PART B. Fast Stochastic Methods

- After having identified hard instances, can we find better algorithms for solving them?
- Answer: Yes (at least for half of them...
Standard Procedures For SAT

• **Systematic search** for a satisfying assignment.

• Interesting situation:
  
  –Davis-Putnam (DP) procedure, proposed in 1960, is still the fastest complete method!

  –Backtrack-style procedure with unit propagation.
    
    SAT Competition 1992; DIMACS Challenge 1993 / 1994
    
    Essentially: Systematic exhaustive search! (but remember ---
    It’s an NP-complete problem!)
• DP provides very challenging benchmark for comparisons with other systematic (complete) procedures.

Not just on random formulas!

• Many other methods have been tried, e.g.,
  1) Backtracking with sophisticated heuristics
     (Purdom 1984; Zabih and McAllester 1988; Andre and Dubois 1993; Bhom 1992; Crawford and Auton 1993; Freeman 1993, etc.)
  2) Translations to integer programming
     (Jeroslow 1986; Hooker 1988; Karmarkar et al. 1992; Gu 1993)
3) Exploiting hidden structure
   (Stamm 1992; Larrabee 1991; Gallo and Urbani 1989; Boros et al. 1993)

4) Limited resolution at the backtrack nodes
   (Billionet and Sutter 1992; van Gelder and Tsuji 1993)

• And others!

Open Question: Why don’t they beat DP?

• Let’s try something completely different ...
**Randomized Greedy Local Search: GSAT**

Begin with a random truth assignment.

Flip the value assigned to the variable that yields greatest number of satisfied clauses.

Repeat until a model is found, or have performed specified maximum number of flips.

If model is still not found, repeat entire process, starting from different random assignment.

(Selman, Levesque, and Mitchell 1992)
How Well Does It Work?

- First intuition: Will get **stuck in local minimum**, with a few unsatisfied clauses.

- No use for **almost** satisfying assignments. E.g., a plan with a “magic” step is useless. Contrast with optimization problems.

- Surprise: It often finds **global** minimum! I.e., finds satisfying assignments.

- Inspired by local search for CSP initially used on N-Queens: **Heuristic Repair Method**. (Minton et al. 1991)
GSAT outperforms Davis-Putnam on, e.g.:

- **Hard random formulas**
  - DP: up to 400 vars; GSAT: 2000+ var formulas.

- **Boolean encodings of graph coloring problems.**
  - GSAT competitive with direct encodings.

- **Encodings of Boolean circuit synthesis and diagnosis problems.**
Mystery of very high dimensional search spaces
**Improvements Of Basic Local Search**

Issue: How to move more quickly to successively lower plateaus?

Idea: Introduce *uphill* moves ("noise") to escape from long plateaus (or true local minima).

Noise strategies:

- **a) Simulated Annealing**
  (Kirkpatrick et al. 1982)

- **b) Biased Random Walk**
  (Selman, Kautz, and Cohen 1993)
Simulated Annealing

- Noise model based on statistical mechanics.
- Pick a random variable
  \( \delta = \) change in number of unsatisfied clauses
  
  If \( \delta < 0 \) make flip (“downward”)
  
  else flip with probability \( e^{-\delta/T} \) (“upward”).

  Slowly decrease \( T \) from high temperature to near zero.
Random Walk

• Random walk SAT algorithm:

1) Pick random truth assignment.

2) Repeat until all clauses are satisfied:
   Flip random variable from unsatisfied clause.

• Solves 2SAT in $O(n^2)$ flips. (Papadimitriou 1992)
  (very elegant argument)

• Does not work for hard k-SAT (k >= 3).
**Biased Random Walk**

1) With probability $p$, “walk”, i.e., flip variable in some unsatisfied clause.

2) With probability $1-p$, “greedy move”, i.e., flip variable that yields greatest number of satisfied clauses.
# Experimental Results: Hard Random 3SAT

Biased Walk better than Sim. Ann. better than Basic GSAT better than DP.

<table>
<thead>
<tr>
<th>vars (vars)</th>
<th>basic time</th>
<th>basic eff.</th>
<th>walk time</th>
<th>walk eff.</th>
<th>Sim. Ann. time</th>
<th>Sim. Ann. eff.</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>.4</td>
<td>.12</td>
<td>.2</td>
<td>1.0</td>
<td>.6</td>
<td>.88</td>
</tr>
<tr>
<td>200</td>
<td>22</td>
<td>.01</td>
<td>4</td>
<td>.97</td>
<td>21</td>
<td>.86</td>
</tr>
<tr>
<td>400</td>
<td>122</td>
<td>.02</td>
<td>7</td>
<td>.95</td>
<td>75</td>
<td>.93</td>
</tr>
<tr>
<td>600</td>
<td>1471</td>
<td>.01</td>
<td>35</td>
<td>1.0</td>
<td>427</td>
<td>.3</td>
</tr>
<tr>
<td>800</td>
<td>*</td>
<td>*</td>
<td>286</td>
<td>.95</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>1000</td>
<td>*</td>
<td>*</td>
<td>1095</td>
<td>.85</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>2000</td>
<td>*</td>
<td>*</td>
<td>3255</td>
<td>.95</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>
Other Applications Of GSAT

