The Interface between P and NP

Toby Walsh

Cork Constraint Computation Centre

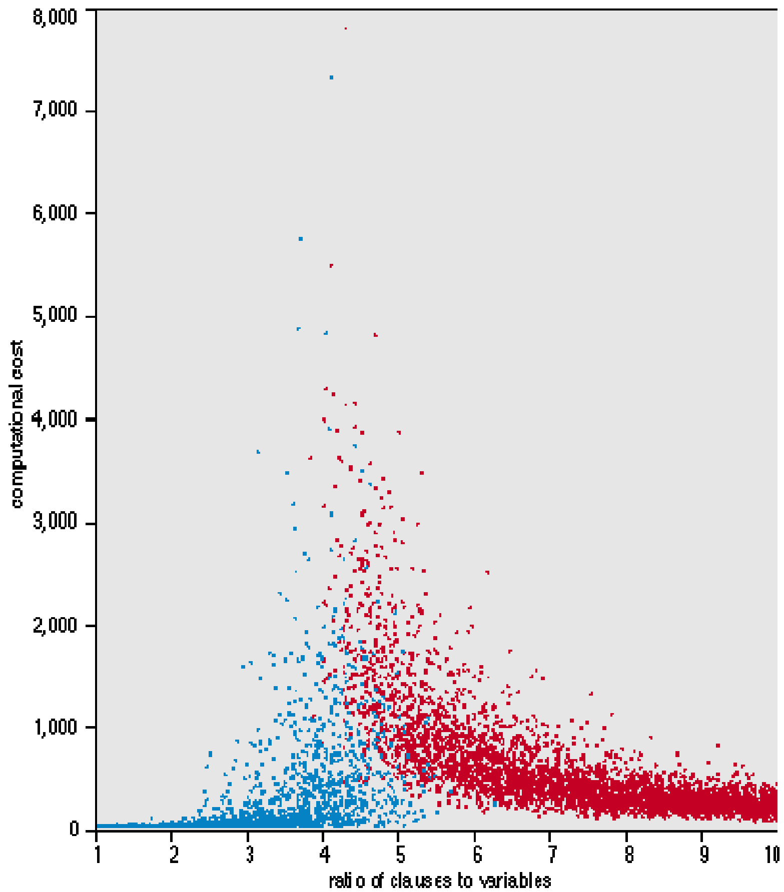
http://4c.ucc.ie/~tw

1/25/2008

3-SAT

- **Where are the hard 3-SAT problems?**
- **Sample randomly generated 3-SAT**
  - Fix number of clauses, l
  - Number of variables, n
  - By definition, each clause has 3 variables
  - Generate all possible clauses with uniform probability

Random 3-SAT



- Which are the hard instances?
  - **around *l/n* = 4.3**

What happens with larger problems?

Why are some dots red and others blue?

This is a so-called “phase transition”

Where did this all start?

- At least as far back as 60s with Erdos & Renyi
  - **thresholds in random graphs**
- Late 80s
  - **pioneering work by Karp, Purdom, Kirkpatrick, Huberman, Hogg …**
- Flood gates burst
  - **Cheeseman, Kanefsky & Taylor's IJCAI-91 paper**

What do we know about this phase transition?

- It's shape
  - **Step function in limit [Friedgut 98]**
- It's location
  - **Theory puts it in interval:**

    $3.42 < l/n < 4.506$
  - **Experiment puts it at:**

    $l/n = 4.2$

3SAT phase transition

- Lower bounds (hard)
  - **Analyse algorithm that almost always solves problem**
  - **Backtracking hard to reason about so typically without backtracking**
    - Complex branching heuristics needed to ensure success
    - But these are complex to reason about

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    No assumptions about the distribution of X *except non-negative!*

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    The expected value of X can be easily calculated

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    E[X] = 2^n * (7/8)^l

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    $E[X] = 2^n * (7/8)^l$

    If E[X] < 1, then prob(X>=1) = prob(SAT) < 1

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    $E[X] = 2^n * (7/8)^l$

    If $E[X] < 1$, then $2^n * (7/8)^l < 1$

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    $E[X] = 2^n * (7/8)^l$

    If $E[X] < 1$, then $2^n * (7/8)^l < 1$

    $n + l \log2(7/8) < 0$

3SAT phase transition

- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **E.g. Markov (or 1st moment) method**

    For any statistic X

    prob(X>=1) <= E[X]

    Let X be the number of satisfying assignments for a 3SAT problem

    $E[X] = 2^n * (7/8)^l$

    If $E[X] < 1$, then $2^n * (7/8)^l < 1$

    $n + l \log2(7/8) < 0$

    $l/n > 1/\log2(8/7) = 5.19\ldots$

3SAT phase transition

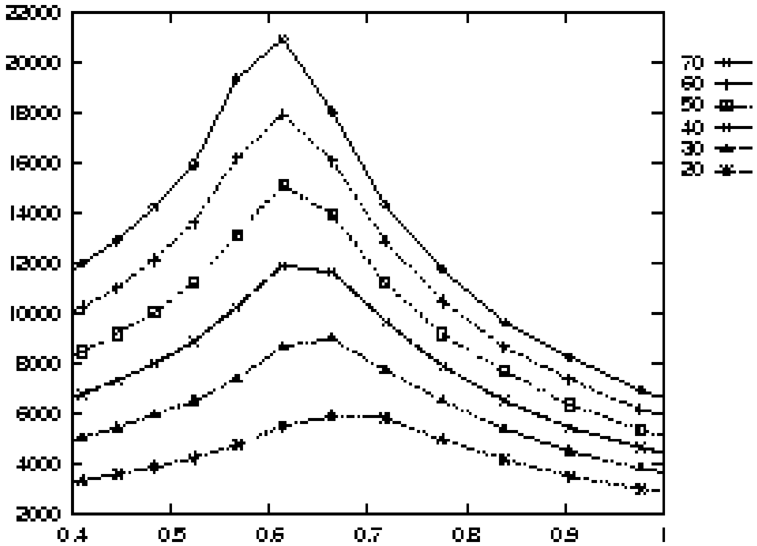
- Upper bounds (easier)
  - **Typically by estimating count of solutions**
  - **To get tighter bounds than 5.19, can refine the counting argument**
    - E.g. not count all solutions but just those maximal under some ordering

