Some practical experiences
of
hard graph problems

Keith Briggs

Keith.Briggs@bt.com

http://keithbriggs.info

BICS Seminar - University of Bath 2006 June 12 1315

hard_graph_problems_Bath_2006_Jun_12.tex TYPESET 2006 JUNE 6 10:17 IN PDFLATEX ON A LINUX SYSTEM

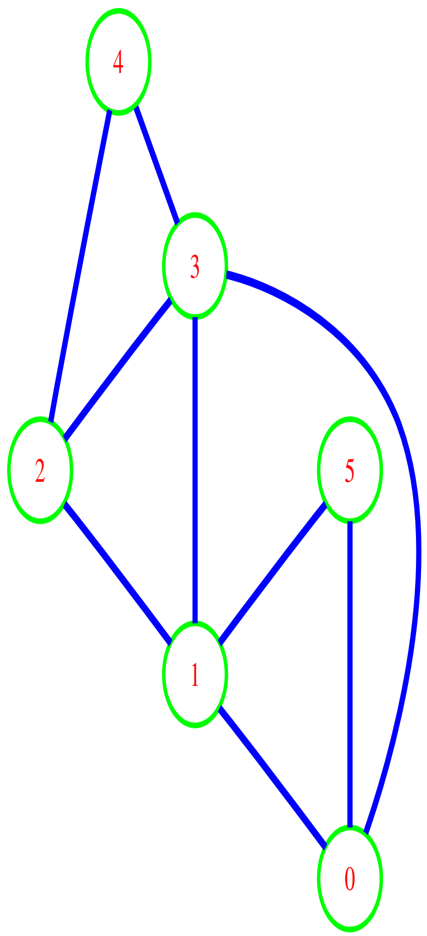
BT Research at Martlesham, Suffolk

- Cambridge-Ipswich high-tech corridor
- 2000 technologists
- 15 companies
- UCL, Univ of Essex

Talk outline

- graph concepts and problems
- chromatic number and clique number
- relaxations and optimization formulations
- performance in practice
- random k-sat
- Hamiltonian paths
- theme - Never mind the theory - how do things work in practice?

Graphs concepts



- *clique* - a complete subgraph
- *maximal clique* - a clique that cannot be extended to a larger one
- *lonely set* - a pairwise disjoint set of nodes (stable set, independent set)
- *colouring* - an assignment of colours to nodes in which no neighbours have the same colour
- *chromatic number* χ - the number of colours in a colouring with a minimal number of colours
- *loneliness* α - the number of nodes in a largest lonely set
- *clique number* ω - the number of nodes in a largest maximal clique

Hard graph problems

- finding χ, α and ω is proven to be NP-complete
  - *this means that it unlikely that any algorithm exists which runs in time which is a polynomial function of the number of nodes*
- we therefore have two options:
  - *use a heuristic, which is probably fast but may give the wrong answer*
  - *use an exact algorithm, and try to make it as fast as possible by clever coding*
- the theory is well developed and presented in many places, but little practical experience gets reported
- therefore, I tried exact algorithms for these problems to determine how big the problems can be in practice, and compared the timings with approximate (relaxed) algorithms

Chromatic number χ

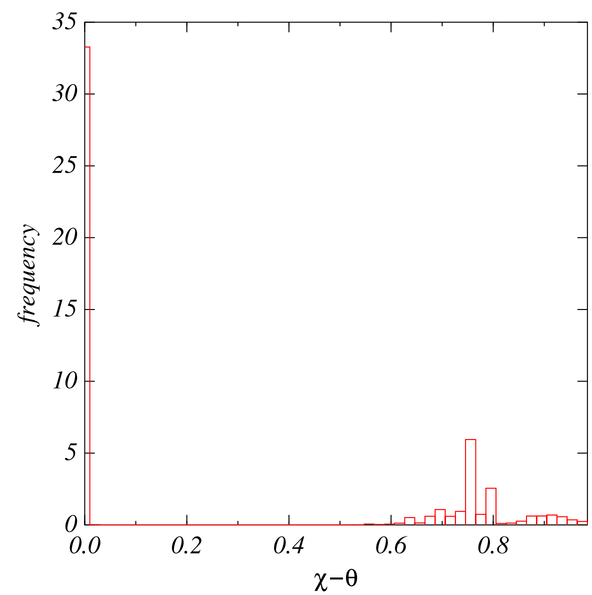
- many papers appeared in the 1980s about backtracking (branch-and-bound) methods. Some had errors
  - *idea: start to compute all colourings, but abort one as soon as it is worse than the best so far*
- can be combined with heuristics (greedy colourings) and exact bounds like $\omega \leqslant \chi \leqslant \Delta+1$, where Δ is the maximum degree
- tradeoff in using heuristics depends on type of graph
- in practice (with a well-written C program), up to 100 nodes is ok, and up to 200 for very sparse or very dense graphs
- best results are in a PhD by Chiarandini (Darmstadt 2005) `http://www.imada.sdu.dk/~marco/public.php`
- determining χ may be easy for many real-world graphs with specific structures (Coudert, DAC97)

