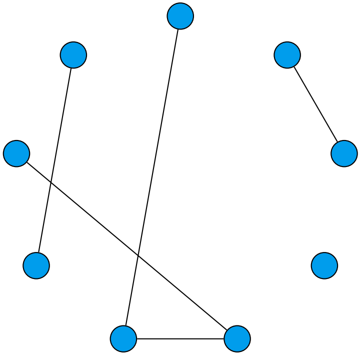
Ciaran McCreesh William Pettersson Patrick Prosser

University of Glasgow

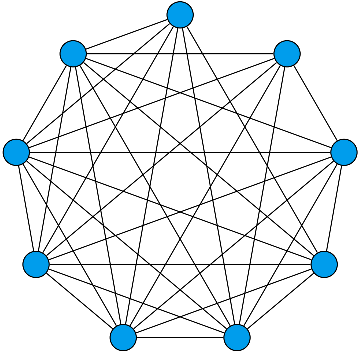
Cliques



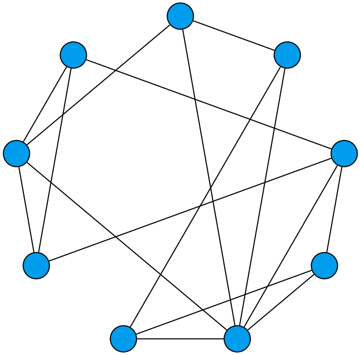
Cliques



Cliques

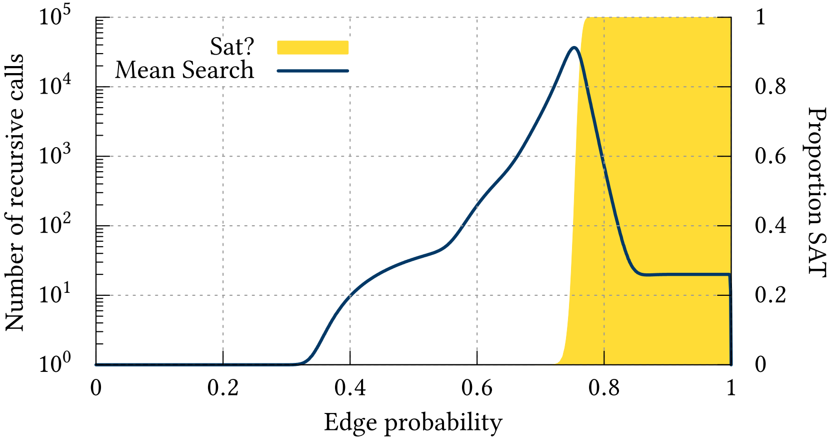


Cliques



Cliques

Does $G(150, x)$ contain a clique of twenty vertices?



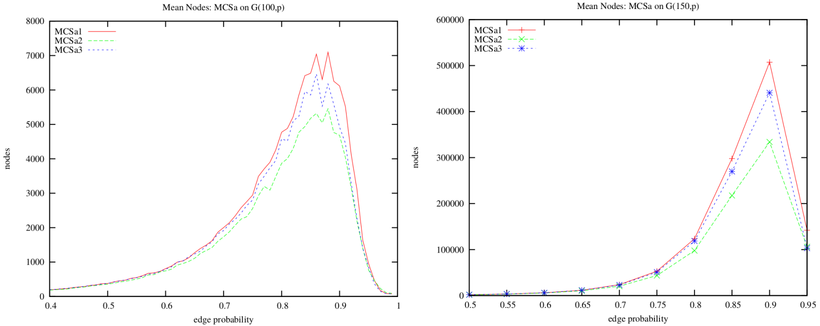
Intuition

- Low density means no occurrences, and we can quickly show we run out of edges after doing a bit of branching.
- High density means lots of occurrences, so wherever we look, it's easy to find one of them.
- If we expect there to be just one solution, it's really hard to find it if it exists, and really hard to rule it out if it doesn't exist.

Intuition

- Low density means no occurrences, and we can quickly show we run out of edges after doing a bit of branching.
- ~~High density means lots of occurrences, so wherever we look, it's easy to find one of them.~~
- If we expect there to be just one solution, it's really hard to find it if it exists, and really hard to rule it out if it doesn't exist.

Optimisation, an Incomplete Picture



Patrick Prosser: Exact Algorithms for Maximum Clique: A Computational Study. Algorithms 5(4): 545-587 (2012). Both plots have 100 samples per density step. The left-hand plot seems to go up in density steps of 0.01, and the right-hand plot, 0.05.

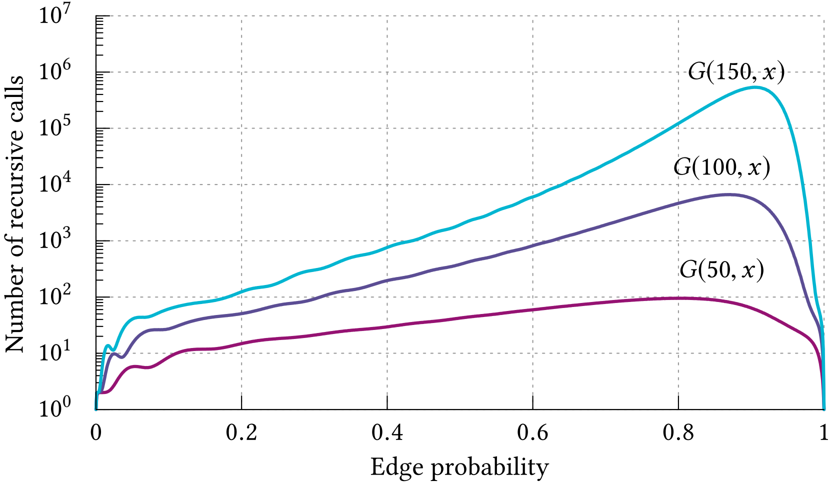
Which is the Hardest Density?

- Which density is hardest, for the optimisation problem?
- Does this change depending upon the number of vertices? The algorithm used? The random graph model selected?
- Is this the same as the hardest density for the decision problem, if we can also pick the decision number? And if so, which decision number do we pick?