Random 2-SAT

- 2-SAT is P
  - **linear time algorithm**
- Random 2-SAT displays "classic" phase transition
  - ***l/n* < 1, almost surely SAT**
  - ***l/n* > 1, almost surely UNSAT**
  - **complexity peaks around *l/n*=1**

x1 v x2, -x2 v x3, -x1 v x3, …

Phase transitions in P

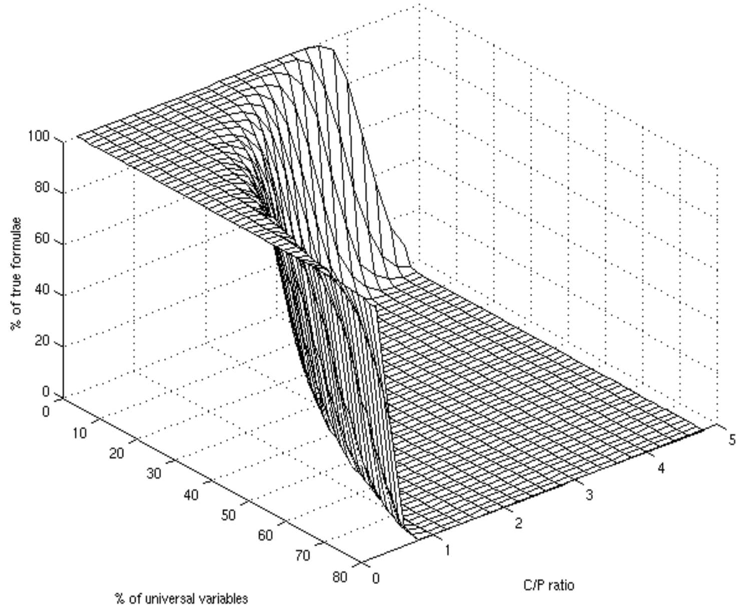


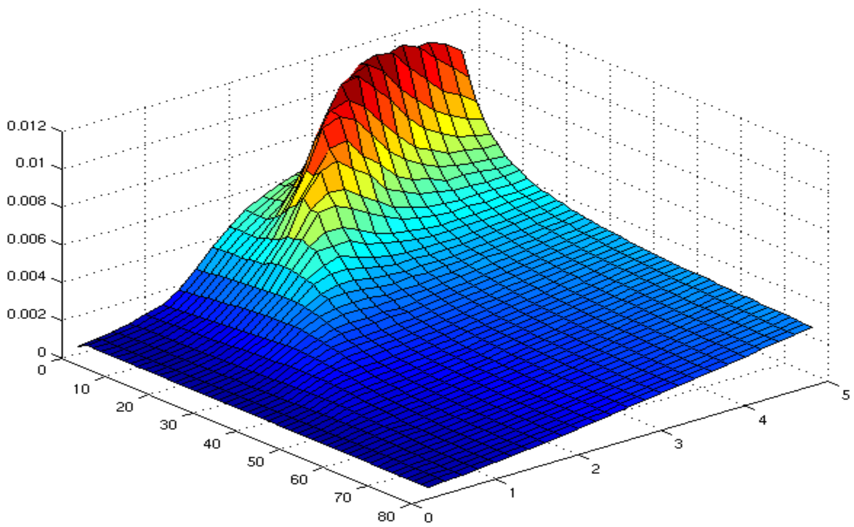
- 2-SAT
  - *l/n*=1
- Horn SAT
  - transition not “sharp”
- Arc-consistency
  - rapid transition in whether problem can be made AC
  - peak in (median) checks

Phase transitions above NP

- PSpace
  - **QSAT (SAT of QBF)**

∀x1 ∃x2 ∀x3 . x1 v x2 & -x1 v x3





Phase transitions above NP

- PSpace-complete
  - **QSAT (SAT of QBF)**
  - **stochastic SAT**
  - **modal SAT**
- PP-complete
  - **polynomial-time probabilistic Turing machines**
  - **counting problems**
  - **#SAT(>= 2^n/2)**

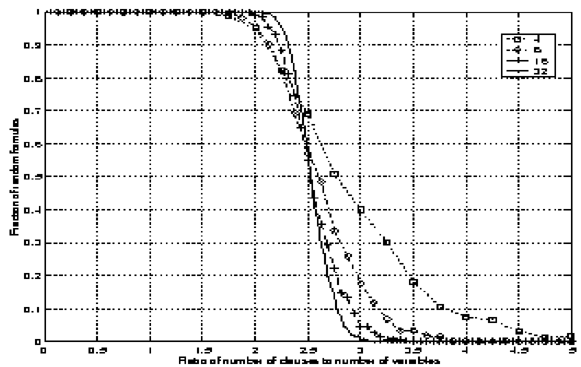


Figure 1: Phase Transition Graphs

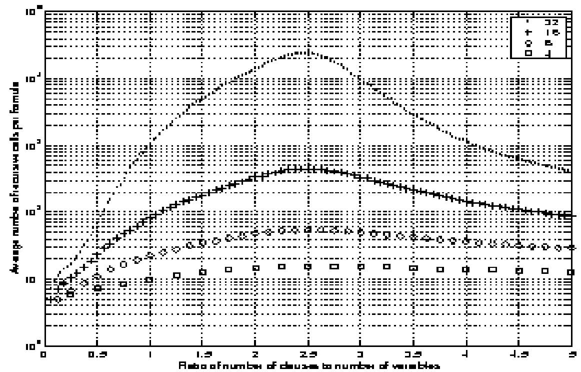


Figure 2: Performance Graphs

Exact phase boundaries in NP

- Random 3-SAT is only known within bounds
  - **3.42 < *l/n* < 4.506**
- Recent result gives an exact NP phase boundary
  - **1-in-*k* SAT at *l/n* = 2/*k*(*k*-1)**

Are there any NP phase boundaries known exactly?

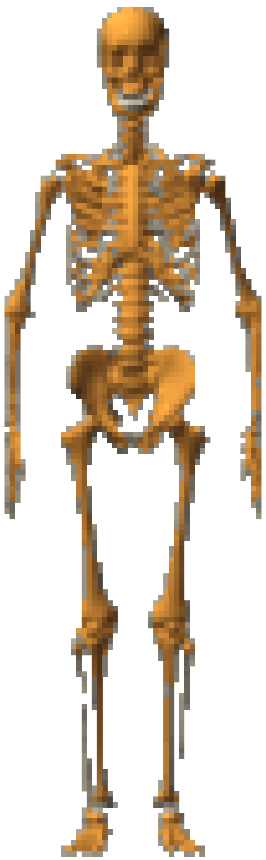
Backbone

- Variables which take fixed values in all solutions
  - **alias unit prime implicates**
- Let f_k be fraction of variables in backbone
  - **in random 3-SAT**

    $l/n < 4.3$, f_k vanishing (otherwise adding clause could make problem unsat)

    $l/n > 4.3$, $f_k > 0$

discontinuity at phase boundary!



Backbone

- Search cost correlated with backbone size
  - **if f_k non-zero, then can easily assign variable “wrong” value**
  - **such mistakes costly if at top of search tree**
- One source of “thrashing” behaviour
  - **can tackle with randomization and rapid restarts**

Can we adapt algorithms to offer more robust performance guarantees?

