



Algorithms: Design
and Analysis, Part II

Local Search

Analysis of
Papadimitriou's Algorithm

Papadimitriou's Algorithm

n = number of variables

Repeat $\log_2 n$ times:

- Choose random initial assignment
- Repeat $2n^2$ times:
 - If current assignment satisfies all clauses, halt + report this
 - Else, pick arbitrary unsatisfied clause and flip the value of one of its variables [choose between the two uniformly at random]

Report “unsatisfiable”

Obvious good points:

- (1) Runs in polynomial time
- (2) Always correct on unsatisfiable instances

Satisfiable Instances

Theorem: For a satisfiable 2-SAT instance with n variables, Papadimitriou's algorithm produces a satisfying assignment with probability $\geq 1 - \frac{1}{n}$.

Proof: First focus on a single iteration of the outer for loop.

Fix an arbitrary satisfying assignment a^* .

Let a_t = algorithm's assignment after inner iteration t
($t = 0, 1, \dots, 2n^2$) [a random variable]

Let X_t = number of variables on which a_t and a^* agree.
($X_t \in \{0, 1, \dots, n\}$)

Note: If $X_t = n$, algorithm halts with satisfying assignment a^* .

Proof of Theorem (con'd)

Key point: Suppose a_t not a satisfying assignment and algorithm picks unsatisfied clause with variables x_i, x_j .

Note: Since a^* is satisfying, it makes a different assignment than x_i or x_j (or both).

Consequence of algorithm's random variable flip:

(1) If a^* and a_t differ on both x_i & x_j , then $X_{t+1} = X_t + 1$ (100% probability)

(2) If a^* and a_t differ on exactly one of x_i, x_j , then

$$X_{t+1} = \begin{cases} X_t + 1 & (50\% \text{ probability}) \\ X_t - 1 & (50\% \text{ probability}) \end{cases}$$

Quiz: Connection to Random Walks

Question: The random variables $X_0, X_1, \dots, X_{2n^2}$ behave just like a random walk of the nonnegative integers except that:

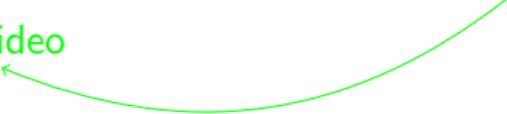


- A) Sometimes move right with 100% probability (instead of 50%)
- B) Might have $X_0 > 0$ instead of $X_0 = 0$
- C) Might stop early, before $X_t = n$
- D) All of the above

Completing the Proof

Consequence: Probability that a single iteration of the outer for loop finds a satisfying assignment is $\geq \Pr[T_n \leq 2n^2] \geq 1/2$

from last video



Thus:

$$\begin{aligned}\Pr[\text{algorithm fails}] &\leq \Pr[\text{all } \log_2 n \text{ independent trials fail}] \\ &\leq \left(\frac{1}{2}\right)^{\log_2 n} \\ &= \frac{1}{n}. \quad \text{QED!}\end{aligned}$$