cs5050

04 22 14

(0)   + lower   bounds   " no better than "

Solve N P Complete problems

+ Simple   generate + test (recursive knapsack)

all possible solutions   Is this the actual Sol?
                                      pass the test
variable assignment
n variables 2^n
n node 3^n

(1) Give something up   + Solve Simpler Problem — Integer knapsack (polynomial)
                                                          ↳ sizes of knapsack
  + Correctness — wrong answer   and object is Int.

  + Completeness — Cannot solve all problems

  + Optimality — near optimal, not the "best"

                                                    $f(n)$ is the number of
                                                       call the alg. makes
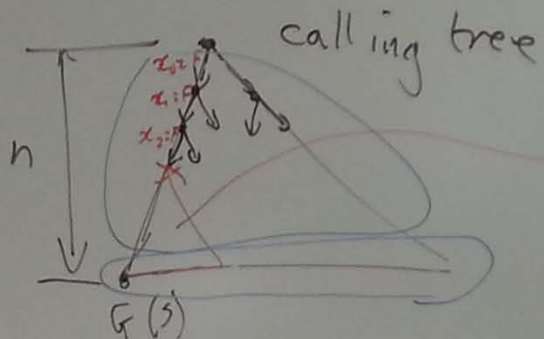 Solve (   )   for problem size n
      if nothing to do
         evaluate complete possible solution
                                                                      — my call
      else set variable to T $f(n) = 2f(n-1) + 1$
            Solve ( )
            set variable F
            solve

) P.S. A

call $G(s)$ 2^n times for SAT $f(1)=1$, $f(n)=2f(n-1)+1$

3^n 3 Color $f(1)=1$ $f(n)=3f(n-1)+1$

$$G(s) = (x_0^F + x_1^F + x_2^F) \wedge \cdots \cdots$$
$$68$$

and

calling tree

avoid $2^{(n-3)}$

n

x_2:
x_1:
x_2:

$G(s)$

time to solve

Simple

with early terminate

hard

all problems easy
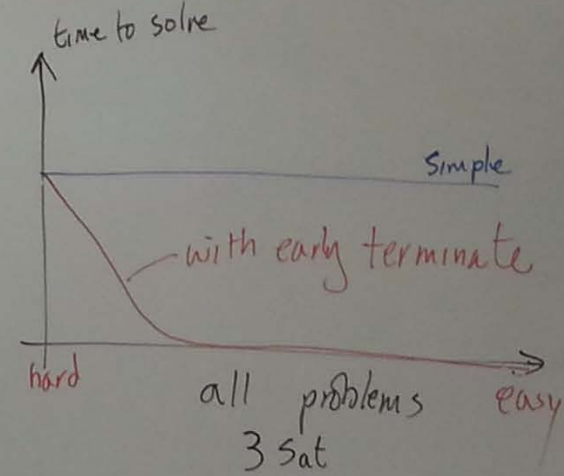
3 Sat

Complete   Randomness

SAT
    start   with random
        assignment of variables

IF fail try again
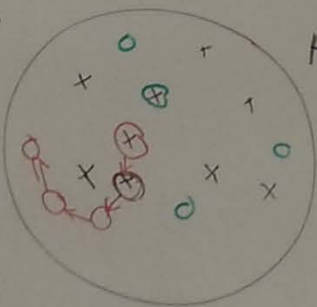
    Monte Carlo
                              Complete
2^n
                      All possible variable
                          assignment
n = 100

2^{100}

3SAT — Change evaluation
                              currently T or F
    clauses

(_) $\wedge$ (_) $\wedge$ (_) $\wedge$ (_) $\wedge$ (_) $\wedge$ (
    T   T   T   T F
Count how many are false

    + make random change here

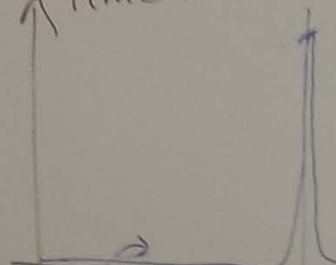   + pick changes that
        decrease # of clauses false

    Hill Climbing

Hard and Easy Problems   3SAT

time to solve   (4)

nothing works

easily False

easily true

anything works

$\dfrac{\# \text{ of clauses}}{\# \text{ of variables}}$ increase — more constrained