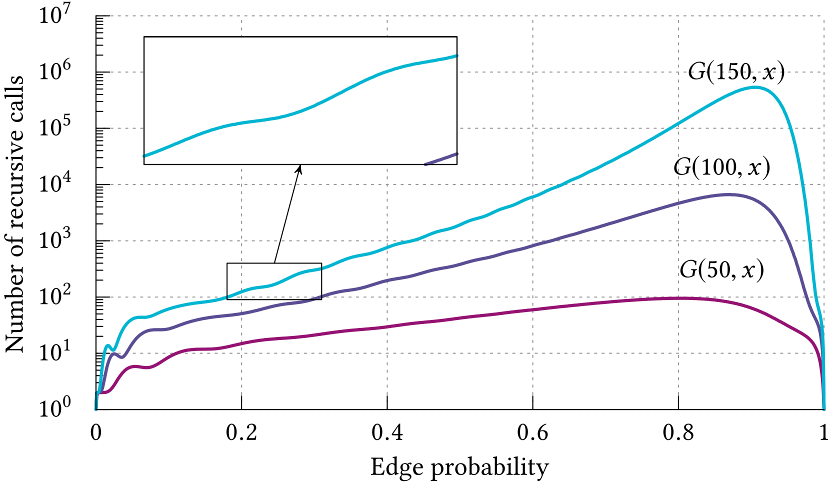
Really Big Experiments

- Increase density from 0 to 1 in steps of 0.001? This is around one pixel per step.
- Mean runtimes seem to settle down at around 10,000 samples. We probably want 100,000 samples to be safe.
- Back of the envelope feasibility estimates: 18 years.
- Conveniently, this is around 150,000 core hours.
  - …And the rest of the paper is a bit below 1,000,000 core hours.