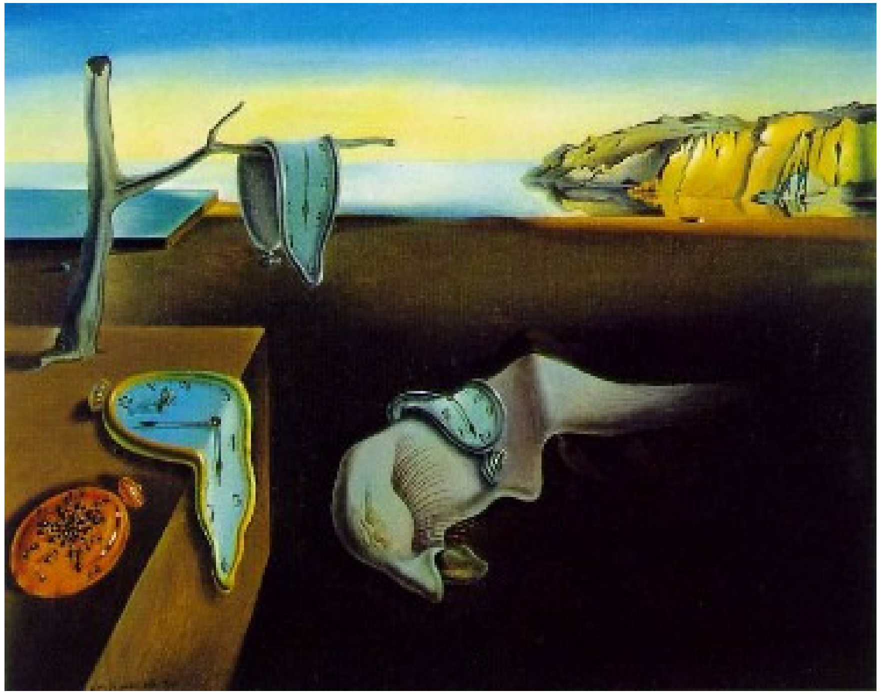
Backbone

- Backbones observed in structured problems
  - **quasigroup completion problems (QCP)**

- Backbones also observed in optimization and approximation problems
  - **coloring, TSP, blocks world planning …**

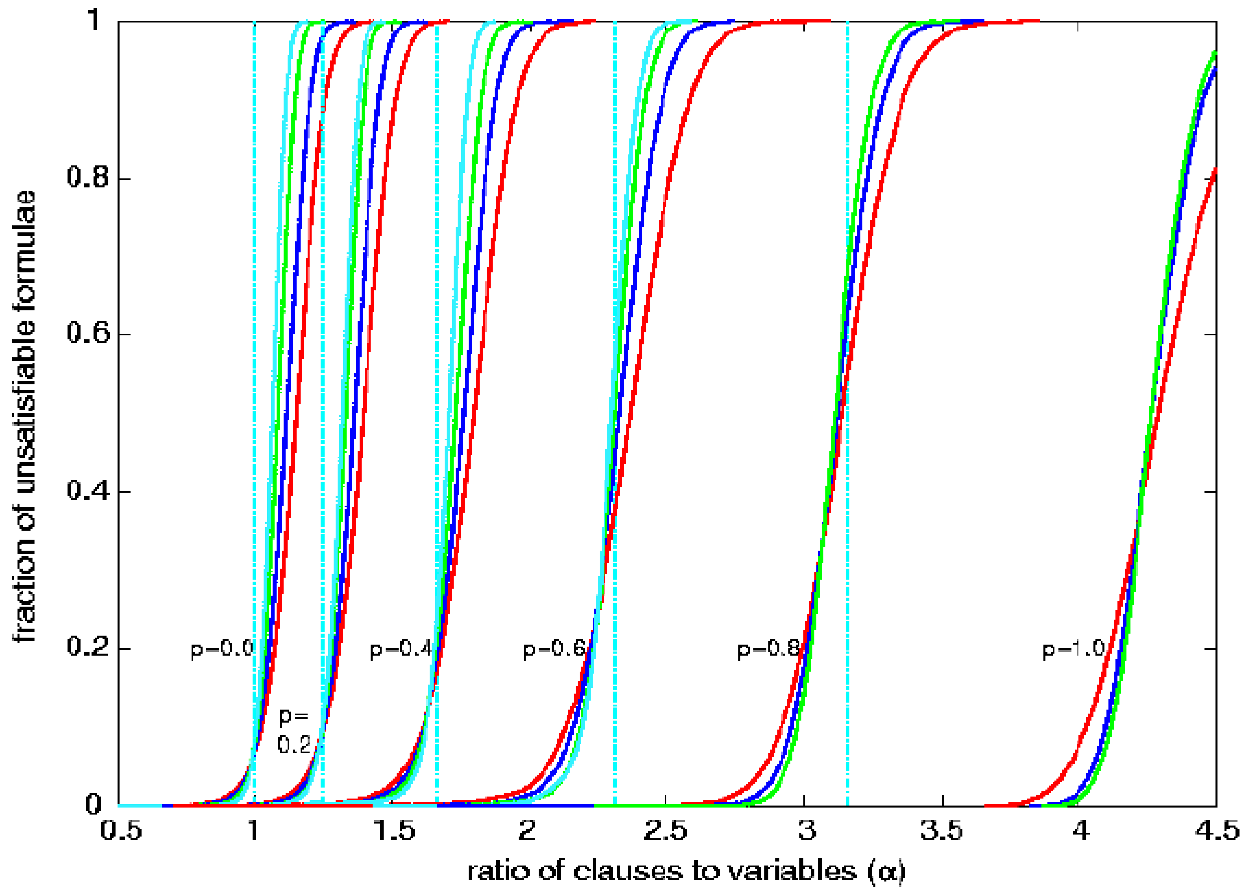
Can we adapt algorithms to identify and exploit the backbone structure of a problem?

