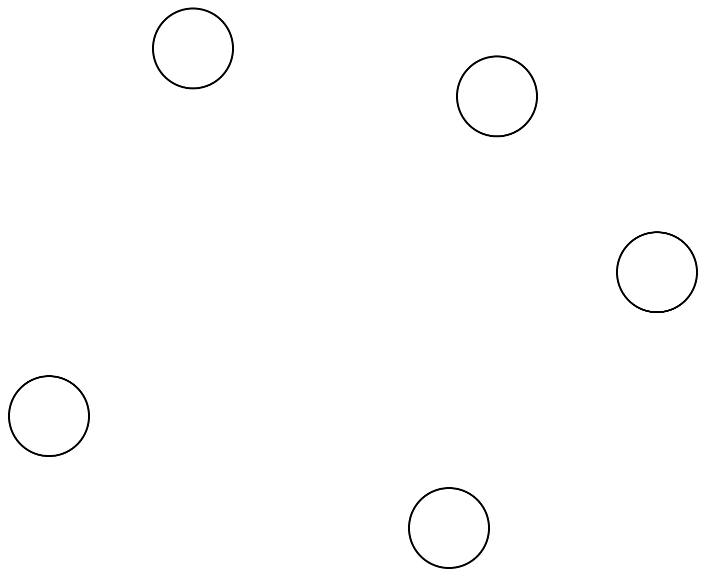
Where are the hard problems?

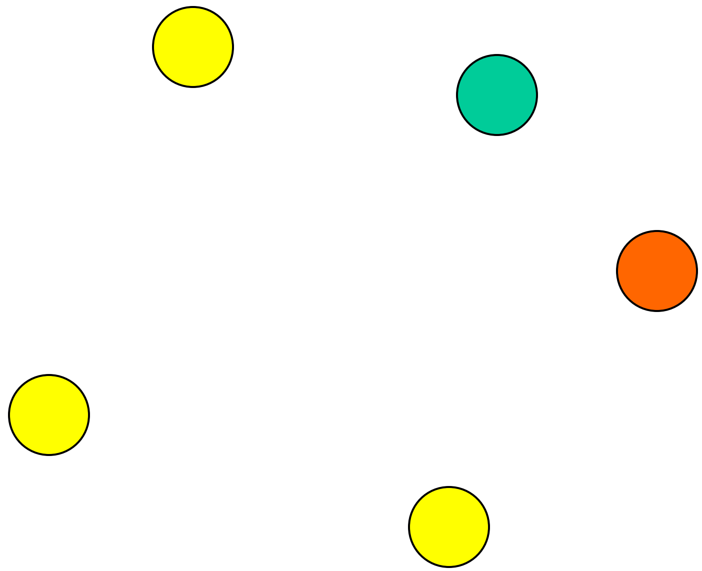
Remember Graph Colouring?

Remember 3Col?

3 Colour me?

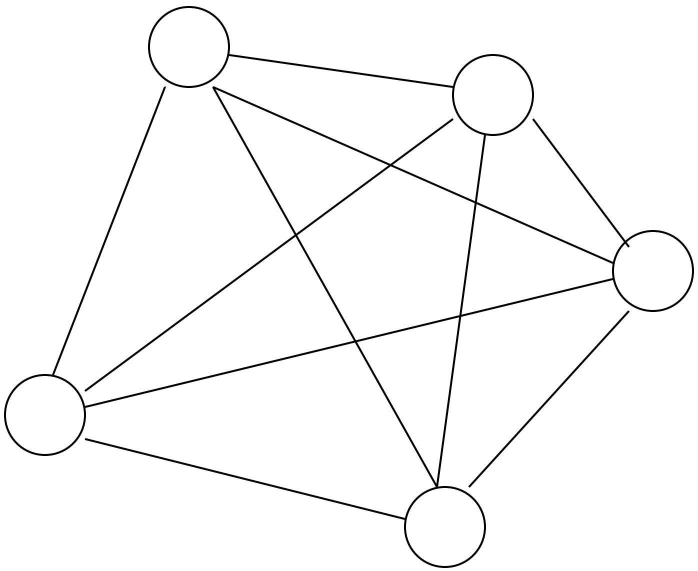


3 Colour me?

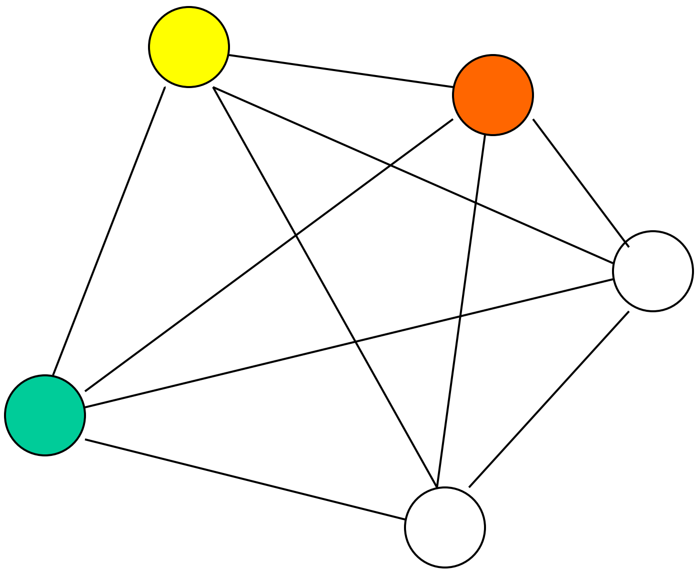


Easy?

3 Colour me?

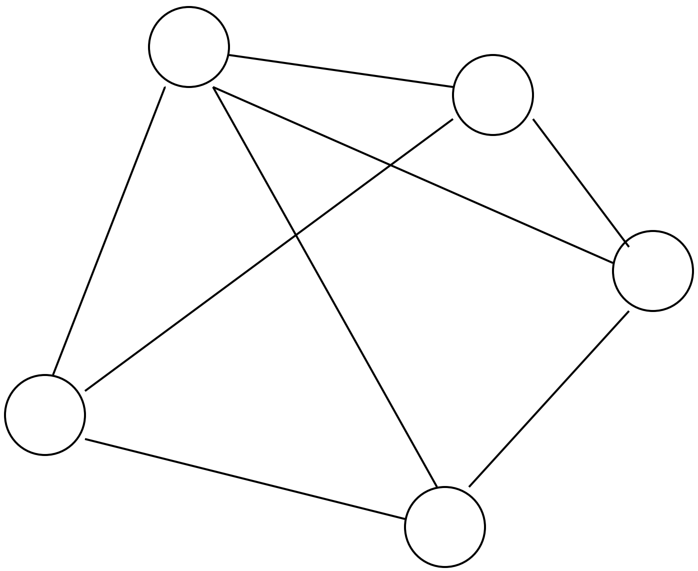


3 Colour me?

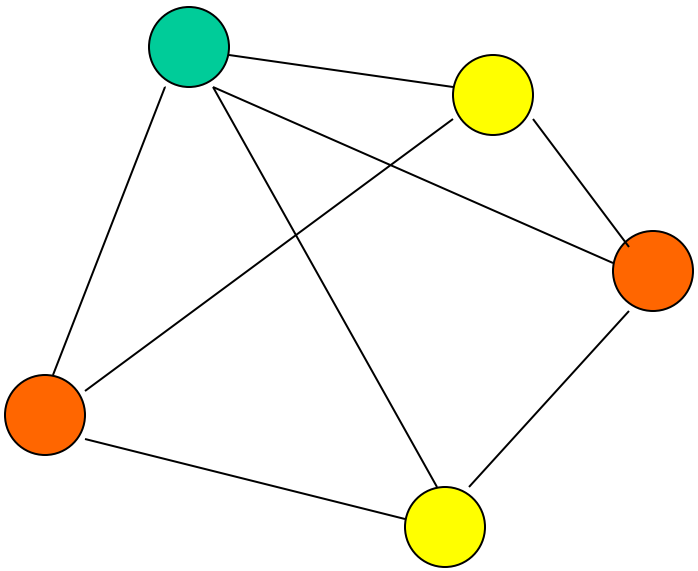


Easy?