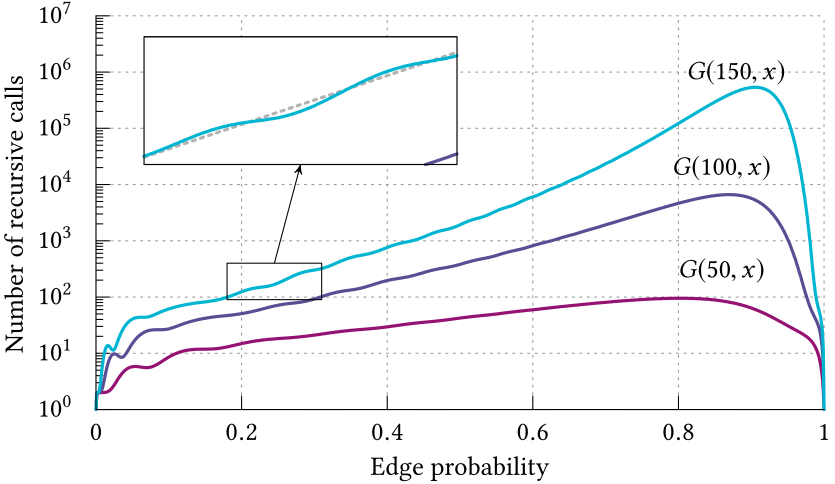
Optimisation, Refined



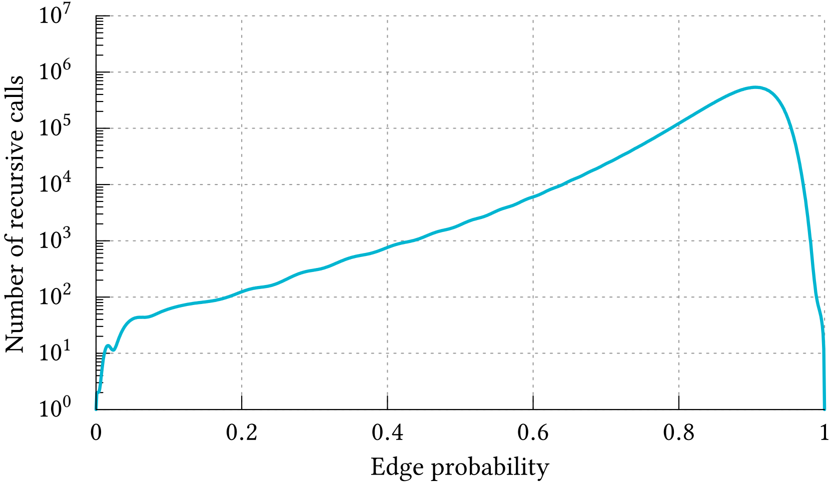
Optimisation, Refined



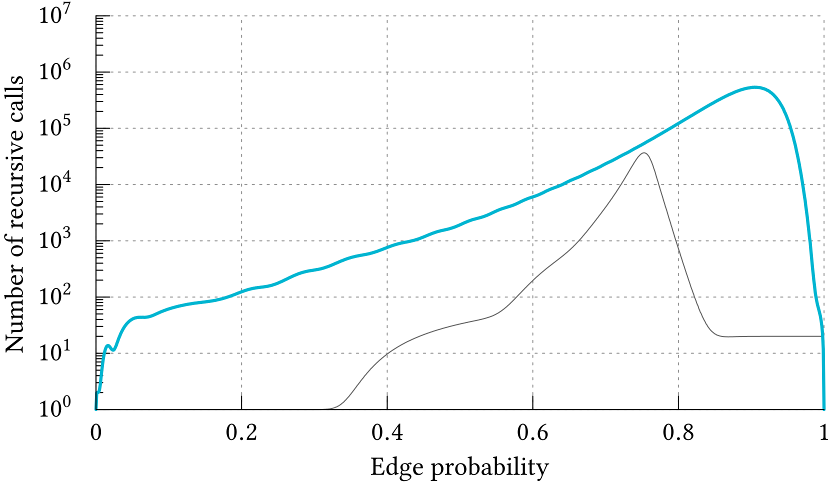
Optimisation, Refined



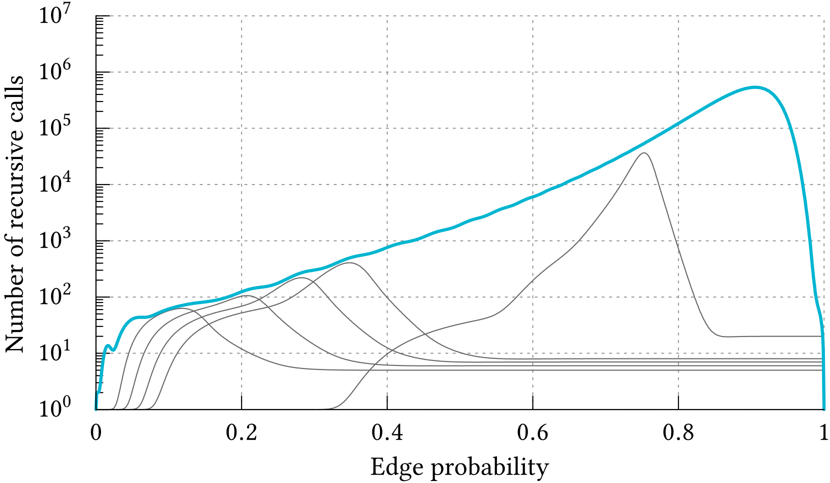
Optimisation versus Decision



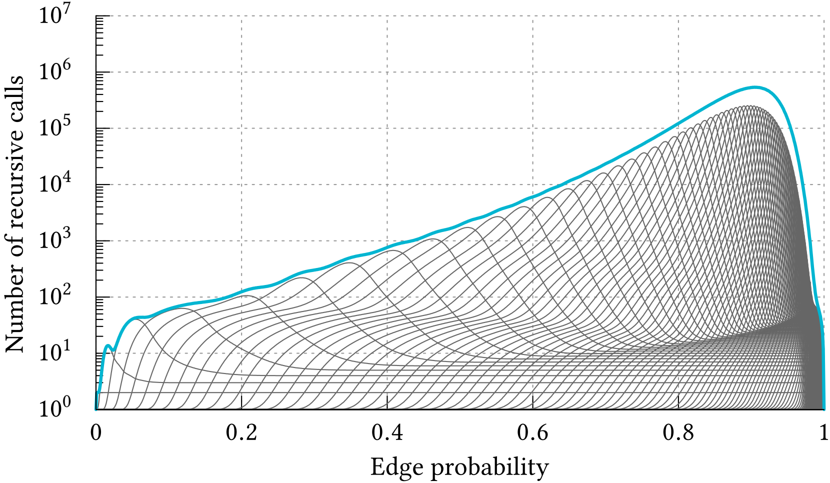
Optimisation versus Decision



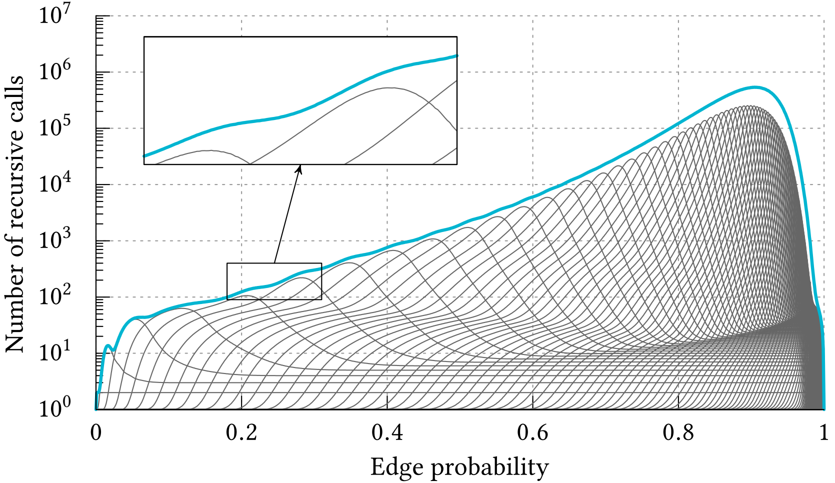
Optimisation versus Decision