Loneliness number α and clique number ω

- best algorithm I found was one by Tsukiyama, Ide, Ariyoshi, & Shirakawa (SIAM J. Computing **6** 505-517 (1977))
- can use graph complementation to flip these two calculations
- in practice (with a well-written C program), up to 100 nodes is ok, and up to 200 for very sparse or very dense graphs

Relaxations and semidefinite programming

- idea: formulate as an integer linear program (still hard), then relax constraints to obtain a semidefinite program (SDP, easy to solve)
- SDP: provides the *Lovász θ number* for a graph. This number is an upper bound for the clique number of a graph, and a lower bound for the chromatic number
- best SDP code: DSDP5.8 by Benson
  `http://www-unix.mcs.anl.gov/DSDP/`
- distribution of $\chi-\theta$ on $G\{n,p\}$:



LP formulation of chromatic number and clique number

- let B be the 0-1 matrix with n rows and whose columns indicate the lonely sets (in practice, ok to use only maximal lonely sets). Finding B is slow
- chromatic number χ is the solution of the 0-1 ILP

$$\begin{aligned} \text{minimize} \quad & 1^T x \\ \text{subject to} \quad & Bx \geqslant 1 \end{aligned}$$

- clique number ω is the solution of the 0-1 ILP

$$\begin{aligned} \text{maximize} \quad & y^T 1 \\ \text{subject to} \quad & y^T B \leqslant 1 \end{aligned}$$

- solving the 0-1 ILPs is hard, so we don't try

Fractional chromatic number χ_f

- used by McDiarmid for a radio channel assignment problem in which the *demand* (required number of channels) at each node varies

- χ_f is the solution of the LP (ordinary LP, so easy)

$$\begin{aligned} \text{minimize} \quad & 1^T x \\ \text{subject to} \quad & Bx \geqslant 1 \\ & x \geqslant 0 \end{aligned}$$

- ω_f is the solution of the LP (ordinary LP, so easy)

$$\begin{aligned} \text{maximize} \quad & y^T 1 \\ \text{subject to} \quad & y^T B \leqslant 1 \\ & y \geqslant 0 \end{aligned}$$