3 Colour me?

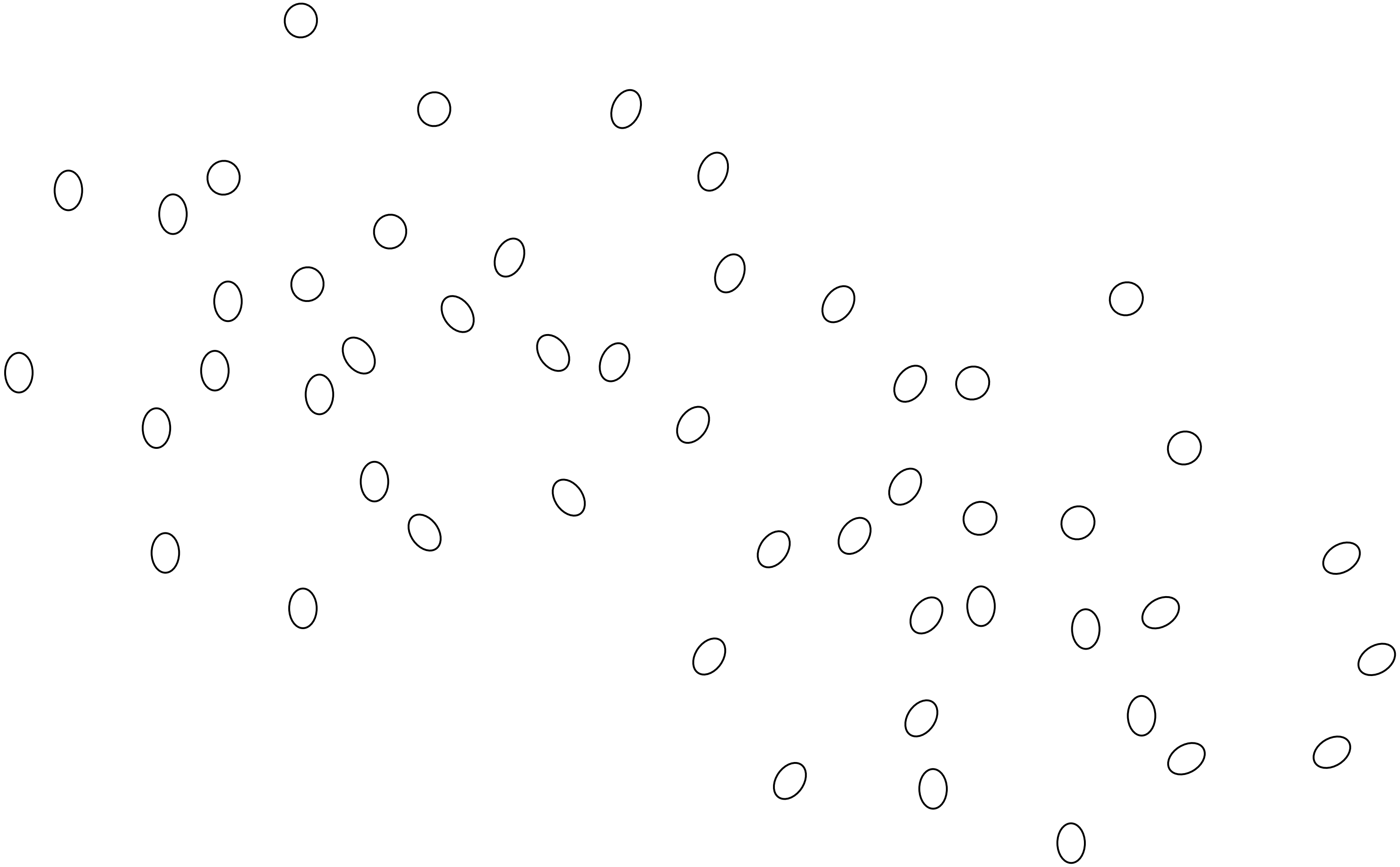


3 Colour me?



Easy?

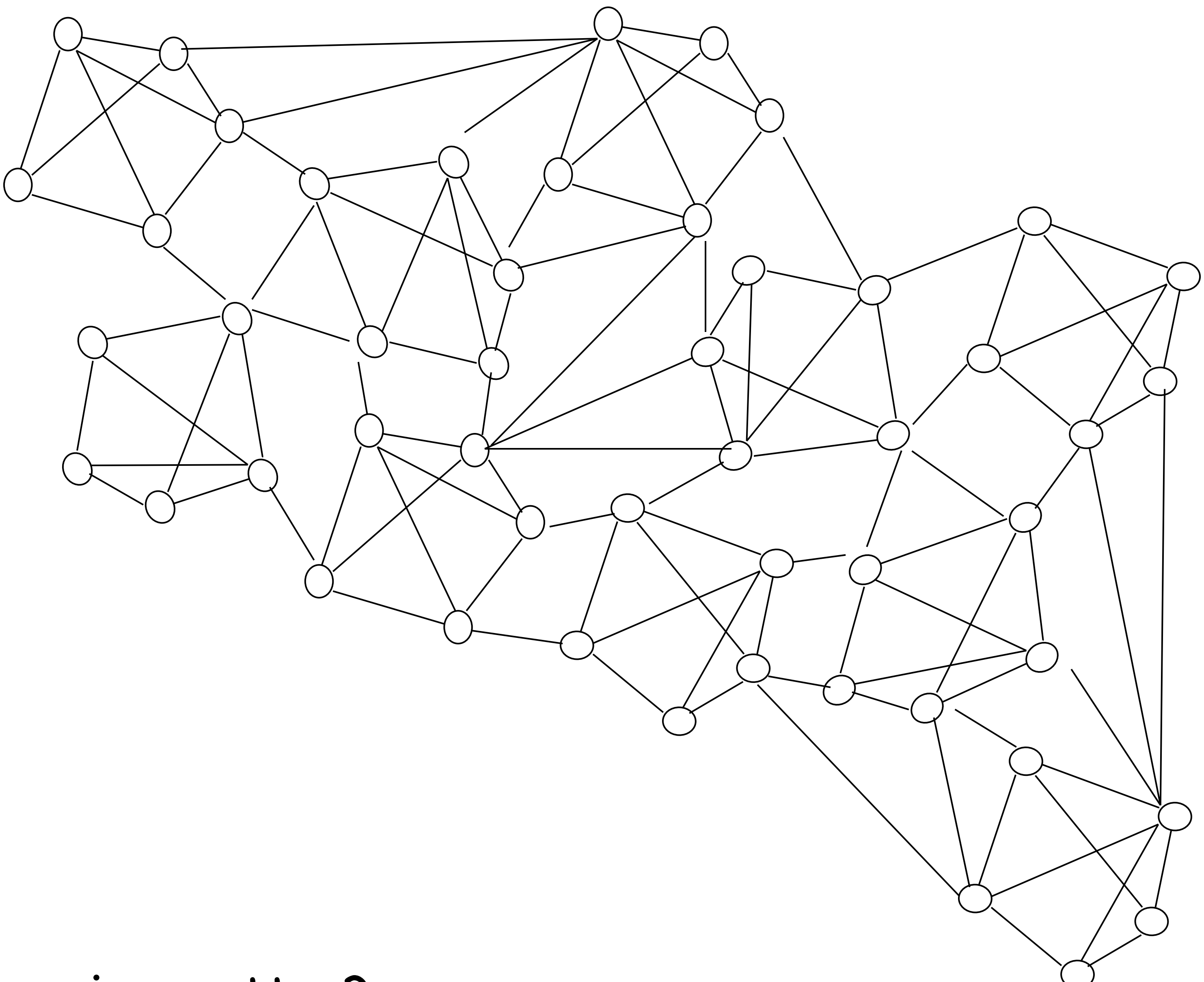
3 Colour me?



Does Size Matter?

Easy?

3 Colour me?



Does size matter?

So, Where are the hard problems?

Dr. Peter Cheeseman

RIACS

Affiliation: Research Institute for Advanced Computer Science
Postal address: Computational Sciences Division - Code IC
NASA Ames Research Center, MS 269-2
Moffett Field, CA 94035-1000
Voice: (650) 604-4946
Fax: (650) 604-3594
Electronic mail: cheesem@ptolemy.arc.nasa.gov

Career Summary

Intelligent Systems Division - Windows Internet Explorer

http://ti.arc.nasa.gov/

heeseman kanefsky and Taylor

Page Tools

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Search: + GO

+ABOUT NASA | +LATEST NEWS | +MULTIMEDIA | +MISSIONS | +MY NASA | +WORK FOR NASA

+ NASA Home
+ Ames Home

Intelligent Systems Division

+ Home
+ Organization
+ News
+ Research Areas
+ Publications
+ Software

Internal Resources

+ IT Systems Help

Search Intelligent Systems Division

GO

News

Technical Research Areas

Autonomous Systems and Robotics
Development of technologies required for systems that can adapt their behavior to complex, rapidly changing environments
+ Visit ASR

Collaborative & Assistant Systems
Information technologies and collaboration tools that facilitate the specialized work of distributed teams in NASA mission settings
+ Visit CAS

Discovery and Systems Health
Tools and methods for systems health management; large-scale science and aeronautical data analysis and data mining
+ Visit DaSH

Robust Software Engineering
Increased software quality, reliability, and productivity through research done in the context of NASA applications
+ Visit RSE

Highlights

Mission Technologies

+ LCROSS
+ Phoenix Mars Lander
+ Mars Exploration Rovers
+ Mars Science Laboratory

Featured Projects

LCROSS Flight Director's Blog
+ Crewed NEO Mission Feasibility Study
+ Robotic Recon Experiment at Black Point Lava Flow
+ Planetary Content
+ Spaceflight Training Management

Events

ADT 2009
Location: Venice, Italy
Date: 10-21-09 to 10-23-09

EMICS 2009

start | slides | 2 Internet ... | emacs@BIAK | Microsoft Po... | 11:13

Planning and Scheduling

Overview

The Planning and Scheduling Group builds automated planning and scheduling systems for NASA missions. These planning and scheduling systems are essential components of autonomous spacecraft, deep space probes, planetary rovers, and autonomous vehicles. They are also integral elements of ground operations tools for space missions. Planning and Scheduling Group activities include: Conducting basic research into computational methods used in automated planning and scheduling systems. Designing and implementing automated planners as components of autonomous systems. Designing and implementing planning and scheduling tools used by mission operators.

Project List (Active)

Activity Planning and Sequencing for Mars Science
Project Lead: James Kurien

Constraint Based Planning
Project Lead: Conor McGann
Development of planning techniques and software for domains with complex temporal and resource constraints.

