15-252 Lecture 14

(Fast) Exponential Time Algorithms

3-SAT is NP-complete $\Rightarrow$ Don't expect polynomial time algorithm

ETH (Exponential Time Hypothesis)

3-SAT requires $2^{\alpha n}$ time for some $\alpha > 0$

(No $2^{n/\log n}$ time algorithm)

Naive algo (brute force): $2^n \mathrm{poly}(n)$ time
($n = \#$ vars)

Still interesting (and important) to beat brute force
- 2^n and $2^{n/2}$ are quite different
- Interesting algorithmic ideas

"Fine grained complexity"
- P vs NP (P vs Exp time)
  Coarse distinction

Ultimate dream
- Alg with runtime c^n $(c > 1)$
- "Hard" to solve in $(c-\varepsilon)^n$ time

We are <u>very far</u> from such a picture

- SAT (CNF Satisfiability with unbounded width clauses) can't be solved in $(2-\varepsilon)^n$ time for any $\varepsilon > 0$

Strong Exponential Time Hypothesis (SETH)

Today's lecture: Some algorithms for 3-SAT

Trivial: $O(m 2^n)$ time $\quad n = \#\text{vars}$

$m = \#\text{clauses}$

$(x_1 \vee x_2 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1 \vee x_7) \wedge (x_9 \vee x_3 \vee \bar{x}_1) \wedge \ldots$

Local search Suppose we knew an assignment A that is close to a satisfying assignment.

(in Hamming distance)

A & A^* differ in r variables

satisfying

Don't know A^*

① Start with assignment A

② While $\exists$ at least one unsatisfied clause in A and haven't looped for $> r$ steps

(A): $x \vee y \vee z$

a) Pick an arbitrary unsatisfied clause $x \vee y \vee z$

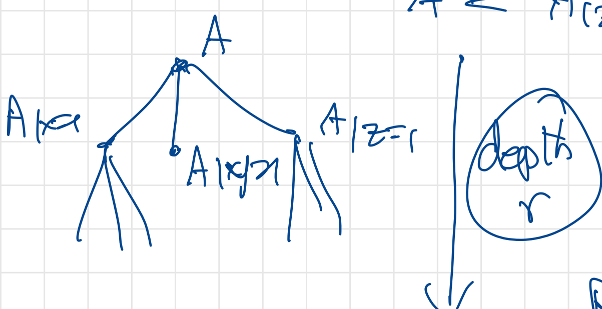
(A*) $x \vee y \vee z$ is true

b) Branch on each of the vars x, y, z

$A \leftarrow A_{|x=1}$

$A \leftarrow A_{|y=1}$

$A \leftarrow A_{|z=1}$

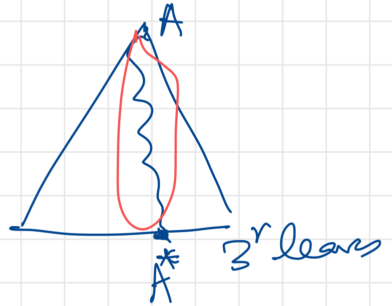


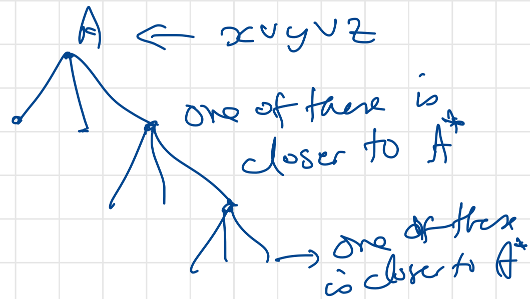
At each node of tree, check if assignment is satisfying & if so halt

Runtime. $3^r \text{poly}(n)$

<u>Claim</u>: If algo didn't terminate before depth r, then one of the leaves will be A^*.

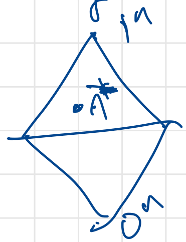
Pf.





How to pick starting assignment A?

