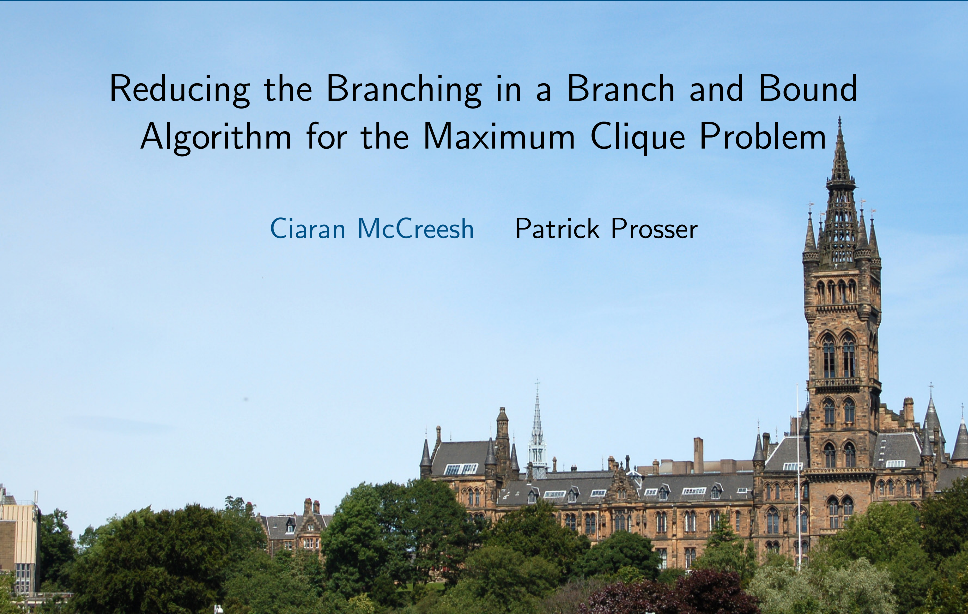
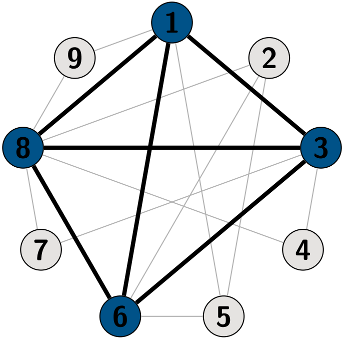
University of Glasgow

Reducing the Branching in a Branch and Bound Algorithm for the Maximum Clique Problem

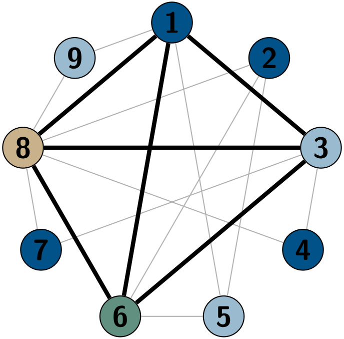
Ciaran McCreesh Patrick Prosser



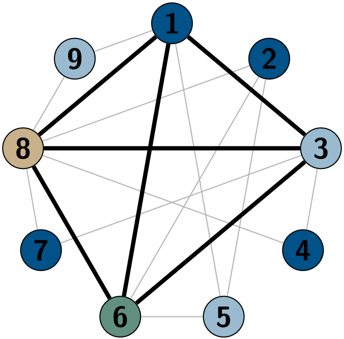
The Maximum Clique Problem



Branch and Bound



Iteration Order

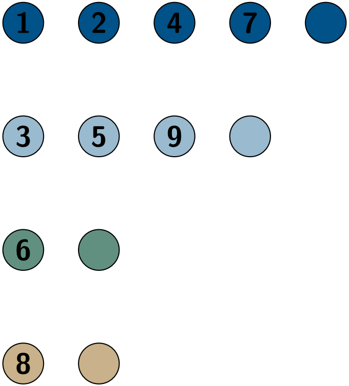


1	2	4	7	3	5	9	6	8
1	1	1	1	2	2	2	3	4

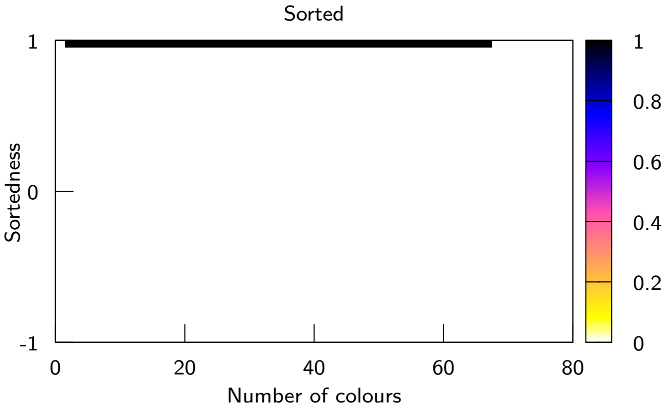
Why?