Decision Theoretic Planning for Planetary Exploration

Team

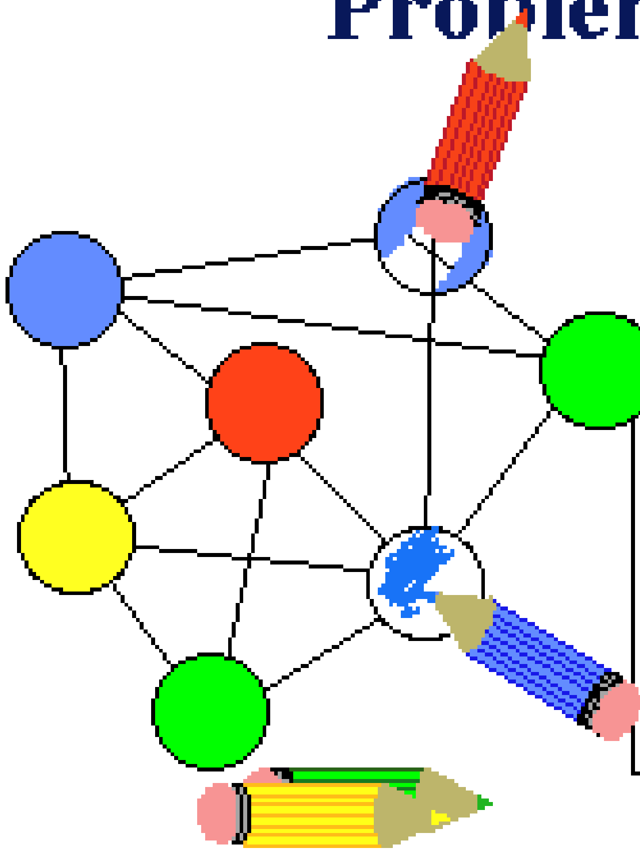
Group Lead
Jeremy Frank

Group Members
Andrew Bachmann
John L Bresina
Kevin Greene
Peter A. Jarvis
Bob Kanefsky
Lina Khatib
James Kurien
Elif Kurklu
Tony Lindsey
Conor McGann
Robert Morris
Paul Morris
David Smith
Stephen Wragg

Search
Intelligent Systems Division
GO

Ames Research Center
Artificial Intelligence Branch

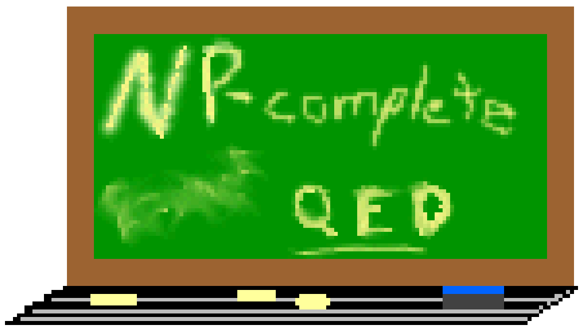
Where the *Really* Hard Problems Are



Peter Cheeseman
RIACS
Bob Kanefsky
Sterling Software
William Taylor
Sterling Software

Problem:

We were working on a
new algorithm for constraint-satisfaction problems.
We needed **hard problems** to try it out on.



An NP-complete problem like Graph-Colorability should provide plenty of hard cases, right?

Surprise!

There's already a backtracking algorithm (by Brélaz) that solves large random graphs in **linear** time.

Q: What does this say about NP problems?

Wots NP?

Nondeterministic Polynomial
Problems that cannot be solved in polynomial (P) time
… as far as we know

NP-Complete (NPC)
If a polytime alg can be found for any NPC problem
Then it can be adapted for all NPC problems

Theory shows NP problems are worst case exponential (if P≠NP)

but says nothing about complexity in a particular case!

Intractability Theorem *The problem of determining whether a proposition is necessarily true in a nonlinear plan whose action representation is sufficiently strong to represent conditional actions ... is NP hard.*

Hoping for the best amounts to arguing that for the particular cases that come up in practice, extensions to current planning techniques will happen to be efficient. My intuition is that this is not the case.

— D. Chapman, A.I.J. (1987)

Where are the hard problems in NP?

- Pathological?
- Unpredictable?

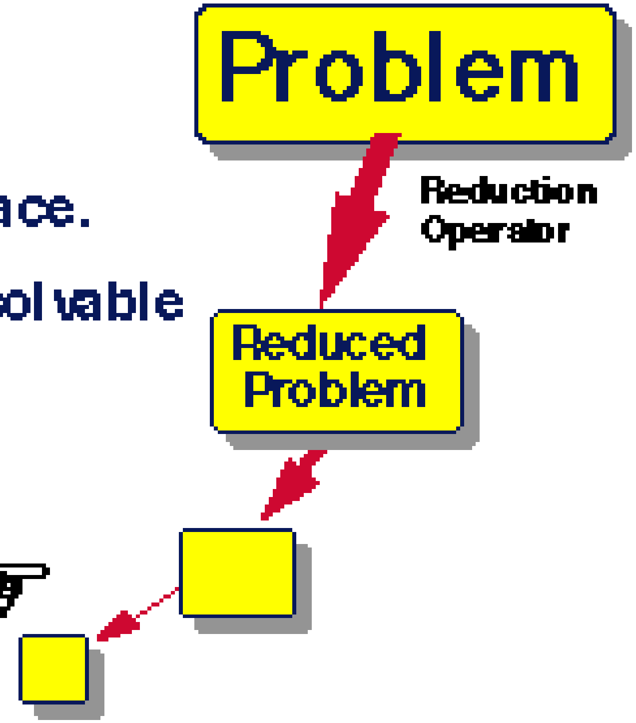
<u>Partial answer:</u>

Not in 'trivial' part of space.

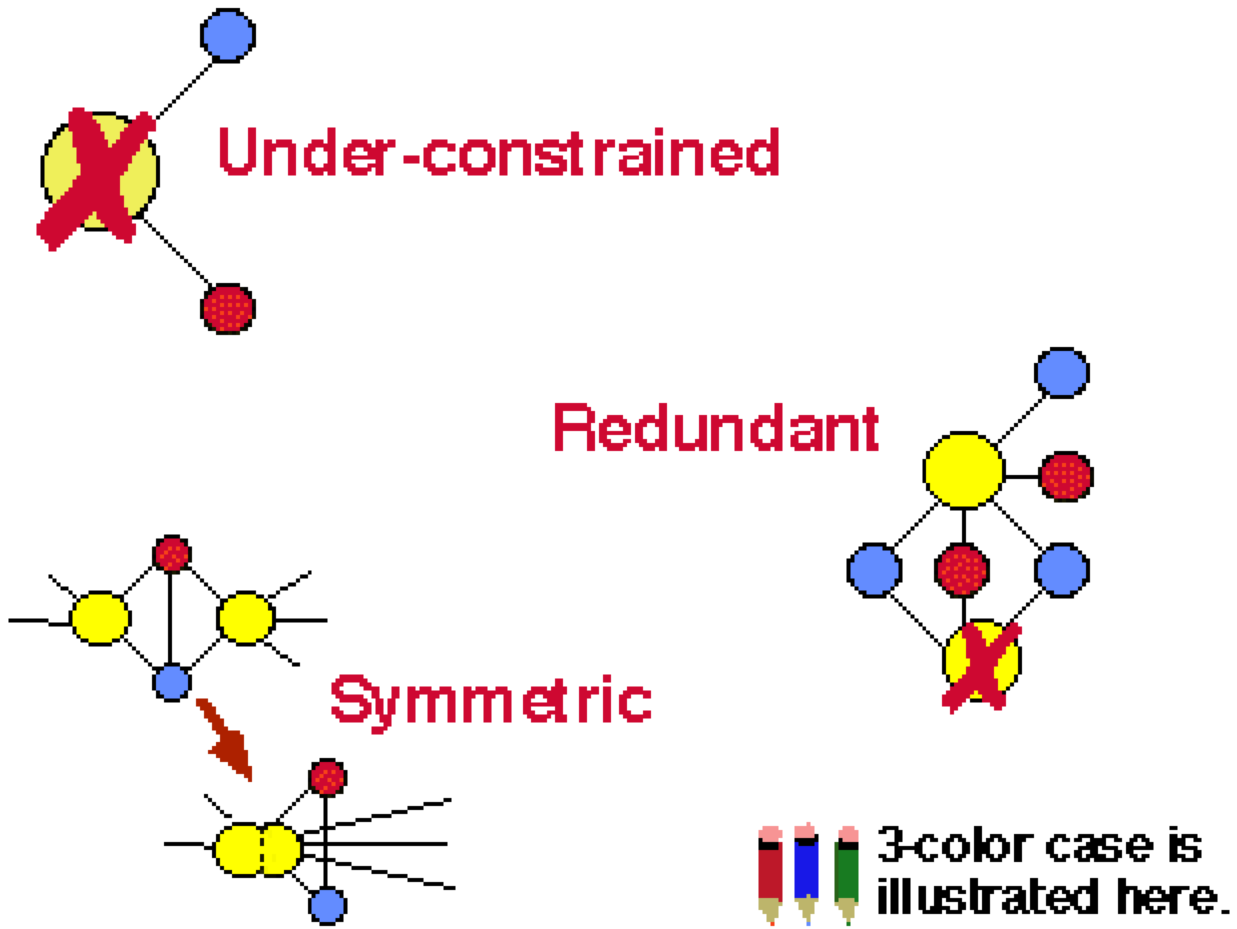
Eliminate all problems solvable without search.

Reduction Operators.