Try $A = 0^n$ and $A = 1^n$



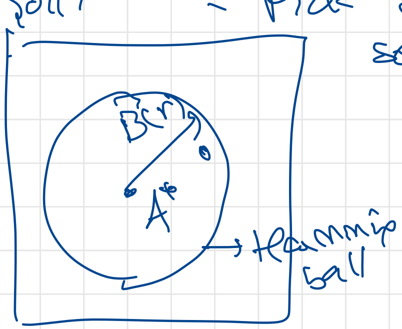
One of those is within Ham Dist $\frac{n}{2}$ of A^*

<u>Cor</u>: $3^{n/2}$ poly(n) time algo.

$\to (1.73)^n$

Randomized variant

- Pick several random starts A and search within radius r $\to$ 3^r time



$$\frac{|B(r)|}{2^n} = p = \text{prob}$$

that random A is within r of A^*

$$|B(r)| = \sum_{i=0}^{r} \binom{n}{i}$$

$$\#\text{ trials} \approx \frac{1}{p} = \frac{2^n}{|B(r)|}$$

$$\underbrace{\left(n \cdot \frac{2^n}{\binom{n}{r}} \right)}_{\geq \#\text{ trials}} \cdot \underbrace{3^r}_{\text{Runtime of each trial}}$$

Optimize in r

$r = \frac{n}{4}$ is best choice

Randomized Runtime : $(1.5)^n \text{poly}(n)$

Random walk algorithm (Schöning 1999)

Fact: Rand. algo (polynomial) that succeeds with prob p $\left(= \left(\frac{2}{3}\right)^n\right)$

$\Rightarrow$ Rand algo of runtime $\frac{\text{poly}(n)}{p}$ that succeeds with prob $1 - 2^{-n}$

---

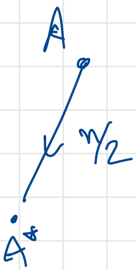
1. Pick a random initial assignment A
2. While there is at least one unsatisfied clause in A & haven't run for ~~n~~ $3n$ steps already
   (a) Pick an arbitrary unsatisfied clause
   (b) Flip the value of a random var. of that clause

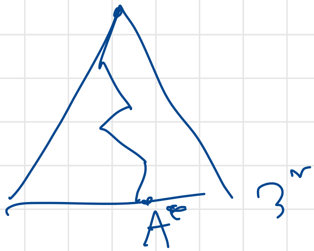
Analysis #1 Fix some A^* that is satisfying

E = event that A & A^* agree on $\geq \frac{n}{2}$ vars

$\Pr[E] \geq 1/2$

$\Pr[\text{algo succeeds} \mid E] \geq \left(\frac{1}{3}\right)^{n/2}$





$\Pr[\text{algo succeeds}] \geq \frac{1}{2} \cdot \left(\frac{1}{3}\right)^{n/2}$

$\Rightarrow 3^{n/2} \text{poly}(n)$ alg..

Better analysis of same algo:

$$\Pr[\text{algo succeeds}] \geq \sum_{k=0}^{n} \underbrace{\binom{n}{k} 2^{-n}}_{\Pr[\text{dist}(A, A^*) = k]} \left(\frac{1}{3}\right)^k$$

$$= 2^{-n} \sum_{k=0}^{n} \binom{n}{k} \left(\frac{1}{3}\right)^k$$

$$= 2^{-n} \left(1 + \frac{1}{3}\right)^n = 2^{-n} \left(\frac{4}{3}\right)^n$$

$$= \left(\frac{2}{3}\right)^n$$

$\Rightarrow$ Algo with runtime $\left(\frac{3}{2}\right)^n \text{poly}(n)$

An improved algorithm

Small change: Run loop for $3n$ steps instead of n

Analysis: Instead of a beeline from A to A^* analyze the chance of making at most k incorrect steps within first $3k$ steps

$$\Pr[\text{algo succeeds}] \geq \sum_{k=0}^{n} \binom{n}{k} 2^{-n} \binom{3k}{k} \left(\frac{2}{3}\right)^{k} \left(\frac{1}{3}\right)^{2k}$$

Stirling's formula

Compute this

$$\geq \left(\frac{3}{4}\right)^n \frac{1}{100\sqrt{n}}$$

Gives $(4/3)^n \, \text{poly}(n)$ time algo

Almost the best known runtime which is $\approx (1.31)^n$