"In procedure EXPAND(R, No), after applying NUMBER-SORT more than once, a maximum clique contains a vertex p in R such that $No[p] \geq \omega(R)$. It is generally expected that a vertex p in R such that $No[p] = Max\{No[q] | q \in R\}$ has a high probability of belonging to a maximum clique." (Tomita et al., 2007)

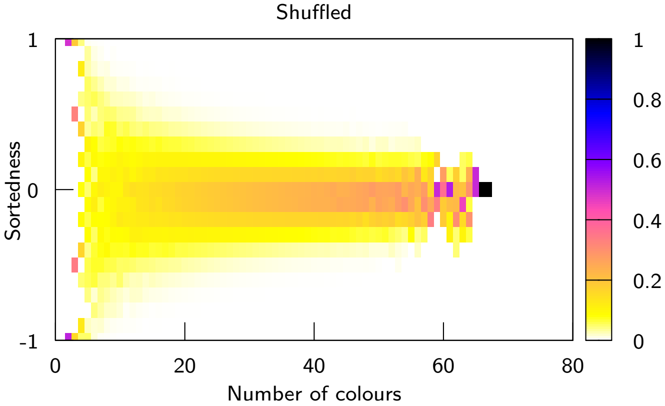
An Observation



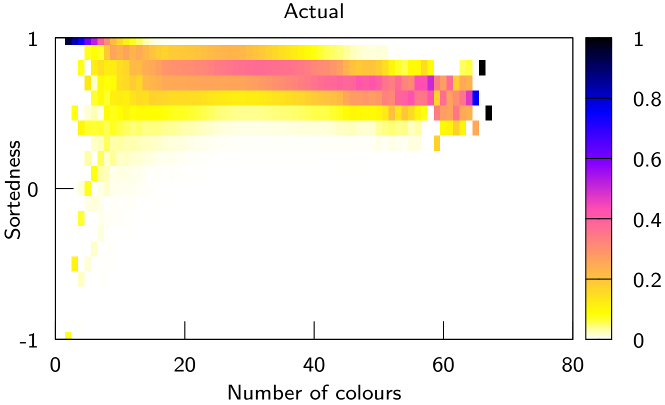
Are Later Colour Classes Smaller?



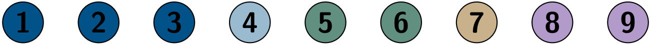
Are Later Colour Classes Smaller?