(Can be applied recursively)

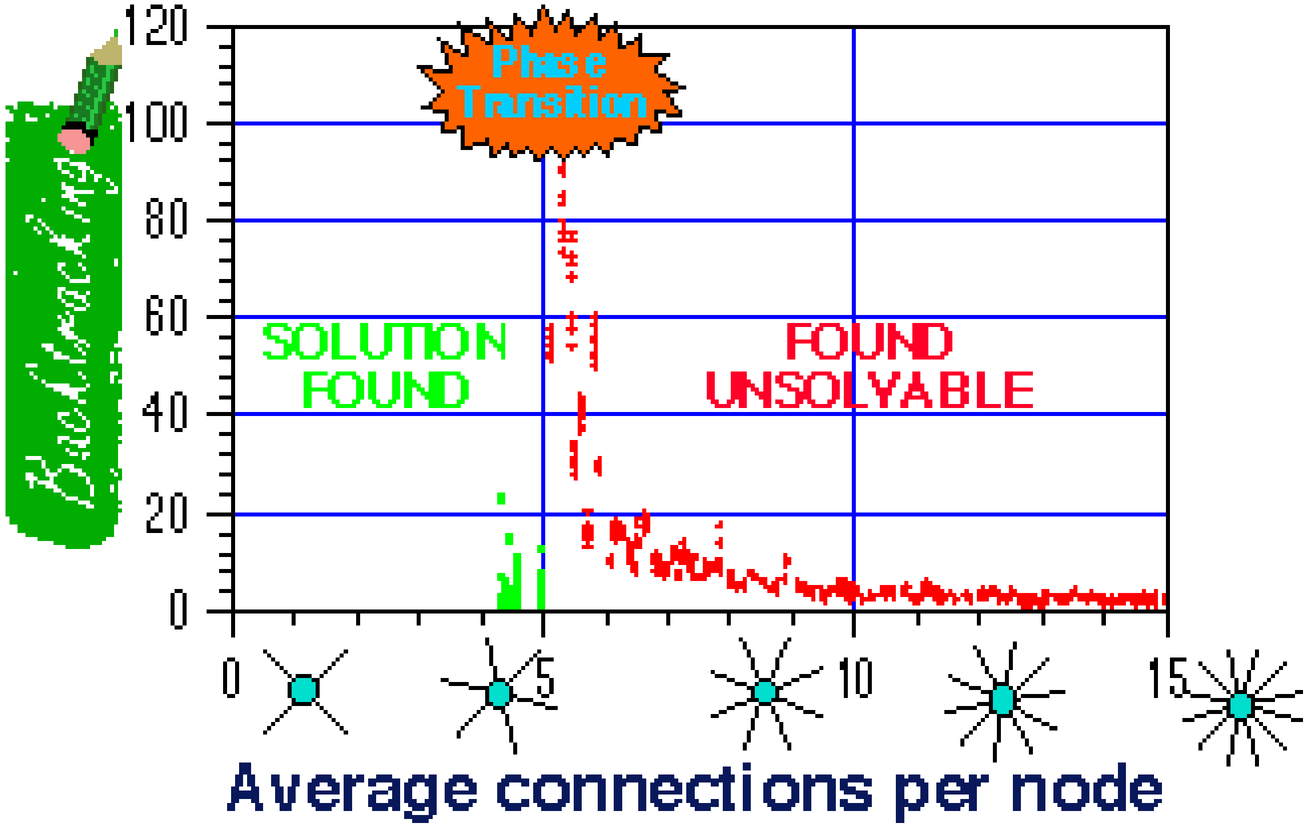


Graph Reduction Operations



3-colorability cost and solvability

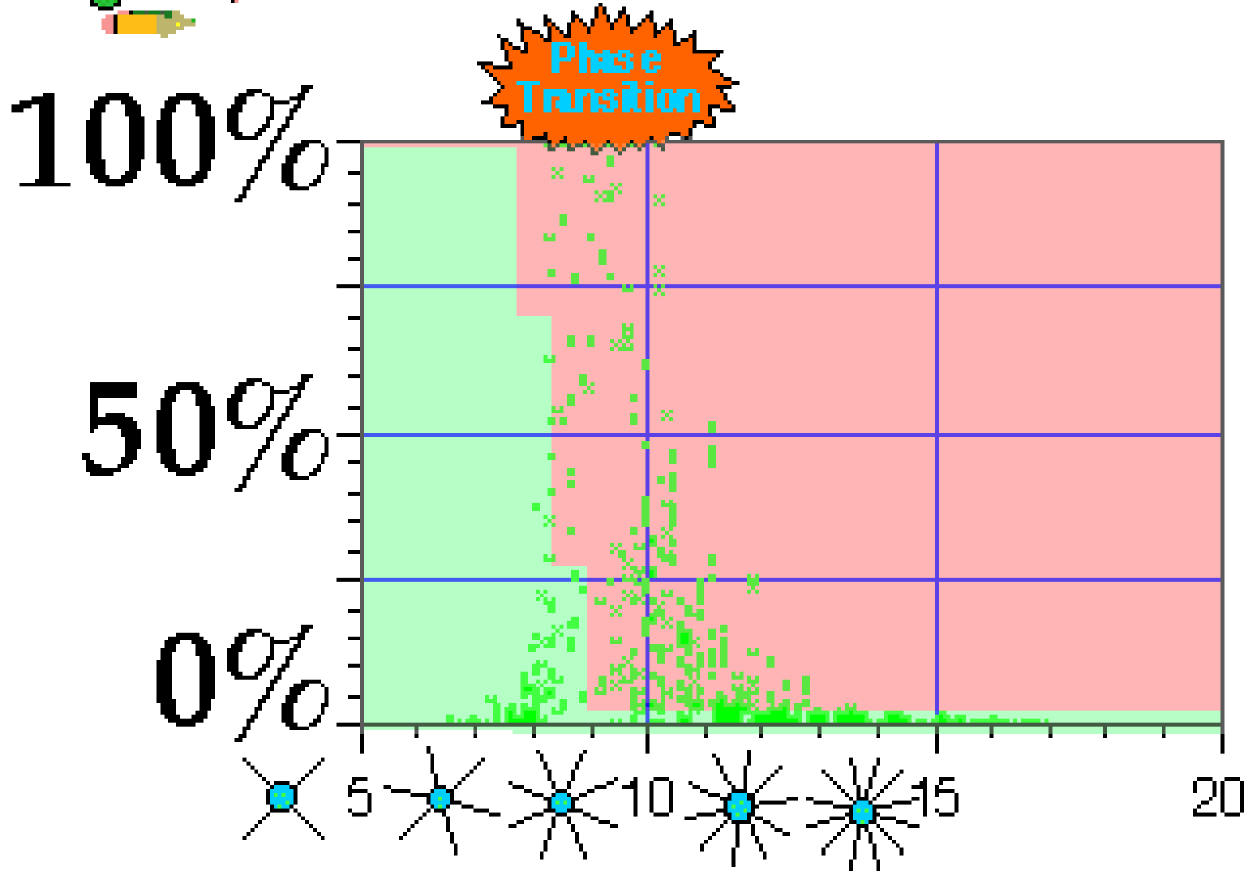
Brélaz's Algorithm



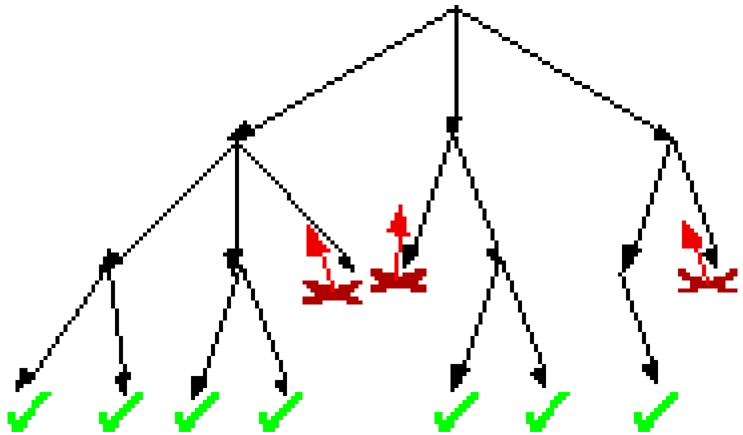
prevalence of 4-colorable graphs

4-colorability

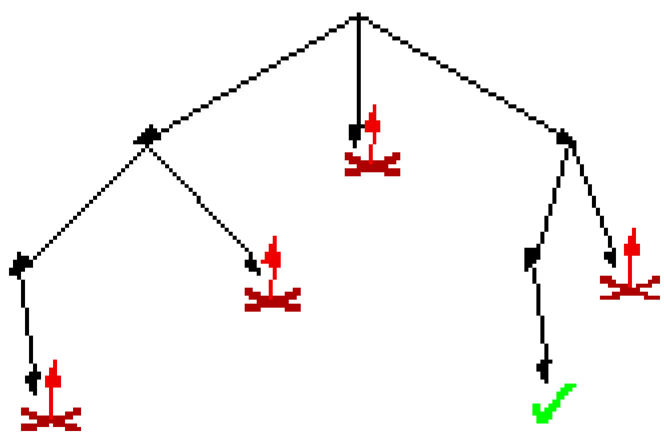
(~~solvable~~, irreducible 144-node graphs)



Under-constrained = almost any solution works

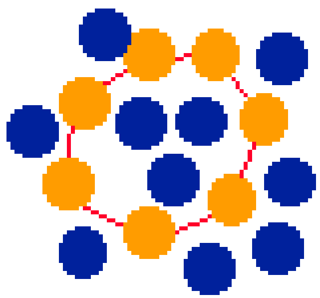


Over-constrained = few false trails.

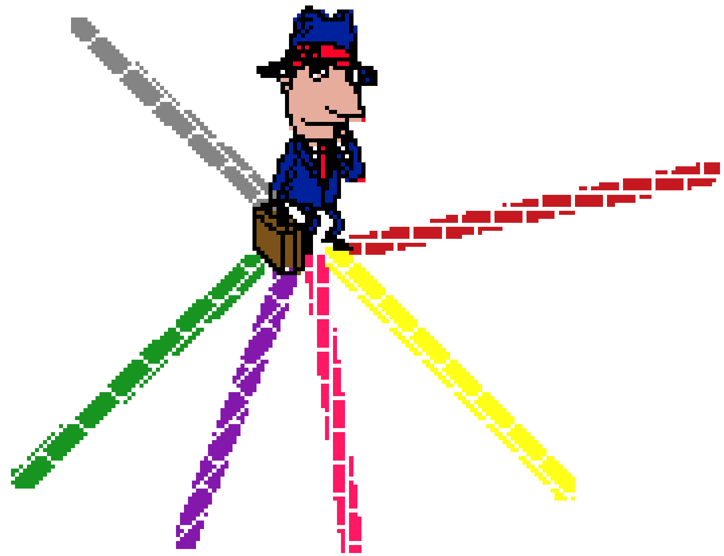


Do other NP problems have phase transitions?

<u>Hamilton Circuit</u>



<u>Traveling Salesman</u>



<u>*k*sat</u>

$a \vee b \vee \neg c$

&

$d \vee \neg a \vee e$

Wot's SAT?

Toby?

