NP-completeness, computational complexity, and phase transitions: kSAT and Number Partitioning

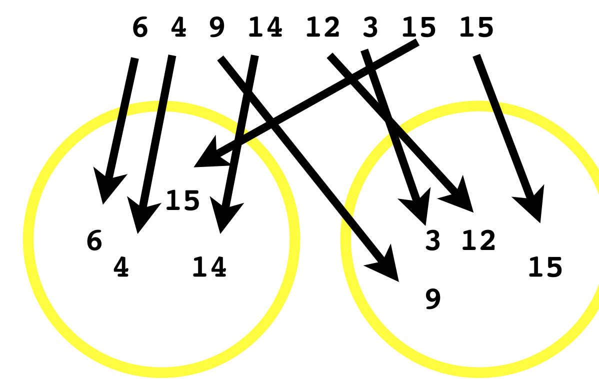
Phys 7682 / CIS 6229: Computational Methods for Nonlinear Systems

- Computational complexity
  - study of how resources required to solve a problem (e.g., CPU time, memory) scale with the size of the problem
  - e.g., polynomial time algorithm ($t \sim N \log N$, $t \sim N^2$) vs. exponential time algorithm ($t \sim 2^N$, $t \sim e^N$)
- Complexity classes
  - P: set of problems solvable in time polynomial in problem size on a deterministic sequential machine
  - NP (non-deterministic polynomial): set of problems for which a solution can be verified in polynomial time
  - NP-Complete: set of problems that are in NP, and are NP-hard (i.e., that every other problem in NP is reducible to it in polynomial time)
    - a polynomial time algorithm to solve one NP-complete problem would constitute a polynomial time algorithm to solve all of them
    - no known polynomial time algorithms for NP-complete problems
    - exponential runtimes consider worst case scenario; increasing interest in typical case complexity

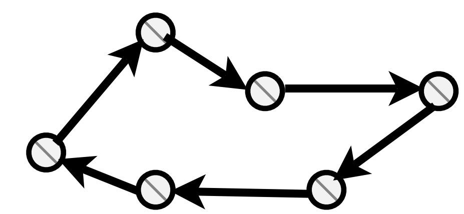
NP-complete problems

$(x_1 \vee x_2 \vee -x_4) \wedge$
$(x_2 \vee -x_3 \vee -x_5) \wedge$
$(x_3 \vee x_4 \vee x_5) \wedge$
...
$(x_4 \vee -x_8 \vee x_N)$

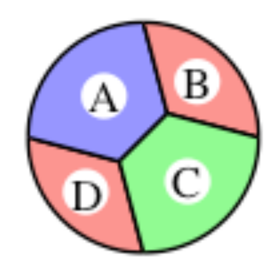
- Thousands of problems proven to be NP-complete (see, e.g., Garey and Johnson, *Computers and Intractability*, or Skiena, *The Algorithm Design Manual*)
    - typically phrased as "decision problems" with yes/no answer
- *Satisfiability (SAT)*: given a set U of boolean variables, and a set of clauses C over U, is there a satisfying truth assignment for C?
- *Partitioning*: given a finite set A and a size $s(a) \in Z^+$ for each $a \in A$, is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$?



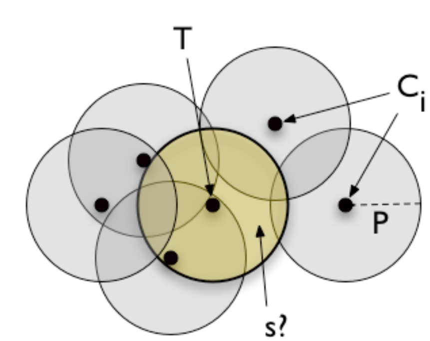
- *Traveling Salesman*: given a set C of m cities, distance $d(c_i,c_j) \in Z^+$ for each pair of cities $c_i,c_j \in C$, and a positive integer B, is there a tour of C having length B or less?



- *Graph K-colorability*: given a graph G=(V,E), and a positive integer $K \leq |V|$, is G K-colorable, i.e., does there exist a function $f: V \rightarrow \{1,2,\ldots,K\}$ such that $f(u) \neq f(v)$ whenever $\{u,v\} \in E$?