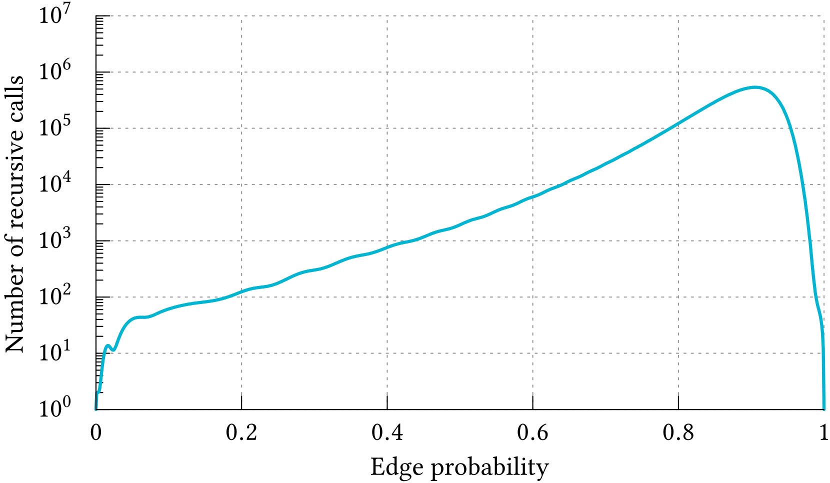
Optimisation versus Decision



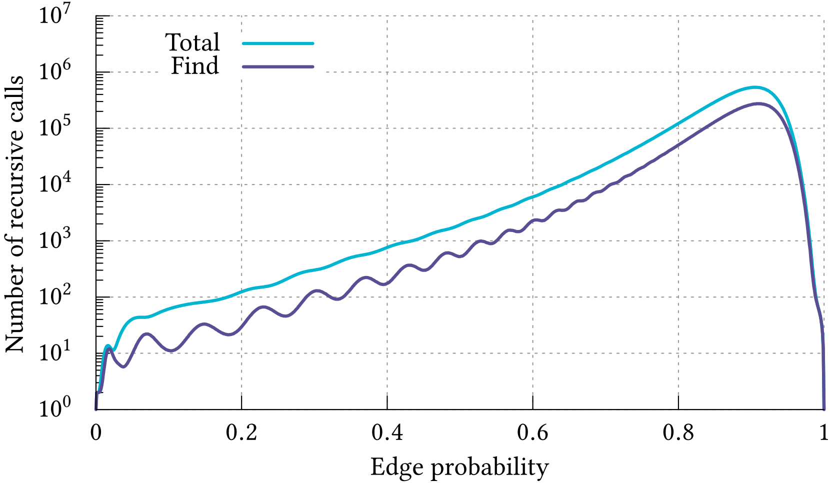
Optimisation versus Decision



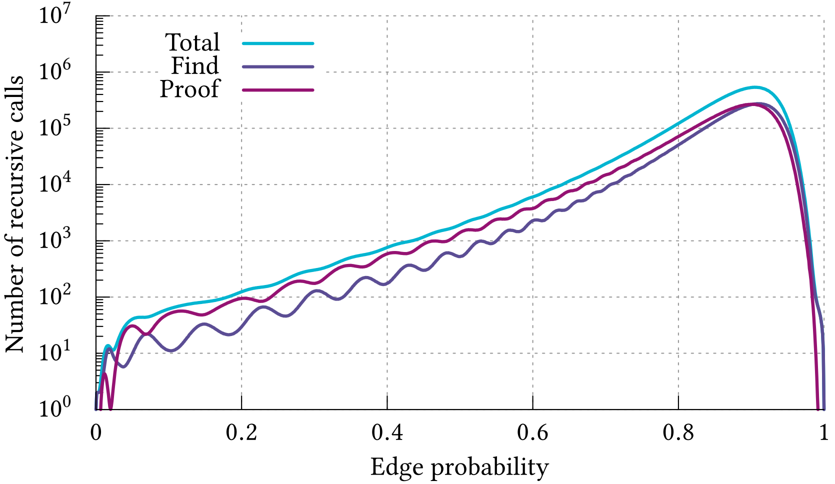
Finding versus Proving Optimality



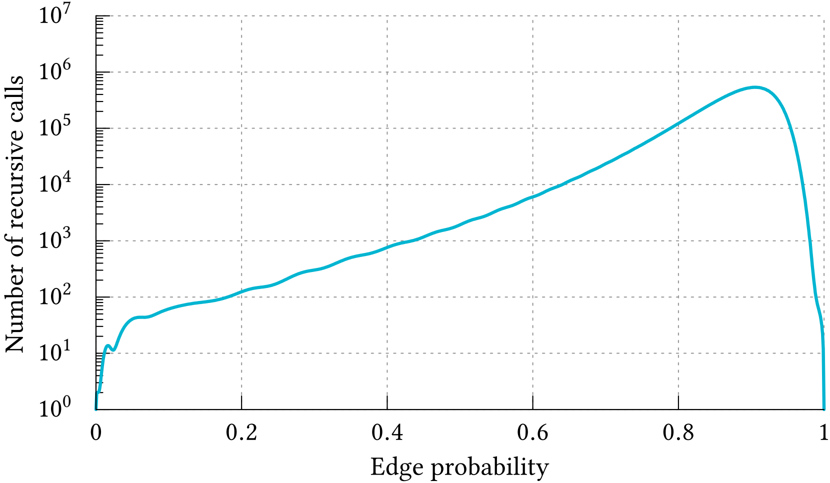
Finding versus Proving Optimality



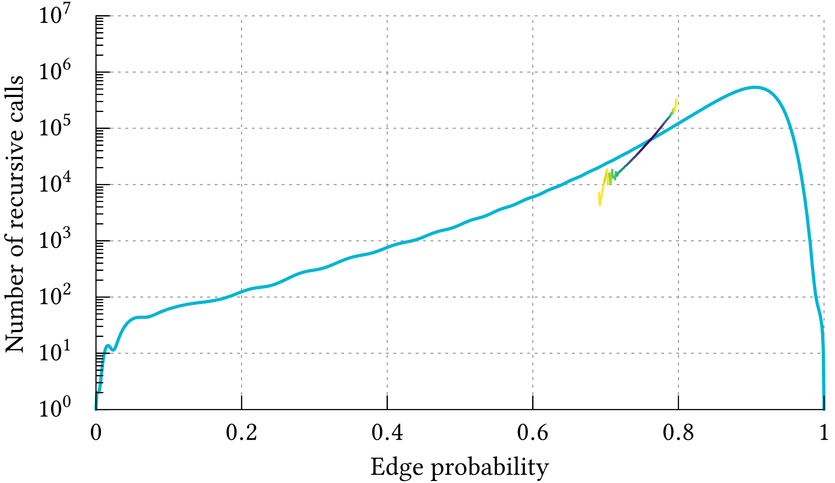
Finding versus Proving Optimality



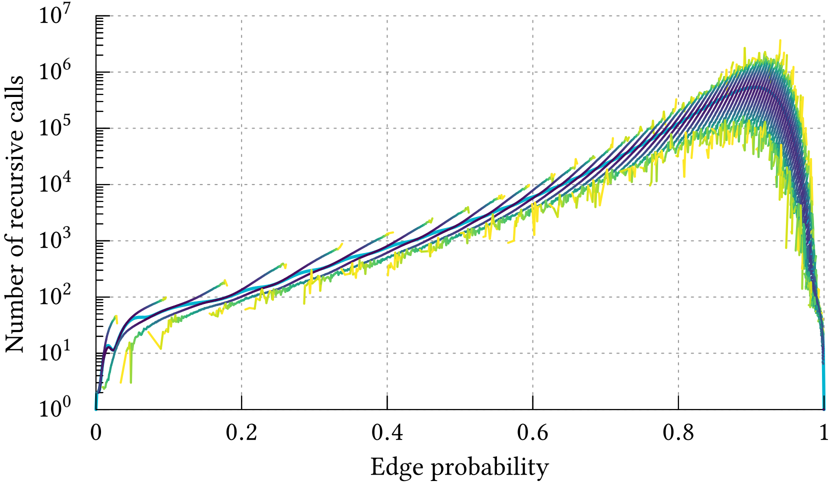