Propositional Satisfiability

- SAT
  - does a truth assignment exist that satisfies a propositional formula?
  - special type of constraint satisfaction problem
    - Variables are Boolean
    - Constraints are formulae
  - NP-complete

- 3-SAT
  - formulae in clausal form with 3 literals per clause
  - remains NP-complete

(x1 v x2) & (-x2 v x3 v -x4)

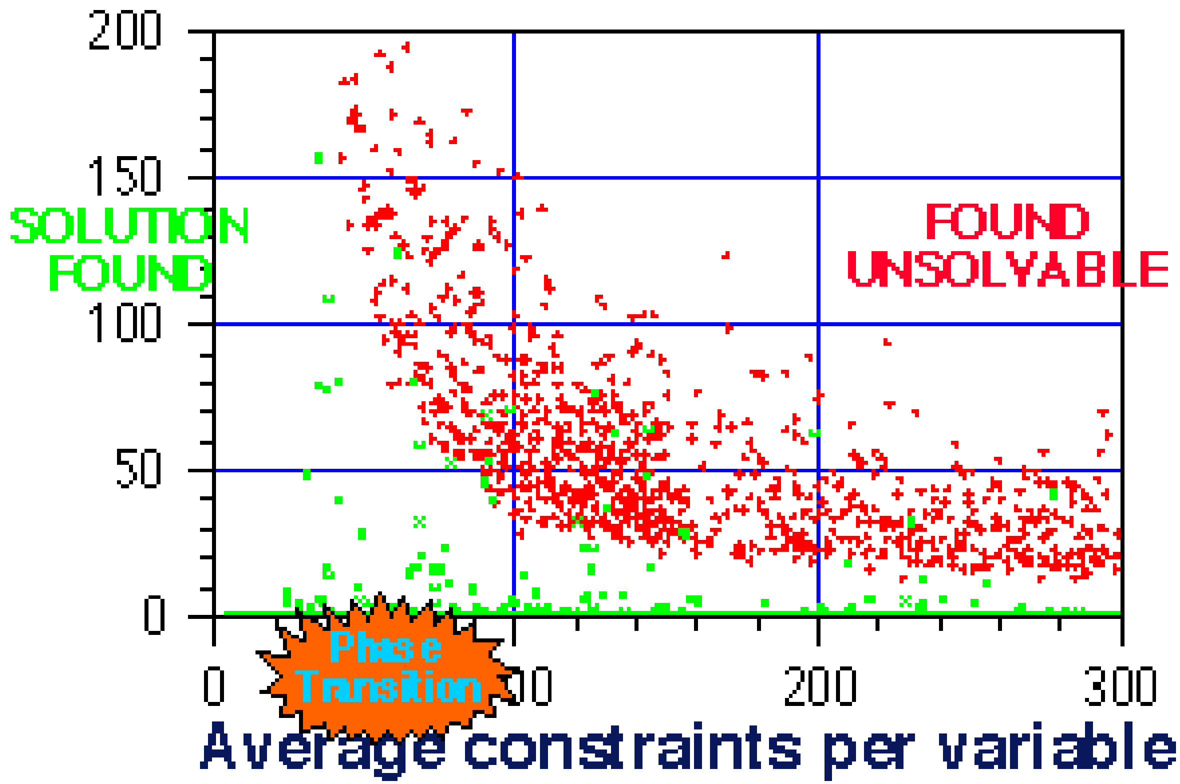
x1/ True, x2/ False, ...

Wots complexity of 3SAT?

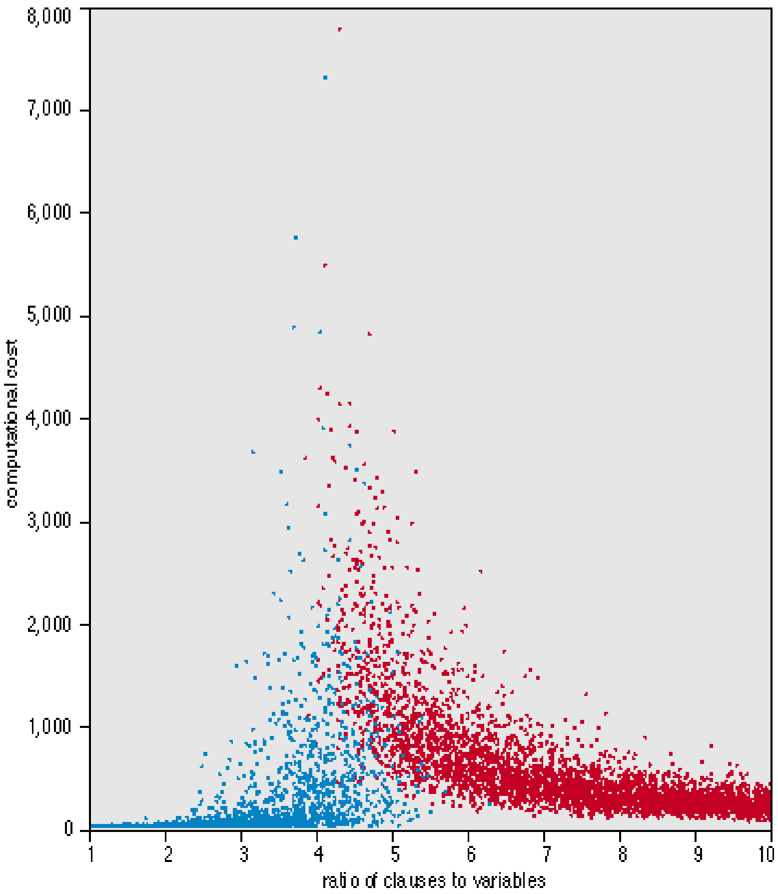
4-sat cost

a,b,c,...,y
25 var.

&



Random 3-SAT



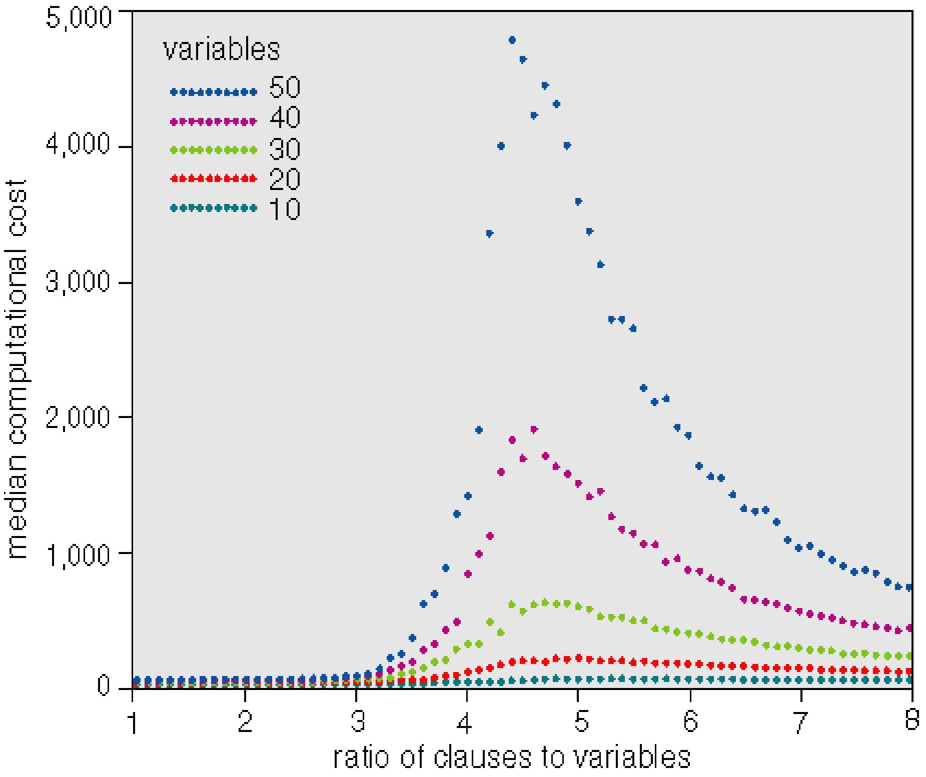
- Random 3-SAT
  - sample uniformly from space of all possible 3-clauses
  - *n* variables, *l* clauses
- Which are the hard instances?
  - around $l/n = 4.3$

What happens with larger problems?

Why are some dots red and others blue?