2+p-SAT

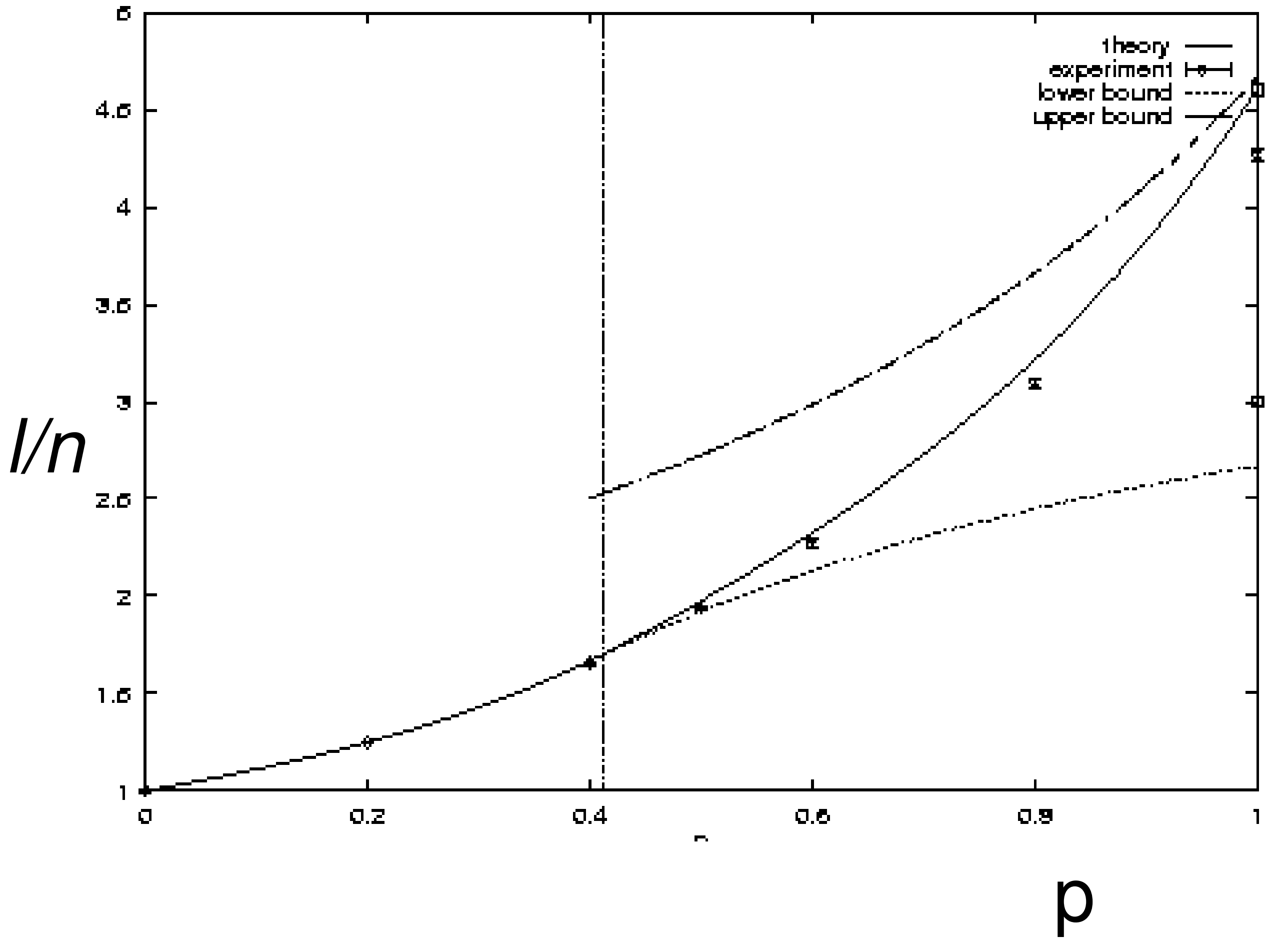


- Morph between 2-SAT and 3-SAT
  - **fraction p of 3-clauses**
  - **fraction (1-p) of 2-clauses**
- 2-SAT is polynomial (linear)
  - **phase boundary at *l/n* =1**
  - **but no backbone discontinuity here!**
- 2+p-SAT maps from P to NP
  - **p>0, 2+p-SAT is NP-complete**

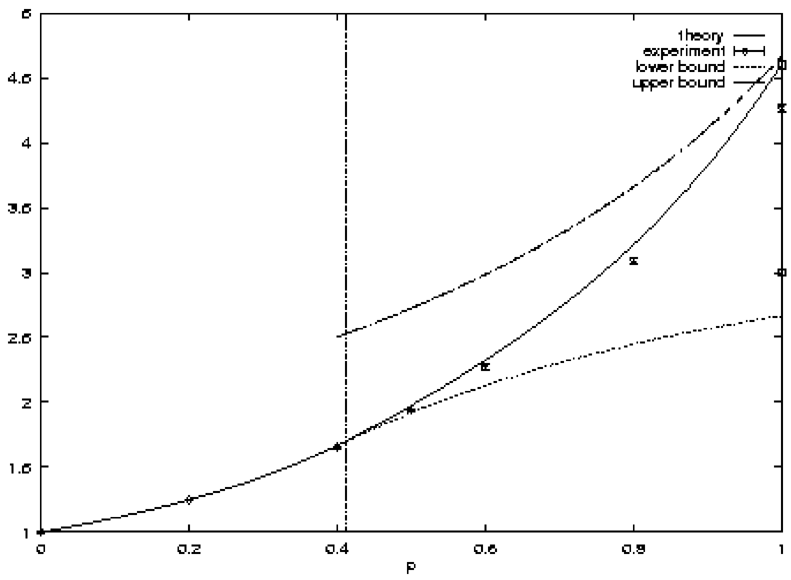
2+p-SAT phase transition



2+p-SAT phase transition



2+p-SAT phase transition

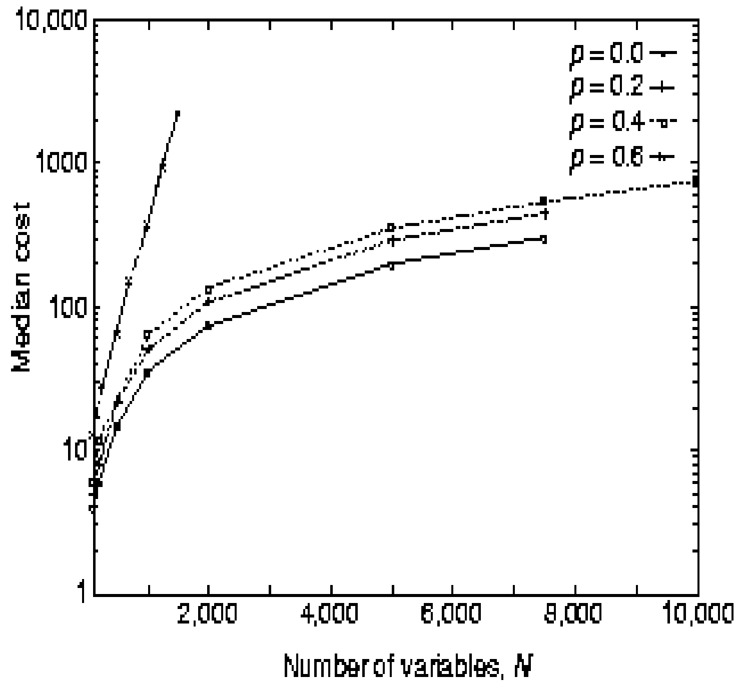


- Lower bound
  - are the 2-clauses (on their own) UNSAT?
  - n.b. 2-clauses are much more constraining than 3-clauses
- p <= 0.4
  - transition occurs at lower bound
  - 3-clauses are not contributing!

2+p-SAT backbone

- f_k becomes discontinuous for p>0.4
  - **but NP-complete for p>0 !**
- search cost shifts from linear to exponential at p=0.4
- similar behavior seen with local search algorithms

