CS124 Lecture 16

Randomness

2SAT (at most two literals in a clause)

$(x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (x_1 \vee x_2) \wedge (x_4 \vee \bar{x}_3) \wedge (x_4 \vee \bar{x}_1)$

T F F T
$x_1\ x_2\ x_3\ x_4$

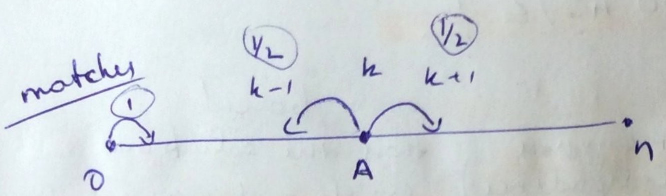
Randomized alg, $O(n^2)$.

start with any truth assg.
if satisfied, done!
if not satisfied, take any unsatisfied clause
flip a coin, flip a variable
repeat until bored
↳ unsatisfiable

assume the formula is satisfiable
solution S - truth assg.
current truth assg. A.
\# matches between S and A ← closeness measure
= \# of variables → done!

worst case assumptions
1) single solution
2) each can correctly change 1 variable

worst case:
pick an unsatisfied clause
↳ at least one of vars needs to change.
maybe both, but in the worst case only one



reduce worst case to random walk analysis

⇒ in the worst case, with 1/2 prob. \# of matches increases and with 1/2 prob. \# of matches decreases

⇒ \# of steps to get to n?

$T(i)$ = expected time to get to n from i

$T(n) = 0$ recurrence
$T(0) = 1 + T(1)$
$T(i) = \frac{1}{2} T(i-1) + \frac{1}{2} T(i+1) + 1$

approach: solve for specific n, see pattern

soln $T(i) = n^2 - i^2$

⇒ regardless of start $\leq n^2$ expected steps

$n = 3$

$T(0) = T(1) + 1$

$T(1) = \frac{1}{2} T(0) + \frac{1}{2} T(2) + 1$

$T(2) = \frac{1}{2} T(1) + \frac{1}{2} \cdot 0 + 1$

$T(0) = 9$

$T(1) = 8$

$T(2) = 5$

pattern

$T(0) = n^2$

$T(1) = n^2 - 1$

$T(2) = n^2 - 4$

[redacted] ;

$\boxed{T(i) = n^2 - i^2}$

check recurrence solution

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

$$= n^2 - i^2 \checkmark$$

Markov's inequality

non-negative random quantity X

$X \quad E[X]$

$$Pr\left(X \geq k\, E[X]\right) \leq \frac{1}{k}$$

e.g. $k = 2$

prob X is $\geq$ twice the average is $\leq \frac{1}{2}$

[redacted]

Expected time from $i = 0$

n^2

$$Pr(\underbrace{\text{steps to solution}}_{\text{fail when soln exists}} \geq 100\, n^2) \leq \frac{1}{100}$$

prob. bound that the solution exists and was not found

improve prob. bound (recognize independence)

After $2n^2$ steps

Pr (fail when solution exists) $\leq \frac{1}{2}$

indep.

restart (alg. can start anywhere)

Pr (fail after next $2n^2$ steps, when solution exists) $\leq \frac{1}{2}$

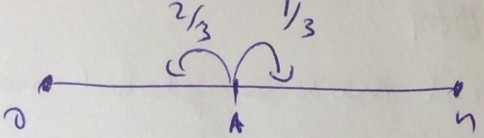
;

after $100 n^2$ steps

Pr (fail after all $2n^2$ step trials) $\leq 2^{-50}$

→ use the same idea for 3SAT (NP-Complete) etc.—

3SAT $O\left(\left(\frac{4}{3}\right)^n poly(n)\right)$



Linear Programming

profit

100 ← x_1 : # of product 1

600 ← x_2 : # of product 2

1400 ← x_3 : # of product 3

constraints:

$x_1, x_2, x_3 \geq 0$

$x_1 \leq 200$

$x_2 \leq 300$

$x_1 + x_2 + x_3 \leq 400$

$x_2 + 3x_3 \leq 600$

goal:

max $100x_1 + 600x_2 + 1400x_3$

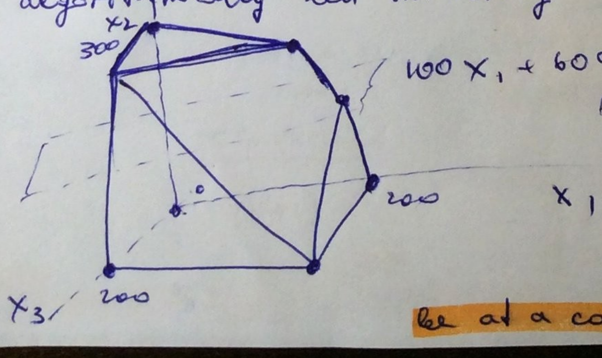
$x_2 = 300$, $x_3 = 100$, $x_1 = 0$ profit: 320 000

linear program:
- linear constraints
- linear objective function

LP's are solvable, provably poly-time algs
↑ not used in practice

Simplex alg. widely used, but all known "basic" versions are exponential time

↳ algorithmically less interesting



$100x_1 + 600x_2 + 1400x_3 = y$

↑ plane, move parallel versions to max profit

↳ maximum will be at a corner of a box

greedy alg.

Simplex

- start at a corner of the box
- look locally for a better corner until cannot find a better corner

↳ OPT

local max = global max

Many problems can be formulated as LPs but LP is often not an efficient way to solve

↳ **LP often used as baseline and solvability proof**

Integer LP : NP-hard

not clear how simplex would work