Difficulty by Solution Size



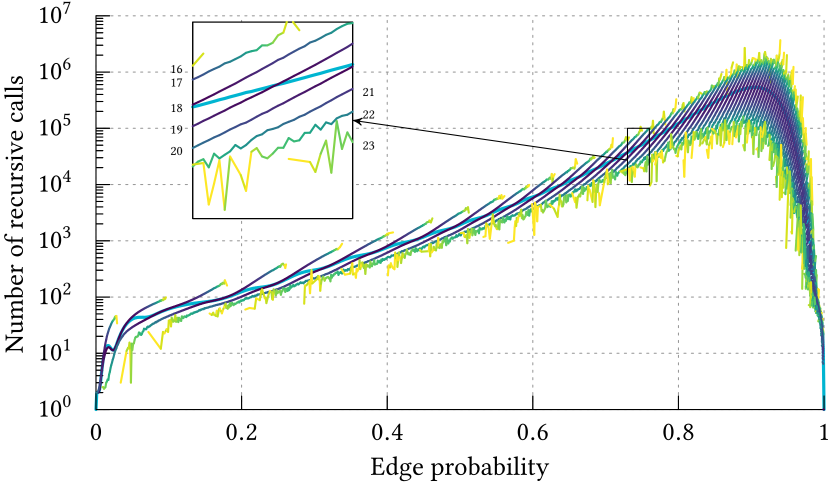
Difficulty by Solution Size



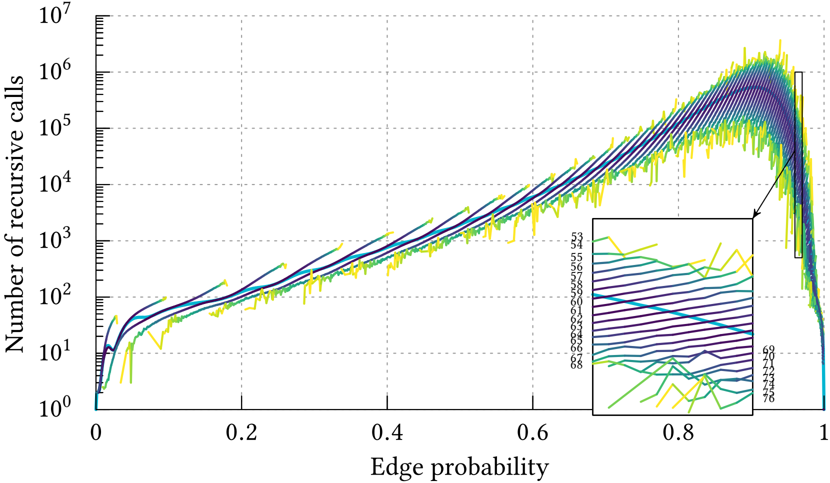
Difficulty by Solution Size



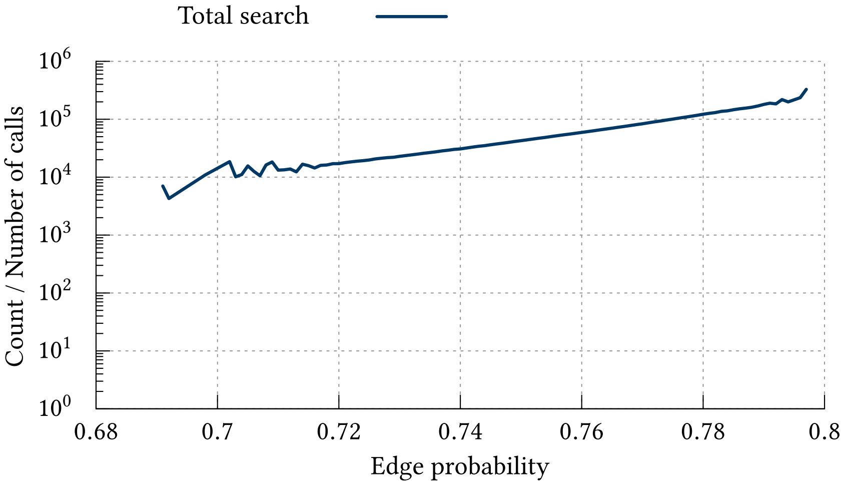
Difficulty by Solution Size



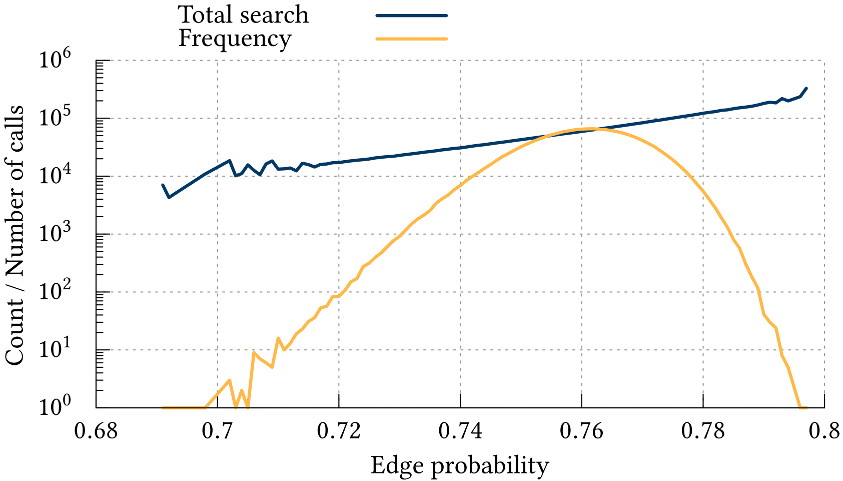
Difficulty by Solution Size



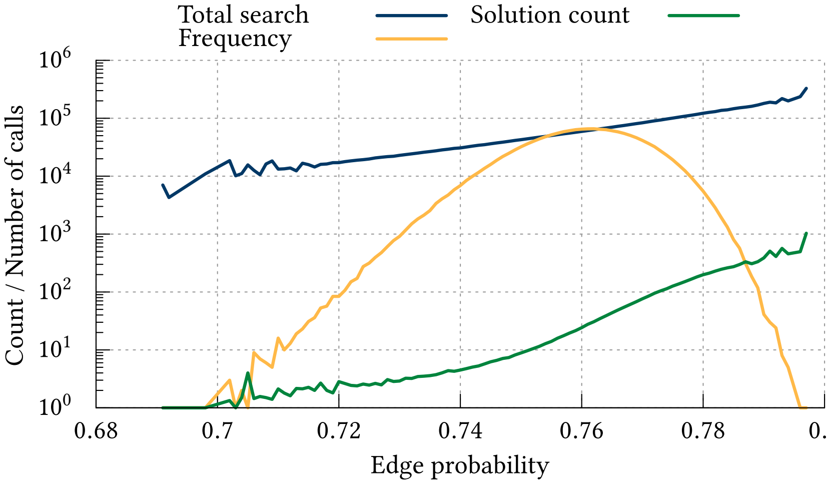
Difficulty by Optimal Solution Frequency