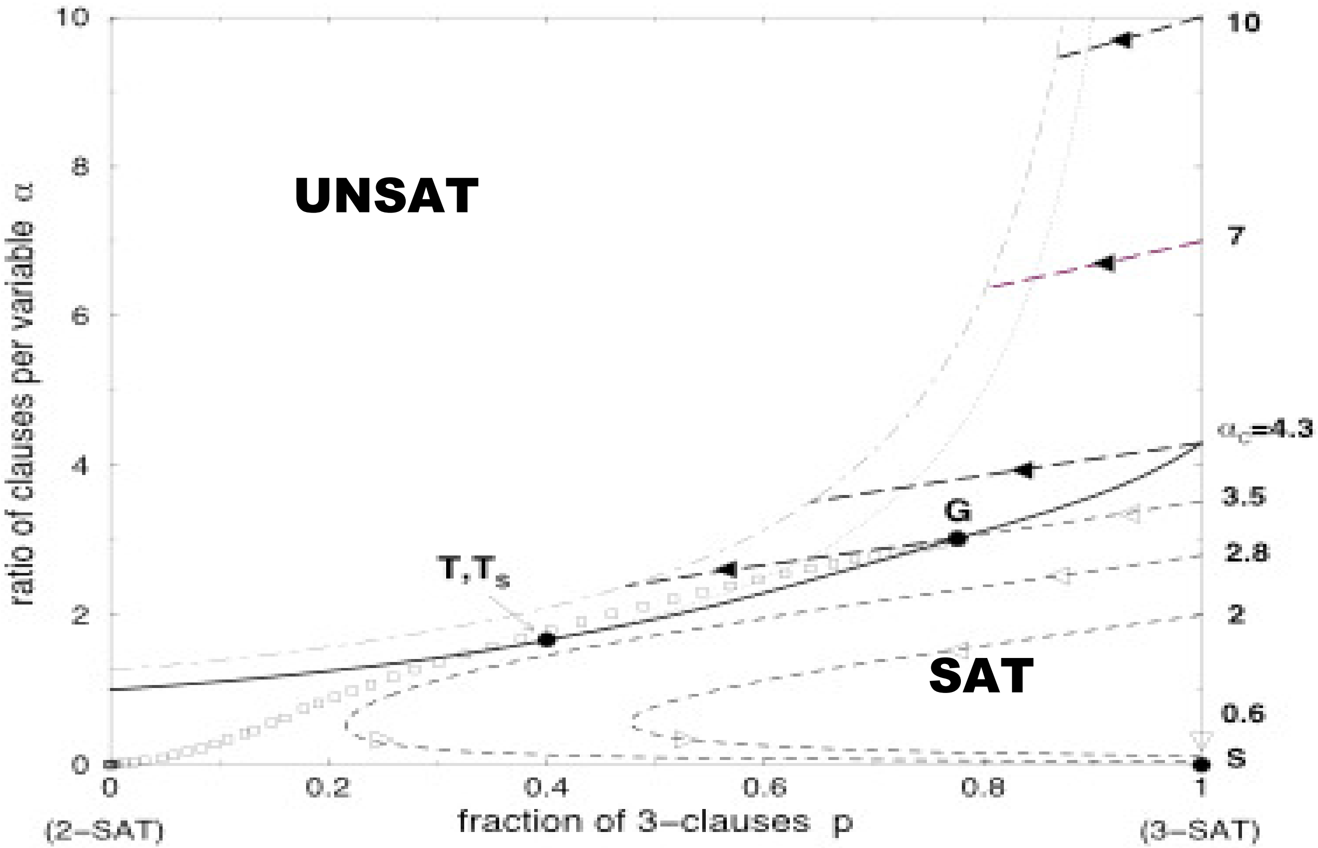
Search cost against n



2+p-SAT trajectories

- Input 3-SAT to a SAT solver like Davis Putnam
- REPEAT assign variable
  - **Simplify all unit clauses**
  - **Leaving subproblem with a mixture of 2 and 3-clauses**
- For a number of branching heuristics (e.g random,..)
  - **Assume subproblems sample uniformly from 2+p-SAT space**
  - **Can use to estimate runtimes!**

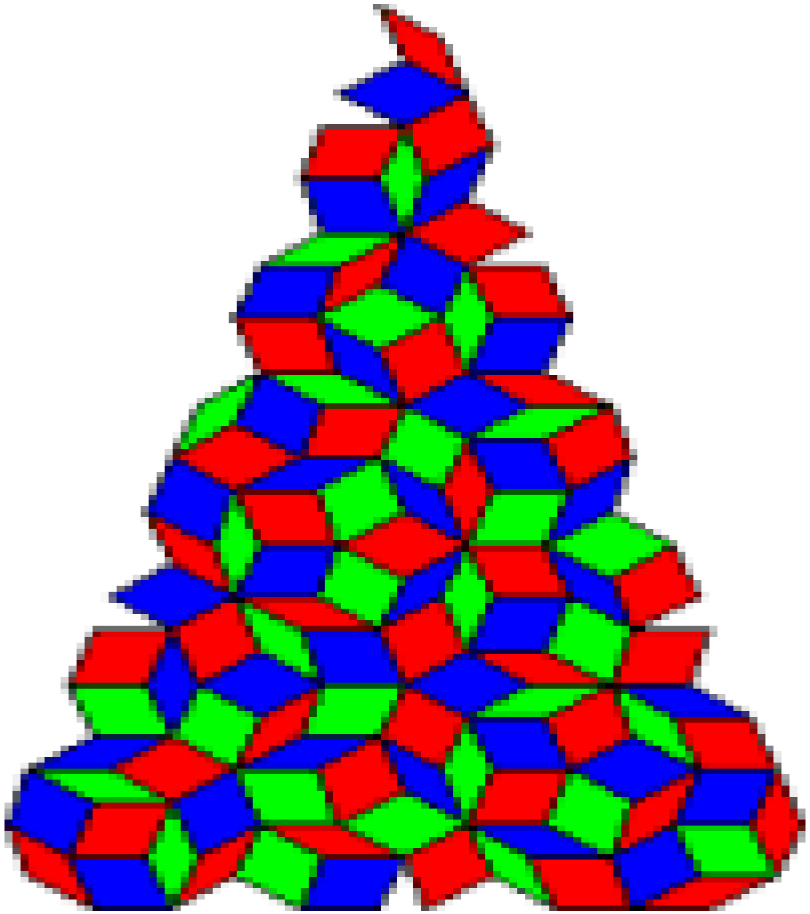
2+p-SAT trajectories



Beyond 2+p-SAT

- Optimization
  - **MAX-SAT**
- Other decision problems
  - **2-COL to 3-COL**
  - **Horn-SAT to 3-SAT**
  - **XOR-SAT to 3-SAT**
  - **1-in-2-SAT to 1-in-3-SAT**
  - **NAE-2-SAT to NAE-3-SAT**
  - **..**

COL



- Graph colouring
  - ***Can we colour graph so that neighbouring nodes have different colours?***
- In *k*-COL, only allowed *k* colours
  - **3-COL is NP-complete**
  - **2-COL is P**

Random COL

- Sample graphs uniformly
  - ***n*** **nodes and** ***e*** **edges**
- Observe colourability phase transition
  - **random 3-COL is "sharp",** ***e/n*** **$=_{approx}$ 2.3**
  - ***BUT*** **random 2-COL is not "sharp"**