Random 3-SAT

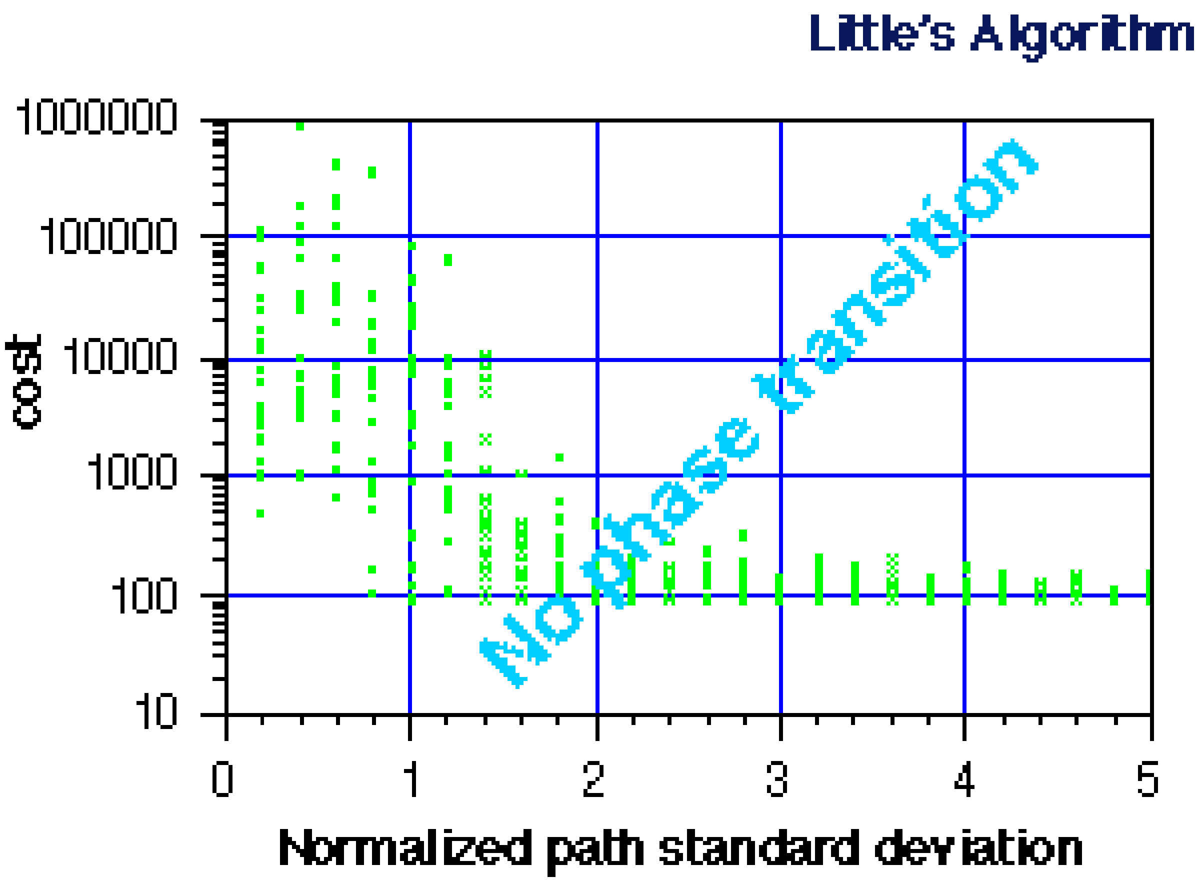
- Varying problem size, *n*
- Complexity peak appears to be largely invariant of algorithm
  - backtracking algorithms like Davis-Putnam
  - local search procedures like GSAT



What's so special about 4.3?

Traveling Salesman

cost for 48 cities



ELSEVIER

Artificial Intelligence 88 (1996) 349-358

Artificial Intelligence

Research Note

The TSP phase transition

Ian P. Gent [a,*], Toby Walsh [b,c,1]

[a] *Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, UK*
[b] *Mechanized Reasoning Group, IRST, Trento, Italy*
[c] *DIST, University of Genoa, Genoa, Italy*

Received October 1995; revised July 1996

Abstract

The traveling salesman problem is one of the most famous combinatorial problems. We identify

Ian P. Gent [a,*], Toby Walsh [b,c,1]

[a] *Department of Computer Science, University of Strathclyde, Glasgow G1 1XH, UK*
[b] *Mechanized Reasoning Group, IRST, Trento, Italy*
[c] *DIST, University of Genoa, Genoa, Italy*

Received October 1995; revised July 1996

Abstract

The traveling salesman problem is one of the most famous combinatorial problems. We identify a natural parameter for the two-dimensional Euclidean traveling salesman problem. We show that for random problems there is a rapid transition between soluble and insoluble instances of the decision problem at a critical value of this parameter. Hard instances of the traveling salesman problem are associated with this transition. Similar results are seen both with randomly generated problems and benchmark problems using geographical data. Surprisingly, finite-size scaling methods developed in statistical mechanics describe the behaviour around the critical value in random problems. Such phase transition phenomena appear to be ubiquitous. Indeed, we have yet to find an NP-complete problem which lacks a similar phase transition.

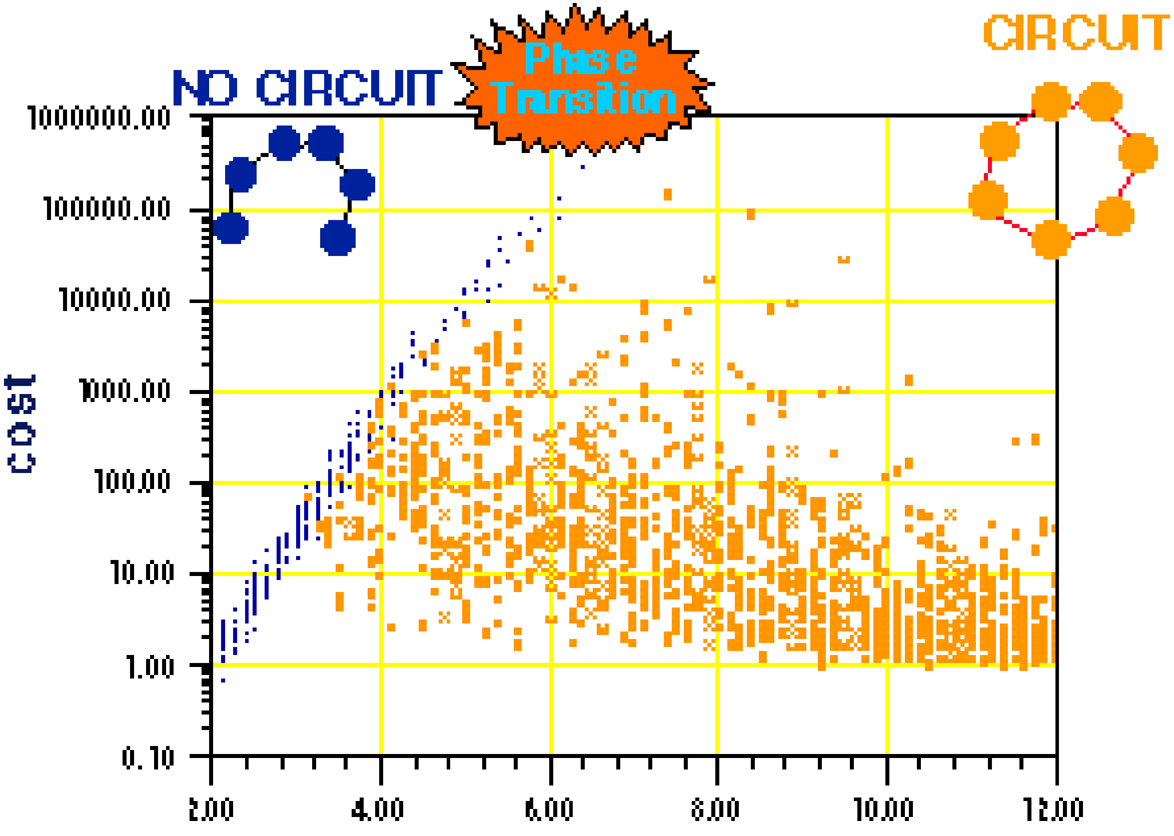
Keywords: NP-complete problems; Complexity; Traveling salesman problem; Search phase transitions; Finite-size scaling; Eeasy and hard instances

---

1. Introduction

There are many useful connections between statistical mechanics and a wide variety

Hamilton Circuit search cost



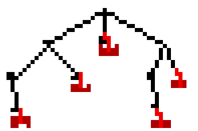
Why “Phase Transition”?

- Transition separates 2 regions

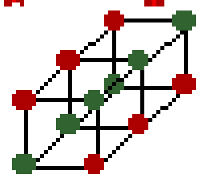
Under-constrained — Over-constrained

SOLID — Liquid

- Strong “fluctuations” near transition



Depth of backtrack before failure



Crystal nuclei

- Sharpness of transition increases with size (boundary effects)

- Phase transitions = macroscopic order

  Average over individual interactions.

Speculation

- All hard problems occur at a phase transition.
  - Hard problems are predictable.
- Much more to complexity than P or NP.
  - Depth of backtrack related to order of constraints.
  - Depth of backtrack maximum at phase transitions.

- CKT were first to report the phenomenon
- Were they the first to see it?

Feldman and Golumbic 1990
Student Scheduling Problems

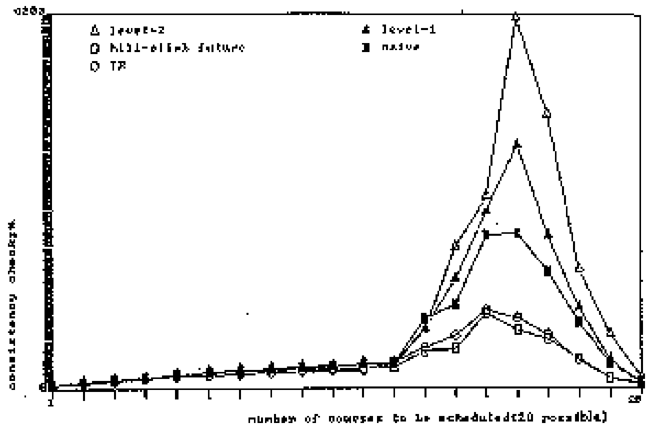


Fig. 4. Ordering algorithms – forward checking.

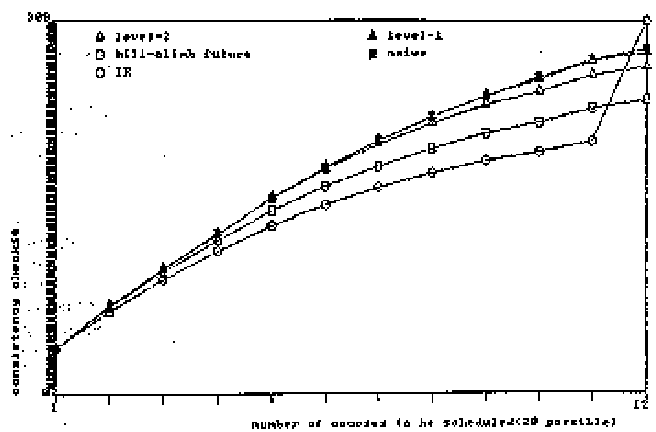


Fig. 5. Ordering algorithms – forward checking.