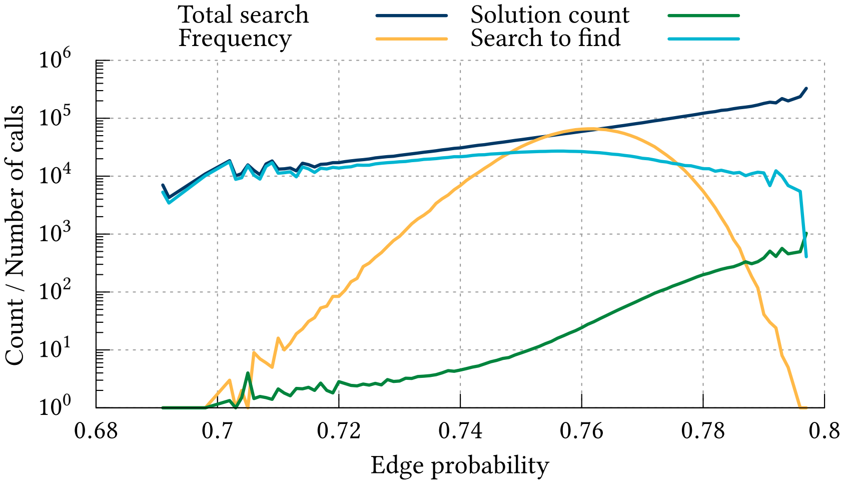
Difficulty by Optimal Solution Frequency



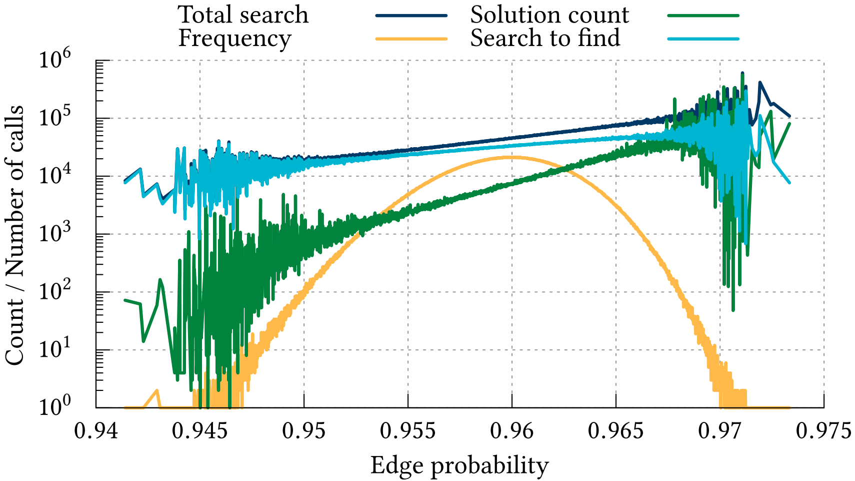
Difficulty by Optimal Solution Frequency



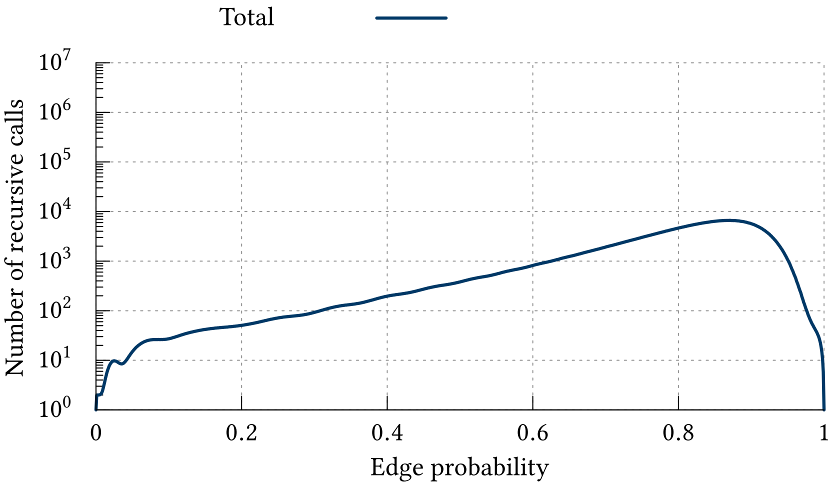
Difficulty by Optimal Solution Frequency



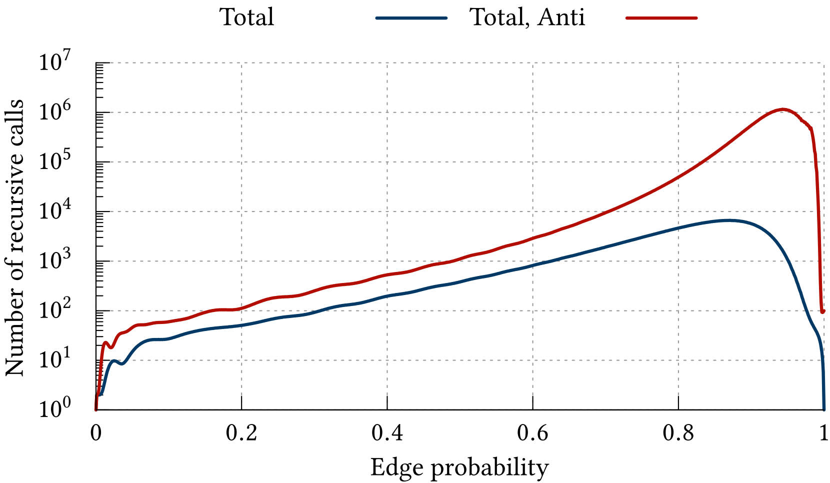
Difficulty by Optimal Solution Frequency