As n->oo *prob(2-COL @ e/n=0) = 1*

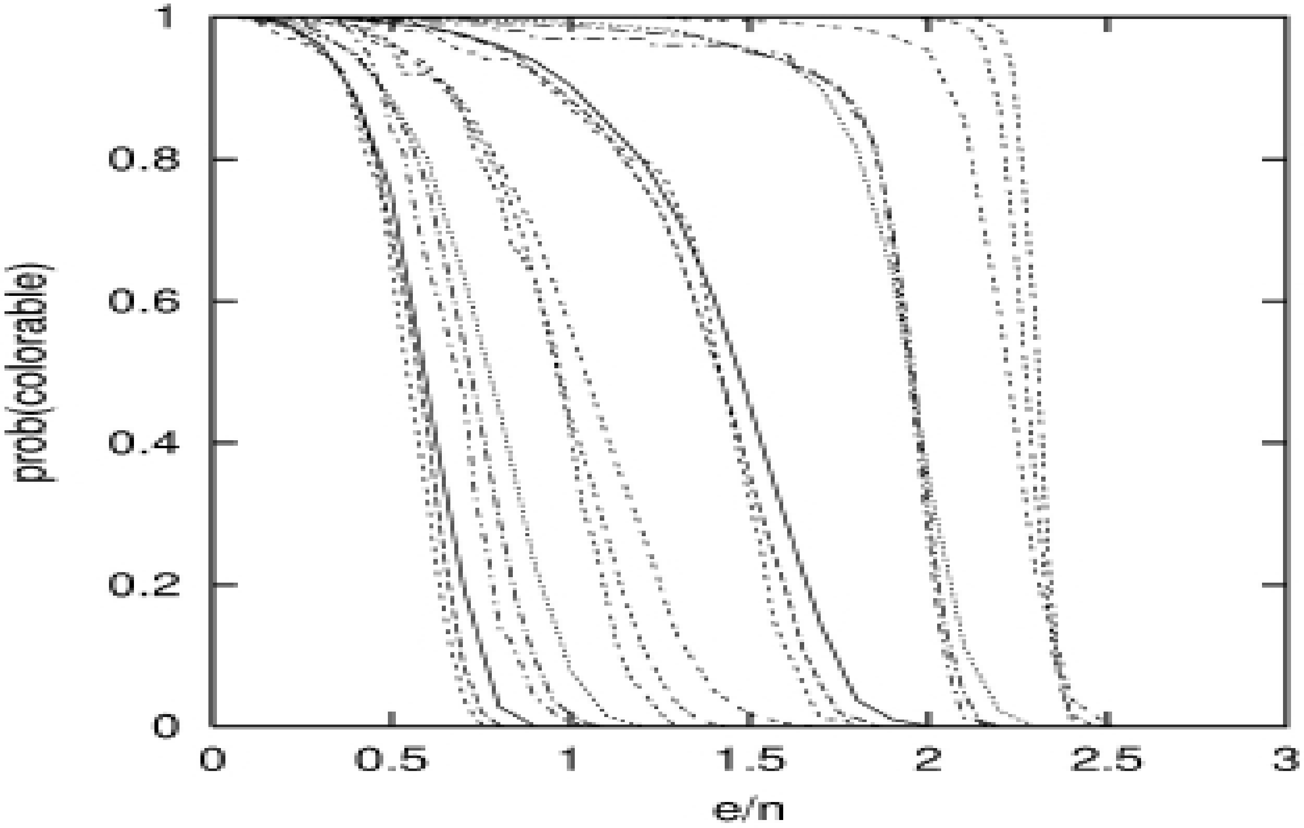
prob(2-COL @ e/n=0.45) $=_{approx}$ 0.5

prob(2-COL @ e/n=1) = 0

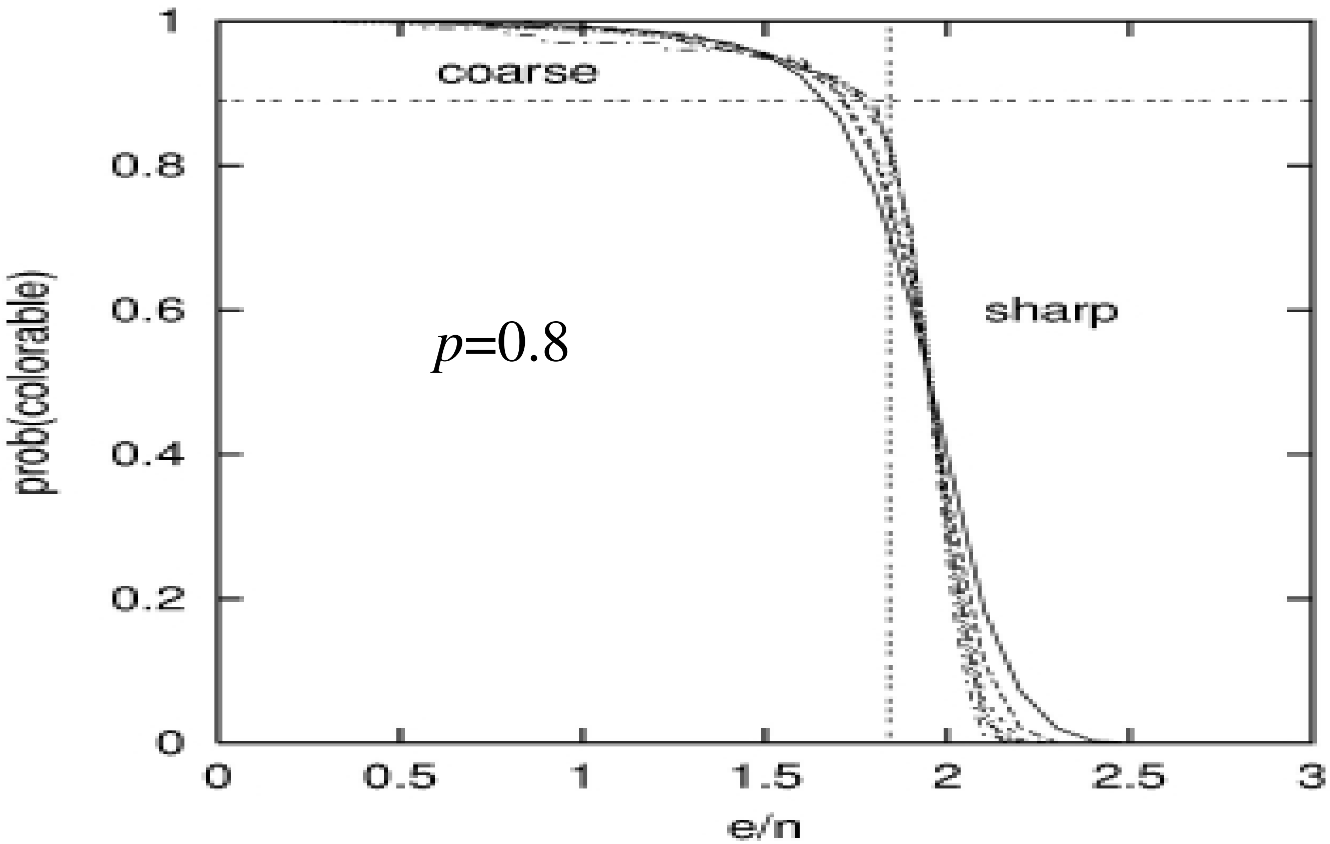
2+*p*-COL

- Morph from 2-COL to 3-COL
  - **fraction *p* of 3 colourable nodes**
  - **fraction *(1-p)* of 2 colourable nodes**
- Like 2+*p*-SAT
  - **maps from P to NP**
  - **NP for any fixed *p>0***
- Unlike 2+*p*-SAT
  - **maps from coarse to sharp transition**