- *Sequence Niche*: given a sequence $T \in \{0,1\}^L$, a set of sequences $C_i \in \{0,1\}^L$ for $i=1,\ldots,N$ and a positive integer $P \leq L$, is there a sequence $s \in \{0,1\}^L$ such that $|s-T| \leq P$ and $|s-C_i| > P$ for all $i=1,\ldots,N$?



kSAT

- SAT (logical satisfiability)
  - given a set of logical clauses in conjunctive normal form (CNF) over a set of boolean variables, is there a variable assignment that satisfies all clauses?
- kSAT
  - restrict all clauses to length k
  - NP-complete for all $k \geq 3$
  - in P for $k = 2$
- 2^N possible assignments for N variables
  - exhaustive enumeration only an option for very small systems

$(x_1 \vee x_2 \vee -x_4) \wedge$
$(x_2 \vee -x_3 \vee -x_5) \wedge$
$(x_3 \vee x_4 \vee x_5) \wedge$
...
$(x_4 \vee -x_8 \vee x_N)$

M clauses

k variables per clause,
N variables total

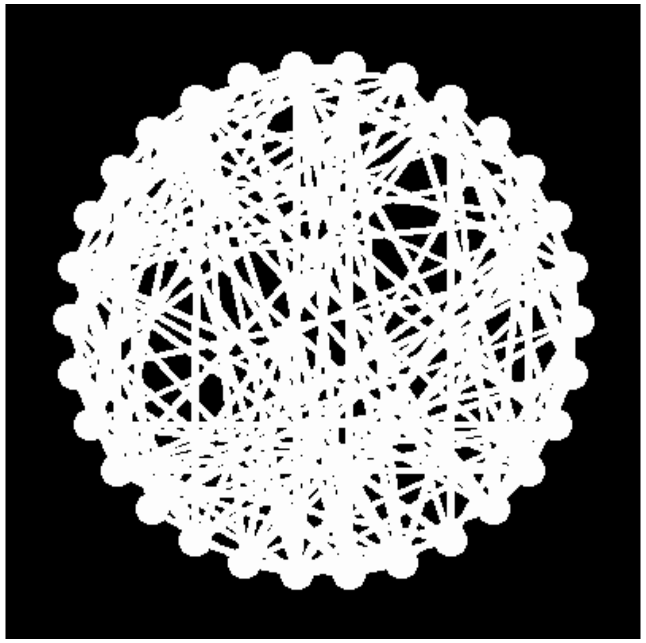
$\wedge$ = AND,
$\vee$ = OR,
$-$ = NOT,
x_i = True or False

Some algorithms for kSAT

- Davis-Putnam (+ modifications)
  - complete: can determine whether or not there is a solution for any instance
  - recursive: set a variable, eliminate resolved clauses, call itself on reduced problem
    - either assignment or contradiction is found
    - backtrack if contradiction is found
  - lots of heuristics (variable ordering, MOMS, random restarts) to prune the exponential search tree
- WalkSAT
  - randomly flips variables in unsatisfied clauses
  - incomplete: cannot determine that there is no solution
- Survey Propagation (SP)
  - based on "cavity method" developed to study the statistical mechanics of spin glasses
  - fast, complicated, and incomplete

$$(x_1 \vee x_2 \vee -x_4) \wedge$$
$$(x_2 \vee -x_3 \vee -x_5) \wedge$$
$$(x_3 \vee x_4 \vee x_5) \wedge$$
$$\ldots$$
$$(x_4 \vee -x_8 \vee x_N)$$

$\wedge$ = AND,
$\vee$ = OR,
- = NOT,
x_i = True or False



Phase transitions in random SAT problems

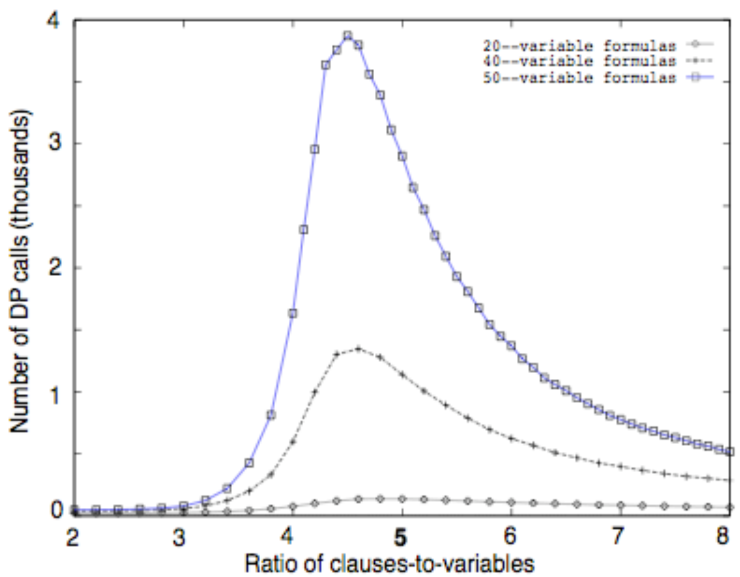
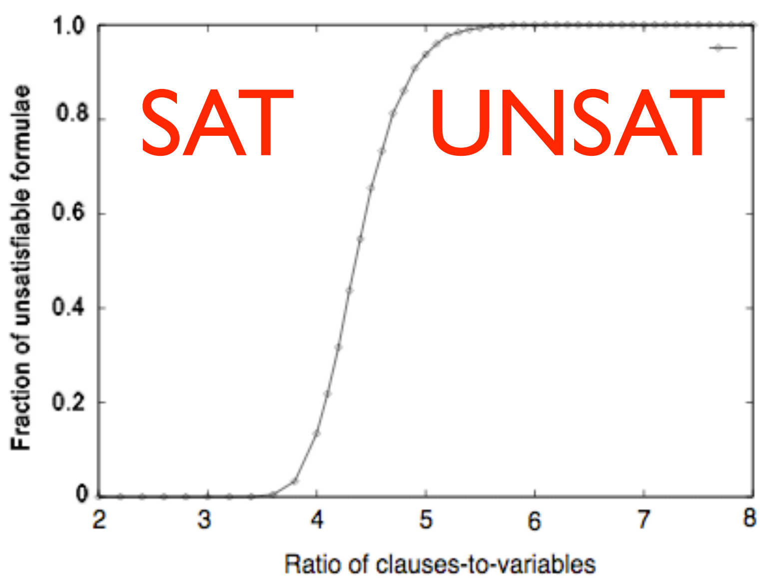