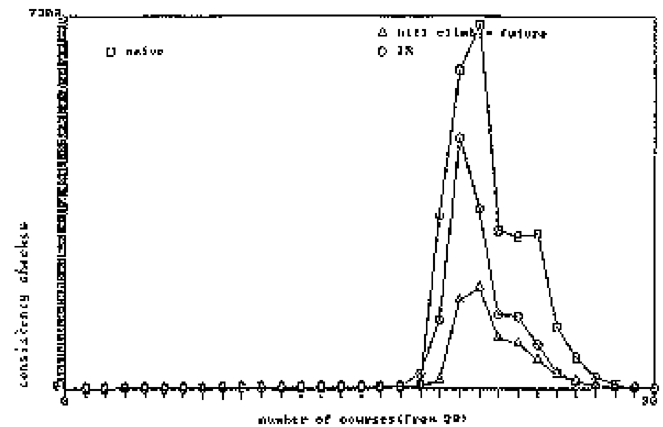


Fig. 6. Ordering algorithms – forward checking.

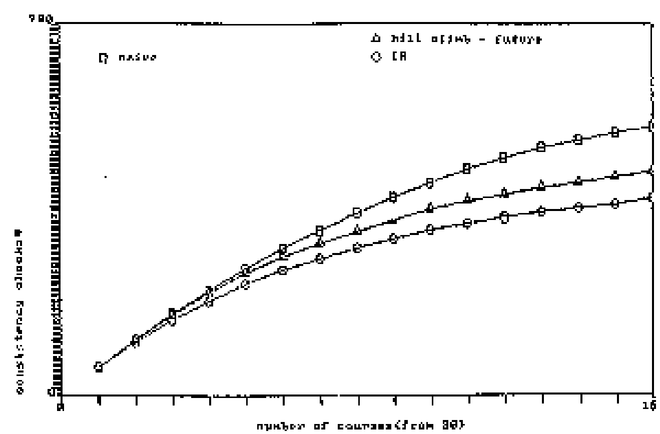


Fig. 7. Ordering algorithms – forward checking.

Wait a minute! 1990? Real problems?

Gaschnig
PhD thesis 1979
2nd last page

Figure 4.4.3-1 Dependence of mean number of pair-tests (T_f) on degree of constraint (L)
150 randomly generated SAPs of size N = k_i = 10 for each plotted point
1350 s.e. per algorithm, 5400 s.e. total
upper solid curve: BACKTRACK; middle: BACKJUMP; lower: BACKMARK
first solution

L	BACKTRACK	BACKMARK	BACKJUMP	DEEB
.000	100.000	100.000	100.000	100.000
.100	223.000	186.000	189.000	378.100
.200	452.000	289.000	373.000	1162.000
.300	985.000	524.000	794.000	3776.000
.400	2546.000	1046.000	1945.000	4568.000
.500	8710.000	2687.000	6259.000	7552.000
.600	35590.000	7791.000	22551.000	16360.000
.650	15893.000	3046.000	9836.000	
.700	9143.000	751.000	2346.000	3152.000
.800	352.000	137.000	171.000	2686.000
.900	67.400	67.200	67.400	2743.000
1.000	45.000	45.000	45.000	1305.000

My favourite! Gaschnig's random 10 queens

Gaschnig 1979

Log of search effort against constraint tightness

Algorithm independent phenomena

Figure 4.4.3-1 Dependence of mean number of pair-tests (T_i) on degree of constraint (L)
150 randomly generated SAPs of size $N = k_i = 10$ for each plotted point
1350 s.e. per algorithm, 5400 s.e. total
upper solid curve: BACKTRACK; middle: BACKJUMP; lower: BACKMARK
first solution

L	BACKTRACK	BACKMARK	BACKJUMP	DEEB
.000	100.000	100.000	100.000	100.000
.100	223.000	186.000	183.000	378.100
.200	452.000	299.000	373.000	1162.000
.300	985.000	524.000	794.000	3776.000
.400	2546.000	1046.000	1945.000	4568.000
.500	8718.000	2682.000	6259.000	7552.000
.600	35598.000	7791.000	22551.000	16360.000
.650	15893.000	3046.000	9536.000	
.700	9143.000	751.000	2346.000	3152.000
.800	352.000	137.000	171.600	2686.000
.900	67.400	67.200	67.400	2743.000
1.000	45.000	45.000	45.000	1305.000

Rotate to view!

Gaschnig's Thesis, page 179

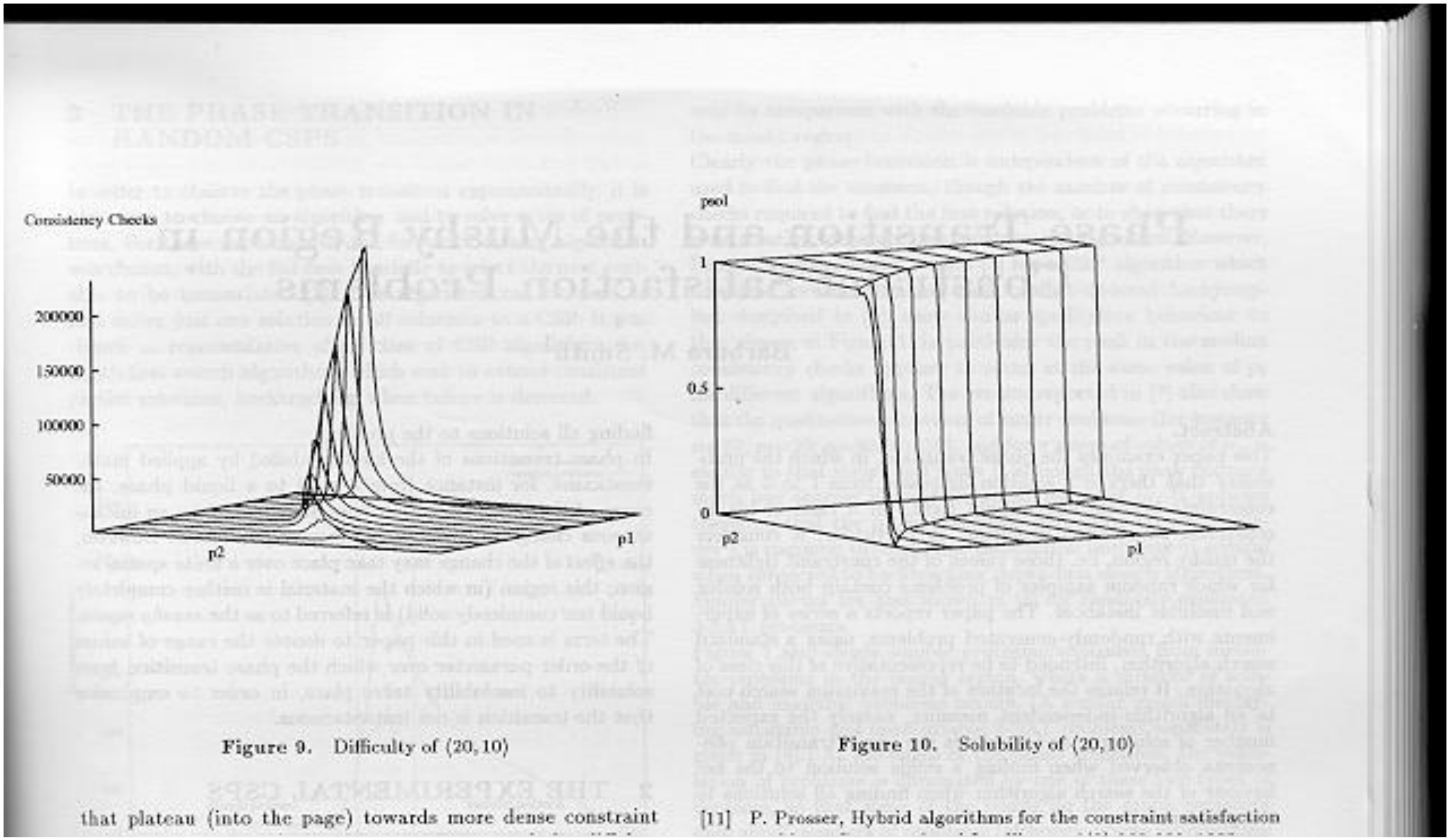
4.4.3 Cost as a Function of L: A sharp Peak at L = ~0.6

- Random CSP's <n,m,p1,p2>

  - n the number of variables
  - m domain size
  - p1 the probability of a constraint
    - between variables Vi and Vj
    - p2 probability Vi=x and Vj=y are in conflict

- <20,10,1.0,0>
  - easy soluble clique
- <20,10,1.0,1.0>
  - easy insoluble clique
- <20,10,1.0,0.2>
  - hard, phase transition, clique
- <20,10,0.5,0.37>
  - Drosophilia

ECAI94, random csp’s



1994, PT for CSP, show it exists, try and locate it (bms also at ECAI94)
And lunch with Barbara, Toby, and Ian

Frost and Dechter
AAAI94

K	N	C	C/N	C	C/N	C	C/N	C	C/N
		T = 1/9		T = 2/9		T = 3/9		T = 4/9	
3	25	199	7.96	89	3.56	51	2.04	31	1.24
3	30	236	7.87	104	3.47	59	1.97	36	1.20
3	35	272	7.77	120	3.43	68	1.94	41	1.17
3	40	310	7.75	137	3.43	76	1.90	45	1.13
3	50	380	7.60	166	3.32	91	1.82	53	1.06
3	60	454	7.57	196	3.27	106	1.77	62	1.03
3	75	565	7.53	244	3.25	132	1.76	74	0.99
3	100	747	7.47	317	3.17	169	1.69	92	0.92
3	125	927	7.42	394	3.15	207	1.66	109	0.87
3	150	1100	7.40	468	3.12	242	1.61	127	0.85
3	175	1290	7.37	546	3.12	281	1.61	146	0.83
3	200	1471	7.36	623	3.11	318	1.59	159	0.80
3	225			697	3.10	353	1.57	176	0.78
3	250			773	3.09	390	1.56	193	0.77
3	275			847	3.08	425	1.54	205	0.75
		T = 4/36		T = 8/36		T = 12/36		T = 16/36	
6	15	**	**	102	6.80	62	4.13	41	2.73
6	25	**	**	165	6.60	100	4.00	65	2.60
6	35	500	14.29	228	6.51	137	3.91	89	2.54
6	50	710	14.20	325	6.50	193	3.87	125	2.51
6	60	852	14.20	389	6.48	231	3.85	150	2.50
		T = 9/81		T = 18/81		T = 27/81		T = 36/81	
9	15	**	**	**	**	79	5.27	53	3.53
9	25	**	**	211	8.44	128	5.12	87	3.48
9	35	**	**	294	8.40	178	5.09	119	3.40

Figure 1: The "C" columns show values of C which empirically produce 50% solvable problems, using the model described in the text and the given values of N, K, and T. The "C/N" column shows the value from the "C" column to its left, divided by the current value for N. "**" indicates that at this setting of N, K and T, even the maximum possible value of C produced only satisfiable instances. A blank entry signifies that problems generated with these parameters were too large to run.