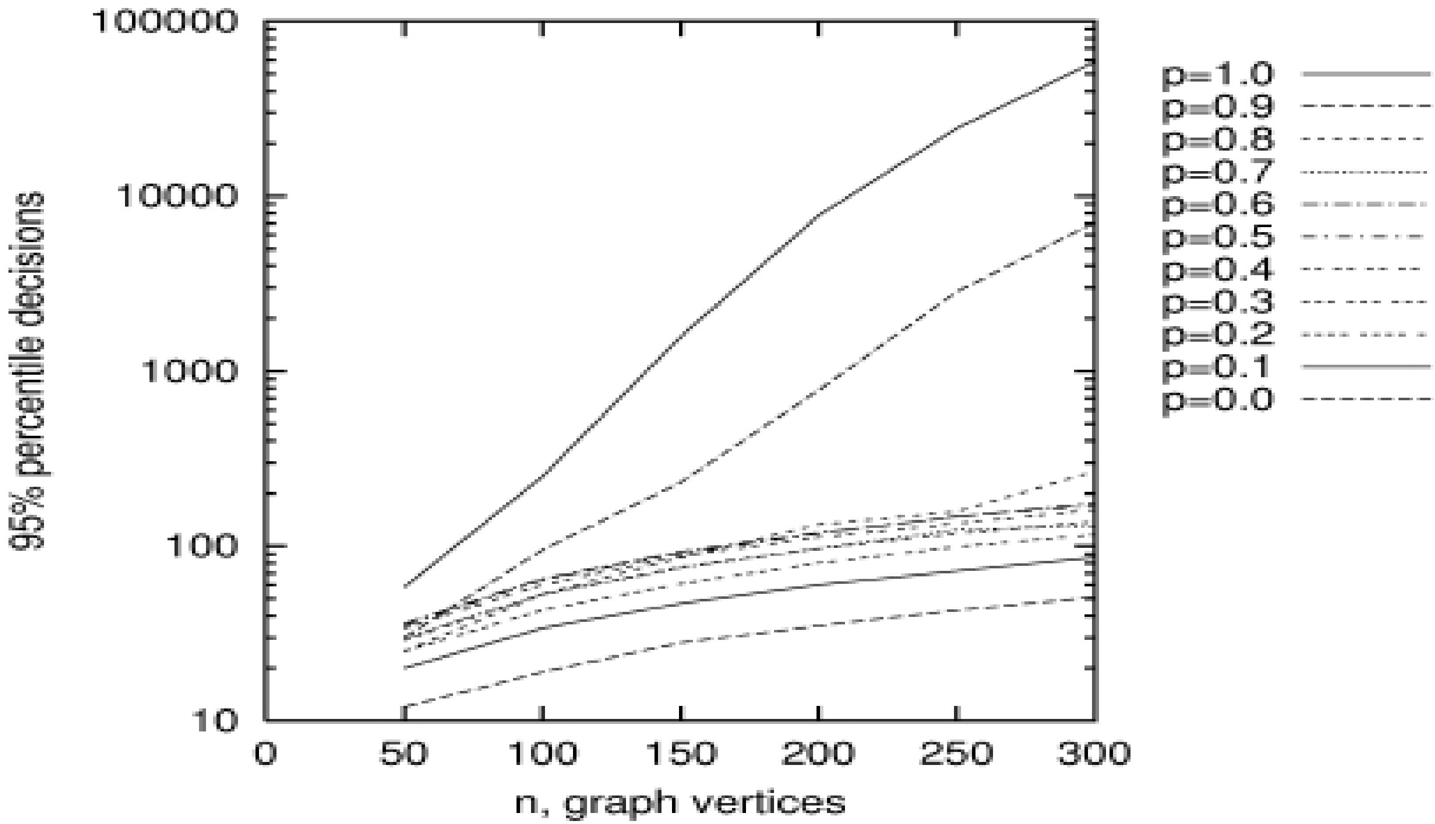
2+*p*-COL



2+*p*-COL sharpness



2+*p*-COL search cost



2+*p*-COL

- Sharp transition for $p>0.8$
- Transition has coarse and sharp regions for $0<p<0.8$
- Problem hardness appears to increase from polynomial to exponential at $p=0.8$
- 2+p-COL behaves like 2-COL for $p<0.8$
  - ***NB sharpness alone is not cause of complexity since 2-SAT has a sharp transition!***

Location of phase boundary

- For sharp transitions, like 2+*p*-SAT:

As n->oo, if l/n = c+epsilon, then UNSAT

l/n = c-epsilon, then SAT

- For transitions like 2+*p*-COL that may be coarse, we identify the start and finish:
  - ***delta2+p = sup{e/n | prob(2+p-colourable) = 1}***
  - ***gamma2+p = inf{e/n | prob(2+p-colourable) = 0}***

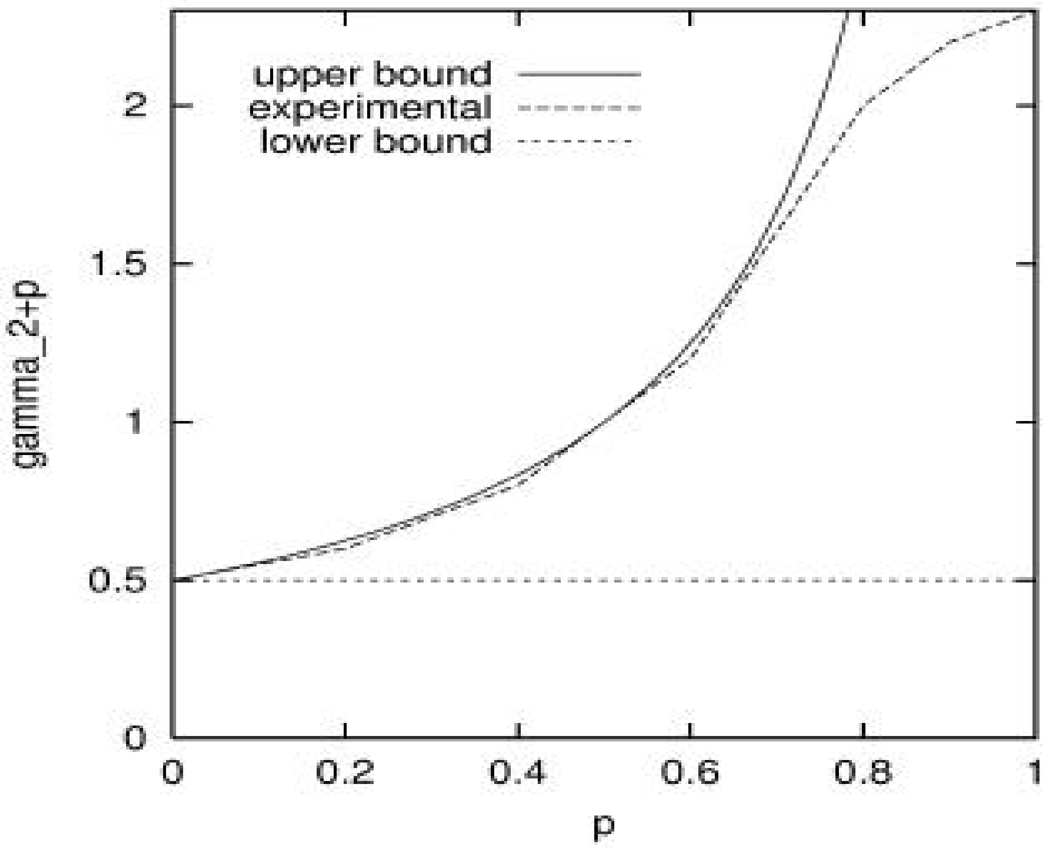
Basic properties

- **monotonicity: delta <= gamma**
- **sharp transition iff delta=gamma**
- **simple bounds:**

  delta_2+p = 0 for all p<1

  gamma_2 <= gamma_2+p <= min(gamma_3,gamma_2/1-p)