Figure 1.1: Solving 3SAT instances.

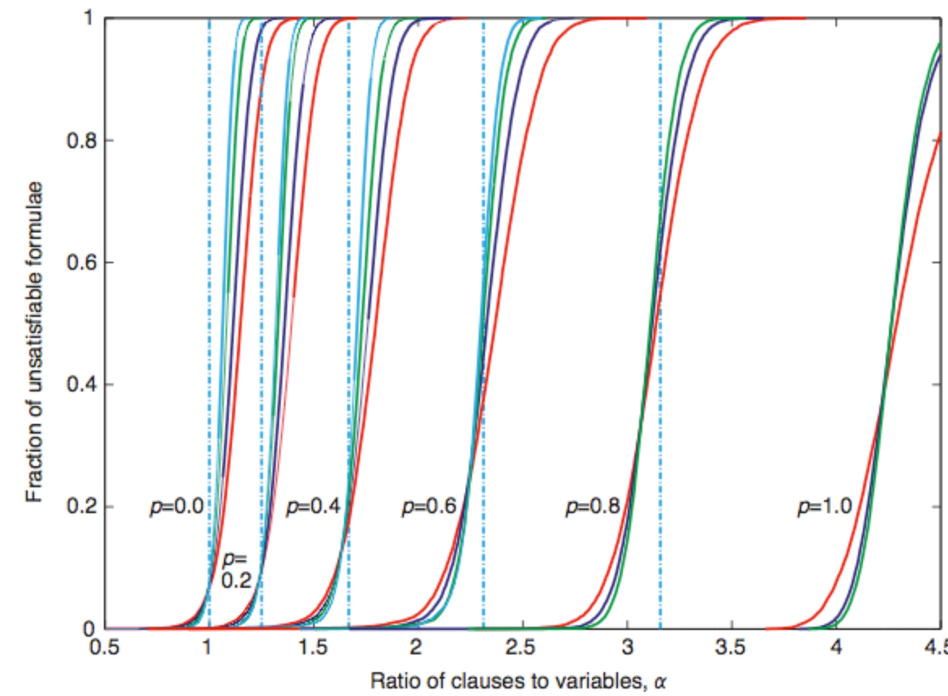


3-SAT

solving 3SAT problems gets hard near the SAT-UNSAT transition

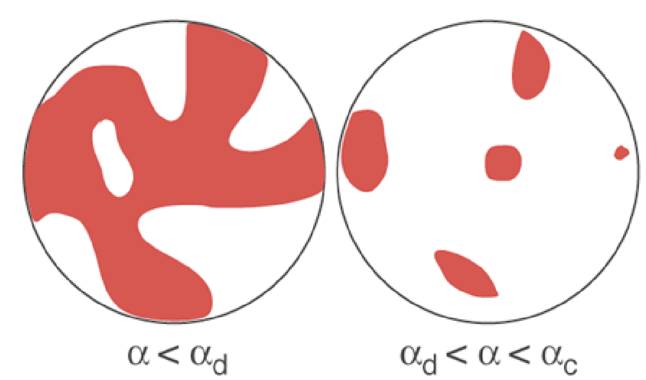
(# DP calls = # of recursive calls in Davis-Putnam algorithm)

Kirkpatrick and Selman (2001)



Monasson *et al.* (1999)

2+p-SAT



Mézard (2003)

fragmentation of solution space (hard SAT phase)