1994 again, Frost and Dechter tabulate, use this for comparison of algs (CKT's first goal!)

Arc-consistency and arc-consistency again 187

$K_{-2} = 0.600$

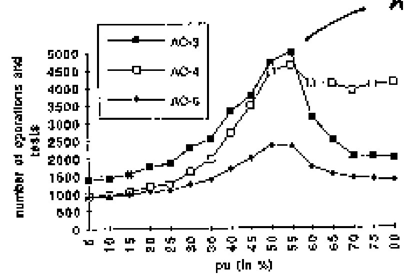
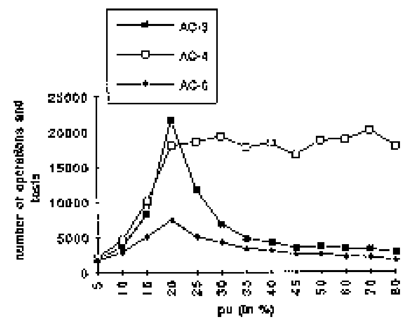


Fig. 4. AC-3, AC-4 and AC-6 on randomly generated CNs with 20 variables having 5 possible values, where the probability *pc* to have a constraint between two variables is 50%.



$K_2 = 0.73$

Fig. 5. AC-3, AC-4 and AC-6 on randomly generated CNs with 12 variables having 16 possible values, where the probability *pc* to have a constraint between two variables is 50%.

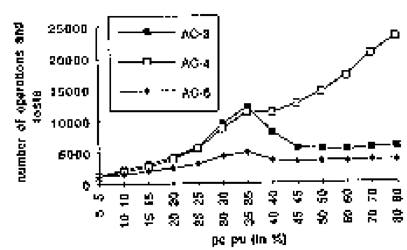


Fig. 6. AC-3, AC-4 and AC-6 on randomly generated CNs with 18 variables having 9 possible values.

— p_1 made fixed!

if $p_1 = 0.5$

then $K_2 = 0.953$

if $p_1 = 0.35$

then $K_2 = 0.67$

1994 again! A problem in P

Constrainedness

$$\kappa = 1 - \frac{\log_2(< Sol >)}{N}$$

<Sol> is expected number of solutions
N is log_2 of the size of the state space

---

k = 0, all states are solutions, easy, underconstrained

k = ∞ , <Sol> is zero, easy, overconstrained

k = 1, critically constrained, 50% solubility, hard

Applied to: CSP, TSP, 3-SAT, 3-COL, Partition, HC, …?

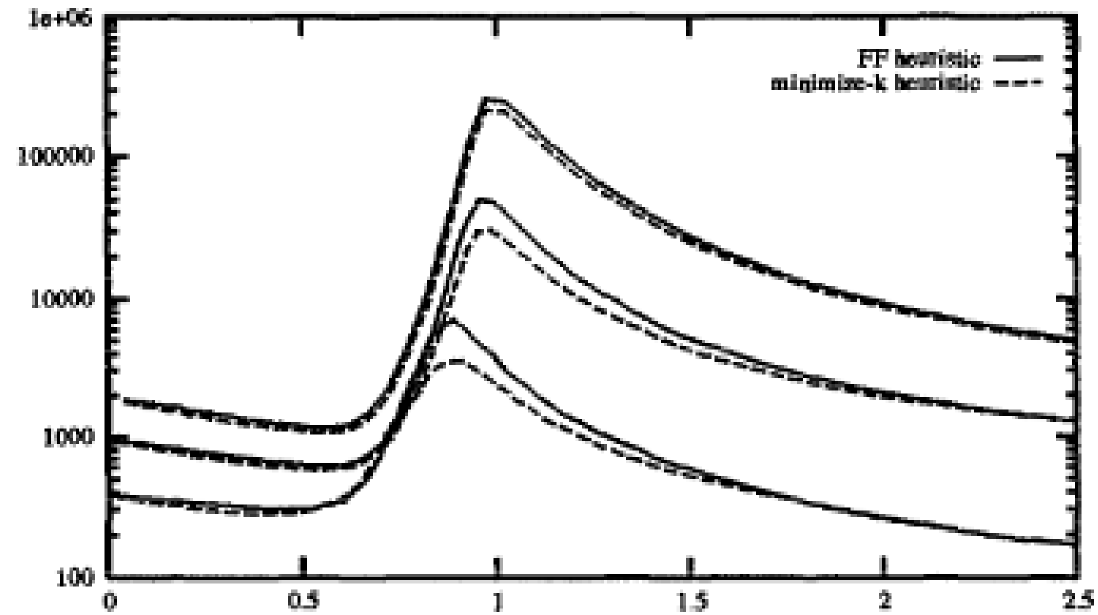
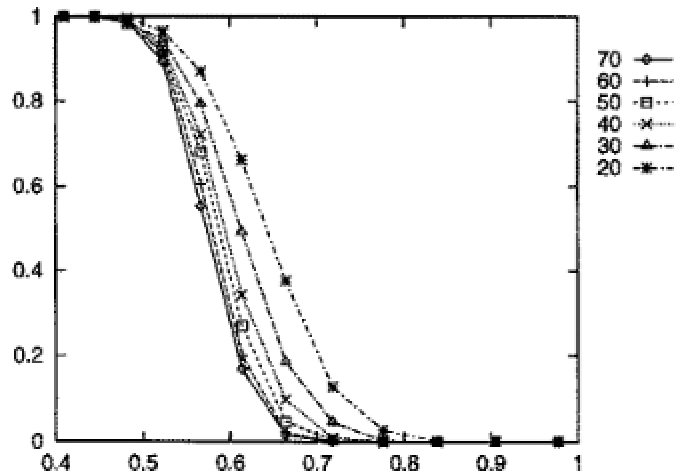


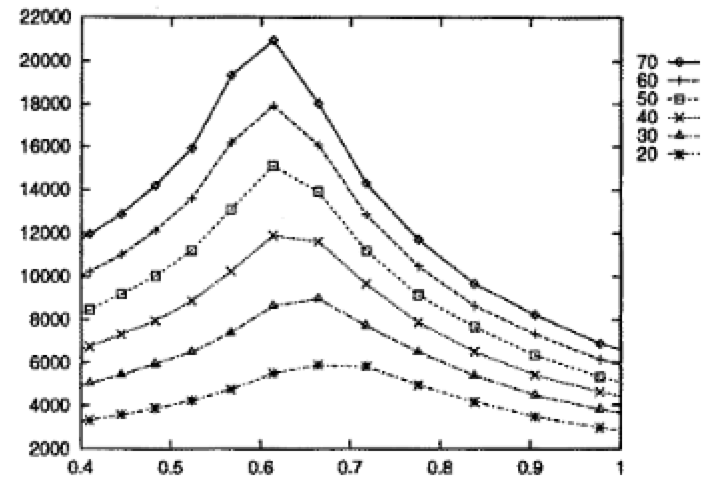
Figure 4. Fail First (FF) and minimize-κ heuristics applied to $\langle 20, 10, p_1 \rangle$ problems using fc-cbj. Mean search effort on y-axis, κ on x-axis. Contours for $p_1 = 1.0$ (top), $p_1 = 0.5$ (middle), $p_1 = 0.2$ (bottom).

$$\kappa = \frac{-\sum_{c \in C} \log_2(1 - p_c)}{\sum_{v \in V} \log_2(m_v)}$$

The Constrainedness of Arc Consistency*

Ian P. Gent, Ewan MacIntyre, Patrick Prosser, Paul Shaw, and Toby Walsh





$$\kappa_{ac} = \frac{-\sum_{c \in C} m_x \log_2(1 - {p_c}^{\frac{m_y}{2}}) + m_y \log_2(1 - {p_c}^{\frac{m_x}{2}})}{2 \sum_{v \in V} m_v}$$

- 1994
  - critical ratio of clauses to variables in 3SAT
- 1995
  - applied techniques from statistical mechanics to analysis
- 1996
  - Kappa, a theory of constrainedness
    - applies in CSP, 3-SAT NumPart, TSP!, ...
  - kappa based heuristics
  - P/NP phase transition (2+p)-SAT
    - At p ~0.4

- 1997
  - Kappa holds in P, achieving arc-consistency
  - Empirically derive complexity of AC3
  - Derive existing heuristics for revision ordering in AC3
- 1998
  - Expectation of better understanding of behaviour of algorithms and heuristic
  - What happens inside search?

- 1999
  - Kappa for QSAT
- 2000
  - the backbone
- 2001
  - backbone heuristics
- 2000 and beyond
  - Physics takes over?
  - New problems and richer behaviour
- 2019
  - Optimisation

Conclusion?

- More to it than just P and NP
- we are now learning about the structure of problems
- the behaviour of algorithms
- using this to solve the problems!

Where are the hard problems?