2+*p*-COL phase boundary



XOR-SAT

- XOR-SAT
  - **Replace or by xor**
  - **XOR *k*-SAT is in P for all *k***
- Phase transition
  - **XOR 3-SAT has sharp transition**
  - **0.8894 <= *l/n* <= 0.9278 *[Creognou et al 2001]***
  - **Statistical mechanics gives *l/n* = 0.918 *[Franz et al 2001]***

XOR Truth Table

Layer 1	Layer 2	Output
0	0	0
0	1	1
1	0	1
1	1	0

XOR-SAT to SAT

- Morph from XOR-SAT to SAT
  - **Fraction *(1-p)* of XOR clauses**
  - **Fraction *p* of OR clauses**
- NP-complete for all *p*>0
  - **Phase transition occurs at:**
  - **0.92 <= *l/n* <= min(0.92/1-*p*, 4.3)**
- Upper bound appears loose for all p>0
  - **Polynomial subproblem does not dominate!**
  - **3-SAT contributes (cf 2+*p*-SAT, 2+*p*-COL)**

Other morphs between P and NP

- NAE 2+*p*-SAT
  - **NAE = not all equal**
  - **NAE 2-SAT is P, NAE 3-SAT is NP-complete**
- 1-in-2+*p*-SAT
  - **1-in-*k* SAT = exactly one in *k* literals true**
  - **1-in-2 SAT is P, 1-in-3 SAT is NP-complete**
- …

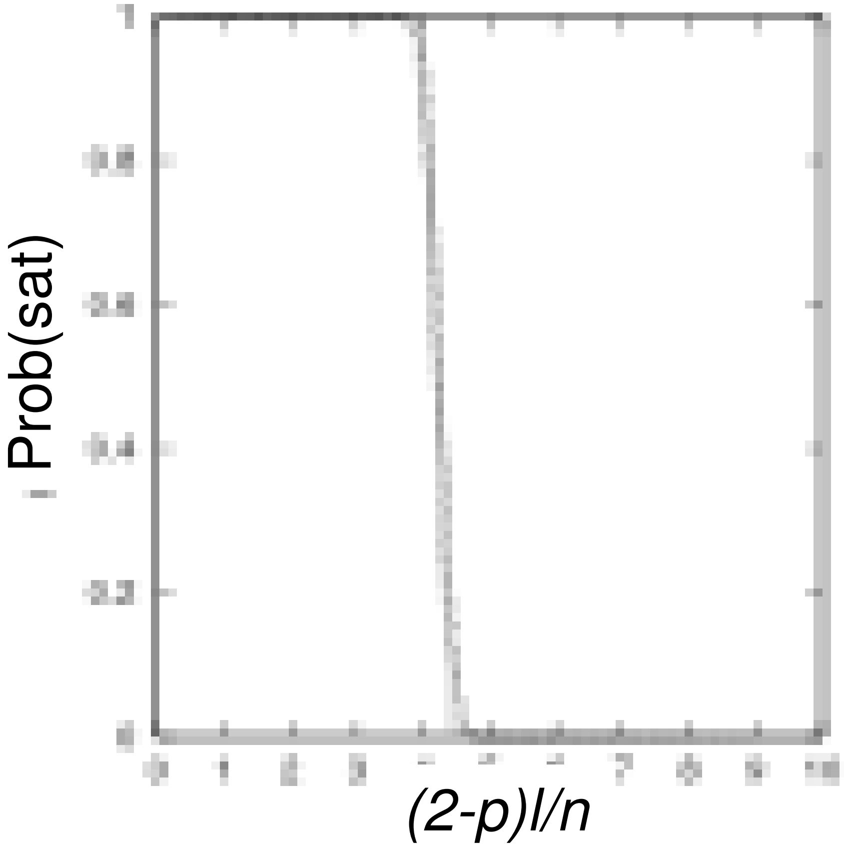
NAE to SAT

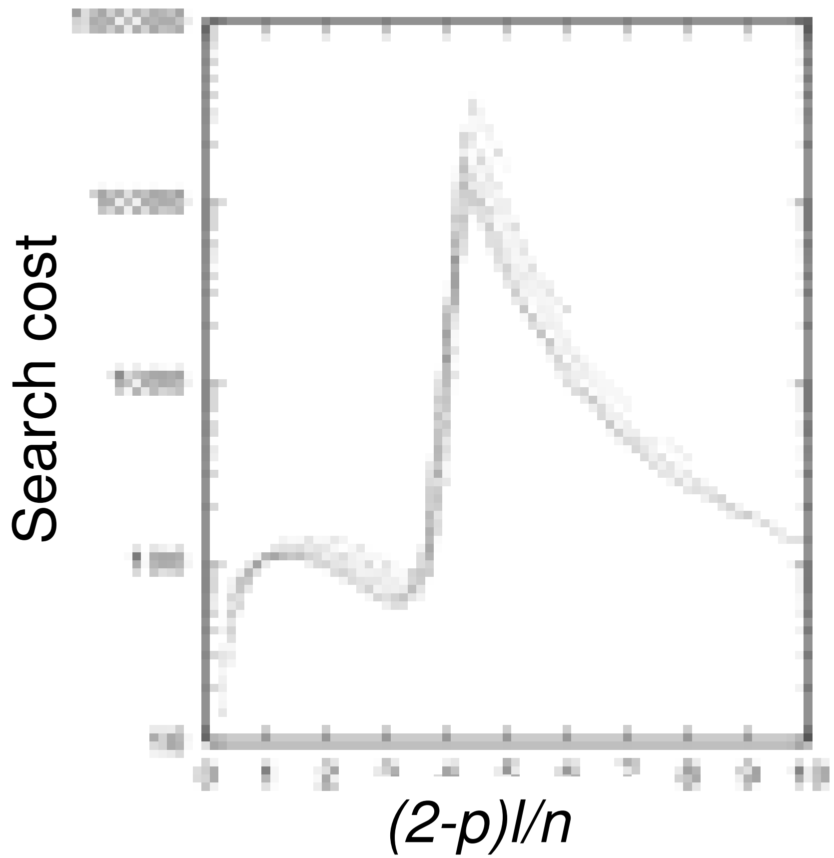
- Morph between two NP-complete problems
  - **Fraction *(1-p)* of NAE 3-SAT clauses**
  - **Fraction *p* of 3-SAT clauses**
- Each NAE 3-SAT clause is equivalent to two 3-SAT clauses
  - **NAE 3-SAT phase transition occurs around *l/n* = 2.1**
    - Tantalisingly close to half of 4.2
  - **NAE(a,b,c) = or(a,b,c) & or(-a,-b,-c)**
    - Can we ignore many of the correlations that this encoding of NAE SAT into SAT introduces?

NAE to SAT

- Compute "effective" clause size
  - **Consider *(1-p)l* NAE 3-SAT clauses and *pl* 3-SAT clauses**
  - **These behave like *2(1-p)l* 3-SAT clauses and *pl* 3-SAT clauses**
  - **That is, *(2-p)l* 3-SAT clauses**
  - **Hence, effective clause to variable ratio is *(2-p)l/n***
- Plot prob(satisfiable) and search cost against *(2-p)l/n*

NAE to SAT





Conclusions

- There's rich structure to be found between P and NP
- Problem classes like 2+p-SAT and 2+p-COL help us understand the onset of intractability
- NP-completeness isn't everything!
  - **Next lecture: the impact that structure has on problem hardness**