Typical results

- ★ I have programmed all the methods

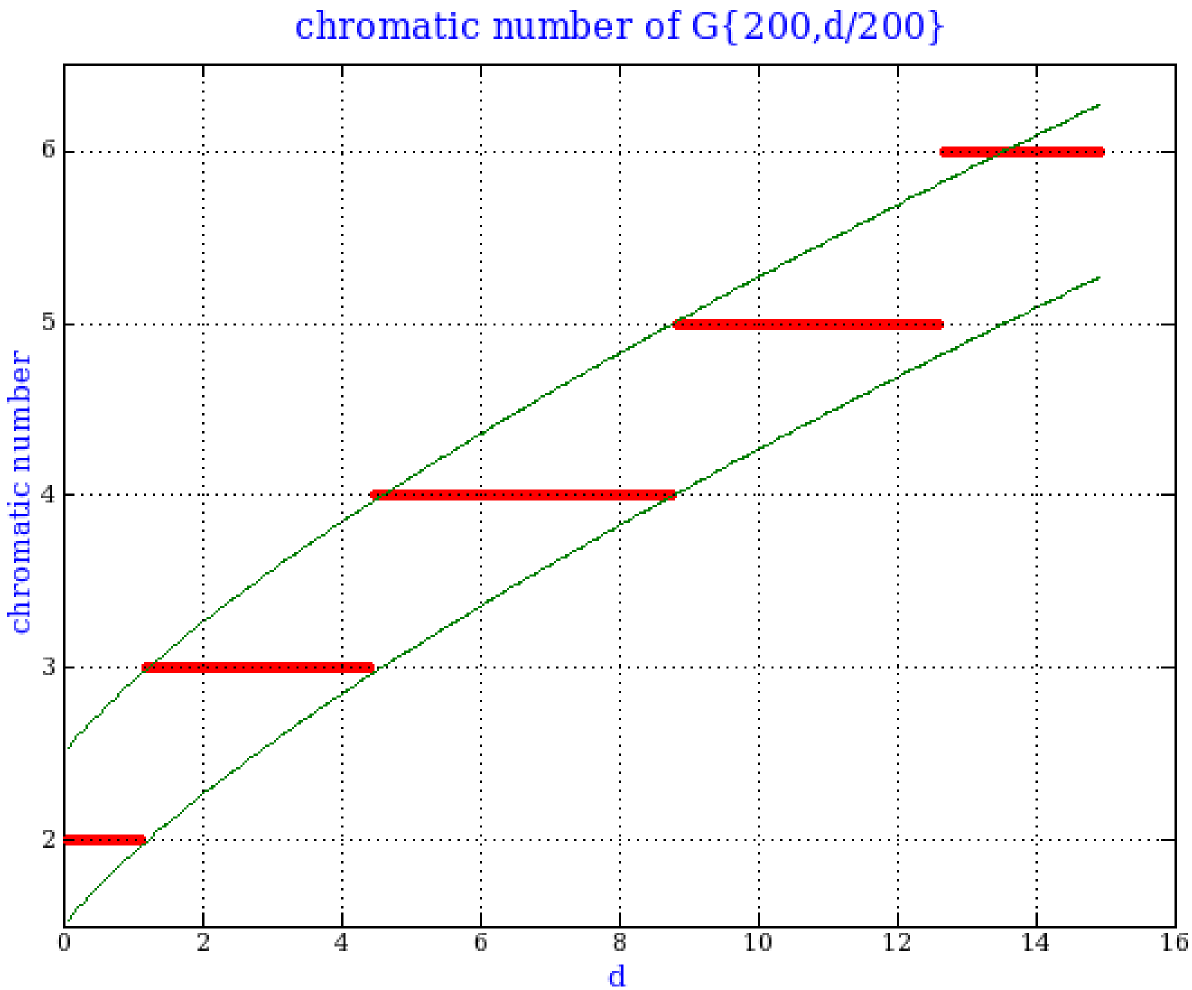
graph	n	p or m	α	ω	χ_f	χ	θ	
g1	10	0.5	4	4	4	4	4	medium
g2	10	0.9	3	7	7	7	7	dense
g3	10	0.1	9	2	2	2	2	sparse
g4	50	5	45	2	2	2	2	sparse, big
g5	50	100	23	3	3	3	3	medium density, big
g6	50	1000	4	14	16.5	17	15.36	high density, big

- ★ theorem: we always have $\omega \leqslant \omega_\mathsf{f} \leqslant \chi_\mathsf{f} \leqslant \chi$
- ★ recall $\omega \leqslant \theta \leqslant \chi$