- **VLSI circuit diagnosis**
  SAT formulation by Larrabee (1992)
  approx. 10,000 var 5,000 clause problems

- **Planning and scheduling**
  approx. 20,000 var 100,000 clause problems
  (Crawford and Baker 1994)

- **Finite algebra**
  search for algebraic structures
  GSAT+walk outperforms systematic method on large instances. Currently exploring remaining open problems.
  (Fujita et al. 1993)
For other work on stochastic, incomplete methods, see e.g.:

Adorf and Johnston 1990; Beringer et al. 1994; Davenport et al. 1994 (GENET); Kask and Dechter 1995; Ginsberg and McAllester 1994; Gu 1992; Hampson and Kibler 1993; Konolige 1994; Langley 1992; Minton et al. 1991; Morris 1993; Pinkas and Dechter 1993; Resende and Feo 1993; Spears 1995, and others!
• GSAT-style procedures are now a promising alternative to systematic methods.

• Drawback: cannot show unsatisfiability.
Showing UNSAT / Inconsistencies

Given the success of stochastic search methods on satisfiable instances, a natural question is:

Can we do something similar for unsatisfiable instances?
To show a set of clauses \( S \) unsatisfiable, we need to demonstrate ("prove") that none of the \( 2^N \) truth assignments satisfies \( S \).

This “truth-table” method is very time consuming.

Compare this with having to check a single satisfying assignment to verify the satisfiability of a formula.

*Can we do better? --- Surprisingly difficult!*
Length Of Proofs

- Best know improvement on truth tables: resolution
  - Resolve clauses until empty clause is reached.
  - Widely used in automated theorem proving.

- DP is a form of resolution.
Limitations Of Resolution

- **Method can’t “count”!** Pigeon-hole formulas: *Can’t place N+1 objects in N holes.*
  - Shortest resolution proof is exponentially long.
  - (Cook / Karp 1972; Haken 1985)

- **Random unsat formulas: exponential size proofs.**
  - Explains why we can’t push DP over 400 vars:
    - 400 vars requires search tree of about 10 million nodes
    - 1000 vars unsat requires $10^{15}$ nodes!
  - (Chvatal and Szemeredi 1988; Crawford 1995)
Stochastic Search For Proofs

- **GSAT**: start with random truth assignment (size linear in $N$), and try to “fix” it.

- **Proposal for UNSAT**: start with random proof structure, and try to fix it.

- Completely unfeasible if the structure that we’re fixing has trillions of nodes (exponential in $N$).

- **We need short proofs!** (O($N$) or something...) (Using abstractions / symmetries?)
Recap Of Results

A) Computationally hard problem instances

- Hardest ones are critically-constrained.
- Under- and over-constrained ones can be surprisingly easy.
- Critically-constrained instances at phase-transition boundaries.

Properties of transition can be analyzed with tools from statistical physics.
B) Stochastic Search Methods

- **GSAT**: Randomized local search for SAT testing. Viable alternative to systematic, complete methods.

- **Progress**:
  - 1991: 10 vars, 500 clause theories.
  - 2001: 100,000+ vars, up to 1,000,000+ clauses

- **Approaches size of practical applications.**
  E.g. in scheduling, planning, diagnosis, circuit design, and constraint-logic programming.
Impact And Future Directions

Fast Incomplete Methods

– Shift in Reasoning and Search from Systematic / Complete methods to Stochastic / Incomplete methods.

– Key issue: Better scaling properties.

– Analogy in OR: Shift from finding optimal to finding approximate solns.

– Also, little progress on heuristic guidance of complete methods. DP still rules…
Impact, Cont.

Message for KR&R

- Asymmetry between our ability to show \textit{satisfiability vs. unsatisfiability}, argues for \textit{model-finding} (show sat) over \textit{theorem proving} (show unsat).

- Examples:
  - Vivid repr. (Levesque 1985)
  - Planning (Kautz and Selman 1992)
  - Abduction / diagnosis / deduction
    - Model-based repr. versus formula-based repr. (Kautz, Kearns, and Selman 1994; Khardon and Roth 1994)
    - Case-based reasoning (Kolodner 1991)
Some Challenges

• Fast incomplete strategies for UNSAT (deduction)?
  Need for short proofs. Human proofs $O(N)$? Need automatic discovery of abstractions, symmetries, useful lemmas...

• Need for more model-based reformulations:
  Where solutions are compact structures --- allowing for randomized local search strategies.

• Can we syntactically characterize the class of instances solved by incomplete, stochastic methods?
  Running algorithm may be the best and only characterization!
Possible Limits Of Syntactic Characterization

Would suggest fundamental role for incomplete methods.