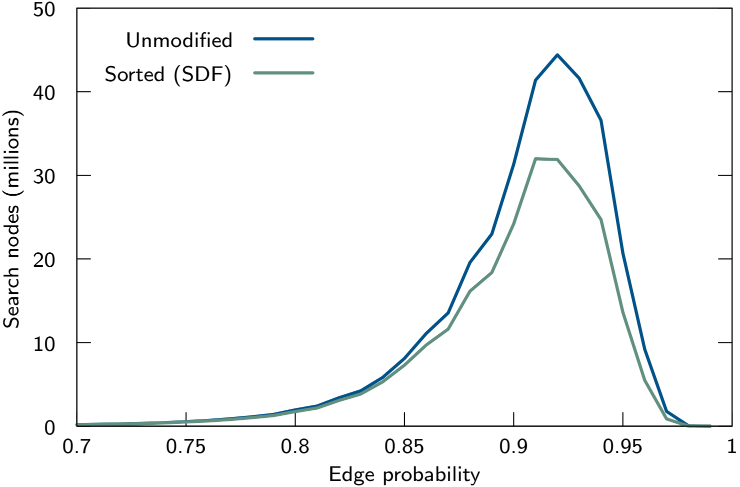
Later Colour Classes are Smaller



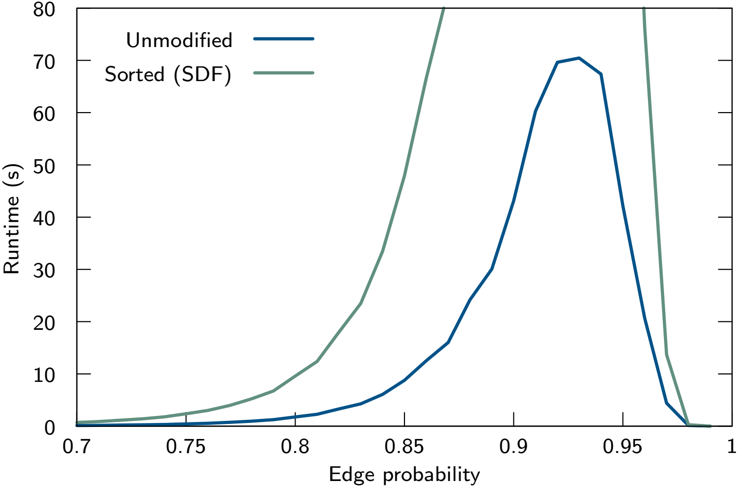
Later Singletons have More Filtering Power



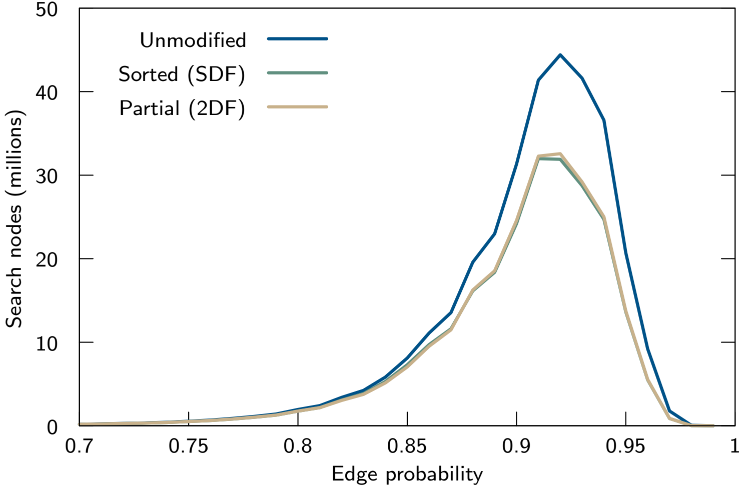
Smallest Domain First (SDF)



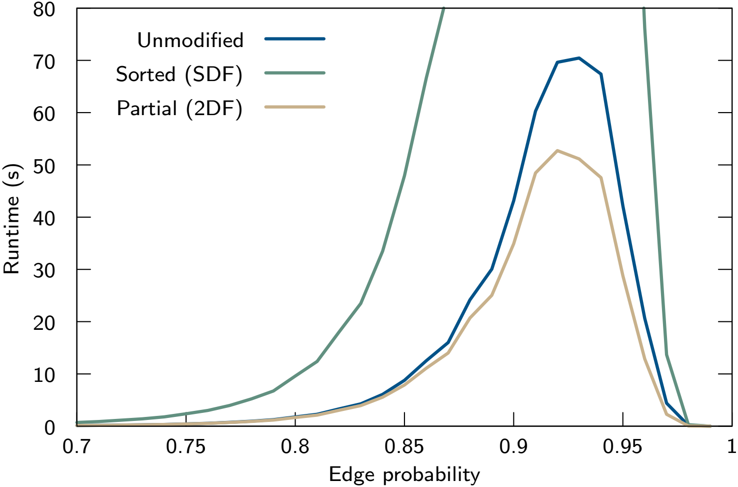
SDF is Expensive



Domains of Size 2 First (2DF)



2DF is Cheap



Unmodified
Sorted (SDF)
Partial (2DF)
Runtime (s)
0
10
20
30
40
50
60
70
80
0.7
0.75
0.8
0.85
0.9
0.95
1
Edge probability

Standard Benchmark Instances

- Modest improvements to both search space size and node counts in 39 out of 50 medium-sized DIMACS instances, minimal effect in 10 more, and a large slowdown in 1.
- Improvements of 10% to 25% for the smallest ten BHOSLIB instances.
- A full sort is typically slightly better for the search space size, but around five times as expensive.

But more importantly. . .

- By rephrasing these algorithms using language and techniques from CP, we now understand more about why they work, and what we should be preserving when making changes.

http://dcs.gla.ac.uk/~ciaran
c.mccreesh.1@research.gla.ac.uk