Achlioptas & Naor

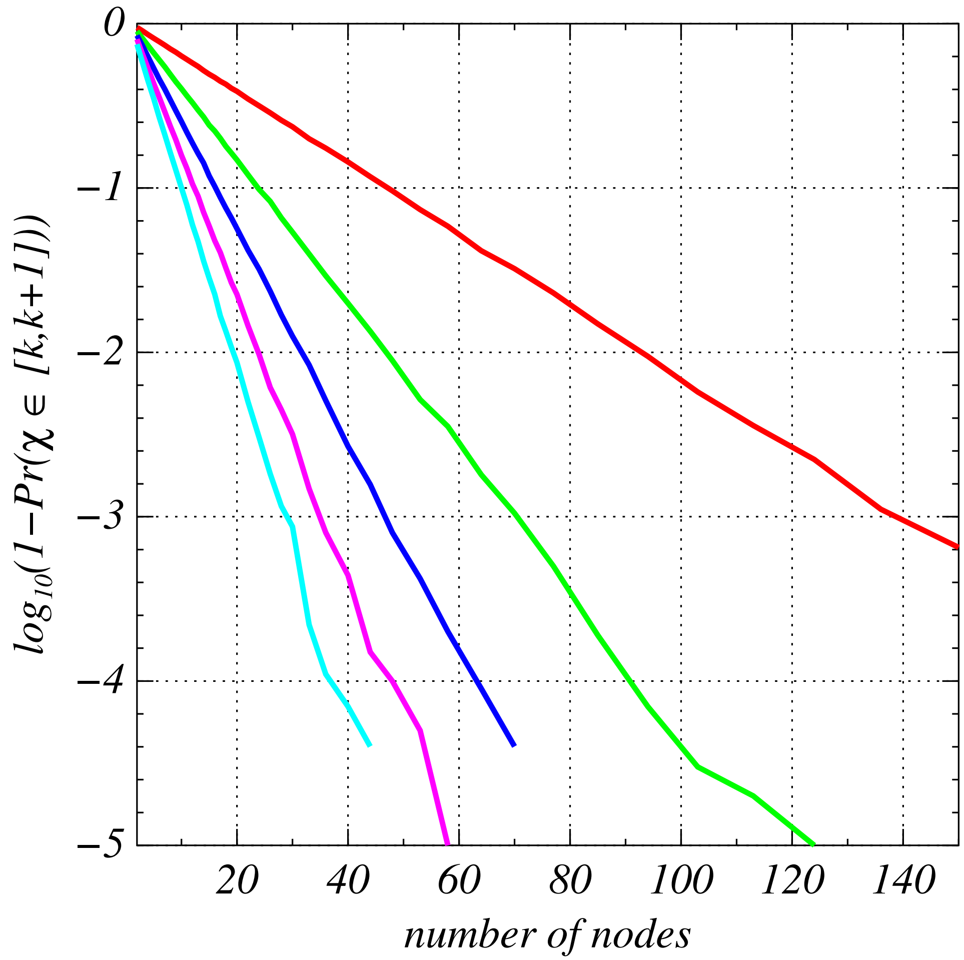
- *The two possible values of the chromatic number of a random graph* Annals of Mathematics, **162** (2005) `http://www.cs.ucsc.edu/~optas/`
- the authors show that for fixed d, as $n \to \infty$, the chromatic number of $G\{n, d/n\}$ is either k or $k+1$, where k is the smallest integer such that $d < 2k\log(k)$. In fact, this means that k is given by $\lceil d/(2W(d/2)) \rceil$
- $G\{n, p\}$ means the random graph on n nodes and each possible edge appears independently with probability p

Achlioptas & Naor cotd.