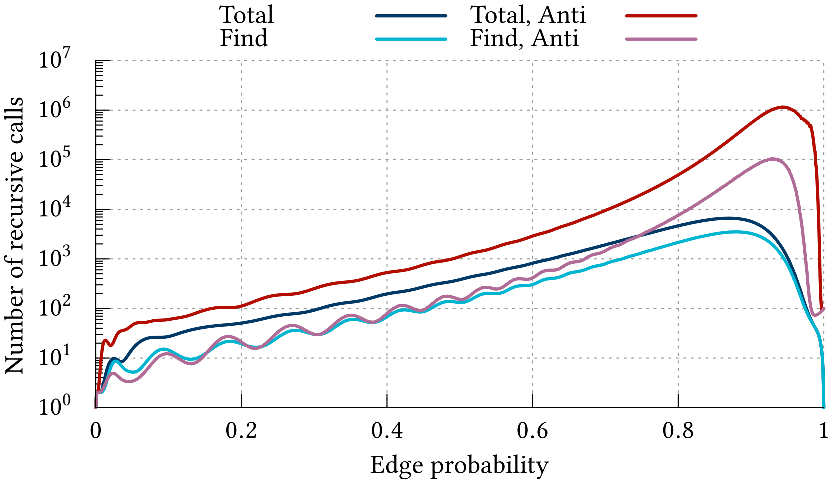
Search Order



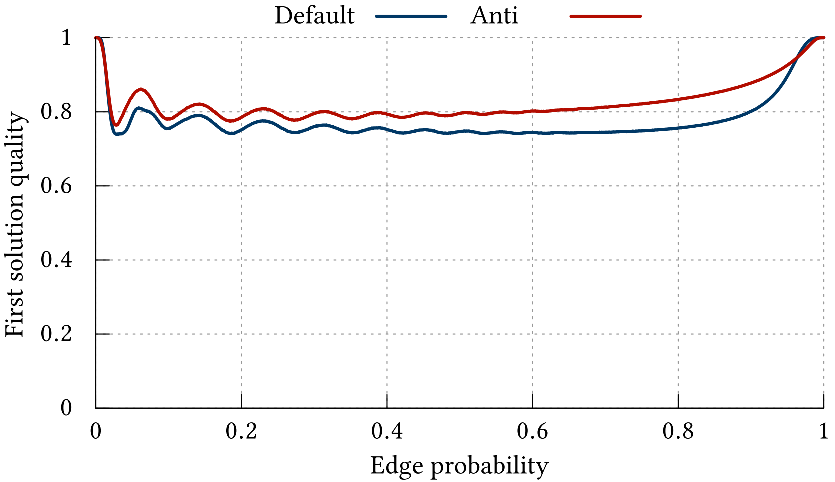
Search Order



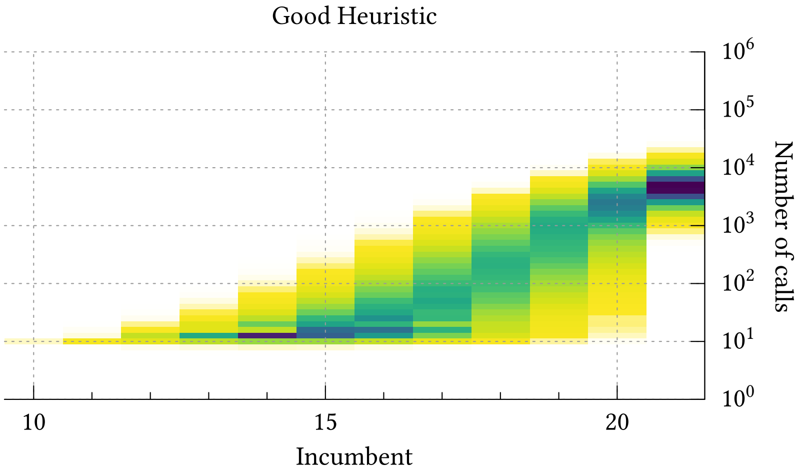
Search Order



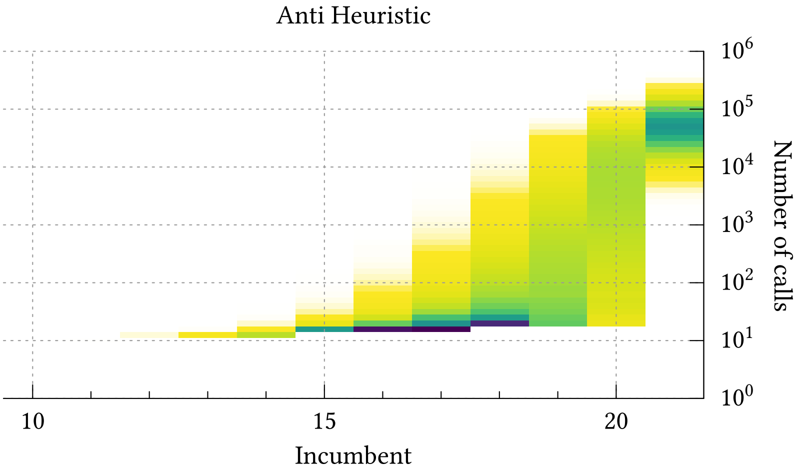
First Solution Quality



Solution Quality over Time



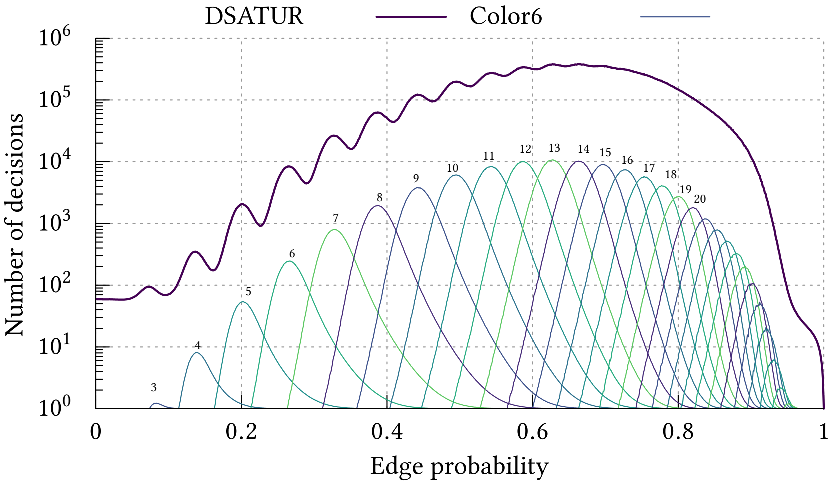
Solution Quality over Time



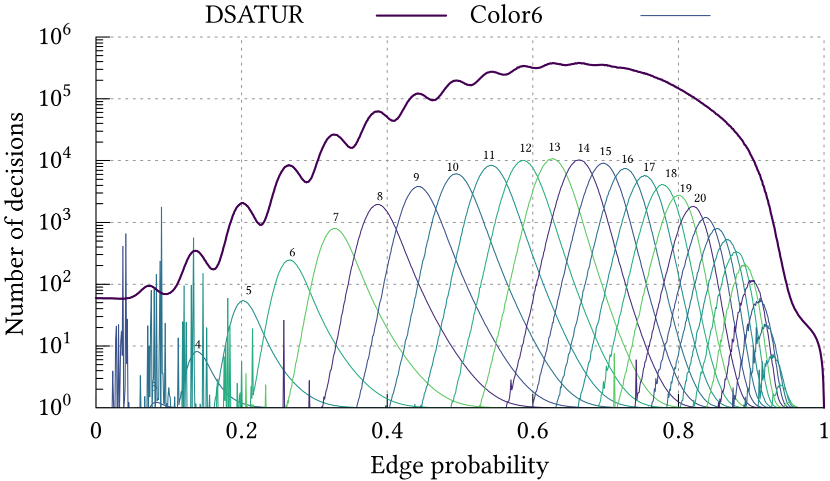
What about Other Approaches?

- All of this behaviour is common to several very different algorithms.
- Although some solvers have additional weird behaviour…

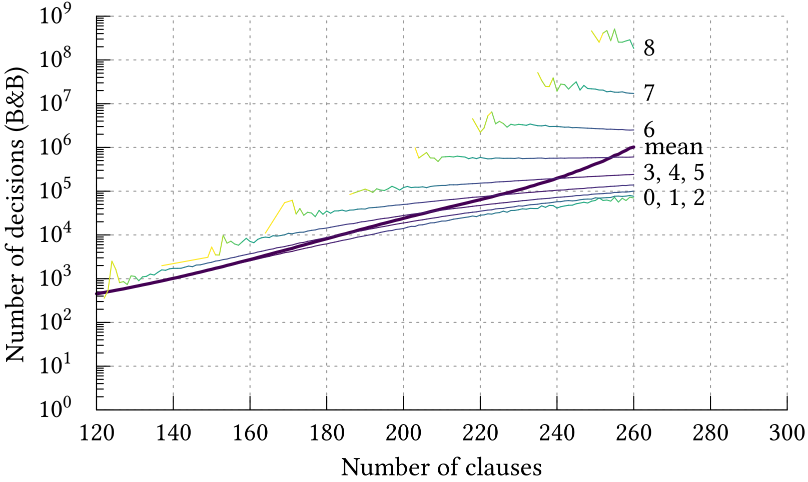
What about Other Problems?



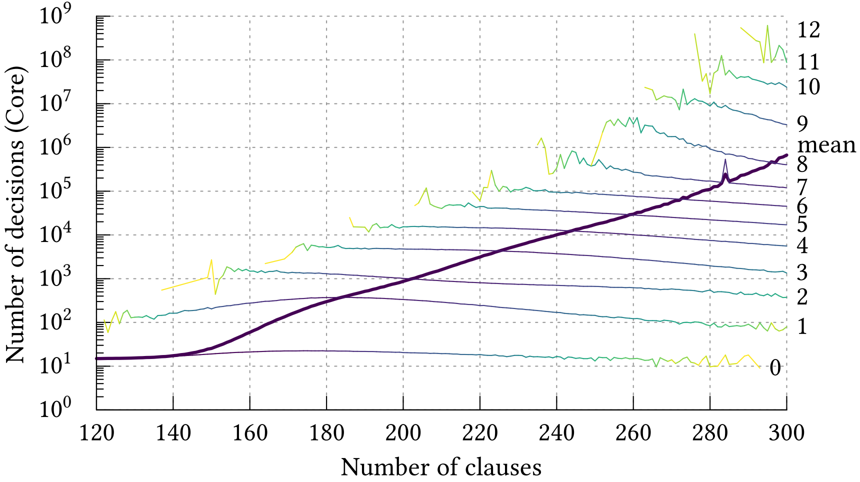
What about Other Problems?



What about Other Problems?



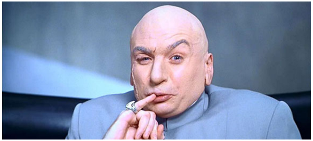
What about Other Problems?



So What?

- There is a tradeoff between proof behaviour and anytime behaviour.
- We probably want a different approach to search and heuristics to deal with this.
- We need more care in how we report results.
- Much larger sample sizes are needed for experiments on randomly generated instances.

One Million Core Hours



- A million core hours is relatively accessible and worth using.
- Outliers are interesting!
- There are an awful lot of buggy solvers out there.
  - Typical error rate: incorrect solutions for one in a thousand instances.
- Various common design choices make running these kinds of experiments annoying.
  - Use of file extensions to determine input file format.
  - Temporary file names based upon the model name.
  - Outputting to log files with a fixed name.
  - Timing code based upon CPU usage, not wallclock time.

http://www.dcs.gla.ac.uk/~ciaran

ciaran.mccreesh@glasgow.ac.uk