Achlioptas & Naor - my conjecture

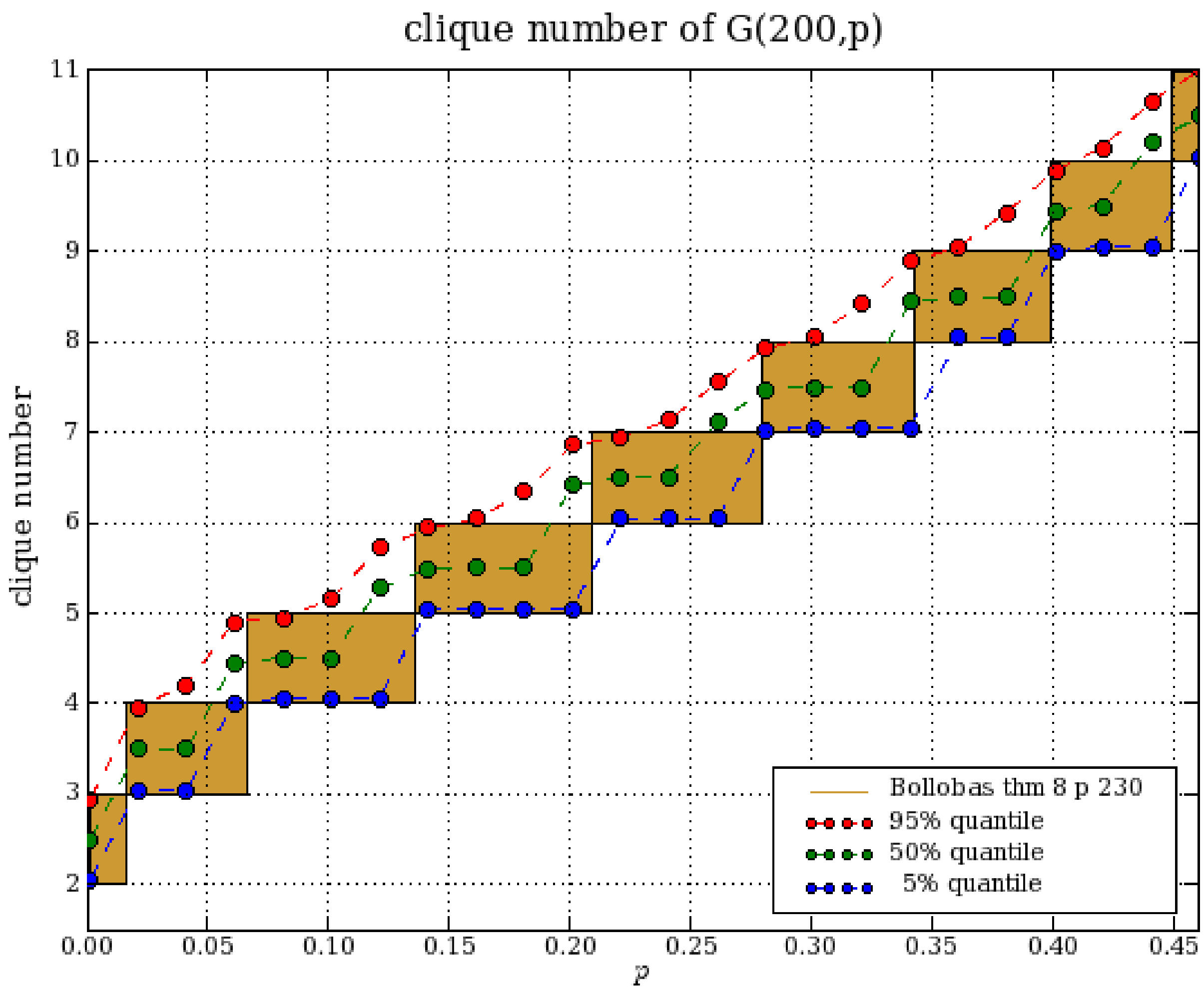
★ the next graph (each point is the average of 1 million trials) suggests that for small d, we have Pr $[\chi \in [k, k+1]] \sim 1 - \exp(-dn/2)$



Cliques

- In *Modern graph theory*, page 230, Bollobás shows that the clique number of $G(n,p)$ as $n \to \infty$ is almost surely d or $d+1$, where d is the greatest natural number such that $\binom{n}{d} p^{\binom{d}{2}} \geqslant \log(n)$
- How accurate is this formula when n is small?
- We have $d = 2\log(n)/\log(1/p) + \mathcal{O}(\log\log(n))$.

Cliques - simulation results



Counting graphs

Number of graphs on n nodes with chromatic number k:

k \ n =	1	2	3	4	5	6	7	8	9	10	
2	0	1	2	6	12	34	87	302	1118	5478	A076278
3	0	0	1	3	16	84	579	5721	87381	2104349	A076279
4	0	0	0	1	4	31	318	5366	155291	7855628	A076280
5	0	0	0	0	1	5	52	867	28722	1919895	A076281
6	0	0	0	0	0	1	6	81	2028	115391	A076282
7	0	0	0	0	0	0	1	7	118	4251	
8	0	0	0	0	0	0	0	1	8	165	
9	0	0	0	0	0	0	0	0	1	9	
10	0	0	0	0	0	0	0	0	0	1	
11	0	0	0	0	0	0	0	0	0	0	

(A-numbers from `http://www.research.att.com/~njas/sequences/`)

Counting graphs cotd.

Number of graphs on n nodes with clique number k:

k \ n =	1	2	3	4	5	6	7	8	9	10	
2	0	1	2	6	13	37	106	409	1896	12171	A052450
3	0	0	1	3	15	82	578	6021	101267	2882460	A052451
4	0	0	0	1	4	30	301	4985	142276	7269487	A052452
5	0	0	0	0	1	5	51	842	27107	1724440	A077392
6	0	0	0	0	0	1	6	80	1995	112225	A077393
7	0	0	0	0	0	0	1	7	117	4210	A077394
8	0	0	0	0	0	0	0	1	8	164	
9	0	0	0	0	0	0	0	0	1	9	
10	0	0	0	0	0	0	0	0	0	1	

(A-numbers from `http://www.research.att.com/~njas/sequences/`)

Another NP-complete problem: k-sat

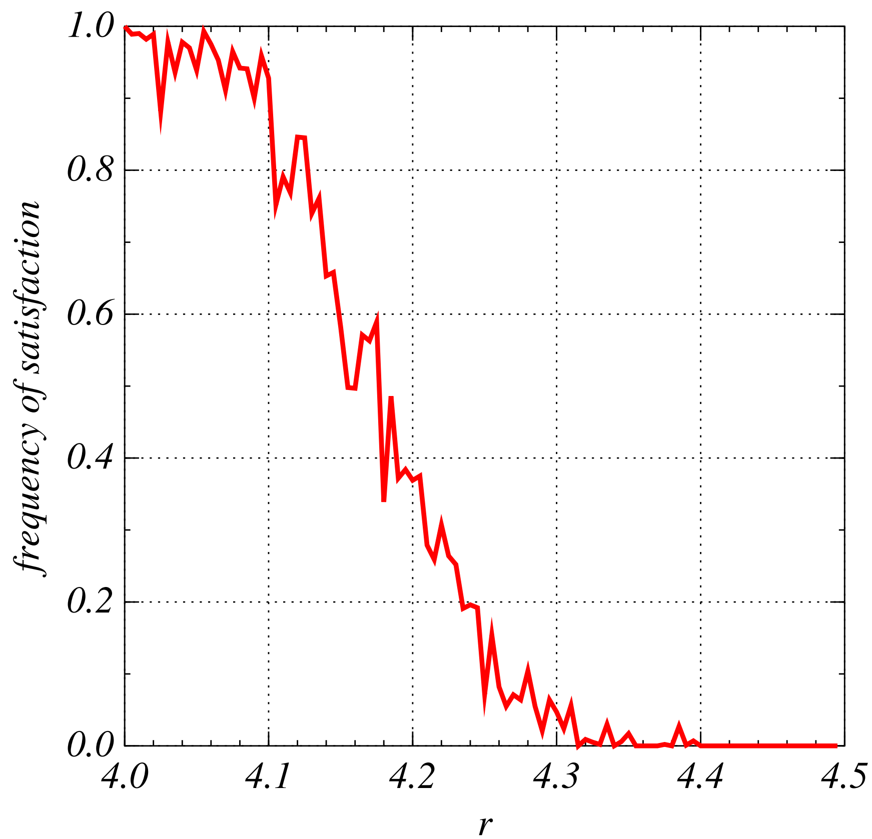
- n Boolean variables
- Boolean function f in CNF: consists of the "and" ($\wedge$) of a number of clauses
- each clause is the "or" ($\vee$) of k variables or their negations
- e.g. $f(x) = (x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \overline{x_3}) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee x_2 \vee x_3)$
- k-sat: find an assignment x of the variables such that $f(x) = 1$
- useful for finding feasible points in scheduling problems etc.
- hard to find a solution, easy to verify a proposed solution
- 2-sat is easy, k-sat for $k > 3$ can be reduced to 3-sat
- therefore, we can use heuristics to find solutions

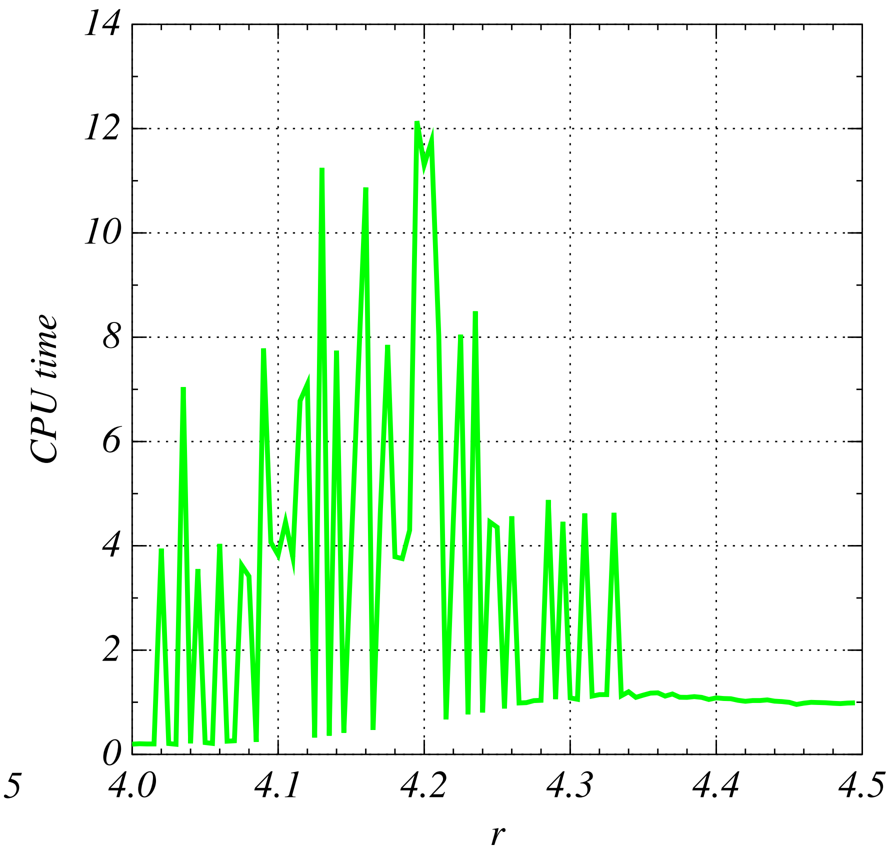
Random k-sat

- choose clauses randomly, with r being the ratio of the number of clauses to the number of variables
- recent (2002) big breakthrough: *survey propagation* by Mézard, Parisi & Zecchina (`http://www.sciencemag.org/cgi/content/abstract/297/5582/812` and many other articles in Nature and Science): physics-inspired heuristic that works (at least for random 3-sat) even for $n > 10^5$
  - *my experience: works worse than other heuristics on small, structured problems*
- there is a phase transition near $r_c = 4.26$, where random 3-sat jumps from being almost surely satisfiable to almost surely unsatisfiable

Random 3-sat phase transition

- I computed this with the survey propagation heuristic - 1000 variables x_i, 100 trials for each value of r:





Hamiltonian path problem

- find a path in a graph that visits every node once and only once
- NP-complete
- can encode as a k-sat problem (Knuth *Boolean Basics* problem 40 - errors!) and use heuristic
- let p_{uv} mean $u < v$ in a ordering of the nodes, and let q_{uvw} mean $u < v < w$
- they express the constraints that consecutive nodes are adjacent (i.e. non-adjacent nodes ($\nsim$) are non-consecutive)
- the graph has a Hamiltonian path iff the set of clauses on the next page is satisfiable
- recall that $(\overline{x} \vee y) \equiv (x \Rightarrow y)$
- can we do Hamiltonian circuits this way?

Hamiltonian path encoding

- ★ $p_{uv} \vee p_{vu}$ for all pairs $u \neq v$ (i.e. either $u < v$ or $v < u$)
- ★ $\overline{p}_{uv} \vee \overline{p}_{vu}$ for all pairs $u \neq v$ (i.e. not both $u < v$ and $v < u$)
- ★ $\overline{p}_{uv} \vee \overline{p}_{vw} \vee p_{uw}$ for all pairs $u \neq v, u \neq w, v \neq w$ (i.e. $u < v$ and $v < w \Rightarrow u < w$)
- ★ $\overline{q}_{uvw} \vee p_{uv}$ for $u \nsim w$ (i.e. $u < v < w \Rightarrow u < v$)
- ★ $\overline{q}_{uvw} \vee p_{vw}$ for $u \nsim w$ (i.e. $u < v < w \Rightarrow v < w$)
- ★ $q_{uvw} \vee \overline{p}_{uv} \vee \overline{p}_{vw}$ (i.e. $\overline{u < v < w} \Rightarrow u > v$ or $v > w$)
- ★ $q_{uvw} \vee q_{wvu}$ for $u \nsim w$ and $\forall v \notin \{u, w\}$
- ★ I wrote a python program to translate any given graph into these clauses and solve them with a satisfier
- ★ `http://www.research.att.com/~njas/sequences/A115065`

The future

- work on making the practical algorithms faster, by using more efficient data structures etc.
- understand the accuracy of the θ bound
- what is the distribution of $\theta - \chi$ for standard ensembles of random graphs?
- study properties of special graphs arising in real applications - scale-free, unit-disk etc.
- understand the k-sat phase transition
- encode more real problems as k-sat
- see the book *Perfect graphs* (ed. J L Ramírez Alfonsín & B A Reed) for lots more on applications to networks